Learning from weak representations using distance functions and generative models

Thesis for the degree of

DOCTOR of PHILOSOPHY

by

Aharon Bar-Hillel

SUBMITTED TO THE SENATE OF

THE HEBREW UNIVERSITY OF JERUSALEM

October 2006

This work was carried out under the supervision of Prof. Daphna Weinshall

Acknowledgements

To be inserted.

Abstract

This thesis discusses two different problem domains, in which learning takes place using a weak initial representation and partial supervision. The first is distance function learning from data augmented with equivalence constraints, which are constraints stating whether two points come from the same or a different source. The second problem is learning recognition of visual object class categories, where the natural image representation is an unordered set of patch descriptors. A short introduction describes the similarities between the two problems, which can be viewed as instances of representation learning. Then the two problems are discussed in two separate chapters:

Chapter 2 - Learning with equivalence constraints: We consider distance function learning as learning of a classifier defined over point pairs, and investigate its theoretical relations to the multi-class learning problem. Specifically we show that the two problems are equivalent in terms of learnability, and that a good solution for the former (a good distance function) leads to a solution of the latter. We then present two methods for parametric distance function learning from positive equivalence constraints. The first algorithm, termed RCA, learns a Mahalanobis metric and its optimality for this parametric family is proven under several criteria. The second method, termed coding similarity, is derived based on informational-theoretic considerations, and it leads to a non-Mahalanobis parametric form. This similarity is analytically shown to have deep connections with the Fisher Linear Discriminant(FLD) and RCA, and it has an empirical advantage over Mahalanobis metrics in several applications.

Chapter 3 - Learning object class recognition: Unlike traditional learning from ordered vectors, this problem is naturally posed as learning from sets of features with internal relations. We suggest an approach which combines a relational generative, part-based object model with a discriminative, boosting-based optimization method. The learning complexity is linear in the number of model parts and image features, compared to the exponential complexity of traditional methods for relational model learning. The improved efficiency allows scalable learning of relational models with many parts. Based on this algorithm, we then suggest a two stage method for the recognition of subordinate classes (e.g. cross motorcycles or dining tables). In the first stage a model of the basic category is learned, and it is used to form a part-based vector representation for images. Classification is then done by applying SVM to the new representation. We show that this method allows inter-task knowledge transfer and that it outperforms simpler methods which do not take advantage of the similarity between subordinate categories.

This thesis is based on the following publications:

Chapter 2

[A] A. Bar-Hillel and D. Weinshall: “Learning with equivalence constraints, and the relation to multiclass Classification ”, in *the Sixteenth Annual Conference On Learning Theory (COLT)*, 640-654, Springer 2003.

[B] A. Bar-Hillel, T. Hertz, N. Shental and D. Weinshall: “Learning a Mahalanobis metric from equivalence constraints”, in *Journal of Machine Learning Research* 6(Jun): 937-965, 2005.

Chapter 3

[C] A. Bar-Hillel, T. Hertz and D. Weinshall: “Object class recognition by boosting a part based model”, in *Conference on Computer Vision and Pattern Recognition (CVPR)*, volume I, 702-709, 2005.

[D] A. Bar-Hillel, T. Hertz and D. Weinshall: “Efficient learning of relational object class models”, in *International Conference of Computer Vision (ICCV)*, volume II, 1762-1769, 2005.

[E] A. Bar-Hillel and D. Weinshall:“Subordinate class recognition using relational object models”, accepted for publication in *Advances in Neural Information Processing Systems (NIPS)*, 2006.

It also includes the technical report:

[F] A. Bar-Hillel and D. Weinshall: “Learning distance function by coding similarity”, Technical report, 2006.

Contents

Chapter 1

Introduction

This thesis presents work done in two seemingly unrelated research domains: learning distance functions from equivalence constraints, and learning object models from unsegmented images. However, my interest in these research domains derives from the same source; that is, my deep conviction about the role of representation learning. Both tasks can be seen from a wider perspective as instances of learning an intermediate means (distance function/object model) of clustering or classification, using partial supervision (equivalence constraints/unsegmented images). Similar issues arise in both domains, as knowledge transfer between related learning tasks and the division of labor between representation and prediction learning.

I start the introduction by presenting and motivating the general notion of representation learning, which is the unifying aspect of the work presented here, in section 1.1. I then continue with more specific introductions for the two chapters of the thesis. Section 1.2 starts with a discussion of the distance, metric and similarity concepts, and their relations with representation, feature synthesis and classification. I then survey the relevant literature on distance function learning. In section 1.3 I discuss the representation problem in the object recognition domain. Here the gap between the initial representation and a useful one is especially pronounced, and the debate regarding proper representation is at the core of contemporary research. I briefly present the problem and the relevant literature, with emphasis on the research areas to which my work belongs.

1.1 Learning from a weak representation

The notions of 'weak representation' and 'Representation leaning', while discussed in the literature [12,31,135], are relatively vague, and do not have an agreed formal definition. I first outline the rationale for the introduction of these concepts in section 1.1.1. Then, in section 1.1.2 I characterize representation learning as applying to both distance function and input-data transformation learning. In 1.1.3 I consider the conditions under which

representation learning should be separated from classifier learning. Section 1.1.4 provides a short summary of research included in the next chapters, with an emphasis on the 'representation learning' perspective.

1.1.1 Motivation

Several observations prompted my interest in representation learning. The first, basic one is that in my opinion, the traditional supervised learning problem is to a large extent solved. In saying that, I refer to a binary classification problem with a large labeled training set and a reasonable data representation. Learning tools developed over the last few decades, specifically SVMs and boosting algorithms, are usually able to solve such problems efficiently and accurately. Specifically, SVMs [118, 144] combine very good empirical performance with a well developed theory, including a large margin aspect which provides generalization guaranties. A combination of empirical success and margin-based generalization bounds have also been documented for boosting, or combined classifiers in general [10].

However, in all but the simplest real world problems, the human machine learning expert has to devise a proper representation when solving a new problem. Preparing the data for learning often requires complex processing, chosen in a trial-and-error process by the human expert. Without such a representation choice step, classification results are at best mediocre. The intuition here is that a large portion of the 'learning' is actually done by the human designer, and not by the machine. From an engineering perspective, automating this second order representation learning is the next desirable step in the automation of learning.

Another source (and for me, the most profound) of inspiration for representation learning is the human-machine performance comparison. Current machine classifiers may outperform human for problems with large data sets and many features. Humans are clearly superior in learning problems which have some similarity to problems already encountered by them, and with learning from a small sample. This generalization between related tasks implies that humans pervasively use second order learning and representation learning. Such issues, also termed in the literature 'learning to learn' [135], 'inductive transfer' [31] or 'learning with points sets' [102], are attracting growing attention in the machine learning and computer vision communities both from the theoretical [13, 16] and the practical [53, 56, 101] points of view. Thus the tasks of understanding the human and improving the machines can both benefit from research on these lines.

1.1.2 Representation Learning

Clustering and classification algorithms take a description of a set of data instances as input. In most cases, each data instance is described using an ordered feature vector (for example, this is the input for decision trees [114]

or k-means clustering [50]). Alternatively, some algorithms accept as input a matrix of distances or similarities between the input data instances. Examples are graph based clustering algorithms, a wide family of algorithms including agglomerative [50], spectral [128] and stochastic [24, 61] formulations, or the Support Vector Machine which requires only the Gram matrix for learning. Formally, the task of a learning algorithm A is to find a mapping $f : X \to Y$ from data instances in X to a finite label set $Y = \{1, 2, ..M\}$. The mapping is defined on the whole data domain for classification, and on the data instances alone for clustering.

Successful application of a learning algorithm typically relies on several underlying assumptions. Some examples are: Classes should preferably be connected and convex, or at least not too fragmented. Vector entrees with the same index in different instances should be measurements of 'the same quantity', i.e. should have the same relation w.r.t to the prediction label (this is the basis of the 'feature' notion). A reasonable portion of the features should be relevant to the prediction label. For distance based algorithms, the distance between two instances should in general decrease with the likelihood of their belonging to the same cluster. When some of these assumptions fail, we say that the representation is 'weak'.

A weak representation can be improved by changing the input given to the learning algorithm. One possibility is to apply a pre-processing transformation T to the data instances before these are subjected to the learning algorithm. In the case of classification, the learned classifier f can then be applied to the original domain via the composition $f \cdot T(x)$. For distance based algorithms a second possibility exists; namely, altering the distance function used. While distance functions operate on pairs of points and not on single points like a data transformation, the functional role they play as an input modifier is similar. Again, a classifier can be applied to the data using a composition $f \cdot d(x) \stackrel{\Delta}{=} f(d(x_1, x), .., d(x_n, x))$. In both cases it is well known empirically that classifier performance critically depends on the transformation/distance function used. Most traditional algorithms are very sensitive to the data representation, while SVM and graph-based clustering rely heavily on the quality of the kernel/distance function used. The parallelism between distance and transformation learning can be made formal for certain restricted distance families, which are formally equivalent to specific transformation families. This is the case with kernels in general and specifically with Mahalanobis metrics, which can be equated with linear data transformations.

Based on this functional similarity, I use the term 'representation learning' to denote both distance function and transformation learning. Such learning aims to reduce the human role in the choice of data pre-processing, and can be regarded as 'pre-process' learning. In both cases of transformation and distance learning, it is formally easy to see that if the family of considered representations is not limited, then choosing an optimal, ideal representation makes classifier learning redundant. For example, one may learn a representation transformation which sends each

data instance to its label. A similar construction is possible for distance functions, and we discuss the theoretical connections between classifier and distance function learning extensively in paper [A] in this thesis. Of course, in practical cases representation learning is limited to a predefined family of hypotheses. The fuzzy borderline between the two, however, raises a question regarding the need for two separate learning stages (i.e. representation and clustering/classification) and the division of labor between them.

1.1.3 Do we need a separate stage of representation learning?

The notion of two sequential learning stages, i.e. representation learning followed by clustering/ classification (I use the term 'prediction' for clustering/classification in what follows), immediately raises some difficulties. One problem is the choice of optimization goal for the representation learning stage. In classification the learning task is well defined: learn a classifier with minimal generalization error. The optimization argument can be defined with clear relation to this error. Usually some smooth loss function of the training error (that is, a statistical estimator of the generalization error) is used, possibly with a regularization term. This is the discriminative choice, used for example by boosting and SVMs, and the minimization of such a directly relevant loss function is an important ingredient in the power of these methods. In contrast, representation learning learns only an intermediate construct, and the utility of a specific representation for the final classification is hard to predict a-priori. Hence it is unclear how to define a good optimization argument, and the problem, like clustering, is an ill-posed problem.

A related claim may be stated as follows: Assume that our goal is to find a classifier $h : X \to Y$ which minimizes some loss $L_1(h)$ over $h \in H$. In a two stage approach we first choose a representation (transformation or distance function) $g \in G$ by minimizing some loss $L_2(g)$ and then learn a classifier on the transformed input, so the final classifier is a composition $h = f \cdot g$. Compare this to a single stage approach, in which we choose $h \in H = F \cdot G$ which minimize $L_1(h)$ directly. Assume that the learning algorithms are able to find the optimal functions f^*, g^*, h^* which achieve the minimal losses. In this case we can see that a direct single stage approach is equal or better than a two stage one:

$$L_1(f^*) = \min_{f \in F} L_1(f \cdot g^*) \geq \min_{f \in F, g \in G} L_1(f \cdot g) = L_1(h^*)$$

These arguments show that under ideal conditions direct learning is preferable to a two stage approach. Why should we therefore turn to a sequential, two stage learning process? There are several possible answers to this question.

- **Algorithmic reasons:** While direct minimization leads to better losses, it requires search in a more complex function space $F \cdot G$. This search is often intractable, leading to high computational complexities and sub-optimal solutions. Splitting the problem into two separate search stages may alleviate this problem.

- **Design related reasons:** The separation of learning into two separate modules of representation and prediction learning creates a more modular system, which has several advantages. First, it is easier in this way to reuse existing prediction (i.e. clustering and classification) algorithms, and design only the representation learning module. The design, research and programming of the two separate modules is easier than the construction of a complete unified system learning directly from weak representation, and it leads to specialization. Finally, a two stage approach creates a flexible system, in which the same representation learning module can be used with various prediction algorithms and vice versa. In this way, it is easier to extend existing algorithms to new domain (by replacing the representation learning module) and new tasks (by replacing the prediction module)

- **'Learning to learn' reasons:** Separating the learning process into two stages opens the door to knowledge transfer between tasks. In this scenario the representation function g is jointly learnt from samples of several related learning tasks. Hence g is a shared component, learnt from a relatively large sample not available for each learning problem alone. Successful learning of g allows prediction learning in each of the related tasks to begin with improved representation. This in turn can lead to higher accuracies with smaller samples.

The papers included in this thesis are mostly related to representation learning of two kinds: improving the distance function for clustering, and creating a meaningful ordered vector representation for images. In both cases the dilemma of choosing between a one or two stage approach appears, and the argumentations sketched out above play an important role.

1.1.4 The included papers

Here I summarize the main contributions of the papers and sections included in this thesis, from the 'representation learning' point of view:

- "Learning with Equivalence Constraints, and the relation to Multiclass Classification" - In this paper we examine whether distance function learning can, at least in theory, completely replace classifier learning. We consider binary distance functions, where an ideal function should return '1' for pairs of points from the same class, and '0' for pairs from different classes. The natural learning inputs are equivalence constraints stating that two points are from the same class (a 'positive' constraint) or not (a 'negative' constraint). We establish the connections between a classification learning problem and the induced distance learning problem in terms of error and required sample size, and show that a concept class is learnable iff its induced distance functions class is learnable. We then show algorithmically how a learned binary distance function

with small error can be used to produce clustering and classifiers with small errors.

- "Learning a Mahalanobis metric from equivalence constraints" - We suggest an algorithm (termed RCA) for Mahalanobis metric learning, which improves clustering results using K-means [B] and graph-based clustering [F]. Since the distance hypotheses family is limited to Mahalanobis metrics, the algorithm can be regarded as learning a linear data transformation, and it is therefore not limited to distance-based algorithms. The algorithm uses positive equivalence constraints alone, and it is conceptually simple and computationally efficient. It is shown to be optimal with respect to several criteria, including information preservation between initial and final representation, minimization of the average distance between constrained points, and generative model estimation under simple Gaussian assumptions. It is empirically shown to improve clustering results, including dramatic improvement in a face recognition application.

- "Learning distance function by Coding similarity" - Here we define 'similarity' and its goals using information-theoretic terms. The similarity between two instances is related to the gain in coding length obtained by moving from independent to joint encoding of the pair. We then suggest a simple algorithm for distance function estimation from positive equivalence constraints, under simplified Gaussian assumptions. The resulting distance is shown to have close relations with other techniques, i.e. FLD and RCA. It is not a metric, and so its usage for clustering is limited to distance based (graph based) clustering algorithms. However, this distance is shown to be empirically superior to RCA and another algorithm in such clustering, and it enables knowledge transfer between tasks in a face retrieval problem.

- "Efficient learning of relational object class models" - When learning from unsegmented images, the preliminary image representation used in most of the current work is an unordered set of image patches with various locations and scales. This representation is extremely weak, as it is both unordered and highly redundant (usually hundreds of patches represent a single image). In this work our main tool for representation transformation is a generative, part-based object model. When applied to an image, the model selects features and orders them into ordered part vectors, which are then used for image classification. We consider models of varying complexity, with an emphasis on relational models, in which both the appearance of parts and their relative positions are described. In this work we do not use a two-stage approach which separates representation from classifier learning. Instead, model learning and classification are jointly optimized to minimize a discriminative loss. The optimization method is based on a non-trivial boosting extension, iterating between weak hypothesis learning and inference of the object's size and location. The combination of generative models with discriminative optimization solves an inherent problem in traditional

maximum-likelihood learning of such models, and it is the main contribution of the work.

- "Subordinate class recognition using relational object models" - In this paper we suggest a two stage learning method for the recognition of subordinate image categories, such as cross motorcycles or dining tables. The method is inspired by cognitive psychology observations regarding the primacy of basic level categories in object recognition and the structural similarity between sub-ordinate categories of the same basic category. In the first representation learning stage, a model of the basic category is learned using images from several subordinates. In the classification stage part vectors, created using the model, are subjected to an SVM for sub-ordinate level classification. The empirical accuracy obtained in this method is often higher than other methods which do not use the representation/classification split or do not rely on the joint sample from all subordinates during representation learning. Thus the main contribution of the work is in showing the utility of inter-task transfer between sub-ordinate category recognition tasks.

The paper descriptions given above focus on the relatively abstract perspective of 'representation learning'. While this may provide an interesting global view of the thesis, the papers are not presented in the context in which they were originally conceived and published. A more detailed presentation of the problem domain and related work is given next in section 1.2 for distance function learning, and in section 1.3 for visual object recognition.

1.2 Learning distance functions

The importance of the distance function for learning algorithms has been gradually acknowledged in the last 20-25 years, and many distance learning techniques have been suggested for the improvement of classification, retrieval and clustering. In section 1.2.1 I briefly consider concepts and techniques related to distance learning. Then in section 1.2.2 I describe the literature regarding fully supervised distance learning, i.e. learning from labels. In section 1.2.3 I describe the main techniques developed for distance learning from the partial information of equivalence constraints.

1.2.1 Distance learning: related concepts and techniques

I discuss the relations between the concepts of distance, metric and similarity in section 1.2.1.1. Then I present distance function learning in the context of related techniques in section 1.2.1.2.

1.2.1.1 Related concepts

The basic intuition for the distance concept is geometric, starting with Euclidean spatial distance. This intuition is captured in the mathematical, axiomatized notions of a metric and an inner product. These notions are very

useful since they extend a large portion of traditional Euclidean geometry to large, flexible families of spaces. Specifically useful is the inner product ('kernel') concept, generalizing the Euclidean dot product. The existence of an inner product endows a space with an induced metric and geometry, and so allows the application of a large 'kernel-based' algorithm family including the SVM, SVR(Support Vector Regression) and others [118]. Therefore the similarity between two points is often measured using a kernel $K(x, y)$, and the distance between them is given by the induced metric $\sqrt{K(x, x) + K(y, y) - 2K(x, y)}$.

While paving the way to powerful algorithms, the 'metric' and 'kernel' notions are limited by their axioms. There are several contexts in which these requirements are not naturally met, and more general families of distance functions should be discussed. Robust distance functions, which ignore outliers and irrelevant aspects in the matching of two items tend to violate the triangle inequality [77]. This is mainly since the matched aspects of an instance depend on the second instance, and vary between distance computations. Such distance functions commonly arise in machine vision problems which require part-based comparison, and in human similarity judgments. Specifically, it was shown in [141] that human similarity judgments often violate the symmetry and triangle inequality metric requirements. In other contexts it is the self-similarity requirement ($d(x, x') \geq 0$ with equality iff $x = x'$) which poses problems. In [99] it is shown that the optimal distance function for nearest-neighbor classification is not metric due to violation of this requirement. In [A] we showed that classifier learning can be replaced by distance function learning, but the relevant distance functions are binary functions which violate the self similarity requirement. Finally, in many practical cases (specifically for image comparisons) distance computations rely on a complex process, and it is very hard to ensure that they obey the metric axioms.

Some attempts have been made to define a general, widely applicable notion of non-metric similarity. Using several intuitive axioms, Lin [92] derived an information theoretic based similarity definition for discrete value vectors. This similarity can be easily learned from event frequencies, and it has been successfully applied to document retrieval [3]. In [82], the similarity between two items is defined based on the likelihood of a shared common generative source for them. In [F] we suggest an information-theoretic definition of similarity which is different from the one used in [92], and is very similar to the one used in [82]. Unlike previous papers, this work suggests an efficient similarity learning method for continuous variable vectors.

1.2.1.2 Related techniques

The research regarding distance function learning has some natural overlap with several related research areas. In general, any algorithm which learns a representation transformation of the input data $T(x)$, can be regarded as a distance learning algorithm, where the distance learnt is the distance in the new data space $d(T(x), T(x'))$.

However in many cases, and specifically when the transformation is relatively simple, this aspect is disregarded.

- **Feature selection** - this is probably the simplest form of representation learning, and it has been researched extensively in recent years (See [87] for a review of traditional methods, [68] for a more updated summary). As in distance learning, it includes stand-alone representation learning algorithms (termed "filters" in this contexts), and others, which are optimized in conjunction with a specific classifier ("Wrappers"). While some of the optimization arguments suggested rely primarily on distances in the reduced space (e.g. [62]), usually the relation to distance learning is not mentioned.

- **Feature weighting**- In this framework one learns weights for the input features, which are then usually used in a KNN algorithm (see [151] for a review). While this is a richer input transformation than feature selection (it includes feature selection as a specific case), it is still very limited in terms of distance function learning, and can be equated with learning of a diagonal Mahalanobis metric. Recent work of this sort is sometimes labeled "feature selection" (e.g. the "Simba" algorithm in [62]) and sometimes "distance learning" (see [120, 158]).

- **Linear projections**- Learning a linear projection A is equivalent to learning a low rank Mahalanobis metric $B = A^t A$. The most popular projections are the Linear Discriminant Analysis (LDA or FLD), first suggested in [58], or its extensions (such as [117]). Learning LDA from equivalence constraints is considered in [18],[B],[F], with differences in estimation method and in post-projection treatment. While in [18] the constraint-based LDA is considered as a distance metric in itself, in [B] it is shown to be the optimal projection to precede another regular Mahalanobis metric, termed RCA. In [F] it is shown to be the optimal projection for yet another non-Mahalanobis distance measure, i.e. Gaussian coding similarity.

Another related corpus of literature deals with hand-designed distance functions for specific tasks, such as visual object recognition [17, 66]. Such methods do not involve automated learning, though often several parameters are determined using cross-validation. A more general approach is the "tangent distance" suggested by [130], in which the distance function is adapted to a set of invariance transformations of the input data.

1.2.2 Learning from labels

Considerable work has been done regarding distance function learning in the fully supervised scenario, with the aim of improving classification. Most of these methods require explicit labels for training, but some actually learn from equivalence constraints, which can easily be extracted from labels. With some exceptions, we focus in this section on the former set of methods, and defer the discussion of the latter to the next section. Interestingly,

the distance learned from labels are almost always metric. The literature can be divided into two rather different branches, pursued by different research communities: Metric learning for KNN classifiers, and Kernel learning for SVMs.

1.2.2.1 Metrics for nearest neighbor classification

Learning a metric for KNN is not radically different from metric learning for clustering or retrieval, but the task differences lead to a subtle difference in the characteristics of the optimization argument. Specifically, in KNN only relations between 'near' points count. Consider for example a metric in which a certain class is manifested in several distinct clusters, which are relatively well separated from other classes. This may be a good representation for KNN, but bad for clustering and instance based retrieval, since each instance is connected only to a small portion of the relevant class.

Improving NN classification using distance learning was already studied by Short and Fukunaga in [129], where the optimal metric for NN is characterized in terms of the local class densities and their gradients. This is a local metric, which has to be estimated for each point separately, and an algorithm is suggested for its estimation and incorporation into the NN classification. Other methods which adapt the metric on a local basis have been suggested more recently, based on different ideas. Specifically, local LDA adaptation was suggested in [69], and computing distances from local label-dependent hyperplanes was suggested in [146].

While not optimal, learning a global Mahalanobis metric is simpler, and this is the more popular alternative. In [96], the "$0-1$" error on nearest neighbor is smoothed using a stochastic formulation, and optimization of this argument is done using gradient descent. Only a diagonal Mahalanobis metric is learned. In [64] a similar method is used to learn a full Mahalanobis metric. Yet another stochastic and smooth loss is suggested in [63], with the advantage of leading to a convex optimization problem. In [150] the optimization cost is based on a notion of large NN classification margin, and the optimization is done via semi-definite programming (SDP). In addition, there are methods which were published as learning Mahalanobis metric from labels, with classification in mind, but actually do not rely on full labels but on equivalence constraint information. In [163], distances between pairs of points are modified according to the label equivalence (i.e. the distance between points with the same label is shrunk, and the opposite for points with different labels), and a linear transformation is sought which approximates the modified distances. In [125] the problem is presented as a large margin classification problem over point pairs. The resulting SDP problem formulation is solved using an online algorithm.

1.2.2.2 Kernel learning for SVMs

Learning the kernel for SVM classification has a relatively short history, and it is still a controversial issue. There are some encouraging theoretical and empirical results, but still no persuasive examples of empirical improvements in the 'large data set' scenario (as far as I know). From the theory point of view, Srebro and Ben-David [131] have recently shown that for a kernel family with pseudo-dimension d, and a classifier learned with margin γ, the gap between the training error and generalization error is bounded by $\sqrt{O((d+1/\gamma^2)/n)}$, with n denoting the training sample size. Since the equivalent bound for a single kernel is $\sqrt{O(1/\gamma^2)/n}$, and since we can bound the pseudo-dimension of several interesting kernel families, this is an encouraging bound.

In practice, kernel learning methods usually learn the Gram matrix, and not a kernel function defined on the whole product space of point pairs. Such methods are limited to a transductive learning scenario. Cristianini et al. [42] first approached this problem using the influential concept of "Kernel alignment". Alternative approaches suggested include learning the kernel matrix by boosting [39], semi-definite programming [86], Gradient descent of generalization bounds [26] and generative modeling [164]. Another set of research papers has dealt with learning the parameters (usually 2-3 parameters at most) for a pre-defined kernel family. In common practice such assessment is made using cross validation, but alternative methods include statistical estimation [109] or gradient descent of generalization error estimates [35], where the latter method has been used to learn a large number of parameters.

There has been only a little work on kernel function learning in the inductive setting. In [73] we presented a kernel function learning algorithm, based on product-space boosting. The method is an adaptation of a previous method for distance learning from equivalence constraints [71], and it relies on a weak learner which incorporates equivalence constraints into a mixture of Gaussians fitting process [126]. This algorithm is shown to enable significant knowledge transfer between related classification tasks with small samples. Another algorithm which may be used for kernel learning is presented in [157], where an existing kernel matrix is modified, and then approximated by a learned Mahalanobis metric in the induced feature space.

1.2.3 Learning from equivalence constraints

In recent years there has been a growing interest in learning from partial information in the form of equivalence constraints. Formally, these are triplets (x_1, x_2, y), where x_1, x_2 are data points and $y \in \{+1, -1\}$ is a label indicating whether the two points are similar (from the same class/cluster) or dissimilar. There are two essential sources for the interest in such constraints: their availability in some learning contexts and the fact that they are a natural input for distance function learning. Regarding availability, there are several scenarios in which

equivalence constraints can be obtained automatically or with low cost in human labor, while labels are not readily available [70]. In video [159] or surveillance [70] applications one can often achieve such constraints based on the Markovian dependency between successive frames of a movie. For example, it is often possible to automatically identify that the human face in two such frames is the same, based on low level continuity cues. Other scenarios are information retrieval with user feedback [2, 38], or distributed learning with many uncoordinated teachers [70]. In these cases, equivalence constraints can be acquired in a cheap and robust manner, while labels are much harder to achieve. The other reason for the extensive usage of equivalence constraints is that when we regard the distance function learning as a classification problem on the product space (i.e., with pairs of points as input), the natural 'labels' are equivalence constraints [A]. Because of this characteristic of equivalence constraints, fully supervised distance learning algorithms, which receive labeled points as input, often turns them into equivalence constraints and operate directly on the latter (for example [37, 113, 125, 163]).

While binary equivalence constraints (positive or negative) have received most of the attention, several related supervision forms have also been considered. Specifically, soft constraints, indicating the likelihood of several points being together were considered in [88]. In general such constraints are less appealing since they are harder to obtain in real life scenarios and harder to incorporate, potentially leading to hard inference problems. Another form of supervision, which can be regarded as even weaker than equivalence constraints are 'relative comparisons', used to learn distances mainly in retrieval contexts [4, 115, 120]. These are triplets of the form "A is more similar to B than to C". Notice that such triplets can always be extracted from labels, and sometimes even from a set of positive and negative equivalence constraints (if certain points appear in both positive and negative constraints), but not the other way around. For information retrieval, in which the order of the retrieved items is the important aspect, such triplets seem like natural supervision.

In what follows, I will consider how equivalence constraints have been used for distance function learning in the domains of information retrieval 1.2.3.1 and semi-supervised clustering 1.2.3.2

1.2.3.1 Image retrieval and verification

In a common paradigm for information retrieval, and specifically for image retrieval, distances between a query and items in the data base are computed, and the most similar items are returned. A closely related task is verification, where the query is accompanied by a conjectured identity and the task is to verify whether the query indeed has that identity. This task is mostly employed in face recognition, and its implementation usually relies on distance computation as well. The retrieval task is rather similar to classification: the question can be posed as "which items are from the same class as the query?". One might therefore expect that distance function leaning

for retrieval should not be that different from distance learning for classification. This however, is *not* the case, due to several emphasis differences between classification and practical image retrieval.

As I noted in section 1.2.2.1, distance learning for KNN emphasizes relations between 'near' points, while in retrieval, at least a-priori, all the pairs of points have equal importance. Due to this difference, metric learning algorithms for KNN usually rely on explicit labels, from which only the relevant equivalence constraints are extracted. Distance learning algorithms for image retrieval, even when they are declared as 'learning from labels', are usually defined and operate with equivalence constraint input [37, 98, 113]. Other differences in learning arise due to the difference between the data domains used in traditional classification and the much more problematic input of object and face image retrieval. While traditional classification is usually done with a pre-defined vector representation with several dozens of features at most, for images usually no such representation exists and thousands of features are usually considered. Due to the increased input complexity, Mahalanobis metric learning, which is the most popular approach for classification (see section 1.2.2.1), is usually prohibitive for learning distance from images. Instead, often distance functions are learned which are not metric [99, 113], and feature selection and synthesis is an important aspect of the distance learning [4, 99]. Finally, inter-task knowledge transfer, usually not considered in traditional classification, arises naturally in several retrieval applications, specifically in face retrieval and verification [37, 99, 113]. In such application one has to learn a distance function from a set of faces, and apply it to faces which were not seen during the training period.

Recently, some distance learning methods have been specifically designed in the context of face and object retrieval. In [98, 99] a non-metric distance is learned which is a linear combination of elementary distance functions. The method is based on a discriminative probabilistic model and its optimization, for a given set of elementary distance functions, is a convex problem. The set of elementary distance functions is chosen with a greedy local search. In [4] a similar linear combination of elementary distance functions is learned using product-space boosting from relative comparison triplets. In [37] a distance function of the form $d(I_1, I_2) = ||G(I_1) - G(I_2)||$ is learned from equivalence constraints, where G is a non-linear transformation of the image implemented as a convolutional network. This distance is clearly metric, and its optimization is done using back propagation gradient descent. Another suggestion, put forward by [113], is to learn the discrimination between 'I1 and I2 are the same' and 'I1 and I2 are different' in the *difference space*, i.e. the space of vector differences $I_1 - I_2$. The method is applied to face images, properly aligned and projected using PCA, and the distinction is learned using standard SVM.

When the input data are simpler than real images, learning a Mahalanobis metric has been considered for visual recognition and retrieval tasks [56, 115]. In [56] an online algorithm developed for classification (POLA, [125]) is

used for character recognition. The algorithm recently suggested in [115] learns a Mahalanobis metric with low L_1 norm from relative comparisons, using linear programming optimization. However, while its declared aim is retrieval, it was only empirically tested with traditional low dimensional data sets from the UCI repository [23]. Other distance learning algorithms which have been used for image retrieval are the Distboost algorithm [72], the LLMA algorithm [34], RCA [B], and Gaussian Coding Similarity [F]. These algorithms have also been used for semi-supervised clustering, and I defer their discussion to the next section.

1.2.3.2 Semi-supervised clustering

In the scenario we discuss in this section a data set of n unlabeled data points is augmented with a (usually small) set of equivalence constraints. In contrast to labeled data, from which $O(n^2)$ equivalence constraints can be extracted, the amounts of equivalence constraints considered in this task are usually a fraction of n, and at most linear in n with small constants. The task is clustering, i.e. partitioning of the data into mutually exclusive sets according to a 'natural' equivalence relation. This original clustering task is clearly ill-posed, but with the advent of semi-supervision it is less so, as the partition chosen should strive to obey the constraints, and it is hence less arbitrary. In practice, clustering and related distance learning algorithms are tested on labeled data sets, for which the required partition is known (though the labels are hidden from the clustering algorithms), and performance is judged based on agreement with the known labelling.

Equivalence constraints have been incorporated into the clustering process in two distinct ways: direct incorporation and distance function learning. In direct incorporation, clustering algorithms are altered in order to use the information contained in the equivalence constraints, and prefer partitions which obey them. Several known clustering algorithms have been augmented in this way, including K-means [148], complete linkage [83], EM of a Gaussian mixture model [126] and Normalized-cut [19]. In the second way, a distance function is learned from the equivalence constraints, and this distance function is then used in a second clustering stage. For graph-based clustering techniques (which are solely distance based), the distance function is used to compute the input distance matrix. For other clustering techniques, which require explicit vector representation, the transformation compatible with the distance matrix is used to re-present the data before clustering. Of course, in this case, only distance functions for which an equivalent transformation is readily available can be used. The two methods of distance learning and direct incorporation can be used together. In [21], the two methods were jointly and separately used with the k-means clustering algorithm. The results indicate that the contributions of the direct incorporation and distance learning do not completely overlap, and joint usage is preferable. The empirical results in [B], which were also obtained with k-means, show a general performance advantage of distance learning methods over direct

incorporation for this algorithm.

Mahalanobis Metric learning. Like in the classification task, the Mahalanobis metrics family has received considerable research attention. Since a learned Mahalanobis metric A directly entails a linear data transformation $B = A^{-\frac{1}{2}}$, it is conveniently used with standard vector-based clustering algorithms. Usually such metrics are used in conjunction with K-means, or the constrained K-means [148] algorithm. The first algorithms suggested for Mahalanobis metric from equivalence constraints were suggested in [127,158]. In [158], the metric is learned from positive and negative constraints using PSD convex optimization. The suggested algorithm is iterative, based on interleaved steps of projections and gradient descent. In [127] the simpler Relevant Components Analysis (RCA) method was initially presented, learning a Mahalanobis metric from positive constraints alone. This method is further developed and analyzed in [6] and in [B]. The cost suggested by RCA is optimized in a closed form solution, which require a single matrix inversion. This algorithm was later expanded to include negative constraints in [160]. In [156] it was expanded using a boosting process to include both types of constraints, as well as unlabelled data. In [139] a kernelized version of RCA was presented, and in [152] RCA and its kernel extension were analyzed from a quantum physics perspective. In [18] a low-rank Mahalanobis metric is suggested, based on estimation of the LDA projection from positive equivalence constraints. As mentioned in section 1.2.1.2, similar constraint-based LDA has been shown to be an optimal pre-processing stage for techniques discussed in this thesis, RCA [B] and Gaussian Coding similarity [F].

Several kernel-oriented methods have been suggested for Mahalanobis metric learning. These methods can be used to learn a simple Mahalanobis metric on the input space, or a Mahalanobis metric on a high dimensional space, computed via the kernel trick. In [84] a metric is learned from positive and negative constraints by demanding that the (kernel mediated) distance between dissimilar points should be enlarged by a certain margin and vice versa, thus increasing the kernel alignment [42]. The resulting optimization is a quadratic programming similar to the ν-SVM algorithm, but over point pairs. A similar formulation is studied in [157], but here a linear transformation is applied to (kernel mediated) pairwise similarities to achieve improved kernel alignment. A third kernel oriented, margin-based approach to Mahalanobis metric learning from constraints is POLA [125], mentioned earlier in the context of fully supervised learning.

Non-linear methods. As Mahalanobis metrics correspond to a globally linear transformation of the input space, they are sometimes not expressive enough for complex distance function learning. One possibility for learning non-linear metrics is by learning a Mahalanobis metric in a high dimensional space via the kernel trick, as discussed in the previous paragraph. Several other solutions have been suggested that learn a non-linear distance

function directly in the input space. The Distboost algorithm [71] learns a non-metric distance function from positive and negative equivalence constraints using product space boosting. The weak hypotheses combined are soft partitions of the feature space learned with a weighted version of the constrained EM algorithm [126]. Since the distance is non-metric, there is no explicit data transformation associated with it, and it has been used with graph-based clustering algorithms, which are purely distance based. The LLMA (Locally Linear Metric Adaptation) algorithm [32] learns an input transformation which is globally non-linear as a combination of local linear transformation. Only positive constraints are used, and the cost tries to bring these points closer together while keeping the local neighborhood topology structure undistorted. A kernel-based version of this algorithm was later presented in [33]. Finally, the Gaussian Coding Similarity (GCS), described in this thesis [F], is a non-metric distance function derived from information-theoretic principles and Gaussian assumptions. Like RCA, this is a relatively simple and efficient technique, requiring only the estimation and inverse of two covariance matrices. It was used to improve graph-based clustering results, as well as for image retrieval.

1.3 Learning Image classification

Perhaps more than in any other research domain, the representation problem lies in the core of research in visual object recognition. This problem has been studied for decades, with many different approaches and representations. A brief survey of several methods put forward in the last 20 years is given in section 1.3.1. I believe that unlike the traditional view of classification in machine learning, image understanding tasks require a joint solution for the classification and feature extraction tasks, with context and feedback playing important roles. While these notions are very old, their applicability to current learning in vision is highly controversial. In section 1.3.2 I present a suggested viewpoint for feature extraction and classification, and try to argue for its validity and usefulness.

In the last few years, a major research trend has focused on the problem of object class recognition from unsegmented images, with the images initially represented via sets of local patch descriptors. In this research context, the more general debate regarding the utility of context focuses mainly on questions regarding the modeling of spatial relations between object parts. This research context is presented in section 1.3.3 with an emphasis on generative model learning, in which the contribution of papers [C],[D] is made. In section 1.3.4 I survey the recent literature regarding discrimination between similar classes and knowledge transfer between recognition tasks, which is mostly relevant to paper [E] in this thesis.

1.3.1 Suggested approaches to object recognition

Image understanding requires answers to high semantic questions such as "what are the objects in this image?", "where are they?", "what do they do?". There is an enormous gap between these questions and the initial pixel image data representation. These initial data are formed by a complex interplay of object viewpoint, camera position and illumination conditions, which renders isolated single pixel values meaningless w.r.t the high semantic variables. I cannot hope to cover all the relevant literature in this brief introduction. Instead I merely outline several approaches which have been influential in my scientific education. Specifically I consider earlier approaches based on 3D models in section 1.3.1.1, and approaches based on 2D appearance or shape matching in section 1.3.1.2.

1.3.1.1 Object recognition using 3D models

Traditional object recognition systems typically relied on a data base of 3D object models, where recognition requires a match between the input 2D image and model from the data base. Typical processing stages in such a system include extraction of low level geometrical features, perceptual grouping, data base search and indexing, and finally a verification alignment stage. While this description is clearly schematic, it does grossly characterize several systems [95, 122] and provides a convenient thematic framework for the presentation of several lines of work.

Perceptual grouping. The first stage toward recognition in such systems is the extraction of low level geometric features from the image, e.g. edge maps, edgels or 'interest points' obtained using an interest point detector [119]. These low level features are usually not distinctive enough, and so perceptual grouping techniques can be applied to gather them into more sematic groups, which can be used as keys for a data base indexing mechanism. Perceptual grouping relies on basic Gestalt principles such as smoothness and proximity [124], parallelism and co-linearity [95]. Specifically, in [124] edgels are combined to form salient (i.e. long and smooth) curves using an efficient recursive technique. In the SCERPO system presented in [95] straight lines are gathered into 'meaningful' quadruplets, used later in a probabilistic indexing system. In [122] contours are extracted using a stick growing method. The most salient ones are then described using appearance based patches and serve as indexing keys. While such low level grouping was useful in several systems, using perceptual grouping to obtain higher level 3D structures has not been that successful. A known attempt to base recognition on simple 3D elements is Biederman's 'Recognition By Components (RBC)' or 'Geons' Theory [20] which was a source of inspiration for several systems (a similar notion was suggested earlier by Binford [22]). In this representation objects are described as ensembles of simple parts, termed 'Geons', which have a certain degree of viewpoint invariance. However, as discussed

in [45], these systems apparently failed because of two main difficulties: an inability to extract Geons reliably, and the instability of Geon decomposition.

Indexing and correspondence. After extracting a set of features from the image, these features have to be matched against similar features in the 3D model database. The match can be based on appearance similarity [122] or more commonly on geometric properties, hopefully invariants [155]. Specifically in the 'Geometric hashing' technique of [155], k-tuples of interest points are used to produce geometric invariants, by expressing one point in the reference frame induced by the others (3 points are enough for affine invariance). These invariants are then matched against the database and vote for the object identity and pose. In [138] this framework is expanded to work with lines as input and with a smooth, probabilistic voting schema. One main problem of this method is the high sensitivity of the invariants. While successful for single object recognition, the invariants are usually not stable enough to allow for recognition of an object class.

Alignment and verification Given a correspondence between several image features (points or lines) and 3D model features, a final matching can be done by solving for the optimal transformation which brings the 3D model as close as possible to the image. The computed transformation can then be used to project other points in the model and verify their existence in the image. Such a verification stage, used for example in [11, 76, 95], is often the most robust phase of the system, and it enables the rejection of most false alarms. This step is usually relatively expensive, so alignment can only be applied to a small set of filtered correspondence hypotheses. The main drawback of this method is its reliance on explicit 3D object models, which cannot be learned automatically and are hard to encode manually. A certain remedy for this last point is studied in [142], where a 3D model is represented using a set of 2D object images, and the input image is matched to a linear combination of the 2D model images.

1.3.1.2 2D shape and appearance based approaches

During the 1990s new techniques for object recognition became popular including shape-based image retrieval and appearance-based classification. These techniques do not rely on a 3d model database, and they gradually introduced machine learning techniques into computer vision.

Shape representation and comparison. In this context "shape" denotes the outline contour of an object in an image. Such contours can be extracted by applying an edge detector to an image, followed by some perceptual grouping into point sets, lines or curves. Several simple distances were successfully used to compare shapes repre-

sented as points sets, including the Chamfer distance, originally presented in [7] and the Hausdurff distance [75]. The former, which is more immune to clutter (as it involves a mean operation between pairwise point distances where the Hausdorff distance takes the maximum), compares favorably with the modern shape context representation (see below) in [134]. A more sophisticated edit distance between shapes was suggested by Gdalyahu and Weinshall in [60]. Here the shapes are represented as polygons, with the length and absolute orientation describing the polygon edge constituents. The distance between two shapes is computed using dynamic programming. A different shape representation is obtained when one considers the shape's internal skeleton instead of the boundary, i.e. the medial axis [165] or its shock graph extension [121]. These structures are often regarded as more stable with respect to noisy segmentation than the contour line. Comparing two such graphs is highly non trivial, and in [121] it is done using a graph edit distance algorithm. Finally, Belongie et al [15] suggest representing a shape using a set of feature descriptors localized on the shape edge points, termed 'shape context'. This descriptor measures the distribution of other shape edge points using a log-polar histogram. Two shapes are compared by matching the two sets of feature descriptors, and this can be done optimally using the Hungarian method for two sets of equal size and a one-to-one correspondence. In general, while some good methods for shape comparison were suggested, the main problem with this school is the difficulty to reliably extract shape contour from cluttered natural images.

Appearance based methods. During the 1990s, an influential line of work dropped 3D models altogether and resorted to direct appearance based comparison between images (or between an image and an appearance model/prototype). This allows learning techniques to be used naturally. The comparison is usually done with global templates [50], hence such method depends on a good alignment between the compared images. In [28] such a template matching approach was compared to an approach based on geometrical features in a face recognition task, and exhibited superior performance. Appearance based comparison is often done in a reduced sub-space, in order to ignore irrelevant variability. Standard techniques used to find appropriate subspaces for face recognition are PCA [140] and LDA [14]. The limitation of using a global template can be partially overcome by using a sliding window approach, in which windows from all possible locations at several scales are compared to the object template. Such approaches were successfully used for face detection [133, 147]. Other appearance-based approaches use histograms of local appearance features as the main descriptor. I defer discussion of such approaches to the next section.

1.3.2 Context and feedback

I now try to sketch out the main differences between two possible approaches to feature extraction and classification: the traditional mechanisms of machine learning, and the (so I believe) required mechanisms for image understanding. This done in section 1.3.2.1, and I then argue for the second option in 1.3.2.2.

1.3.2.1 A contextual view of feature extraction

Feature extraction and representation is an issue of growing interest in machine learning, as partially surveyed in section 1.2. However, in most of the cases the features are simple constant functions of the data, and all of them are applied to the data at the test phase. 'Which features to calculate' is not a function of the exemplar to be classified. Intuitively, this is not the right policy when it comes to visual object concepts. When I am presented with a picture of a cat, features such as size, head shape and whiskers are the ones I use, and these features are not the ones I need when the picture is of a can of olives. I can't even compute these features for a can of olives. Representation learning in this case involves more than learning intermediate features, and can be viewed as a control learning problem: We need to learn a strategy to select which features to calculate as a function of the test data. When considering a test data example, a decision has to be made as to which features should be computed to represent it. These new features should then be used to decide which features to compute next. Trying to compute all the features that might be relevant to some label in advance is not computationally feasible, nor required, for tasks such as object recognition.

The dependence of feature extraction on the class label suggests that classification and feature extraction should be done together in a feedback loop, where models of the presumed objects and their location guide the search for relevant features in a top-down fashion. Such guidance is possible due to a rich set of appearance and spatial relations which typically exist between scenes, objects and parts. Viewed this way, classifying an example is not a matter of calculating a fixed set of features and returning the label of the region in which the feature vector lies. Instead, classifying is done after a dynamic process, in which models that received some support in the initial data led to the calculation of additional features, and competed until one of them won. Such a view of classification can explain our inability to compare two very different objects: it is much harder (even meaningless) to compare a dog and an olive can, while comparing a dog and a cat is easy and natural. If features are computed 'on demand' by a learnt model, a dog and a can live in different feature spaces, and no natural distance between them can be determined.

1.3.2.2 The case for context

The view of classification and feature extraction in humans as a joint, iterative process is hundreds of years old. In the writings of the philosopher Immanuel Kant [81] the application of a concept to a sensation manifold (i.e. for us, pixels) is termed 'judgment', and it is described as a mutual, loopy interaction between the two. Recent human vision research assigns an important role to prior shape expectations in the task of object segregation. Peterson et al. [112] showed that familiarity with a shape enhanced the tendency to interpret it as a figure, even when traditional segmentation biases such as convexity and enclosure specify it as ground. Another line of studies, conducted by Needham et al. [106, 107] showed effects of prior object knowledge on object segregation in 4.5 month old infants. From a physiological perspective, massive top down fibers are known to exist in the visual ventral pathway [80], which is the main path used for object recognition in humans. The fibers exist along the entire path, from IT to V4, V2 and V1, and given their dominance top down guidance of intermediate representation in humans seems fairly plausible.

The view of classification and feature extraction as a joint process has been influential from the early days of computer vision, and can be seen clearly in more recent annotation systems developed for image understanding such as SCHEMA and CITE [46, 48, 49]. Specifically, in [48] learning an object recognition scheme is explicitly considered as a 'control' problem, with bottom-up and top-down information channels. However, the usefulness of context and top-down feedback for contemporary machine vision technique is controversial, and there are good arguments for both sides. Frequent segmentation failures of objects [25], shapes [111] and parts [45] in bottom-up methods seems to indicate a need for top-down guidance. Some positive evidence for the utility of context comes from methods which use a graphical model to represent spatial relations between object parts (this framework will be described in length in section 1.3.3.3). Comparisons made between models with varying degree graph connectivity [40],[D], indicate that a higher degree of spatial relations provides better recognition performance, though the increase is slight for high connectivity. Another recent line of work by Torralba et al. [104, 136] shows the utility of a global context variable, modeling the scene type. An interesting phenomenon seen in [136] is that the spatial context is very helpful for the detection of small items, such as a computer mouse, but less helpful for larger objects like a monitor.

On the other hand, recognition systems built according to the feed-forward paradigm are usually much simpler and more efficient. Since this paradigm is in accordance with current machine learning tools and theory, it allows for easier integration of powerful learning techniques. Currently such systems obtain the best results in the task of object class recognition on the common benchmarks of the Caltech 101 data set [90] or the Pascal challenge [5]. The situation however, is usually the reverse for localization tasks(see the results of [5]). In [153] a comparison

is made between a context variable computed using simple feedforward operations on the surrounding object area, and a context variable based on detection of related objects in a street scenario. Both kinds of context are found to make a marginal contribution to an appearance based object classifier, and specifically the contribution of related objects context is lower than that of the simpler context. I believe these results can be explained by the relatively weak contextual connection between the objects considered (compared for example with the spatial relations between parts in an object, which are often much tighter), and the relatively low difficulty of the task for the appearance based classifier.

1.3.3 Parts and wholes

Over the last several years initial image representation as a set of patch descriptors has gained considerable popularity. Such a representation currently dominates research in object and object class recognition, and give a new twist to the controversy regarding the importance of context and feedback. In 1.3.3.1 I describe this representation and the types of part based object representations typically learned from it. In this recognition scenario, incorporation of contextual information is usually done by inclusion of spatial part relations in the object description, which influence the choice of relevant image features. I describe algorithms which do not incorporate such information in section 1.3.3.2, and algorithms that do include it in 1.3.3.3.

1.3.3.1 The 'set of patches' representation

The recent popularity of the 'set of patches' representation starts with articles by Sali and Ullman [116] and Burl et a. [29]. It can be thought of as a compromise between earlier representations. On the axis of local-versus-global features, it stands between global templates, which are not flexible enough to capture inner-class variability, and local geometric features such as edgels and interest points, which are not discriminative enough. This trade-off motivates using patch features of intermediate complexity in [116, 143]. The trade-off presented in [29] is different, where a part based model describing part appearance and spatial part relations is considered as a compromise between purely appearance based and purely geometric methods.

In several papers, the set of patches extracted from an image are chosen using manual segmentation [40,52,85]. In this case, the relevant object parts are chosen a-priori, and each image can be represented using an ordered feature vector by concatenating the chosen parts in a pre-defined order. This ordered vector has high semantic value, and learning in this scenario can be done using traditional machine learning tools. However, in most of the work done, such manual intervention is not assumed, and only image labels are supplied. In this case the learning problem is much harder.

Without manual feature segmentation, features are extracted for each image automatically. The locations and scales of the patches are chosen using interest point detectors (e.g. in [54, 110]), on edge locations [97], or at random [79]. The patches extracted are represented using simple 'patch descriptors'. Standard dimensionality reduction techniques were used such as PCA [54] and DCT [C,D], but gradient-based descriptors as SIFT [94] are more popular, as they seem to have more discriminative power [100]. The result is an image representation as an unordered set of features, where the semantic roles of the features are unknown. Typically the set is also large (usually hundreds of features per image are considered) and highly redundant. The learning problem is naturally considered as a supervised learning problem with unordered sets. Alternatively, if we consider the problem as predicting hidden labels for each patch (i.e. 1 if it is an object patch and 0 if it is not), the problem can be regarded as a semi-supervised problem in a Multiple Instance Learning (MIL) framework (see [36] for details).

While other approaches are possible, a natural approach to the learning problem in this case introduces an ordered vector intermediate representation, whose elements are termed 'Parts'. In this view, the image features extracted do not have known semantic roles, but the parts are fixed semantic carriers, with a fixed predictive function w.r.t the unknown label. In a face recognition example the parts are eyes, nose and mouth, and they are implemented by features chosen from the unordered set. In a context-free model, the features implementing each part are chosen independently, e.g. the chosen eye does not affect the choice of a nose. In a contextual model, the choice of features implementing each part depends on the chosen features for other parts. Usually spatial dependence is considered, demanding that the parts are places in a specific spatial relation to each other. Enforcing such dependence between the parts is often done using some form of feedback inference, though other techniques are possible.

Due to these considerations, the distinction between contextual and non-contextual models roughly corresponds to the distinction between simple feed-forward methods and algorithms which include feedback, and to the distinction between appearance based algorithms and those which model spatial part relations. I review these two algorithm families in the next subsections.

1.3.3.2 Context free, feed forward systems

The simplest object model used for class recognition from feature sets is the 'bag of features' model. The object is described by a set of independent parts, each described using an appearance template. Each part is implemented and scored by one or more features from the image. Learning such model is done in a discriminative fashion, with an emphasis on good parts selection, where 'good parts' are those that tend to appear in object images and not in background images. Despite the limited expressiveness of such models, their simplicity allows efficient and

accurate minimization of tight classification related loss functions, and they have been used with great success in the last few years. Currently such methods have a dominant influence, as can be judged by considering the competing methods in the PASCAL challenge [5], which are mostly of this kind.

'Bag of feature' approaches differ in the type of features used and their extraction, and in the details of the feature voting mechanisms, but they primarily differ in the learning and part selection method. In the line of work of Ullman et al. [116, 143, 145] part prototypes are image fragments selected in order to approximately maximize the mutual information between the fragment existence and the class label. The classifier over the chosen feature detectors (i.e. parts) is learned using Naive Bayes [116, 143], or TAN network and SVM [145]. In [110],[C] the combined fragment selection and classification are achieved using a boosting algorithm. While in [110] feature detectors are binary weak hypotheses, in [C] these are probabilistic part models, combined into a generative object model. In [30] a discriminative probabilistic bag-of-features model is presented and classification is done by model averaging, approximated by a Markov chain. Dorko and Schmid [47] used a Gaussian mixture model trained over features from all the images to obtain part prototypes. The most discriminative prototypes are then chosen and combined in a simple voting scheme for the class label. An interesting biologically motivated bag-of-features system was suggested in [123]. This system was later extended and applied to a localization task using the sliding window approach in [105]. In the methods outlined above the part response is computed based on its single best match with an image feature. In a slightly different approach averages of part responses over the features are computed, which results in a histogram representation for the image. Such feature histograms were successfully used in [43, 44].

While the methods mentioned above compare an image to a model, several distance based methods which compare two images directly were recently suggested. Intuitively such a comparison should be based on a correspondence established between the two feature sets, but this is not always the case. For example, the kernel suggested in [154] compares two feature sets based on the principal angle between the linear spaces they span, without correspondence establishment. Grauman and Darrell [66] suggested a pyramid match kernel, which relies on an implicit, approximate feature correspondence. Other distance-based methods do rely on explicit correspondence, and can be regarded as context-free if the matching is established for each feature in the source image independently. Such a method, in which the matching is based on appearance and absolute location is successfully used in [161] over the Caltech101 data set.

Context-free models usually rely on appearance and drop spatial part relations altogether, but this is not always the case. In [1] an intermediate representation is considered which includes, in addition to standard appearance-based feature detectors, binary indicators of spatial relations between detected features. The relation indicators

are computed based on the responses of the feature indicators in a feed-forward manner, so this is a context-free system. This approach was applied to car localization using a sliding window approach. Similar usage of 'late' spatial relation features appeared in [162], where such relations were considered as weak hypotheses in late boosting rounds.

1.3.3.3 Relational models and spatial context

Relational part based object representations typically model the appearance of object parts and their relative locations (often termed 'shape'). Such representations, also termed constellation models, were already suggested by [57], but the recent interest in them started with work by Perona et al. [29, 54, 55, 74, 91, 149]. Initially, appearance models were learned from images with manual segmentations, followed by shape learning from images taken under strictly controlled conditions [29]. However, these restrictions were gradually removed in subsequent work. In [149] learning was already done with unsegmented images, but still in a two stage fashion. In [54] this latter disadvantage was removed, via an EM-based optimization of joint appearance and shape model. In these papers the model studied was a generative model, and classification was done using a likelihood ratio test against a simple background model. In addition, the spatial relations modeled were a full clique, i.e. the location of each part depended on all the other parts. Since several feature candidates are possible for each part, evaluating the joint probability for all the possible image-to-model matching hypotheses is exponential in the number of parts.

In order to improve the computational complexity of generative model evaluation, spatial graphical models of lower connectivity were recently suggested in [55],[D], [41, 93, 132]. In [55] a star-like model was suggested, where the locations of all parts were described relative to a single 'root' part. This allowed evaluation of an existing model with time complexity linear in the number of parts, but learning remained exponential. An alternative technique for learning star models or K-fans generalization of such models was presented in [41]. In this technique efficient EM learning becomes possible based on an approximation allowing two parts to be implemented by a single feature. However, the global maximum likelihood optimum of such approximated learning occurs in models with repetitive parts [D]. In order to avoid this, heuristic laborious initialization of the model is done in [41], to ensure that the EM converges to a local maximum without repetitive parts. The analysis presented in [D] showed that in a purely generative setting one has to chose between exponential learning complexity (as done in [55]) and optimality of models with repetitive parts (as done in [41]). The solution suggested in [D] is based on discriminative optimization of a generative star-like model, which enables linear learning complexity in a model with diverse parts. An alternative solution proposed independently in [93] and [132] allows linear learning complexity by relaxing the constraint stating that a part is to be implemented by a single image feature.

This, however, lead to a less intuitive 'part' concept, implemented in many image patches. Such parts are less convenient for further processing in tasks such as localization or sub-class recognition (as suggested in [E]).

A closely related alternative to generative object models, termed Implicit Shape Model (ISM) was successfully applied in [59, 89] for object localization and segmentation. The object description is similar to [D], i.e. object parts are described by an appearance model and an offset from a reference point on the object. At localization time probabilistic Hough voting is used to identify the most likely object position, and this information is then propagated back to allow object segmentation. In [59] a second verification stage is added, in which object hypotheses generated are further filtered using an SVM classifier. While highly successful in localization tasks, model learning in these systems is done using hand segmented images.

Finally, contextual part relations can also be embedded into a distance-based approach, comparing two images directly. An example of such an approach is presented in [17], in which the distance depends on explicit correspondence established between the feature sets from the two images. The correspondence is found by maximizing a cost including terms scoring the appearance matches and terms scoring the similarity of spatial relations between matched features. The optimization is done via relaxation of integer quadratic programming.

1.3.4 Similar object classes and knowledge transfer

In paper [E] we consider how a model of an object class can be further used to discriminate sub-classes of the class. This is an approach to distinction between similar object classes, based on knowledge transfer between related recognition tasks. In this section I briefly discuss distinction between similar classes, knowledge transfer between tasks, and work combining these notions in the research literature.

Similar object classes. The distinction between similar visual object classes is in general a harder problem than standard object class categorization, and it has received less research attention. Actually, all the work I know of (up to paper [E]) have dealt with the two specific cases of face and car type recognition. Gender discrimination was already considered in the early 1990s using geometric features [27] or the appearance-based approach [65]. The geometric approach of [27] relies on specific face features, as mouth width or distance between the eyes, and it was designed by hand to the specific task. In [65] a neural network is applied to aligned face images. While this method is more general, it requires exact detection and strict alignment of the faces, and it is less applicable to more flexible (non-rigid) objects. In [103] a similar approach to gender discrimination was successfully applied using an SVM classifier and specifically for small (21×12 pixels) images the SVM outperformed average human classification. Similar global template based approaches were applied to person recognition [113] and to ethnic classification [67].

Some work has recently been done on the practical problem of distinction between different kinds of cars. Usually the cars are segmented from a video recording and they share the same viewpoint. In [97] the features used are large SIFT features (covering large portions of the car), located on edge points. Edge points are shown to be more stable than interest point detectors in this kind of data. Object class models are learned based on appearance alone, or on appearance and absolute position, and classification is made using an LRT. In [111] a dense correspondence between the interiors of two objects is established by expanding correspondence of the silhouettes or of the shock graph skeletons. This correspondence is then used for an appearance based comparison.

Learning hypothesis bias for object class recognition. Intuitively, knowledge transfer between object recognition tasks should be primarily useful when the objects are similar. Nevertheless, a large portion of the work done on this subject considers non-similar object classes, with the aim of learning a general representation bias for a large family of classes. Such a general bias for digit recognition was learned in [101] using a prior over alignment transformations, and in [73] using a distance function. For object class recognition, several approaches were suggested which learn a general bias through some form of intermediate representation [9, 108, 137]. In [137] 21 object categories are learned using 'joint boosting', which encourages usage of patch features common to several categories. It is estimated that the number of features required to achieve a certain performance level grows only logarithmically with the number of classes. In [108] an intermediate representation of 'compositions', which are ensembles of proximate patch features, is learned from all object classes together. These compositions are then used as parts in a relational part based model. The model is applied in feed-forward manner, and it is shown to perform much better than a simple bag-of-features composed of the original patches. Bart and Ullman [9] suggested representing a new class image using its similarities to already learned classes, as measured using soft value classifiers trained previously. Nearest neighbor classification in this 'similarities space' representation enables one-shot learning (learning from a single image). A considerable improvement is obtained over independent classifiers trained in isolation from two images (object and background) each.

Several papers have considered learning a general recognition bias via a prior over generative models [74, 91]. In [91] a prior is learned over the parameters of the constellation model from [54], and the model averaging in the classification stage is approximated using a variational technique. The parameter prior for each class is learned from Maximum likelihood models of the other classes. In [74] a two-stage extension of the constellation model approach is suggested. Here the application of the model to an image makes a (soft) selection of relevant image features, and those are then used in a discriminative SVM classification. The method is applied to all the Caltech-101 data set, with the generative models learned from airplanes and faces alone. Clearly the useful knowledge transferred via these models can only be of a very general nature, mostly related to natural object

fragment statistics.

Knowledge transfer between similar tasks. Several recent approaches to learning from similar classes were recently suggested. In [8] Bart and Ullman studied an approach to one-shot learning in which fragment selection for a new class model is done based on similarity with fragments chosen for other classes trained earlier. Unlike in the methods presented in the previous paragraph, only classes which are similar to the new class affect the construction of the new model. In [53] Ferench et al. considered an object identification task, where many small subclasses (corresponding to specific objects) from a known object category have to be identified. Their approach is based on learning a binary distance function, accepting a pair of images and determining whether they have the same identity or not. The distance function decision is based on a set of parts, defined by expected appearance and absolute position (the objects in the images are roughly aligned). The method is applied to faces and cars. An improved discriminative learning technique for a similar binary distance is suggested in [78].

Distinction between similar classes based on a model of the joint class is considered in [51],[E]. In [51] it is claimed that distinction between similar classes is usually based on small features, whose identity can only be determined based on their location with respect to larger features shared by all the classes. For example, an earring may be an important cue for gender recognition. Such features are termed 'satellite features' and the large, stable features are termed 'anchor' features. Recognition and learning in the suggested system consist of three distinct phases, where first anchor patches are found (learnt), followed by detection of satellite features and finally, naive base classification based on the satellite features. The method is tested on cars and faces. In [E] a similar, but more general idea is employed, where the notion of a 'joint class' is equated with 'basic-level class' in the cognitive psychology sense, and the task is posed as a distinction between sub-classes. A two stage method is suggested, where first the object parts are identified using a part-based model of the basic-class (learned using the algorithm from [D]), and then sub-class SVM discrimination is done using a vector of part descriptions. The method is tested on 6 different basic classes, and shown to have a performance advantage over independent classifier learning. While this approach does not include an explicit 'anchor' vs. 'satellite' parts distinction, these part categories roughly coincide with parts learned at earlier and late rounds of the algorithm from [D] respectively. This is because the parts first chosen by the boosting algorithm in [D] are the most characteristic of the class, in terms of both appearance and spatial relations, and the parts found later are less stable and characteristically found only in a subset of the object images. The discrimination relevance of satellite features, shown in [51] can therefore explain the improved sub-class discrimination of the method when large amounts of model parts are used (in contrast, for example, with the performance of basic class recognition, which typically requires half the number of parts).

Chapter 2

Distance Learning with equivalence constraints

This chapter contains the following research papers:

[A] A. Bar-Hillel and D. Weinshall: “Learning with equivalence constraints, and the relation to multiclass Classification ”, in *the Sixteenth Annual Conference On Learning Theory (COLT)*, 640-654, Springer 2003.

[B] A. Bar-Hillel, T. Hertz, N. Shental and D. Weinshall: “Learning a Mahalanobis metric from equivalence constraints”, in *Journal of Machine Learning Research* 6(Jun): 937-965, 2005.

[F] A. Bar-Hillel and D. Weinshall: “Learning distance function by coding similarity”, Technical report, 2006.

Papers [A] and [B] are publications, and they appear in sections 2.1, 2.2 in their original publication format. The technical report [F] was not published, and it appears here in section 2.3

Learning with Equivalence Constraints, and the Relation to Multiclass Learning

Aharon Bar-Hillel and Daphna Weinshall

School of Computer Sci. and Eng. & Center for Neural Computation
Hebrew University, Jerusalem 91904, Israel
{aharonbh,daphna}@cs.huji.ac.il
WWW home page: http://www.ca.huji.ac.il/~daphna

Abstract. We study the problem of learning partitions using equivalence constraints as input. This is a binary classification problem in the product space of pairs of datapoints. The training data includes pairs of datapoints which are labeled as coming from the same class or not. This kind of data appears naturally in applications where explicit labeling of datapoints is hard to get, but relations between datapoints can be more easily obtained, using, for example, Markovian dependency (as in video clips).
Our problem is an unlabeled partition problem, and is therefore tightly related to multiclass classification. We show that the solutions of the two problems are related, in the sense that a good solution to the binary classification problem entails the existence of a good solution to the multiclass problem, and vice versa. We also show that bounds on the sample complexity of the two problems are similar, by showing that their relevant 'dimensions' (VC dimension for the binary problem, Natarajan dimension for the multiclass problem) bound each other. Finally, we show the feasibility of solving multiclass learning efficiently by using a solution of the equivalent binary classification problem. In this way advanced techniques developed for binary classification, such as SVM and boosting, can be used directly to enhance multiclass learning.

1 Introduction

Multiclass learning is about learning a concept over some input space, which takes a discrete set of values $\{0, 1, \ldots, \mathbf{M}-1\}$. A tightly related problem is data partitioning, which is about learning a partitioning of data to $\mathbf{M}$ discrete sets. The latter problem is equivalent to unlabelled multiclass learning, namely, all the multiclass concepts which produce the same partitioning but with a different permutation of labels are considered the same concept.

Most of the work on multiclass partitioning of data has focused on the first variant, namely, the learning of an explicit mapping from datapoints to $\mathbf{M}$ discrete labels. It is assumed that the training data is obtained in the same form, namely, it is a set of datapoints with attached labels taken from the set $\{0, 1, \ldots, \mathbf{M}-1\}$. On the other hand, unlabeled data partitioning requires as

training data only equivalence relations between pairs of datapoints; namely, for each pair of datapoints a label is assigned to indicate whether the pair originates from the same class or not. While it is straightforward to generate such binary labels on pairs of points from multiclass labels on individual points, the other direction is not as simple.

It is therefore interesting to note that equivalence constraints between pairs of datapoints may be easier to obtain in many real-life applications. More specifically, in data with natural Markovian dependency between successive datapoints (e.g., a video clip), there are automatic means to determine whether two successive datapoints (e.g., frames) come from the same class or not. In other applications, such as distributed learning where labels are obtained from many uncoordinated teachers, the subjective labels are meaningless, and the major information lies in the equivalence constraints which the subjective labels impose on the data. More details are given in [12].

Multiclass classification appears like a straightforward generalization of the binary classification problem, where the concept takes only two values $\{0, 1\}$. But while there is a lot of work on binary classification, both theoretical and algorithmic, the problem of multiclass learning is less understood. The VC dimension, for example, can only be used to characterize the learnability and sample complexity of binary functions. Generalizing this notion to multiclass classification has not been straightforward; see [4] for the details regarding a number of such generalizations and the relations between them.

On a more practical level, most of the algorithms available are best suitable (or only work for) the learning of binary functions. Support vector machines (SVM) [14] and boosting techniques [13] are two important examples. A possible solution is to reduce the problem to the learning of a number of binary classifiers ($O(\mathbf{M})$ or $O(\mathbf{M^2})$), and then combine the classifiers using for example a winner-takes-all strategy [7]. The use of error correcting code to combine the binary classifiers was first suggested in [5]. Such codes were used in several successful generalizations to existing techniques, such as multiclass SVM and multiclass boosting [6, 1]. These solutions are hard to analyze, however, and only recently have we started to understand the properties of these algorithms, such as their sample complexity [7]. Another possible solution is to assume that the data distribution is known and construct a generative model, e.g., a Gaussian mixture model. The main drawback of this approach is the strong dependence on the assuption that the distribution is known.

In this paper we propose a different approach to multiclass learning. For each multiclass learning problem, define an equivalent binary classification problem. Specifically, if the original problem is to learn a multiclass classifier over data space X, define a binary classification problem over the product space $X \times X$, which includes all pairs of datapoints. In the binary classification problem, each pair is assigned the value 1 if the two datapoints come from the same class, and 0 otherwise. Hence the problem is reduced to the learning of a *single* binary classifier, and any existing tool can be used. Note that we have eliminated the problem of combining $\mathbf{M}$ binary classifiers. We need to address, however,

the problems of how to generate the training sample for the equivalent binary problem, and how to obtain a partition of X from the learned concept in the product space.

A related idea was explored algorithmically in [11], where multiclass learning was translated to a binary classification problem over the *same* space X, using the difference between datapoints as input. This embedding in rather problematic, however, since the binary classification problem is ill-defined; it is quite likely that the same value would correspond to the difference between two vectors from the same class, and the difference between two other vectors from two different classes.

In the rest of this paper we study the properties of the binary classification problem in the product space, and their relation to the properties of the equivalent multiclass problem. Specifically, in Section 2.1 we define, given a multiclass problem, the equivalent binary classification problem, and state its sample complexity using the usual PAC framework. In Section 2.2 we show that for any solution of the product space problem with error e_{pr}, there is a solution of the multiclass problem with error e_o, such that

$$\frac{e_{pr}}{2} < e_o < \sqrt{2\mathbf{M}e_{pr}}$$

However, under mild assumptions, a stronger version for the right inequality exists, showing that the errors in the original and the product space are lineary related:

$$e_o < (\frac{e_{pr}}{K})$$

where K is the frequency of the smallest class. Finally, in Section 2.3 we show that the sample complexity of the two problems is similar in the following sense: for S_N the Natarajan dimension of the the multiclass problem, S_{VC} the VC-dimension of the equivalent binary problem, and $\mathbf{M}$ the number of classes, the following relation holds

$$\frac{S_N}{f_1(\mathbf{M})} - 1 \leq S_{VC} \leq f_2(\mathbf{M})S_N$$

where $f_1(\mathbf{M})$ is $O(\mathbf{M}^2)$ and $f_2(\mathbf{M})$ is $O(log\mathbf{M})$.

In order to solve a multiclass learning problem by solving the equivalent binary classification problem in the product space, we need to address two problems. First, a sample of independent points in X does not generate an independent sample in the product space. We note, however, that every n independent points in X trivially give $\frac{n}{2}$ independent pairs in the product space, and therefore the bounds above still apply up to a factor of $\frac{1}{2}$. We believe that the bounds are actually better, since a sample of n independent labels gives an order of $\mathbf{M}n$ non-trivial labels on pairs. By non-trivial we mean that given less than $\mathbf{M}n$ labels on pairs of points from $\mathbf{M}$ classes of the same size, we cannot deterministically derive the labels of the remaining pairs. This problem is more acute in

the other direction, namely, it is actually not possible to generate a set of labels on individual points from a set of equivalence constraints on pairs of points.

Second, and more importantly, the approximation we learn in the product space may not represent any partition. A binary product space function f represents a partition only if it is an indicator of an equivalence relation, i.e. the relation $f(x1, x2) = 1$ is reflexive, symetric and transitive. It can be readily shown that f represents a partition, i.e., $\exists g, s.t. f = U(g))$ iff the function $1 - f$ is a binary metric. While this condition holds for our target concept, it doesn't hold for its approximation in the general case, and so an approximation will not induce any obvious partition on the original space.

To address this problem, we show in section 3 how an ε-good hypothesis f in the product space can be used to build an original space classifier with error linear in ε. First we show how f enables us, under certain conditions, to partition data in the original space with error linear in ε. Given the partitioning, we claim that a classifier can be built by using f to compare new presented data points to the partitioned sample. A similar problem was studied in [2], using the same kind of approximation. However, different criteria are optimized in the two papers: in [2] $e_{pr}(\bar{g}, f)$ is minimized (i.e., the product space error), while in our work a partition g is sought which minimizes $e_o(g, c)$ (i.e., the error in the original space of g w.r.t. the original concept).

2 From M-partitions to binary classifiers

In this section we show that multiclass classification can be translated to binary classification, and that the two problems are equivalent in many ways. First, in section 2.1 we formulate the binary classification problem whose solution is equivalent to a given multiclass problem. In section 2.2 we show that the solutions of the two problems are closely related: a good hypothesis for the multiclass problem provides a good hypothesis for the equivalent binary problem, and vice versa. Finally, in section 2.3 we show that the sample complexity of the two problems is similar.

2.1 PAC framework in the product space

Let us introduce the following notations:

- X: the input space.
- $\mathbf{M}$: the number of classes.
- D: the sampling distribution (measure) over X.
- c: a target concept over X; it is a labeled partition of X, $c : X \to \{0, \ldots, \mathbf{M}-1\}$. For each such concept, $c^{-1}(j) \in X$ denotes the cluster of points labeled j by c.
- $\mathcal{H}$: a family of hypotheses; each hypothesis is a function $h : X \to \{0, \ldots, \mathbf{M}-1\}$.

– $e(h, c)$: the error in X of a hypothesis $h \in \mathcal{H}$ with respect to c, defined as

$$e(h, c) = D(c(x) \neq h(x))$$

Given an unknown target concept c, the learning task is to find a hypothesis $h \in \mathcal{H}$ with low error $e(h, c)$. Usually it is assumed that a set of labeled datapoints is given during training. In this paper we do not assume to have access to such training data, but instead have access to a set of labeled equivalence constraints on pairs of datapoints. The label tells us whether the two points come from the same (unknown) class, or not. Therefore the learning problem is transformed as follows:

For any hypothesis h (or c), define a functor U which takes the hypothesis as an argument and outputs a function $\bar{h}, \bar{h} : X \times X \rightarrow \{0, 1\}$. Specifically:

$$\bar{h}(x, y) = 1_{h(x)=h(y)}$$

Thus $\bar{h}$ expresses the implicit equivalence relations induced by the concept h on pairs of datapoints.

The functor U is not injective: two different hypotheses h_1 and h_2 may result in the same $\bar{h}$. This, however, happens only when h_1 and h_2 differ only by a permutation of their corresponding labels, while representing the same partition; $\bar{h}$ therefore represents an unlabeled partition.

We can now define a second notion of error between unlabeled partitions over the product space $X \times X$:

$$e(\bar{h}, \bar{c}) = D \times D(\bar{h}(x, y) \neq \bar{c}(x, y))$$

where $\bar{c}$ is obtained from c by the functor U. This error measures the probability of disagreement between $\bar{h}$ and $\bar{c}$ with regard to equivalence queries. It is a rather intuitive measure for the comparison of unlabeled partitions. The problem of learning a partition can now be cast as a regular PAC learning problem, since $\bar{h}$ and $\bar{c}$ are binary hypotheses. Specifically:

> Let $X \times X$ denote the input space, $D \times D$ denote the sampling probability over the input space, and $\bar{c}$ denote the target concept. Let the hypotheses family be the family $\bar{\mathcal{H}} = \{\bar{h} : h \in \mathcal{H}\}$.[1]

Now we can use the VC dimension and PAC-learning theory on sample complexity, in order to characterize the sample complexity of learning the binary hypothesis $\bar{h}$. More interestingly, we can then compare our results with results on sample complexity obtained directly for the multiclass problem.

2.2 The Connection between solution quality of the two problems

In this section we show that a good (binary) hypothesis in the product space can be used to find a good hypothesis (partition) in the original space, and

[1] Note that $\bar{\mathcal{H}}$ is of the same size as $\mathcal{H}$ only when $\mathcal{H}$ does not contain hypotheses which are identical with respect to the partition of X.

vice versa. Note that the functor U, which connects hypotheses in the original space to hypotheses in the product space, is not injective, and therefore it has no inverse. Therefore, in order to asses the difference between two hypotheses $\bar{h}$ and $\bar{c}$, we must choose h and c such that $\bar{h} = U(h)$ and $\bar{c} = U(c)$, and subsequently compute $e(h, c)$.

We proceed by showing three results: Thm. 1 shows that in general, if we have a hypothesis in product space with some error ε, there is a hypothesis in the original space with error $O(\sqrt{M\varepsilon})$. However, if ε is small with respect to the smallest class probability K, Thm. 2 shows that the bound is linear, namely, there is a hypothesis in the original space with error $O(\frac{\varepsilon}{K})$. In most cases, this is the range of interest. Finally, Thm. 3 shows the other direction: if we have a hypothesis in the original space with some error ε, its product space hypothesis $U(h) = \bar{h}$ has an error smaller than 2ε.

Before proceeding we need to introduce some more notations: Let c and h denote two partitions of X into $\mathbf{M}$ classes. Define the joint distribution matrix $P = \{p_{ij}\}_{i,j=0}^{\mathbf{M}-1}$ as follows:

$$p_{ij} \stackrel{\Delta}{=} D(c(x) = i, h(x) = j)$$

Using this matrix we can express the probability of error in the original space and the product space.

1. The error in X is

$$e(h, c) = D(c(x) \neq h(x)) = \sum_{i=0}^{\mathbf{M}-1} \sum_{j \neq i} p_{ij}$$

2. The error in the product space is

$$\begin{aligned} e(\bar{h}, \bar{c}) &= D \times D([\bar{c}(x, y) = 1 \wedge \bar{h}(x, y) = 0] \vee [\bar{c}(x, y) = 0 \wedge \bar{h}(x, y) = 1]) \\ &= \sum_{i=0}^{\mathbf{M}-1} \sum_{j=0}^{\mathbf{M}-1} D(c(x) = i, h(x) = j) \cdot (D(\{y | c(y) = i, h(y) \neq j\}) \\ &\qquad + D(\{y | c(y) \neq i, h(y) = j\})) \\ &= \sum_{i=0}^{\mathbf{M}-1} \sum_{j=0}^{\mathbf{M}-1} p_{ij} \left(\sum_{k \neq i} p_{kj} + \sum_{k \neq j} p_{ik} \right) \end{aligned}$$

Theorem 1. *For any two product space hypotheses $\bar{h}, \bar{c}$, there are h, c such that $\bar{h} = U(h), \bar{c} = U(c)$ and*

$$e(h, c) \leq \sqrt{2\mathbf{M} e(\bar{h}, \bar{c})}$$

where $\mathbf{M}$ is the number of equivalence classes of $\bar{h}, \bar{c}$.

The proof appears in the technical report [3], appendix A. We note that the bound is tight as a function of ε since there are indeed cases where $e(c,h) = O(\sqrt{(}e(\bar{h},\bar{c}))$. A simple example of such 3-class problem occurs when the matrix of joint distribution is the following:

$$P = \begin{pmatrix} 1-3q & 0 & 0 \\ 0 & q & q \\ 0 & 0 & q \end{pmatrix}$$

Here $e(c,h) = q$ and $e(\bar{c},\bar{h}) = 4q^2$. The next theorem shows, however, that this situation cannot occur if $e(\bar{c},\bar{h})$ is small compared to the smallest class frequency.

Theorem 2. *Let c denote a target partition and h a hypothesis, and let $\bar{c},\bar{h}$ denote the corresponding hypotheses in the product space. Denote the size of the minimal class of c by $K = \min_{i\in\{0,\ldots,\mathbf{M}-1\}} D(c^{-1}(i))$, and the product space error $\varepsilon = e(\bar{c},\bar{h})$.*

$$\varepsilon < \frac{K^2}{2} \quad \Longrightarrow \quad e(f\circ h, c) < \frac{\varepsilon}{K} \tag{1}$$

where $f : \{0,\ldots,\mathbf{M}-1\} \to \{0,\ldots,\mathbf{M}-1\}$ is a bijection matching the labels of h and c.

Proof. We start by showing that if the theorem's condition holds, then there is a natural correspondence between the classes of c and h:

Lemma 1. *If the condition in (1) holds, then there exists a bijection $J : \{0,\ldots,\mathbf{M}-1\} \to \{0,\ldots,\mathbf{M}-1\}$ such that*

- $p_{i,J(i)} > \sqrt{\frac{\varepsilon}{2}}$
- $p_{i,l} < \sqrt{\frac{\varepsilon}{2}}$ *for all* $l \neq J(i)$
- $p_{l,J(i)} < \sqrt{\frac{\varepsilon}{2}}$ *for all* $l \neq i$

Proof. Denote the class probabilities as $p_i^c = D(c^{-1}(i))$; clearly

$$p_i^c = \sum_{j=0}^{\mathbf{M}-1} p_{ij}$$

We further define for each class i of c its internal error $\varepsilon_i = \sum_{j=0}^{\mathbf{M}-1} p_{ij}(p_i^c - p_{ij})$. The rationale for this definition follows from the following inequality:

$$\varepsilon = \sum_{i=0}^{\mathbf{M}-1}\sum_{j=0}^{\mathbf{M}-1} p_{ij}\Big(\sum_{\substack{k=0\\k\neq i}}^{\mathbf{M}-1} p_{kj} + \sum_{\substack{k=0\\k\neq j}}^{\mathbf{M}-1} p_{ik}\Big) \geq \sum_{i=0}^{\mathbf{M}-1}\sum_{j=0}^{\mathbf{M}-1} p_{ij}(p_i^c - p_{ij}) = \sum_{i=0}^{\mathbf{M}-1} \varepsilon_i$$

We first observe that each row in matrix P contains at least one element bigger than $\sqrt{\frac{\varepsilon}{2}}$. Assume to the contrary that no such element exists in class i; then

$$\varepsilon \geq \varepsilon_i = \sum_{j=0}^{\mathbf{M}-1} p_{ij}(p_i^c - p_{ij}) > \sum_{j=0}^{\mathbf{M}-1} p_{ij}(\sqrt{2\varepsilon} - \sqrt{\frac{\varepsilon}{2}}) = \sqrt{\frac{\varepsilon}{2}}\sum_{j=0}^{\mathbf{M}-1} p_{ij} \geq \sqrt{\frac{\varepsilon}{2}}\cdot\sqrt{2\varepsilon} = \varepsilon$$

in contradiction.

Second, we observe that the row element bigger than $\sqrt{\frac{\varepsilon}{2}}$ is unique. This follows from the following argument: for any two elements p_{ij_1}, p_{ij_2} in the same row:

$$\varepsilon \geq \sum_{j=0}^{\mathbf{M}-1} p_{ij}(\sum_{\substack{k=0 \\ k\neq i}}^{\mathbf{M}-1} p_{kj} + \sum_{\substack{k=0 \\ k\neq j}}^{\mathbf{M}-1} p_{ik}) \geq \sum_{j=0}^{\mathbf{M}-1} p_{ij} \sum_{\substack{k=0 \\ k\neq j}}^{\mathbf{M}-1} p_{ik} \geq 2p_{ij_1}p_{ij_2}$$

Hence it is not possible that both the elements p_{ij_1} and p_{ij_2} are bigger than $\sqrt{\frac{\varepsilon}{2}}$. The uniqueness of an element bigger than $\sqrt{\frac{\varepsilon}{2}}$ in a column follows from an analogous argument with regard to the columns, which completes the proof of the lemma.

Denote $f = J^{-1}$, and let us show that $\sum_{i=0}^{\mathbf{M}-1} p_{i,f(i)} > 1 - \frac{\varepsilon}{K}$. We start by showing that $p_{i,f(i)}$ cannot be 'too small':

$$\varepsilon_i = \sum_{j=0}^{\mathbf{M}-1} p_{ij}(p_i^c - p_{ij}) = p_{i,f(i)}(p_i^c - p_{i,f(i)}) + \sum_{\substack{j=0 \\ j\neq f(i)}}^{\mathbf{M}-1} p_{ij}(p_i^c - p_{ij})$$

$$\geq p_{i,f(i)}(p_i^c - p_{i,f(i)}) + \sum_{\substack{j=0 \\ j\neq f(i)}}^{\mathbf{M}-1} p_{ij}p_{i,f(i)} = 2p_{i,f(i)}(p_i^c - p_{i,f(i)})$$

This gives a quadratic inequality

$$p_{i,f(i)}^2 - p_i^c p_{i,f(i)} + \frac{\varepsilon_i}{2} \geq 0$$

which holds for $p_{i,f(i)} \geq \frac{p_i^c + \sqrt{(p_i^c)^2 - 2\varepsilon_i}}{2}$ or for $p_{i,f(i)} \leq \frac{p_i^c - \sqrt{(p_i^c)^2 - 2\varepsilon_i}}{2}$. Since

$$\sqrt{(p_i^c)^2 - 2\varepsilon_i} = \sqrt{(p_i^c)^2(1 - \frac{2\varepsilon_i}{(p_i^c)^2})} = p_i^c\sqrt{1 - \frac{2\varepsilon_i}{(p_i^c)^2}} > p_i^c(1 - \frac{2\varepsilon_i}{(p_i^c)^2}) = p_i^c - \frac{2\varepsilon_i}{p_i^c}$$

it must hold that either $p_{i,f(i)} > p_i^c - \frac{\varepsilon_i}{p_i^c}$ or $p_{i,f(i)} < \frac{\varepsilon_i}{p_i^c}$. But the second possiblity that $p_{i,f(i)} < \frac{\varepsilon_i}{p_i^c}$ leads to contradiction with condition (1) since

$$\sqrt{\frac{\varepsilon}{2}} < p_{i,f(i)} < \frac{\varepsilon_i}{p_i^c} \leq \frac{\varepsilon}{K} \quad \Longrightarrow \quad K < \sqrt{2\varepsilon}$$

Therefore $p_{i,f(i)} > p_i^c - \frac{\varepsilon_i}{p_i^c}$.

Summing the inequalities over i, we get

$$\sum_{i=0}^{\mathbf{M}-1} p_{i,f(i)} > \sum_{i=0}^{\mathbf{M}-1} p_i^c - \frac{\varepsilon_i}{p_i^c} \geq 1 - \sum_{i=0}^{\mathbf{M}-1} \frac{\varepsilon_i}{K} \geq 1 - \frac{\varepsilon}{K}$$

This completes the proof of the theorem since

$$e(J\circ h,c)=p(J\circ h(x)\neq c(x))=1-p(J\circ h(x)=c(x))$$
$$=1-\sum_{i=0}^{\mathbf{M}-1}p(\{c(x)=i,h(x)=J^{-1}(i))=1-\sum_{i=0}^{\mathbf{M}-1}p_{i,f(i)}<\frac{\varepsilon}{K}=\frac{e(\bar{c},\bar{h})}{K}$$

Corollary 1. *If the classes are equiprobable, namely* $\frac{1}{K}=\mathbf{M}$*, we get a bound of* $\mathbf{M}\varepsilon$ *on the error in the original space.*

Corollary 2. *As* $K\to\sqrt{2\varepsilon}$*, the lowest allowed value according to the theorem condition, we get an error bound approaching* $\frac{\varepsilon}{\sqrt{2\varepsilon}}=\sqrt{\frac{\varepsilon}{2}}$*. Hence the linear behavior of the bound on the original space error is lost near this limit, in accordance with Thm. 1.*

A bound in the other direction is much simpler to achieve:

Theorem 3. *For every two labeled partitions* h,c*: if* $e(h,c)<\varepsilon$ *then* $e(\bar{h},\bar{c})<2\varepsilon$*.*

Proof.

$$e(\bar{h},\bar{c})=\sum_{i=0}^{\mathbf{M}-1}\sum_{j=0}^{\mathbf{M}-1}p_{ij}\left[\sum_{k\neq i}p_{kj}+\sum_{k\neq j}p_{ik}\right]$$
$$=\sum_{i=0}^{\mathbf{M}-1}p_{ii}[\sum_{k\neq i}p_{ki}+\sum_{k\neq i}p_{ik}]+\sum_{i=0}^{\mathbf{M}-1}\sum_{j\neq i}p_{ij}[\sum_{k\neq i}p_{kj}+\sum_{k\neq j}p_{ik}]$$
$$\leq\sum_{i=0}^{\mathbf{M}-1}p_{ii}\cdot\varepsilon+\sum_{i=0}^{\mathbf{M}-1}\sum_{j\neq i}p_{ij}\leq\varepsilon+\varepsilon=2\varepsilon$$

2.3 The connection between sample size complexity

Several dimension-like measures of the sample complexity exist for multiclass probelms. However, these measures can be shown to be closely related [4]. We use here the Natarajan dimension, denoted as $S_N(\mathcal{H})$, to characterize the sample size complexity of the hypotheses family $\mathcal{H}$ [10, 4]. Since $\bar{\mathcal{H}}$ is binary, its sample size is characterized by its VC dimension $S_{VC}(\bar{\mathcal{H}})$ [14]. We will now show that each of these dimensions bounds the other up to a scaling factor which depends on $\mathbf{M}$. Specifically, we will prove the following double inequality:

$$\frac{S_N(\mathcal{H})}{f_1(\mathbf{M})}-1\leq S_{VC}(\bar{\mathcal{H}})\leq f_2(\mathbf{M})S_N(\mathcal{H}) \tag{2}$$

where $f_1(\mathbf{M})=O(\mathbf{M}^2)$ and $f_2(\mathbf{M})=O(log\mathbf{M})$.

Theorem 4. *Let $S_N^U(\mathcal{H})$ denote the uniform Natarajan dimension of $\mathcal{H}$ as defined by Ben-David et al. [4]; then*

$$S_N^U(\mathcal{H}) - 1 \leq S_{VC}(\bar{\mathcal{H}})$$

Proof. Let d denote the uniform Natarajan dimension, $d = S_N^U(\mathcal{H})$. It follows that there are $k, l \in \{0, \ldots, \mathbf{M}-1\}$ and $\{x_i\}_{i=1}^d$ points in X such that

$$\{0,1\}^d \subseteq \{(\psi_{k,l} \circ h(x_1), \ldots, \psi_{k,l} \circ h(x_d)) | h \in \mathcal{H}\}$$

where $\psi_{k,l} : \{0, \ldots, \mathbf{M}-1\} \to \{0, 1, *\}$, $\psi_{k,l}(k) = 1$, $\psi_{k,l}(l) = 0$, and $\psi_{k,l}(u) = *$ for every $u \neq k, l$.

Next we show that the set of product space points $\{\bar{x}_i = (x_i, x_{i+1})\}_{i=1}^{d-1}$ is VC-shattered by $\bar{\mathcal{H}}$. Assume an arbitrary $\bar{b} \in \{0,1\}^{d-1}$. Since by definition $\{x_i\}_{i=1}^d$ is $\psi_{k,l}$-shattered by $\mathcal{H}$, we can find $h \in \mathcal{H}$ which assigns $h(x_1) = l$ and gives the following assignments over the points $\{x_i\}_{i=2}^d$:

$$h(x_i) = \begin{pmatrix} k \; if \;\; h(x_{i-1}) = l \;\; and \; \bar{b}(i-1) = 0 \\ l \;\; if \;\; h(x_{i-1}) = l \;\; and \; \bar{b}(i-1) = 1 \\ l \;\; if \; h(x_{i-1}) = k \; and \; \bar{b}(i-1) = 0 \\ k \; if \; h(x_{i-1}) = k \; and \; \bar{b}(i-1) = 1 \end{pmatrix}$$

By construction $(\bar{h}(\bar{x}_1), \ldots, \bar{h}(\bar{x}_{d-1})) = \bar{b}$. Since $\bar{b}$ is arbitrary, $\{\bar{x}_i\}_{i=1}^{d-1}$ is shattered by $\bar{\mathcal{H}}$, and hence $S_{vc}(\bar{\mathcal{H}}) \geq d - 1$.

The relation between the uniform Natarajan dimension and the Natarajan dimension is given by theorem 7 in [4]. In our case it is

$$S_N(H) \leq \frac{M(M-1)}{2} S_N^U(H)$$

Hence the proof of theorem 4 gives us the left bound of inequality 2.

Theorem 5. *Let $d_{pr} = S_N(\mathcal{H})$ denote the Natarajan dimension of $\mathcal{H}$, and $d_o = S_{VC}(\bar{\mathcal{H}})$ denote the VC dimension of $\bar{\mathcal{H}}$. Then*

$$S_{VC}(\bar{\mathcal{H}}) \leq 4.87 S_N(\mathcal{H}) \log(\mathbf{M}+1)$$

Proof. Let $X_{pr} = \{\bar{x}_i = (x_i^1, x_i^2)\}_{i=1}^{d_{pr}}$ denote a set of points in the product space which are shattered by $\bar{\mathcal{H}}$. Let $X_o = \{x_1^1, x_1^2, x_2^1, \ldots, x_{d_{pr}}^1, x_{d_{pr}}^2\}$ denote the corresponding set of points in the original space.

There is a set $Y_{pr} = \{\bar{h}_j\}_{j=1}^{2^{d_{pr}}}$ of $2^{d_{pr}}$ hypotheses in $\bar{\mathcal{H}}$, which are different from each other on X_{pr}. For each hypothesis $\bar{h}_j \in Y_{pr}$ there is a hypothesis $h \in \mathcal{H}$ such that $\bar{h} = U(h)$. If $\bar{h}_1 \neq \bar{h}_2 \in Y_{pr}$ then the corresponding h_1, h_2 are different on X_o. To see this, note that $\bar{h}_1 \neq \bar{h}_2$ implies the existence of $\bar{x}_i = (x_i^1, x_i^2) \in X_{pr}$ on which $\bar{h}_1(\bar{x}_i) \neq \bar{h}_2(\bar{x}_i)$. It is not possible in this case that both $h_1(x_i^1) = h_2(x_i^1)$ and $h_1(x_i^2) = h_2(x_i^2)$. Hence there are $2^{d_{pr}}$ hypotheses in $\mathcal{H}$ which are different on X_o, from which it follows that

$$|\{(h(x_1^1), h(x_1^2), \ldots, h(x_{d_{pr}}^1), h(x_{d_{pr}}^2)) | h \in \mathcal{H}\}| \geq 2^{d_{pr}} \tag{3}$$

The existence of an exponential number of assignments of $\mathcal{H}$ on the set X_o is not possible if $|X_o|$ is much larger than the Natarajan dimension of $\mathcal{H}$. We use Thm. 9 in [4] (proved in [8]) to argue that if the Natarajan dimension of $\mathcal{H}$ is d_o, then

$$|\{(h(x_1^1), h(x_1^2), \ldots, h(x_{d_{pr}}^1), h(x_{d_{pr}}^2)) | h \in \mathcal{H}\}| \leq (\frac{2d_{pr}e(\mathbf{M}+1)^2}{2d_o})^{d_o} \qquad (4)$$

where $\mathbf{M}$ is the number of classes. Combining (3) and (4) we get

$$2^{d_{pr}} \leq (\frac{2d_{pr}e(\mathbf{M}+1)^2}{2d_o})^{d_o}$$

Here the term on left side is exponential in d_{pr}, and the term on the right side is polynomial. Hence the inequality cannot be true asymptotically and d_{pr} is bounded.

We can find a convenient bound by following the proof of Thm. 10 in [4]. The algebraic details completing the proof are left for the technical report [3], appendix B.

Corollary 3. *$\bar{\mathcal{H}}$ is learnable iff $\mathcal{H}$ is learnable.*

3 From product space approximations to original space classifiers

In section 3.1 we present an algorithm to partition a data set Y using a product space function which is ε-good over $Y \times Y$. f should only satisfy $e(f, \bar{c}) < \varepsilon$, but it doesn't have to be an equivalence relation indicator, and so in general there is no h such that $f = U(h)$. The partition generated is shown to have an error linear in ε. Then in section 3.2 we briefly discuss (without proof) how an ε-good product space hypothesis can be used to build a classifier with error $O(\varepsilon)$.

3.1 Partitioning using a product space hypothesis

Assume we are given a data set $Y = \{x_i\}_{i=1}^N$ of points drawn independently from the distribution over X. Let f denote a learned hypothesis from $\bar{\mathcal{H}}$, and denote the error of f over the product space $Y \times Y$ by

$$\varepsilon = e(\bar{c}, f) = \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} 1_{\bar{c}(x_i, y_j) \neq f(x_i, y_j)}$$

Denote by K the frequency $\frac{classsize}{N}$ of the smallest class in Y.

Note that since no explicit labels are given, we can only hope to find an approximation to c over Y up to a permutation of the labels. The following theorem shows that if ε is small enough compared to K and given f, there is a simple algorithm which is guaranteed to achieve an approximation to the partition represented by concept c with error linear in ε.

Theorem 6. *Using the notation defined above, if the following condition hold*

$$\varepsilon < \frac{K^2}{6}, \tag{5}$$

then we can find a partition g of Y with a simple procedure, such that $e(c, J \circ g) < \frac{6\varepsilon}{K}$. J here denotes a permutation $J : \{0, \ldots, \mathbf{M}-1\} \to \{0, \ldots, \mathbf{M}-1\}$ matching the labels of c and g.

In order to present the algorithm and prove the error bound as stated above, we first define several simple concepts.

Define the 'fiber' of a point $x \in Y$ under a function $h : X \times X \to \{0, 1\}$ as the following restriction of h:

$$fiber^h(x) : Y \to \{0, 1\}, \quad [fiber^h(x)](y) = h(x, y)$$

$fiber^h(x)$ is an indicator function of the points in Y which are in the same class with x according to h.

Let us now define the distance between two fibers. For two indicator functions $I_1, I_2 : Y \to \{0, 1\}$ let us measure the distance between them using the $L1$ metric over Y:

$$d(I_1, I_2) = Prob(I_1(x) \neq I_2(x)) = \frac{1}{N} \sum_{i=1}^{N} 1_{I_1(x_i) \neq I_2(x_i)} = \frac{1}{N} \sum_{i=1}^{N} |I_1(x_i) - I_2(x_i)|$$

Given two fibers $fiber^h(x), fiber^h(z)$ of a product space hypothesis, the L_1 distance between them has the form of

$$d(fiber^h(x), fiber^h(z)) = \frac{\#(Nei^h(x) \Delta Nei^h(z))}{N}$$

where $Nei^h(x) = \{y | h(x, y) = 1\}$. This gives us an intuitive meaning to the inter-fiber distance, namely, it is the frequency of sample points which are neighbors of x and not of z or vice versa.

The operator taking a point $x \in Y$ to $fiber^h(x)$ is therefore an embedding of Y in the metric space $L_1(Y)$. In the next lemma we see that if the conditions of Thm. 6 hold, most of the data set is well separated under this embedding, in the sense that points from the same class are near while points from different classes are far. This allows us to define a simple algorithm which uses this separability to find a good partitioning of Y, and prove that its error is bounded as required.

Lemma 2. *There is a set of 'good' points $\mathcal{G} \in Y$ such that $|Y \backslash \mathcal{G}| \leq \frac{3\varepsilon}{K} N$ (i.e., the set is large), and for every two points $x, y \in \mathcal{G}$:*

$$c(x) = c(y) \implies d(fiber^f(x), fiber^f(y)) < \frac{2K}{3}$$
$$c(x) \neq c(y) \implies d(fiber^f(x), fiber^f(y)) \geq \frac{4K}{3}$$

Proof. Define the 'good' set $\mathcal{G}$ as

$$\mathcal{G} = \{x | d(fiber^f(x), fiber^c(x)) < \frac{K}{3}\}$$

We start by noting that the complement of $\mathcal{G}$, the set of 'bad' points $\mathcal{B} = \{x | d(fiber^f_x, fiber^c_x) \geq \frac{K}{3}\}$, is small as the lemma requires. The argument is the following

$$\varepsilon = e(\bar{c}, f) = \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N} 1_{\bar{c}(x_i,y_j)=f(x_i,y_y)} = \frac{1}{N^2}[\sum_{x_i\in\mathcal{B}}\sum_{j=1}^{N} 1_{\bar{c}(x_i,y_j)=f(x_i,y_y)} + \sum_{x_i\in\mathcal{G}}\sum_{j=1}^{N} 1_{\bar{c}(x_i,y_j)=f(x_i,y_y)}] \geq \frac{1}{N^2}\sum_{x_i\in\mathcal{B}}\frac{K}{3}N = \frac{K}{3N}|\mathcal{B}|$$

Next, assume that $c(x) = c(y)$ holds for two points $x, y \in \mathcal{G}$. Since $fiber^c(x) = fiber^c(y)$ we get

$$d(fiber^f(x), fiber^f(y)) \leq d(fiber^f(x), fiber^c(x)) + d(fiber^c(x), fiber^c(y)) + d(fiber^c(y), fiber^f(y)) < \frac{K}{3} + 0 + \frac{K}{3} = \frac{2K}{3}$$

Finally, if $c(x) \neq c(y)$ then $fiber^c(x)$ and $fiber^c(y)$ are indicators of disjoint sets, each bigger or equal to K. Hence $d(fiber^c(x), fiber^c(y)) \geq 2K$ and we get

$$\begin{aligned} 2K &\leq d(fiber^c(x), fiber^c(y)) \\ &\leq d(fiber^c(x), fiber^f(x)) + d(fiber^f(x), fiber^f(y)) + d(fiber^f(y), fiber^c(y)) \\ &\leq \frac{K}{3} + d(fiber^f(x), fiber^f(y)) + \frac{K}{3} \\ &\implies d(fiber^f(x), fiber^f(y)) \geq \frac{4K}{3} \end{aligned}$$

It follows from the lemma that over the 'good' set $\mathcal{G}$, which contains more than $(1 - \frac{3\varepsilon}{K})N$ points, the classes are very well separated. Each class is concentrated in a $\frac{K}{3}$-ball and the different balls are $\frac{4K}{3}$ distant from each other. Intuitively, under such conditions almost any reasonable clustering algorithm can find the correct partitioning over this set; since the size of the remaining set of 'bad' points $\mathcal{B}$ is linear in ε, the total error is expected to be linear in ε too.

However, in order to prove a worst case bound we still face a certain problem. Since we do not know how to tell $\mathcal{G}$ from $\mathcal{B}$, the 'bad' points might obscure the partition. We therefore suggest the following greedy procedure to define a partition g over Y:

- Compute the fibers $fiber^f(x)$ for all $x \in Y$.
- Let $i = 0$, $S_0 = Y$; while $|S_i| > \frac{KN}{2}$ do:

- for each point $x \in S_i$, compute the set of all points lying inside a sphere of radius $\frac{2K}{3}$ around x:

$$B_{\frac{2k}{3}}(x) = \{y \in S_i : d(fiber^f(x), fiber^f(y)) < \frac{2K}{3}\}$$

- find $z = \arg\max_{x \in S_i} |B_{\frac{2K}{3}}(x)|$ and define $g(y) = i$ for every $y \in B_{\frac{2k}{3}}(z)$;
- remove the points of $B_{\frac{2k}{3}}(z)$ from S_i: let $S_{i+1} = S_i \backslash B_{\frac{2k}{3}}(z)$, and $i = i+1$.

– Let $\mathbf{M}_g$ denote the number of rounds completed. Denote the domain on which g has been defined so far as G_0. Define g for the remaining points in $Y \backslash G_0$ as follows:

$$g(x) = \arg\min_{i \in \{0,\ldots,\mathbf{M}_g - 1\}} d(fiber^f(x), I_{\{g^{-1}(i)\}})$$

where $I_{\{g^{-1}(i)\}}$ is the indicator function of cluster i of g. Note, however, that the way g is defined over this set is not really important since, as we shall see, the set is small.

The proof for the error bound of g starts with two lemmas:

1. The first lemma uses lemma 2 to show that each cluster defined by g intersects only a single set of the form $c^{-1}(i) \cap \mathcal{G}$.
2. The second lemma shows that due to the greedy nature of the algorithm, the sets $g^{-1}(i)$ chosen at each step are big enough so that each intersects at least one of the sets $\{c^{-1}(j) \cap \mathcal{G}\}_{j=1}^{\mathbf{M}-1}$.
   It immediately follows that each set $g^{-1}(i)$ intersects a single set $\{c^{-1}(j) \cap \mathcal{G}\}$, and a match between the clusters of g and the classes of c can be established, while $Y \backslash G_0$ can be shown to be $O(\varepsilon)$ small.
3. Finally, the error of g is bounded by showing that if $x \in G_0 \cap \mathcal{G}$ then x is classified correctly by g.

Details of the lemmas and proofs are given in the technical report [3], appendix C, which completes the proof of Thm. 6.

3.2 Classifing using a product space hypothesis

Given an ε good product space hypothesis f, we can build a multiclass classifier as follows: Sample N unlabeled data points $Y = \{x_i\}_{i=1}^N$ from X and partition them using the algorithm presented in the previous subsection. A new point Z is classified as a member of the class l where

$$l = \arg\min_{i \in \{0,..,M-1\}} d(fiber^f(z), I_{g^{-1}(i)})$$

The following theorem bounds the error of such a classifier

Theorem 7. *Assume the error probability of f over $X \times X$ is $e(f, \bar{c}) = \varepsilon < \frac{K^2}{8}$. For each $\delta > 0$, $l > 4$: if $N > \frac{3l}{K(\frac{l}{4}-1)^2} log(\frac{1}{\delta})$, then the error of the classifier proposed is lower than $\frac{l\varepsilon}{K} + \delta$*

The proof is omitted.

4 Concluding remarks

We showed in this paper that learning in the product space produces good multiclass classifiers of the original space, and that the sample complexity of learning in the product space is comparable to the complexity of learning in the original space. We see the significance of these results in two aspects: First, since learning in the product space always involves only binary functions, we can use the full power of binary classification theory and its many efficient algorithms to solve multiclass classification problems. In contrast, the learning toolbox for multiclass problems in the original space is relatively limited. Second, the automatic acquisition of product space labels is plausible in many domains in which the data is produced by some Markovian process. In such domains the learning of interesting concepts without any human supervision may be possible.

References

1. E. Allwein, R. Schapire, and Y. Singer. Reducing Multiclass to Binary: A Unifying Approach for Margin Classifiers. *Journal of Machine Learning Research*, 1:113-141, 2000.
2. N. Bansal, A. Blum, and S. Chawla. Correlation Clustering. In Proc. FOCS 2002, pages 238-247.
3. A. Bar-Hillel, and D. Weinshall. Learning with Equivalence Constraints. HU Technical Report 2003-38, in http://www.cs.huji.ac.il/~daphna.
4. S. Ben-David, N. Cesa-Bianchi, D. Haussler, and P. H. Long. Characterizations of learnability for classes of 0, . . . , n-valued functions.
5. T. G. Dietterich, and G. Bakiri. Solving multiclass learning problems via error-correcting output codes. *Journal of Artifical Intelligence Research*, 2:263-286, 1995.
6. V. Guruswami and Amit Sahai. Multiclass learning, Boosting, and Error-Correcting codes. In Proc. COLT, 1999.
7. S. Har-Peled, D. Roth, D. Zimak. Constraints classification:A new approach to multiclass classification and ranking. In Proc. NIPS, 2002.
8. D. Haussler and P.M. Long. A generalization of Sauer's lemma. Technical Report UCSU-CRL-90-15. UC Santa Cruz, 1990.
9. M. J. Kearns and U. V. Vazirani. An Introduction to Computational Learning Theory. MIT Press, 1994.
10. B. K. Natarajan. On learning sets and functions. *Machine Learning*, 4:67-97, 1989.
11. P. J. Phillips. Support vector machines applied to face recognition. In M. C. Mozer, M. I. Jordan, and T. Petsche, editors, *Advances in Neural Information Processing Systems 11*, page 803. MIT Press, 1998.
12. T. Hertz, N. Shental, A. Bar-Hillel, and D. Weinshall. Enhancing Image and Video Retrieval: Learning via Equivalence Constraints. In Proc. of IEEE Conference on Computer Vision and Pattern Recognition, 2003.
13. R. E. Schapire. A brief introduction to boosting. In Proc. of the Sixteenth International Joint Conference on Artificial Intelligence, 1999.
14. V. N. Vapnik. The Nature of Statistical Learning. Springer, 1995.

Journal of Machine Learning Research x (2005) 1-29 Submitted 7/04; Published 4/05

Learning a Mahalanobis Metric from Equivalence Constraints

Aharon Bar-Hillel AHARONBH@CS.HUJI.AC.IL
Tomer Hertz TOMBOY@CS.HUJI.AC.IL
Noam Shental FENOAM@CS.HUJI.AC.IL
Daphna Weinshall DAPHNA@CS.HUJI.AC.IL
School of Computer Science & Engineering and Center for Neural Computation
The Hebrew University of Jerusalem
Jerusalem, Israel 91904

Editor: Greg Ridgeway

Abstract

Many learning algorithms use a metric defined over the input space as a principal tool, and their performance critically depends on the quality of this metric. We address the problem of learning metrics using side-information in the form of equivalence constraints. Unlike labels, we demonstrate that this type of side-information can sometimes be automatically obtained without the need of human intervention. We show how such side-information can be used to modify the representation of the data, leading to improved clustering and classification.

Specifically, we present the Relevant Component Analysis (RCA) algorithm, which is a simple and efficient algorithm for learning a Mahalanobis metric. We show that RCA is the solution of an interesting optimization problem, founded on an information theoretic basis. If dimensionality reduction is allowed within RCA, we show that it is optimally accomplished by a version of Fisher's linear discriminant that uses constraints. Moreover, under certain Gaussian assumptions, RCA can be viewed as a Maximum Likelihood estimation of the within class covariance matrix. We conclude with extensive empirical evaluations of RCA, showing its advantage over alternative methods.

Keywords: clustering, metric learning, dimensionality reduction, equivalence constraints, side information.

1. Introduction

A number of learning problems, such as clustering and nearest neighbor classification, rely on some a priori defined distance function over the input space. It is often the case that selecting a "good" metric critically affects the algorithms' performance. In this paper, motivated by the wish to boost the performance of these algorithms, we study ways to learn a "good" metric using side information.

One difficulty in finding a "good" metric is that its quality may be context dependent. For example, consider an image-retrieval application which includes many facial images. Given a query image, the application retrieves the most similar faces in the database according to some pre-determined metric. However, when presenting the query image we may be interested in retrieving other images of the same person, or we may want to retrieve other faces with the same facial expression. It seems difficult for a pre-determined metric to be suitable for two such different tasks.

In order to learn a context dependent metric, the data set must be augmented by some additional information, or side-information, relevant to the task at hand. For example we may have access to the labels of *part* of the data set. In this paper we focus on another type of side-information,

in which *equivalence constraints* between a few of the data points are provided. More specifically we assume knowledge about small groups of data points that are known to originate from the same class, although their label is unknown. We term these small groups of points *"chunklets"*.

A key observation is that in contrast to explicit labels that are usually provided by a human instructor, in many unsupervised learning tasks equivalence constraints may be extracted with minimal effort or even automatically. One example is when the data is inherently sequential and can be modelled by a Markovian process. Consider for example movie segmentation, where the objective is to find all the frames in which the same actor appears. Due to the continuous nature of most movies, faces extracted from successive frames in roughly the same location can be assumed to come from the same person. This is true as long as there is no scene change, which can be robustly detected (Boreczky and Rowe, 1996). Another analogous example is speaker segmentation and recognition, in which the conversation between several speakers needs to be segmented and clustered according to speaker identity. Here, it may be possible to automatically identify small segments of speech which are likely to contain data points from a single yet *unknown* speaker.

A different scenario, in which equivalence constraints are the natural source of training data, occurs when we wish to learn from several teachers who do not know each other and who are not able to coordinate among themselves the use of common labels. We call this scenario 'distributed learning'.[1] For example, assume that you are given a large database of facial images of many people, which cannot be labelled by a small number of teachers due to its vast size. The database is therefore divided (arbitrarily) into P parts (where P is very large), which are then given to P teachers to annotate. The labels provided by the different teachers may be inconsistent: as images of the same person appear in more than one part of the database, they are likely to be given different names. Coordinating the labels of the different teachers is almost as daunting as labelling the original data set. However, equivalence constraints can be easily extracted, since points which were given the same tag by a certain teacher are known to originate from the same class.

In this paper we study how to use equivalence constraints in order to learn an optimal Mahalanobis metric between data points. Equivalently, the problem can also be posed as learning a good representation function, transforming the data representation by the square root of the Mahalanobis weight matrix. Therefore we shall discuss the two problems interchangeably.

In Section 2 we describe the proposed method–the Relevant Component Analysis (RCA) algorithm. Although some of the interesting results can only be proven using explicit Gaussian assumptions, the optimality of RCA can be shown with some relatively weak assumptions, restricting the discussion to linear transformations and the Euclidean norm. Specifically, in Section 3 we describe a novel information theoretic criterion and show that RCA is its optimal solution. If Gaussian assumptions are added the result can be extended to the case where dimensionality reduction is permitted, and the optimal solution now includes Fisher's linear discriminant (Fukunaga, 1990) as an intermediate step. In Section 4 we show that RCA is also the optimal solution to another optimization problem, seeking to minimize within class distances. Viewed this way, RCA is directly compared to another recent algorithm for learning Mahalanobis distance from equivalence constraints, proposed by Xing et al. (2003). In Section 5 we show that under Gaussian assumptions RCA can be interpreted as the maximum-likelihood (ML) estimator of the within class covariance matrix. We also provide a bound over the variance of this estimator, showing that it is at most twice the variance of the ML estimator obtained using the fully labelled data.

1. A related scenario (which we call 'generalized relevance feedback'), where users of a retrieval engine are asked to annotate the retrieved set of data points, has similar properties.

The successful application of RCA in high dimensional spaces requires dimensionality reduction, whose details are discussed in Section 6. An online version of the RCA algorithm is presented in Section 7. In Section 8 we describe extensive empirical evaluations of the RCA algorithm. We focus on two tasks–data retrieval and clustering, and use three types of data: (a) A data set of frontal faces (Belhumeur et al., 1997); this example shows that RCA with partial equivalence constraints typically yields comparable results to supervised algorithms which use fully labelled training data. (b) A large data set of images collected by a real-time surveillance application, where the equivalence constraints are gathered automatically. (c) Several data sets from the UCI repository, which are used to compare between RCA and other competing methods that use equivalence constraints.

Related work

There has been much work on learning representations and distance functions in the supervised learning settings, and we can only briefly mention a few examples. Hastie and Tibshirani (1996) and Jaakkola and Haussler (1998) use labelled data to learn good metrics for classification. Thrun (1996) learns a distance function (or a representation function) for classification using a "leaning-to-learn" paradigm. In this setting several related classification tasks are learned using several labelled data sets, and algorithms are proposed which learn representations and distance functions in a way that allows for the transfer of knowledge between the tasks. In the work of Tishby et al. (1999) the joint distribution of two random variables X and Z is assumed to be known, and one seeks a compact representation of X which bears high relevance to Z. This work, which is further developed in Chechik and Tishby (2003), can be viewed as supervised representation learning.

As mentioned, RCA can be justified using information theoretic criteria on the one hand, and as an ML estimator under Gaussian assumptions on the other. Information theoretic criteria for unsupervised learning in neural networks were studied by Linsker (1989), and have been used since in several tasks in the neural network literature. Important examples are self organizing neural networks (Becker and Hinton, 1992) and Independent Component Analysis (Bell and Sejnowski, 1995)). Viewed as a Gaussian technique, RCA is related to a large family of feature extraction techniques that rely on second order statistics. This family includes, among others, the techniques of Partial Least-Squares (PLS) (Geladi and Kowalski, 1986), Canonical Correlation Analysis (CCA) (Thompson, 1984) and Fisher's Linear Discriminant (FLD) (Fukunaga, 1990). All these techniques extract linear projections of a random variable X, which are relevant to the prediction of another variable Z in various settings. However, PLS and CCA are designed for regression tasks, in which Z is a continuous variable, while FLD is used for classification tasks in which Z is discrete. Thus, RCA is more closely related to FLD, as theoretically established in Section 3.3. An empirical investigation is offered in Section 8.1.3, in which we show that RCA can be used to enhance the performance of FLD in the fully supervised scenario.

In recent years some work has been done on using equivalence constraints as side information. Both positive ('a is similar to b') and negative ('a is dissimilar from b') equivalence constraints were considered. Several authors considered the problem of semi-supervised clustering using equivalence constraints. More specifically, positive and negative constraints were introduced into the complete linkage algorithm (Klein et al., 2002), the K-means algorithm (Wagstaff et al., 2001) and the EM of a Gaussian mixture model (Shental et al., 2003). A second line of research, to which this work belongs, focuses on learning a 'good' metric using equivalence constraints. Learning a Mahalanobis metric from both positive and negative constraints was addressed in the work of Xing et al. (2003),

presenting an algorithm which uses gradient ascent and iterative projections to solve a convex non linear optimization problem. We compare this optimization problem to the one solved by RCA in Section 4, and empirically compare the performance of the two algorithms in Section 8. The initial description of RCA was given in the context of image retrieval (Shental et al., 2002), followed by the work of Bar-Hillel et al. (2003). Recently Bilenko et al. (2004) suggested a K-means based clustering algorithm that also combines metric learning. The algorithm uses both positive and negative constraints and learns a single or multiple Mahalanobis metrics.

2. Relevant Component Analysis: the algorithm

Relevant Component Analysis (RCA) is a method that seeks to identify and down-scale global unwanted variability within the data. The method changes the feature space used for data representation, by a global linear transformation which assigns large weights to "relevant dimensions" and low weights to "irrelevant dimensions" (see Tenenbaum and Freeman, 2000). These "relevant dimensions" are estimated using *chunklets*, that is, small subsets of points that are known to belong to the same although *unknown* class. The algorithm is presented below as Algorithm 1 (Matlab code can be downloaded from the authors' sites).

Algorithm 1 The RCA algorithm

Given a data set $X = \{x_i\}_{i=1}^{N}$ and n chunklets $C_j = \{x_{ji}\}_{i=1}^{n_j}$ $j = 1 \dots n$, do

1. Compute the within chunklet covariance matrix (Figure 1d).

$$\hat{C} = \frac{1}{N} \sum_{j=1}^{n} \sum_{i=1}^{n_j} (x_{ji} - m_j)(x_{ji} - m_j)^t \tag{1}$$

   where m_j denotes the mean of the j'th chunklet.

2. If needed, apply dimensionality reduction to the data using $\hat{C}$ as described in Algorithm 2 (see Section 6).

3. Compute the whitening transformation associated with $\hat{C}$: $W = \hat{C}^{-\frac{1}{2}}$ (Figure 1e), and apply it to the data points: $X_{new} = WX$ (Figure 1f), where X refers to the data points after dimensionality reduction when applicable. Alternatively, use the inverse of $\hat{C}$ in the Mahalanobis distance: $d(x_1, x_2) = (x_1 - x_2)^t \hat{C}^{-1} (x_1 - x_2)$.

More specifically, points x_1 and x_2 are said to be related by a positive constraint if it is known that both points share the same (unknown) label. If points x_1 and x_2 are related by a positive constraint, and x_2 and x_3 are also related by a positive constraint, then a chunklet $\{x_1, x_2, x_3\}$ is formed. Generally, chunklets are formed by applying transitive closure over the whole set of positive equivalence constraints.

The RCA transformation is intended to reduce clutter, so that in the new feature space, the inherent structure of the data can be more easily unravelled (see illustrations in Figure 1a-f). To this end, the algorithm estimates the within class covariance of the data $cov(X|Z)$ where X and Z describe the data points and their labels respectively. The estimation is based on positive equiva-

lence constraints only, and does not use any explicit label information. In high dimensional data, the estimated matrix can be used for semi-supervised dimensionality reduction. Afterwards, the data set is whitened with respect to the estimated within class covariance matrix. The whitening transformation W (in Step 3 of Algorithm 1) assigns lower weights to directions of large variability, since this variability is mainly due to within class changes and is therefore "irrelevant" for the task of classification.

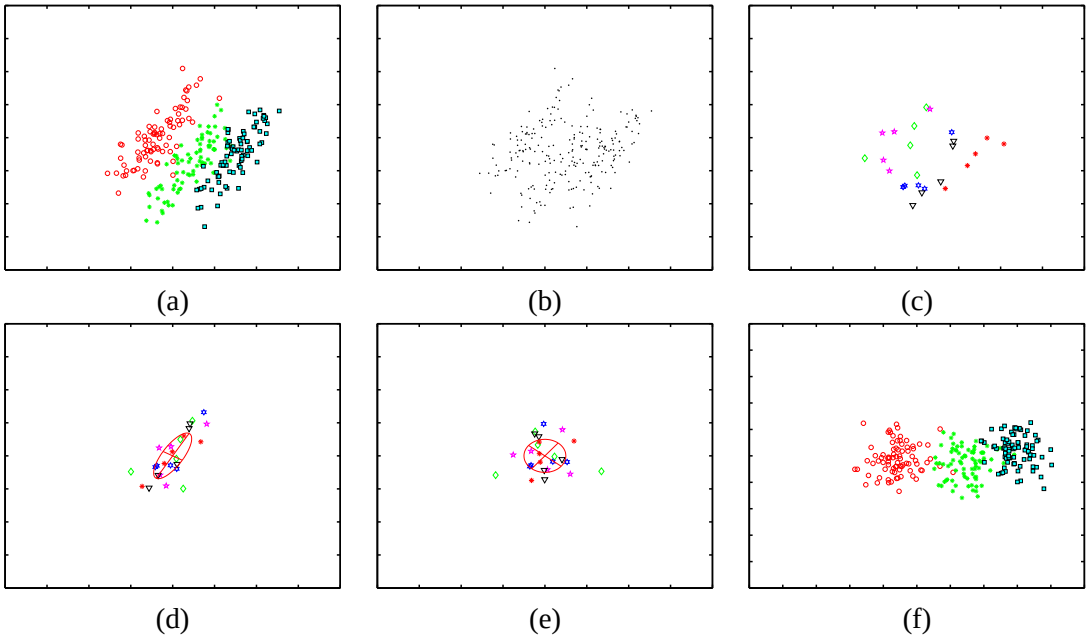


Figure 1: An illustrative example of the RCA algorithm applied to synthetic Gaussian data. (a) The fully labelled data set with 3 classes. (b) Same data unlabelled; clearly the classes' structure is less evident. (c) The set of chunklets that are provided to the RCA algorithm (points that share the same color and marker type form a chunklet). (d) The centered chunklets, and their empirical covariance. (e) The whitening transformation applied to the chunklets. (f) The original data after applying the RCA transformation.

The theoretical justifications for the RCA algorithm are given in Sections 3-5. In the following discussion, the term 'RCA' refers to the algorithm either with or without dimensionality reduction (optional Step 2). Usually the exact meaning can be readily understood in context. When we specifically discuss issues regarding the use of dimensionality reduction, we may use the explicit terms 'RCA with (or without) dimensionality reduction'.

RCA does not use negative equivalence constraints. While negative constraints clearly contain useful information, they are less informative than positive constraints (see counting argument below). They are also much harder to use computationally, due partly to the fact that unlike positive constraints, negative constraints are not transitive. In our case, the naïve incorporation of negative constraints leads to a matrix solution which is the difference of two positive definite matrices, and as a results does not necessarily produce a legitimate Mahalanobis metric. An alternative approach, which modifies the optimization function to incorporate negative constraints, as used for example by Xing et al. (2003), leads to a non-linear optimization problem with the usual associated drawbacks

of increased computational load and some uncertainty about the optimality of the final solution.[2] In contrast, RCA is the closed form solution of several interesting optimization problem, whose computation is no more complex than a single matrix inversion. Thus, in the tradeoff between runtime efficiency and asymptotic performance, RCA chooses the former and ignores the information given by negative equivalence constraints.

There is some evidence supporting the view that positive constraints are more informative than negative constraints. Firstly, a simple counting argument shows that positive constraints exclude more labelling possibilities than negative constraints. If for example there are M classes in the data, two data points have M^2 possible label combinations. A positive constraint between the points reduces this number to M combinations, while a negative constraint gives a much more moderate reduction to $M(M-1)$ combinations. (This argument can be made formal in information theoretic terms.) Secondly, empirical evidence from clustering algorithms which use both types of constraints shows that in most cases positive constraints give a much higher performance gain (Shental et al., 2003; Wagstaff et al., 2001). Finally, in most cases in which equivalence constraints are gathered automatically, only positive constraints can be gathered.

Step 2 of the RCA algorithm applies dimensionality reduction to the data if needed. In high dimensional spaces dimensionality reduction is almost always essential for the success of the algorithm, because the whitening transformation essentially re-scales the variability in all directions so as to equalize them. Consequently, dimensions with small total variability cause instability and, in the zero limit, singularity.

As discussed in Section 6, the optimal dimensionality reduction often starts with Principal Component Analysis (PCA). PCA may appear contradictory to RCA, since it eliminates principal dimensions with small variability, while RCA emphasizes principal dimensions with small variability. One should note, however, that the principal dimensions are computed in different spaces. The dimensions eliminated by PCA have small variability in the original data space (corresponding to $Cov(X)$), while the dimensions emphasized by RCA have low variability in a space where each point is translated according to the centroid of its own chunklet (corresponding to $Cov(X|Z)$). As a result, the method ideally emphasizes those dimensions with large total variance, but small within class variance.

3. Information maximization with chunklet constraints

How can we use chunklets to find a transformation of the data which improves its representation? In Section 3.1 we state the problem for general families of transformations and distances, presenting an information theoretic formulation. In Section 3.2 we restrict the family of transformation to non-singular linear maps, and use the Euclidean metric to measure distances. The optimal solution is then given by RCA. In Section 3.3 we widen the family of permitted transformations to include non-invertible linear transformations. We show that for normally distributed data RCA is the optimal transformation when its dimensionality reduction is obtained with a constraints based Fisher's Linear Discriminant (FLD).

2. Despite the problem's convexity, the proposed gradient based algorithm needs tuning of several parameters, and is not guaranteed to find the optimum without such tuning. See Section 8.1.5 for relevant empirical results.

3.1 An information theoretic perspective

Following Linsker (1989), an information theoretic criterion states that an optimal transformation of the input X into its new representation Y, should seek to maximize the mutual information $I(X,Y)$ between X and Y under suitable constraints. In the general case a set $X = \{x_i\}$ of data points in $\mathcal{R}^D$ is transformed into the set $Y = \{f(x_i)\}$ of points in $\mathcal{R}^K$. We seek a deterministic function $f \in F$ that maximizes $I(X,Y)$, where F is the family of permitted transformation functions (a "hypotheses family").

First, note that since f is deterministic, maximizing $I(X,Y)$ is achieved by maximizing the entropy $H(Y)$ alone. To see this, recall that by definition

$$I(X,Y) = H(Y) - H(Y|X)$$

where $H(X)$ and $H(Y|X)$ are differential entropies, as X and Y are continuous random variables. Since f is deterministic, the uncertainty concerning Y when X is known is minimal, thus $H(Y|X)$ achieves its lowest possible value at $-\infty$.[3] However, as noted by Bell and Sejnowski (1995), $H(Y|X)$ does not depend on f and is constant for every finite quantization scale. Hence maximizing $I(X,Y)$ with respect to f can be done by considering only the first term $H(Y)$.

Second, note also that $H(Y)$ can be increased by simply 'stretching' the data space. For example, if $Y = f(X)$ for an invertible continuous function, we can increase $H(Y)$ simply by choosing $Y = \lambda f(X)$ for any $\lambda > 1$. In order to avoid the trivial solution $\lambda \to \infty$, we can limit the distances between points contained in a single chunklet . This can be done by constraining the average distance between a point in a chunklet and the chunklet's mean. Hence the optimization problem is:

$$\max_{f \in F} H(Y_f) \quad s.t. \ \frac{1}{N}\sum_{j=1}^{n}\sum_{i=1}^{n_j} ||y_{ji} - m_j^y|| \leq \kappa \tag{2}$$

where $\{y_{ji}\}_{j=1,i=1}^{n\ ,\ n_j}$ denote the set of points in n chunklets after the transformation, m_j^y denotes the mean of chunklet j after the transformation, and κ is a constant.

3.2 RCA: the optimal linear transformation for the Euclidean norm

Consider the general problem (2) for the family F of invertible linear transformations, and using the squared Euclidean norm to measure distances. Since f is invertible, the connection between the densities of $Y = f(X)$ and X is expressed by $p_y(y) = \frac{p_x(x)}{|J(x)|}$, where $|J(x)|$ is the Jacobian of the transformation. From $p_y(y)dy = p_x(x)dx$, it follows that $H(Y)$ and $H(X)$ are related as follows:

$$H(Y) = -\int_y p(y)\log p(y)dy = -\int_x p(x)\log\frac{p(x)}{|J(x)|}dx = H(X) + \langle \log|J(x)|\rangle_x$$

For the linear map $Y = AX$ the Jacobian is constant and equals $|A|$, and it is the only term in $H(Y)$ that depends on the transformation A. Hence Problem (2) is reduced to

$$\max_A log|A| \quad s.t. \ \frac{1}{N}\sum_{j=1}^{n}\sum_{i=1}^{n_j} ||y_{ji} - m_j^y||_2^2 \leq \kappa$$

3. This non-intuitive divergence is a result of the generalization of information theory to continuous variables, that is, the result of ignoring the discretization constant in the definition of differential entropy.

Multiplying a solution matrix A by $\lambda > 1$ increases both the $log|A|$ argument and the constrained sum of within chunklet distances. Hence the maximum is achieved at the boundary of the feasible region, and the constraint becomes an equality. The constant κ only determines the scale of the solution matrix, and is not important in most clustering and classification tasks, which essentially rely on relative distances. Hence we can set $\kappa = 1$ and solve

$$\max_A log|A| \qquad s.t. \;\; \frac{1}{N}\sum_{j=1}^{n}\sum_{i=1}^{n_j}||y_{ji} - m_j^y||_2^2 = 1 \tag{3}$$

Let $B = A^tA$; since B is positive definite and $\log|A| = \frac{1}{2}\log|B|$, Problem (3) can be rewritten as

$$\max_{B \succ 0} log|B| \qquad s.t. \;\; \frac{1}{N}\sum_{j=1}^{n}\sum_{i=1}^{n_j}||x_{ji} - m_j||_B^2 = 1 \tag{4}$$

where $||.||_B$ denotes the Mahalanobis distance with weight matrix B. The equivalence between the problems is valid since for any $B \succ 0$ there is an A such that $B = A^tA$, and so a solution to (4) gives us a solution to (3) (and vice versa).

The optimization problem (4) can be solved easily, since the constraint is linear in B. The solution is $B = \frac{1}{D}\hat{C}^{-1}$, where $\hat{C}$ is the average chunklet covariance matrix (1) and D is the dimensionality of the data space. This solution is identical to the Mahalanobis matrix compute by RCA up to a global scale factor, or in other words, RCA is a scaled solution of (4).

3.3 Dimensionality reduction

We now solve the optimization problem (4) for the family of general linear transformations, that is, $Y = AX$ where $A \in \mathcal{M}_{K\times D}$ and $K \leq D$. In order to obtain workable analytic expressions, we assume that the distribution of X is a multivariate Gaussian, from which it follows that Y is also Gaussian with the following entropy

$$H(Y) \;\; = \;\; \frac{D}{2}\log 2\pi e + \frac{1}{2}\log|\Sigma_y| = \frac{D}{2}\log 2\pi e + \frac{1}{2}\log|A\Sigma_x A^t|$$

Following the same reasoning as in Section 3.2 we replace the inequality with equality and let $\kappa = 1$. Hence the optimization problem becomes

$$\max_A \log|A\Sigma_x A^t| \qquad s.t. \;\; \frac{1}{N}\sum_{j=1}^{n}\sum_{i=1}^{n_j}||x_{ji} - m_j||_{A^tA}^2 = 1 \tag{5}$$

For a given target dimensionality K, the solution of the problem is Fisher linear discriminant (FLD),[4] followed by the whitening of the within chunklet covariance in the reduced space. A sketch of the proof is given in Appendix A. The optimal RCA procedure therefore includes dimensionality reduction. Since the FLD transformation is computed based on the estimated within chunklet covariance matrix, it is essentially a semi-supervised technique, as described in Section 6. Note that after the FLD step, the within class covariance matrix in the reduced space is always diagonal, and Step 3 of RCA amounts to the scaling of each dimension separately.

4. Fisher Linear Discriminant is a linear projection A from $\mathcal{R}^D$ to $\mathcal{R}^K$ with $K < D$, which maximizes the determinant ratio $\max_{A\in\mathcal{M}_{K\times D}} \frac{AS_tA^t}{AS_wA^t}$, where S_t and S_w denote the total covariance and the within class covariance respectively.

4. RCA and the minimization of within class distances

In order to gain some intuition about the solution provided by the information maximization criterion (2), let us look at the optimization problem obtained by reversing the roles of the maximization term and the constraint term in problem (4):

$$\min_B \frac{1}{N}\sum_{j=1}^{n}\sum_{i=1}^{n_j}||x_{ji}-m_j||_B^2 \quad s.t.\ |B|\geq 1 \tag{6}$$

We interpret problem (6) as follows: a Mahalanobis distance B is sought, which minimizes the sum of all within chunklet squared distances, while $|B|\geq 1$ prevents the solution from being achieved by "shrinking" the entire space. Using the Kuhn-Tucker theorem, we can reduce (6) to

$$\min_B \sum_{j=1}^{n}\sum_{i=1}^{n_j}||x_{ji}-m_j||_B^2-\lambda\log|B| \quad s.t.\ \lambda\geq 0,\ \ \lambda\log|B|=0 \tag{7}$$

Differentiating this Lagrangian shows that the minimum is given by $B=|\hat{C}|^{\frac{1}{D}}\hat{C}^{-1}$, where $\hat{C}$ is the average chunklet covariance matrix. Once again, the solution is identical to the Mahalanobis matrix in RCA up to a scale factor.

It is interesting, in this respect, to compare RCA with the method proposed recently by Xing et al. (2003). They consider the related problem of learning a Mahalanobis distance using side information in the form of pairwise constraints (Chunklets of size >2 are not considered). It is assumed that in addition to the set of positive constraints Q_P, one is also given access to a set of negative constraints Q_N–a set of pairs of points known to be dissimilar. Given these sets, they pose the following optimization problem.

$$\min_B \sum_{(x_1,x_2)\in Q_P}||x_1-x_2||_B^2 \qquad s.t.\ \sum_{(x_1,x_2)\in Q_N}||x_1-x_2||_B\geq 1,\quad B\succeq 0 \tag{8}$$

This problem is then solved using gradient ascent and iterative projection methods.

In order to allow a clear comparison of RCA with (8), we reformulate the argument of (6) using only within chunklet pairwise distances. For each point x_{ji} in chunklet j we have:

$$x_{ji}-m_j=x_{ji}-\frac{1}{n_j}\sum_{k=1}^{n_j}x_{jk}=\frac{1}{n_j}\sum_{k=1}^{n_j}(x_{ji}-x_{jk})$$

Problem (6) can now be rewritten as

$$\min_B \frac{1}{N}\sum_{j=1}^{n}\frac{1}{n_j^2}\sum_{i=1}^{n_j}||\sum(x_{ji}-x_{jk})||_B^2 \quad s.t.\ |B|\geq 1 \tag{9}$$

When only chunklets of size 2 are given, as in the case studied by Xing et al. (2003), (9) reduces to

$$\min_B \frac{1}{2N}\sum_{j=1}^{n}||x_{j1}-x_{j2}||_B^2 \quad s.t.\ |B|\geq 1 \tag{10}$$

Clearly the minimization terms in problems (10) and (8) are identical up to a constant ($\frac{1}{2N}$). The difference between the two problems lies in the constraint term: the constraint proposed by Xing et al. (2003) uses pairs of dissimilar points, whereas the constraint in the RCA formulation affects global scaling so that the 'volume' of the Mahalanobis neighborhood is not allowed to shrink indefinitely. As a result Xing et al. (2003) are faced with a much harder optimization problem, resulting in a slower and less stable algorithm.

5. RCA and Maximum Likelihood: the effect of chunklet size

We now consider the case where the data consists of several normally distributed classes sharing the same covariance matrix. Under the assumption that the chunklets are sampled i.i.d. and that points within each chunklet are also sampled i.i.d., the likelihood of the chunklets' distribution can be written as:

$$\prod_{j=1}^{n}\prod_{i=1}^{n_j}\frac{1}{(2\pi)^{\frac{D}{2}}|\Sigma|^{\frac{1}{2}}}\exp\left(-\tfrac{1}{2}(x_{ji}-m_j)^t\Sigma^{-1}(x_{ji}-m_j)\right)$$

Writing the log-likelihood while neglecting constant terms and denoting $B=\Sigma^{-1}$, we obtain:

$$\sum_{j=1}^{n}\sum_{i=1}^{n_j}||x_{ji}-m_j||_B^2 - N\log|B| \tag{11}$$

where N is the total number of points in chunklets. Maximizing the log-likelihood is equivalent to minimizing (11), whose minimum is obtained when B equals the RCA Mahalanobis matrix (1). Note, moreover, that (11) is rather similar to the Lagrangian in (7), where the Lagrange multiplier is replaced by the constant N. Hence, under Gaussian assumptions, the solution of Problem (7) is probabilistically justified by a maximum likelihood formulation.

Under Gaussian assumptions, we can further define an *unbiased* version of the RCA estimator. Assume for simplicity that there are N constrained data points divided into n chunklets of size k each. The *unbiased* RCA estimator can be written as:

$$\hat{C}(n,k)=\frac{1}{n}\sum_{j=1}^{n}\frac{1}{k-1}\sum_{i=1}^{k}(x_{ji}-m_i)(x_{ji}-m_i)^t$$

where $\hat{C}(n,k)$ denotes the empirical mean of the covariance estimators produced by each chunklet. It is shown in Appendix B that the variance of the elements $\hat{C}_{ij}$ of the estimating matrix is bounded by

$$Var(\hat{C}_{ij}(n,k)) \le (1+\frac{1}{k-1})Var(\hat{C}_{ij}(1,nk)) \tag{12}$$

where $\hat{C}_{ij}(1,nk)$ is the estimator when all the $N=nk$ points are known to belong to the same class, thus forming the best estimate possible from N points. This bound shows that the variance of the RCA estimator rapidly converges to the variance of the best estimator, even for chunklets of small size. For the smallest possible chunklets, of size 2, the variance is only twice as high as the best possible.

6. Dimensionality reduction

As noted in Section 2, RCA may include dimensionality reduction. We now turn to address this issue in detail. Step 3 of the RCA algorithm decreases the weight of principal directions along which the within class covariance matrix is relatively high, and increases the weight of directions along which it is low. This intuition can be made precise in the following sense:

Denote by $\{\lambda^i\}_{i=1}^D$ the eigenvalues of the within class covariance matrix, and consider the squared distance between two points from the same class $||x_1 - x_2||^2$. We can diagonalize the within class covariance matrix using an orthonormal transformation which does not change the distance. Therefore, let us assume without loss of generality that the covariance matrix is diagonal.

Before whitening, the average squared distance is $E[||x_1 - x_2||^2] = 2\sum_{j=1}^D \lambda^j$ and the average squared distance in direction i is $E[(x_1^i - x_2^i)^2] = 2\lambda^i$. After whitening these values become $2D$ and 2, respectively. Let us define the weight of dimension i, $W(i) \in [0,1]$, as

$$W(i) = \frac{E[(x_1^i - x_2^i)^2]}{E[||x_1 - x_2||^2]}$$

Now the ratio between the weight of each dimension before and after whitening is given by

$$\frac{W_{before}(i)}{W_{after}(i)} = \frac{\lambda^i}{\frac{1}{D}\sum_{j=1}^D \lambda^j} \tag{13}$$

In Equation (13) we observe that the weight of each principal dimension increases if its initial within class variance was lower than the average, and vice versa. When there is high irrelevant noise along several dimensions, the algorithm will indeed scale down noise dimensions. However, when the irrelevant noise is scattered among many dimensions with low amplitude in each of them, whitening will amplify these noisy dimensions, which is potentially harmful. Therefore, when the data is initially embedded in a high dimensional space, the optional dimensionality reduction in RCA (Step 2) becomes mandatory.

We have seen in Section 3.3 that FLD is the dimensionality reduction technique which maximizes the mutual information under Gaussian assumptions. Traditionally FLD is computed from fully labelled training data, and the method therefore falls within supervised learning. We now extend FLD, using the same information theoretic criterion, to the case of partial supervision in the form of equivalence constraints. Specifically, denote by S_t and S_w the estimators of the total covariance and the within class covariance respectively. FLD maximizes the following determinant ratio

$$\max_{A \in \mathcal{M}_{K \times D}} \frac{AS_tA^t}{AS_wA^t} \tag{14}$$

by solving a generalized eigenvector problem. The row vectors of the optimal matrix A are the first K eigenvectors of $S_w^{-1}S_t$. In our case the optimization problem is of the same form as in (14), with the within chunklet covariance matrix from (1) playing the role of S_w. We compute the projection matrix using SVD in the usual way, and term this FLD variant cFLD (constraints based FLD).

To understand the intuition behind cFLD, note that both PCA and cFLD remove dimensions with small total variance, and hence reduce the risk of RCA amplifying irrelevant dimensions with small variance. However, unsupervised PCA may remove dimensions that are important for the

discrimination between classes, if their total variability is low. Intuitively, better dimensionality reduction can be obtained by comparing the total covariance matrix (used by PCA) to the within class covariance matrix (used by RCA), and this is exactly what the partially supervised cFLD is trying to accomplish in (14).

The cFLD dimensionality reduction can only be used if the rank of the within chunklet covariance matrix is higher than the dimensionality of the initial data space. If this condition does not hold, we use PCA to reduce the original data dimensionality as needed. The procedure is summarized below in Algorithm 2.

Algorithm 2 Dimensionality reduction: Step 2 of RCA

Denote by D the original data dimensionality. Given a set of chunklets $\{C_j\}_{j=1}^{n}$ do

1. Compute the rank of the estimated within chunklet covariance matrix $R = \sum_{j=1}^{n}(|C_j| - 1)$, where $|C_j|$ denotes the size of the j'th chunklet.
2. If ($D > R$), apply PCA to reduce the data dimensionality to αR, where $0 < \alpha < 1$ (to ensure that cFLD provides stable results).
3. Compute the total covariance matrix estimate S_t, and estimate the within class covariance matrix using $S_w = \hat{C}$ from (1). Solve (14), and use the resulting A to achieve the target data dimensionality.

7. Online implementation of RCA

The standard RCA algorithm presented in Section 2 is a batch algorithm which assumes that all the equivalence constraints are available at once, and that all the data is sampled from a stationary source. Such conditions are usually not met in the case of biological learning systems, or artificial sensor systems that interact with a gradually changing environment. Consider for example a system that tries to cluster images of different people collected by a surveillance camera in gradually changing illumination conditions, such as those caused by night and day changes. In this case different distance functions should be used during night and day times, and we would like the distance used by the system to gradually adapt to the current illumination conditions. An online algorithm for distance function learning is required to achieve such a gradual adaptation.

Here we briefly present an online implementation of RCA, suitable for a neural-network-like architecture. In this implementation a weight matrix $W \in \mathcal{M}_{D\times D}$, initiated randomly, is gradually developed to become the RCA transformation matrix. In Algorithm 3 we present the procedure for the simple case of chunklets of size 2. The extension of this algorithm to general chunklets is briefly described in Appendix C.

Assuming local stationarity, the steady state of this stochastic process can be found by equating the mean update to 0, where the expectation is taken over the next example pair (x_1^{T+1}, x_2^{T+1}). Using the notations of Algorithm 3, the resulting equation is

$$E[\eta(W - yy^tW)] = 0 \quad \Rightarrow \quad E[I - yy^t] = I - WE[hh^t]W^t = 0 \quad \Rightarrow \quad W = PE[hh^t]^{-\frac{1}{2}}$$

where P is an orthonormal matrix $PP^t = I$. The steady state W is the whitening transformation of the correlation matrix of h. Since $h = 2(x_1 - \frac{(x_1+x_2)}{2})$, it is equivalent (up to the constant 2) to the

Algorithm 3 Online RCA for point pairs

Input: a stream of pairs of points (x_1^T, x_2^T), where x_1^T x_2^T are known to belong to the same class.
Initialize W to a symmetric random matrix with $||W|| << 1$.
At time step T do:

- receive pair x_1^T, x_2^T;
- let $h = x_1^T - x_2^T$;
- apply W to h, to get $y = Wh$;
- update $W = W + \eta(W - yy^tW)$.

where $\eta > 0$ determines the step size.

distance of a point from the center of its chunklet. The correlation matrix of h is therefore equivalent to the within chunklet covariance matrix. Thus W converges to the RCA transformation of the input population up to an orthonormal transformation. The resulting transformation is geometrically equivalent to RCA, since the orthonormal transformation P preserves vector norms and angles.

In order to evaluate the stability of the online algorithm we conducted simulations which confirmed that the algorithm converges to the RCA estimator (up to the transformation P), if the gradient steps decrease with time ($\eta = \eta_0/T$). However, the adaptation of the RCA estimator for such a step size policy can be very slow. Keeping η constant avoids this problem, at the cost of producing a noisy RCA estimator, where the noise is proportional to η. Hence η can be used to balance this tradeoff between adaptation, speed and accuracy.

8. Experimental Results

The success of the RCA algorithm can be measured directly by measuring neighborhood statistics, or indirectly by measuring whether it improves clustering results. In the following we tested RCA on three different applications using both direct and indirect evaluations.

The RCA algorithm uses only partial information about the data labels. In this respect it is interesting to compare its performance to unsupervised and supervised methods for data representation. Section 8.1 compares RCA to the unsupervised PCA and the fully supervised FLD on a facial recognition task, using the YaleB data set (Belhumeur et al., 1997). In this application of face recognition, RCA appears very efficient in eliminating irrelevant variability caused by varying illumination. We also used this data set to test the effect of dimensionality reduction using cFLD, and the sensitivity of RCA to average chunklet size and the total amount of points in chunklets.

Section 8.2 presents a more realistic surveillance application in which equivalence constraints are gathered automatically from a Markovian process. In Section 8.3 we conclude our experimental validation by comparing RCA with other methods which make use of equivalence constraints in a clustering task, using a few benchmark data sets from the UCI repository (Blake and Merz, 1998). The evaluation of different metrics below is presented using *cumulative neighbor purity* graphs, which display the average (over all data points) percentage of correct neighbors among the first k neighbors, as a function of k.

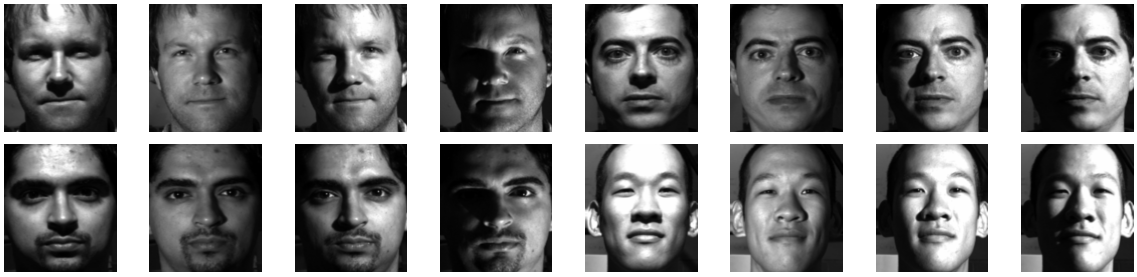


Figure 2: A subset of the YaleB database which contains 1920 frontal face images of 30 individuals taken under different lighting conditions.

8.1 Applying RCA to facial recognition

The task here is to classify facial images with respect to the person photographed. In these experiments we consider a retrieval paradigm reminiscent of nearest neighbor classification, in which a query image leads to the retrieval of its nearest neighbor or its K-nearest neighbors in the data set. Using a facial image database, we begin by evaluating nearest neighbor classification with the RCA distance, and compare its performance to supervised and unsupervised learning methods. We then move on to address more specific issues: In 8.1.4 we look more closely at the two steps of RCA, Step 2 (cFLD dimensionality reduction) and Step 3 (whitening w.r.t. $\hat{C}$), and study their contribution to performance in isolation. In 8.1.5 the retrieval performance of RCA is compared with the algorithm presented by Xing et al. (2003). Finally in 8.1.6 we evaluate the effect of chunklets sizes on retrieval performance, and compare it to the predicted effect of chunklet size on the variance of the RCA estimator.

8.1.1 THE DATA SET

We used a subset of the yaleB data set (Belhumeur et al., 1997), which contains facial images of 30 subjects under varying lighting conditions. The data set contains a total of 1920 images, including 64 frontal pose images of each subject. The variability between images of the same person is mainly due to different lighting conditions. These factors caused the variability among images belonging to the same subject to be greater than the variability among images of different subjects (Adini et al., 1997). As preprocessing, we first automatically centered all the images using optical flow. Images were then converted to vectors, and each image was represented using its first 60 PCA coefficients. Figure 2 shows a few images of four subjects.

8.1.2 OBTAINING EQUIVALENCE CONSTRAINTS

We simulated the *'distributed learning'* scenario presented in Section 1 in order to obtain equivalence constraints. In this scenario, we obtain equivalence constraints using the help of T teachers. Each teacher is given a random selection of L data points from the data set, and is asked to give his own labels to all the points, effectively partitioning the data set into equivalence classes. Each teacher therefore provides both positive and negative constraints. Note however that RCA only uses the positive constraints thus gathered. The total number of points in chunklets grows linearly with

TL, the number of data points seen by all teachers. We control this amount, which provides a loose bound on the number of points in chunklets, by varying the number of teachers T and keeping L constant. We tested a range of values of T for which TL is 10%, 30%, or 75% of the points in the data set.[5]

The parameter L controls the distribution of chunklet sizes. More specifically, we show in Appendix D that this distribution is controlled by the ratio $r = \frac{L}{M}$ where M is the number of classes in the data. In all our experiments we have used $r = 2$. For this value the expected chunklet size is roughly 2.9 and we typically obtain many small chunklets. Figure 3 shows a histogram of typical chunklet sizes, as obtained in our experiments.[6]

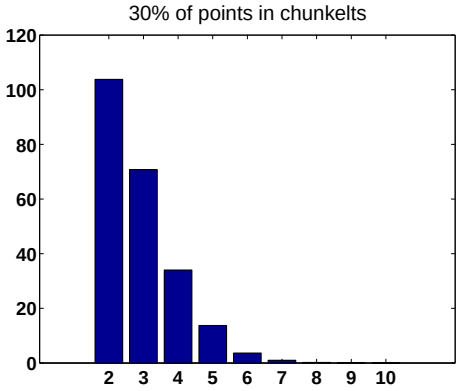


Figure 3: Sample chunklet size distribution obtained using the distributed learning scenario on a subset of the yaleB data set with 1920 images from $M = 30$ classes. L is chosen such that $r = \frac{L}{M} = 2$. The histogram is plotted for distributed learning with 30% of the data points in chunklets.

8.1.3 RCA ON THE CONTINUUM BETWEEN SUPERVISED AND UNSUPERVISED LEARNING

The goal of our main experiment in this section was to assess the relative performance of RCA as a semi-supervised method in a face recognition task. To this extent we compared the following methods:

- Eigenfaces (Turk and Pentland, 1991): this unsupervised method reduces the dimensionality of the data using PCA, and compares the images using the Euclidean metric in the reduced space. Images were normalized to have zero mean and unit variance.

- Fisherfaces (Belhumeur et al., 1997): this supervised method starts by applying PCA dimensionality reduction as in the Eigenfaces method. It then uses all the data labels to compute the FLD transformation (Fukunaga, 1990), and transforms the data accordingly.

5. In this scenario one usually obtains mostly 'negative' equivalence constraints, which are pairs of points that are known to originate from different classes. RCA does *not* use these 'negative' equivalence constraints.
6. We used a different sampling scheme in the experiments which address the effect of chunklet size, see Section 8.1.6.

- RCA: the RCA algorithm with dimensionality reduction as described in Section 6, that is, PCA followed by cFLD. We varied the amount of data in constraints provided to RCA, using the *distributed learning* paradigm described above.

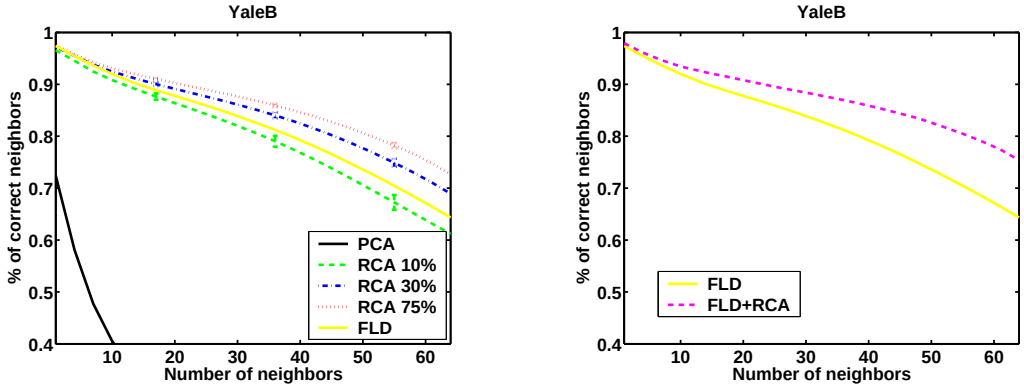


Figure 4: Left: Cumulative purity graphs for the following algorithms and experimental conditions: Eigenface (PCA), RCA 10%, RCA 30%, RCA 75%, and Fisherface (FLD). The percentages stated for RCA are the fractions of data points presented to the 'distributed learning' oracle, as discussed in Section 8.1.2. The data was reduced to dimension 60 using PCA for all the methods. It was then further reduced to dimension 30 using cFLD in the three RCA variants, and using FLD for the Fisherface method. Results were averaged over 50 constraints realizations. The error bars give the Standard Errors of the Mean (SEMs). Right: Cumulative purity graphs for the fully supervised FLD, with and without fully labelled RCA. Here RCA dramatically enhances the performance of FLD.

The left panel in Figure 4 shows the results of the different methods. The graph presents the performance of RCA for low, moderate and high amounts of constrained points. As can be seen, even with low amounts of equivalence constraints the performance of RCA is much closer to the performance of the supervised FLD than to the performance of the unsupervised PCA. With Moderate and high amounts of equivalence constraints RCA achieves neighbor purity rates which are higher than those achieved by the fully supervised Fisherfaces method, while relying only on fragmentary chunklets with unknown class labels. This somewhat surprising result stems from the fact that the fully supervised FLD in these experiments was not followed by whitening.

In order to clarify this last point, note that RCA can also be used when given a fully labelled training set. In this case, chunklets correspond uniquely and fully to classes, and the cFLD algorithm for dimensionality reduction is equivalent to the standard FLD. In this setting RCA can be viewed as an augmentation of the standard, fully supervised FLD, which whitens the output of FLD w.r.t the within class covariance. The right panel in Figure 4 shows comparative results of FLD with and without whitening in the fully labelled case.

In order to visualize the effect of RCA in this task we also created some "RCAfaces", following Belhumeur et al. (1997): We ran RCA on the images after applying PCA, and then reconstructed the images. Figure 5 shows a few images and their reconstruction. Clearly RCA dramatically reduces

the effect of varying lighting conditions, and the reconstructed images of the same individual look very similar to each other. The Eigenfaces (Turk and Pentland, 1991) method did not produce similar results.

Figure 5: Top: Several facial images of two subjects under different lighting conditions. Bottom: the same images from the top row after applying PCA and RCA and then reconstructing the images. Clearly RCA dramatically reduces the effect of different lighting conditions, and the reconstructed images of each person look very similar to each other.

8.1.4 SEPARATING THE CONTRIBUTION OF THE DIMENSIONALITY REDUCTION AND WHITENING STEPS IN RCA

Figure 4 presents the results of RCA including the semi-supervised dimensionality reduction of cFLD. While this procedure yields the best results, it mixes the separate contributions of the two main steps of the RCA algorithm, that is, dimensionality reduction via cFLD (Step 2) and whitening of the inner chunklet covariance matrix (Step 3). In the left panel of Figure 6 these contributions are isolated.

It can be seen that when cFLD and whitening are used separately, they both provide considerable improvement in performance. These improvements are only partially dependent, since the performance gain when combining both procedures is larger than either one alone. In the right panel of Figure 6 we present learning curves which show the performance of RCA with and without dimensionality reduction, as a function of the amount of supervision provided to the algorithm. For small amounts of constraints, both curves are almost identical. However, as the number of constraints increases, the performance of RCA dramatically improves when using cFLD.

8.1.5 COMPARISON WITH THE METHOD OF XING ET AL.

In another experiment we compared the algorithm of Xing et al. (2003) to RCA on the YaleB data set using code obtained from the author's web site. The experimental setup was the one described in Section 8.1.2, with 30% of the data points presented to the distributed learning oracle. While RCA uses only the positive constraints obtained, the algorithm of Xing et al. (2003) was given both the positive and negative constraints, as it can make use of both. Results are shown in Figure 7, showing that this algorithm failed to converge when given high dimensional data, and was outperformed by RCA in lower dimensions.

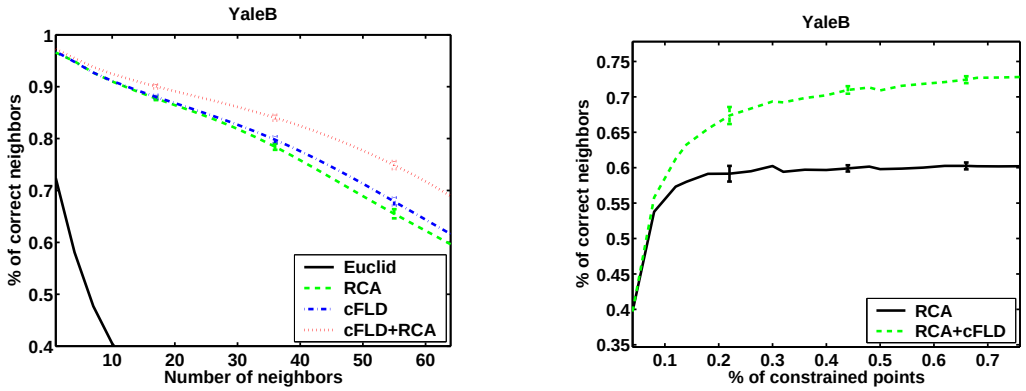


Figure 6: Left: Cumulative purity graphs for 4 experimental conditions: original space, RCA without cFLD, cFLD only, and RCA with cFLD (using the Euclidean norm in all cases). The data was reduced to 60 dimensions using unsupervised PCA. The semi supervised techniques used constraints obtained by distributed learning with 30% of the data points. RCA without cFLD was performed in the space of 60 PCA coefficients, while in the last 2 conditions dimensionality was further reduced to 30 using the constraints. Results were averaged over 50 constraints realizations. Right: Learning curves–neighbor purity performance for 64 neighbors as a function of the amount of constraints. The performance is measured by averaging (over all data points) the percentage of correct neighbors among the first 64 neighbors. The amount of constraints is measured using the percentage of points given to the distributed learning oracle. Results are averaged over 15 constraints realizations. Error bars in both graphs give the standard errors of the mean.

8.1.6 THE EFFECT OF DIFFERENT CHUNKLET SIZES

In Section 5 we showed that RCA typically provides an estimator for the within class covariance matrix, which is not very sensitive to the size of the chunklets. This was done by providing a bound on the variance of the elements in the RCA estimator matrix $\hat{C}(n,k)$. We can expect that lower variance of the estimator will go hand in hand with higher purity performance. In order to empirically test the effect of chunklets' size, we fixed the number of equivalence constraints, and varied the size of the chunklets S in the range $\{2-10\}$. The chunklets were obtained by randomly selecting 30% of the data (total of $P = 1920$ points) and dividing it into chunklets of size S.[7]

The results can be seen in Figure 8. As expected the performance of RCA improves as the size of the chunklets increases. Qualitatively, this improvement agrees with the predicted improvement in the RCA estimator's variance, as most of the gain in performance is already obtained with chunklets of size $S = 3$. Although the bound presented is not tight, other reasons may account for the difference between the graphs, including the weakness of the Gaussian assumption used to derive the bound (see Section 9), and the lack of linear connection between the estimator's variance and purity performance.

7. When necessary, the remaining $mod(0.3P, S)$ points were gathered into an additional smaller chunklet.

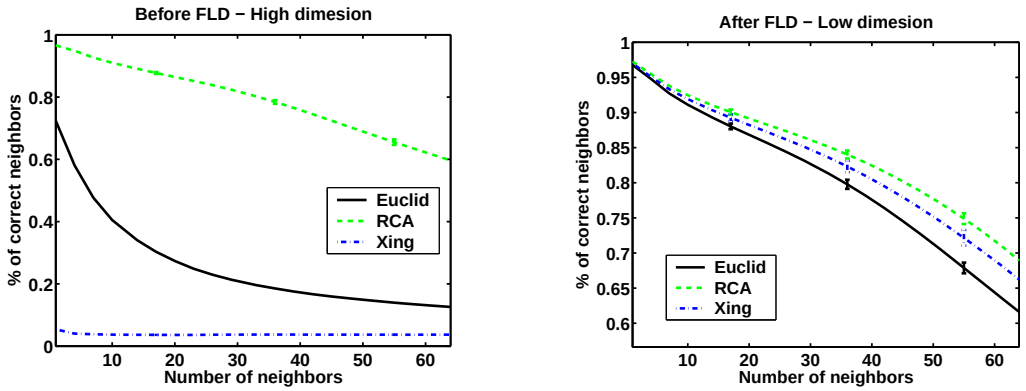


Figure 7: The method of Xing et al. (2003) and RCA on the YaleB facial image data set. Left: Neighbor purity results obtained using 60 PCA coefficients. The algorithm of Xing et al. (2003) failed to converge and returned a metric with chance level performance. Right: Results obtained using a 30 dimensional representation, obtained by applying cFLD to the 60 PCA coefficients. Results are averaged over 50 constraints realizations. The error bars give the standard errors of the mean.

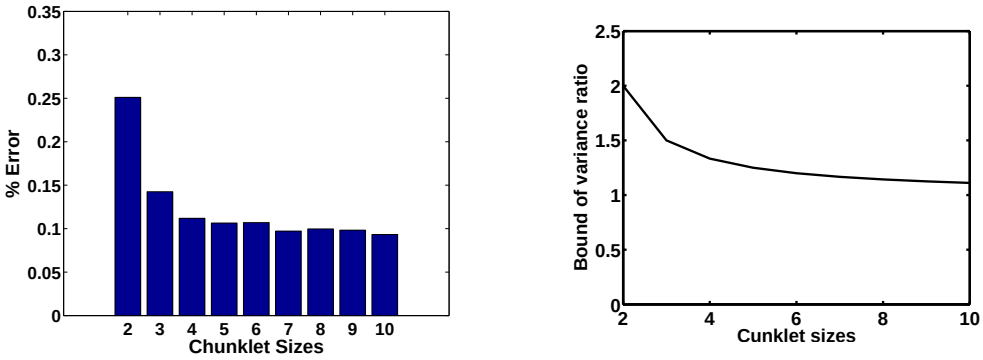


Figure 8: Left: Mean error rate on all 64 neighbors on the yaleB data set when using 30% of the data in chunklets. In this experiment we varied the chunklet sizes while fixing the total amount of points in chunklets. Right: the theoretical bound over the ratio between the variance of the RCA matrix elements and the variance of the best possible estimator using the same number of points (see inequality 12). The qualitative behavior of the graphs is similar, seemingly because a lower estimator variance tends to imply better purity performance.

8.2 Using RCA in a surveillance application

In this application, a stationary indoor surveillance camera provided short video clips whose beginning and end were automatically detected based on the appearance and disappearance of moving targets. The database therefore included many clips, each displaying only one person of unknown

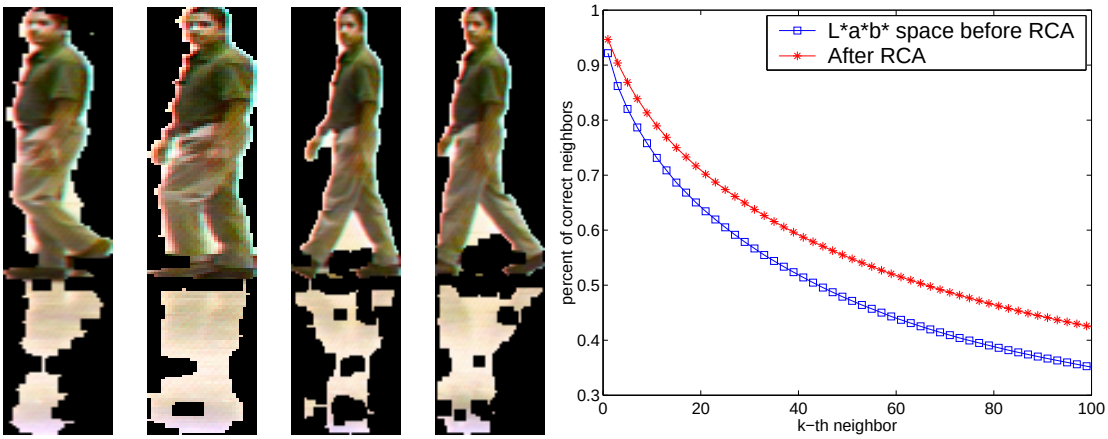


Figure 9: Left: several images from a video clip of one intruder. Right: cumulative neighbor purity results before and after RCA.

identity. Effectively each clip provided a chunklet. The task in this case was to cluster together all clips in which a certain person appeared.

The task and our approach: The video clips were highly complex and diversified, for several reasons. First, they were entirely unconstrained: a person could walk everywhere in the scene, coming closer to the camera or walking away from it. Therefore the size and resolution of each image varied dramatically. In addition, since the environment was not constrained, images included varying occlusions, reflections and (most importantly from our perspective) highly variable illumination. In fact, the illumination changed dramatically across the scene both in intensity (from brighter to darker regions), and in spectrum (from neon light to natural lighting). Figure 9 shows several images from one input clip.

We sought to devise a representation that would enable the effective clustering of clips, focusing on color as the only low-level attribute that could be reliably used in this application. Therefore our task was to accomplish some sort of color constancy, that is, to overcome the general problem of irrelevant variability due to the varying illumination. This is accomplished by the RCA algorithm.

Image representation and RCA Each image in a clip was represented by its color histogram in $L^*a^*b^*$ space (we used 5 bins for each dimension). We used the clips as chunklets in order to compute the RCA transformation. We then computed the distance between pairs of images using two methods: *L1* and RCA (Mahalanobis). We used over 6000 images from 130 clips (chunklets) of 20 different people. Figure 9 shows the cumulative neighbor purity over all 6000 images. One can see that RCA makes a significant contribution by bringing 'correct' neighbors closer to each other (relative to other images). However, the effect of RCA on retrieval performance here is lower than the effect gained with the YaleB data base. While there may be several reasons for this, an important factor is the difference between the way chunklets were obtained in the two data sets. The automatic gathering of chunklets from a Markovian process tends to provide chunklets with dependent data points, which supply less information regarding the within class covariance matrix.

8.3 RCA and clustering

In this section we evaluate RCA's contribution to clustering, and compare it to alternative algorithms that use equivalence constraints. We used six data sets from the UCI repository. For each data set we randomly selected a set Q_P of pairwise positive equivalence constraints (or chunklets of size 2). We compared the following clustering algorithms:

a. K-means using the default Euclidean metric and no side-information (Fukunaga, 1990).

b. Constrained K-means + Euclidean metric: the K-means version suggested by Wagstaff et al. (2001), in which a pair of points $(x_i, x_j) \in Q_P$ is always assigned to the same cluster.

c. Constrained K-means + the metric proposed by Xing et al. (2003): The metric is learnt from constraints in Q_P. For fairness we replicated the experimental design employed by Xing et al. (2003), and allowed the algorithm to treat all unconstrained pairs of points as negative constraints (the set Q_N).

d. Constrained K-means + RCA: Constrained K-means using the RCA Mahalanobis metric learned from Q_P.

e. EM: Expectation Maximization of a Gaussian Mixture model (using no side-information).

f. Constrained EM: EM using side-information in the form of equivalence constraints (Shental et al., 2003), when using the RCA distance metric as the initial metric.

Clustering algorithms a and e are unsupervised and provide respective lower bounds for comparison with our algorithms d and f. Clustering algorithms b and c compete fairly with our algorithm d, using the same kind of side information.

Experimental setup To ensure fair comparison with Xing et al. (2003), we used exactly the same experimental setup as it affects the gathering of equivalence constraints and the evaluation score used. We tested all methods using two conditions, with: (i) "little" side-information Q_P, and (ii) "much" side-information. The set of pairwise similarity constraints Q_P was generated by choosing a random subset of all pairs of points sharing the same class identity c_i. Initially, there are N 'connected components' of unconstrained points, where N is the number of data points. Randomly choosing a pairwise constraint decreases the number of connected components by 1 at most. In the case of "little" ("much") side-information, pairwise constraints are randomly added until the number of different connected components K_c is roughly $0.9N$ ($0.7N$). As in the work of Xing et al. (2003), no negative constraints were sampled.

Following Xing et al. (2003) we used a normalized accuracy score, the "Rand index" (Rand, 1971), to evaluate the partitions obtained by the different clustering algorithms. More formally, with binary labels (or two clusters), the accuracy measure can be written as:

$$\sum_{i>j} \frac{1\{1\{c_i = c_j\} = 1\{\hat{c}_i = \hat{c}_j\}\}}{0.5m(m-1)}$$

where $1\{\dot{}\}$ denotes the indicator function $(1\{True\} = 1), 1\{False\} = 0)$, $\{\hat{c}_i\}_{i=1}^m$ denotes the cluster to which point x_i is assigned by the clustering algorithm, and c_i denotes the "correct" (or

desirable) assignment. The score above is the probability that the algorithm's decision regarding the label equivalence of two points agrees with the decision of the "true" assignment c. [8]

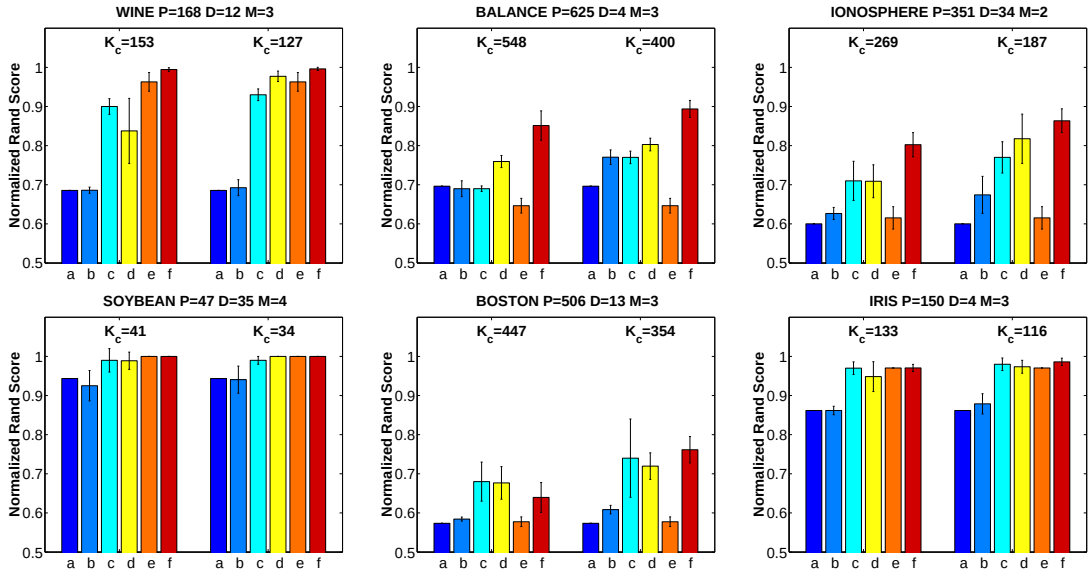


Figure 10: Clustering accuracy on 6 UCI data sets. In each panel, the six bars on the left correspond to an experiment with "little" side-information, and the six bars on the right correspond to "much" side-information. From left to right the six bars correspond respectively to the algorithms described in the text, as follows: (a) K-means over the original feature space (without using any side-information). (b) Constrained K-means over the original feature space. (c) Constrained K-means over the feature space suggested by Xing et al. (2003). (d) Constrained K-means over the feature space created by RCA. (e) EM over the original feature space (without using any side-information). (f) Constrained EM (Shental et al., 2003) over the feature space created by RCA. Also shown are P–the number of points, M–the number of classes, D–the dimensionality of the feature space, and K_c–the mean number of connected components. The results were averaged over 20 realizations of side-information. The error bars give the standard deviations. In all experiments we used K-means with multiple restarts as in done by Xing et al. (2003).

Figure 10 shows comparative results using six different UCI data sets. Clearly the RCA metric significantly improved the results over the original K-means algorithms (both the constrained and unconstrained versions). Generally in the context of K-means, we observe that using equivalence constraints to find a better metric improves results much more than using this information to constrain the algorithm. RCA achieves comparable results to those reported by Xing et al. (2003), despite the big difference in computational cost between the two algorithms (see Section 9.1).

8. As noted by Xing et al. (2003), this score should be normalized when the number of clusters is larger than 2. Normalization is achieved by sampling the pairs (x_i, x_j) such that x_i and x_j are from the same cluster with probability 0.5 and from different clusters with probability 0.5, so that "matches" and "mismatches" are given the same weight.

The last two algorithms in our comparisons use the EM algorithm to compute a generative Gaussian Mixture Model, and are therefore much more computationally intensive. We have added these comparisons because EM implicitly changes the distance function over the input space in a locally linear way (that is, like a Mahalanobis distance). It may therefore appear that EM can do everything that RCA does and more, without any modification. The histogram bins marked by (e) in Figure 10 clearly show that this is not the case. Only when we add constraints to the EM, and preprocess the data with RCA, do we get improved results as shown by the histogram bins marked by (f) in Figure 10.

9. Discussion

We briefly discuss running times in Section 9.1. The applicability of RCA in general conditions is then discussed in 9.2.

9.1 Runtime performance

Computationally RCA relies on a few relatively simple matrix operations (inversion and square root) applied to a positive-definite square matrix, whose size is the reduced dimensionality of the data. This can be done fast and efficiently and is a clear advantage of the algorithm over its competitors.

9.2 Using RCA when the assumptions underlying the method are violated

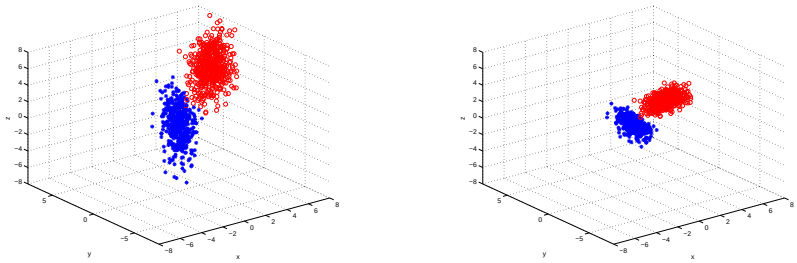


Figure 11: Extracting the shared component of the covariance matrix using RCA: In this example the data originates from 2 Gaussian sources with the following diagonal covariance matrices: $diag(C_1) = (\epsilon, 1, 2)$ and $diag(C_2) = (1, \epsilon, 2)$. (a) The original data points (b) The transformed data points when using RCA. In this example we used all of the points from each class as a single chunklet and therefore the chunklet covariance matrix is the average within-class covariance matrix. As can be seen RCA clearly down-scales the irrelevant variability in the Z axis, which is the shared component of the 2 classes covariance matrices. Specifically, the eigenvalues of the covariance matrices for the two classes are as follows (for $\epsilon = 0.1$): class 1–$(3.947, 1.045, 0.009)$ before RCA, and $(1.979, 1.001, 0.017)$ after RCA; class 2–$(3.953, 1.045, 0.010)$ before RCA, and $(1.984, 1.001, 0.022)$ after RCA. In this example, the condition numbers increased by a factor of 3.78 and 4.24 respectively for both classes.

In order to obtain a strict probabilistic justification for RCA, we listed in Section 5 the following assumptions:

1. The classes have multi-variate normal distributions.

2. All the classes share the same covariance matrix.

3. The points in each chunklet are an i.i.d sample from the class.

What happens when these assumptions do not hold?

The first assumption gives RCA its probabilistic justification. Without it, in a distribution-free model, RCA is the best linear transformation optimizing the criteria presented in Sections 3-4: maximal mutual information, and minimal within-chunklet distance. These criteria are reasonable as long as the classes are approximately convex (as assumed by the use of the distance between chunklet's points and chunklet's means). In order to investigate this point empirically, we used Mardia's statistical tests for multi-variate normality (Mardia, 1970). These tests (which are based on skewness and kurtosis) showed that all of the data sets used in our experiments are significantly non-Gaussian (except for the Iris UCI data set). Our experimental results therefore clearly demonstrate that RCA performs well when the distribution of the classes in the data is not multi-variate normal.

The second assumption justifies RCA's main computational step, which uses the empirical average of all the chunklets covariance matrices in order to estimate the global within class covariance matrix. When this assumption fails, RCA effectively extracts the shared component of all the classes covariance matrices, if such component exists. Figure 11 presents an illustrative example of the use of RCA on data from two classes with different covariance matrices. A quantitative measure of RCA's partial success in such cases can be obtained from the change in the *condition number* (the ratio between the largest and smallest eigenvalues) of the within-class covariance matrices of each of the classes, before and after applying RCA. Since RCA attempts to whiten the within-class covariance, we expect the condition number of the within-class covariance matrices to decrease. This is indeed the case for the various classes in all of the data sets used in our experimental results.

The third assumption may break down in many practical applications, when chunklets are automatically collected and the points within a chunklet are no longer independent of one another. As a result chunklets may be composed of points which are rather close to each other, and whose distribution does not reflect all the typical variance of the true distribution. In this case RCA's performance is not guaranteed to be optimal (see Section 8.2).

10. Conclusion

We have presented an algorithm which uses side-information in the form of equivalence constraints, in order to learn a Mahalanobis metric. We have shown that our method is optimal under several criteria. Our empirical results show that RCA reduces irrelevant variability in the data and thus leads to considerable improvements in clustering and distance based retrieval.

Appendix A. Information Maximization with non-invertible linear transformations

Here we sketch the proof of the claim made in Section 3.3. As before, we denote by $\hat{C}$ the average covariance matrix of the chunklets. We can rewrite the constrained expression from Equation 5 as:

$$\frac{1}{N}\sum_{j=1}^{n}\sum_{i=1}^{n_j}(x_{ji}-m_j)^t A^t A(x_{ji}-m_j) = tr(A^t A\hat{C}) = tr(A^t\hat{C}A)$$

Hence the Lagrangian can be written as:

$$\log|A\Sigma_x A^t| - \lambda(tr(A\hat{C}A^t)-1)$$

Differentiating the Lagrangian w.r.t A gives

$$\Sigma_x A^t(A\Sigma_x A^t)^{-1} = \lambda\hat{C}A^t$$

Multiplying by A and rearranging terms, we get: $\frac{I}{\lambda} = A\hat{C}A^t$. Hence as in RCA, A must whiten the data with respect to the chunklet covariance $\hat{C}$ in a yet to be determined subspace. We can now use the equality in (5) to find λ.

$$\begin{aligned} tr(A\hat{C}A^t) = tr(\frac{I}{\lambda}) = \frac{K}{\lambda} = 1 &\Longrightarrow \lambda = K \\ &\Longrightarrow A\hat{C}A^t = \frac{1}{K}I \end{aligned}$$

where K is the dimension of the projection subspace.

Next, since in our solution space $A\hat{C}A^t = \frac{1}{K}I$, it follows that $\log|A\hat{C}A^t| = K\log\frac{1}{K}$ holds for all points. Hence we can modify the maximization argument as follows

$$\log|A\Sigma_x A^t| \quad = \quad \log\frac{|A\Sigma_x A^t|}{|A\hat{C}A^t|} + K\log\frac{1}{K}$$

Now the optimization argument has a familiar form. It is known (Fukunaga, 1990) that maximizing the determinant ratio can be done by projecting the space on the span of the first K eigenvectors of $\hat{C}^{-1}\Sigma_x$. Denote by G the solution matrix for this unconstrained problem. This matrix orthogonally diagonalizes both $\hat{C}$ and Σ_x, so $G\hat{C}G^t = \Lambda_1$ and $G\Sigma_x G^t = \Lambda_2$ for Λ_1, Λ_2 diagonal matrices. In order to enforce the constraints we define the matrix $A = \sqrt{\frac{1}{K}}\Lambda_1^{-0.5}G$ and claim that A is the solution of the constrained problem. Notice that the value of the maximization argument does not change when we switch from A to G since A is a product of G and another full ranked matrix. It can also be shown that A satisfies the constraints and is thus the solution of the Problem (5).

Appendix B. Variance bound on the RCA covariance estimator

In this appendix we prove Inequality 12 from Section 5. Assume we have $N = nk$ data points $X = \{x_{ji}\}_{i=1,j=1}^{n,k}$ in n chunklets of size k each. We assume that all chunklets are drawn independently

from Gaussian sources with the same covariance matrix. Denoting by m_i the mean of chunklet i, the unbiased RCA estimator of this covariance matrix is

$$\hat{C}(n,k) = \frac{1}{n}\sum_{j=1}^{n}\frac{1}{k-1}\sum_{i=1}^{k}(x_{ji}-m_i)(x_{ji}-m_i)^T$$

It is more convenient to estimate the convergence of the covariance estimate for data with a diagonal covariance matrix. We hence consider a diagonalized version of the covariance, and return to the original covariance matrix toward the end of the proof. Let U denote the diagonalization transformation of the covariance matrix C of the Gaussian sources, that is, $UCU^t = \Lambda$ where Λ is a diagonal matrix with $\{\lambda_i\}_{i=1}^{D}$ on the diagonal. Let $Z = UX = \{z_{ji}\}_{i=1,j=1}^{n,k}$ denote the transformed data. Denote the transformed within class covariance matrix estimation by $\hat{C}^u(n,k) = U\hat{C}(n,k)U^t$, and denote the chunklet means by $m_i^u = Um_i$. We can analyze the variance of $\hat{C}^u$ as follows:

$$\begin{aligned} var(\hat{C}^u(n,k)) &= var[\frac{1}{n}\sum_{i=1}^{n}\frac{1}{k-1}\sum_{j=1}^{k}(z_{ji}-m_i^u)(z_{ji}-m_i^u)^T] \\ &= \frac{1}{n}var[\frac{1}{k-1}\sum_{j=1}^{k}(z_{ji}-m_1^u)(z_{ji}-m_1^u)^T] \end{aligned} \tag{15}$$

The last equality holds since the summands of the external sum are sample covariance matrices of independent chunklets drawn from sources with the same covariance matrix.

The variance of the sample covariance, assessed from k points, for diagonalized Gaussian data is known to be (Fukunaga, 1990)

$$var(\hat{C}_{ii}) = \frac{2\lambda_i^2}{k-1};\;\; var(\hat{C}_{ij}) = \frac{\lambda_i\lambda_j}{k};\;\; cov(\hat{C}_{ij},\hat{C}_{kl}) = 0$$

hence (15) is simply:

$$var(\hat{C}^u_{ii}) = \frac{2\lambda_i^2}{n(k-1)};\;\; var(\hat{C}^u_{ij}) = \frac{\lambda_i\lambda_j}{nk};\;\; cov(\hat{C}^u_{ij},\hat{C}^u_{kl}) = 0$$

Replacing $N = nk$, we can write

$$var(\hat{C}^u_{ii}) = \frac{2\lambda_i^2}{N(1-\frac{1}{k})};\;\; var(\hat{C}^u_{ij}) = \frac{\lambda_i\lambda_j}{N};\;\; cov(\hat{C}^u_{ij},\hat{C}^u_{kl}) = 0$$

and for the diagonal terms $\hat{C}^u_{ii}$

$$var(\hat{C}^u(\frac{N}{k},k)_{ii}) = \frac{2\lambda_i^2}{N(1-\frac{1}{k})} = \frac{k}{k-1}\frac{2\lambda_i^2}{N} \le \frac{k}{k-1}\frac{2\lambda_i^2}{N-1} = \frac{k}{k-1}var(\hat{C}^u(1,N)_{ii})$$

This inequality trivially holds for the off-diagonal covariance elements.

Getting back to the original data covariance, we note that in matrix elements notation $\hat{C}_{ij} = \sum_{q,r=1}^{D} \hat{C}^u_{qr} U_{iq} U_{jr}$ where D is the data dimension. Therefore

$$\frac{var[\hat{C}_{ij}(n,k)]}{var[\hat{C}_{ij}(1,nk)]} = \frac{\sum_{q,r=1}^{D} var[\hat{C}^u(n,k)_{qr} U_{iq} U_{jr}]}{\sum_{q,r=1}^{D} var[\hat{C}^u(1,nk)_{qr} U_{iq} U_{jr}]} \leq \frac{\sum_{q,r=1}^{D} \frac{k}{k-1} var[\hat{C}^u(1,nk)_{qr} U_{iq} U_{jr}]}{\sum_{q,r=1}^{D} var[\hat{C}^u(1,nk)_{qr} U_{iq} U_{jr}]} = \frac{k}{k-1}$$

where the first equality holds because $cov(\hat{C}^u_{ij}, \hat{C}^u_{kl}) = 0$.

Appendix C. Online RCA with chunklets of general size

The online RCA algorithm can be extended to handle a stream of chunklets of varying size. The procedure is presented in Algorithm 4.

Algorithm 4 Online RCA for chunklets of variable size

Input: a stream of chunklets where the points in a chunklet are known to belong to the same class.
Initialize W to a symmetric random matrix with $||W|| << 1$.
At time step T do:

- receive a chunklet $\{x_1^T, ..., x_n^T\}$ and compute its mean $m^T = \frac{1}{n} \sum_{i=1}^{n} x_i^T$;
- compute n difference vectors $h_i^T = x_i^T - m^T$;
- transform h_i^T using W, to get $y_i^T = W h_i^T$;
- update $W = W + \eta \sum_{i=1}^{n} (W - y_i^T (y_i^T)^t W)$.

where $\eta > 0$ determines the step size.

The steady state of the weight matrix W can be analyzed in a way similar to the analysis in Section 3. The result is $W = PE[\frac{1}{n} \sum_{i=1}^{n} (x_i^T - m^T)(x_i^T - m^T)^t]^{-\frac{1}{2}}$ where P is an orthonormal matrix, and so W is equivalent to the RCA transformation of the current distribution.

Appendix D. The expected chunklet size in the distributed learning paradigm

We estimate the expected chunklet size obtained when using the distributed learning paradigm introduced in Section 8. In this scenario, we use the help of T teachers, each of which is provided with a random selection of L data points. Let us assume that the data contains M equiprobable classes, and that the size of the data set is large relative to L. Define the random variables x_i^j as the number of points from class i observed by teacher j. Due to the symmetry among classes and among teachers, the distribution of x_i^j is independent of i and j, thus defined as x. It can be well approximated by a Bernoulli distribution $B(L, \frac{1}{M})$, while considering only $x \geq 2$ (since $x = 0, 1$ do not form chunklets). Specifically,

$$p(x = i | x \neq 0, 1) = \frac{1}{1 - p(x = 0) - p(x = 1)} \begin{pmatrix} L \\ i \end{pmatrix} (\frac{1}{M})^i (1 - \frac{1}{M})^{L-i} \quad i = 2, 3, ..$$

We can approximate $p(x=0)$ and $p(x=1)$ as

$$p(x=0) = (1-\frac{1}{M})^{L} \approx e^{-\frac{L}{M}} \quad , \quad p(x=1) = \frac{L}{M}(1-\frac{1}{M})^{L-1} \approx \frac{L}{M}e^{-\frac{L}{M}}$$

Using these approximations, we can derive an approximation for the expected chunklet size as a function of the ratio $r = \frac{L}{M}$

$$E(x|x \neq 0, x \neq 1) = \frac{\frac{L}{M} - p(x=1)}{1 - p(x=0) - p(x=1)} \simeq \frac{r(1-e^{-r})}{1-(r+1)e^{-r}}$$

References

Y. Adini, Y. Moses, and S. Ullman. Face recognition: The problem of compensating for changes in illumination direction. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 19(7):721–732, 1997.

A. Bar-Hillel, T. Hertz, N. Shental, and D. Weinshall. Learning distance functions using equivalence relations. In *International Conference on Machine Learning (ICML)*, pages 11–18, 2003.

S. Becker and G.E. Hinton. A self-organising neural network that discovers surfaces in random-dot stereograms. *Nature*, 355:161–163, 1992.

P.N. Belhumeur, J. Hespanha, and D. Kriegman. Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 19(7):711–720, 1997.

A.J. Bell and T.J. Sejnowski. An information-maximization approach to blind separation and blind deconvolution. *Neural Computation*, 7(6):1129–1159, 1995.

M. Bilenko, S. Basu, and R.J. Mooney. Integrating constraints and metric learning in semi-supervised clustering. In *International Conference on Machine Learning (ICML)*, pages 81–88, 2004.

C.L. Blake and C.J. Merz. UCI repository of machine learning databases, 1998. URL `http://www.ics.uci.edu/~mlearn/MLRepository.html`.

J. S. Boreczky and L. A. Rowe. Comparison of video shot boundary detection techniques. *SPIE Storage and Retrieval for Still Images and Video Databases IV*, 2664:170–179, 1996.

G. Chechik and N. Tishby. Extracting relevant structures with side information. In *Advances in Neural Information Processing Systems (NIPS)*, pages 857–864, 2003.

K. Fukunaga. *Statistical Pattern Recognition*. Academic Press, San Diego, 2nd edition, 1990.

P. Geladi and B. Kowalski. Partial least squares regression: A tutorial. *Analytica Chimica Acta*, 185:1–17, 1986.

T. Hastie and R. Tibshirani. Discriminant adaptive nearest neighbor classification and regression. In *Advances in Neural Information Processing Systems (NIPS)*, pages 409–415, 1996.

T. Jaakkola and D. Haussler. Exploiting generative models in discriminative classifiers. In *Advances in Neural Information Processing Systems (NIPS)*, pages 487–493, 1998.

D. Klein, S. Kamvar, and C. Manning. From instance-level constraints to space-level constraints: Making the most of prior knowledge in data clustering. In *International Conference on Machine Learning (ICML)*, pages 307–314, 2002.

R. Linsker. An application of the principle of maximum information preservation to linear systems. In *Advances in Neural Information Processing Systems (NIPS)*, pages 186–194, 1989.

K.V. Mardia. Measures of multivariate skewness and kurtosis with applications. *Biometrika*, 36: 519–530, 1970.

W.M. Rand. Objective criteria for the evaluation of clustering method. *Journal of the American Statistical Association*, 66(366):846–850, 1971.

N. Shental, A. Bar-Hillel, T. Hertz, and D. Weinshall. Computing gaussian mixture models with em using equivalence constraints. In *Advances in Neural Information Processing Systems (NIPS)*, pages 465–472, 2003.

N. Shental, T. Hertz, D. Weinshall, and M. Pavel. Adjustment learning and relevant component analysis. In *European Conf. on Computer Vision (ECCV)*, pages 776–792, 2002.

J.B. Tenenbaum and W.T. Freeman. Separating style and content with bilinear models. *Neural Computation*, 12(6):1247–1283, 2000.

B. Thompson. *Canonical correlation analysis: Uses and interpretation*. Sage Publications, Beverly Hills, 1984.

S. Thrun. Is learning the n-th thing any easier than learning the first? In *Advances in Neural Information Processing Systems (NIPS)*, pages 640–646, 1996.

N. Tishby, F.C. Pereira, and W. Bialek. The information bottleneck method. In *Proc. of the 37-th Annual Allerton Conference on Communication, Control and Computing*, pages 368–377, 1999.

M.A. Turk and A.P. Pentland. Face recognition using Eigenfaces. In *Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 586–591, 1991.

K. Wagstaff, C. Cardie, S. Rogers, and S. Schroedl. Constrained K-means clustering with background knowledge. In *International Conference on Machine Learning (ICML)*, pages 577–584, 2001.

E.P. Xing, A.Y. Ng, M.I. Jordan, and S. Russell. Distance metric learning with application to clustering with side-information. In *Advances in Neural Information Processing Systems (NIPS)*, pages 505–512, 2003.

Learning Distance Function by Coding Similarity

Aharon Bar Hillel Daphna Weinshall
The Hebrew university of Jerusalem
Jerusalem 91904
aharonbh,daphna@cs.huji.ac.il

September 26, 2006

Abstract

We consider the problem of learning a similarity function from a set of positive equivalence constraints, i.e. 'similar' point pairs. We define the similarity in information theoretic terms, as the gain in coding length when shifting from independent encoding of the pair to joint encoding. Under simple Gaussian assumptions, this formulation leads to a non-Mahalanobis similarity function which is efficient and simple to learn. This function can be viewed as a likelihood ratio test, and we show that the optimal similarity preserving projection of the data is a variant of Fisher Linear Discriminant. We also show that under some naturally occurring sampling conditions of equivalence constraints, this function converges to a known Mahalanobis distance (RCA). The suggested similarity function exhibits superior performance over alternative Mahalanobis distances learnt from the same data. Its superiority is demonstrated in the context of image retrieval and graph based clustering, using a large number of data sets - a facial image database, animal images, digits (MNist) and data sets from the UCI repository.

1 Introduction

Similarity functions play a key role in several learning and information processing tasks. One example is data retrieval, where similarity is used to rank items in the data base according to their similarity to some query item. In unsupervised graph based clustering, items are only known to the algorithm via the similarities between them, and the quality of the similarity function directly determines the quality of the clustering results. Finally, similarity functions are employed in several prominent techniques of supervised learning, from nearest neighbor classification to kernel machines. In the latter the similarity takes the form of a kernel function, and its choice is known to be a major design decision.

Good similarity functions can be designed by hand [13, 9] or learnt from data [6, 1, 15, 10, 11]. As in other contexts, learning can help; so far, the utility of distance function learning has been demonstrated in the context of image retrieval (e.g., [16]) and clustering (e.g., [15]). Since a similarity function operates on pairs of points, the natural input to a distance learning algorithm consists of equivalence constraints, which are pairs of points labeled as 'similar' or 'not-similar' (henceforth called positive and negative equivalence constraints respectively). Several scenarios have been discussed in which constraints, which offer a relatively weak form of supervision, are readily available, while labels are much harder to achieve [17]. For example, given temporal data such as video or surveillance data, constraints may be automatically obtained based on temporal coherence.

In this paper we derive a similarity measure from general principles, and propose a simple and practical similarity learning algorithm. In general the notion of similarity is somewhat vague, involving possibly conflicting intuitions. One intuition is that similarity should be related to commonalities, i.e., two objects are similar when they share many features. This direction was studied by [4], and

is most applicable to items described using discrete features, where the notion of common features is natural. Another intuition, suggested by [2], measures similarity by the plausibility of a common generative process. This notion draws attention to models of the hidden sources and the processes generating the visible items, which has two drawbacks: First, these models and processes are relatively complex and hard to estimate from data, specifically from equivalence constraints (indeed, learning is not discussed in [2]). Second, if such models are already known, the tasks of clustering and classification can be readily done using the models directly, and so similarity judgments are not required.

Our approach focuses on the purposes of similarity judgment, which is usually to decide whether two items belong to the same class. Hence, like [2], we relate similarity to the probability of two items belonging to the same cluster. However, unlike [2], we model the joint distribution of 'pairs from the same cluster' directly, and estimate it from positive equivalence constraints. Given this distribution, the notion of similarity is related to the shared information between two points, or equivalently, to the information one conveys about the other. This notion is formally presented in Section 2.1.

One of the main contributions of this paper is the specific application of this abstract similarity notion to continuous variables. Specifically, in Section 2.2 we develop this notion under Gaussian assumptions, deriving a simple similarity formula which is nevertheless different from a Mahalanobis metric. Intuitively, in the Gaussian setting the similarity between two point x and x' is computed by using x' to predict x via linear regression. The similarity is then related to $log\ p(x|x')$, which encodes the error of the prediction. Now learning the similarity requires only the estimation of two correlation matrices, which can be readily estimated from equivalence constraints.

The suggested similarity is strongly related to Fisher Linear Discriminant (FLD). The matrices employed in its computation are those involved in FLD, i.e., the within-class and between-class scatter matrices [5]. In Section 3 we show that FLD can be derived from our similarity as the optimal linear projection. Specifically, when coding similarity is regarded as a likelihood ratio test, FLD is the projection maximizing the expected margin of the test. In addition, we explore the connection between coding similarity and the Mahalanobis metric. We show that in a certain large sample limit, coding similarity converges to the Mahalanobis metric estimated by the RCA algorithm [1].

To evaluate our method, in Section 4 we experimentally explore two tasks: semi-supervised graph based clustering, and retrieval from a facial image database. Graph based clustering is evaluated using data sets from the UCI repository [3], as well as two harder data sets: the MNist data set of hand-written digits [18], and a data set of animal images [16]. We used the YaleB data set of facial images [8] for face retrieval experiments. In both tasks Gaussian Coding Similarity (GCS) significantly outperforms the Mahalanobis metric, learnt by two readily available algorithms [6, 1]. Note that the method of [6] employs both positive and negative equivalence constraints and is based on non-linear optimization (thus its computation is rather complex), while [1] offers a closed-form solution based on positive constraints alone. The computational cost of coding similarity is low, similar to RCA [1] and much smaller than in the method of [6].

2 Similarity based on Coding Length

We introduce the general notion of similarity based on coding gain in Section 2.1, and consider the implementation of this notion under Gaussian assumptions in Section 2.2.

2.1 General definition

Intuitively, two items are similar if they share common aspects, whereby one can be used to predict some details of the the other. Learning similarity is learning what aspects tend to be shared more then others, hence it is naturally related to the joint distribution $p(x, x'|H_1)$, where H_1 is the hypothesis stating that the two points originate from the same source. We estimate $p(x, x'|H_1)$, and define the similarity between two items x, x' to be the information one conveys about the other. We measure this information using the coding length, i.e. the negative logarithm of an event [14]. The similarity is therefore defined by the gain in coding length obtained by encoding x when x' is known.

$$codsim(x, x') = cl(x) - cl(x|x', H_1) = \log p(x|x', H_1) - \log p(x) \quad (1)$$

As stated in [2], this measurement can be also viewed as a log-likelihood ratio statistic:

$$\log p(x|x', H_1) - \log p(x) = \log \frac{p(x, x'|H_1)}{p(x)p(x')} = \log \frac{p(x, x'|H_1)}{p(x, x'|\mathrm{H}_0)} \tag{2}$$

where H_0 denotes the hypothesis stating independence between the points. The coding similarity is therefore the optimal statistic for determining whether two points are drawn from the same class or independently. We can also see from this last equation that it is a symmetric function.

Consider data sampled from several sources in R^d, i.e. $p(x) = \sum_{k=1}^{M} \alpha_k p(x|h_i)$, where $p(x|h_i)$ denotes the distribution of the i-th source. A simple form for $p(x, x'|H_1)$ is obtained when the two points are conditionally independent given the hidden source:

$$p(x, x'|H_1) = \sum_{k=1}^{M} \alpha_k p(x|h_k)p(x'|h_k) \tag{3}$$

This distribution, defined over pairs, corresponds to sampling pairs by first choosing the hidden source, followed by the independent choice of two points from this source. In Section 3.2 we discuss a common case in which the conditional independence between the points is violated.

2.2 Gaussian coding similarity

We now develop the coding similarity notion under some simplifying Gaussian assumptions:

- $p(x, x'|H_1)$ is Gaussian (in R^{2d})
- $p(x) = \int\limits_{x} p(x, x'|H) = \int\limits_{x'} p(x, x'|H)$

The second assumption is the reasonable (though not always trivially satisfied) demand that $p(x)$ should be the marginal distribution of $p(x, x'|H_1)$ w.r.t both arguments. It is clearly satisfied for distribution (3). It follows from the first assumption that $p(x)$ is also Gaussian (in R^d). The first assumption is clearly a simplification of the true data density in all but the most trivial cases. However, its value lies in its simplicity, which leads to a coding scheme that is efficient and easy to estimate. While clearly inaccurate, we propose here that this model can be very useful.

We assume w.l.o.g that the data's mean is 0 (otherwise, we can subtract it from the data), and so we can parameterize the two distributions using two matrices. Denoting the Gaussian distribution by $G(\cdot|\mu, \Sigma)$ we have

$$\begin{aligned} &p(x) = G(x|0, \Sigma_x) \\ &p(x, x'|H) = G(x, x'|0, \Sigma_{2x}) \qquad \Sigma_{2x} = \begin{pmatrix} \Sigma_x & \Sigma_{xx'} \\ \Sigma_{xx'} & \Sigma_x \end{pmatrix} \end{aligned} \tag{4}$$

where $\Sigma_x = E[xx^t]$, $\Sigma_{xx'} = E[x(x')^t]$. The conditional density $p(x|x', H)$ is also Gaussian $G(x|Mx', \Sigma_{x|x'})$ [12], with $M, \Sigma_{x|x'}$ given by

$$M = \Sigma_{xx'}\Sigma_x^{-1} \qquad \Sigma_{x|x'} = \Sigma_x - \Sigma_{xx'}\Sigma_x^{-1}\Sigma_{xx'} \tag{5}$$

Plugging this into Eq. (1), we get the following expression for Gaussian coding similarity:

$$\begin{aligned} \log p(x|x', H_1) - \log p(x) &= \log G(x|Mx', \Sigma_{x|x'}) - \log G(x|0, \Sigma_x) \\ &= \frac{1}{2}[\log \frac{|\Sigma_x|}{|\Sigma_{x|x'}|} + x^t\Sigma_x^{-1}x - (x - Mx')^t\Sigma_{x|x'}^{-1}(x - Mx')] \end{aligned} \tag{6}$$

The Gaussian coding similarity can be easily and almost instantaneously learnt from a set of positive equivalence constraints, as summarized in Algorithm 1. Learning includes the estimation of several statistics, mainly the matrices Σ_x,$\Sigma_{xx'}$, from which the matrices $M, \Sigma_{x|x'}$ are computed. Notice that each constraint is considered twice, once as (x, x') and once as (x', x), to ensure symmetry and to satisfy the marginalization demand. Given those statistics, similarity is computed using Eq. (8), which is based on Eq. (6) but with the multiplicative and additive constants removed.

Algorithm 1 Gaussian coding similarity

Learning procedure:

Input: a set of equivalence constraints $\{x_i, x'_i\}_{i=1}^N$, and optionally a dimension parameter k.

1. Compute the mean $Z = \frac{1}{2N}\sum_{i=1}^N [x_i + x'_i]$ and subtract it from the training data
2. Estimate $\Sigma_x, \Sigma_{xx'}$

$$\Sigma_x = \frac{1}{2N}\sum_{i=1}^N [x_i x_i^t + x'_i x_i'^t] \qquad \Sigma_{xx'} = \frac{1}{2N}\sum_{i=1}^N [x_i x_i'^t + x'_i x_i^t] \tag{7}$$

3. If dimensionality reduction is required, find the k eigenvectors with the highest eigenvalues of $\Sigma_x^{-1}\Sigma_{xx'}$ and put them into $A \in M_{d\times k}$.
   Let $\Sigma_x = A^t\Sigma_x A$, $\Sigma_{xx'} = A^t\Sigma_{xx'}A$, $Z = ZA$
4. Compute M and $\Sigma_{x|x'}$ according to Eq. (5).

Return $Z, M, \Sigma_x^{-1}, \Sigma_{x|x'}^{-1}$ and A (if computed).

Similarity computation for a pair (x, x'):

If A is defined, let $x = xA - Z$, $x' = x'A - Z$ else let $x = x - Z$, $x' = x' - Z$.

Return $$codsim(x, x') = x^t\Sigma_x^{-1}x - (x - Mx')^t\Sigma_{x|x'}^{-1}(x - Mx') \tag{8}$$

3 Relation to other methods

In this section we provide some analysis connecting Gaussian coding similarity as defined above to other known learning techniques. In Section 3.1 we discuss the underlying connection between GCS and FLD dimensionality reduction. In Section 3.2 we show that under certain estimation conditions, the dominant term in GCS behaves like a Mahalanobis metric, and specifically that it converges to the RCA metric [1].

3.1 Dimensionality reduction and the optimality of FLD

As defined in Eqs. (5)-(8), the coding similarity depends on two matrices only - the data covariance matrix Σ_x and the covariance between pairs from the same source $\Sigma_{xx'}$. To establish the connection to FLD, let us first consider the expected value of $\Sigma_{xx'}$ under distribution (3):

$$\begin{aligned} E_{p(x,x'|H_1)}[x(x')^t] &= \int_x\int_{x'}\sum_{k=1}^M \alpha_k p(x|h_k)p(x'|h_k)x(x')^t \\ &= \sum_{k=1}^M \alpha_k E_{p(x|h_k)}[x]\cdot E_{p(x'|h_k)}[(x')^t] = \sum_{k=1}^M \alpha_k m_k m_k^t \end{aligned} \tag{9}$$

The expected value above, which gives the convergence limit of $\Sigma_{xx'}$ as estimated in Eq. (7), is essentially the between-class scatter matrix S_B used in FLD [5]. The main difference between the estimation of $\Sigma_{xx'}$ in Eq. (7) and the traditional estimation of S_B is the training data, equivalence constraints vs. labels respectively. Also, while S_B is always of rank $k-1$ and so is its estimator based on labeled data, our estimator from Eq. (7) is usually of full rank.

When the data distributions $p(x, x'|H_1)$ and $p(x)$ lie in high dimensional space, in many cases the projection into a lower dimensional space may increase learning accuracy (by dropping irrelevant dimensions) and computational efficiency. We now characterize the notion of *optimal dimensionality reduction* based on the 'natural margin' of the likelihood ratio test. This test gives the optimal rule [14] for deciding between two hypotheses H_0 and H_1, where the data comes from a mixture $p(x) = \alpha p(x|H_0) + (1-\alpha)p(x|H_1)$:

$$decide\ H_1 \quad \Longleftrightarrow \quad \log\frac{p(X|H_1)}{p(X|H_0)} > \log\frac{1-\alpha}{\alpha} \tag{10}$$

Hypothesis margin: Let the label of point x be 1 if hypothesis H_1 is true, and -1 if H_0 is true. The natural margin of a point x can be defined as $y_i(\log \frac{p(x_i|h_1)}{p(x_i|h_0)} - \log \frac{1-\alpha}{\alpha})$.

Given this definition, the expected margin of the test is

$$\begin{aligned} & E_x[y(x)(\log \frac{p(x|h_1)}{p(x|h_0)} - \log \frac{1-\alpha}{\alpha})] \qquad (11) \\ = \quad & \alpha \int_x p(x|h_1)[\log \frac{p(x|h_1)}{p(x|h_0)} - \log \frac{1-\alpha}{\alpha}]dx - (1-\alpha) \int_x p(x|h_0)[\log \frac{p(x|h_1)}{p(x|h_0)} - \log \frac{1-\alpha}{\alpha}]dx \\ = \quad & \alpha D_{kl}[p(x|h_1)||p(x|h_0)] + (1-\alpha) D_{kl}[p(x|h_0)||p(x|h_1)] + (1-2\alpha) \log \frac{1-\alpha}{\alpha} \end{aligned}$$

Optimal dimensionality reduction: $A \in M_{d\times k}$ is the optimal linear projection from dimension d to k if it maximizes the expected margin defined above.

Theorem 1. *Assume Gaussian distributions $p(x, x'|H_1)$ in R^{2d} and $p(x)$ in R^d, and a linear projection $A \in M_{d\times k}$ where $z = A^t x$. For all $0 \le \alpha \le 1$, the optimal A*

$$A^* = \arg\max_{A \in M_{d\times k}} \alpha D_{kl}[p(z, z'|H_1)||p(z, z'|H_0)] + (1-\alpha) D_{kl}[p(z, z'|H_0)||p(z, z'|H_1)] \qquad (12)$$

is the FLD transformation. Thus A is composed of the k eigenvectors of $\Sigma_x^{-1}\Sigma_{xx'}$ with the highest eigenvalues.

The proof of this theorem is relatively complex and we only describe here a very general sketch. Since the distributions involved in Eq. (12) are Gaussian, the $D_{kl}[\cdot||\cdot]$ can be written in closed-form for which we can write an upper bound, where we use $A^t\Sigma_{x|x'}A$ to approximate $\Sigma_{z|z'}$. The approximate bound, for fixed α, can be shown to obtain its maximal value at the k eigenvectors of $\Sigma_{x|x'}^{-1}\Sigma_x$ with the highest eigenvalues. These vectors, in turn, are identical to the highest eigenvectors of $\Sigma_x^{-1}\Sigma_{xx'}$. Finally, it is shown that the optimum of the upper bound is also obtained by the original expression, using the same matrix A.

3.2 The Mahalanobis limit

Above we have considered specifically coding similarity with pairs distribution of the form (3). However, in practice the source of equivalence constraints may not produce an unbiased sample from distribution (3). Specifically, equivalence constraints which are obtained automatically, e.g. from a surveillance camera, are often biased and tend to include only very similar (close in the Euclidean sense) points in each pair. This happens since constraints are extracted based on temporal proximity, and hence include highly dependent points. When the points in all pairs are very close to each other, the best regression from one to the other is close to the identity matrix. The following theorem states that under these conditions, coding similarity converges to a Mahalanobis metric.

Theorem 2. *Assume that equivalence constraints are generated by sampling the first point x from $p(x)$ and then x' from a small neighborhood of x. Denote $\Delta = (x - x')/2$. Assume that the covariance matrix $\Sigma_\Delta < \epsilon\Sigma_x$, where $\epsilon > 0$ and $A < B$ stands for "$B - A$ is a p.s.d matrix". Then*

$$codsim(x, x') \underset{\varepsilon \to 0}{\to} - (x - x')^t (4\Sigma_\Delta)^{-1}(x - x') \qquad (13)$$

where the limit is in the sense that $g(x) \to f(x)$ iff $g(x)/f(x) \to 1$.

Proof. We concentrate on approximating the second term in Eq. (8), which involves both x and x'. Denote $\bar{x} = (x + x')/2$, so $x = \bar{x} + \Delta$, $x' = \bar{x} - \Delta$. We get the following estimates for Σ_x , $\Sigma_{xx'}$:

$$\begin{aligned} \Sigma_x &= \frac{1}{2}E(\bar{x} - \Delta)(\bar{x} - \Delta)^t + \frac{1}{2}E(\bar{x} + \Delta)(\bar{x} + \Delta)^t = \Sigma_{\bar{x}} + \Sigma_\Delta \\ \Sigma_{xx'} &= E(\bar{x} + \Delta)(\bar{x} - \Delta)^t = \Sigma_{\bar{x}} - \Sigma_\Delta \end{aligned}$$

We therefore see that $\Sigma_{xx'} = \Sigma_x - 2\Sigma_\Delta$, and obtain the following approximations for M,$\Sigma_{x|x'}$:

$$\begin{aligned} M &= \Sigma_{xx'}\Sigma_x^{-1} = (\Sigma_x - 2\Sigma_\Delta)\Sigma_x^{-1} = I - 2\Sigma_\Delta\Sigma_x^{-1} \ge I - 2\varepsilon \\ \Sigma_{x|x'} &= \Sigma_x - (I - 2\Sigma_\Delta\Sigma_x^{-1})(\Sigma_x - 2\Sigma_\Delta) = 4\Sigma_\Delta - 4\Sigma_\Delta\Sigma_x^{-1}\Sigma_\Delta \ge 4\Sigma_\Delta(I - \varepsilon) \end{aligned}$$

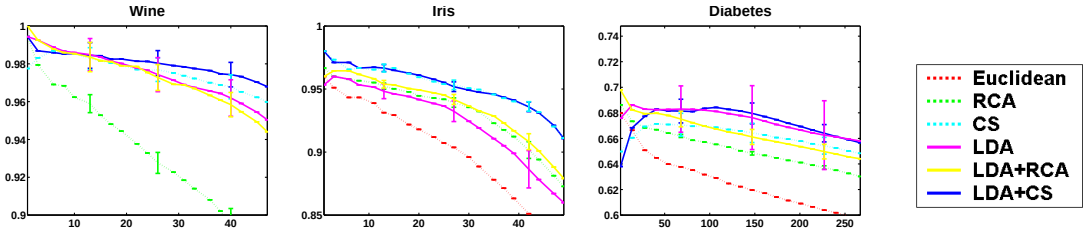


Figure 1: Cumulative neighbor purity curves for 5 data sets from the UCI repository. The Y axis shows the percentage of correct neighbors vs the number of neighbors considered (the X axis). In each graph we compare 3 methods: Gaussian coding similarity, RCA and the Euclidean metric, with and without constraint-based LDA. Results were averaged over 50 realizations.

Since the bounded matrix above ($\Sigma_\Delta \Sigma_x^{-1}$) is p.s.d, we get that $M = I + O(\epsilon)$ and $\Sigma_{x|x'} = 4\Sigma_\Delta(I + O(\epsilon))$.

Returning to Eq. (8), we note that the first term $0 < x^t \Sigma_x^{-1} x < \epsilon x^t \Sigma_\Delta^{-1} x$ is negligible w.r.t the second in the limit of $\epsilon \to 0$. Therefore the first term can be rigorously ommitted, and we get:

$$\begin{aligned} codsim(x, x') &\approx -(x - [I + O(\varepsilon)]x')^t \left[4\Sigma_\Delta(I + O(\varepsilon)\right]^{-1} (x - [I + O(\varepsilon)]x') \\ &\underset{\varepsilon \to 0}{\to} -(x - x')^t (4\Sigma_\Delta)^{-1}(x - x') \end{aligned}$$

Note that $codsim(x, x')$ is negative as appropriate, since it measures similarity rather than distance. □

The Mahalanobis matrix $\Sigma_\Delta = E_{p(x,x'|H_1)}[x - (x + x')/2]$ is actually the inner chunklet covariance matrix, as defined in [1]. It is therefore the RCA transformation, estimated from the population of 'near' point pairs.

4 Experimental validation

Following [17], we obtained equivalence constraints by simulating a *distributed learning* scenario, in which small subsets of labels are provided by a number of uncoordinated teachers. Accordingly, we randomly chose small subsets of data points from the dataset and partitioned each subset into equivalence classes. The constraints obtained from all the subsets are gathered and used by the coding similarity and competing methods. In all the experiments we chose the size of each subset to be $s = 2M$, where M is the number of classes in the data. In each experiment we used $\frac{N}{s}$ subsets, where N is the total number of points in the data. Notice that the number of equivalence constraints thus provided typically includes only a small fraction of all possible pairs of data points, which is $\mathcal{O}(N^2)$. Whenever tested, the method of [6] is given both the positive and the negative constraints, while coding similarity and RCA use only the positive constraints.

We first examine experimentally the relation between coding similarity, FLD and RCA, and report comparative results in Section 4.1. We then present experiments in semi-supervised clustering in Section 4.2, where the data is augmented by equivalence constraints. Finally, in Section 4.3 we test Gaussian coding similarity in a face retrieval task.

4.1 Coding similarity and FLD

Given the fundamental relation between GCS and constraints-based FLD, we first verify that GCS indeed offers independent contribution to similarity judgment. We are also interested in the comparison to the RCA algorithm, for which constraints-based FLD is also known to be the optimal dimensionality reduction [1]. Thus we have compared the Euclidean, RCA and GCS distances with and without constraints-based FLD over 9 data sets from the UCI repository. Figure 1 shows cumulative neighbor purity plots for 3 representative data sets, where the purity is averaged over all the points in the data, used as queries.

Several effects can be seen in this experiment: First, the Gaussian coding similarity has a significant advantage over RCA and the Euclidean metric before FLD is applied (in at least 7 of the 9 data sets).

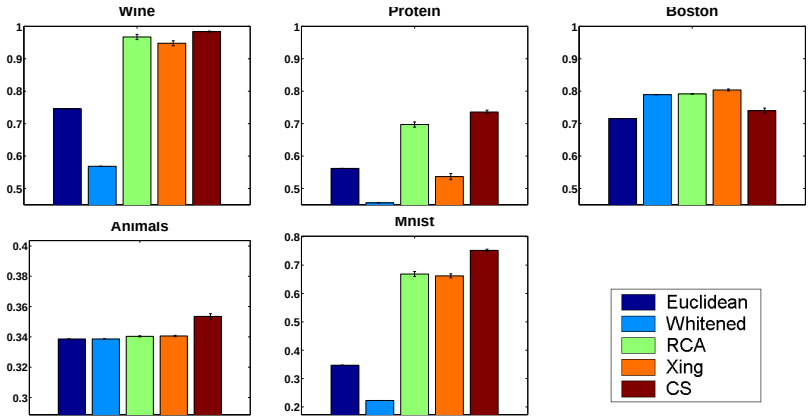


Figure 2: Clustering performance with the average linkage clustering algorithm using several similarity functions, with 5 data sets: 3 from the UCI repository [3], one collection of animal images, and digits from the MNist database. Performance is measured using the $F_{\frac{1}{2}} = \frac{2PR}{R+P}$ score, where P denotes precision rate and R denotes recall rate. The results are averaged over 50 constraint realizations. (Similar results were obtained using other agglomerative clustering algorithms.)

Second, the constraints-based FLD usually enhances purity for all three methods. Finally, in most cases, coding similarity offers additional contribution to neighbor purity beyond the improvement obtain by FLD, or by FLD+RCA.

4.2 Clustering with equivalence constraints

Graph based clustering includes a rich family of algorithms, among them are the relatively simple agglomerative linkage algorithms used here [5]. In graph based clustering, pairwise similarity is the sole source of information regarding the clustered data. Given equivalence constraints, one can adapt the similarity function to the specific problem, and improve clustering results considerably. In our experiments, we have evaluated clustering results using the following distance functions: a) the Euclidean metric, b) the 'whitening' Mahalanobis metric (Σ_x^{-1}), c) a Mahalanobis metric learnt by non-linear optimization [6] d) the RCA metric [1] (inverse inner class covariance, as estimated from the constraints), and e) Gaussian coding similarity. The distance functions were evaluated by applying the agglomerative average linkage algorithm to the similarity graphs produced. Clustering performance was assessed by computing the match between clustering results and the real data labels (known for all the data sets).

We have experimented with the UCI data sets and added two harder data sets, with 10 classes each: A subset of the MNist digits data set [18], and a data set of animals images [16]. For MNist, we randomly chose 50 instances from each digit, and represented the data using 50 PCA dimensions. The animals data set includes 565 images taken from a commercial CD. As in [16] we represent the images using Color Coherence Histograms (CCV) [7], containing information about color distribution and color continuity in the image. The vectors were then reduced to 100 PCA dimensions.

We tested the different similarities in the original space first, and after reducing the data dimension to the number of classes using constrained-based FLD. The results after FLD , which are usually better, are summarized in Figure 2 for 3 representative UCI datasets, and the two image datasets. Overall, coding similarity (rightmost bar, in brown) has an advantage over all the Mahalanobis metrics in 7 of the 11 data sets. The results in the original space show a similar trend.

4.3 Facial image retrieval

We tested the retrieval performance of coding similarity using the YaleB data set [8]. This data set contains 64 images per person of 30 people, where the variability is mainly due to change of illumination. The images were aligned using optical flow, and then reduced to 60 PCA dimensions. From each class, we randomly chose 48 images to be part of the 'data base', and used the remaining 16 as queries presented to the data base. We learned two Mahalanobis distances [6, 1] and coding simi-

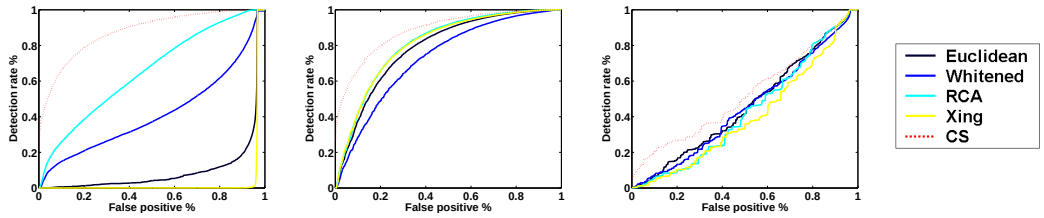


Figure 3: ROC curves for several methods in a face retrieval task. **Left:** Retrieval of test images from constrained classes using 60 PCA dimensions. **Middle:** Retrieval of images from constrained classes using 18 FLD dimensions. **Right:** Retrieval of test queries from unconstrained classes using 18 FLD dimensions. Results were averaged over 20 constraints realizations.

larity using constraints obtained from 25 of the 30 classes, and then evaluated retrieval performance for images from both constrained and unconstrained classes. Notice that for unconstrained classes the task is much harder, and any success shows inter-class generalization, since images from these classes were not used during training.

The performance of the three learning methods, as well as the Euclidean and whitened distances which provide the baseline, are shown in Figure 3. We can see that coding similarity is clearly superior to other methods in the retrieval of faces from known classes. In contrast to other methods, it operates well even in the original 60 dimensional space. It also has a small advantage in the 'learning-to-learn' scenario, i.e., in the retrieval of faces from unconstrained classes.

5 Summary

We described a new measure of similarity between two datapoints, based on the gain in coding length of one point when the other is known. This similarity measure can be efficiently computed from positive equivalence constraints. We showed the relation of this measure to Fisher Linear Discriminant, and to a known Mahalanobis distance that can also be learnt efficiently. We demonstrated overall superior performance in clustering and retrieval, using a battery of experiments on a large number of datasets.

References

[1] Bar Hillel A., Hertz T., Shental N., and Weinshall D. Learning a mahalanobis metric from equivalence constraints. *JMLR*, 6(Jun):937–965, 2005.

[2] Kemp C., Bernstein A., and Tenenbaum J. B. A generative theory of similarity. In *The Twenty-Seventh Annual Conference of the Cognitive Science Society*, 2005.

[3] Blake C.L. and Merz C.J. UCI repository of machine learning databases, 1998.

[4] Lin D. An information-theoretic definition of similarity. In *ICML*, 1998.

[5] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. John Wiley and Sons Inc., 2001.

[6] Xing E.P., Ng A.Y., Jordan M.I., and Russell S. Distance metric learnign with application to clustering with side-information. In *NIPS*, volume 15. The MIT Press, 2002.

[7] Pass G., Zabih R., and Miller J. Comparing images using color coherence vectors. In *ACM Multimedia*, pages 65–73, 1996.

[8] A.S. Georghiades, P.N. Belhumeur, and D.J. Kriegman. From few to many: Generative models for recognition under variable pose and illumination. In *IEEE Int. Conf. on Automatic Face and Gesture Recognition*, pages 277–284, 2000.

[9] Zhang H., Berg A.C., Maire M., and Malik J. Svm-knn: Discriminative nearest neighbor classification for visual category recognition. In *CVPR*, 2006.

[10] Bilenko M., Basu S., and Mooney R.J. Integrating constraints and metric learning in semi-supervised clustering. In *ICML*, pages 81–88, 2004.

[11] Cristianini N., Kandola J., Elissee A., and Shawe-Taylor J. On kernel target alignment. In *NIPS*, volume 14, Cambridge, MA, 2002. MIT Press.

[12] Barnett S. *Matrix methods for engineers and scientists*. McGraw Hill, 1979.

[13] Belongie S., Malik J., and Puzicha J. Shape matching and object recognition using shape context. *PAMI*, 24:509–522, 2002.

[14] Cover T. and Thomas J. *Elements of Information Theory*. Wiley, 1991.

[15] Hertz T., Bar-Hillel A., and Weinshall D. Boosting margin based distance functions for clustering. In *ICML*, 2004.

[16] Hertz T., Bar-Hillel A., and Weinshall D. Learning distance functions for image retrieval. In *CVPR*, 2004.

[17] Hertz T., Shental S., Bar-Hillel A., and Weinshall D. Enhancing image and video retrieval: Learning with equivalence constraints. In *CVPR*, 2003.

[18] LeCun Y., Bottou L., Bengio Y., and Haffner P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, November 1998.

Chapter 3

Learning object class recognition

This chapter is based on the following pulications:

[C] A. Bar-Hillel, T. Hertz and D. Weinshall: "Object class recognition by boosting a part based model", in *Conference on Computer Vision and Pattern Recognition (CVPR)*, volume I, 702-709, 2005.

[D] A. Bar-Hillel, T. Hertz and D. Weinshall: "Efficient learning of relational object class models", in *International Conference of Computer Vision (ICCV)*, volume II, 1762-1769, 2005.

[E] A. Bar-Hillel and D. Weinshall: "Subordinate class recognition using relational object models", accepted for publication in *Advances in Neural Information Processing Systems (NIPS)*, 2006.

Section 3.1 is a unified report summarizing publications [C] and [D], and adds to them some formal analysis and empirical results. Section 3.2 contains publication [E] in the format in which it was accepted for publication.

Efficient Learning of Relational Object Class Models

Aharon Bar-Hillel **Daphna Weinshall**
Computer Science Department and the center for neural computation
The Hebrew University of Jerusalem
Jerusalem, Israel 91904

Abstract

We present an efficient method for learning part-based object class models. The models include part appearance, as well as location and scale relations between parts. The object class is generatively modeled using a simple Bayesian network with a central hidden node containing location and scale information, and nodes describing object parts. The model's parameters, however, are optimized to reduce a loss function of the training error, as in discriminative methods. We show how boosting techniques can be extended to optimize the relational model proposed, with complexity linear in the number of parts and the number of features per image. This efficiency allows our method to learn relational models with many parts and features. The method has an advantage over purely generative and purely discriminative approaches, since the former are limited to a small number of parts and features, while the latter neglect geometrical relations between parts. Experimental results are described, using some bench-mark data sets and three sets of newly collected data, showing the relative merits of our method in recognition and localization tasks.

1 Introduction

One of the important organization principles of object recognition is the categorization of objects into object classes. Categorization is a hard learning problem due to the large inner-class variability of object classes, in addition to the "common" object recognition problems of varying pose and illumination. Recently, there has been a growing interest in the task of object class recognition [21, 20, 23, 2, 10, 6, 18] which can be defined as follows: given an image, determine whether the object of interest appears in the image. In many cases the localization of the object in the image is also sought.

Following previous work [20, 17], we represent an object using a part-based model (see illustration in Figure 1). Such models can capture the essence of most object classes, since they represent both parts' appearance and invariant relations of location and scale between the parts. Part-based models are somewhat resistant to various sources of variability such as within-class variance, partial occlusion and articulation, and they are potentially convenient for indexing in a more complex system [8, 6].

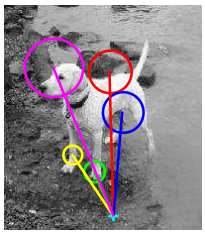


Figure 1: Dog image with our learnt part-based model drawn on top. Each circle represents a part in the model. The parts relative location and scale are related to one another through a hidden center (better viewed in color).

Part-based approaches to object class recognition can be crudely divided into two types: (1) 'generative-model-based' methods [20, 6, 18, 16, 21] and (2) 'discriminative-model-free' methods [2, 1, 10, 27, 19]. In the 'Generative-model based' approach, a probabilistic model of the object class is learnt by likelihood maximization. The likelihood ratio test is used to classify new images. The main advantage of this approach is the ability to model relations between object parts. In addition, domain knowledge can be incorporated into the model's structure and priors. 'Discriminative-model-free' methods seek a classification rule which discriminates object images from background images. The main advantage of discriminative methods is the direct minimization of a classification-based error function, which typically leads to superior classification results [5]. Additionally since these methods are model-free, they are usually computationally efficient.

In our current work, we suggest to combine the two approaches in order to enjoy the benefits of both worlds: the modeling power of the generative approach, with the accuracy and efficiency of discriminative optimization. We motivate this idea in Section 2 using general considerations, and as a solution to some problems encountered in related work. Our argument relies on two basic claims. The first is that feature relations are powerful cues for recognition, and perhaps indispensable cues for semantical recognition-related tasks like object localization or part identification. Proper representation of such relations requires a generative approach. On the other hand, we argue that generative learning procedures are inadequate in the specific context of learning from unsegmented images, due to essential computational and functional reasons. We therefore propose to replace maximum-likelihood optimization in the generative learning, by the discriminative optimization of the classifiers' parameters.

Specifically, we suggest a novel learning method for classifiers based on a simple part based model. The model, described in Section 3, is a simple 'star'-like Bayesian network, with a central hidden node describing the objects location and scale. The location and scale of the different parts depend only on the central hidden variable, and so the parts are conditionally independent given this variable. Such a model allows us to represent part relations with low inference computational complexity. Models of similar topology are implicitly or explicitly considered in [8, 6, 21]. While using a generative object model, we optimize its parameters by minimizing a loss over the training error, as done in discriminative learning. We show how a standard boosting approach can be naturally extended to learn such a model with conditionally independent parts. Learning time is linear in the number of parts and the number of feature extracted per image. Beyond this extension, we consider a wider family of gradient descent optimization algorithms, of which the extended boosting is a special case. Optimal performance is empirically achieved using algorithms from this family that are close to the extended boosting, but not identical to it. The discriminative optimization methods are discussed in Section 4.

Our experimental results are described in Section 5. We compare the recognition performance of our algorithm to several state-of-the-art generative and discriminative methods, using the benchmark data sets of [20]. When similar features are used by all methods, our method usually outperforms purely generative and discriminative competitors. When compared with generative methods [20, 21], our model is learned more efficiently and includes more selective features, which allows for the construction of models with many discriminative parts. When compared to a purely discriminative approach [2], our model's main advantage is the inclusion of informative part relations. This information improves recognition performance, and allows localization and identification of multiple instances in an image.

In order to challenge our method, we collected three more difficult data sets containing images of chairs, dogs and humans, with matching backgrounds (we have made this data publicly available online). We used these data sets to test the algorithm's performance under harder conditions, with high visual similarity between object and background, and large pose and scale variability. We investigated the relative contribution of the appearance, location and scale components of our model, and showed the importance of incorporating location relations between object parts. In another experiment we investigated the contribution of using large numbers of parts and features, and demonstrated their relative merits. We experimented with a generic interest point detector [26], as well as with a discriminative interest point detector [11]; our results show a small advantage for the latter. In addition, we showed that the classifiers learnt perform well against new, unseen backgrounds. Finally, we demonstrate the utility of the model in a localization task, using the UIUC cars side benchmark dataset [24]. In this task we efficiently scan the image to find the exact location

of one or more object instances. The localization performance achieved is comparable to the best available methods.

2 Why mix discriminative learning with generative modeling: motivation and related work

In this section we describe the main arguments for combining generative and discriminative methods in the context of learning from unsegmented images. In Section 2.1 we review the distinction between the generative and discriminative paradigms, and assess the relative merits of each approach in general. We next discuss the specific problem of learning from unsegmented images in Section 2.2, and characterize it as learning from unordered feature sets, rather than data vectors. In Section 2.3 we claim that relations between features, best represented in a generative framework, are useful in the context of learning from unordered sets, and are specifically important for semantical recognition-related tasks. In Section 2.4 we argue that generative maximum-likelihood learning is highly problematic in the context of learning from unsegmented images. Specifically, we argue that such learning suffers from inherent computational problems, and that it is likely to exhibit deficient feature pruning characteristics. To solve these problems while keeping the important information of feature relations, we propose to combine the generative treatment of relations with discriminative learning techniques. In Section 2.5 we briefly review how feature relations are handled in related discriminative methods.

2.1 Discriminative and generative learning

Generative classifiers learn a model of the probability $p(x|y)$ of input x given label y. They then predict the input labels by using Bayes rule to compute $p(y|x)$ and choosing the most likely label. With 2 classes $y \in \{-1, 1\}$, the optimal decision rule is the log likelihood ratio test, based on the statistic:

$$log \frac{p(x|y=1)}{p(x|y=-1)} - \nu \tag{1}$$

where ν is a constant threshold. The models $p(x|y=1)$ and $p(x|y=-1)$ are learnt in a maximum likelihood framework (or maximum-a-posteriori when a useful prior is available).

Discriminative classifiers do not learn probabilistic class models. Instead, they learn a direct map from the input space X to the labels. The map's parameters are chosen in a way that minimizes the training error, or a smooth loss function of it. With two labels, the classifier often takes the form $sign(f(x))$, with the interpretation that $f(x)$ models the log likelihood ratio statistic.

There are several compelling arguments in the learning literature which indicate that discriminative learning is preferable to generative learning in terms of classification performance. Specifically, learning a direct map is considered an easier task than the reliable estimation of $p(x|y)$ [28]. When classifiers with the same functional form are learned in both ways, it is known that the asymptotic error of a reasonable discriminative classifier is lower or equal to the error achievable by a generative classifier [5].

However, when we wish to design (or choose) the functional form of our classifier, generative models can be very helpful. When building a model of $p(x|y)$ we can use our prior knowledge about the problem's domain to guide our modeling decisions. We can make our assumptions more explicit and gain semantic understanding of the model's components. Specifically, the generative framework readily allows for the modeling of parts relations, while providing us with a rich toolbox of theory and algorithms for inference and relations learning. It is plausible to expect that a carefully designed classifier, whose functional form is determined by generative modeling, will give better performance than a classifier from an arbitrary parametric family.

These considerations suggest that a hybrid path may be beneficial. More specifically, choose the functional form of the classifier using a generative model of the data, then learn the model's parameters in a discriminative setting. While the arguments in favor of this idea as presented so far are very general, we next claim that when learning from images in particular, this idea can overcome several problems in current generative and discriminative approaches.

2.2 Learning from features sets

Our primary problem is object class recognition from unaligned and unsegmented images, which are binary labeled as to whether or not they contain an object from the class. It is therefore a binary classification problem, where the input is a set of features rather than an ordered vector of features, as in standard learning problems. This is a very important difference: vector representation implicitly assumes that measurements of the 'same' quantities are made for all data instances and stored in corresponding indexes of the data vectors. The 'same' features in different data vectors are assumed to have the same fixed, simple relation with the class label (the same 'role'). Such implicit correspondence is often hard to find in bottom up image representation, in particular when feature maps or local descriptors sets are detected with interest point detectors.

Thus we adopt the view of image representation as a set of features. Each feature has a location index, but unlike an element in a vector, its location does not imply a pre-determined fixed 'role' in the representation. Instead, only relations between locations are meaningful. Such representations present a challenge to current learning theory and algorithms, which are well developed primarily for vectorial input.

A second inherent problem arises because the relevant feature set representations usually contain a large number of spurious features. The images are unsegmented, and therefore many features may not represent the object of interest at all (but background information), while many other features may duplicate each other. Thus feature pruning is an important part of the learning problem.

2.3 Semantics and part relations

The lack of feature correspondence between images can be handled in two basic ways: either try to establish correspondence, or give it up to begin with. Without correspondence, images are typically represented by some statistical properties of the feature set, without assigning roles to specific image features. A notable example is the feature histogram, used for example in [10, 4, 13] and most of the methods in [7]. These approaches are relatively simple and in some cases give excellent recognition results. In addition they tend to have good invariance properties, as the use of invariant features directly gives invariant classifiers. Most of these approaches do not consider feature relations, mainly because of their added complexity (an exception is [13]). The main drawback of this framework is the complete lack of image semantics. While good recognition rates can be achieved, further recognition related tasks like localization or part identification cannot be done in this framework, as they require identifying the role of specific features.

The alternative research choice, which we adopt in the current paper, seeks to identify and correspond features with the same 'role' in different images. This is done explicitly in generative modeling approaches [20, 21, 18], using the notion of a probabilistically modeled 'part'. The 'part' is an entity with a fixed role (probabilistically modeled), and its instantiation in each image should be chosen from the set of available image features. Discriminative part based methods [2, 1, 23, 17] use a more implicit 'part' notion, and their degree of commitment to finding semantically similar features in images varies. The important advantage of identifying parts with fixed roles over the images is the ability to perform image understanding tasks beyond mere recognition.

When looking in images for parts with fixed roles, feature relations (mainly location and scale relations) provide a powerful, perhaps indispensable cue. Basing part identity on appearance criteria alone is possible, and in [1, 27] it leads to very good recognition results. However, as reported in [1], the stability of correct part identification is low, and localization results are mediocre. Specifically, it was found that typically less than 50% of the instantiating features were actually located on the object. Instead, many feature rely on the difference in background context between object and non-object images. Conversely, good localization results are reported for methods based on generative models [20, 21]. In [23] a detection task is considered in a discriminative framework. In order to achieve good localization, gross part relations are introduced as additional features into the discriminative classifier.

2.4 Learning in Generative models

We now consider generative model learning when the input is a set of unsegmented images. In this scenario, the model is learnt from a set of object images alone, and its parameters are chosen to maximize the likelihood of the image set (sometimes under a certain prior over models). We describe two inherent problems of this maximum likelihood approach. In Section 2.4.1 we claim that such learning involves an essential tradeoff, where computational efficiency is traded for weaker modeling which allows repetitive parts. In Section 2.4.2 we review how this problem is handled in some current generative models. In Section 2.4.3 we maintain that generative learning is not well adjusted to feature pruning, and becomes problematic when rich image representations are used.

2.4.1 The computational problem

Assume that the image is represented as a set of features, with spatial relations between features as described in Sections 2.2-2.3. Computing a generative model from such a set of features is hard. Denote the feature set of image I by $F(I)$, and the number of features in $F(I)$ by N. While the input is a feature set, the generative model typically specifies the likelihood $P(V|M)$ for an ordered part vector $V = (f_1, .., f_P)$ of P parts. The problem of learning from unordered sets is tackled by considering all the possible vectors V that can be formed using the feature set. Legitimate part vectors should have no repeated features, and there are $O(N^P)$ such vectors. Thus, the image likelihood $P(I|M)$ requires marginalization[1] over all such vectors. Assuming uniform prior over these vectors, we have

$$P(I|M) = \sum_{\substack{V=(x_1,..,x_P)\in F(I)^P \\ x_i \neq x_j \, if \, i \neq j}} P(V|M) \tag{2}$$

Efficient likelihood computation in relational models is only possible via the decomposition of the joint probability using conditional independence assumptions, as done in graphical models. Such decomposition specifies the probability as a product of local terms, each depending on a small subset of parts. For a part vector $V = (f_1, .., f_P)$

$$P(V|M) = \prod_c \Psi_c(V|_{S_c}) \tag{3}$$

where $S_c \subset \{1, .., P\}$ are index subsets and $V|_S = \{f_i : i \in S\}$. Using dynamic programming, inference and marginalization are exponential in the 'induced width' g of the related graphical model, which is usually relatively low (note that for trees, $g = 2$ only).

The summation in Eq. (2) does not lend itself easily to such simplifications, however. We therefore make the following approximation, in which part vectors with repetitive features are allowed

$$P(I|M) = \sum_{\substack{(x_1,..,x_P)\in F(I)^P \\ x_i \neq x_j \;\; for \;\; i \neq j}} \prod_c \Psi_c(V|_{S_c}) \approx \sum_{(x_1,..,x_P)\in F(I)^P} \prod_c \Psi_c(V|_{S_c}) \tag{4}$$

This approximation is essential to make efficient marginalization possible. If feature repetition is not allowed, global dependence emerges between the features assigned to the different parts (as they cannot overlap). As a result we get global constraints, and efficient enumeration becomes impossible.

The approximation in (4) may appear minor, which is indeed the case when a fixed, 'reasonable' part based model is applied to an image. In this case, typically, parts are characterized by different appearance and location models, and part vectors with repetitive parts have low insignificant probability. But during learning, approximation (4) brings about a serious problem: when vectors with feature repetitions are

[1] Alternatively, one may approximate the sum in Eq. (2) by a max operator, looking for the best model interpretation in the image. This does not affect the computation considerations discussed here.

allowed, learning may result in models with many repetitive parts. In fact, standard maximum likelihood has a strong tendency to choose such models. This is because it can easily increase the likelihood by choosing the same parts with high likelihood, over and over again.

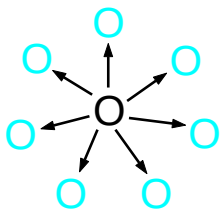


Figure 2: A "star" graphical model. peripheral nodes, shown in blue, are related only via a hidden central node. Such a model is used in our work, as well as in [21]. If (i) feature repetition is allowed (as in Eq. (4)), and (ii) model parameters are chosen to maximize the likelihood of the best object occurrence, then all the peripheral nodes are optimized to represent the same part.

The intuition above can be made precise in the simple case in which a 'star' model is used (see Figure 2) and the sum over all hypotheses is approximated by the single best features vector. In this extreme case, the maximal likelihood is achieved when all the peripheral parts models are identical. We specifically consider this model in Section 3 and prove the last statement in Appendix A. The proof doesn't directly apply when a sum over all the feature vectors is used, but as this sum is usually dominated by only a few vectors, part repetition is likely to occur in this case too.

Thus, in conclusion, we see that in a pure generative framework, one needs to choose between computational efficiency and the risk of part duplication.

2.4.2 How is this computational problem handled?

Several recent approaches use generative modeling for object class recognition [20, 16, 21, 18]. In [20, 16] a full relational model is used. The probability $P((f_1, .., f_P)|M)$ in this model cannot be decomposed into the product of local terms, due to the complex probabilistic dependencies between all of the model's parts (in graphical models terminology the model is a single large clique). As a result, both learning and recognition are exponential in the number of model parts, which limits the number of parts that can be used (up to 7 in [20] and 4 in [16]), and the number of features per image ($N = 30, 20$ respectively).

In [21] a decomposable model is proposed with a 'star'-like topology. This reduces the complexity of recognition significantly. However, learning remains essentially exponential, in order to avoid part repetition in the learnt model.

The computational problem (as well as the feature pruning problem, discussed in the next section) is completely avoided in the case of learning from segmented images, as done in [18]. Here the input is a set of object images, with manually segmented parts and manual part correspondence between images. In this case learning is reduced to standard maximum likelihood estimation of vectorial data.

2.4.3 Feature pruning

We argued in Section 2.2 that feature pruning is necessary when learning from images. P, the number of parts in the model, is often much smaller than the number of features per image N. This is usually not the case in classical applications of generative modeling, in which data is typically described as a relatively small feature vector.

When $P << N$, maximum likelihood chooses to model only parts with high likelihood - often parts which are highly repetitive in images, with repetitive relations. This optimization policy has a number of drawbacks. On the one hand, it introduces a preference for simple parts, as these tend to have low variability through images, which gives rise to high likelihood scores. It also introduces preference for features which are

frequent in natural images, whether they belong to the object or not. On the other hand, there is no explicit preference for discriminative parts, nor any preference for feature diversity. As a result, certain aspects of the object may be extensively described, while others are neglected. The problem may be intuitively summarized by stating that generative methods can describe the data, but they cannot choose what to describe. Additional task related signal, external to the data, is needed, and is most readily provided by labels.

In [20, 16], initial feature pruning is obtained by using the Kadir and Bradey detector [26], which finds relatively diverse, high entropy regions in the image. Explicit preference is given for features with large scale, which tend to be more discriminative. In addition, they limit the number of features per image ($N = 20, 30$). To some extent, the burden of feature pruning is placed on the pre-learning feature detection mechanisms. However, with such a small number of features per image, objects do not always get sufficient coverage. In fact, learning is very sensitive to the fine tuning of the feature pruning mechanism.

In [21], where a 'star'-like decomposable model is used, more parts and features are used in the generative learning experiments. Surprisingly, the results do not show obvious improvement. Increasing the number of parts P and features N_f does not typically reduce the error rates, since many of the additional features turn out to be irrelevant, which makes feature pruning harder. In Section 5 we investigate the impact P and N_f have on performance for models similar to those used by [21], but optimized discriminatively. In our experiments extra information (increased N_f) and modeling power (increased P) clearly lead to better performance.

2.5 Relations in discriminative methods

Many part based object class recognition methods are mostly discriminative [2, 1, 17, 25, 23]. In most of these methods, spatial relations between parts are not considered at all. While some of these systems exhibit state-of-the-art recognition performance, they are usually lacking in further, more semantical tasks as localization and part identification, as described in Section 2.3. In the 'fragment based' approach of [17, 25] relations are not used, but when the same approach is applied to segmentation, which requires richer semantics, fragment relations are incorporated [9].

One way to incorporate part relations into a discriminative setting is used by the object detection system of [23]. The task is localization, and it requires the exact correspondence and the identification of parts. To achieve this, qualitative location relations between fragment features are also considered as features, creating a very large and sparse feature vector. Discriminative learning in this very high dimensional space is then done using a specific feature-efficient learning algorithm. The relational features in this scheme are highly qualitative (for example, 'fragment a in to the left and bottom of fragment b'). Another problem with this approach is that supervised learning from high dimensional sparse vectors is a hard problem, often requiring dimensionality reduction to enable efficient learning.

In this context, our main contribution may be the design of a relatively simple and efficient technique for the introduction of relational information into the discriminative framework of boosting. As such, our work is related to the purely discriminative techniques used in [2, 1]. In spirit, our work has some resemblance to the work of [3], in which relational context information is incorporated into a boosting process. However, the techniques we use and the task we consider are quite different.

3 The generative model

We represent an input image using a set of local descriptors obtained using an interest point detector. Some details regarding this process are given in Section 3.1. We then define a classifier over such sets of features using a generative object model. The model and the resulting classifier are described in Sections 3.2 and 3.3 respectively.

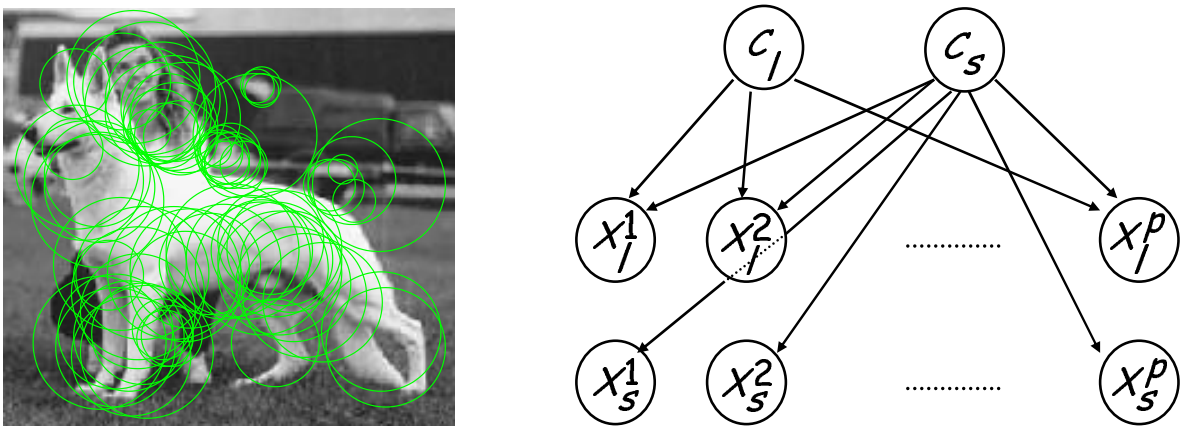


Figure 3: **a)** Output of the KB interest point (or feature) detector, marked with green circles. **b)** A Bayesian network specifying the dependencies between the hidden variables C_l, C_s and the parts scales and locations X_l^k, X_s^k for $k = 1, .., P$. The part appearance variables X_a^k are independent, and so they do not appear in this network.

3.1 Feature extraction and representation

Our feature extraction and representation scheme mostly follows the scheme used in [20]. Initially, images were rescaled to have a uniform horizontal length of 200 pixels. We experimented with two feature detectors: (1) Kadir & Brady (KB) [26], and (2) Gao & Vasconcellos (GV) [11][2]. The KB detector is a generic detector. It searches for circular regions of various scales, that correspond to the maxima of an entropy based score in scale space. The GV detector is a discriminative saliency detector, which searches for features that permit optimal discrimination between the object class and the background class. Given a set of labeled images from two classes, the algorithm finds a set of discriminative filters based on the principle of Maximal Marginal Diversity (MMD). It then identifies circular salient regions at various scales by pooling together the responses of the discriminative filters.

Both detectors produce an initial set of thousands of salient candidates for a typical image (see example in Figure 3a). As in [20], we multiply the saliency score of each candidate patch by its scale, thus creating a preference for large image patches, which are usually more informative. A set of N_f high scoring features with limited overlap is then chosen using an iterative greedy procedure. By varying the amount of overlap allowed between selected features we can vary the number of patches chosen: in our experiments we varied N_f between 13 and 513. After their initial detection, selected regions are cropped from the image and scaled down to 11×11 pixel patches. The patches are then normalized to have zero mean and variance of 1. Finally the patches are represented using their first 15 DCT coefficients (not including the DC).

To complete the representation, we concatenate 3 additional dimensions to each feature, corresponding to the x and y image coordinates of the patch, and its scale respectively. Therefore each image I is represented using an unordered set $F(I)$ of 18 dimensional vectors. Since our suggested algorithm's runtime is only linear in the number of image features, we can represent each image using a large number of features, typically in the order of several hundred features per image.

3.2 Model structure

We consider a part-based model, where each part in a specific image I_i corresponds to a patch feature from $F(I_i)$. Denote the appearance, location and scale components of each vector $x \in F(I)$ by x_a, x_l and x_s respectively (with dimensions 15,2,1), where $x = [x_a, x_l, x_s]$. We can assume that the appearance of different parts is independent, but this is obviously not the case with the parts' scale and location. However, once

[2]We thank Dashan Gao for making his code available to us, and providing useful feedback.

we align the object instances with respect to location and scale, the assumption of part location and scale independence becomes reasonable. Thus we introduce a 3-dimensional hidden variable $C = (C_l, C_s)$, which fixes the location of the object and its scale. Our assumption is that the location and scale of different parts is conditionally independent given the hidden variable C, and so the joint distribution decomposes according to the graph in Figure 3b.

It follows that for a model with P parts, the joint probability of $\{X^k\}_{k=1}^p$ and C takes the form

$$\mathrm{p}(\{\mathrm{X^k}\}_{\mathrm{k=1}}^{\mathrm{P}}, \mathrm{C}|\Theta) = p(C|\Theta)\prod_{k=1}^{P} p(X^k|C,\theta^k) = p(C|\Theta)\prod_{k=1}^{P} \mathrm{p}(\mathrm{X_a^k}|\theta_a^k)p(X_l^k|C_l,C_s,\theta_l^k)p(X_s^k|C_s,\theta_s^k) \quad (5)$$

We assume uniform probability for C and Gaussian conditional distribution for X_a, X_l, X_s as follows:

$$\begin{aligned} P(X_a^k|\theta_a^k) &= G(X_a^k|\mu_a^k,\Sigma_a^k) \quad (6)\\ P(X_l^k|C_l,C_s,\theta_l^k) &= G(\frac{X_l^k - C_l}{C_s}|\mu_l^k,\Sigma_l^k)\\ P(X_s^k|C_s,\theta_s^k) &= G(\log(X_s^k) - \log(C_s)|\mu_s^k,\sigma_s^k) \end{aligned}$$

where $G(\cdot|\mu,\Sigma)$ denotes the Gaussian density with mean μ and covariance matrix Σ. We index the model components a, l, s as $1, 2, 3$ respectively, and denote the log of these probabilities by $LG(x_j|C,\mu_j,\Sigma_j)$ for $j = 1, 2, 3$.

3.3 A model based classifier

As discussed in Section 2.4.1, the likelihood $P(I|M)$ is given by averaging over all the possible part vectors that can be assembled from the feature set $F(I)$ (see Eq. (2)). In our case, we should also average over all the possible values for the hidden variable C. Thus

$$P(I|M) = K_0 \sum_C \sum_{\substack{(x^1,..,x^p)\in F(I)^P \\ x^i \neq x^j \, for i \neq j}} \prod_{k=1}^{P} P(x^k|C,\theta^k) \quad (7)$$

for some constant K_0.

In order to allow efficient likelihood assessment we make the following approximations

$$\begin{aligned} P(I|M) &\approx K_0 \sum_C \sum_{(x^1,..,x^p)\in F(I)^P} \prod_{k=1}^{P} P(x^k|C,\theta^k) \quad (8)\\ &\approx K_0 \max_{C,(x^1,..,x^p)\in F(I)^P} \prod_{k=1}^{P} P(x^k|C,\theta^k) \quad (9)\\ &= K_0 \max_C \prod_{K=1}^{P} \max_{x\in F(I)} P(x|C,\theta^k) \quad (10) \end{aligned}$$

Approximation (8) above was discussed earlier in a more general context (see Eq. (4)), and it is necessary in order to eliminate the global dependency between parts. In approximation (9), averages are replaced by the likelihood of the best feature vector and best hidden C. This approximation is compelling since natural images rarely have two different likely object interpretations. In addition, working with the best single vector uniquely identifies the object's location and scale, as well as the object's parts. Such unique identification is required for most semantical tasks beyond mere recognition. Finally, the approximated likelihood is decomposed into separate maxima over C and the different parts in Eq. (10).

The decomposition of the maximum achieved in Eq. (10) is the key to the efficient likelihood computation. We discretize the hidden variable C and consider only a finite grid of locations and scales, with a total of N_c possible values. Using this decomposition the maximum over the $N_c \cdot N_f^P$ arguments can be computed in $O(N_c N_f P)$ operations. However, we cannot optimizing the parameters of such a model by likelihood maximization. Since feature repetition is allowed, the ML solution will choose the same (best) part p times, as shown in Appendix A.

The natural generative classifier is based on the comparison of the LRT statistic to a constant threshold, and it therefore requires a model of the background in addition to the object model. Modeling general backgrounds is clearly difficult, due to the diversity of objects and scenes that do not share simple common features. We therefore approximate the background likelihood by a constant. Our LRT based classifier thus becomes

$$f(I) = \log P(I|M) - \log(I|BG) - \nu^{'} = \max_{C} \sum_{k=1}^{P} \max_{x \in F(I)} \log\, p(x|C, \theta^k) - \nu \tag{11}$$

for some constant ν.

4 Discriminative optimization

Given a set of labeled images $\{I_i, y_i\}_{i=1}^N$, we wish to find a classifier $f(I)$ of the functional form given in Eq. (11), which minimizes the exponential loss

$$L(f) = \sum_{i=1}^{N} \exp(-y_i f(I_i)) \tag{12}$$

This is the same loss minimized by the the Adaboost algorithm [22]. Its main advantage in our context is that it allows for the determination of the classifier threshold using a closed form formula, as will be described in Section 4.1.

We have considered two possible techniques for the optimization of the loss in Eq. (12): Boosting and gradient descent. In the boosting context, we view the log probability of a part

$$\max_{x \in F(I)} \log\; p(x|C, \theta^k)$$

as a weak hypothesis of a specific functional form. However, the classifier form we use in (Eq. (11)) is rather different from the traditional classifiers built by boosting, which typically have the form $f(I) = \sum_{k=1}^{P} \alpha^k h^k(I)$. Specifically, the classifier (11) does not include part weights α^k, it has an extra threshold parameter ν, and it involves a maximization over C, which depends on all the 'weak' hypotheses. The third point is the most problematic, as it requires optimizing over parts with internal dependencies, which is much harder than optimization over independent parts as in standard boosting.

In order to simplify the presentation, we assume in Section 4.1 a simplified model with no spatial relations between the parts, and show how the problems of parts weights and threshold parameters are coped with, with minor changes to the standard boosting framework. In Section 4.2 we consider the problem of dependent parts, and show how boosting can be naturally extended to handle classifiers as in Eq. (11), despite the dependencies between parts due to the hidden variable C. Finally we consider the optimization from a more general viewpoint of gradient descent in Section 4.3. This allows us to introduce several enhancements to the pure boosting technique.

4.1 Boosting of a probabilistic model

Let us consider a simplified model with parts appearance only (see Eq. (6)). We show how such a classifier can be represented as a sum of weighted 'weak' hypotheses in Section 4.1.1. We then derive the boosting

algorithm as an approximate gradient descent in Section 4.1.2. This derivation is slightly simpler than similar derivations in the literature, and provides the basis for our treatment of related parts, introduced in Section 4.2. In Section 4.1.3 we show how the threshold parameter in our classifier can be readily optimized.

4.1.1 Functional form of the classifier

When there are no relations between parts, the classifier (11) takes the following form

$$f(I) = \sum_{k=1}^{P} \max_{x \in F(I)} \log\, p(x_a|\theta_a^k) - \nu \tag{13}$$

This classifier is easily represented as a sum of weak hypotheses $f(I) = \sum_{k=1}^{P} h^k(I)$ where

$$h^k(I) = \max_{x \in F(I)} \log G(x_a|\theta_a^k) - \nu^k \tag{14}$$

and $\nu = \sum_{k=1}^{P} \nu^k$. Weak hypotheses in this form can be viewed as soft classifiers.

We next represent the classifier in an equivalent functional form in which the covariance scale is transformed to part weight. Now $f(I) = \sum_{k=1}^{P} \alpha^k h^k(I)$ where $h^k(I)$ takes the form

$$h^k(I) = \max_{x \in F(I)} \log G(x_a|\eta_a^k, \Sigma_a^k) - \nu^k, \qquad |\Sigma_a^k| = 1 \tag{15}$$

The equivalence of these forms is shown in Appendix B.

4.1.2 Boosting as approximate gradient descent

Boosting is a common method which learns a classifier of the form $f(x) = \sum_{k=1}^{p} \alpha^k h^k(x)$ in a greedy fashion. Several papers [12, 15] have presented boosting as a greedy gradient descent of some loss function. In particular, the work of [15] has shown that the Adaboost algorithm [29, 22] can be viewed as a greedy gradient descent of the exp loss in Eq. (12), in L^2 function space. In [12] Adaboost is derived using a second order Taylor approximation of the exp loss, which leads to repetitive least square regression problems. We suggest here another variation of the derivation, similar to [12] but slightly simpler. All three approaches lead to an identical algorithm (the discrete Adaboost [29]) when the weak learners are binary with the range $\{+1, -1\}$. For weak learners with a continuous output, our approach and the approach of [15] culminates in the same algorithm, e.g. Adaboost with confidence intervals [22]. However, our approach is simpler, and is later used to derive a boosting version for a model with dependent parts.

Specifically, we derive Adaboost by considering the first order Taylor expansion of the exp loss function. In what follows and throughout this paper, we use superscripts to indicate the boosting round in which a quantity is measured. At the p'th boosting round, we wish to extend the classifier f by $f^p(x) = f^{p-1}(x) + \alpha^p h^p(x)$. We first assume that α^p is infinitesimally small, and look for an appropriate weak hypothesis $h^p(X)$. Since α^p is small, we can approximate Eq. (12) using the first order Taylor expansion.

To begin with, we differentiate $L(f)$ w.r.t. α^p

$$\frac{dL(f)}{d\alpha^p} = -\sum_{i=1}^{N} \exp(-y_i f(x_i)) y_i h^p(x_i) \tag{16}$$

We denote $w_i = \exp(-y_i f(x_i))$, and derive the following Taylor expansion

$$L(f^p) \approx L(f^{p-1}) - \alpha^p \sum_{i=1}^{N} w_i^{p-1} y_i h^p(x_i) \tag{17}$$

Assuming $\alpha^p > 0$, the steepest descent of $L(f)$ is obtained for some weak hypothesis h^p which maximizes the weighted correlation score

$$S(h^p(x)) = \sum_{i=1}^{N} w_i^{p-1} y_i h^p(x_i) \tag{18}$$

This maximization is done by a weak learner, getting as input the weights $\{w_i^{p-1}\}_{i=1}^N$ and the labeled data points. After the determination of $h^p(x)$, the coefficient α^p is determined by the direct optimization of the loss in Eq. (12). This can be done in closed form only for binary weak hypotheses with output in the range of $\{1, -1\}$. In the general case numeric methods are employed, such as line search [22].

4.1.3 Threshold optimization

Maximizing the linear approximation (17) can be problematic when unbounded weak hypotheses are used. In particular, optimizing this criterion w.r.t to the threshold parameter in hypotheses of the form (14) is ill-posed. Substituting (14) into criterion (17), we get the following expression to optimize:

$$\begin{aligned} S(h) &= \sum_{i=1}^{N} w_i y_i (\max_{x \in F(I)} \log G(x_i|\mu_a, \Sigma_a) - \nu) \\ &= C + (\sum_{i:y_i=-1} w_i - \sum_{i:y_i=1} w_i)\nu \end{aligned} \tag{19}$$

where C is independent of ν. If $\sum_{i:y_i=-1} w_i - \sum_{i:y_i=1} w_i \neq 0$, S(h) can be increased indefinitely by sending ν to $+\infty$ or $-\infty$. Such a choice of ν clearly doesn't improve the original (exact) loss (12).

The optimization of the threshold should therefore be done by considering (12) directly. It is based on the following lemma:

Lemma 1. *Consider a function $f : I \rightarrow R$. We wish to minimize the loss (12) of the function $\tilde{f} = f - \nu$ where ν is a constant. Assume that there are both labels $+1$ and -1 in the data set.*

1. *An optimal ν^* exists and is given by*

$$\nu^* = \frac{1}{2} \log \left[\frac{\sum_{\{i:y_i=-1\}}^{N} \exp(f(I_i))}{\sum_{\{i:y_i=1\}}^{N} \exp(-f(I_i))} \right] \tag{20}$$

2. *The optimal $\tilde{f}^* = f - \nu^*$ satisfies*

$$\sum_{\{i:y_i=1\}}^{N} \exp(-\tilde{f}^*(I_i)) = \sum_{\{i:y_i=-1\}}^{N} \exp(\tilde{f}^*(I_i)) \tag{21}$$

3. *The optimal loss $L(f - \nu^*)$ is*

$$2 \left[\sum_{\{i:y_i=1\}}^{N} \exp(-f(I_i)) \cdot \sum_{\{i:y_i=-1\}}^{N} \exp(f(I_i)) \right]^{\frac{1}{2}} \tag{22}$$

The lemma is proved by direct differentiation of the loss w.r.t ν, as sketched in Appendix C.

We use this lemma to determine the threshold after each round of boosting, when $f^p(I) = f^{p-1}(I) + \alpha^p h^p(I)$. Eq. (20) gives a closed form solution for ν once $h^p(I)$ and α^p have been chosen. Eq. (22) gives the optimal score obtained, and it is useful when efficient numeric search for α^p is required. Finally, property (21) implies that after threshold update, the coefficient of ν in Eq. (19) is nullified (the slope of the linear approximation is 0 at the optimum). Hence optimizing the threshold before round p assures that the score $S(h^p)$ does not depend on ν^p. We optimize the threshold in our algorithm during initialization, and after every boosting round (see Algorithm 1). The weak learner can therefore effectively ignore this parameter when choosing a candidate hypothesis.

4.2 Relational model Boosting

We now extend the boosting framework to handle dependent parts in a relational model of the form (11). We introduce part weights into the classifier by applying the transformation described in Eq. (15) to the three model ingredient described in Eq. (6), i.e. appearance, location and scale. The three new weights are summed into a single part weight, leading to the following classifier form

$$f(I) = \max_C \sum_{k=1}^{P} \alpha^k h^k(I, C) - \nu \tag{23}$$

where for $k = 1, .., P$

$$\begin{aligned} h^k(I, C) &= \max_{x \in F(I)} g^k(I, C) \\ g^k(I, C) &= \sum_{j=1}^{3} \frac{\lambda_j^k}{\sum_{j=1}^{3} \lambda_j^k} LG(x_j^k | c, \mu_j^k, \Sigma_j^k) \\ |\Sigma_j^k| &= 1 \ , \ \lambda_j^k > 0 \qquad j = 1, 2, 3 \end{aligned} \tag{24}$$

In this parametrization α^k is the sum of component weights and $\lambda_i / \sum_{j=1}^{3} \lambda_j$ measures the relative weights of the appearance, location and scale. Thus, given an image I, the computation of f requires the computation of the accumulated log-likelihood and its hidden center optimizer, denoted as follows

$$\begin{aligned} ll(I, C) &= \sum_{k=1}^{p} \alpha^k h^k(I, C) \\ C^* &= \arg\max_C ll(I, C) \end{aligned} \tag{25}$$

In order to allow tractable maximization over C, we discretize it and consider only a finite grid of locations and scales with N_c possible values. Under these conditions, the computation of ll and C^* amounts to standard MAP message passing, requiring $O(PN_f N_c)$ operations.

Our suggested boosting method is presented in Algorithm 1. We derive it by replicating the derivation of standard boosting in Eq. (16)-(18). For f of the form (23), the derivative of $L(f)$ w.r.t. α_p is now

$$\frac{dL(f)}{d\alpha^p} = -\sum_{i=1}^{N} w_i y_i h^p(I_i, C_i^*) \tag{26}$$

and using the Taylor expansion (17) we get

$$L(f^p) = L(f^{p-1}) - \alpha^p \sum_{i=1}^{N} w_i^{p-1} y_i h^P(I_i, C_i^{*,p-1}) \tag{27}$$

Algorithm 1 Relational model boosting

Given $\{(I_i, y_i)\}_{i=1}^N$, $y_i \in \{-1, 1\}$, initialize:

$ll(i, c) = 0 \qquad i = 1, .., N$, c in a predefined grid
$\nu = \frac{1}{2} log \frac{\#\{y_i=-1\}}{\#\{y_i=1\}}$
$w_i = exp(\ y_i \cdot \nu\ ) \qquad i = 1, .., N$
$w_i = w_i / \sum_{i=1}^N w_i$
For $k = 1, .., P$

1. Use a weak learner to find a part hypothesis $h^k(I, C)$ which maximizes

$$S(h) = \sum_{i=1}^{N} w_i y_i h(I_i, C_i^*)$$

   (see text for special treatment of round 1).

2. Find optimal α^k by minimizing

$$\sum_{\{i:y_i=1\}}^{N} \exp(-f^0(I_i)) \cdot \sum_{\{i:y_i=-1\}}^{N} \exp(f^0(I_i))$$

   where $f^0(I) = \max_C ll(I, C) + \alpha h^k(I, C))$.

   Update ll and the optimal center C^*

   $ll(i, c) = ll(i, c) + \alpha^k h(i, c)$

   $[f^0(I_i), C_i^*] = \max, \arg\max_c ll(i, c)$

3. Update $f(I_i)$ and the weights $\{w_i\}_{i=1}^N$

   $\nu = \frac{1}{2} \log \left[\frac{\sum_{\{i:y_i=-1\}}^{N} \exp(f^0(I_i))}{\sum_{\{i:y_i=1\}}^{N} \exp(-f^0(I_i))} \right]$

   $f(I_i) = f^0(I_i) - \nu$

   $w_i = exp(-y_i f(I_i))$

   $w_i = w_i / \sum_{i=1}^N w_i$

Output the final hypothesis $f(I) = \max_C \sum_{k=1}^P \alpha_k h_k(I, C) - \nu$

In analogy with the criterion (18), the weak learner should now get as input $\{w_i^{p-1}, C_i^{*,p-1}\}_{i=1}^N$ and try to maximize the score

$$S(h^p) = \sum_{i=1}^{N} w_i^{p-1} y_i h^P(I_i, C_i^{*,p-1}) \tag{28}$$

This task is not essentially harder than the weak learner's task in standard boosting, since the weak learner 'assumes' that the value of the hidden variable C is known and set to its optimal value according to the previous hypotheses. In the first boosting round, when $C^{*,p-1}$ is not yet defined, we only train the appearance component of the hypothesis. The relational components of this part are set to have low weights and default values.

Choosing α^p after the hypothesis $h^p(I,C)$ has been chosen is more demanding than in standard boosting. Specifically, α^p should be chosen to minimize

$$L(\max_C \left[ll^{p-1}(I,C) + \alpha^p h^p(I,C)\right] - \nu^*) \tag{29}$$

Since the optimal value of C depends on α^p, its inference should be repeated whenever a different value is considered for α^p (although the messages $h^p(I,C)$ can be computed only once). After finding the maximum over C, the loss with the optimal threshold can be computed using Eq. (22). The search for the optimal α^p can be done using any line search algorithm, and we implement it using gradient descent as described in Section 4.3.

4.3 Gradient descent

In this section we combine the relational boosting from Section 4.2 with elements from a more general gradient descent perspective. In Section 4.3.1 we describe our implementation of Algorithm 1, in which the weak learner and the part weight optimization are gradient based. In Section 4.3.2 we suggest to supplement Algorithm 1 with feedback elements in the spirit of more traditional gradient descent algorithms. Algorithm 2 presents the resulting algorithm for part optimization.

4.3.1 Gradient-based implementation

Current boosting-based object recognition approaches use a version of what we call "selection-based" weak learners [2, 19, 14]. The weak hypotheses family is finite, and hypotheses are based on a predefined feature set [19] or on the set of features extracted from the training images [2, 14]. The weak learner computes the weighted correlation for all the possible hypotheses and returns the best scoring one. Weak learners of this type, considered in the current paper, sample features from object images (exhaustive search is too expensive computationally); they build part hypotheses based on the feature and the current estimate of the hidden center C^* in the feature's image. However, as a single feature cannot reliably determine the relative weights of the different part components (the covariance scale of appearance, location and scale), several values of these parameters are considered for each feature.

As an alternative, we have considered a second type of weak learners, which we call "gradient-based". A "gradient-based" weak learner uses a hypothesis supplied by the selection learner as its starting point, and tries to improve its score using gradient ascent. Unlike the selection based weak learner, the gradient-based weak learner is not limited to parts based on natural image features, as it searches in the continuum of all possible part models. The relevant gradient is the derivative of the score $S(h^p)$ w.r.t the part parameters, given by the weighted sum

$$\frac{dS(h^p)}{d\theta^p} = \sum_{i=1}^{N} w_i^{p-1} y_i \frac{dh^p(I_i, C_i^{*,p-1})}{d\theta^p} = \sum_{i=1}^{N} w_i^{p-1} y_i \frac{dg^p(I_i, C_i^{*,p-1}, x_i^{*,p})}{d\theta^p} \tag{30}$$

where $x_i^{*,p}$ is the best part candidate in image i

$$x_i^{*,p} = \arg\max_{x \in F(I_i)} g^p(I_i, C_i^{*,p-1})$$

Since the gradient depends on the best part candidates according to the current model, the gradient dynamics iterates between gradient steps in the parameters θ^p and the re-computation of the best part candidates $\{x_i^{*,p}\}_{i=1}^N$. Pseudo code is given in Step 1 of Algorithm 2.

We also use gradient descent dynamics to implement the line search for the optimal part weight α^p. This search method is based on slow, gradual changes in the value of α^p, and hence it allows us to experiment with a feedback mechanism (see Section 4.3.2). The gradient of the loss w.r.t α^p is given in Eq. (26). Notice that the gradient depends on $\{C_i^*\}_{i=1}^N$ and $\{w_i\}_{i=1}^N$, and both are functions of α^p. Hence the gradient dynamics

in this case iterates between gradient steps of α^p, inference of $\{C_i^*\}_{i=1}^N$, and updates of $\{w_i\}_{i=1}^N$. This loop is instantiated in Step 3 of Algorithm 2. The loop must be preceded by the computation of the messages $h(i,c)$ in Step 2.

4.3.2 Gradient-based extension

When the determination of both θ^p and α^p are gradient based, the boosting optimization at round p essentially makes a specific control choice for a unified gradient descent algorithm which optimizes α^p and θ^p together. A more traditional gradient descent algorithm can be constructed by 1) differentiating $L(f)$ directly instead of using its Taylor approximation, and 2) iterating small gradient steps on both α^p and θ^p in a single loop, instead of two separate loops as suggested by boosting. In boosting, the optimization of θ^p is done before setting α^p and there is no feedback between them. Such feedback is plausible in our case, since any change in α^p may induce changes in C^* for some images, and can therefore change the optimal part model of $h^p(I,C)$.

We considered the update steps required for gradient descent of the exact loss (12), without the Taylor approximation implied by the boosting strategy. The gradient of α^p (Eq. (26)) and its treatment remain the same, as α^p was optimized w.r.t the exact loss in the boosting strategy as well. The gradient w.r.t θ^p is

$$\frac{dL(f)}{d\theta^p} = \sum_{i=1}^{N} w_i^p y_i \frac{dh^p(I_i, C_i^{*,p})}{d\theta^p} \tag{31}$$

While this expression is very similar to (30), there is a subtle difference between them. In Eq. (31) w_i and C_i^* are no longer constant as they were in (30), but depend on θ^p and α^p. Exact gradient descent therefore requires the re-computation of w_i, C_i^* at each gradient iteration, which is computationally expensive.

We have experimented in the continuum between the 'boosting' and the 'gradient descent' flavors using Algorithm 2, which encloses the optimization loops of h^p and α^p in a third 'feedback' loop. Setting the outer loop counter K_1 to 1 we get the boosting flavor, i.e., Algorithm 2 implements an inner loop step in Algorithm 1. Setting K_1 to some large value and $K_2 = 1$,$K_3 = 1$, we get exact gradient descent flavor. We found that a good trade-off between complexity and performance is achieved with a version which is rather close to boosting, but still repeats the optimization of α^p and h^p several times to allow mutual cross-talk during the estimation of these parameters. Thus, our final optimization algorithm involves repeated, sequential calls of Algorithm 2.

5 Experimental results

We tested our algorithm in recognition tasks using the Caltech datasets [20], which are publicly available[3], as well as three more challenging data sets we have collected specifically for this evaluation. The former are used as a common benchmark, while the latter are designed to measure the performance limits of the algorithm by challenging it with fairly hard conditions. Localization performance was evaluated using a common benchmark for this task [23][4]. The Datasets are described in Section 5.1. In Section 5.2 we discuss the various algorithm parameters. Recognition results are presented in Section 5.3. In Section 5.4 we report the results of additional experiments, studying the contribution to recognition performance of several modeling factors in isolation. Finally, we report localization results in Section 5.5.

5.1 Datasets

We compare our recognition results with other methods using the Caltech datasets. Four datasets are used: Motorcycles (800 images), Cars rear (800), Airplanes (800) and Faces (435). These datasets contain relatively small variance in scale and location, and the background images do not contain objects similar to the class

[3] http://www.robots.ox.ac.uk/∼vgg/data

[4] http://www.pascal-network.org/challenges/VOC/#UIUC

Algorithm 2 Optimization of part p

Input : $F(I_i)$, y_i , w_i, C_i^* $\quad i = 1,..,N$
$\quad\quad ll(i,c) \quad i = 1,..,N$, $c = 1,..,N_c$
initialize weak hypothesis using a selection learner :
Choose $\theta = \lambda_j, \mu_j, \Sigma_j \quad j = 1,..,3 \quad , \quad \alpha = 0$
Set $[h(i, C_i^*), x^*(i)] = \max, \arg\max_{x \in F(I_i)} g(x, C_i^*)$
where $g(x,c) = \sum_{j=1}^{3} \frac{\lambda_j}{\sum_{j=1}^{3} \lambda_j} LG(x_j | c, \mu_j, \Sigma_j)$

Loop over $1,2,3$ K_1 iterations

1. Loop over a,b K_2 iterations (θ optimization)
   (a) Update weak hypothesis parameters
   $\theta = \theta + \eta \sum_{i=1}^{N} w_i y_i \frac{dg(x_i^*, c_i^*)}{d\theta}$
   (b) Update best part candidates for all images
   $[h(i, C_i^*), x_i^*] = \max, \arg\max_{x \in F(I_i)} g(x, C_i^*)$
2. Compute for all $i, c \quad h(i,c) = \max_{x \in F(I_i)} g(x,c)$
3. Loop over a,b,c K_3 iterations (α optimization)
   (a) Update α : $\quad \alpha = \alpha + \eta \sum_{i=1}^{N} w_i y_i h(i, C_i^*)$
   (b) Update hidden center for all images
   $[f^0(I_i), C_i^*] = \max, \arg\max_{c} ll(i,c) + \alpha h(i,c)$
   (c) Update $f(I_i)$ and the weights
   $\nu = \frac{1}{2} \log \left[\frac{\sum_{\{i: y_i = -1\}}^{N} \exp(f^0(I_i))}{\sum_{\{i: y_i = 1\}}^{N} \exp(-f^0(I_i))} \right]$
   $f(I_i) = f^0(I_i) - \nu$
   $w_i = exp(-y_i f(I_i)) \ , \quad w_i = w_i / \sum_{i=1}^{N} w_i$

Set $ll^p(i,c) = ll(i,c) + \alpha h(i,c)$
Return θ, w_i, C_i^*,$ll^p(i,c) \quad i = 1,..N \quad c = 1,..,N_c$

objects. In order to test recognition performance under harder conditions, we compiled 3 new datasets with matching backgrounds.[5] These datasets contain images of Chairs (800 images), Dogs (500) and Humans (593), and are briefly described below. Our localization experiments were carried using the UIUC cars side data set[23]. The training set here is composed of 550 cars images and 500 background images. The test set includes 170 images, containing altogether 200 cars, with ground truth bounding boxes.

In the Chairs and Dogs datasets, the objects are seen roughly from the same pose, but include large inner class variability, as well as some variability in location and scale. For the Chairs dataset we compiled a background dataset of Furniture which contained images of tables, beds and bookcases (200,200,100 images respectively). When possible (for tables and beds), images were aligned to a viewpoint isomorphic to the viewpoint of the chairs. As background for the Dogs dataset, we compiled two animal datasets: 'Easy

[5]The datasets are available at `http://www.cs.huji.ac.il/~aharonbh/`.

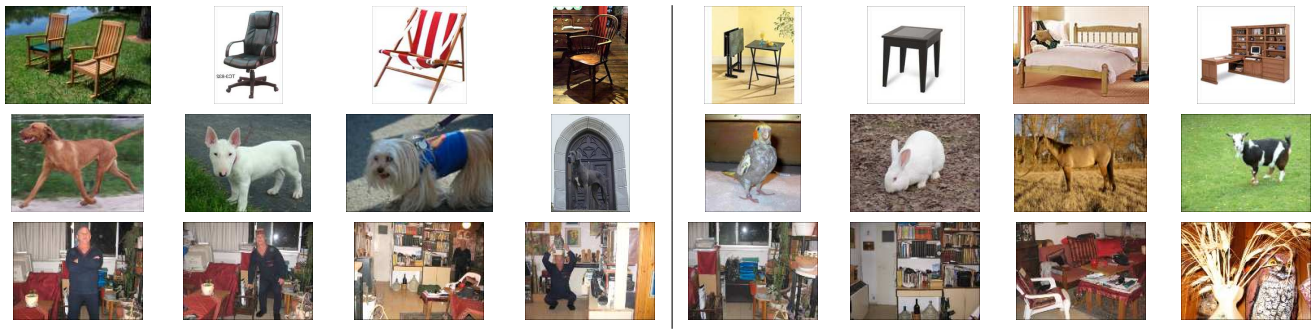


Figure 4: Images from the Chairs, Dogs and Humans datasets and their corresponding backgrounds. Object images appear on the left, background images on the right. In the second row, the two leftmost background images are of 'easy animals' and next are two 'hard animals' images. In the third row, the two leftmost object images belong to the easier image subset. The next two images are hard due to the person's scale and pose.

Animals' contains 500 images of animals not similar to Dogs; 'Hard Animals' contains 250 images from the 'Easy Animals' dataset, and an additional set of 250 images of four-legged animals (such as horses and goats) in a pose isomorphic to the Dogs.

The Humans dataset was designed to include large variability in location, scale and pose. The data contains images of 25 different people. Each person was photographed in 4 different scales (each 1.5 times larger than its predecessor), at various locations and with several articulated poses of the hands and legs. For each person there are several images in which s/he is partially occluded. For this dataset we created a background dataset of 593 images, showing the sites in which the Humans images were taken. Figure 4 shows several images from our datasets.

5.2 Algorithm parameters

We have run a series of preliminary experiments, in order to tune the weak learners' parameters and compare the results when using selection-based vs. gradient-based weak learners. The parameters of the selection based weak learner include the number of image patches it samples, and the number of location/scale models used for each sampled patch. The parameters of the gradient based learner include the step size and stop condition. The gradient based learner is not limited to hypotheses based on object images, and in many cases it chooses exaggerated appearance and location models for the part in order to enhance discriminative power. In the exaggerated appearance models, the brightness contrast is enhanced and the mean patch looks almost like a Black&White mask (see examples in Figure 5b). This tendency for exaggerated appearance model is enhanced when the weight of the location model is relatively weak.

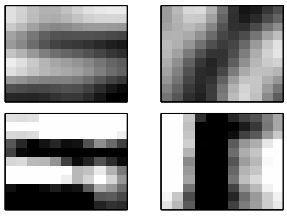
In exaggerated location models, parts are modeled as being much farther from the center than they are in real objects. An example is given in Figure 7, showing a chair model where the tip of the chair's leg is located below its mean location in most images. Still, in most cases gradient based learners give lower error rates than their purely selection-based competitors. Some examples are given in Table 5a. We hence used gradient based learners in the rest of the recognition experiments.

We have also experimented with the learning of covariance matrices for the appearance and location models. To guarantee positive definiteness, we have implemented gradient dynamics for the square root of the covariance matrix. However, we have still observed too much over-fitting in the estimation of the covariance matrices in our experiments. These additional degrees of freedom tended not to improve the test results, while achieving lower training error. The problem was ever more serious with the appearance covariance matrix, where we have sometimes observed reduced performance, and the emergence of unstable models with covariance matrices close to singular. As a result, in the following experiments we fix the covariance matrices to σI. We only learn the covariance scale, which in our model determines the part and component weight parameters.

In the recognition experiments reported in Section 5.3, we constructed models with up to 60 parts using

Data Name	Selection Learner	Gradient Learner
Motorbikes	7.2	6.9
Cars Rear	6.8	2.3
Airplanes	14.2	10.3
Faces	7.9	8.35

a)



b)

Figure 5: **a)** Comparison of error rates obtained by selection-based and gradient based weak learners on the Caltech data sets. The results presented were obtained for object models without a location component, i.e. the models are not relational and classification is based on part appearance alone. **b)** Examples of parts from motorcycle models learnt using the selection-based learner (top) and the gradient-based learner (bottom). The images present reconstructions from the 15 DCT coefficients of the mean appearance vector. The parts presented correspond to motorcycles seat (left) and wheel (right). Clearly, the parts learnt by the gradient learner have much sharper contrasts.

Algorithm 2, with control parameters of $K_1 = 60, K_2 = 100, K_3 = 4$. Each image was represented using at most $N_f = 200$ features (KB detector) or $N_f = 240$ features (GV detector). The hidden center location values were an equally spaced grid of 6×6 positions over the image. The hidden scale center had a single value, or 3 different values with a ratio of 0.63 between successive scales, resulting in a total of $N_c = 36, 108$ values respectively. We randomly selected half of the images from each dataset for training and used the remaining half for testing.

For the localization experiments reported in Section 5.5 we changed several important parameters of the learning process. Model accuracy is more important for this task, and we therefore learn smaller models with $P = 40$ parts, but using a finer location grid of 10×10 possible locations ($N_C = 100$) and $N_f = 400$ features extracted per image. As noted above, the dynamics of the gradient-based location model tends to produce 'exaggerated' models, in which parts are located too far from the objects center. This tendency dramatically reduces the utility of the model for localization. We therefore eliminated the gradient location dynamics in this context, and modified only the part appearance using gradient descent. We found experimentally that increasing the weight of the location component uniformly for all the parts improves the localization results considerably. In the experiments reported below, we multiply the location component weights $\Lambda_2^k \;\; k = 1, .., P$ (see Eq. (**??** for their definition) by a constant factor of 10. Probabilistically, this amounts to smaller location covariance and hence to stricter demands on the accuracy of parts relative locations. Finally, parts without location component (when the location component weight is 0) are ignore; these parts do not convey localization information, and therefore add irrelevant 'noise' to the MAP score.

5.3 Recognition results

As a general remark, we note that our algorithm tends to learn models in which most features have clear semantics in terms of object's parts. Examples of learnt models can be seen in Figures 6-7. In the dog example we can clearly identify parts that correspond to the head, back, legs (both front and back), and the hip. Typically $40 - 50$ out of the 60 parts are similar in quality to the ones shown. The location models are gross, and sometimes exaggerated, but clearly useful. Analysis of the part models shows that in many cases, a distinguished object part (e.g a wheel in the motorcycle model, or an eye in the face model) is modeled using a number of model parts (12 for the wheel, 10 for the eye) with certain internal variation. In this sense our model seems to describe each object part using a mixture model.

In Table 1 we compare our results to those obtained by a purely generative approach [20][6] and a purely discriminative one [2][7] using the Caltech dataset. Both methods learn from an unordered set of local descriptors, obtained using an interest point detector. Following [20], the motorbikes, airplanes and faces

[6]Note that the results reported in [20] (except for the cars data base) were achieved using manually scale-normalized images, while our method did not rely on any such rescaling.

[7]In [1], this approach was reported to give better results using segmentation based features. We did not include these results since we wanted to compare the different learning algorithms using similar features. see also discussion in Section 6

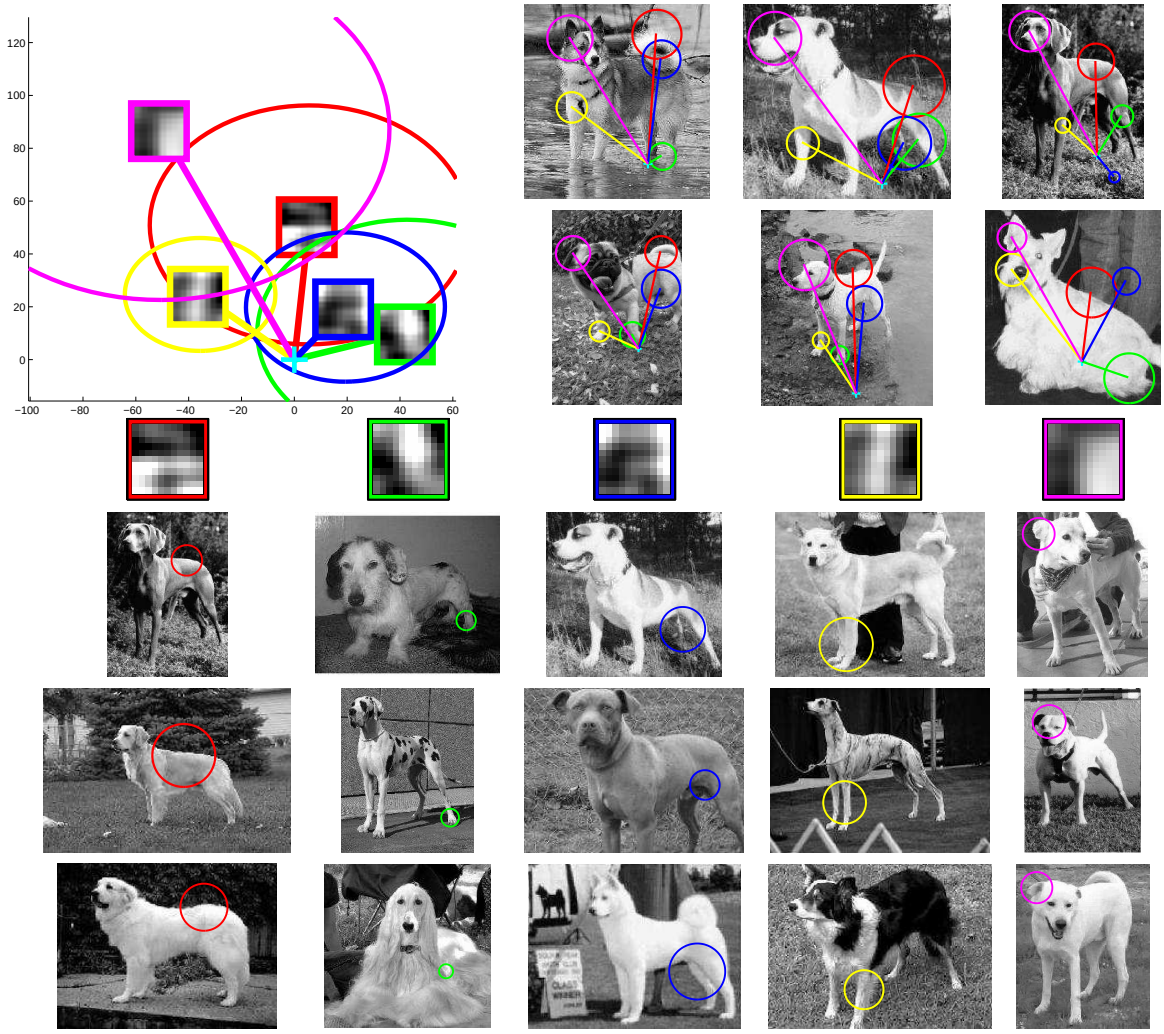


Figure 6: 5 parts from a dog model with 60 parts. The top left drawing shows the modeled locations of the 5 parts. Each part's mean location is surrounded by the 1 std line. The cyan cross indicates the location of the hidden 'center'. The top right pictures show dog test images with the model implementation found. All these dogs were successfully identified except for the one on the right-bottom corner. Below the location model, the parts' mean appearance patches are shown. The last three rows present parts implementations in the 3 test images that got the highest part likelihood. Each column presents the implementations of the part shown above the column. The parts have clear semantic meaning and repetitive locations on the dogs back, hind leg, joint of the hind leg, front leg and head. Most other parts behave similarly to the ones shown.

datasets were tested against office background images, and the Cars rear dataset was tested against road background images. To allow for a clear comparison with [20], we used their exact train and test indexes and the same feature detector (KB).

Our results are given in Table 1. They were obtained without modeling scale, since it did not improve classification results when using the KB detector. This may be partially explained by noting that the Caltech datasets contain relatively small variance in scale. Error rates for our method were computed using the threshold learnt by our boosting algorithm. Results are presented for models with 7 parts (the number of parts used in [20]) and 50 parts. When 7 parts are used, our results are comparable to those of [20]. However, when 50 parts are used our algorithm outperforms both competitors in most cases.

We used the Chairs and Dogs datasets to test the sensitivity of the algorithm to visual similarity between object and background images. We trained the Chairs dataset against the Caltech office background

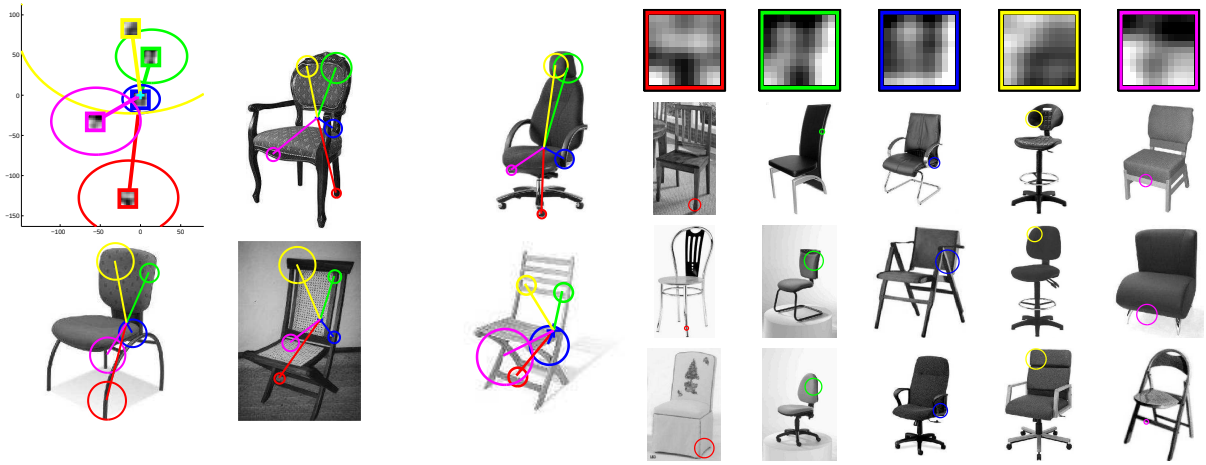


Figure 7: 5 parts from a chair model. The model is presented in the same format as Figure 6. Model parts represent the tip of the chairs leg(first part), edges of the back (second and forth parts), the seat corner(third) and the seat edge(fifth). The location model is exaggerated: The tip of the chairs leg is modeled as being far below its real mean position in object images.

Data Name	**Our model 7 parts**	**Our model 50 parts**	**Fergus et. al**	**Opelt et. al**
Motorbikes	7.8	4.9	7.5	7.8
Cars Rear	1.2	0.6	9.7	8.9
Airplanes	8.6	6.7	9.8	11.1
Faces	9.5	6.3	3.6	6.5

Table 1: Test error rates over the Caltech dataset, showing the results of our method in 2 conditions - using 7 or 50 parts, as well as two other methods - a generative model approach [20] and a discriminative model-free boosting approach [2]. The algorithm's parameters were held constant across all experiments.

dataset, and against the furniture dataset described above. The Dogs dataset was trained against 3 different backgrounds datasets: Caltechs 'office' background, 'Easy Animals' and 'Hard Animals'. The results are summarized in Table 2. As can be seen, our algorithm works well when there are large differences between the object and background images. However, it fails to discriminate, for example, dogs from horses.

Data	**Background**	**Test Error**
Chairs	Office	2.23
Chairs	Furniture	15.53
Dogs	Office	8.61
Dogs	Easy Animals	19.0
Dogs	Hard Animals	34.4
Humans	Sites	34.3
Humans (resticted)	Sites	25.9

Table 2: Error rates with the new datasets of Chairs, Dogs and Humans. Results were obtained using the KB detector (see text for more details).

We used the Humans dataset to test the algorithm's sensitivity to variations in scale and object articulations. In order to obtain reasonable results on this hard dataset, we had to reduce scale variability to 2 scales and restrict the variability in pose to hand gestures only - we denote this dataset by 'Humans restricted' (355 images). The results are shown in Table 2.

The parameters in our model are optimized to minimize training error with respect to a certain background. One may worry that the learnt models describe the background just as well as they describe the object, in which case performance in classification tasks against different backgrounds is expected to be poor.

Data	Original BG		Motorcycles BG	Airplanes BG	Sites BG
Cars	Road	(0.6)	3.0	2.2	6.8
Cars	Office	(1.6)	1.0	0.8	6.4
Chairs	Office	(2.2)	8.0	1.4	6.2
Chairs	Furniture	(15.5)	17.4	4.2	8.4
Dogs	Office	(8.6)	10.3	4.0	12.3
Dogs	Easy animals	(19.0)	15.7	5.7	7.7

Table 3: Generalization results of some learnt models to new backgrounds. Each row describes results of a single class model trained against a specific background and tested against other backgrounds. Test errors were computed using a sample of 100 images from each test background. The classifiers based on learnt models perform well in most of the new classification tasks. There is no apparent connection between the difficulty of the training background and successful generalization to new backgrounds.

Indeed, from a purely discriminative point of view, there is no reason to believe that the learnt classifier will be useful when one of the classes (the background) changes. To investigate this issue, we used the learnt models to classify object images against various background images not seen by the learning algorithm. We found that the learnt models tend to generalize well to the new classification problem, as seen in Table 3. These results show that the models have 'generative' qualities: they seem to capture the 'essence' of the object in a way that does not really depend on the background used.

5.4 Recognition performance analysis

In this section we analyze the contribution to performance of several important modeling factors. Specifically, we consider the contribution of modeling part location and scale, and of increasing the number of model parts and features extracted per image.

5.4.1 Location and scale models

The relational components of the model, i.e. the location and scale of the parts, clearly complicate learning considerably, and it is important to understand if they give any performance gain. Table 4 shows comparative results varying the model complexity. Specifically we present results when using only an appearance model, and when adding location and scale models, using the GV detector [11].[8] We can see that although the appearance model produces very reasonable results, adding a location model significantly improves performance. The additional contribution of the scale model is relatively minor. Additionally, by comparing the results of our full blown model (A+L+S) to those presented in Tables 1-2, we can see that the discriminative GV detector usually provides somewhat better results than those obtained using the generic KB detector.

Data Name	A	A+L	A+L+S
Motorbikes	8.1	3.2	3.5
Cars Rear	4.0	1.4	0.6
Airplanes	15.1	15.1	12.1
Faces	6.1	5.2	3.8
Chairs	16.3	10.8	10.9

Table 4: Errors rates using models of varying complexity. (A) Appearance model alone. (A+L) Appearance and location models. (A+L+S) Appearance, location and scale models. The algorithm's parameters were held constant across all experiments.

[8] Similar experiments with the KB detector yielded similar results, but showed no significant improvement with scale modeling.

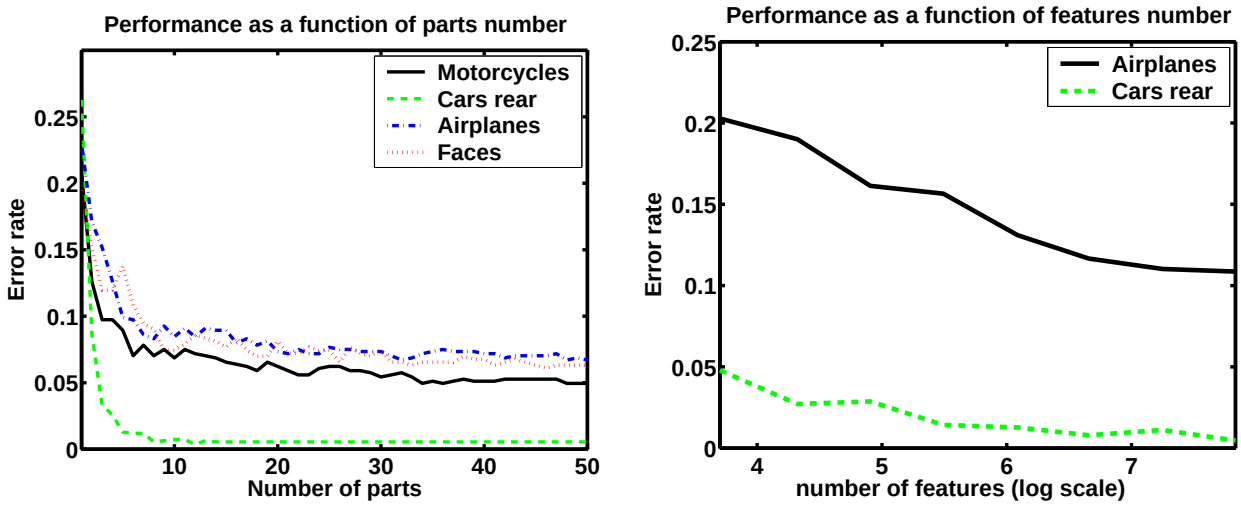


Figure 8: **a)** Error rate as a function of the number of parts P in the model on the Caltech datasets for $N_f = 200$. **b)** Error rate as a function of the number of image features N_f on the Cars rear (easy) and Airplanes (Relatively hard) Caltech datasets, with $P = 30$. In **b)**, the X axis varies between 13 and 228 features in log scale, base 2. All the results were obtained using the KB detector.

5.4.2 Large numbers of parts and features

When hundreds of features are used per image, many features lie in the background of the image, and learning good parts implicitly requires feature pruning. Figure 8 gives error rates as a function of the number of parts and features. Significant performance gains are obtained by scaling up these quantities, indicating that the algorithm is able to find good part models even in the presence of many clutter features. This behavior should be contrasted with the generative learning of a similar model in [21], where increasing the number of parts and features does not usually lead to improved performance. Intuitively, maximum likelihood learning chooses to model features which are frequent in object images, even if these are simple clutter features from the background, while discriminative learning naturally tends to selects more discriminative parts.

5.5 Localization results

Locating an object in a large image is much harder than the binary present/absent detection task. The latter problem is tackled in this paper using a limited set of image features, and a crude grid of possible object locations. For localization we use a similar framework in learning, but turn to a more exhaustive search at the test phase. While searching we do not select representative features, but consider instead as part candidates all the possible image patches at every location and several scales. Object center candidates are taken from a dense grid of possible image locations. To search efficiently, we use the methods proposed in [21, 18], which allow such an exhaustive search in a relatively low computational cost.

The model is applied to an image following a three stage protocol:

1. Convolve the image with the first 15 filters of the DCT base at N_s scales, yielding $N_s \times 15$ coefficient 'activity maps'. We use $N_s = 5$, spanning patch sizes between 5 and 30 pixels.

2. Compute $P \times N_s$ appearance maps by applying the parts appearance models to the vector of DCT coefficients at every image location. The coordinate values (x, y) in map (k, j) contain the log probability of part k with scale j in location (x, y).

3. Apply the relational model to the set of appearance maps, yielding a single log probability map for the 'hidden center' node. To this end, the N_s appearance maps of each part are merged into a single map

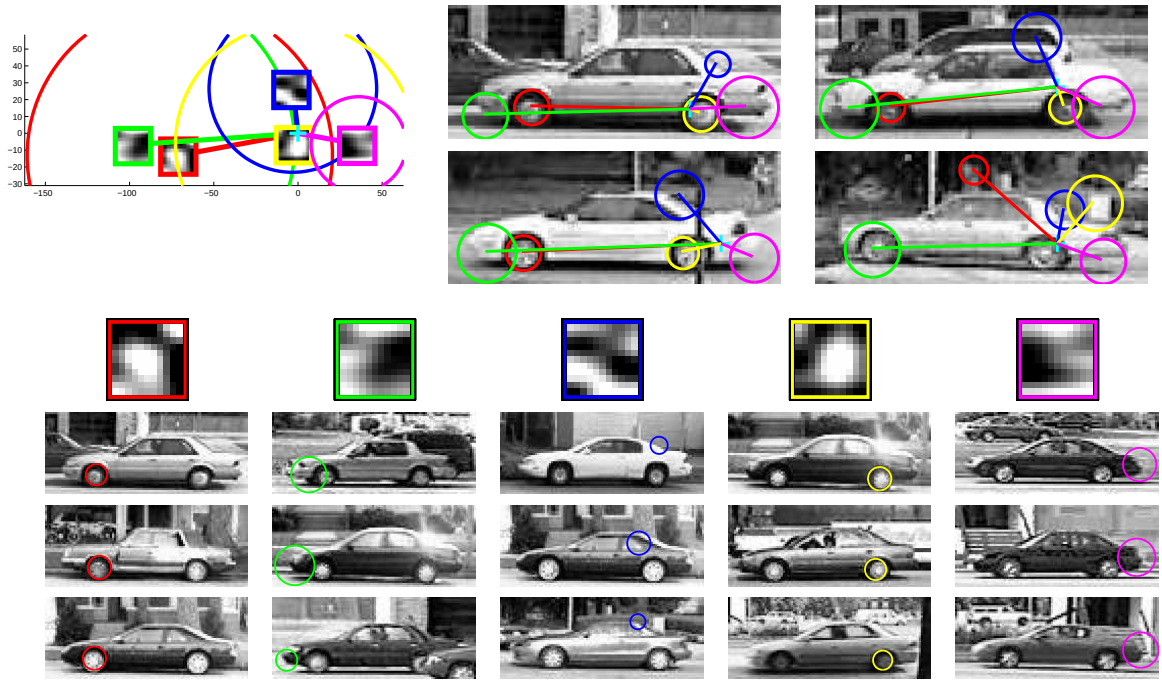


Figure 9: 5 parts from the car side model used in the localization task. The parts shown correspond to the two wheels, front and rear ends, and the top-rear corner. The complete model includes 38 parts, most of them with clear semantics. While the model is not symmetric w.r.t to the x axis, it is not far from being so. It hence happens that a car is successfully detected, but its direction is not properly identified.

by choosing at each coordinate the most likely part scale. We then compute P part message maps, corresponding to the messages $h^k(I, C)$ defined in Eq. (24), by applying the distance transform [18] to the merged appearance maps. Finally the 'hidden center' map is formed as a weighted sum of parts message maps.

The data includes cars facing both directions (i.e. left-to-right and right-to-left). We therefore flip the training images prior to training, such that all cars face the same direction. At the test phase we run the exhaustive search for the learnt model and its mirror image. We detect local maxima in the hidden center map, sort them according to likelihood, and prune neighboring maxima in a way similar to the neighborhood suppression technique suggested in [23]. Figure 10 presents some probability maps and detected cars, illustrating typical successful and problematic detections. Each detection is labeled as hit or miss using the criterion used in [24] (which is slightly different from the one used in [23]), to allow for a fair comparison with other methods. Figure 11 presents a precision-recall curve and a comparison of the achieved localization performance to several recently suggested methods. Our results are comparable to those obtained by the best methods, and are inferior only to a method which uses 'strong' supervision, in the form of images with parts segmentations. In this method part identities are not learnt but chosen manually, and so the learning task is simpler.

6 Discussion

We have presented a method for object class recognition and localization, based on discriminative optimization of a relational generative model. The method combines the natural treatment of spatial part relations, typical to generative classifiers, with the efficiency and pruning ability of discriminative methods. Efficient, scalable learning is achieved by extending boosting techniques to a simple relational model with conditionally dependent parts. In a recognition task, our method compares favorably with several purely generative or

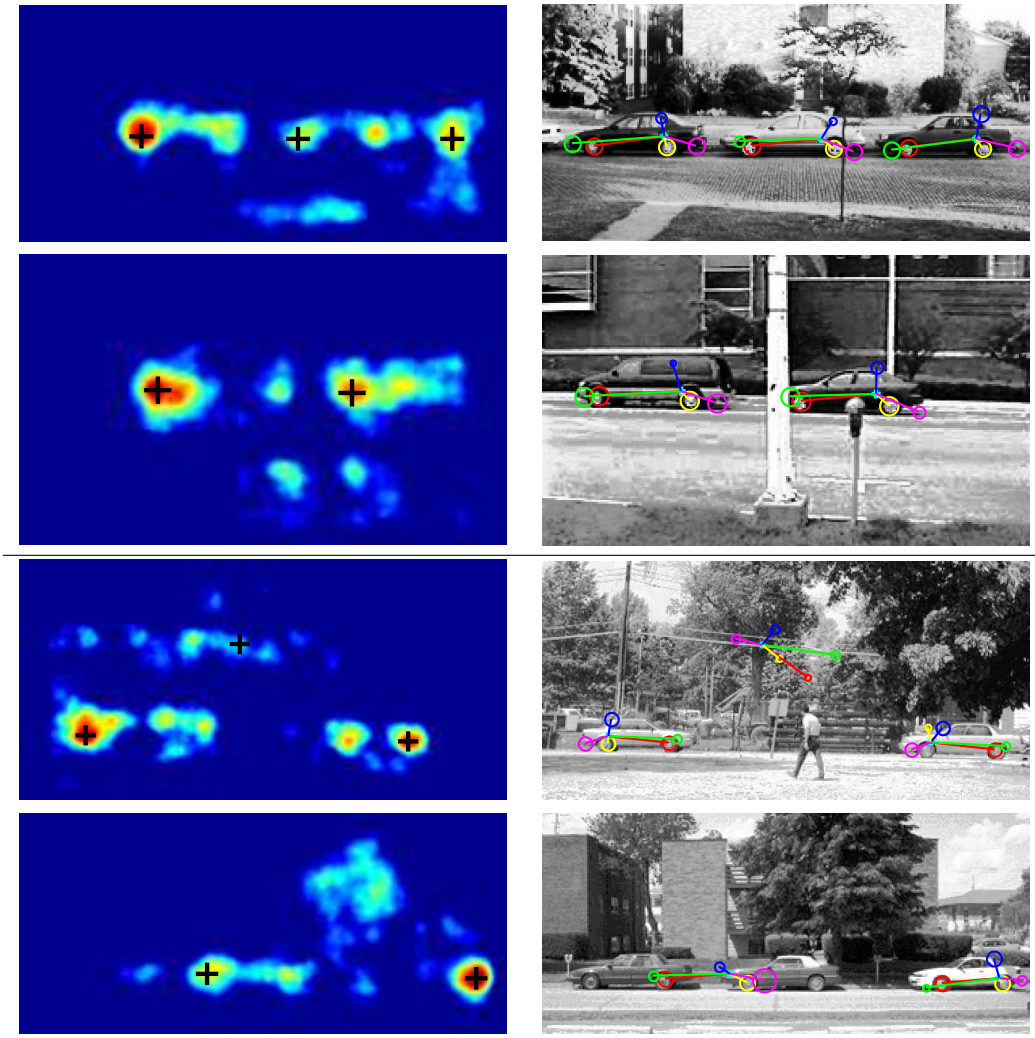
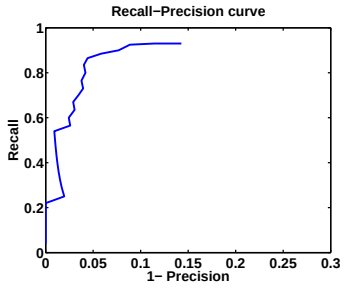


Figure 10: Hidden center probability maps and car detections. In each image pair, the left image shows the probability map for the location of reference point C. The right image shows the 5 parts from Figure 9 superimposed on the detected cars. **Top** Successful detections. Notice that the middle of a gap between two cars tends to emerge as a probable car candidate, as it gains support from both cars. **Bottom** Problematic detections. The third example includes a spurious detection and a car detected using the model of the 'wrong' direction. The bottom example includes a spurious 'middle car' between two real cars. Values in the probability maps were thresholded and linearly transformed for visualization.



a)

Method	Reference	**Equal error rate**
Roth et al.	[24]	0.21
Fergus et al. 2003	[20]	0.115
Fergus et al. 2005	[21]	0.078
Our method	-	0.076
Leibe et. al.*	[6]	0.09 (0.025)

b)

Figure 11: **a)** Recall-Precision curve for cars side detection, using the model shown in Figure 9. **b)** Error error rates (recall=1-precision) obtained on the cars side data by several recent methods.Our performance is comparable to the state-of-the-art methods. images with manually segmented parts. It obtains error rate of 0.09, which is improved to 0.025 using an MDL verification step.

purely discriminative systems recently proposed. In a localization task its performance is comparable to the best available method.

While our recognition results are fair, [1, 27] report better results obtained using discriminative methods which ignore geometric relations and focus instead on feature representation. Specifically, in [1] segmentation based features are used, while in [27] features are based on flexible exhaustive search of 'code book' patches. The recognition performance of these approaches relies on better feature extraction and representation, compared with our simple combination of interest point detection and DCT-based representation. We regard the advances offered by these methods as orthogonal to our main contribution, i.e. the efficient incorporation of geometrical relations. The advantages can be combined by combining better part appearance models and better feature extraction techniques with the relational boosting technique suggested here. We intend to continue our research along these lines.

The complexity of our suggested learning technique is linear in the number of parts and features per image, but it may still be quite expensive. Specifically, the inference complexity of the hidden center C is $O(N_c N_f P)$ where N_c is the number of considered center locations, and this inference is carried for each image many times during learning. This limits us to a relatively crude grid of possible center locations in the learning process, and hence limits the accuracy of the location model learnt. A possible remedy is to consider less exhaustive methods for inferring the optimal hidden center, based on part voting or mean shift mode estimation, as done in [6]. Such 'heuristic' inference solutions may offer enhanced scalability and a more biologically plausible recognition mechanism.

Finally, leaving technical details aside, we regard this work as a contribution to an important debate in learning from unprocessed images, regarding the choice of generative vs. discriminative modeling. We demonstrated that combining generative relational modeling with discriminative optimization can be fruitful and lead to more scalable learning. However, this combination is not free of problems. Our technical problems with covariance matrix learning and the tendency of our technique to produce 'exaggerated' models are two examples. The method proposed here is a step towards the required balance between the descriptive power of generative models and the task relatedness enforced by discriminative optimization.

Acknowledgements

We are grateful to Tomer Hertz, which was involved in early stages of this research for his support and his help in the empirical validation of the method.

A Feature repetition and ML optimization in a star model

Allowing feature repetitions, we derived the likelihood approximation (10) for our star model. For a set of object images $\{I_j\}_{j=1}^n$, This approximation entails the following total data likelihood

$$\begin{aligned}\sum_{j=1}^{n}\log P(I_j|\Theta) &= n\log K_0+\sum_{j=1}^{n}\log[\max_{C}\prod_{K=1}^{P}\max_{x\in F(I_j)}P(x|C,\theta^k)] \\ &= n\log K_0+\sum_{j=1}^{n}\max_{C}\sum_{K=1}^{P}\max_{x\in F(I_j)}\log P(x|C,\theta^k)\end{aligned} \tag{32}$$

The maximum likelihood parameters $\Theta=(\theta_1,..,\theta_P)$ are chosen to maximize this likelihood. In this maximization, we can ignore the constant term $n\log K_0$. To simplify further notation let us denote parts' conditional log likelihood terms by $g_j(C,\theta_k)=\max_{x\in F(I_j)}P(x|C,\theta_k)$. Also denote the vector of the hidden center variables in all images by $\vec{C}=(C_1,..,C_n)$.

$$\max_{\Theta}[\sum_{j=1}^{n}\max_{C}\sum_{K=1}^{P}g_j(C,\theta^k)]=\max_{(C_1,..,C_n)}[\max_{\Theta}\sum_{j=1}^{n}\sum_{K=1}^{P}g_j(C_j,\theta^k)]=\max_{\vec{C}}\sum_{K=1}^{P}\max_{\theta^k}[\sum_{j=1}^{n}g_j(C_j,\theta^k)] \tag{33}$$

For any fixed centers vector $\vec{C}$, and any $1\le k\le P$, the optimal θ^k is determined as $\theta^k=argmax_\theta\, G(\theta,\vec{C})$ where $G(\theta,\vec{C})=\sum_{j=1}^{n}g_j(C_j,\theta)$. Hence, for any $\vec{C}$, the optimal part parameters θ^k are identical, as maxima of the same function. Clearly the maximum over $\vec{C}$ also posses this property.

The proof can be repeated in a similar way for the star model presented in [21], in which the center node is an additional 'landmark' part, as long as the sum over all model interpretations in an image is replaced by the single maximal likelihood interpretation.

B Part weights introduction

Here we establish the functional equivalence between classifiers with and without part weights for weak learners of the form (14). We use the identity

$$\log G(x|\mu,\Sigma)-\nu=\alpha'[\log G(x|\mu',\Sigma')-\nu'] \tag{34}$$

where

$$\mu'=\mu\ ,\quad \Sigma'=\alpha\Sigma\ ,\quad \theta'=\frac{1}{\alpha}[\theta-\frac{(\alpha-1)d}{2}\log 2\pi-\frac{1}{2}(\alpha-\frac{1}{\alpha^d})\log|\Sigma|$$

to introduce part weights into the classifier. This identity is true for all $\alpha>0$. We apply this identity to each part k in the classifier (14), with $\alpha^k=|\Sigma_a^k|^{-1/d}$, to obtain

$$f(I)=\sum_{k=1}^{P}\max_{x\in F(I)}\log G(x|\mu_a^k,\Sigma_a^k)-\nu^k=\sum_{k=1}^{P}\alpha^k[\max_{x\in F(I)}\log G(x|\mu_a^k,\Sigma_a^{k'})-\nu^{k'}] \tag{35}$$

where $\Sigma_a^{k'}=\alpha^k\Sigma_a^k$ is has a fixed determinant of 1 for all parts. The weights α^k therefore (inversely) reflect covariance scale.

C Proof of lemma 1

We differentiate the loss w.r.t. ν

$$0 = \frac{d}{d\nu} \sum_{i=1}^{N} \exp(-y_i[f(I_i) - \nu]) = - \sum_{\{i:y_i=1\}}^{N} \exp(-f(I_i) + \nu) + \sum_{\{i:y_i=-1\}}^{N} \exp(f(I_i) - \nu) \tag{36}$$

For $\tilde{f} = f - \nu$, (36) gives property (21). Solving for ν gives

$$\exp(\nu) \sum_{\{i:y_i=1\}}^{N} \exp(-f(I_i)) = \exp(-\nu) \sum_{\{i:y_i=-1\}}^{N} \exp(f(I_i)) \tag{37}$$

from which (20) follows. Finally, we can compute the loss using the optimal ν^*

$$\begin{aligned} \sum_{i=1}^{N} \exp(-y_i[f(I_i) - \nu^*]) &= \left[\frac{\sum_{\{i:y_i=-1\}}^{N} \exp(f(I_i))}{\sum_{\{i:y_i=1\}}^{N} \exp(-f(I_i))} \right]^{\frac{1}{2}} \sum_{\{i:y_i=1\}}^{N} \exp(-f(I_i)) \\ &+ \left[\frac{\sum_{\{i:y_i=-1\}}^{N} \exp(f(I_i))}{\sum_{\{i:y_i=1\}}^{N} \exp(-f(I_i))} \right]^{-\frac{1}{2}} \sum_{\{i:y_i=-1\}}^{N} \exp(f(I_i)) \end{aligned}$$

from which (22) follows.

References

[1] Opelt A., Fussenegger M., Pinz A., and Auer P. Object recognition with boosting. technical report tr-emt-2004-01. submitted to pami.

[2] Opelt A., Fussenegger M., Pinz A., and Auer P. Weak hypotheses and boosting for generic object detection and recognition. In *ECCV*, 2004.

[3] Torralba A., Murphy K., and Freeman W. Contextual models for object detection using boosted random fields. In *NIPS*, 2004.

[4] Chan A.B., Vasconcelos N., and Moreno P.J. A family of probabilistic kernels based on information divergence, 2004.

[5] NG A.Y. and Jordan M.I. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In *NIPS*, 2001.

[6] Leibe B., Leonardis A., and Schiele B. Combined object categorization and segmentation with an implicit shape model. In *ECCV workshop on statistical learning in computer vision*, 2004.

[7] VOC challenge results of the Pascal visual object classes challenge, 2005.

[8] Lowe D. Local feature view clustering for 3d object recognition. In *CVPR*, pages 682–688, 2001.

[9] Borenstein E., Sharon E., and Ullman S. Combining top-down and bottom-up segmentation. In *IEEE workshop on Perceptual Organization in Computer Vision, (CVPR)*, 2004.

[10] Csurka G., Bray C., Dance C., and Fan L. Visual categorization with bags of keypoints. In *ECCV*, 2004.

[11] D. Gao and N. Vasconcelos. Discriminant saliency for visual recognition from cluttered scenes. In *NIPS*, 2004.

[12] Friedman J. H., Hastie T., and Tibshirani R. Additive logistic regression: a statistical view ofboosting. *Ann. Statist.*, 28:337–407, 2000.

[13] Thureson J. and Carlsson S. Appearance based qualitative image description for object class recognition. In *ECCV*, pages 518–529, 2004.

[14] Murphy K.P., Torralba A., and Freeman W. T. Using the forest to see the trees: a graphical model relating features, objects and scenes. In *NIPS*, 2003.

[15] Mason L., Baxter J., Bartlett P., and Frean M. Boosting algorithms as gradient descent in function space. In *NIPS*, pages 512–518, 2000.

[16] F.F. Li, R. Fergus, and Perona P. A bayesian approach to unsupervised one shot learning of object catgories. In *ICCV*, 2003.

[17] Vidal-Naquet M. and Ullman S. Object recognition with informative features and linear classification. In *ICCV*, 2003.

[18] Feltzenswalb P. and Hutenlocher D. Pictorial structures for object recognition. *IJCV*, 61:55–79, 2005.

[19] Viola P. and Jones M. Rapid object detection using a boosted cascade of simple features. In *CVPR*, 2001.

[20] Fergus R., Perona P., and Zisserman A. Object class recognition by unsupervised scale invariant learning. In *CVPR*. IEEE Computer Society, 2003.

[21] Fergus R., Perona P., and Zisserman A. A sparse object category model for efficient learning and exhaustive recognition. In *CVPR*, 2005.

[22] Schapire R.E. and Singer Y. Improved boosting using confidence-rated predictions. *Machine Learning*, 37(3):297–336, 1999.

[23] Agarwal S., Awan A., and Roth D. Learning to detect objects in images via a sparse, part based representation. *PAMI*, 20(11):1475–1490, 2004.

[24] Agarwal S. and Roth D. Learning a sparse representation for object detection. In *ECCV*, pages 113–130, 2002.

[25] Ullman S., Vidal-Naquet M., and Sali E. Visual features of intermediate complexity and their use in classification. *Nature Neuroscience*, 5:682–687, 2002.

[26] Kadir T. and Brady M. Scale, saliency and image description. *IJCV*, 45(2):83–105, November 2001.

[27] Serre T., Wolf L., and Poggio T. A new biologically motivated framework for robust object recognition. In *CVPR*, 2005.

[28] Vapnik V.N. *Statistical learning theory*. John wiley and sons, 1998.

[29] Freund Y. and Schapire R.E. Experiments with a new boosting algorithm. In *ICML*, pages 148–156, 1996.

Subordinate class recognition using relational object models

Aharon Bar Hillel Daphna Weinshall
The Hebrew university of Jerusalem
Jerusalem 91904
aharonbh,daphna@cs.huji.ac.il

September 26, 2006

Abstract

We address the problem of sub-ordinate class recognition, like the distinction between different types of motorcycles. Our approach is motivated by observations from cognitive psychology, which identify parts as the defining component of basic level categories (like motorcycles), while sub-ordinate categories are more often defined by part properties (like 'jagged wheels'). Accordingly, we suggest a two-stage algorithm: First, a relational part based object model is learnt using unsegmented object images from the inclusive class (e.g., motorcycles in general). The model is then used to build a class-specific vector representation for images, where each entry corresponds to a model's part. In the second stage we train a standard discriminative classifier to classify subclass instances (e.g., cross motorcycles) based on the class-specific vector representation. We describe extensive experimental results with several subclasses. The proposed algorithm typically gives better results than a competing one-step algorithm, or a two stage algorithm where classification is based on a model of the sub-ordinate class.

1 Introduction

Human categorization is fundamentally hierarchical, where categories are organized in tree-like hierarchies. In this organization, higher nodes close to the root describe inclusive classes (like vehicles), intermediate nodes describe more specific categories (like motorcycles), and lower nodes close to the leaves capture fine distinctions between objects (e.g., cross vs. sport motorcycles). Intuitively one could expect such hierarchy to be learnt either bottom-up or top-down (or both), but surprisingly, this is *not* the case. In fact, there is a well defined intermediate level in the hierarchy, called *basic level*, which is learnt first [11]. In addition to learning, this level is more primary than both more specific and more inclusive levels, in terms of many other psychological, anthropological and linguistic measures.

The primary role of basic level categories seems related to the structure of objects in the world. In [13], Tversky & Hemenway promote the hypothesis that the explanation lies in the notion of parts. Their experiments show that basic level categories (like cars and flowers) are often described as a combination of distinctive parts (e.g., stem and petals), which are mostly unique. Higher (super-ordinate and more inclusive) levels are more often described by their function (e.g., 'used for transportation'), while lower (sub-ordinate and more specific) levels are often described by part properties (e.g., red petals) and other fine details. These points are illustrated in Fig. 1.

This computational characterization of human categorization finds parallels in computer vision and machine learning. Specifically, traditional work in pattern recognition focused on discriminating vectors of features, where the features are shared by all objects, with different values. If we make the

Figure 1: **Left** Examples of sub-ordinate and basic level classification. *Top row:* Two motorcycle subordinate classes, sport (right) and cross (left). As members of the same basic level category, they share the same part structure. *Bottom row:* Objects from different basic level categories, like a chair and a face, lack such natural part correspondence. **Right**. Several parts from a learnt motorcycle model as detected in cross and sport motorcycle images. Based on the part correspondence we can build ordered vectors of part descriptions, and conduct the classification in this shared feature space. (Better seen in color)

analogy between features and parts, this level of analysis is appropriate for sub-ordinate categories. In this level different objects share parts but differ in the parts' values (e.g., red petals vs. yellow petals); this is called 'modified parts' in [13].

This discrimination paradigm cannot easily generalize to the classification of basic level objects, mostly because these objects do not share common informative parts, and therefore cannot be efficiently compared using an ordered vector of fixed parts. This problem is partially addressed in a more recent line of work (e.g., [5, 6, 2, 7, 9]), where part-based generative models of objects are learned directly from images. In this paradigm objects are modeled as a set of parts with spatial relations between them. The models are learnt and applied to images, which are represented as unordered feature sets (usually image patches). Learning algorithms developed within this paradigm are typically more complex and less efficient than traditional classifiers learnt in some fixed vector space. However, given the characteristics of human categorization discussed above, this seems to be the correct paradigm to address the classification of basic level categories.

These considerations suggest that sub-ordinate classification should be solved using a two stage method: First we should learn a generative model for the basic category. Using such a model, the object parts should be identified in each image, and their descriptions can be concatenated into an ordered vector. In a second stage, the distinction between subordinate classes can be done by applying standard machine learning tools, like SVM, to the resulting ordered vectors. In this framework, the model learnt in stage 1 is used to solve the correspondence problem: features in the same entry in two different image vectors correspond since they implement the same part. Using this relatively high level representation, the distinction between subordinate categories may be expected to get easier.

Similar notions, of constructing discriminative classifiers on top of generative models, have been recently proposed in the context of object localization [10] and class recognition [7]. The main motivation in these papers was to provide discriminative power to a generative model, optimized by maximum likelihood. Thus the discriminative classifier for a class in [7, 10] uses a generative model of the same class as a representation scheme.[1] In contrast, in this work we use a recent learning algorithm, which already learns a generative relational model of basic categories using a discriminative boosting technique [2]. The new element in our approach is in the learning of a model of one class (the more general basic level category) to allow the efficient discrimination of another class (the more specific sub-ordinates).

Thus our main contribution lies the use of objcet hierarchy, where we represent sub-ordinate classes using models of the more general, basic level class. The approach relies on a specific form of knowledge transfer between classes, and as such it is an instance of the 'learning-to-learn' paradigm. There are several potential benefits to this approach. First and most important is improved accuracy, especially when training data is scarce. For an under-sampled sub-ordinate class, the basic level

[1] An exception to this rule is the Caltech 101 experiment of [7], but there the discriminative classifiers for all 101 classes relies on the same two arbitrary class models.

model can be learnt from a larger sample, leading to a more stable representation for the second stage SVM and lower error rate. A second advantage becomes apparent when scalability is considered: A system which needs to discriminate between many subordinate classes will have to learn and keep considerably less models (only one for each basic level class) if built according to our proposed approach. Such a system can better cope with new subordinate classes, since learning to identify a new class may rely on existing basic class models.

Typically the learning of generative models from unsegmented images is exponential in the number of parts and features [5, 6]. This significantly limits the richness of the generative model, to a point where it may not contain enough detail to distinguish between subclass instances. Alternatively, rich models can be learnt from images with part segmentations [4, 9], but obtaining such training data requires a lot of human labor. The algorithm we use in this work, presented in [2], learns from unsegmented images, and its complexity is linear in the number of model parts and image features. We can hence learn models with many parts, providing a rich object description. In section 3 we discuss the importance of this property.

We briefly describe the model learning algorithm in Section 2.1. The details of the two-stage method are then described in Section 2.2. In Section 3 we describe experiments with sub-classes from six basic level categories. We compare our proposed approach, called BLP (Basic Level Primacy), to a one-stage approach. We also compare to another two-stage approach, called SLP (Subordinate Level Primacy), in which discrimination is done based on a model of the subordinate class. In most cases, the results support our claim and demonstrate the superiority of the BLP method.

2 Algorithms

To learn class models, we use an efficient learning method briefly reviewed in Section 2.1. Section 2.2 describes the techniques we use for subclass recognition.

2.1 Efficient learning of object class models

The learning method from [2] learns a generative relational object model, but the model parameters are discriminatively optimized using an extended boosting process. The class model is learnt from a set of object images and a set of background images. Image I is represented using an unordered feature set $F(I)$ with N_f features extracted by the Kadir & Brady feature detector [8]. The feature set usually contains several hundred features in various scales, with considerable overlap. Features are normalized to uniform size, zero mean and unit variance. They are then represented using their first 15 DCT coefficients, augmented by the image location of the feature and its scale.

The object model is a generative part-based model with P parts (see example in Fig. 2b), where each part is implemented by a single image feature. For each part, its appearance, location and scale are modeled. The appearance of parts is assumed to be independent, while their location and scale are relative to the unknown object location and scale. This dependence is captured by a Bayesian network model, shown in Fig. 2a. It is a star-like model, where the center node is a 3-dimensional hidden node $C = (\vec{C}_l, C_s)$, with the vector $\vec{C}_l$ denoting the unknown object location and the scalar C_s denoting its unknown scale. All the components of the part model, including appearance, relative location and relative log-scale, are modeled using Gaussian distributions with a (scaled) identity covariance matrix.

Based on this model and some simplifying assumptions, the likelihood ratio test classifier is approximated by

$$f(I) = \max_{C} \sum_{k=1}^{P} \max_{x \in F(I)} \log p(x|C, \theta^k) - \nu \qquad (1)$$

This classifier compares the first term, which represents the approximated image likelihood, to a threshold ν. The likelihood term approximates the image likelihood using the MAP interpretation of the model in the image, i.e., it is determined by the single best implementation of model parts by image features. This MAP solution can be efficiently found using standard message passing in time linear in the number of parts P and the number of image features N_f. However, Maximum

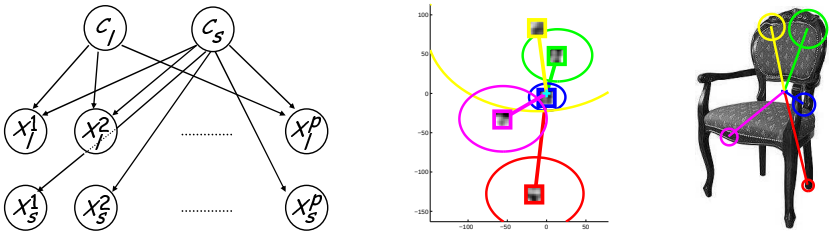


Figure 2: **Left** A Bayesian network specifying the dependencies between the hidden variables C_l, C_s and the parts scale and location X_l^k, X_s^k for $k = 1, .., P$. The part appearance variables X_a^k are independent, and so they do not appear in this network. **Middle** The spatial relations between 5 parts from a learnt chair model. The cyan cross indicates the position of the hidden object center c_l. **Right** The implementations of the 5 parts in a chair image. (Better seen in color)

Likelihood (ML) parameter optimization cannot be used, since the approximation permits part repetition, and as a result the ML solution is vulnerable to repetitive choices of the same part. Instead, the model is optimized to minimize a discriminative loss function.

Specifically, labeling object images by $+1$ and background images by -1, the learning algorithm tries to minimize the exp loss of the margin, $L(f) = \sum_{i=1}^{N} \exp(-y_i f(I_i))$, which is the loss minimized by the Adaboost algorithm [12]. The optimization is done using an extended 'relational' boosting scheme, which generalizes the boosting technique to classifiers of the form (1).

In the relational boosting algorithm, the weak hypotheses (summands in Eq. (1)) are not merely functions of the image I, but depend also on the hidden variable C, which captures the unknown location and scale of the object. In order to find good part hypotheses, the weak learner is given the best current estimate of C, and uses it to guide the search for a discriminative part hypothesis. After the new part hypothesis is added to the model, C is re-inferred and the new estimate is used in the next boosting round. Additional tweaks are added to improve class recognition results, including a gradient descent weak learner and a feedback loop between the optimization of the a weak hypothesis and its weight.

2.2 Subclass recognition

As stated in the introduction, we approach subclass recognition using a two-stage algorithm. In the first stage a model of the basic level class is applied to the image, and descriptors of the identified parts are concatenated into an ordered vector. In the second stage the subclass label is determined by feeding this vector into a classifier trained to identify the subclass. We next present the implementation details of these two stages.

Class model learning Subclass recognition in the proposed framework depends on part consistency across images, and it is more sensitive to part identification failures than the original class recognition task. Producing an informative feature vector is only possible using a rich model with many stable parts. We therefore use a large number of features ($N_f = 400$) per image, and a relatively fine grid of C values, with 10×10 locations over the entire image and 3 scales (a total of $N_c = 300$ possible values for the hidden variable C). We also learn large models with $P = 60$ parts.[2] Note that such large values for N_f and P are not possible in a purely generative framework such as [5, 6] due to the prohibitive computational learning complexity of $O(N_f^P)$.

In [2], model parts are learnt using a gradient based weak learner, which tends to produce exaggerated part location models to enhance its discriminative power. In such cases parts are modeled as being unrealistically far from the object center. Here we restrict the dynamics of the location model in order to produce more realistic and stable parts. In addition, we found out experimentally that when the data contains object images with rich backgrounds, performance of subclass recognition and localization is improved when using models with increased relative location weight. Specifically, a part hypothesis in the model includes appearance, location and scale components with

[2] In comparison, class recognition in [2] was done with $N_f = 200$, $N_c = 108$ and $P = 50$.

relative weights $\lambda_i/(\lambda_1+\lambda_2+\lambda_3),\ \ i=1,2,3$, learnt automatically by the algorithm. We multiply λ_2 of all the parts in the learnt model by a constant factor of 10 when learning from images with rich background. Probabilistically, such increase of λ_2 amounts to smaller location covariance, and hence to stricter demands on the accuracy of the relative locations of parts.

Subclass discrimination Given a learnt object model and a new image, we match for each model part the corresponding image feature which implements it in the MAP solution. We then build the feature vector, which represents the new image, by concatenating all the feature descriptors implementing parts $1,..P$. Each feature is described using a 21-dimensional descriptor including:

- The 15 DCT coefficients describing the feature.
- The relative (x,y) location and log-scale of the feature (relative to the computed MAP value of C).
- A normalized mean of the feature $(m-\hat{m})/std(m)$ where m is the feature's mean (over feature pixels), and $\hat{m}, std(m)$ are the empirical mean and std of m over the P parts in the image.
- A normalized logarithm of feature variance $(v-\hat{v})/std(v)$ with v the logarithm of the feature's variance (over feature pixels) and $\hat{v}, std(v)$ the empirical mean and std of v over image parts.
- The log-likelihood of the feature (according to the part's model).

In the end, each image is represented by a vector of length $21\times P$. The training set is then normalized to have unit variance in all the dimensions, and the standard deviations are stored in order to allow for identical scaling of the test data. Vector representations are prepared in this manner for a training sample including objects from the sub-ordinate class, objects from other sub-ordinate classes of the same basic category, and background images. Finally, a linear SVM [3] is trained to discriminate the target subordinate class images from all other images.

3 Experimental results

Methods: In our experiments, we regard subclass recognition as a binary classification problem in a retrieval scenario. Specifically, The learning algorithm is given a sample of background images, and a sample of unsegmented class images. Images are labeled by the subclass they represent, or as background if they do not contain any object from the inclusive class. The algorithm is trained to identify a specific subclass. In the test phase, the algorithm is given another sample from the same distribution of images, and is asked to identify images from the specific subclass.

Several methodological problems arise in this scenario. First, subclasses are often not mutually exclusive [13], and in many cases there are borderline instances which are inherently ambiguous. This may lead to an ill-defined classification problem. We avoid this problem in the current study by filtering the data sets, leaving only instances with clear-cut subclass affiliation. The second problem concerns performance measurements. The common measure used in related work is the equal error rate of the ROC curve (denoted here EER), i.e., the error obtained when the rate of false positives and the rate of false negatives are equal. However, as discussed in [1], this measure is not well suited for a detection scenario, where the number of positive examples is much smaller than the number of negative examples. A better measure appears to be the equal error rate of the recall-precision curve (denoted here RPC). Subclass recognition has the same characteristics, and we therefore prefer the RPC measure; for completeness, and since the measures do not give qualitatively different results, the EER score is also provided.

The algorithms compared: We compare the performance of the following three algorithms:

- *Basic Level Primacy (BLP)* - The two-stage method for subclass recognition described above, in which a model of the basic level category is used to form the vector representation.
- *Subordinate level primacy (SLP)* - A two-stage method for subclass recognition, in which a model of the sub-ordinate level category is used to form the vector representation.

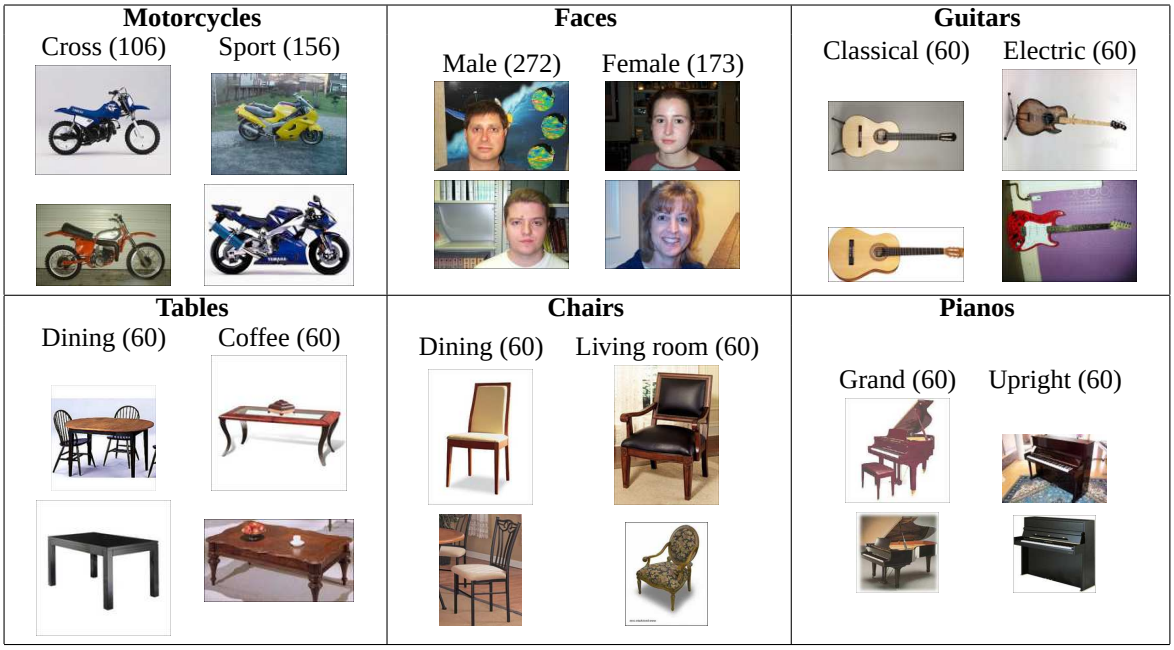


Figure 3: Object images from the subclasses learnt in our experiments. We used 12 subclasses of 6 basic classes. The number of images in each subclass is indicated in the parenthesis next to the subclass name. Individual faces were also considered as subclasses, and the males and females subclasses above include a single example from 4 such individuals.

- *One stage method* - The classification is based on the likelihood obtained by a model of the sub-ordinate class.

The three algorithms use the same training sample in all the experiments. The class models in all the methods were implemented using the algorithm described in Section 2.1, with exactly the same parameters (reported in section 2.2). This algorithm is competitive with current state-of-the-art methods in object class recognition [2].

The third and the second method learn a different model for each subordinate category, and use images from the other sub-ordinate classes as part of the background class during model learning. The difference is that in the third method, classification is done based on the model score (as in [2]), and in the second the model is only used to build a representation, while classification is done with an SVM (as in [7]). The first and second method both employ the distinction between a representation and classification, but the first uses a model of the basic category, and so tries to take advantage of the structural similarity between different subordinate classes of the same basic category.

Datasets We have considered 12 subordinate classes from 6 basic categories. The images were obtained from several sources. Specifically, we have re-labeled subsets of the Caltech Motorcycle and Faces database[3], to obtain the subordinates of sport and cross motorcycles, and male and female faces. For these data sets we have increased the weight of the location model, as mentioned in section 2.2. We took the subordinate classes of grand piano and electric guitar from the Caltech 101 dataset [4] and supplemented them with classes of upright piano and classical guitar collected using google images. Finally, we used subsets of the chairs and furniture background used in [2][5] to define classes of dining and living room chairs, dining and coffee tables. Example images from the data sets can be seen in Fig. 3. In all the experiments, the Caltech office background data was used as the background class. In each experiment half of the data was used for training and the other half for test.

[3] Available at http://www.robots.ox.ac.uk/ vgg/data.html.
[4] Available at http://www.vision.caltech.edu/feifeili/Datasets.htm
[5] Available at http://www.cs.huji.ac.il/ aharonbh/#Data.

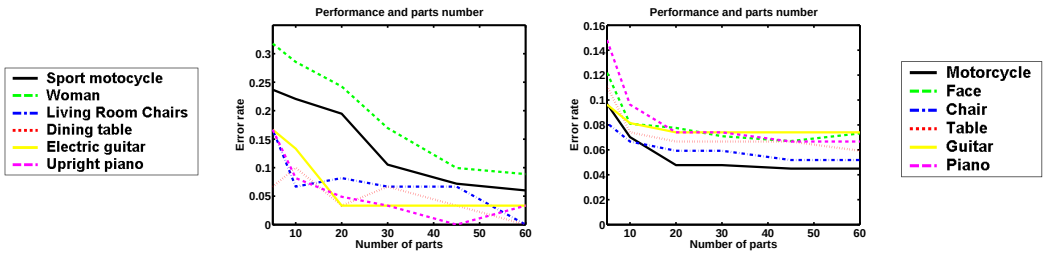


Figure 4: **Left:** RPC error rates as a function of the number of model parts P in the two-stage BLP method, for $5 \leq P \leq 60$. The curves are presented for 6 representative subclasses, one from each basic level category presented in Fig. 3 **Right:** classification error of the first stage classifier as a function of P. This graph reports errors for the 6 basic level models used in the experiments reported on the left graph. In general, while adding only a minor improvement to inclusive class recognition, adding parts beyond 30 significantly improves subclass recognition performance.

In addition, we have experimented with individual faces from the Caltech faces data set. In this experiment each individual is treated as a sub-ordinate class of the Faces basic class. We filtered the faces data to include only people which have at least 20 images. There were 19 such individuals, and we report the results of these experiments using the mean error.

Classification results Table 1 summarizes the classification results. We can see that both two-stage methods perform better than the one-stage method. This shows the advantage of the distinction between representation and classification, which allows the two-stage methods to use the more powerful SVM classifier. When comparing the two two-stage methods, BLP is a clear winner in 7 of the 13 experiments, while SLP has a clear advantage only in a single case. The representation based on the basic level model is hence usually preferable for the fine discriminations required. Overall, the BLP method is clearly superior to the other two methods in most of the experiments, achieving results comparable or superior to the others in 11 of the 13 problems. It is interesting to note that the SLP and BLP show comparable performance when given the individual face subclasses. Notice however, that in this case BLP is far more economical, learning and storing a single face model instead of the 19 individual models used by SLP.

Subclass	**One stage method**		**Subordinate level primacy**		**Basic level primacy**	
Cross motor.	14.5	(12.7)	9.9	(3.5)	**5.5**	**(1.7)**
Sport motor.	10.5	(5.7)	6.6	(5.0)	**4.6**	**(2.6)**
Males	**20.6**	**(12.4)**	24.7	(19.4)	21.9	(16.7)
Females	10.6	(7.1)	10.6	(7.9)	**8.2**	**(5.9)**
Dining chair	6.7	(3.6)	**0**	**(0)**	**0**	**(0)**
Living room chair	6.7	(6.7)	**0**	**(0)**	**0**	**(0)**
Coffee table	13.3	(6.2)	8.4	(6.7)	**3.3**	**(3.6)**
Dining table	6.7	(3.6)	4.9	(3.6)	**0**	**(0)**
Classic guitar	4.9	(3.1)	**3.3**	**(0.5)**	6.7	(3.1)
Electric guitar	6.7	(3.6)	**3.3**	(3.6)	**3.3**	**(2.6)**
Grand piano	10.0	**(3.6)**	10.0	**(3.6)**	**6.7**	(4.0)
Upright piano	**3.3**	(3.6)	10.0	(6.7)	**3.3**	**(0.5)**
Individuals	27.5*	(24.8)*	**17.9***	(7.3)*	19.2*	**(6.5)***

Table 1: Error rates (in percents), when separating subclass images from non-subclass and background images. The main numbers indicate equal error rate of the recall precision curve (RPC). Equal error rate of the ROC (EER) are reported in parentheses. The best result in each row is shown in bold. For the individuals subclasses, the mean over 19 people is reported (marked by $*$). Overall, the BLP method shows a clear advantage.

Performance as a function of number of parts Fig. 4 presents errors as a function of P, the number of class model parts. The graph on the left plots RPC errors of the two stage BLP method on 6 representative data sets. The graph on the right describes the errors of the first stage class models in the task of discriminating the basic level classes background images. While the performance of inclusive class recognition stabilizes after ~ 30 parts, the error rates in subclass recognition continue to drop significantly for most subclasses well beyond 30 parts. It seems that while later boosting

rounds have minor contribution to class recognition in the first stage of the algorithm, the added parts enrich the class representation and allow better subclass recognition in the second stage.

4 Summary and Discussion

We have addressed in this paper the challenging problem of distinguishing between subordinate classes of the same basic level category. We showed that two augmentations contribute to performance when solving such problems: First, using a two-stage method where representation and classification are solved separately. Second, using a larger sample from the more general basic level category to build a richer representation. We described a specific two stage method, and experimentally showed its advantage over two alternative variants.

The idea of separating representation from classification in such a way was already discussed in [7]. However, our method is different both in motivation and in some important technical details. Technically speaking, we use an efficient algorithm to learn the generative model, and are therefore able to use a rich representation with dozens of parts (in [7] the representation typically includes 3 parts). Our experiments show that the large number of model parts i a critical for the success of the two stage method.

The more important difference is that we use the hierarchy of natural objects, and learn the representation model for a more general class of objects - the basic level class (BLP). We show experimentally that this is preferable to using a model of the target subordinate (SLP). This distinction and its experimental support is our main contribution. Compared with the more traditional SLP method, the BLP method suggested here enjoys two significant advantages. First and most importantly, its accuracy is usually superior, as demonstrated by our experiments. Second, the computational efficiency of learning is much lower, as multiple SVM training sessions are typically much shorter than multiple applications of relational model learning. In our experiments, learning a generative relational model per class (or subclass) required 12-24 hours, while SVM training was typically done in a few seconds. This advantage is more pronounced as the number of subclasses of the same class increases. As scalability becomes an issue, this advantage becomes more important.

References

[1] S. Agarwal and D. Roth. Learning a sparse representation for object detection. In *ECCV*, 2002.

[2] A. Bar-Hillel, T. Hertz, and D. Weinshall. Efficient learning of relational object class models. In *ICCV*, 2005.

[3] G.C. Cawley. MATLAB Support Vector Machine Toolbox [http://theoval.sys.uea.ac.uk/˜gcc/svm/toolbox].

[4] P. Feltzenswalb and D. Hutenlocher. Pictorial structures for object recognition. *IJCV*, 61:55–79, 2005.

[5] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale invariant learning. In *CVPR*, 2003.

[6] R. Fergus, P. Perona, and A. Zisserman. A sparse object category model for efficient learning and exhaustive recognition. In *CVPR*, 2005.

[7] AD. Holub, M. Welling, and P. Perona. Combining generative models and fisher kernels for object class recognition. In *International Conference on Computer Vision (ICCV)*, 2005.

[8] T. Kadir and M. Brady. Scale, saliency and image description. *IJCV*, 45(2):83–105, November 2001.

[9] B. Leibe, A. Leonardis, and B. Schiele. Combined object categorization and segmentation with an implicit shape model. In *ECCV workshop on statistical learning in computer vision*, 2004.

[10] Fritz M., Leibe B., Caputo B., and Schiele B. Integrating representative and discriminative models for object category detection. In *ICCV*, pages 1363–1370, 2005.

[11] E. Rosch, C.B. Mervis, W.D. Gray, D.M. Johnson, and P. Boyes-Braem. Basic objects in natural categories. *Cognitive Psychology*, 8:382–439, 1976.

[12] R.E. Schapire and Y. Singer. Improved boosting using confidence-rated predictions. *Machine Learning*, 37(3):297–336, 1999.

[13] B. Tversky and K. Hemenway. Objects, parts, and categories. *Journal of Experimental Psychology: General*, 113(2):169–197, 1984.

Chapter 4

Epilogue

This thesis, and my Ph.D research in general, includes contributions to two research domains: theory and praxis of distance function learning, and visual object class recognition.

- **Distance function learning:**
  - *Theory :* We have shown that multi class classification problems are equivalent in learnability terms to binary distance functions over the product space by showing the connections between the errors and the sample sizes required for the two problems. We showed how a learnt distance function with ε error can be used to find a solution for the corresponding multi-classification problem with error linear in ε.
  - *Algorithms :* We have suggested several algorithms for distance function learning: RCA, Gaussian Coding similarity, Distboost, and Kernelboost, although only the first two are included in this thesis. Specifically, RCA is a simple and efficient algorithm for Mahalanobis metric learning, which has been highly influential in the learning distance community. The Gaussian coding similarity combines a general definition of similarity in information-theoretic terms with a practical, efficient algorithm and good empirical results.
- **Learning object class recognition:**
  - *Object class recognition :* We have studied learning algorithms for part based object recognition based on discriminative optimization of generative models. We considered and compared bag-of-features and relational models. Our method overcomes an inherent problem in maximum likelihood learning of relational models from unsegmented images, and allows efficient learning, which is linear in the number of model parts and image features.

– *Subordinate class recognition :* We suggested a two stage method for the discrimination between similar sub-classes of an object class, based on insights from human cognitive psychology. By taking advantage of a natural object hierarchy, our method allows more accurate learning with fewer object models. To the best of our knowledge, this is the first method yielding such results over a wide range of object classes.

Appendix A

Proof completion for article A

A.1 Proof of Theorem 1

In order to prove this theorem, we first describe a procedure for finding c and h such that their labels are matched. We then look for a lower bound on the ratio

$$\frac{e(\bar{h},\bar{c})}{e(h,c)^2} = \frac{\sum_{i=0}^{\mathbf{M}-1}\sum_{j=0}^{\mathbf{M}-1} p_{ij}(\sum_{k\neq i} p_{kj} + \sum_{k\neq j} p_{ik})}{(1-\sum_{i=0}^{\mathbf{M}-1} p_{ii})^2} \tag{A.1}$$

for the c, h described, where the expressions for the errors are those presented earlier. Finally, we use the properties of the suggested match between c and h to bound the ratio.

Let c, h denote any two original space hypotheses such that $\bar{c} = U(c), \bar{h} = U(h)$. We wish to match the labels of h with the labels of c, using a permutation of the labels of h. If we look at the matrix P, such a permutation is a permutation of the columns, but since the order of the labels in the rows is arbitrary, we may permute the rows as well. Note that the product space error is invariant to permutations of the rows and columns, but the original space error is not: it only depends on the mass of the diagonal, and so we seek permutations which maximize this mass. We suggest an $\mathbf{M}$-step greedy procedure to construct the permuted matrix.

Specifically, denote by $f_r : \{0,..,\mathbf{M}-1\} \to \{0,..,\mathbf{M}-1\}$ and $f_c : \{0,\ldots,\mathbf{M}-1\} \to \{0,\ldots,\mathbf{M}-1\}$ the permutations of the rows and the columns respectively. In step $0 \leq k \leq \mathbf{M}-1$ we extend the definition of both permutations by finding the row and column to be mapped to row and column k. In the first step we find the largest element p_{ij} of the matrix P, and define $f_r(i) = 0, f_c(j) = 0$. In the k-th step we find the largest element of the sub matrix of P with rows $\{0,\ldots,\mathbf{M}-1\}\backslash f_r^{-1}(0,\ldots,k-1)$ and columns $\{0,\ldots,\mathbf{M}-1\}\backslash f_c^{-1}(0,\ldots,k-1)$ (rows and columns not already 'used'). Denoting this element as p_{ij}, we then define $f_r(i) = k$ and $f_c(j) = k$.

Without loss of generality, let P denote the joint distribution matrix after applying the permutations thus defined to the original P's rows and columns. By construction, P now has the property:

$$\forall 0 \leq i \leq \mathbf{M}-1, \;\; \forall j,k \geq i, \quad p_{ii} \geq p_{jk} \tag{A.2}$$

In order to bound the ratio (A.1), we bound the nominator from below as follows

$$\begin{aligned} e(\bar{h},\bar{c}) &= \sum_{j=0}^{\mathbf{M}-1}\sum_{i=0}^{\mathbf{M}-1} p_{ij} \sum_{\substack{k=0 \\ k\neq i}}^{\mathbf{M}-1} p_{kj} + \sum_{i=0}^{\mathbf{M}-1}\sum_{j=0}^{\mathbf{M}-1} p_{ij} \sum_{\substack{k=0 \\ k\neq j}}^{\mathbf{M}-1} p_{ik} \\ &\geq \sum_{j=0}^{\mathbf{M}-1}[\sum_{i\geq j}^{\mathbf{M}-1} p_{ij} \sum_{\substack{k\geq j \\ k\neq i}}^{\mathbf{M}-1} p_{kj}] + \sum_{i=0}^{\mathbf{M}-1}[\sum_{j\geq i}^{\mathbf{M}-1} p_{ij} \sum_{\substack{k\geq i \\ k\neq j}}^{\mathbf{M}-1} p_{ik}] \\ &= \sum_{j=0}^{\mathbf{M}-1}[\sum_{i\geq j}^{\mathbf{M}-1} p_{ij}[\sum_{k\geq j}^{\mathbf{M}-1} p_{kj} - p_{ij}]] + \sum_{i=0}^{\mathbf{M}-1}[\sum_{j\geq i}^{\mathbf{M}-1} p_{ij}[\sum_{k\geq i}^{\mathbf{M}-1} p_{ik} - p_{ij}]] \end{aligned}$$

and then use constraint (A.2) to improve the bound

$$\geq \sum_{j=0}^{\mathbf{M}-1}[\sum_{i\geq j}^{\mathbf{M}-1} p_{ij}[\sum_{k\geq j}^{\mathbf{M}-1} p_{kj} - p_{jj}]] + \sum_{i=0}^{\mathbf{M}-1}[\sum_{j\geq i}^{\mathbf{M}-1} p_{ij}[\sum_{k\geq i}^{\mathbf{M}-1} p_{ik} - p_{ii}]]$$

The denominator in (A.1) is the square of $e(c,h)$, which can be written as

$$e(h,c) = 1 - \sum_{k=0}^{\mathbf{M}-1} p_{ii} = \sum_{k=0}^{\mathbf{M}-1}[\sum_{i>k} p_{ik} + \sum_{j>k} p_{kj}]$$

To simplify the notations, denote

$$\begin{aligned} \mathbf{M}_k^v &= \sum_{j>k} p_{jk} \qquad 0 \leq k \leq \mathbf{M}-1 \\ \mathbf{M}_k^h &= \sum_{i>k} p_{ki} \qquad 0 \leq k \leq \mathbf{M}-1 \end{aligned}$$

Changing variables from $\{p_{ij}\}_{i,j=0}^{\mathbf{M}-1}$ to $\{\mathbf{M}_k^v, \mathbf{M}_k^h, p_{kk}\}_{k=0}^{\mathbf{M}-1}$, ratio (A.1) becomes

$$\frac{e(\bar{h},\bar{c})}{e(h,c)^2} = \frac{\sum_{k=0}^{\mathbf{M}-1} (\mathbf{M}_k^v + p_{kk})\mathbf{M}_k^v + (\mathbf{M}_k^h + p_{kk})\mathbf{M}_k^h}{(\sum_{k=0}^{\mathbf{M}-1} \mathbf{M}_k^v + \mathbf{M}_k^h)^2}$$

Now we use the inequality $\sum_{i=1}^{N} a_i^2 \geq \frac{1}{N}(\sum_{i=1}^{N} a_i)^2$ (for positive arguments) twice to get the required bound:

$$\begin{aligned}
\frac{\sum_{k=0}^{\mathbf{M}-1}(\mathbf{M}_k^v + p_{kk})\mathbf{M}_k^v + (\mathbf{M}_k^h + p_{kk})\mathbf{M}_k^h}{(\sum_{k=0}^{\mathbf{M}-1}\mathbf{M}_k^v + \mathbf{M}_k^h)^2} &\geq \frac{\sum_{k=0}^{\mathbf{M}-1}(\mathbf{M}_k^v)^2 + (\mathbf{M}_k^h)^2}{(\sum_{k=0}^{\mathbf{M}-1}\mathbf{M}_k^v + \mathbf{M}_k^h)^2} \geq \frac{\sum_{k=0}^{\mathbf{M}-1}\frac{1}{2}(\mathbf{M}_k^v + \mathbf{M}_k^h)^2}{(\sum_{k=0}^{\mathbf{M}-1}\mathbf{M}_k^v + \mathbf{M}_k^h)^2} \\
&\geq \frac{\frac{1}{2\mathbf{M}}(\sum_{k=0}^{\mathbf{M}-1}\mathbf{M}_k^v + \mathbf{M}_k^h)^2}{(\sum_{k=0}^{\mathbf{M}-1}\mathbf{M}_k^v + \mathbf{M}_k^h)^2} = \frac{1}{2\mathbf{M}}
\end{aligned}$$

A.2 Completion of the proof of Theorem 5

We have observed that

$$2^{d_{pr}} \leq (\frac{2d_{pr}e(\mathbf{M}+1)^2}{2d_o})^{d_o}$$

Following the proof of Thm. 10 in [4], let us write

$$d_{pr}\ln 2 \leq d_o[\ln\frac{d_{pr}}{d_o} + \ln(e(\mathbf{M}+1)^2)]$$

Using the inequality $\ln(x) \leq xy - \ln(ey)$ which is true for all $x, y \geq 0$, we get

$$\begin{aligned}
d_{pr}\ln 2 &\leq d_o[\frac{d_{pr}}{d_o}y - \ln ey + \ln e(\mathbf{M}+1)^2] \\
&\leq d_{pr}y + d_o\ln\frac{(\mathbf{M}+1)^2}{y} \\
&\leq \frac{d_o}{ln(2)-y}ln\frac{(\mathbf{M}+1)^2}{y} = \frac{2d_o ln(\mathbf{M}+1) - d_o lny}{ln(2)-y} \\
&= \frac{2ln2d_o log_2(\mathbf{M}+1) - d_o lny}{ln(2)-y}
\end{aligned}$$

If we limit ourselves to $y < 1$ then $(-d_o \ln y) \geq 0$, and therefore we can multiply this expression by $\log_2(\mathbf{M}+1) > 1$ and keep the inequality. Hence

$$d_{pr} \leq \frac{(2\ln 2 - \ln y)}{\ln 2 - y} d_o \log_2(\mathbf{M}+1)$$

Finally we choose $y = 0.34$ to get the bound

$$d_{pr} \leq 4.87 d_o \log_2(\mathbf{M}+1)$$

A.3 Completion of the proof of Theorem 6

Lemma 1. *Each cluster $g^{-1}(i)\ \ i = 0, \ldots, \mathbf{M}_g$ intersects at most one of the sets $\{c^{-1}(j) \cap \mathcal{G}\}_{j=0}^{\mathbf{M}-1}$.*

Proof. According to Lemma 2 , $c^{-1}(j_1) \cap \mathcal{G}$ and $c^{-1}(j_2) \cap \mathcal{G}$ for $j_1 \neq j_2$ are two $\frac{K}{3}$-balls that are separated by a distance $\frac{4K}{3}$ in the $L1$ metric space. By construction $g^{-1}(i)$ is an open ball with diameter $\frac{4K}{3}$. Hence it cannot intersect more than one of the sets $\{c^{-1}(j) \cap \mathcal{G}\}_{j=0}^{\mathbf{M}-1}$. □

Lemma 2. *The labeling function g as defined above has the following properties:*

1. *g defines an $\mathbf{M}$-class partition, i.e., $\mathbf{M}_g = \mathbf{M}$.*
2. *There is a bijection $J : \{0, \ldots, \mathbf{M}-1\} \to \{0, \ldots, \mathbf{M}-1\}$ matching the sets $\{g^{-1}(i)\}_{i=0}^{\mathbf{M}-1}$ and $\{c^{-1}(i) \cap \mathcal{G}\}_{i=0}^{\mathbf{M}-1}$ such that*

$$g^{-1}(i) \cap (c^{-1}(J(i)) \cap \mathcal{G}) \neq \phi$$
$$g^{-1}(i) \cap (c^{-1}(l) \cap \mathcal{G}) = \phi \quad for\ \ l \neq J(i)$$

3. *$|Y \backslash G_0| < \frac{3\varepsilon}{K}$.*

Proof. Assume without loss of generality that the classes are ordered according to their size, i.e.

$$|c^{-1}(j) \cap \mathcal{G}| \geq |c^{-1}(j+1) \cap \mathcal{G}|,\ \ j = 0, \ldots, \mathbf{M} - 2$$

We claim that

$$|g^{-1}(i)| \geq |c^{-1}(i) \cap \mathcal{G}|,\ \ i = 0, \ldots, \mathbf{M} - 1$$

To see this, note that since each of the sets $\{g^{-1}(j)\}_{j=0}^{i-1}$ intersects at most one of the sets $\{c^{-1}(j) \cap \mathcal{G}\}_{j=0}^{i}$, there is at least one $l \in \{1, \ldots, i\}$ such that $c^{-1}(l) \cap \mathcal{G}$ has not yet been touched in the i-th step. This set is contained in a $\frac{k}{3}$-ball, and hence it is contained in $B_{\frac{2K}{3}}(x)$ for each of its members. Therefore, at this step our greedy procedure must choose a set of the form $B_{\frac{2K}{3}}(x)$ such that

$$|B_{\frac{2K}{3}}(x)| = \max_{y \in S_{i-1}} |B_{\frac{2K}{3}}(y)| \geq |c^{-1}(l) \cap \mathcal{G}| \geq |c^{-1}(i) \cap \mathcal{G}| \tag{A.3}$$

Using condition (5) in the theorem, we can see that the set is bigger than the algorithm's stopping condition:

$$|c^{-1}(i) \cap \mathcal{G}| \geq |c^{-1}(i)| - |\mathcal{B}| \geq KN - \frac{3\varepsilon N}{K} > KN - \frac{KN}{2} = \frac{KN}{2} \tag{A.4}$$

and therefore the algorithm cannot stop just yet. Furthermore, it follows from the same condition that $\frac{KN}{2} > \frac{3\varepsilon N}{K}$, and thus the chosen set is bigger than $\mathcal{B}$. This implies that it has non empty intersection with $c^{-1}(j)$ for some $j \in \{0, \ldots, \mathbf{M}-1\}$. We have already shown that this j is unique, and so we can define the matching $J(i) = j$.

After $\mathbf{M}$ steps we are left with the set $S_{\mathbf{M}}$ with size

$$\begin{aligned} |S_{\mathbf{M}}| &= N - \sum_{i=0}^{\mathbf{M}-1} |g^{-1}(i)| \leq N - \sum_{i=0}^{\mathbf{M}-1} |c^{-1}(i) \cap \mathcal{G}| \\ &= N - |\mathcal{G}| \leq N - (N - \frac{3\varepsilon N}{K}) = \frac{3\varepsilon N}{K} < \frac{KN}{2} \end{aligned}$$

where the last inequality once again follows from condition (5) . The algorithm therefore stops at this step, which completes the proof of claims (1) and (3) of the lemma.

To prove claim (2) note that since $|S_{\mathbf{M}}| \leq \frac{3\varepsilon N}{K}$, it is too small to contain a whole set of the form $c^{-1}(i) \cap \mathcal{G}$. We know from Eq. (A.4) that for all i $|c^{-1}(i) \cap \mathcal{G}| > \frac{KN}{2} > \frac{3\varepsilon N}{K}$. Hence, for each $j \in 0, \ldots, \mathbf{M}-1$ there is an $i \in 0, \ldots, \mathbf{M}-1$ such that

$$(c^{-1}(j) \cap \mathcal{G}) \cap g^{-1}(i) \neq \phi$$

Therefore $J : \{0, \ldots, \mathbf{M}-1\} \rightarrow \{0, \ldots, \mathbf{M}-1\}$ which was defined above is a surjection, and since both its domain and range are of size $\mathbf{M}$, it is also a bijection. □

We can now complete the proof of Thm. 6 :

Proof. Let $\tilde{g}$ denote the composition $J \circ g$. We will show that for $x \in G_0$, $fiber^f(x)$ of a point x on which $\tilde{g}(x)$ makes an error is far from the 'true' fiber $fiber^c(x)$ and hence x is in $\mathcal{B}$. This will prove that $e(c, \tilde{g}) < \frac{3\varepsilon}{K}$ over G_0. Since the remaining domain $Y \backslash G_0$ is known from Lemma 2 to be smaller than $\frac{3\varepsilon}{K}N$, this concludes the proof.

Assume $c(x) = i, \tilde{g}(x) = j$ and $i \neq j$ hold for a certain point $x \in G_0$. x is in $\tilde{g}^{-1}(j) \cap G_0$ which is the set chosen by the algorithm at step $i = J^{-1}(j)$. This set is known to contain a point $z \in c^{-1}(j) \cap \mathcal{G}$. On the one hand, z is in $\tilde{g}^{-1}(j)$, which is a ball of radius $\frac{2K}{3}$. Thus $d(fiber^f(x), fiber^f(z)) < \frac{4K}{3}$. On the other hand, z is in $c^{-1}(j) \cap \mathcal{G}$ and hence $d(fiber^f(z), fiber^c(z)) < \frac{K}{3}$. We can use the triangle inequality to get

$$\begin{aligned} d(fiber^f(x), fiber^c(z)) &\leq d(fiber^f(x), fiber^f(z)) + d(fiber^f(z), fiber^c(z)) \\ &< \frac{4K}{3} + \frac{K}{3} = \frac{5K}{3} \end{aligned}$$

This inequality implies that $fiber^f(x)$ is far from the 'true' fiber $fiber^c(x)$:

$$\begin{aligned} 2K &\leq d(fiber^c(x), fiber^c(z)) \\ &\leq d(fiber^c(x), fiber^f(x)) + d(fiber^f(x), fiber^c(z)) \\ &< d(fiber^c(x), fiber^f(x) + \frac{5K}{3} \\ &\implies d(fiber^c(x), fiber^f(x) > \frac{K}{3} \end{aligned}$$

and hence $x \in \mathcal{B}$. □

Appendix B

Coding similarity: FLD as a margin optimizer

In this appendix we prove theorem 1 from section 1.3, stating that FLD projection maximizes the expected margin of the Gaussian Coding similarity. We will need several lemmas before we prove the main theorem.

Lemma 3. *For two matrices of the form* $M_1 = \begin{pmatrix} A & 0 \\ 0 & A \end{pmatrix}, M_2 = \begin{pmatrix} A & B \\ B & A \end{pmatrix}$ *where* $A, B \in M_{d\times d}$ *and* A *is an invertible matrix,*

1. $\frac{|M_2|}{|M_1|} = \frac{|A-BA^{-1}B|}{|A|}$
2. $tr(M_1 M_2^{-1}) = 2tr(A(A - BA^{-1}B)^{-1})$

Proof. 1.

$$\frac{|M_2|}{|M_1|} = |M_1^{-1}M_2| = |\begin{pmatrix} A^{-1} & 0 \\ 0 & A^{-1} \end{pmatrix}\begin{pmatrix} A & B \\ B & A \end{pmatrix}| = |\begin{pmatrix} I & A^{-1}B \\ A^{-1}B & I \end{pmatrix}|$$

we multiply this matrix from the left with $\begin{pmatrix} I & 0 \\ A^{-1}B & -I \end{pmatrix}$ to get

$$(-1)^D|\begin{pmatrix} I & A^{-1}B \\ A^{-1}B & I \end{pmatrix}| = |\begin{pmatrix} I & 0 \\ A^{-1}B & -I \end{pmatrix}\begin{pmatrix} I & A^{-1}B \\ A^{-1}B & I \end{pmatrix}| =$$
$$|\begin{pmatrix} I & A^{-1}B \\ 0 & (A^{-1}B)^2 - I \end{pmatrix}| = |(A^{-1}B)^2 - I|$$

Hence We got on the one hand $\frac{|M_2|}{|M_1|} = (-1)^D|(A^{-1}B)^2 - I| = |I - (A^{-1}B)^2|$. On the other hand

$$\frac{|A - BA^{-1}B|}{|A|} = |A^{-1}(A - BA^{-1}B)| = |I - (A^{-1}B)^2|$$

2. The proof stems from the following fact:

   For a matrix of the form $M_2 = \begin{pmatrix} A & B \\ B & A \end{pmatrix}$, $M_2^{-1} = \begin{pmatrix} C & D \\ D & C \end{pmatrix}$ with $C = (A - BA^{-1}B)^{-1}, D = -A^{-1}B(A - BA^{-1}B)^{-1}$

   This fact can be verified by multiplying the matrices.

$$tr(M_1 M_2^{-1}) = tr(M_2^{-1} M_1) = tr(\begin{pmatrix} C & D \\ D & C \end{pmatrix} \begin{pmatrix} A & 0 \\ 0 & A \end{pmatrix}) = tr(\begin{pmatrix} CA & DA \\ DA & CA \end{pmatrix}$$
$$= 2tr(CA) = 2tr(AC) = 2tr(A(A - BA^{-1}B)^{-1})$$

□

The following lemma is a known result in P.S.D. matrix theory:

Lemma 4. (Lemma on the Schur complement)

Let $A = \begin{pmatrix} B & C^t \\ C & D \end{pmatrix}$ *be a symmetric matrix with* $k \times k$ B *and* $l \times l$ *block* D*. Assume that* $B > 0$*. Then* $A \geq 0$ *iff* $D - CB^{-1}C^t \geq 0$*.*

The matrix $D - CB^{-1}C$ is called the Schur complement of B in A.

Lemma 5. *Let* $\Sigma_{x|x'} = \Sigma_x - \Sigma_{xx'}\Sigma_x^{-1}\Sigma_{xx'}$ *be the conditional covariance matrix of a multi dimensional Gaussian distribution p(x,x') in* R^d *(the notations* Σ_x *,* $\Sigma_{xx'}$ *are as defined in section 1.3). For any projection matrix* $A \in M_{d \times k}$

$$\Sigma_x - \Sigma_{xx'}A(A^t\Sigma_x A)^{-1}A^t\Sigma_{xx'} \geq \Sigma_{x|x'} \tag{B.1}$$

Proof. Rearranging terms, we have to prove that

$$\Sigma_{xx'}\Sigma_x^{-1}\Sigma_{xx'} - \Sigma_{xx'}A(A^t\Sigma_x A)^{-1}A^t\Sigma_{xx'} \geq 0$$

We apply the Schur lemma to the following matrix

$$M = \begin{pmatrix} A^t\Sigma_x A & A^t \\ A & \Sigma_x^{-1} \end{pmatrix}$$

The Schur complement of Σ_x^{-1} in M is $A^t\Sigma_x A - A^t\Sigma_x A = 0 \geq 0$. Hence , by Schur's lemma $M \geq 0$.

Now $A^t\Sigma_x A \geq 0$ since $A^t\Sigma_x A = (A^t(\Sigma_x^{0.5})^t)\Sigma_x^{0.5}A = B^tB$. Using Schur's lemma again we get that the Schur complement of $A^t\Sigma_x A$ in M is PSD, i.e. $\Sigma_x^{-1} - A(A^t\Sigma_x A)^{-1}A^t \geq 0$. Since $\Sigma_{xx'}$ is symmetric, we can multiply the inequality from the left and the right to get $\Sigma_{xx'}\Sigma_x^{-1}\Sigma_{xx'} - \Sigma_{xx'}A(A^t\Sigma_x A)^{-1}A^t\Sigma_{xx'} \geq 0$ as required. □

Lemma 6. *Let $\Sigma_{x|x'} = \Sigma_x - \Sigma_{xx'}\Sigma_x^{-1}\Sigma_{xx'}$ be the conditional covariance matrix of a Gaussian distribution p(x,x') in R^d, and let $A \in M_{d\times k}$ be any projection matrix. Denote the covariance matrices after projection by $\Sigma_z = A^t\Sigma_x A$, $\Sigma_{zz'} = A^t\Sigma_{xx'}A$ and $\Sigma_{z|z'} = \Sigma_z - \Sigma_{zz'}\Sigma_z^{-1}\Sigma_{zz'}$. In addition denote $\Sigma_{app} = A^t\Sigma_{x|x'}A$. For all* $0 \leq \alpha \leq 1$

$$\frac{1-2\alpha}{2}\log\frac{|\Sigma_{z|z'}|}{|\Sigma_z|} + (1-\alpha)tr(\Sigma_z\Sigma_{z|z'}^{-1}) \leq \frac{1-2\alpha}{2}\log\frac{|\Sigma_{app}|}{|\Sigma_z|} + (1-\alpha)tr(\Sigma_z\Sigma_{app}^{-1})$$

Proof. We have to show that

$$\frac{2\alpha-1}{2}\log\frac{|\Sigma_{z|z'}|}{|\Sigma_{app}|} + (1-\alpha)tr(\Sigma_z(\Sigma_{app}^{-1} - \Sigma_{z|z'}^{-1})) \geq 0$$

First we show it for $\alpha > \frac{1}{2}$: In this case $\frac{2\alpha-1}{2}, 1-\alpha$ are both positive, and so it is enough to show that $\log\frac{|\Sigma_{z|z'}|}{|\Sigma_{app}|}$, $tr(\Sigma_z(\Sigma_{app}^{-1} - \Sigma_{z|z'}^{-1}))$ are positive. Both of the inequalities result from lemma 5. Multiplying inequality B.1 by A^t, A from right and left respectively we get

$$\Sigma_{z|z'} = A^t\Sigma_x A - A^t\Sigma_{xx'}A(A\Sigma_x A^t)^{-1}A^t\Sigma_{xx'}A \geq A^t\Sigma_x A - A^t\Sigma_{xx'}\Sigma_x^{-1}\Sigma_{xx'}A = \Sigma_{app}$$

The determinant is monotone in its argument and so $|\Sigma_{z|z'}| \geq |\Sigma_{app}|$, which proves the claim for the determinant ratio. Regarding the trace:

$$\begin{aligned}\Sigma_{z|z'} \geq \Sigma_{app} &\rightarrow \Sigma_{z|z'}^{-1} \leq \Sigma_{app}^{-1} \\ \rightarrow \Sigma_{app}^{-1} - \Sigma_{z|z'}^{-1} \geq 0 &\rightarrow \Sigma_z(\Sigma_{app}^{-1} - \Sigma_{z|z'}^{-1}) \geq 0 \\ &\rightarrow tr(\Sigma_z(\Sigma_{app}^{-1} - \Sigma_{z|z'}^{-1})) \geq 0\end{aligned}$$

Where multiplying by Σ_z keeps the inequality since it is symmetric.

Now the case of $0 \leq \alpha \leq \frac{1}{2}$: In this range $\frac{1-2\alpha}{2(1-\alpha)} \leq 1$ and $\log\frac{|\Sigma_{z|z'}|}{|\Sigma_{app}|} \geq 0$, so it's enough to prove

$$tr(\Sigma_z(\Sigma_{app}^{-1} - \Sigma_{z|z'}^{-1})) \geq \log\frac{|\Sigma_{z|z'}|}{|\Sigma_{app}|}$$

Simplifying notation, denote $A = \Sigma_z\Sigma_{z|z'}^{-1}$, $B = \Sigma_z\Sigma_{app}^{-1}$. In terms of these matrices, we have to show $tr(B -$ A$) \geq \log\frac{|B|}{|A|}$.

Since $\Sigma_{z|z'} \geq \Sigma_{app}$, we get $\Sigma_{z|z'}^{-1} \leq \Sigma_{app}^{-1}$ and thus $A = \Sigma_z\Sigma_{z|z'}^{-1} \leq \Sigma_z\Sigma_{app}^{-1} = B$. Also, we can see that $A \geq I$

$$A^{-1} = \Sigma_{z|z'}\Sigma_z^{-1} = (\Sigma_z - \Sigma_{zz'}\Sigma_z^{-1}\Sigma_{zz'})\Sigma_z^{-1} = I - (\Sigma_{zz'}\Sigma_z^{-1})^2 \leq I$$

We can use this fact as follows:

$$tr(B-A) \geq tr(A^{-1}(B-A)) = tr(A^{-1}B - I) = \sum_{i=1}^{N} (\lambda_i - 1)$$

where $\{\lambda_i\}_{i=1}^N$ are the eigenvalues of $A^{-1}B$. Using the inequality $x \geq \log(1+x)$ we can proceed to

$$tr(B-A) \geq \sum_{i=1}^{N} (\lambda_i - 1) \geq \sum_{i=1}^{N} \log(\lambda_i) = \log|A^{-1}B| = \log\frac{|B|}{|A|}$$

as required. □

Lemma 7. *Let $A \in M_{d\times k}$ be any projection matrix and $\Sigma_z = A^t\Sigma_x A, \Sigma_{app} = A^t\Sigma_{x|x'}A$ as was defined in lemma 6. Then for all $0 \leq \alpha \leq 1$ the maximum*

$$\max_A \; [\; \frac{1-2\alpha}{2} \log\frac{|\Sigma_{app}|}{|\Sigma_z|} + (1-\alpha) tr(\Sigma_z \Sigma_{app}^{-1}) \;] \tag{B.2}$$

is obtained by placing in A the k eigenvectors of $\Sigma_{x|x'}^{-1}\Sigma_x$ with the highest eigenvalues.

Proof. Differentiating the argument of Eq. B.2 w.r.t A we get

$$\begin{aligned}
&(1-2\alpha)(\Sigma_{x|x'}A\Sigma_{app}^{-1} - \Sigma_x A\Sigma_z^{-1}) + 2(1-\alpha)(-\Sigma_{x|x'}A\Sigma_{app}^{-1}\Sigma_z\Sigma_{app}^{-1} + \Sigma_x A\Sigma_{app}^{-1}) \\
&= \Sigma_{x|x'}A[(1-2\alpha)\Sigma_{app}^{-1} - 2(1-\alpha)\Sigma_{app}^{-1}\Sigma_z\Sigma_{app}^{-1}] + \Sigma_x A[-(1-2\alpha)\Sigma_z^{-1} + 2(1-\alpha)\Sigma_{app}^{-1}]
\end{aligned}$$

Multiplying by Σ_{app} from the right, $\Sigma_{x|x'}^{-1}$ from the left, and equating to 0 we get

$$\Sigma_{x|x'}^{-1}\Sigma_x A = A[(1-2\alpha) - 2(1-\alpha)\Sigma_{app}^{-1}\Sigma_z] \cdot [(1-2\alpha)\Sigma_z^{-1}\Sigma_{app} - 2(1-\alpha)]^{-1}$$

The left side can be simplified by extracting a $\Sigma_{app}^{-1}\Sigma_z$ from the first matrix:

$$\Sigma_{x|x'}^{-1}\Sigma_x A = A\Sigma_{app}^{-1}\Sigma_z[(1-2\alpha)\Sigma_z^{-1}\Sigma_{app} - 2(1-\alpha)] \cdot [(1-2\alpha)\Sigma_z^{-1}\Sigma_{app} - 2(1-\alpha)]^{-1} = A\Sigma_{app}^{-1}\Sigma_z$$

We now use the simultaneous diagonalization of Σ_z, Σ_{app}. There exists an invertible matrix $B \in M_{k\times k}$ such that $B^t\Sigma_{app}B = I, B^t\Sigma_z B = \Lambda$ where Λ is a diagonal matrix. Simple algebra shows that $\Sigma_{app}^{-1}\Sigma_z = B\Lambda B^{-1}$, i.e. B contains the eigenvectors of $\Sigma_{app}^{-1}\Sigma_z$. Rewriting the last equation, we get

$$\Sigma_{x|x'}^{-1}\Sigma_x AB = AB\Lambda$$

The matrix AB contains eigenvectors of $\Sigma_{x|x'}^{-1}\Sigma_x$. On the other hand, the argument of B.2 is invariant to the replacement of A by AB, as both the trace and the determinant terms are invariant with respect to this transformation. Hence there is an optimal solution A which contains eigenvectors of $\Sigma_{x|x'}^{-1}\Sigma_x$. Moreover, we see that the

eigenvalues of $\Sigma_{app}^{-1}\Sigma_z$ at this solution are also eigenvalues of $\Sigma_{x|x'}^{-1}\Sigma_x$. We will use this fact now to determine which eigenvalues (and corresponding eigenvectors) of $\Sigma_{x|x'}^{-1}\Sigma_x$ should be chosen for A.

Denote the eigenvalues of $\Sigma_{app}^{-1}\Sigma_z$ by $\{\mu_i\}_{i=1}^k$. The maximization argument in B.2 can be expressed using these eigenvalues:

$$\frac{2\alpha-1}{2}\log|\Sigma_{app}^{-1}\Sigma_z| + (1-\alpha)tr(\Sigma_{app}^{-1}\Sigma_z) = \frac{2\alpha-1}{2}\sum_{i=1}^{k}\ln\mu_i + (1-\alpha)\sum_{i=1}^{k}\mu_i$$
$$= \sum_{i=1}^{k}[(1-\alpha)\mu_i + (\alpha-\frac{1}{2})\ln\mu_i] \triangleq \sum_{i=1}^{k} f(\mu_i)$$

Exploring $f(\mu)$, we can see that it isn't monotonic. However, we are only interested in the behavior of $f(\mu)$ in the range of the eigenvalues of $\Sigma_{x|x'}^{-1}\Sigma_x$ and these are all bigger than 1. To see this note that they are given by expressions of the form

$$\frac{v^t\Sigma_x v}{v^t\Sigma_{x|x'}v} = \frac{v^t\Sigma_x v}{v^t(\Sigma_x - \Sigma_{xx'}\Sigma_x^{-1}\Sigma_{xx'})v}$$

where v is the corresponding eigenvector. Since $\Sigma_x > 0$ and $\Sigma_x^{-1} > 0$, clearly $\Sigma_{xx'}\Sigma_x^{-1}\Sigma_{xx'} > 0$ and so $\Sigma_x - \Sigma_{xx'}\Sigma_x^{-1}\Sigma_{xx'} < \Sigma_x$. Hence $v^t(\Sigma_x - \Sigma_{xx'}\Sigma_x^{-1}\Sigma_{xx'})v < v^t\Sigma_x v$ for any v.

For $\mu > 0$, $f(\mu)$'s monotonic increase zone is:

$$\frac{df}{d\mu} = 1 - \alpha + \frac{(\alpha-0.5)}{\mu} \geq 0 \leftrightarrow \mu > \mu_0 = \frac{0.5-\alpha}{1-\alpha}$$

The denominator of μ_0 is always positive. For $\alpha > \frac{1}{2}$ the nominator is negative and hence $f(\mu)$ is increasing for all $\mu > 0$. For $0 \leq \alpha \leq \frac{1}{2}$ we have $\mu_0 = \frac{0.5-\alpha}{1-\alpha} \leq \frac{0.5}{0.5} = 1$, and so $f(\mu)$ is increasing for all $\mu \geq 1$. Hence the matrix A maximizing B.3 is the choice of the D eigenvectors of $\Sigma_{x|x'}^{-1}\Sigma_x$ with the highest eigenvalues. □

We can now prove our main theorem.

Theorem 8. *Assume Gaussian distributions $p(x, x'|H_1)$ in R^{2d} and $p(x)$ in R^d, and a linear projection $A \in M_{d\times k}$ where $z = A^t x$. For all $0 \leq \alpha \leq 1$, the optimal A*

$$A^* = \arg\max_{A\in M_{d\times k}} \alpha D_{kl}[p(z,z'|H_1)||p(z,z'|H_0)] + (1-\alpha)D_{kl}[p(z,z'|H_0)||p(z,z'|H_1)]$$

is the FLD transformation. Thus A is composed of the k eigenvectors of $\Sigma_x^{-1}\Sigma_{xx'}$ with the highest eigenvalues.

Proof. Without loss of generality, we can assume that $x \sim N(0, \Sigma_x)$. We therefore have $P(x, x'|H_1) = N(0, \Sigma_{2x})$, where $\Sigma_{2x} = \begin{pmatrix} \Sigma_x & \Sigma_{xx'} \\ \Sigma_{xx'} & \Sigma_x \end{pmatrix}$. $\Sigma_{xx'}$ is the covariance matrix $\mathrm{cov}(x_1, x_2)$ where x_1, x_2 come from the same

class. We have seen in equation 9 in section 1.3 that this covariance matrix is symmetric, and that it equals $S_{between}$ in Fisher's terminology. In addition, $P(x, x'|H_0) = P(x)P(x') = N(0, \Sigma_{X^2})$ where $\Sigma_{x^2} = \begin{pmatrix} \Sigma_x & 0 \\ 0 & \Sigma_x \end{pmatrix}$. The $D_{kl}[\cdot||\cdot]$ distance between two Gaussians $(\mu_1, \Sigma_1), (\mu_2, \Sigma_2)$ is given by

$$D_{kl}[\mathrm{N}(\mu_1, \Sigma_1)||N(\mu_2, \Sigma_2)] = \frac{1}{2}[\log \frac{|\Sigma_2|}{|\Sigma_1|} + tr(\Sigma_1\Sigma_2^{-1} + \Sigma_2^{-1}(\mu_2 - \mu_1)(\mu_2 - \mu_1)^t) - d]$$

Using this formula, we obtain the following expression for the expected margin from Eq. 11 in section 3.1:

$$\frac{\alpha}{2}[\log \frac{|\Sigma_{x^2}|}{|\Sigma_{2x}|} + tr(\Sigma_{2x}\Sigma_{x^2}^{-1}) - 2N] + \frac{1-\alpha}{2}[\log \frac{|\Sigma_{2x}|}{|\Sigma_{x^2}|} + tr(\Sigma_{x^2}\Sigma_{2x}^{-1}) - 2N] + (1-2\alpha)\log\frac{1-\alpha}{\alpha}$$

Since $tr(\Sigma_{2x}\Sigma_{x^2}^{-1}) = tr(\Sigma_{x^2}^{-1}\Sigma_{2x})$ and

$$\Sigma_{x^2}^{-1}\Sigma_{2x} = \begin{pmatrix} \Sigma_x & 0 \\ 0 & \Sigma_x \end{pmatrix}^{-1} \begin{pmatrix} \Sigma_x & \Sigma_{xx'} \\ \Sigma_{xx'} & \Sigma_x \end{pmatrix} = \begin{pmatrix} I & \Sigma_x^{-1}\Sigma_{xx'} \\ \Sigma_x^{-1}\Sigma_{xx'} & I \end{pmatrix}$$

We get $tr(\Sigma_{2x}\Sigma_{x^2}^{-1}) = N$. The expected margin is therefore reduced to

$$\frac{1-2\alpha}{2}\log\frac{|\Sigma_{2x}|}{|\Sigma_{x^2}|} + \frac{1-\alpha}{2}tr(\Sigma_{x^2}\Sigma_{2x}^{-1}) - N + (1-2\alpha)\log\frac{1-\alpha}{\alpha}$$

Applying a linear transformation $A \in M_{d\times k}$ to x_1, x_2 is the same as applying $B = \begin{pmatrix} A & 0 \\ 0 & A \end{pmatrix}$ to the concatenated vector (x_1, x_2). The resulting distributions in R^{2k} are also Gaussians and the expected margin is

$$\frac{1-2\alpha}{2}\log\frac{|\Sigma_{2z}|}{|\Sigma_{z^2}|} + \frac{1-\alpha}{2}tr(\Sigma_{z^2}\Sigma_{2z}^{-1}) - N + (1-2\alpha)\log\frac{1-\alpha}{\alpha}$$

where $\Sigma_{z^2}, \Sigma_{2z}$ are defined in terms of $\Sigma_z = A^t\Sigma_x A$ and $\Sigma_{zz'} = A^t\Sigma_{xx'}A$ in an analogous manner to the definitions of $\Sigma_{x^2}, \Sigma_{2x}$. In order to maximize this expression w.r.t A it is easier to move to $d \times d$ matrices instead of the $2d \times 2d$ matrices $\Sigma_{z^2}, \Sigma_{2z}$. Using lemma 3 we can rewrite the expected margin after transformation as follows:

$$f(A) = \frac{1-2\alpha}{2}\log\frac{|\Sigma_{z|z'}|}{|\Sigma_z|} + (1-\alpha)tr(\Sigma_z\Sigma_{z|z'}^{-1}) - N + (1-2\alpha)\log\frac{1-\alpha}{\alpha}$$

where $\Sigma_{z|z'} = \Sigma_z - \Sigma_{zz'}\Sigma_z^{-1}\Sigma_{zz'}$ can be identified as the covariance matrix of the conditional Gaussian $P(z_1|z_2)$. Computing derivatives for this expression is intricate, so we replace it with the upper bound suggested by lemma 6, in which $\Sigma_{app} = A^t\Sigma_{x|x'}A$ replaces $\Sigma_{z|z'}$:

$$g(A) = \frac{1-2\alpha}{2}\log\frac{|\Sigma_{app}|}{|\Sigma_z|} + (1-\alpha)tr(\Sigma_z\Sigma_{app}^{-1}) + C$$

where $C = -N + (1-2\alpha)\log\frac{1-\alpha}{\alpha}$ is constant w.r.t. A. According to lemma 7 the maximum of $g(A)$ is obtained by the k highest eigenvectors of $\Sigma_x^{-1}\Sigma_{xx'}$, and we denote this matrix by A^*. Since $f(A) \leq g(A) \leq g(A^*)$ for all A, it is enough to show that $f(A^*) = g(A^*)$. Since the only difference between $f(A)$ and $g(A)$ is in the substitution of $\Sigma_{z|z'}(A)$ with $\Sigma_{app}(A)$, it is enough to show that $\Sigma_{z|z'}(A^*) = \Sigma_{app}(A^*)$.

Denote by B be the mutual diagonalization matrix of $\Sigma_x, \Sigma_{xx'}$, i.e.

$$B^t\Sigma_x B = I, \;\; B^t\Sigma_{xx'}B = \Lambda = diag(\{\lambda_i\}), \;\; \Sigma_x^{-1}\Sigma_{xx'} = B\Lambda B^{-1}$$

B diagonalize $\Sigma_{x|x'}$:

$$B^t\Sigma_{x|x'}B = B^t(\Sigma_x - \Sigma_{xx'}\Sigma_x^{-1}\Sigma_{xx'})B = I - B^t\Sigma_{xx'}BB^{-1}\Sigma_x^{-1}\Sigma_{xx'}B = I - \Lambda^2$$

As we have seen in lemma 7, A^* can be expressed using the first k columns of B ($A^* = B[:, 1:k]$ in Matlab notation). Since $B^t\Sigma_{x|x'}B = I - \Lambda^2$, we get that $\Sigma_{app}(A^*) = (A^*)^t\Sigma_{x|x'}A^* = I_k - \Lambda_k^2$, where Λ_k is the restriction of Λ to the first k rows and columns ($\Lambda_k = \Lambda[1:k, 1:k]$). On the other hand

$$\Sigma_{z|z'}(A^*) = (A^*)^t\Sigma_x A^* - (A^*)^t\Sigma_{xx'}A^*[(A^*)^t\Sigma_x A^*]^{-1}(A^*)^t\Sigma_{xx'}A^* = I_k - \Lambda_k I_k^{-1}\Lambda_k = I_k - \Lambda_k^2$$

hence $\Sigma_{z|z'}(A^*) = \Sigma_{app}(A^*)$ and the optimal A^* is composed of the first k eigenvectors of $\Sigma_{x|x'}^{-1}\Sigma_x$.

Finally, we claim that these vectors are identical with the first k (in descending order of the eigenvalues) eigenvectors of $\Sigma_x^{-1}\Sigma_{xx'}$. Since B defined above diagonalizes both Σ_x and $\Sigma_{x|x'}$, it contains the eigenvectors of $\Sigma_{x|x'}^{-1}\Sigma_x$:

$$B^{-1}\Sigma_{x|x'}^{-1}\Sigma_x B = B^{-1}\Sigma_{x|x'}^{-1}(B^t)^{-1}B^t\Sigma_x B = (B^t\Sigma_{x|x'}B)^{-1}\cdot I = (I-\Lambda^2)^{-1}$$

So every eigenvector of $\Sigma_x^{-1}\Sigma_{xx'}$ with an eigenvalue λ is also an eigenvector of $\Sigma_{x|x'}^{-1}\Sigma_x$ with eigenvalue $\frac{1}{1-\lambda^2}$. Assuming the matrix is in general position (i.e. $\Sigma_x^{-1}\Sigma_{xx'}$, $\Sigma_{x|x'}^{-1}\Sigma_x$ each has d eigenvectors with distinct eigenvalues), the opposite also holds, i.e. every eigenvector of $\Sigma_{x|x'}^{-1}\Sigma_x$ is an eigenvector of $\Sigma_x^{-1}\Sigma_{xx'}$. Moreover, Since $\frac{1}{1-\lambda^2}$ monotonically increases in λ, the order of the eigenvalues is the same for both matrices. We hence get that the optimal A^* holds the first eigenvectors of $\Sigma_x^{-1}\Sigma_{xx'} = \Sigma_{Total}^{-1}\Sigma_{Between}$ in Fisher's terminology, which are the FLD projection. □

Bibliography

[1] S. Agarwal and D. Roth. Learning a sparse representation for object detection. In *European Conf. on Computer Vision (ECCV)*, pages 113–130, 2002.

[2] C. Aggarwal. Towards systematic design of distance functions for data mining applications. In *Conf. on Knowledge Discovery and Data Mining (KDD)*, pages 9–19, 2003.

[3] J. A. Aslam and M. Frost. An information-theoretic measure for document similarity. In *the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 449–450, New York, NY, USA, 2003. ACM Press.

[4] V. Athitsos, J. Alon, S. Sclaroff, and G. Kollios. Boostmap: A method for efficient approximate similarity rankings. In *Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2004.

[5] Multiple authors. *VOC challenge results of the Pascal visual object classes challenge*. http://www.pascal-network.org/challenges/VOC/voc/, 2005.

[6] A. Bar-hillel, T. Hertz, N. Shental, and D. Weinshall. Learning distance functions using equivalence relations. In *International Conference on Machine Learning (ICML)*, 2003.

[7] H.G. Barrow, J.M. Tenenbaum, R.C. Bolles, and H.C. Wolf. Parametric correspondence and chamfer matching: Two new techniques for image matching. *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 659–663, 1977.

[8] E. Bart and S. Ullman. Cross-generalization: learning novel classes from a single example by feature replacement. In *Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2005.

[9] E. Bart and S. Ullman. Single-example learning of novel classes using representation by similarity. In *British Machine Vision Conference (BMVC)*, 2005.

[10] P. Bartlett, Y. Freund, W. S. Lee, and R. E. Schapire. Boosting the margin: a new explanation for the effectiveness of voting methods. *Ann. Statist.*, 26, no. 5:1651–1686, 1998.

[11] R. Basri and S. Ullman. The alignment of objects with smooth surfaces. *Computer Vision, Graphics, and Image Processing: Image Understanding*, 57(3):331–345, 1993.

[12] J. Baxter. Learning internal representations. In *Annual Conference On Learning Theory (COLT)*, 1995.

[13] J. Baxter. Theoretical models of learning to learn. In *T. Mitchell and S. Thrun (editors) Learning to Learn. Kluwer, Boston, 1997.*, 1997.

[14] P.N. Belhumeur, J. Hespanha, and D. Kriegman. Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 19(7):711–720, 1997.

[15] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 24(24):509–522, 2002.

[16] S. Ben-David and R. Schuller. Exploiting task relatedness for multitask learning. In *Annual Conference On Learning Theory (COLT)*, 2003.

[17] A. C. Berg, T. L. Berg, and J. Malik. Shape matching and object recognition using low distortion correspondence. In *Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2005.

[18] T. De Bie, M. Momma, and N. Cristianini. Efficiently learn the metric with side information. *Lecture Notes in Artificial Intelligence*, 2842:175 – 189, 2003.

[19] T. De Bie, J. Suykens, and B. De Moor. Learning from general label constraints. In *the joint IAPR international workshops on Syntactical and Structural Pattern Recognition (SSPR) and and Statistical Pattern Recognition (SPR)*, Lisbon, August 2004.

[20] I. Biederman. Human image understanding: Recent research and a theory. *Computer vision, Graphics, and image processing*, 32:29–73, 1985.

[21] M. Bilenko, S. Basu, and R. Mooney. Integrating constraints and metric learning in semi-supervised clustering. In *International Conference on Machine Learning (ICML)*, pages 81–88, 2004.

[22] T.O. Binford. Visual perception by computer. *IEEE Conf. on Systems and Controls*, 1971.

[23] C.L. Blake and C.J. Merz. *UCI Repository of machine learning databases. University of California, Irvine, Dept. of Information and Computer Sciences.* http://www.ics.uci.edu/∼mlearn/MLRepository.html., 1998.

[24] M. Blatt, S. Wiseman, and E. Domany. Data clustering using a model granular magnet. *Neural Computation*, 9(8):1805–1842, 1997.

[25] E. Borenstein and S. Ullman. Class-specific, top-down segmentation. In *European Conf. on Computer Vision (ECCV)*, volume 2, pages 109–124, 2002.

[26] O. Bousquet and D.J.L. Herrmann. On the complexity of learning the kernel matrix. In *Advances in Neural Information Processing Systems (NIPS)*, 2002.

[27] R. Brunelli and T. Poggio. Hyperbf networks for gender classification. In *the DARPA Image Understanding Workshop*, pages 311–314, 1992.

[28] R. Brunelli and T. Poggio. Face recognition: features versus templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 15(10):1042–1052, 1993.

[29] M. Burl, T. K. Leung, M. Weber, and P. Perona. *Recognition of Visual Object Classes*. *a chapter in* From Segmentation to Interpretation and Back: Mathematical Methods in Computer Vision, Springer, in press, 1996.

[30] P. Carbonetto, G. Dork, and C. Schmid. *Bayesian learning for weakly supervised object classification*. Technical Report, INRIA Rhne-Alpes, 2004.

[31] R. Caruana. Multitask learning. *Machine Learning*, 28:41–75, 1997.

[32] H. Chang and D.Y. Yeung. Locally linear metric adaptation for semi-supervised clustering. In *International Conference on Machine Learning (ICML)*, 2004.

[33] H. Chang and D.Y. Yeung. Kernel-based metric adaptation with pairwise constraints. In *International Conference on Machine Learning and Cybernetics (ICMLC)*, pages 721–730, 2005.

[34] H. Chang and D.Y. Yeung. Stepwise metric adaptation based on semi-supervised learning for boosting image retrieval performance. In *British Machine Vision Conference (BMVC)*, 2005.

[35] O. Chapelle, V. Vapnik, O. Bousquet, and S. Mukherjee. Choosing multiple parameters for support vector machines. *Machine Learning*, 46(1–3):131–159, 2002.

[36] Y. Chen, J. Bi, and J. Z. Wang. Miles: Multiple-instance learning via embedded instance selection. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 28(12), 2006.

[37] S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In *Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2005.

[38] D. Cohn, R. Caruana, and A. McCallum. Semi-supervised clustering with user feedback. Technical report, citeseer.ist.psu.edu/cohn03semisupervised.html 2003.

[39] K. Crammer, J. Keshet, and Y. Singer. Kernel design using boosting. In *Advances in Neural Information Processing Systems (NIPS)*, volume 15, 2002.

[40] D. Crandall, P. Felzenszwalb, and D. Huttenlocher. Spatial priors for part-based recognition using statistical models. In *Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2005.

[41] D. Crandall and D. Huttenlocher. Weakly supervised learning of part-based spatial models for visual object recognition. In *European Conf. on Computer Vision (ECCV)*, 2006.

[42] N. Cristianini, J. Shawe-Taylor, A. Elisseeff, and J. Kandola. On kernel-target alignment. In *Advances in Neural Information Processing Systems (NIPS)*, 2001.

[43] G. Csurka, C. Bray, C. Dance, and L. Fan. Visual categorization with bags of keypoints. In *European Conf. on Computer Vision (ECCV)*, 2004.

[44] T. Deselaers, D. Keysers, and H. Ney. Discriminative training for object recognition using image patches. In *Conf. on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 157–162, 2005.

[45] S.J. Dickinson, R. Bergevin., I. Biederman, J.O. Eklundh, R. Munck-Fairwood, A.K. Jain, and A.P. Pentland. Panel report: The potential of geons for generic 3-d object recognition. *Image and Vision Computing*, 15(4):277–292, 1997.

[46] C. Dillon and T. Caelly. Leaning image annotation: the cite system. *Videre*, 1:90–121, 1998.

[47] G. Dork and C. Schmid. Object class recognition using discriminative local features. *Submitted to IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2006.

[48] B. Draper. *Learning Control Strategies for Object Recognition*. Symbolic Visual Learning, Ikeuchi and Veloso (eds.), Oxford University Press, 1996.

[49] B. Draper, B. Collins, J. Brolio, A. Hanson, and E. Riseman. The schema system. *International Journal of computer vision (IJCV)*, 2:209–250, 1989.

[50] R.O Duda, P.E. Hart, and D.G. Stork. *Pattern Classification*. John Wiley and Sons Inc., 2001.

[51] B. Epshtein and S. Ullmam. Satellite features for the classification of visually similar classes. In *Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2006.

[52] P. Feltzenswalb and D. Hutenlocher. Pictorial structures for object recognition. *International Journal of computer vision (IJCV)*, 61:55–79, 2005.

[53] A. Ferencz, E. Learned-Miller, and J. Malik. Building a classification cascade for visual identification from one example. In *International Conference on Computer Vision (ICCV)*, pages 286–293, 2005.

[54] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale invariant learning. In *Conf. on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, 2003.

[55] R. Fergus, P. Perona, and A. Zisserman. A sparse object category model for efficient learning and exhaustive recognition. In *Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2005.

[56] M. Fink. Object classification from a single example utilizing class relevance pseudo-metrics. In *Advances in Neural Information Processing Systems (NIPS)*, 2004.

[57] M.A. Fischler and R.A. Elschlager. The representation and matching of pictrial structures. *IEEE transactions of computer*, 22(1):67–92, 1973.

[58] R.A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7:179–188, 1936.

[59] M. Fritz, B. Leibe, B. Caputo, and B. Schiele. Integrating representative and discriminant models for object category detection. In *International Conference on Computer Vision (ICCV)*, pages 1363–1370, 2005.

[60] Y. Gdalyahu and D. Weinshall. Flexible syntactic matching of curves and its application to automatic hierarchical classification of silhouettes. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 21(12), 1999.

[61] Y. Gdalyahu, D. Weinshall, and M. Werman. Self organization in vision: stochastic clustering for image segmentation, perceptual grouping, and image database organization. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 23(10):1053–1074, 2001.

[62] R. Gilad-Bachrach, A. Navot, and N. Tishby. Margin based feature selection - theory and algorithms. In *International Conference on Machine Learning (ICML)*, 2004.

[63] A. Globerson and S. Roweis. Metric learning by collapsing classes. In *Advances in Neural Information Processing Systems (NIPS)*, volume 19, 2005.

[64] J. Goldberger, S. Roweis, G. Hinton, and R. Salakhutdinov. Neighbourhood component analysis. In *Advances in Neural Information Processing Systems (NIPS)*, 2004.

[65] B. Golomb, D. Lawrence, and T. Sejnowski. Sexnet: a neural network identifies sex from human faces. In *Advances in Neural Information Processing Systems (NIPS)*, pages 572–577, 1991.

[66] K. Grauman and T. Darrell. The pyramid match kernel: Discriminative classification with sets of image features. In *International Conference on Computer Vision (ICCV)*, 2005.

[67] S. Gutta, P. J. Wechsler, and J. Phillips. Gender and ethnic classification. In *International Conference on Automatic Face and Gesture Recognition*, pages 194–199, 1998.

[68] I. Guyon, S. Gunn, M. Nikravesh, and L. Zadeh. *Feature extraction, foundations and applications*. Springer, 2006.

[69] T. Hastie and R. Tibshirani. Discriminant adaptive nearest neighbor classification and regression. In David S. Touretzky, Michael C. Mozer, and Michael E. Hasselmo, editors, *Advances in Neural Information Processing Systems (NIPS)*, volume 8, pages 409–415. The MIT Press, 1996.

[70] T. Hertz, A. Bar-Hillel, N. Shental, and D. Weinshall. Enhancing image and video retrieval: Learning via equivalence constraints. In *Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2003.

[71] T. Hertz, A. Bar-Hillel, and D. Weinshall. Boosting margin based distance functions for clustering. In *International Conference on Machine Learning (ICML)*, 2004.

[72] T. Hertz, A. Bar-Hillel, and D. Weinshall. Learning distance functions for image retrieval. In *Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2004.

[73] T. Hertz, A. Bar Hillel, and D. Weinshall. Learning a kernel function for classification with small training samples. In *International Conference on Machine Learning (ICML)*, 2006.

[74] A. Holub, M. Welling, and P. Perona. Combining generative models and fisher kernels for object class recognition. In *International Conference on Computer Vision (ICCV)*, 2005.

[75] D. P. Huttenlocher, G. A. Klanderman, and W. A. Rucklidge. Comparing images using the hausdorff distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 15(9):850–863, 1993.

[76] D. P. Huttenlocher and S. Ullman. Object recognition using alignment. In *International Conference on Computer Vision (ICCV)*, pages 102–111, 1987.

[77] D. W. Jacobs, D. Weinshall, and Y. Gdalyahu. Classification with non metric distances: Image retrieval and class representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 22(6):583–600, 2000.

[78] V. Jain, A. Ferencz, and E. Learned-Miller. Discriminative training of hyper-feature models for object identification. In *British Machine Vision Conference (BMVC)*, 2006.

[79] F. Jurie and B. Triggs. Creating efficient codebooks for visual recognition. In *International Conference on Computer Vision (ICCV)*, 2005.

[80] E. R. Kandel, J. H. Schwartz, and T. M. Jessell. *Principles of neural science, Forth edition.* Mcgraw hill, 2000.

[81] I. Kant. *The Critique of Judgement*. Hackett, 1791.

[82] C. Kemp, A. Bernstein, and J.B. Tenenbaum. A generative theory of similarity. In *CogSci*, 2005.

[83] D. Klein, S. Kamvar, and C. Manning. From instance-level constraints to space-level constraints: Making the most of prior knowledge in data clustering. In *International Conference on Machine Learning (ICML)*, 2002.

[84] J. T. Kwok and I. W. Tsang. Learning with idealized kernels. In *International Conference on Machine Learning (ICML)*, pages 400–407, 2003.

[85] X. Lan and D. Huttenlocher. Beyond trees: Common factor models for 2d human pose recovery. In *International Conference on Computer Vision (ICCV)*, pages 470–477, 2005.

[86] G.R.G. Lanckriet, N. Christianini, P.L. Bartlett, L.E. Ghaoui, and M.I. Jordan. Learning the kernel matrix with semi-definite programming. In *International Conference on Machine Learning (ICML)*, pages 323–330, 2002.

[87] P. Langley. Selection of relevant features in machine learning. In *proceedings of the AAAI symposium on relevance, New Orleans*. LA:AAAI press, 1994.

[88] M. Law, A. Topchy, and A. K. Jain. Model-based clustering with probabilistic constraints. In *Proceedings of SIAM Data Mining*, pages 641–645, 2005.

[89] B. Leibe, A. Leonardis, and B. Schiele. Combined object categorization and segmentation with an implicit shape model. In *ECCV workshop on statistical learning in computer vision*, 2004.

[90] F. F. Li, R. Fergus, and P. Perona. Learning generative visual models from few training examples: an incremental bayesian approach tested on 101 object categories. In *CVPR Workshop on Generative-Model Based Vision*, 2004.

[91] F.F. Li, R. Fergus, and Perona P. A bayesian approach to unsupervised one shot learning of object catgories. In *International Conference on Computer Vision (ICCV)*, 2003.

[92] D. lin. An information theoretic definition of similarity. In *International Conference on Machine Learning (ICML)*, 1998.

[93] N. Loeff, H. Arora, A. Sorokin, and D. Forsyth. Efficient unsupervised learning for localization and detection in object categories. In *Advances in Neural Information Processing Systems (NIPS)*, 2005.

[94] D. G. Lowe. Object recognition from local scale-invariant features. In *International Conference on Computer Vision (ICCV)*, pages 1150–1157, 1999.

[95] D.G. Lowe. Three-dimensional object recognition from single twodimensional images. *Artificial Intelligence*, 31(3):355–395, 1987.

[96] D.G. Lowe. Similarity metric learning for variable kernel classifier. *Neural computation*, 7(1):72–85, 1995.

[97] X. Ma and W. E. L. Grimson. Edge-based rich representation for vehicle classification. In *International Conference on Computer Vision (ICCV)*, 2005.

[98] S. Mahamud and M. Hebert. Minimum risk distance measure for object recognition. In *International Conference on Computer Vision (ICCV)*, 2003.

[99] S. Mahamud and M. Hebert. The optimal distance measure for object detection. In *Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages I: 248–255, 2003.

[100] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. In *Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2003.

[101] E. Miller, N. Matsakis, and P. Viola. Learning from one example through shared densities on transforms. In *Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 464–471, 2000.

[102] T. Minka and R. Picard. *Learning how to learn is learning with point sets*. Unpublished manuscript. Available at http://wwwwhite.media.mit.edu/ tpminka/papers/learning.html., 1997.

[103] B. Moghaddam and M. Yang. Gender classification with support vector machines. In *IEEE Intl. Conf. on Automatic Face and Gesture Recognition*, pages 306–311, 2000.

[104] K.P. Murphy, A. Torralba, and W. T. Freeman. Using the forest to see the trees: a graphical model relating features, objects and scenes. In *Advances in Neural Information Processing Systems (NIPS)*, 2003.

[105] J. Mutch and D. G. Lowe. Multiclass object recognition with sparse, localized features. *Conf. on Computer Vision and Pattern Recognition (CVPR)*, 1:11–18, 2006.

[106] A. Needham. Object recognition and object segregation in 4.5-month-old infants. *Journal of experimental child psychology*, 78:3–24, 2001.

[107] A. Needham and R. Baillargeon. Effects of prior experience in 4.5-months-old infants' object segregation. *Infant behavior and development*, 21:1–24, 1998.

[108] B. Ommer and J. M. Buhmann. Learning compositional categorization models. In *European Conf. on Computer Vision (ECCV)*, 2006.

[109] C. S. Ong, A. J. Smola, and R. C. Williamson. Superkernels. In *Advances in Neural Information Processing Systems (NIPS)*, 2002.

[110] A. Opelt, M. Fussenegger, A. Pinz, and P. Auer. Weak hypotheses and boosting for generic object detection and recognition. In *European Conf. on Computer Vision (ECCV)*, 2004.

[111] O. C. Ozcanli, A. Tamrakar, B. B. Kimia, and J. L. Mundy. Augmenting shape with appearance in vehicle category recognition. In *Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2006.

[112] M.A. Peterson and B.S. Gibson. shape recognition contributions to figure-ground organization in three dimensional displays. *Cognitive Psychology*, 25:383–429, 1993.

[113] P. J. Phillips. Support vector machines applied to face recognition. In *Advances in Neural Information Processing Systems (NIPS)*, volume 11. MIT press, 1999.

[114] J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1:81–106, 1986.

[115] R. Rosales and G. Fung. Learning sparse metrics via linear programming. In *Conf. on Knowledge Discovery and Data Mining (KDD)*, pages 367–373, New York, NY, USA, 2006. ACM Press.

[116] E. Sali and S. Ullman. Combining class specific fragments for object classification. In *British Machine Vision Conference (BMVC)*, volume 1, pages 203–213, 1999.

[117] G. Saon, M. Padmanabhan, R. Gopinath, and S. Chen. Maximum likelihood discriminant feature spaces. In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2000.

[118] B. Schlkopf and A. Smola. *Learning with Kernels*. MIT Press, Cambridge, MA, 2002.

[119] C. Schmid, R. Mohr, and C. Bauckhage. Comparing and evaluating interest points. In *International Conference on Computer Vision (ICCV)*, 1998.

[120] M. Schultz and T. Joachim. Learning a distance metric from relative comparisons. In *Advances in Neural Information Processing Systems (NIPS)*, 2003.

[121] T. B. Sebastian, P. N. Klein, and B. B. Kimia. Recognition of shapes by editing shock graphs. In *International Conference on Computer Vision (ICCV)*, pages 755–762, 2001.

[122] A. Selinger and R. C. Nelson. A perceptual grouping hierarchy for appearance-based 3d object recognition. *Computer Vision and Image Understanding (CVIU)*, 76(1):83–92, 1999.

[123] T. Serre, L. Wolf, and T. Poggio. A new biologically motivated framework for robust object recognition. In *Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2005.

[124] A. Sha'ashua and S. Ullman. Structural saliency: The detection of globally salient structures using a locally connected network. In *International Conference on Computer Vision (ICCV)*, pages 321–327, 1988.

[125] S. Shalev-Shwartz, Y. Singer, and A. Y. Ng. Online and batch learning of pseudo-metrics. In *International Conference on Machine Learning (ICML)*, 2004.

[126] N. Shental, T. Hertz, A. Bar-Hilel, and D. Weinshall. Computing gaussian mixture models with EM using equivalence constraints. In *ICML workshop: The continuum from labeled to unlabeled data in machine learning and data mining*, 2003.

[127] N. Shental, T. Hertz, D. Weinshall, and M. Pavel. Adjustment learning and relevant component analysis. In *European Conf. on Computer Vision (ECCV)*, volume 4, 2002.

[128] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 22(8):888–905, 2000.

[129] R. D. Short and K. Fukunaga. The optimal distance measure for nearest neighbor classification. *IEEE Transactions on Information Theory*, 27(5):622–627, 1981.

[130] P. Simard, Y. Le Cun, J. Denker, and B. Victorri. Transformation invariance in pattern recognition — tangent distance and tangent propagation. *Lecture Notes in Computer Science*, 1524:239–274, 1998.

[131] N. Srebro and S. Ben-David. Learning bounds for support vector machines with learned kernels. In *Annual Conference On Learning Theory (COLT)*, 2006.

[132] E. Sudderth, A. Torralba, W. Freeman, and A. Willsky. Learning hierarchical models of scenes, objects, and parts. In *International Conference on Computer Vision (ICCV)*, 2005.

[133] K. K. Sung and T. Poggio. Example-based learning for view-based human face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 20(1):39–51, 1998.

[134] A. Thayananthan, B. Stenger, P. Torr, and R. Cipolla. Shape context and chamfer matching in cluttered scenes. In *Conf. on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 127–133, 2003.

[135] S. Thrun and L.Y. Pratt. *Learning To Learn*. Kluwer Academic Publishers, Boston, MA, 1998.

[136] A. Torralba, K. P. Murphy, and W. T. Freeman. Contextual models for object detection using boosted random fields. In *Advances in Neural Information Processing Systems (NIPS)*, 2004.

[137] A. Torralba, K. P. Murphy, and W. T. Freeman. Sharing features: efficient boosting procedures for multiclass object detection. In *Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2004.

[138] F. C. Tsai. *A Probabilistic Approach to Geometric Hashing Using Line Features*. PH.D thesis, technical Report 640, Robotics Research Laboratory, N.Y.U., 1993.

[139] I. W. Tsang, P. M. Cheung, and J. T. Kwok. Kernel relevant component analysis for distance metric learning. In *IEEE International Joint Conference on Neural Networks (IJCNN)*, pages 954–959, 2005.

[140] M. Turk and A.P. Pentland. Eigenfaces for recognition. *Cogneuro*, 3(1):71–96, 1991.

[141] A. Tversky. Features of similarity. *Psychological review*, 84(4):327–352, 1977.

[142] S. Ullman and R. Basri. Recognition by linear combinations of models. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 13(10):992–1006, 1991.

[143] S. Ullman, M. Vidal-Naquet, and E. Sali. Visual features of intermediate complexity and their use in classification. *Nature Neuroscience*, 5:682–687, 2002.

[144] V.N. Vapnik. *Statistical learning theory*. John wiley and sons, 1998.

[145] M. Vidal-Naquet and S. Ullman. Object recognition with informative features and linear classification. In *International Conference on Computer Vision (ICCV)*, 2003.

[146] P. Vincent and Y. Bengio. K-local hyperplane and convex distance nearest neighbor algorithms. In *Advances in Neural Information Processing Systems (NIPS)*, volume 14, 2002.

[147] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2001.

[148] K. Wagstaff, C. Cardie, S. Rogers, and S. Schroedl. Constrained K-means clustering with background knowledge. In *International Conference on Machine Learning (ICML)*, pages 577–584. Morgan Kaufmann, San Francisco, CA, 2001.

[149] M. Weber, M. Welling, and P. Perona. Unsupervised learning of models for recognition. In *European Conf. on Computer Vision (ECCV)*, 2000.

[150] K. Q. Weinberger, J. Blitzer, and L. K. Saul. Distance metric learning for large margin nearest neighbor classification. In *Advances in Neural Information Processing Systems (NIPS)*, volume 18. MIT Press: Cambridge, MA, 2006.

[151] D. Wettschereck, D. Aha, and T. Mohri. A review and empirical evaluation of feature weighting methods for a class of lazy learning algorithms. *Artificail intelligence review*, 11:273–314, 1997.

[152] L. wolf. *Learning using the Born rule*. Technical report MIT-CSail-TR-2006-036, 2006.

[153] L. Wolf and S. Bileschi. A critical view of context. *International Journal of computer vision (IJCV)*, 2006.

[154] L. Wolf and A. Shashua. Learning over sets using kernel principal angles. *Journal of Machine Learning Research*, 4(10):913–931, 2003.

[155] H. J. Wolfson. Model-based object recognition by geometric hashing. In *European Conf. on Computer Vision (ECCV)*, pages 526–536, 1990.

[156] F. Wu, Y. Zhou, and C. Zhang. Self-enhanced relevant component analysis with side-information and unlabeled data. In *IEEE International Joint Conference on Neural Networks (IJCNN)*, volume 2, pages 1347–1351, 2004.

[157] G. Wu, E.Y. Chang, and N. Panda. Formulating distance functions via the kernel trick. In *Conf. on Knowledge Discovery and Data Mining (KDD)*, pages 703–709, 2005.

[158] E.P Xing, A.Y. Ng, M.I. Jordan, and S. Russell. Distance metric learnign with application to clustering with side-information. In *Advances in Neural Information Processing Systems (NIPS)*, volume 15. The MIT Press, 2002.

[159] R. Yan, J. Zhang, J. Yang, and A. Hauptmann. A discriminative learning framework with pairwise constraints for video object classification. *Conf. on Computer Vision and Pattern Recognition (CVPR)*, 02:284–291, 2004.

[160] D. Y. Yeung and H. Chang. Extending the relevant component analysis algorithm for metric learning using both positive and negative equivalence constraints. *Pattern Recognition*, 39(5):1007–1010, 2006.

[161] H. Zhang, A. C. Berg, M. Maire, and J. Malik. Svm-knn: Discriminative nearest neighbor classification for visual category recognition. In *Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 2126–2136. IEEE Computer Society, 2006.

[162] W. Zhang, Y. Bing, G. J. Zelinsky, and D. Samaras. Object class recognition using multiple layer boosting with heterogeneous features. In *Conf. on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 323–330, 2005.

[163] Z. Zhang, J. T. Kwok, and D. Y. Yeung. Parametric distance metric learning with label information. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1450–1452, 2003.

[164] Z. Zhang, J.T. Kwok, and D. Y. Yeung. Model-based transductive learning of the kernel matrix. *Machine Learning*, 63(1):69–101, April 2006.

[165] S. Zhu and A. Yuille. Forms: A flexible object recognition and modeling system. *International Journal of computer vision (IJCV)*, 20(3):187–212, 1996.

תוכן העניינים

טלאולוגי של הרמות המופשטות יותר ואפיון של תת-מחלקות ע"י פרטים דקים יחסית, שבד"כ אינם חלקים אלא תיאורי חלקים. בהתאמה אנו מציעים ללמוד מודל מבוסס חלקים של המחלקה הבסיסית, ולהשתמש במודל על מנת לבנות ייצוג המכיל את תיאורי החלקים שזוהו. ייצוג זה, שהוא ווקטור סדור המכיל פרטים עם סמנטיקה גבוהה יחסית משמש בשלב השני להפרדה בין תת-מחלקות של המחלקה הבסיסית. מודל המחלקה הבסיסית נלמד מתמונות ממספר תת-מחלקות באמצעות האלגוריתם שהוצג במאמר הקודם בתזה זו. את ההבחנה בשלב השני אפשר לעשות באמצעים סטנדרטיים של למידת מכונה, ואנו משתמשים במכונת וקטורים תומכים (SVM) ליניארית. בניגוד לעבודות קודמות בנושאים דומים, שנבחנו רק בתמונות של פרצופים או מכוניות, אנו בוחנים אמפירית את השיטה המוצעת על פני מגוון רחב של תת-מחלקות מ- 6 קטגוריות בסיסיות. דיוק הסיווג המתקבל בשיטה המוצעת טוב מזה המושג בשיטות מקבילות בהן נלמד מודל נפרד עבור כל תת-מחלקה, ודיוק זה מושג ביעילות יחסית, שכן יש ללמוד פחות מודלים. התרומה המרכזית כאן היא בהצעת המכניזם ובהדגמה התועלת של העברת מידע בין משימות למידה על סמך ההיררכיה המושגית הטבעית.

4. <u>פרקי סיום:</u>

בפרק הסיום של העבודה מופיע סיכום קצר, בהיקף של עמוד וחצי, המתאר את התרומות העיקריות של העבודה. לאחר מכן מופיעים שני נספחים, המכילים השלמת הוכחות שנדחו מן הטקסט העיקרי מטעמי שיפור הקריאות. נספח A מכיל השלמת 3 הוכחות שנדחו מפרק 2.1 , המכיל את המאמר התיאורטי "למידה עם אילוצי שקילות, ויחסיה עם סיווג רב-תיוגים". נספח B מכיל הוכחה של הקשר בין Gaussian coding similarity, היא מידת הדומי המוצעת בפרק 2.3 , לטכניקה ותיקה ומוכרת להורדת מימד - Fisher Linear Discriminant (FLD), שהוצעה לראשונה כבר ב-1936. אנו מראים בנספח זה ש- FLD היא ההטלה הממקסמת את מרווח השוליים (margin) הממוצע המוגדר באמצעות Gaussian coding similarity. ההוכחות בשני הנספחים מופיעות כאו לראשונה (הן הופיעו לפני כן בדוחות טכניים בלבד). לאחר הנספחים מופיע הביבליוגרפיה של פרק המבוא. הביבליוגרפיות של המאמרים השונים נשמרו כחטיבה אחת עימם.

"למידה יעילה של מודלי אובייקטים הכוללים יחסים" - העבודה מציעה אלגוריתמים ללמידה של זיהוי אובייקטים מתמונות לא מעובדות, המיוצגות באמצעות קבוצות של פיסות תמונה. הכלי המרכזי לשיפור הייצוג הוא מודל אובייקט גנרטיבי, המאפשר זיהוי של חלקי אובייקט, ולצורך זה נבדקים הן מודלי "שק חלקים" והן מודלים הכוללים יחסים מרחביים. רעיון טכני מרכזי בעבודה הוא אופטימיזציה דיסקרימינטיבית של המודל הגנרטיבי, המבוצעת באמצעות הרחבה של טכניקת ה- boosting. תרומתו של שילוב טכני זה באה לידי ביטוי בעיקר בלמידה של מודלים הכוללים יחסים מרחביים. אנו מציגים ניתוח המראה כי למידה של מודלים כאילו באמצעות טכניקת maximum likelihood המסורתית היא בעיתית (בהקשר של למידה מקבוצת פיסות תמונה), כיון שהיא נקלעת לדילמה שקרניה הן סיבוכיות למידה מעריכית מחד וחלקים החוזרים על עצמם במודל האופטימאלי מאידך. האופטימיזציה הדיסקרימינטיבית שאנו מציעים מאפשרת למידה יעילה של מודל בלי חזרת חלקים על עצמם, כאשר הסיבוכיות היא ליניארית במספר חלקי המודל ובמספר פיסות התמונה בייצוג ההתחלתי. בכך אנו מציעים פתרון נוח לבעיה המופיעה במספר עבודות מחקר עכשוויות.
האופטימיזציה הדיסקרימינטיבית באלגוריתם המוצע מכלילה את טכניקת ה- boosting באופן המאפשר תלות בין ההיפותזות החלשות המאוגדות במסווג. כל היפותזה כזו, המתארת חלק של האובייקט תלויה בצורת החלק, כמו גם במיקומו וגודלו יחסית למיקום והגודל המשוערים של האובייקט. תהליך הלמידה מבוסס על למידת חלק, הסקה מחדש של מיקום האובייקט וגודלו בכל תמונות האימון ע"י פעפוע אמונות (belief propagation), וחוזר חלילה. ניסויים אמפיריים מראים יתרון של השיטה המוצעת על פני למידה של מודל גנרטיבי בשיטות המסורתיות, בעיקר בשל היכולת ללמוד מודל בעל עשרות חלקים בשיטה המוצעת, לעומת חלקים ספורים בשיטות קודמות. ניסויים נוספים עם מאגר תמונות של מכוניות, הנכללים בתזה זו לראשונה, מראים כי המודלים הנלמדים יכולים לשמש למציאת מיקומו המדויק של אובייקט.

"זיהוי של תת-מחלקות של אובייקטים ויזואליים באמצעות מודלי אובייקט הכוללים יחסים" – אנו מציעים שיטה דו-שלבית לזיהוי תת-מחלקות של אובייקטים. ההשראה לשיטה מגיעה משני רעיונות שמקורם בפסיכולוגיה קוגניטיבית. ראשית, קיומה של רמה במרכז היררכית המושגים שלנו שהיא קודמת לרמות אחרות מבחינת זיהוי ולמידת אובייקטים. רמה זו נקראת הרמה הבסיסית (the basic level), והיא כוללת מושגים כמו "מכונית" או "אופנוע", בניגוד ל"כלי רכב" המופשט (super-ordinate), או "מכונית משפחתית" ו"אופנוע שטח" שהם תת-מחלקות (sub-ordinate). התצפית השנייה מצביעה על היות מושגי הרמה הבסיסית מאופיינים באמצעות חלקים וקונפיגורצית חלקים, בניגוד לאפיון

כיצד פונקצית מרחק בעלת טעות קטנה יכולה לשמש לצברור ובניית מסווגים עם טעות קטנה (ליניארית בגודל הטעות של פונקצית המרחק).

"למידת מרחק מהלנוביס באמצעות אילוצי שקילות" – אנו מציגים אלגוריתם פשוט ויעיל (הנקרא relevant component analysis (RCA)) ללמידת מטריקת מהלנוביס מאילוצים חיוביים בלבד, המשפר ביצועי צברור ואחזור מידע. האלגוריתם הוא אופטימאלי לפי מספר קריטריונים כולל שמירה של אינפורמציה משותפת בין הייצוג ההתחלתי והסופי, מזעור המרחק הממוצע בין קלטים דומים, ומידול גנרטיבי תחת הנחות גאוסיניות. נידונים גם הורדת מימד רלוונטית וגרסא on-line של האלגוריתם. בחינה אמפירית מראה כי האלגוריתם משפר ביצועי צברור באמצעות אלגוריתמי K-means ו- EM, ואחזור מידע כולל שיפור גדול בבעיית אחזור פרצופים. יתרונותיו המרכזי של האלגוריתם הם בפשטותו הרעיונית וביעילותו החישובית לעומת מתחריו.

"למידה של פונקצית מרחק באמצעות דומי מבוסס קידוד" - במחקר זה אנו מגדירים את מושג ה-"דומי" ומטרותיו באמצעות מונחים מתורת האינפורמציה. הדומי בין שני קלטים מזוהה עם רווח באורך הקידוד הנוצר במעבר מקידוד שלהם כבלתי תלויים לקידוד משותף. בעוד ההגדרה היא כללית מאוד, אנו מציעים אלגוריתם למידה פשוט להערכת פונקצית הדומי מאילוצי שקילות חיוביים בלבד, תחת הנחות גאוסיניות פשוטות. המרחק המתקבל אינו מטרי, אך הוא ניתן לפירוש כמבחן יחס ניראות (Likelihood Ratio Test), ויש לו קשרים עמוקים עם מטריקת ה-RCA ועם הורדת מימד באמצעות FLD. ניסויים אמפיריים מראים יתרון של פונקצית הדומי המוצעת על פני RCA וטכניקה נוספת בצברור מבוסס גרפים (graph based clustering) וכן יכולת הכללה בין משימות בבעיית אחזור פרצופים.

3. <u>מחקרים בנושא זיהוי מחלקות אובייקטים ויזואליים:</u>

פרק זה כולל שני מחקרים. הראשון עוסק בלמידת זיהוי של מחלקות אובייקטים (כגון אופנועים או פרצופים) באמצעות מודלים גנרטיביים. מחקר זה התפרסם במקור בשני כנסים, CVPR 2005 ו- ICCV 2005 . כאן אני מביא גרסא מאוחדת של מחקרים אילו, הכוללת תוצאות תיאורטיות ואמפיריות נוספות. המחקר השני נוגע בהבחנה בין תת-מחלקות (כגון בין אופנועי שטח ואופנועי כביש), והוא התקבל לפרסום בכנס NIPS 2006.

בשנים האחרונות מתרכז חלק ניכר מהמחקר בלמידת זיהוי של מחלקות אובייקטים, וזאת על סמך תמונות לא מעובדות. הייצוג ההתחלתי של התמונה הוא בד"כ כקבוצה לא סדורה של פיסות תמונה (patches) ממיקומים וגדלים שונים. בהקשר זה המחלוקת בנושאי ההקשר מתבטאת בהבדל בין שני סוגי ייצוגי אובייקט: מודל "שק חלקים" (bag of features) ומודל הכולל יחסים. מודלים מן הסוג הראשון מתארים אובייקט כאוסף של חלקים ומפרטים כיצד נראה כל חלק בנפרד. מודלים מן הסוג השני כוללים תיאור זה, ומוסיפים עליו תיאור של מיקום החלקים אחד ביחס למשנהו, ובכך הם מאפשרים משוב בין קביעת זהותו של חלק אחד לקביעת זהותו של האחר, ובין זיהוי האובייקט לזיהוי חלקיו באופן כללי.

תרומתי המחקרית ניתנת בהקשר מחקר זה, ואני מתאר אותו בהרחבה בפרק 1.3.3. לאחר מכן אני סוקר ספרות הנוגעת ללמידה של מחלקות אובייקטים דומים ולהעברת יידע בין משימות למידת זיהוי אובייקט בכלל. זוהי ספרות הרלוונטית בעיקר למחקר [E] המתואר בפרק 3.2.

2. <u>מחקרים בנושא למידת פונקציות מרחק:</u>

פרק זה מציג שלושה מחקרים. הראשון הוא מחקר תיאורטי, שהתפרסם בכנס COLT 2003, וענינו השקילות בין למידת מסווגים ולמידת פונקציות מרחק בינאריות. השני מציג אלגוריתם ללמידת מטריקת מהלנוביס (Mahalanobis) מאילוצי שקילות, והוא פורסם בעיתון JMLR ב- 2005 . השלישי מציע אלגוריתם ללמידת פונקצית מרחק שאינה מטריקה, והוא מופיע פה לראשונה. להלן אפרט מעט על תוכנם של המאמרים.

"למידה עם אילוצי שקילות, ויחסיה עם סיווג רב-תיוגים" – במחקר זה אנו שואלים באיזו מידה למידת פונקצית מרחק, לפחות באופן עקרוני, יכולה להחליף לחלוטין למידת מסווגים. אנו עוסקים בפונקציות מרחק בינאריות המחזירות "1" עבור זוג נקודות שהן מאותו הסוג, ו-"0" עבור זוגות של נקודות שהן מסוגים שונים. בעיית למידת סיווג ניתנת להצגה באופן טבעי כבעיית למידת פונקצית מרחק מסוג זה. אנו מאפיינים את הקשרים בין שתי הבעיות: בין טעויות הסיווג בשני המרחבים, ובין גדלי המדגמים הנדרשים ללמידה של שתיהן. בפרט קשרים אילו מוכיחים כי מחלקת מושגים היא למידה (learnable) אם ורק אם מחלקת פונקציות המרחק המקבילה היא למידה. בנוסף, אנו מראים אלגוריתמית

ישירות רק על מרחקים בין קלטים, אפשר לשפר את הייצוג על ידי למידת פונקצית מרחק טובה, אשר מציינת קלטים עם אותו התיוג כקרובים וקלטים עם תיוגים שונים כרחוקים זה מזה. בכל מקרה, נשאלת השאלה העקרונית אם למידת הייצוג צריכה להיעשות בנפרד, כתהליך מקדים ללמידת המסווג (או הצברור), או שמא עדיפה אופטימיזציה משותפת, בה הלמידה מייצוג חלש מתבצעת בהליך יחיד הכולל גם שיפור הייצוג במידת הצורך. שאלה זו נידונה בפרק 1.1.3 ויש בה פנים לכאן ולכאן. באופן כללי המחקר בעבודה זו מתבצע בגישה הראשונה, של למידת ייצוג נפרדת, למעט האלגוריתמים המוצעים ללמידה של עצמים מהרמה הבסיסית (basic level) בפרק 3.1.

הספרות הרלוונטית ללמידת פונקציות מרחק נידונה בפרק 1.2 של העבודה. ראשית נידונים מושגים קרובים כמטריקה (metric), מכפלה פנימית וגרעין(kernel) ודומי (similarity) וטכניקות קרובות כסינון מאפיינים (feature selection) והטלות ליניאריות. בהמשך אני מציג את הספרות ללמידת מרחק בצורה מפוקחת, הכוללת למידת מטריקה לתיוג על סמך שכנים קרובים (nearest neighbor classification) ולמידת גרעין עבור מכונות וקטורים תומכים (SVM). עיקר עבודתי במהלך הדוקטורט נוגע ללמידת פונקצית מרחק בצורה מפוקחת למחצה, כאשר הפיקוח ניתן באמצעות אילוצי שקילות על זוגות קלטים המציינים אם שני הקלטים הם מאותו הסוג או לא. אילוצים כאילו ניתנים להפקה בצורה אוטומטית בנסיבות מסוימות, בהם הקלטים מגיעים ממקור מרקובי כגון סרט וידאו. בנסיבות אחרות הם ניתנים להשגה במאמץ אנושי קטן יחסית. השימוש בהם ללמידת פונקציות מרחק הוא טבעי מאוד. אני סוקר את המוטיבציה לשימוש באילוצים אילו ואת הספרות הנוגעת לשימוש בהם בפרק 1.2.3 .

בפרק 1.3 אני סוקר את הספרות הרלוונטית לחלקה השני של העבודה, העוסק בזיהוי אובייקטים ויזואליים. אני מתחיל בסקירה היסטורית של גישות שהוצעו לבעיית זיהוי האובייקט, היא בעיה ותיקה הנחקרת עשרות שנים, וממשיך בדיון תמאטי על תפקידם של הקשר ומשוב בזיהוי אובייקט ע"י אדם ומכונה. הפרדיגמה השלטת כיום בלמידת מכונה רואה במציאת הייצוג (feature extraction) תהליך קודם להפעלת המסווג, המשתמש בייצוג על מנת לקבוע את התיוג. תמונה זו אינה הולמת עדויות מצטברות על זיהוי אובייקטים בבני אדם, המצביעות על קיום תהליך משוב בו השערת הסיווג משפיעה על בחירת ייצוגי הביניים ולהפך. רעיונות כאילו, של משוב בין זיהוי וסגמנטציה ובין זיהוי חלקי האובייקט לזיהוי האובייקט ומיקומו הם רעיונות ותיקים בראיה ממוחשבת, אך יישובם עם הטכניקות הקיימות ללמידת מכונה קשה ומצריך מחקר ניכר. לעתים קרובות פשוט יותר ללמוד זיהוי אובייקט בפרדיגמה נטולת המשוב, שהיא פשוטה ומבוססת יותר, והמחלוקת בנושאים אילו רחוקה מסיומה. עמדתי נוטה יותר לכיוון מחייבי הדגש על הקשר ומשוב.

תקציר העבודה

תרומתה המדעית של עבודה זו מפוצלת בין שתי בעיות בהן מתבצעת למידה מייצוג התחלתי חלש ובאמצעות פיקוח חלקי. הבעיה הראשונה היא למידה של פונקציות מרחק, על מנת לשפר את ביצועיהם של אלגוריתמים מבוססי מרחק ללמידה מפוקחת, צברור (clustering) ואחזור מידע. הבעיה השנייה היא למידת מסווגים (classifiers) של קטגוריות עצמים ויזואליות, כאשר הקלט מורכב מתמונות לא מעובדות. בבעיה הראשונה אנו משפרים את הייצוג באמצעות שיפור פונקצית המרחק, בעוד בשנייה הכלי המרכזי המשמש אותנו בעבודה זו הוא מודל אובייקט גנרטיבי.

העבודה מבוססת על חמישה מאמרים שפורסמו ועל דו"ח טכני אחד. פרק הפתיחה דן באספקט המאחד של העבודה, הוא למידה ושיפר של ייצוגים חלשים, ולאחר מכן סוקר את הספרות המדעית בשני תחומי המחקר של למידת פונקציות מרחק וזיהוי אובייקטים ויזואליים. פרקים 2,3 הם עיקר העבודה כאשר פרק 2 מכיל את המחקרים הנוגעים בלמידת מרחק ופרק 3 את הפרסומים שעניינם זיהוי מחלקות אובייקטים ויזואליים. בסוף העבודה מופיעים סיכום קצר של התרומות העיקריות, נספחים המשלימים הוכחות המופיעות במאמרים וביבליוגרפיה. להלן אתן תיאור קצר של התכנים בפרקי העבודה השונים.

1. <u>מבוא</u>

למידת מכונה כיום מאפשרת במקרים רבים פתרון של בעיות סיווג ביעילות ודיוק מרשימים. עם זאת, שימוש בכלי הסיווג הקיימים מחייב ייצוג טוב של הקלט לאלגוריתם הלמידה, בו הקשר בין נתוני הקלט והסיווג הנדרש אינו מורכב מדי. בפועל המידע בבעיות מעשיות רבות אינו מיוצג היטב, והפעלה של אלגוריתמי למידה מחייבת עבודה מקדימה רבה של מומחה הלמידה האנושי על מנת לשפר את הייצוג. כך לעתים קרובות נראה שעיקר הלמידה מתבצע ע"י המומחה האנושי ולא ע"י המכונה. אוטומציה של תהליך שיפור הייצוג היא לפיכך אתגר קשה ובעל ערך מעשי רב. אוטומציה כזו כרוכה בחלקה בהכללה על פני קבוצת בעיות למידה, בניגוד להכללה על פני קלטים המתבצעת בבעיית למידה בודדת. הכללה כזו נראית כיום כעומדת במוקד מותר האדם על המכונה.

למידת ייצוג היא למידה של טרנספורמציה המעבירה את הייצוג ההתחלתי (ה"חלש") לייצוג טוב יותר. אפשרות אחת היא ללמוד טרנספורמציה הפועלת על הקלטים עצמם, אך קיימת גם אפשרות אחרת. היות ואלגוריתמי למידה רבים מסתמכים

תודות

יופיעו בעותק הסופי.

עבודה זו נעשתה בהדרכתה של פרופסור דפנה ויינשל

למידה מייצוגים חלשים

באמצעות

פונקציות מרחק ומודלים גנרטיביים

חיבור לשם קבלת תואר דוקטור לפילוסופיה

מאת

אהרון בר הלל

הוגש לסנאט האוניברסיטה העברית בשנת תשס"ז (2006)