Journal of Machine Learning Research 8 (2007) 2265-2295 Submitted 6/06; Revised 5/07; Published 9/07

Euclidean Embedding of Co-occurrence Data

Amir Globerson* GAMIR@CSAIL.MIT.EDU
Computer Science and Artificial Intelligence Laboratory
Massachusetts Institute of Technology
Cambridge, MA 02139

Gal Chechik* GAL@AI.STANFORD.EDU
Department of Computer Science
Stanford University
Stanford CA, 94306

Fernando Pereira PEREIRA@CIS.UPENN.EDU
Department of Computer and Information Science
University of Pennsylvania
Philadelphia PA, 19104

Naftali Tishby TISHBY@CS.HUJI.AC.IL
School of Computer Science and Engineering and
The Interdisciplinary Center for Neural Computation
The Hebrew University of Jerusalem
Givat Ram, Jerusalem 91904, Israel

Editor: John Lafferty

Abstract

Embedding algorithms search for a low dimensional continuous representation of data, but most algorithms only handle objects of a single type for which pairwise distances are specified. This paper describes a method for embedding objects of different types, such as images and text, into a single common Euclidean space, based on their co-occurrence statistics. The joint distributions are modeled as exponentials of Euclidean distances in the low-dimensional embedding space, which links the problem to convex optimization over positive semidefinite matrices. The local structure of the embedding corresponds to the statistical correlations via random walks in the Euclidean space. We quantify the performance of our method on two text data sets, and show that it consistently and significantly outperforms standard methods of statistical correspondence modeling, such as multidimensional scaling, IsoMap and correspondence analysis.

Keywords: embedding algorithms, manifold learning, exponential families, multidimensional scaling, matrix factorization, semidefinite programming

1. Introduction

Embeddings of objects in a low-dimensional space are an important tool in unsupervised learning and in preprocessing data for supervised learning algorithms. They are especially valuable for exploratory data analysis and visualization by providing easily interpretable representations of the

∗. Both authors contributed equally. Gal Chechik's current address is Google Inc., 1600 Amphitheatre Parkway, Mountain View, CA, 94043.

relationships among objects. Most current embedding techniques build low dimensional mappings that preserve certain relationships among objects. The methods differ in the relationships they choose to preserve, which range from pairwise distances in multidimensional scaling (MDS) (Cox and Cox, 1984) to neighborhood structure in locally linear embedding (Roweis and Saul, 2000) and geodesic structure in IsoMap (Tenenbaum et al., 2000). All these methods operate on objects of a single type endowed with a measure of similarity or dissimilarity.

However, embedding should not be confined to objects of a single type. Instead, it may involve different types of objects provided that those types share semantic attributes. For instance, images and words are syntactically very different, but they can be associated through their meanings. A joint embedding of different object types could therefore be useful when instances are mapped based on their semantic similarity. Once a joint embedding is achieved, it also naturally defines a measure of similarity between objects of the same type. For instance, joint embedding of images and words induces a distance measure between images that captures their semantic similarity.

Heterogeneous objects with a common similarity measure arise in many fields. For example, modern Web pages contain varied data types including text, diagrams and images, and links to other complex objects and multimedia. The objects of different types on a given page have often related meanings, which is the reason they can be found together in the first place. In biology, genes and their protein products are often characterized at multiple levels including mRNA expression levels, structural protein domains, phylogenetic profiles and cellular location. All these can often be related through common functional processes. These processes could be localized to a specific cellular compartment, activate a given subset of genes, or use a subset of protein domains. In this case the specific biological process provides a common "meaning" for several different types of data.

A key difficulty in constructing joint embeddings of heterogeneous objects is to obtain a good similarity measure. Embedding algorithms often use Euclidean distances in some feature space as a measure of similarity. However, with heterogeneous object types, objects of different types may have very different representations (such as categorical variables for some and continuous vectors for others), making this approach infeasible.

The current paper addresses these problems by using object co-occurrence statistics as a source of information about similarity. We name our method *Co-occurrence Data Embedding*, or **CODE**. The key idea is that objects which co-occur frequently are likely to have related semantics. For example, images of dogs are likely to be found in pages that contain words like $\{$*dog, canine, bark*$\}$, reflecting a common underlying semantic class. Co-occurrence data may be related to the geometry of an underlying map in several ways. First, one can simply regard co-occurrence rates as approximating pairwise distances, since rates are non-negative and can be used as input to standard metric-based embedding algorithms. However, since co-occurrence rates do not satisfy metric constraints, interpreting them as distances is quite unnatural, leading to relatively poor results as shown in our experiments.

Here we take a different approach that is more directly related to the statistical nature of the co-occurrence data. We treat the observed object pairs as drawn from a joint distribution that is determined by the underlying low-dimensional map. The distribution is constructed such that a pair of objects that are embedded as two nearby points in the map have a higher statistical interaction than a pair that is embedded as two distant points. Specifically, we transform distances into probabilities in a way that decays exponentially with distance. This exponential form maps sums of distances

into products of probabilities, supporting a generative interpretation of the model as a random walk in the low-dimensional space.

Given empirical co-occurrence counts, we seek embeddings that maximize the likelihood of the observed data. The log-likelihood in this case is a non-concave function, and we describe and evaluate two approaches for maximizing it. One approach is to use a standard conjugate gradient ascent algorithm to find a local optimum. Another approach is to approximate the likelihood maximization using a convex optimization problem, where a convex non-linear function is minimized over the cone of semidefinite matrices. This relaxation is shown to yield similar empirical results to the gradient based method.

We apply CODE to several heterogeneous embedding problems. First, we consider joint embeddings of two object types, namely *words-documents* and *words-authors* in data sets of documents. We next show how CODE can be extended to jointly embed more than two objects, as demonstrated by jointly embedding words, documents, and authors into a single map. We also obtain quantitative measures of performance by testing the degree to which the embedding captures *ground-truth* structures in the data. We use these measures to compare CODE to other embedding algorithms, and find that it consistently and significantly outperforms other methods.

An earlier version of this work was described by Globerson et al. (2005).

2. Problem Formulation

Let X and Y be two categorical variables with finite cardinalities $|X|$ and $|Y|$. We observe a set of pairs $\{x_i, y_i\}_{i=1}^n$ drawn IID from the joint distribution of X and Y. The sample is summarized via its empirical distribution[1] $\bar{p}(x,y)$, which we wish to use for learning about the underlying unknown joint distribution of X and Y. In this paper, we consider models of the unknown distribution that rely on a joint embedding of the two variables. Formally, this embedding is specified by two functions $\phi : X \to \mathbb{R}^q$ and $\psi : Y \to \mathbb{R}^q$ that map both categorical variables into the common low dimensional space $\mathbb{R}^q$, as illustrated in Figure 1.

The goal of a joint embedding is to find a geometry that reflects well the statistical relationship between the variables. To do this, we model the observed pairs as a sample from the parametric distribution $p(x,y;\phi,\psi)$, abbreviated $p(x,y)$ when the parameters are clear from the context. Thus, our models relate the probability $p(x,y)$ of a pair (x,y) to the embedding locations $\phi(x)$ and $\psi(y)$.

In this work, we focus on the special case in which the model distribution depends on the squared Euclidean distance $d_{x,y}^2$ between the embedding points $\phi(x)$ and $\psi(y)$:

$$d_{x,y}^2 \quad = \quad \|\phi(x) - \psi(y)\|^2 = \sum_{k=1}^{q} (\phi_k(x) - \psi_k(y))^2 \ .$$

Specifically, we consider models where the probability $p(x,y)$ is proportional to $e^{-d_{x,y}^2}$, up to additional factors described in detail below. This reflects the intuition that closer objects should co-occur more frequently than distant objects. However, a major complication of embedding models is that the embedding locations $\phi(x)$ and $\psi(y)$ should be insensitive to the marginals $p(x) = \sum_y p(x,y)$ and $p(y) = \sum_x p(x,y)$. To see why, consider a value $x \in X$ with a low marginal probability $p(x) \ll 1$, which implies a low $p(x,y)$ for all y. In a model where $p(x,y)$ is proportional to $e^{-d_{x,y}^2}$ this will force

1. The empirical distribution $\bar{p}(x,y)$ is proportional to the number of times the pair (x,y) was observed. The representations $\{(x_i,y_i)\}_{i=1}^n$ and $\bar{p}(x,y)$ are equivalent up to a multiplicative factor.

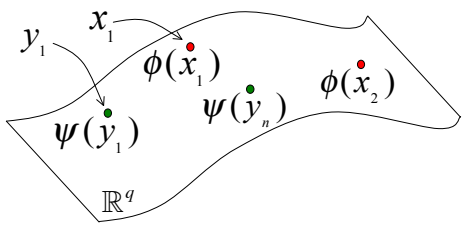

Figure 1: Embedding of X and Y into the same q-dimensional space. The embeddings functions $\phi : X \to \mathbb{R}^q$ and $\psi : Y \to \mathbb{R}^q$ determine the position of each instance in the low-dimensional space.

$\phi(x)$ to be far away from all $\psi(y)$. Such an embedding would reflect the marginal of x rather than its statistical relationship with all the other y values.

In what follows, we describe several methods to address this issue. Section 2.1 discusses symmetric models, and Section 2.2 conditional ones.

2.1 Symmetric Interaction Models

The goal of joint embedding is to have the geometry in the embedded space reflect the statistical relationships between variables, rather than just their joint probability. Specifically, the location $\phi(x)$ should be insensitive to the marginal $p(x)$, which just reflects the chance of observing x rather than the statistical relationship between x and different y values. To achieve this, we start by considering the ratio

$$r_p(x,y) = \frac{p(x,y)}{p(x)p(y)} \ , \ p(x) = \sum_y p(x,y) \ , \ p(y) = \sum_x p(x,y)$$

between the joint probability of x and y and the probability of observing that pair if the occurrences of x and y were independent. This ratio is widely used in statistics and information theory, for instance in the mutual information (Cover and Thomas, 1991), which is the expected value of the log of this ratio: $I(X;Y) = \sum_{x,y} p(x,y) \log \frac{p(x,y)}{p(x)p(y)} = \sum_{x,y} p(x,y) \log r_p(x,y)$. When X and Y are statistically independent, we have $r_p(x,y) = 1$ for all (x,y), and for any marginal distributions $p(x)$ and $p(y)$. Otherwise, high (low) values of $r_p(x,y)$ imply that the probability of $p(x,y)$ is larger (smaller) than the probability assuming independent variables.

Since $r_p(x,y)$ models statistical dependency, it is a natural choice to construct a model where $r_p(x,y)$ is proportional to $e^{-d^2_{x,y}}$. A first attempt at such a model is

$$p(x,y) = \frac{1}{Z} p(x)p(y)e^{-d^2_{x,y}} \ \ , \ \ p(x) = \sum_y p(x,y) \ \ , \ \ p(y) = \sum_x p(x,y) \ , \tag{1}$$

where Z is a normalization term (partition function). The key difficulty with this model is that $p(x)$ and $p(y)$, which appear in the model, are dependent on $p(x,y)$. Hence, some choices of $d^2_{x,y}$ lead to invalid models. As a result, one has to choose $p(x)$, $p(y)$ and $d^2_{x,y}$ jointly such that the

$p(x,y)$ obtained is consistent with the given marginals $p(x)$ and $p(y)$. This significantly complicates parameter estimation in such models, and we do not pursue them further here.[2]

To avoid the above difficulty, we use instead the ratio to the empirical marginals $\bar{p}(x)$ and $\bar{p}(y)$

$$\bar{r}_p(x,y) = \frac{p(x,y)}{\bar{p}(x)\,\bar{p}(y)}, \quad \bar{p}(x) = \sum_y \bar{p}(x,y) \ , \quad \bar{p}(y) = \sum_x \bar{p}(x,y) \ .$$

This is a good approximation of r_p when $\bar{p}(x)$, $\bar{p}(y)$ are close to $p(x), p(y)$, which is a reasonable assumption for the applications that we consider. Requiring $\bar{r}_p(x,y)$ to be proportional to $e^{-d^2_{x,y}}$, we obtain the following model

$$p_{MM}(x,y) \equiv \frac{1}{Z}\,\bar{p}(x)\,\bar{p}(y)\bar{e}^{d^2_{x,y}} \quad \forall x \in X, \forall y \in Y \ , \tag{2}$$

where $Z = \sum_{x,y} \bar{p}(x)\,\bar{p}(y)\bar{e}^{d^2_{x,y}}$ is a normalization term. The subscript MM reflects the fact that the model contains the marginal factors $\bar{p}(x)$ and $\bar{p}(y)$. We use different subscripts to distinguish between the models that we consider (see Section 2.3). The distribution $p_{MM}(x,y)$ satisfies $\bar{r}_{p_{MM}}(x,y) \propto e^{-d^2_{x,y}}$, providing a direct relation between statistical dependencies and embedding distances. Note that $p_{MM}(x,y)$ has zero probability for any x or y that are not in the support of $\bar{p}(x)$ or $\bar{p}(y)$ (that is, $\bar{p}(x) = 0$ or $\bar{p}(y) = 0$). This does not pose a problem because such values of X or Y will not be included in the model to begin with, since we essentially cannot learn anything about them when variables are purely categorical. When the X or Y objects have additional structure, it may be possible to infer embeddings of unobserved values. This is discussed further in Section 9.

The model $p_{MM}(x)$ is symmetric with respect to the variables X and Y. The next section describes a model that breaks this symmetry by conditioning on one of the variables.

2.2 Conditional Models

A standard approach to avoid modeling marginal distributions is to use conditional distributions instead of the joint distribution. In some cases, conditional models are a more plausible generating mechanism for the data. For instance, a distribution of authors and words in their works is more naturally modeled as first choosing an author according to some prior and then generating words according to the author's vocabulary preferences. In this case, we can use the embedding distances $d^2_{x,y}$ to model the conditional word generation process rather than the joint distribution of authors and words.

The following equation defines a distance-based model for conditional co-occurrence probabilities:

$$p_{CM}(y|x) \equiv \frac{1}{Z(x)}\,\bar{p}(y)\bar{e}^{d^2_{x,y}} \quad \forall x \in X, \forall y \in Y \ . \tag{3}$$

$Z(x) = \sum_y \bar{p}(y)\bar{e}^{d^2_{x,y}}$ is a partition function for the given value x, and the subscript CM reflects the fact that we are conditioning on X and multiplying by the marginal of Y. We can use $p_{CM}(y|x)$ and the empirical marginal $\bar{p}(x)$ to define a joint model $p_{CM}(x,y) \equiv p_{CM}(y|x)\,\bar{p}(x)$ so that

$$p_{CM}(x,y) = \frac{1}{Z(x)}\,\bar{p}(x)\,\bar{p}(y)\bar{e}^{d^2_{x,y}} \ . \tag{4}$$

2. Equation 1 bears some resemblance to copula based models (Nelsen, 1999) where joint distributions are modeled as a product of marginals and interaction terms. However, copula models are typically based on continuous variables with specific interaction terms, and thus do not resolve the difficulty mentioned above.

This model satisfies the relation $\bar{p}_{CM}(x,y) \propto \frac{1}{Z(x)} e^{-d^2_{x,y}}$ between statistical dependency and distance. This implies that for a given x, the nearest neighbor $\psi(y)$ of $\phi(x)$ corresponds to the y with the largest dependency ratio $\bar{p}_{CM}(x,y)$.

A GENERATIVE PROCESS FOR CONDITIONAL MODELS

One advantage of using a probabilistic model to describe complex data is that the model may reflect a mechanism for generating the data. To study such a mechanism here, we consider a simplified conditional model

$$p_{CU}(y|x) \equiv \frac{1}{Z(x)} e^{-d^2_{x,y}} = \frac{1}{Z(x)} e^{-\|\phi(x)-\psi(y)\|^2} . \tag{5}$$

We also define the corresponding joint model $p_{CU}(x,y) = p_{CU}(y|x)\,\bar{p}(x)$, as in Equation 4.

The model in Equation 5 states that for a given x, the probability of generating a given y is proportional to $e^{-d^2_{x,y}}$. To obtain a generative interpretation of this model, consider the case where every point in the space $\mathbb{R}^q$ corresponds to $\psi(y)$ for some y (that is, ψ is a surjective map). This will only be possible if there is a one to one mapping between the variable Y and $\mathbb{R}^q$, so for the purpose of this section we assume that Y is not discrete. Sampling a pair (x,y) from $p_{CU}(x,y)$ then corresponds to the following generative procedure:

- Sample a value of x from $\bar{p}(x)$.
- For this x, perform a random walk in the space $\mathbb{R}^q$ starting at the point $\phi(x)$ and terminating after a fixed time T.[3]
- Denote the termination point of the random walk by $\mathbf{z} \in \mathbb{R}^q$.
- Return the value of y for which $\mathbf{z} = \psi(y)$.

The termination point of a random walk has a Gaussian distribution, with a mean given by the starting point $\phi(x)$. The conditional distribution $p_{CU}(y|x)$ has exactly this Gaussian form, and therefore the above process generates pairs according to the distribution $p_{CU}(x,y)$. This process only describes the generation of a single pair (x_i,y_i), and distinct pairs are assumed to be generated IID. It will be interesting to consider models for generating sequences of pairs via one random walk.

A generative process for the model p_{CM} in Equation 3 is less straightforward to obtain. Intuitively, it should correspond to a random walk that is weighted by some prior over Y. Thus, the random walk should be less likely to terminate at points $\psi(y)$ that correspond to low $\bar{p}(y)$. The multiplicative interaction between the exponentiated distance and the prior makes it harder to define a generative process in this case.

2.3 Alternative Models

The previous sections considered several models relating distributions to embeddings. The notation we used for naming the models above is of the form p_{AB} where A and B specify the treatment of the X and Y marginals, respectively. The following values are possible for A and B:

- C : The variable is conditioned on.

3. Different choices of T will correspond to different constants multiplying $d^2_{x,y}$ in Equation 5. We assume here that T is chosen such that this constant is one.

- M : The variable is not conditioned on, and its observed marginal appears in the distribution.
- U : The variable is not conditioned on, and its observed marginal does not appear in the distribution.

This notation can be used to define models not considered above. Some examples, which we also evaluate empirically in Section 8.5 are

$$\begin{aligned} p_{UU}(x,y) &\equiv \frac{1}{Z}e^{-d^2_{x,y}}, \\ p_{MC}(x|y) &\equiv \frac{1}{Z(y)}\,\bar{p}(x)e^{-d^2_{x,y}}, \\ p_{UC}(x|y) &\equiv \frac{1}{Z(y)}e^{-d^2_{x,y}}. \end{aligned} \tag{6}$$

2.4 Choosing the "Right" Model

The models discussed in Sections 2.1-2.3 present different approaches to relating probabilities to distances. They differ in their treatment of marginals, and in using distances to model either joint or conditional distributions. They thus correspond to different assumptions about the data. For example, conditional models assume an asymmetric generative model, where distances are related only to conditional distributions. Symmetric models may be more appropriate when no such conditional assumption is valid. We performed a quantitative comparison of all the above models on a task of word-document embedding, as described in Section 8.5. Our results indicate that, as expected, models that address both marginals (such as p_{CM} or p_{MM}), and that are therefore directly related to the ratio $\bar{p}(x,y)$, outperform models which do not address marginals. Although there are two possible conditional models, conditioning on X or on Y, for the specific task studied in Section 8.5 one of the conditional models is more sensible as a generating mechanism, and indeed yielded better results.

3. Learning the Model Parameters

We now turn to the task of learning the model parameters $\{\phi(x),\psi(y)\}$ from empirical data. In what follows, we focus on the model $p_{CM}(x,y)$ in Equation 4. However, all our derivations are easily applied to the other models in Section 2. Since we have a parametric model of a distribution, it is natural to look for the parameters that maximize the log-likelihood of the observed pairs $\{(x_i,y_i)\}_{i=1}^n$. For a given set of observed pairs, the average log-likelihood is[4]

$$\ell(\phi,\psi) = \frac{1}{n}\sum_{i=1}^{n}\log p_{CM}(x_i,y_i)\ .$$

The log-likelihood may equivalently be expressed in terms of the distribution $\bar{p}(x,y)$ since

$$\ell(\phi,\psi) = \sum_{x,y}\bar{p}(x,y)\log p_{CM}(x,y)\ .$$

4. For conditional models we can consider maximizing only the conditional log-likelihood $\frac{1}{n}\sum_{i=1}^n \log p(y_i|x_i)$. This is equivalent to maximizing the joint log-likelihood for the model $p(y|x)\bar{p}(x)$, and we prefer to focus on joint likelihood maximization so that a unified formulation is used for both joint and conditional models.

As in other cases, maximizing the log-likelihood is also equivalent to minimizing the KL divergence D_{KL} between the empirical and the model distributions, since $\ell(\phi,\psi)$ equals $D_{\mathrm{KL}}[\,\bar{p}(x,y)|p_{CM}(x,y)]$ up to an additive constant.

The log-likelihood in our case is given by

$$\begin{aligned}\ell(\phi,\psi) &= \sum_{x,y}\bar{p}(x,y)\log p_{CM}(x,y)\\ &= \sum_{x,y}\bar{p}(x,y)\Big(-d^2_{x,y}-\log Z(x)+\log\bar{p}(x)+\log\bar{p}(y)\Big)\\ &= -\sum_{x,y}\bar{p}(x,y)d^2_{x,y}-\sum_{x}\bar{p}(x)\log Z(x)+const\,, \end{aligned}\tag{7}$$

where $const=\sum_y\bar{p}(y)\log\bar{p}(y)+\sum_x\bar{p}(x)\log\bar{p}(x)$ is a constant term that does not depend on the parameters $\phi(x)$ and $\psi(y)$.

Finding the optimal parameters now corresponds to solving the following optimization problem

$$(\phi^*,\psi^*)=\arg\max_{\phi,\psi}\ell(\phi,\psi)\,.\tag{8}$$

The log-likelihood is composed of two terms. The first is (minus) the mean distance between x and y. This will be maximized when all distances are zero. This trivial solution is avoided because of the *regularization* term $\sum_x\bar{p}(x)\log Z(x)$, which acts to increase distances between x and y points.

To characterize the maxima of the log-likelihood we differentiate it with respect to the embeddings of individual objects $(\phi(x),\psi(y))$, and obtain the following gradients

$$\begin{aligned}\frac{\partial\ell(\phi,\psi)}{\partial\phi(x)} &= 2\,\bar{p}(x)\Big(\langle\psi(y)\rangle_{\bar{p}(y|x)}-\langle\psi(y)\rangle_{p_{CM}(y|x)}\Big)\,,\,,\\ \frac{\partial\ell(\phi,\psi)}{\partial\psi(y)} &= 2p_{CM}(y)\Big(\psi(y)-\langle\phi(x)\rangle_{p_{CM}(x|y)}\Big)-2\,\bar{p}(y)\Big(\psi(y)-\langle\phi(x)\rangle_{\bar{p}(x|y)}\Big)\,,\end{aligned}$$

where $p_{CM}(y)=\sum_x p_{CM}(y|x)\,\bar{p}(x)$ and $p_{CM}(x|y)=\frac{p_{CM}(x,y)}{p_{CM}(y)}$.

Equating the $\phi(x)$ gradient to zero yields:

$$\langle\psi(y)\rangle_{p_{CM}(y|x)}=\langle\psi(y)\rangle_{\bar{p}(y|x)}\,.\tag{9}$$

If we fix ψ, this equation is formally similar to the one that arises in the solution of conditional maximum entropy models (Berger et al., 1996). However, there is a crucial difference in that the exponent of $p_{CM}(y|x)$ in conditional maximum entropy is linear in the parameters (ϕ in our notation), while in our model it also includes quadratic (norm) terms in the parameters. The effect of Equation 9 can then be described informally as that of choosing $\phi(x)$ so that the expected value of ψ under $p_{CM}(y|x)$ is the same as its empirical average, that is, placing the embedding of x closer to the embeddings of those y values that have stronger statistical dependence with x.

The maximization problem of Equation 8 is not jointly convex in $\phi(x)$ and $\psi(y)$ due to the quadratic terms in d^2_{xy}.[5] To find the local maximum of the log-likelihood with respect to both $\phi(x)$ and $\psi(y)$ for a given embedding dimension q, we use a conjugate gradient ascent algorithm with random restarts.[6] In Section 5 we describe a different approach to this optimization problem.

5. The log-likelihood *is* a convex function of $\phi(x)$ for a constant $\psi(y)$, as noted in Iwata et al. (2005), but is not convex in $\psi(y)$ for a constant $\phi(x)$.

6. The code is provided online at `http://ai.stanford.edu/~gal/`.

4. Relation to Other Methods

In this section we discuss other methods for representing co-occurrence data via low dimensional vectors, and study the relation between these methods and the CODE models.

4.1 Maximizing Correlations and Related Methods

Embedding the rows and columns of a contingency table into a low dimensional Euclidean space was previously studied in the statistics literature. Fisher (1940) described a method for mapping X and Y into scalars $\phi(x)$ and $\psi(y)$ such that the correlation coefficient between $\phi(x)$ and $\psi(y)$ is maximized. The method of Correspondence Analysis (CA) generalizes Fisher's method to non-scalar mappings. More details about CA are given in Appendix A. Similar ideas have been applied to more than two variables in the Gifi system (Michailidis and de Leeuw, 1998). All these methods can be shown to be equivalent to the more widely known *canonical correlation analysis* (CCA) procedure (Hotelling, 1935). In CCA one is given two continuous multivariate random variables X and Y, and aims to find two sets of vectors, one for X and the other for Y, such that the correlations between the projections of the variables onto these vectors are maximized. The optimal projections for X and Y can be found by solving an eigenvalue problem. It can be shown (Hill, 1974) that if one represents X and Y via indicator vectors, the CCA of these vectors (when replicated according to their empirical frequencies) results in Fisher's mapping and CA.

The objective of these correlation based methods is to maximize the correlation coefficient between the embeddings of X and Y. We now discuss their relation to our distance-based method. First, the correlation coefficient is invariant under affine transformations and we can thus focus on centered solutions with a unity covariance matrix solutions: $\langle\phi(X)\rangle = 0, \langle\psi(Y)\rangle = 0$ and $\mathrm{Cov}(\phi(X)) = \mathrm{Cov}(\psi(Y)) = I$. In this case, the correlation coefficient is given by the following expression (we focus on $q = 1$ for simplicity)

$$\rho(\phi(x),\psi(y)) = \sum_{x,y} \bar{p}(x,y)\phi(x)\psi(y) = -\frac{1}{2}\sum_{x,y} \bar{p}(x,y)d^2_{x,y} + 1 \ .$$

Maximizing the correlation is therefore equivalent to minimizing the mean distance across all pairs. This clarifies the relation between CCA and our method: Both methods aim to minimize the average distance between X and Y embeddings. However, CCA forces embeddings to be centered and scaled, whereas our method introduces a global regularization term related to the partition function.

A kernel variant of CCA has been described in Lai and Fyfe (2000) and Bach and Jordan (2002), where the input vectors X and Y are first mapped to a high dimensional space, where linear projection is carried out. This idea could possibly be used to obtain a kernel version of correspondence analysis, although we are not aware of existing work in that direction.

Recently, Zhong et al. (2004) presented a co-embedding approach for detecting unusual activity in video sequences. Their method also minimizes an averaged distance measure, but normalizes it by the variance of the embedding to avoid trivial solutions.

4.2 Distance-Based Embeddings

Multidimensional scaling (MDS) is a well-known geometric embedding method (Cox and Cox, 1984), whose standard version applies to same-type objects with predefined distances. MDS embedding of heterogeneous entities was studied in the context of modeling ranking data (Cox and

Cox, 1984, Section 7.3). These models, however, focus on specific properties of ordinal data and therefore result in optimization principles and algorithms different from our probabilistic interpretation.

Relating Euclidean structure to probability distributions was previously discussed by Hinton and Roweis (2003). They assume that distances between points in some X space are given, and the exponent of these distances induces a distribution $p(x=i|x=j)$ which is proportional to the exponent of the distance between $\phi(i)$ and $\phi(j)$. This distribution is then approximated via an exponent of distances in a low dimensional space. Our approach differs from theirs in that we treat the joint embedding of two different spaces. Therefore, we do not assume a metric structure between X and Y, but instead use co-occurrence data to learn such a structure. The two approaches become similar when $X=Y$ and the empirical data *exactly* obeys an exponential law as in Equation 3.

Iwata et al. (2005) recently introduced the Parametric Embedding (PE) method for visualizing the output of supervised classifiers. They use the model of Equation 3 where Y is taken to be the class label, and X is the input features. Their embedding thus illustrates which X values are close to which classes, and how the different classes are inter-related. The approach presented here can be viewed as a generalization of their approach to the unsupervised case, where X and Y are arbitrary objects.

An interesting extension of locally linear embedding (Roweis and Saul, 2000) to heterogeneous embedding was presented by Ham et al. (2003). Their method essentially forces the outputs of two locally linear embeddings to be aligned such that corresponding pairs of objects are mapped to similar points.

A Bayesian network approach to joint embedding was recently studied in Mei and Shelton (2006) in the context of collaborative filtering.

4.3 Matrix Factorization Methods

The empirical joint distribution $\bar{p}(x,y)$ can be viewed as a matrix $\bar{P}$ of size $|X| \times |Y|$. There is much literature on finding low rank approximations of matrices, and specifically matrices that represent distributions (Hofmann, 2001; Lee and Seung, 1999). Low rank approximations are often expressed as a product UV^T where U and V are two matrices of size $|X| \times q$ and $|Y| \times q$ respectively.

In this context CODE can be viewed as a special type of low rank approximation of the matrix $\bar{P}$. Consider the symmetric model p_{UU} in Equation 6, and the following matrix and vector definitions:[7]

- Let Φ be a matrix of size $|X| \times q$ where the i^{th} row is $\phi(i)$. Let Ψ be a matrix of size $|Y| \times q$ where the i^{th} row is $\psi(i)$.
- Define the column vector $\mathbf{u}(\Phi) \in \mathbb{R}^{|X|}$ as the set of squared Euclidean norms of $\phi(i)$, so that $u_i(\phi) = \|\phi(i)\|^2$. Similarly define $\mathbf{v}(\psi) \in \mathbb{R}^{|Y|}$ as $v_i(\psi) = \|\psi(i)\|^2$.
- Denote the k-dimensional column vector of all ones by $\mathbf{1}_k$.

Using these definitions, the model p_{UU} can then be written in matrix form as

$$\log P_{UU} = -\log Z + 2\Phi\Psi^T - \mathbf{u}(\Phi)\mathbf{1}_{|Y|}^T - \mathbf{1}_{|X|}\mathbf{v}(\Psi)^T$$

where the optimal Φ and Ψ are found by minimizing the KL divergence between $\bar{P}$ and P_{UU}.

7. We consider p_{UU} for simplicity. Other models, such as p_{MM}, have similar interpretations.

The model for $\log P_{UU}$ is in fact low-rank, since the rank of $\log P_{UU}$ is at most $q+2$. However, note that P_{UU} itself will not necessarily have a low rank. Thus, CODE can be viewed as a low-rank matrix factorization method, where the structure of the factorization is motivated by distances between rows of Φ and Ψ, and the quality of the approximation is measured via the KL divergence.

Many matrix factorization algorithms (such as Lee and Seung, 1999) use the term $\Phi\Psi^T$ above, but not the terms $\mathbf{u}(\Phi)$ and $\mathbf{v}(\Psi)$. Another algorithm that uses only the $\Phi\Psi^T$ term, but is more closely related to CODE is the sufficient dimensionality reduction (SDR) method of Globerson and Tishby (2003). SDR seeks a model

$$\log P_{SDR} = -\log Z + \Phi\Psi^T + \mathbf{a}\mathbf{1}_{|Y|}^T + \mathbf{1}_{|X|}\mathbf{b}^T$$

where $\mathbf{a}, \mathbf{b}$ are vectors of dimension $|X|, |Y|$ respectively. As in CODE, the parameters $\Phi, \Psi, \mathbf{a}$ and $\mathbf{b}$ are chosen to maximize the likelihood of the observed data.

The key difference between CODE and SDR lies in the terms $\mathbf{u}(\Phi)$ and $\mathbf{v}(\Psi)$ which are *non-linear* in Φ and Ψ. These arise from the geometric interpretation of CODE that relates distances between embeddings to probabilities. SDR does not have such an interpretation. In fact, the SDR model is invariant to the translation of either of the embedding maps (for instance, $\phi(x)$), while fixing the other map $\psi(y)$. Such a transformation would completely change the distances $d_{x,y}^2$ and is clearly not an invariant property in the CODE models.

5. Semidefinite Representation

The CODE learning problem in Equation 8 is not jointly convex in the parameters ϕ and ψ. In this section we present a convex relaxation of the learning problem. For a sufficiently high embedding dimension this approximation is in fact exact, as we show next. For simplicity, we focus on the p_{CM} model, although similar derivations may be applied to the other models.

5.1 The Full Rank Case

Locally optimal CODE embeddings $\phi(x)$ and $\psi(y)$ may be found using standard unconstrained optimization techniques. However, the Euclidean distances used in the embedding space also allow us to reformulate the problem as constrained convex optimization over the cone of positive semidefinite (PSD) matrices (Boyd and Vandenberghe, 2004).

We begin by showing that for embeddings with dimension $q = |X| + |Y|$, maximizing the CODE likelihood (see Equation 8) is equivalent to minimizing a certain convex non-linear function over PSD matrices. Consider the matrix A whose columns are all the embedded vectors $\phi(x)$ and $\psi(y)$

$$A \equiv [\phi(1), \ldots, \phi(|X|), \psi(1), \ldots, \psi(|Y|)] \ .$$

Define the Gram matrix G as

$$G \equiv A^T A \ .$$

G is a matrix of the dot products between the coordinate vectors of the embedding, and is therefore a symmetric PSD matrix of rank $\leq q$. Conversely, any PSD matrix of rank $\leq q$ can be factorized as $A^T A$, where A is some embedding matrix of dimension q. Thus we can replace optimization over matrices A with optimization over PSD matrices of rank $\leq q$. Note also that the distance between two columns in A is *linearly* related to the Gram matrix via $d_{xy}^2 = g_{xx} + g_{yy} - 2g_{xy}$, and thus the embedding distances are linear functions of the elements of G.

Since the log-likelihood function in Equation 7 depends only on the distances between points in X and in Y, we can write it as a function of G only.[8] In what follows, we focus on the negative log-likelihood $f(G) = -\ell(G)$

$$f(G) = \sum_{x,y} \bar{p}(x,y)(g_{xx} + g_{yy} - 2g_{xy}) + \sum_x \bar{p}(x) \log \sum_y \bar{p}(y) e^{-(g_{xx}+g_{yy}-2g_{xy})} .$$

The likelihood maximization problem can then be written in terms of constrained minimization over the set of rank q positive semidefinite matrices[9]

$$\begin{array}{ll} \min_G & f(G) \\ \text{s.t.} & G \succeq 0 \\ & \text{rank}(G) \leq q \; . \end{array} \tag{10}$$

Thus, the CODE log-likelihood maximization problem in Equation 8 is equivalent to minimizing a nonlinear objective over the set of PSD matrices of a constrained rank.

When the embedding dimension is $q = |X| + |Y|$ the rank constraint is always satisfied and the problem reduces to

$$\begin{array}{ll} \min_G & f(G) \\ \text{s.t.} & G \succeq 0 \; . \end{array} \tag{11}$$

The minimized function $f(G)$ consists of two convex terms: The first term is a linear function of G; the second term is a sum of $\log \sum \exp$ terms of an affine expression in G. The $\log \sum \exp$ function is convex (Boyd and Vandenberghe, 2004, Section 4.5), and therefore the function $f(G)$ is convex. Moreover, the set of constraints is also convex since the set of PSD matrices is a convex cone (Boyd and Vandenberghe, 2004). We conclude that when the embedding dimension is of size $q = |X| + |Y|$ the optimization problem of Equation 11 is convex, and thus has no local minima.

5.1.1 ALGORITHMS

The convex optimization problem in Equation 11 can be viewed as a PSD constrained geometric program.[10] This is not a semidefinite program (SDP, see Vandenberghe and Boyd, 1996), since the objective function in our case is non-linear and SDPs are defined as having both a linear objective and linear constraints. As a result we cannot use standard SDP tools in the optimization. It seems like such Geometric Program/PSD problems have not been dealt with in the optimization literature, and it will be interesting to develop specialized algorithms for these cases.

The optimization problem in Equation 11 can however be solved using any general purpose convex optimization method. Here we use the *projected gradient* algorithm (Bertsekas, 1976), a simple method for constrained convex minimization. The algorithm takes small steps in the direction of the negative objective gradient, followed by a Euclidean projection on the set of PSD matrices. This projection is calculated by eliminating the contribution of all eigenvectors with negative eigenvalues to the current matrix, similarly to the PSD projection algorithm of Xing et al. (2002). Pseudo-code for this procedure is given in Figure 2.

In terms of complexity, the most time consuming part of the algorithm is the eigenvector calculation which is $O((|X| + |Y|)^3)$ (Pan and Chen, 1999). This is reasonable when $|X| + |Y|$ is a few thousands but becomes infeasible for much larger values of $|X|$ and $|Y|$.

8. We ignore the constant additive terms.
9. The objective $f(G)$ is *minus* the log-likelihood, which is why minimization is used.
10. A geometric program is a convex optimization problem where the objective and the constraints are $\log \sum \exp$ functions of an affine function of the variables (Chiang, 2005).

Input: Empirical distribution $\bar{p}(x,y)$. A step size ε.

Output: PSD matrix of size $|X|+|Y|$ that solves the optimization problem in Equation 11.

Initialize: Set G_0 to the identity matrix of size $|X|+|Y|$.

Iterate:

- Set $\hat{G}_{t+1} = G_t - \varepsilon \bigtriangledown f(G_t)$.
- Calculate the eigen-decomposition of $\hat{G}_{t+1}$: $\hat{G}_{t+1} = \sum_k \lambda_k \mathbf{u}_k \mathbf{u}_k^T$.
- Set $G_{t+1} = \sum_k \max(\lambda_k, 0) \mathbf{u}_k \mathbf{u}_k^T$.

Figure 2: A projected gradient algorithm for solving the optimization problem in Equation 11. To speed up convergence we also use an Armijo rule (Bertsekas, 1976) to select the step size ε at every iteration.

5.2 The Low-Dimensional Case

Embedding into a low dimension requires constraining the rank, but this is difficult since the problem in Equation 10 is not convex in the general case. One approach to obtaining low rank solutions is to optimize over a full rank G and then project it into a lower dimension via spectral decomposition as in Weinberger and Saul (2006) or classical MDS. However, in the current problem, this was found to be ineffective.

A more effective approach in our case is to regularize the objective by adding a term $\lambda \mathrm{Tr}(G)$, for some constant $\lambda > 0$. This keeps the problem convex, since the trace is a linear function of G. Furthermore, since the eigenvalues of G are non-negative, this term corresponds to ℓ_1 regularization on the eigenvalues. Such regularization is likely to result in a sparse set of eigenvalues, and thus in a low dimensional solution, and is indeed a commonly used trick in obtaining such solutions (Fazel et al., 2001). This results in the following regularized problem

$$\begin{array}{ll} \min_G & f(G) + \lambda \mathrm{Tr}(G) \\ \text{s.t.} & G \succeq 0 \ . \end{array} \tag{12}$$

Since the problem is still convex, we can again use a projected gradient algorithm as in Figure 2 for the optimization. We only need to replace $\nabla f(G_t)$ with $\lambda I + \nabla f(G_t)$ where I is an identity matrix of the same size as G.

Now suppose we are seeking a q dimensional embedding, where $q < |X|+|Y|$. We would like to use λ to obtain low dimensional solutions, but to choose the q dimensional solution with maximum log-likelihood. This results in the PSD-CODE procedure described in Figure 3. This approach is illustrated in Figure 4 for $q = 2$. The figure shows log-likelihood values of regularized PSD solutions projected to two dimensions. The values of λ which achieve the optimal likelihood also result in only two significant eigenvalues, showing that the regularization and projection procedure indeed produces low dimensional solutions.

PSD-CODE

Input: A set of regularization parameters $\{\lambda_i\}_{i=1}^n$, an embedding dimension q, and empirical distribution $\bar{p}(x,y)$.

Output: A q dimensional embedding of X and Y

Algorithm

- For each value of λ_i:
  - Use the projected gradient algorithm to solve the optimization problem in Equation 12 with regularization parameter λ_i. Denote the solution by G.
  - Transform G into a rank q matrix G_q by keeping only the q eigenvectors with the largest eigenvalues.
  - Calculate the likelihood of the data under the model given by the matrix G_q. Denote this likelihood by ℓ_i.
- Find the λ_i which maximizes ℓ_i, and return its corresponding embedding.

Figure 3: The PSD-CODE algorithm for finding a low dimensional embedding using PSD optimization.

The PSD-CODE algorithm was applied to subsets of the databases described in Section 7 and yielded similar results to those of the conjugate-gradient based algorithm. We believe that PSD algorithms may turn out to be more efficient in cases where relatively high dimensional embeddings are sought. Furthermore, with the PSD formulation it is easy to introduce additional constraints, for example on distances between subsets of points (Weinberger and Saul, 2006). Section 6.1 considers a model extension that could benefit from such a formulation.

6. Using Additional Co-occurrence Data

The methods described so far use a single co-occurrence table of two objects. However, in some cases we may have access to additional information about (X,Y) and possibly other variables. Below we describe extensions of CODE to these settings.

6.1 Within-Variable Similarity Measures

The CODE models in Section 2 rely only on the co-occurrence of X and Y but assume nothing about similarity between two objects of the same type. Such a similarity measure may often be available and could take several forms. One is a distance measure between objects in X. For example, if $\mathbf{x} \in \mathbb{R}^p$ we may take the Euclidean distance $\|\mathbf{x}_i - \mathbf{x}_j\|_2$ between two vectors $\mathbf{x}_i, \mathbf{x}_j \in \mathbb{R}^p$ as a measure of similarity. This information may be combined with co-occurrence data either by requiring the

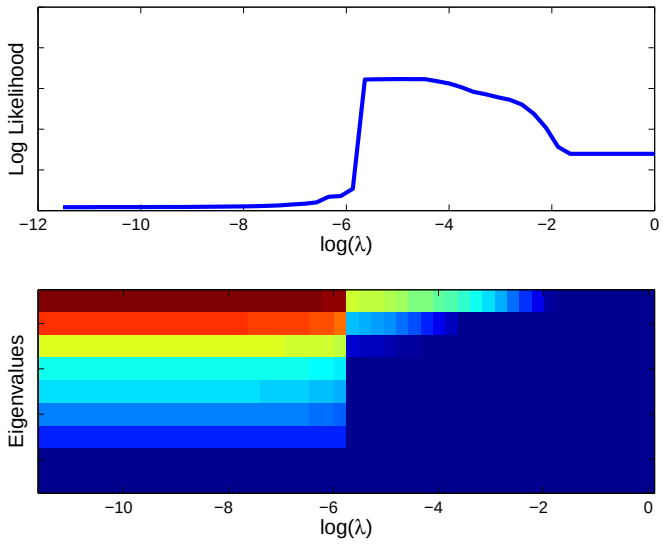

Figure 4: Results for the PSD-CODE algorithm. Data is the 5×4 contingency table in Greenacre (1984) page 55. **Top**: The log-likelihood of the solution projected to two dimensions, as a function of the regularization parameter λ. **Bottom**: The eigenvalues of the Gram matrix obtained using the PSD algorithm for the corresponding λ values. It can be seen that solutions with two dominant eigenvalues have higher likelihoods.

CODE map to agree with the given distances, or by adding a term which penalizes deviations from them.

Similarities between two objects in X may also be given in the form of co-occurrence data. For example, if X corresponds to words and Y corresponds to authors (see Section 7.1), we may have access to joint statistics of words, such as bigram statistics, which give additional information about which words should be mapped together. Alternatively, we may have access to data about collaboration between authors, for example, what is the probability of two authors writing a paper together. This in turn should affect the mapping of authors.

The above example can be formalized by considering two distributions $\bar{p}(x^{(1)}, x^{(2)})$ and $\bar{p}(y^{(1)}, y^{(2)})$ which describe the *within-type* object co-occurrence rates. One can then construct a CODE model as in Equation 3 for $p(x^{(1)}|x^{(2)})$

$$p(x^{(1)}|x^{(2)}) = \frac{\bar{p}(x^{(1)})}{Z(x^{(1)})} e^{-\|\phi(x^{(1)}) - \phi(x^{(2)})\|^2} .$$

Denote the log-likelihood for the above model by $l_x(\phi)$, and the corresponding log-likelihood for $p(y^{(1)}|y^{(2)})$ by $\ell_y(\psi)$. Then we can combine several likelihood terms by maximizing some weighted combination $\ell(\phi, \psi) + \lambda_x \ell_x(\phi) + \lambda_y \ell_y(\psi)$, where $\lambda_x, \lambda_y \geq 0$ reflect the relative weight of each information source.

6.2 Embedding More than Two Variables

The notion of a common underlying semantic space is clearly not limited to two objects. For example, texts, images and audio files may all have a similar meaning and we may therefore wish to embed all three in a single space. One approach in this case could be to use joint co-occurrence statistics $\bar{p}(x,y,z)$ for all three object types, and construct a geometric-probabilistic model for the distribution $p(x,y,z)$ using three embeddings $\phi(x), \psi(y)$ and $\xi(z)$ (see Section 9 for further discussion of this approach). However, in some cases obtaining joint counts over multiple objects may not be easy. Here we describe a simple extension of CODE to the case where more than two variables are considered, but empirical distributions are available only for *pairs* of variables.

To illustrate the approach, consider a case with k different variables $X^{(1)},\ldots,X^{(k)}$ and an additional variable Y. Assume that we are given empirical joint distributions of Y with each of the X variables $\bar{p}(x^{(1)},y),\ldots,\bar{p}(x^{(k)},y)$. It is now possible to consider a set of k CODE models $p(x^{(i)},y)$ for $i=1,\ldots,k$,[11] where each $X^{(i)}$ will have an embedding $\phi^{(i)}(x^{(i)})$ but all models will share the same $\psi(y)$ embedding. Given k non-negative weights $w_1,\ldots,w_k$ that reflect the "relative importance" of each $X^{(i)}$ we can consider the total weighted log-likelihood of the k models given by

$$\ell(\phi^{(1)},\ldots,\phi^{(k)},\psi) = \sum_i w_i \sum_{x^{(i)},y} \bar{p}(x^{(i)},y)\log p(x^{(i)},y)\ .$$

Maximizing the above log-likelihood will effectively combine structures in all the input distributions $\bar{p}(x^{(i)},y)$. For example if $Y=y$ often co-occurs with $X^{(1)}=x^{(1)}$ *and* $X^{(2)}=x^{(2)}$, likelihood will be increased by setting $\psi(y)$ to be close to both $\phi^{(1)}(x^{(1)})$ and $\phi^{(2)}(x^{(2)})$.

In the example above, it was assumed that only a single variable, Y, was shared between different pairwise distributions. It is straightforward to apply the same approach when more variables are shared: simply construct CODE models for all available pairwise distributions, and maximize their weighted log-likelihood.

Section 7.2 shows how this approach is used to successfully embed three different objects, namely authors, words, and documents in a database of scientific papers.

7. Applications

We demonstrate the performance of co-occurrence embedding on two real-world types of data. First, we use documents from NIPS conferences to obtain documents-word and author-word embeddings. These embeddings are used to visualize various structures in this complex corpus. We also use the multiple co-occurrence approach in Section 6.2 to embed authors, words, and documents into a single map. To provide quantitative assessment of the performance of our method, we apply it to embed the document-word 20 Usenet newsgroups data set, and we use the embedding to predict the class (newsgroup) for each document, which was not available when creating the embedding. Our method consistently outperforms previous unsupervised methods evaluated on this task.

In most of the experiments we use the conditional based model of Equation 4, except in Section 8.5 where the different models of Section 2 are compared.

11. This approach applies to all CODE models, such as p_{MM} or p_{CM}.

7.1 Visualizing a Document Database: The NIPS Database

Embedding algorithms are often used to visualize structures in document databases (Hinton and Roweis, 2003; Lin, 1997; Chalmers and Chitson, 1992). A common approach in these applications is to obtain some measure of similarity between objects of the same type such as words, and approximate it with distances in the embedding space.

Here we used the database of all papers from the NIPS conference until 2003. The database was based on an earlier database created by Roweis (2000), that included volumes 0-12 (until 1999).[12] The most recent three volumes also contain an indicator of the document's topic, for instance, AA for *Algorithms and Architectures*, LT for *Learning Theory*, and NS for *Neuroscience*, as shown in Figure 5.

We first used CODE to embed documents and words into $\mathbb{R}^2$. The results are shown in Figures 5 and 6. The empirical joint distribution was created as follows: for each document, the empirical distribution $\bar{p}(word|doc)$ was the number of times a word appeared in the document, normalized to one; this was then multiplied by a uniform prior $\bar{p}(doc)$ to obtain $\bar{p}(doc, word)$. The CODE model we used was the conditional word-given-document model $p_{CM}(doc, word)$. As Figure 5 illustrates, documents with similar topics tend to be mapped next to each other (for instance, AA near LT and NS near VB), even though the topic labels were not available to the algorithm when learning the embeddings. This shows that words in documents are good indicators of the topics, and that CODE reveals these relations. Figure 6 shows the joint embedding of documents and words. It can be seen that words indeed characterize the topics of their neighboring documents, so that the joint embedding reflects the underlying structure of the data.

Next, we used the data to generate an authors-words matrix $\bar{p}(author, word)$ obtained from counting the frequency with which a given author uses a given word. We could now embed authors and words into $\mathbb{R}^2$, by using CODE to model words given authors $p_{CM}(author, word)$. Figure 7 demonstrates that authors are indeed mapped next to terms relevant to their work, and that authors working on similar topics are mapped to nearby points. This illustrates how co-occurrence of words and authors can be used to induce a metric on authors alone.

These examples show how CODE can be used to visualize the complex relations between documents, their authors, topics and keywords.

7.2 Embedding Multiple Objects: Words, Authors and Documents

Section 6.2 presented an extension of CODE to multiple variables. Here we demonstrate that extension in embedding three object types from the NIPS database: words, authors, and documents. Section 7.1 showed embeddings of $(author, word)$ and $(doc, word)$. However, we may also consider a joint embedding for the objects $(author, word, doc)$, since there is a common semantic space underlying all three. To generate such an embedding, we apply the scheme of Section 6.2 with $Y \equiv word, X^{(1)} \equiv doc$ and $X^{(2)} \equiv author$. We use the two models $p_{CM}(author, word)$ and $p_{CM}(doc, word)$, that is, two conditional models where the *word* variable is conditioned on the *doc* or on the *author* variables. Recall that the embedding of the words is assumed to be the same in both models. We seek an embedding of all three objects that maximizes the weighted sum of the log-likelihood of these two models.

Different strategies may be used to weight the two log-likelihoods. One approach is to assign them equal weight by normalizing each by the total number of joint assignments. This corresponds

12. The data is available online at `http://ai.stanford.edu/~gal/`.

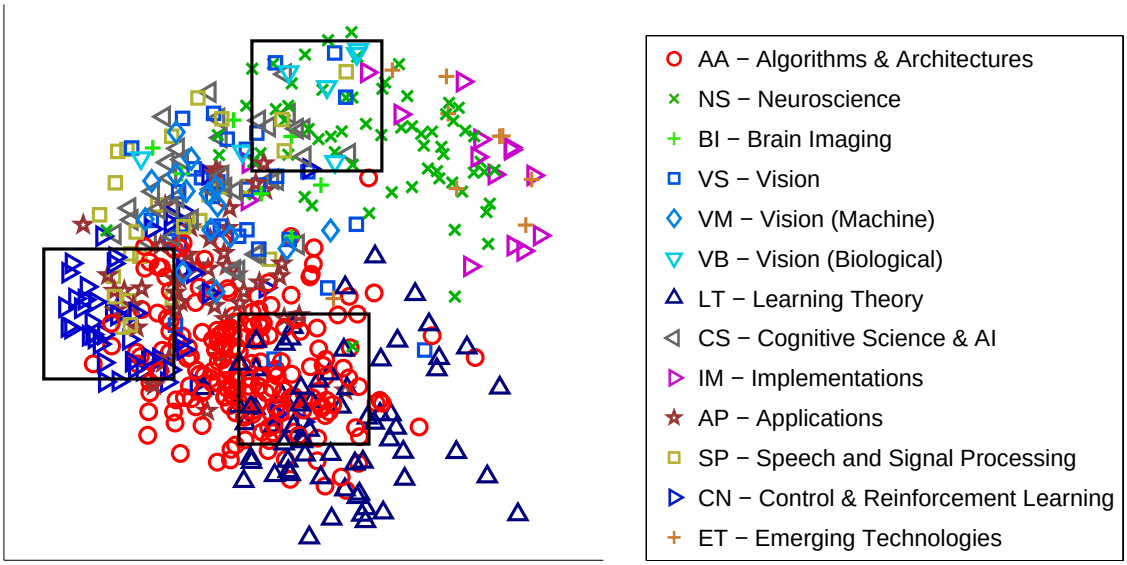

Figure 5: CODE embedding of 2483 documents and 2000 words from the NIPS database (the 2000 most frequent words, excluding the first 100, were used). Embedded documents from NIPS 15-17 are shown, with colors indicating the topic of each document. The word embeddings are not shown.

to choosing $w_i = \frac{1}{|X||Y^{(i)}|}$. For example, in this case the log-likelihood of $p_{CM}(author, word)$ will be weighted by $\frac{1}{|word||author|}$.

Figure 8 shows three insets of an embedding that uses the above weighting scheme.[13] The insets roughly correspond to those in Figure 6. However, here we have all three objects shown on the same map. It can be seen that both authors and words that correspond to a given topic are mapped together with documents about this topic.

It is interesting to study the sensitivity of the result to the choice of weights w_i. To evaluate this sensitivity, we introduce a quantitative measure of embedding quality: the *authorship* measure. The database we generated also includes the Boolean variable $isauthor(doc, author)$ that encodes whether a given author wrote a given document. This information is not available to the CODE algorithm and can be used to evaluate the documents-authors part of the authors-words-documents embedding. Given an embedding, we find the k nearest authors to a given document and calculate what fraction of the document's authors is in this set. We then average this across all k and all documents. Thus, for a document with three authors, this measure will be one if the three nearest authors to the document are its actual authors.

We evaluate the above authorship measure for different values of w_i to study the sensitivity of the embedding quality to changing the weights. Figure 9 shows that for a very large range of w_i values the measure is roughly constant, and it degrades quickly only when close to zero weight is

13. The overall document embedding was similar to Figure 5 and is not shown here.

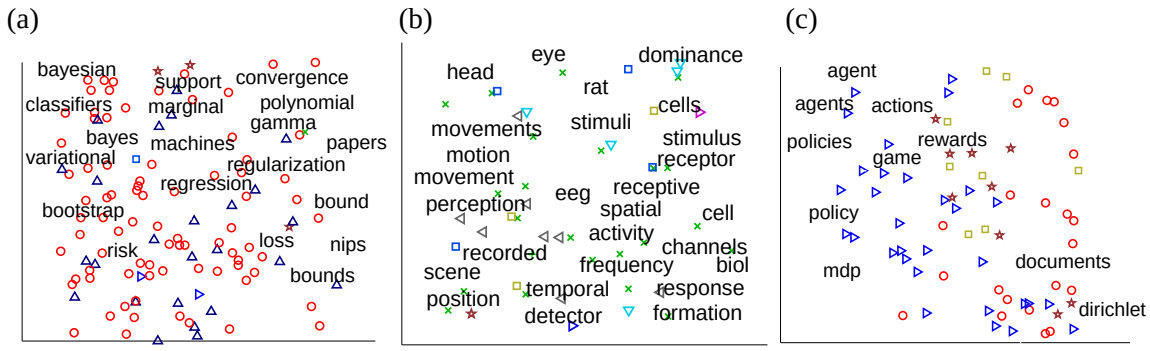

Figure 6: Each panel shows in detail one of the rectangles in Figure 5, and includes both the embedded documents and embedded words. **(a)** The border region between Algorithms and Architectures (AA) and Learning Theory (LT), corresponding to the bottom rectangle in Figure 5. **(b)** The border region between Neuroscience NS and Biological Vision (VB), corresponding to the upper rectangle in Figure 5. **(c)** Control and Reinforcement Learning (CN) region (left rectangle in Figure 5).

assigned to either of the two models. The stability with respect to w_i was also verified visually; embeddings were qualitatively similar for a wide range of weight values.

8. Quantitative Evaluation: The 20 Newsgroups Database

To obtain a quantitative evaluation of the effectiveness of our method, we apply it to a well controlled information retrieval task. The task contains known classes which are not used during learning, but are later used to evaluate the quality of the embedding.

8.1 The Data

We applied CODE to the widely studied 20 newsgroups corpus, consisting of 20 classes of 1000 documents each.[14] This corpus was further pre-processed as described by Chechik and Tishby (2003).[15] We first removed the 100 most frequent words, and then selected the next k most frequent words for different values of k (see below). The data was summarized as a count matrix $n(doc, word)$, which gives the count of each word in a document. To obtain an equal weight for all documents, we normalized the sum of each row in $n(doc, word)$ to one, and multiplied by $\frac{1}{|doc|}$. The resulting matrix is a joint distribution over the document and word variables, and is denoted by $\overline{p}(doc, word)$.

8.2 Methods Compared

Several methods were compared with respect to both homogeneous and heterogeneous embeddings of words and documents.

14. Available from `http://kdd.ics.uci.edu`.
15. Data set available from `http://ai.stanford.edu/~gal/`.

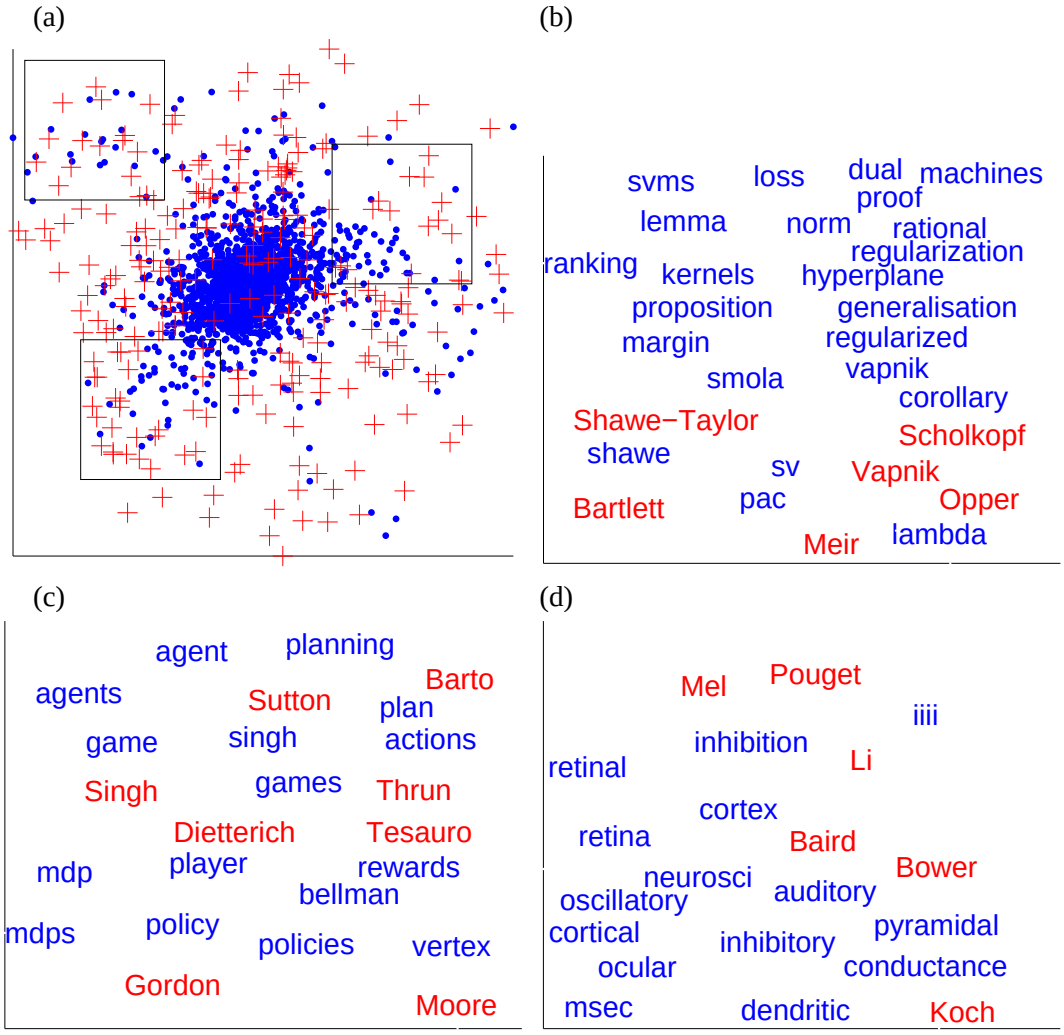

Figure 7: CODE embedding of 2000 words and 250 authors from the NIPS database (the 250 authors with highest word counts were chosen; words were selected as in Figure 5). The top left panel shows embeddings for authors (red crosses) and words (blue dots). Other panels show embedded authors (only first 100 shown) and words for the areas specified by rectangles (words in blue font, authors in red). They can be seen to correspond to learning theory (b), control and reinforcement learning (c) and neuroscience (d).

- **Co-Occurrence Data Embedding (CODE)**. Modeled the distribution of words and documents using the conditional word-given-document model $p_{CM}(doc, word)$ of Equation 4. Models other than p_{CM} are compared in Section 8.5.

- **Correspondence Analysis (CA)**. Applied the CA method to the matrix $\bar{p}(doc, word)$. Appendix A gives a brief review of CA.

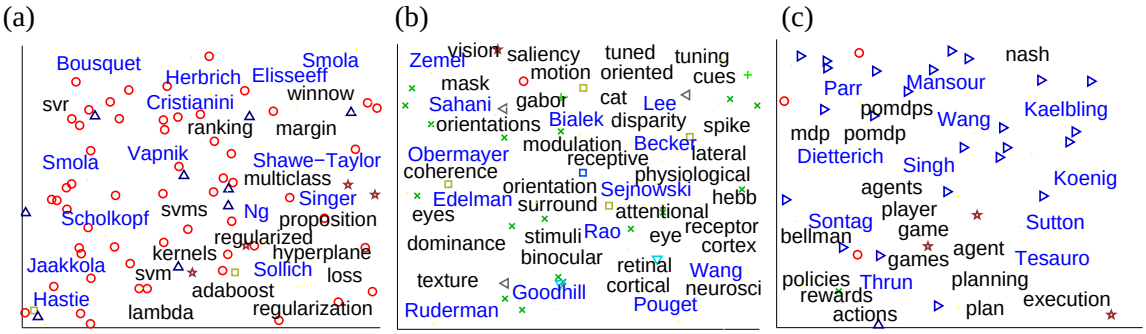

Figure 8: Embeddings of authors, words, and documents as described in Section 7.2. Words are shown in black and authors in blue (author names are capitalized). Only documents with known topics are shown. The representation of topics is as in Figure 5. We used 250 authors and 2000 words, chosen as in Figures 5 and 7. The three figures show insets of the complete embedding, which roughly correspond to the insets in Figure 6. **(a)** The border region between Algorithms and Architectures (AA) and Learning Theory (LT). **(b)** The border region between Neuroscience NS and Biological Vision (VB). **(c)** Control and Reinforcement Learning (CN) region.

- **Singular value decomposition (SVD)**. Applied SVD to two count-based matrices: $\bar{p}(doc, word)$ and $\log(\bar{p}(doc, word)+1)$. Assume the SVD of a matrix P is given by $P = USV^T$ (where S is diagonal with eigenvalues sorted in a decreasing order). Then the document embedding was taken to be $U\sqrt{S}$. Embeddings of dimension q were given by the first q columns of $U\sqrt{S}$. An embedding for words can be obtained in a similar manner, but was not used in the current evaluation.

- **Multidimensional scaling (MDS)**. MDS searches for an embedding of objects in a low dimensional space, based on a predefined set of pairwise distances (Cox and Cox, 1984). One heuristic approach that is sometimes used for embedding co-occurrence data using standard MDS is to calculate distances between row vectors of the co-occurrence matrix, which is given by $\bar{p}(doc, word)$ here. This results in an embedding of the row objects (documents). Column objects (words) can be embedded similarly, but there is no straightforward way of embedding both simultaneously. Here we tested two similarity measures between row vectors: The Euclidean distance, and the cosine of the angle between the vectors. MDS was applied using the implementation in the MATLAB Statistical Toolbox.

- **Isomap**. Isomap first creates a graph by connecting each object to m of its neighbors, and then uses distances of paths in the graph for embedding using MDS. We used the MATLAB implementation provided by the Isomap authors (Tenenbaum et al., 2000), with $m = 10$, which was the smallest value for which graphs were fully connected.

Of the above methods, only CA and CODE were used for joint embedding of words and documents. The other methods are not designed for joint embedding and were only used for embedding documents alone.

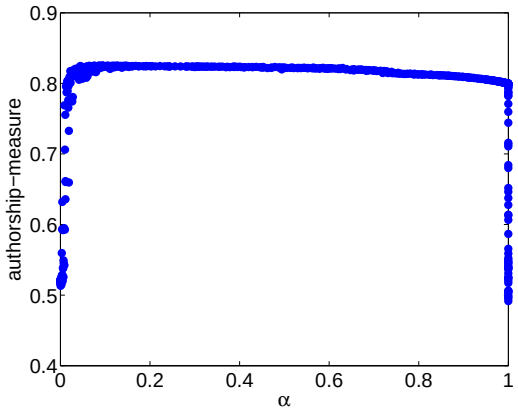

Figure 9: Evaluation of the authors-words-documents embedding for different likelihood weights. The X axis is a number α such that the weight on the $p_{CM}(doc, word)$ log-likelihood is $\frac{\alpha}{|word||doc|}$ and the weight on $p_{CM}(author, word)$ is $\frac{1-\alpha}{|author||doc|}$. The value $\alpha = 0.5$ results in equal weighting of the models after normalizing for size, and corresponds to the embedding shown in Figure 8. The Y axis is the authorship measure reflecting the quality of the joint document-author embedding.

All methods were also tested under several different normalization schemes, including TF/IDF weighting, and no document normalization. Results were consistent across all normalization schemes.

8.3 Quality Measures for Homogeneous and Heterogeneous Embeddings

Quantitative evaluation of embedding algorithms is not straightforward, since a ground-truth embedding is usually not well defined. Here we use the fact that documents are associated with class labels to obtain quantitative measures.

For the *homogeneous* embedding of the document objects, we define a measure denoted by *doc-doc*, which is designed to measure how well documents with identical labels are mapped together. For each embedded document, we measure the fraction of its neighbors that are from the same newsgroup. This is repeated for all neighborhood sizes,[16] and averaged over all documents and sizes, resulting in the *doc-doc* measure. The measure will have the value one for *perfect* embeddings where same topic documents are always closer than different topic documents. For a random embedding, the measure has a value of $1/(\sharp\text{newsgroups})$.

For the *heterogeneous* embedding of documents *and* words into a joint map, we defined a measure denoted by *doc-word*. For each document we look at its k nearest words and calculate their probability under the document's newsgroup.[17] We then average this over all neighborhood sizes of up to 100 words, and over all documents. It can be seen that the *doc-word* measure will be high if documents are embedded near words that are common in their class. This implies that by looking at the words close to a given document, one can infer the document's topic. The *doc-word* measure

16. The maximum neighborhood size is the number of documents per topic.
17. This measure was normalized by the maximum probability of any k words under the given newsgroup, so that it equals one in the optimal case.

could only be evaluated for CODE and CA since these are the only methods that provided joint embeddings.

8.4 Results

Figure 10 (top) illustrates the joint embedding obtained for the CODE model $p_{CM}(doc, word)$ when embedding documents from three different newsgroups. It can be seen that documents in different newsgroups are embedded in different regions. Furthermore, words that are indicative of a newsgroup topic are mapped to the region corresponding to that newsgroup.

To obtain a quantitative estimate of homogeneous document embedding, we evaluated the *doc-doc* measure for different embedding methods. Figure 11 shows the dependence of this measure on neighborhood size and embedding dimensionality, for the different methods. It can be seen that CODE is superior to the other methods across parameter values.

Table 1 summarizes the *doc-doc* measure results for all competing methods for seven different subsets.

Newsgroup Sets	CODE	Isomap	CA	MDS-e	MDS-c	SVD	SVD-l
`comp.os.ms-windows.misc`, `comp.sys.ibm.pc.hardware`	68*****	65	56	54	53	51	51
`talk.politics.mideast`, `talk.politics.misc`	85*****	83	66	45	73	52	52
`alt.atheism, comp.graphics`, `sci.crypt`	66*****	58	52	53	62	51	51
`comp.graphics`, `comp.os.ms-windows.misc`	76	77*****	55	55	53	56	56
`sci.crypt, sci.electronics`	84*****	83	83	65	58	56	56
`sci.crypt, sci.electronics`, `sci.med`	82*****	77	76	51	53	40	50
`sci.crypt, sci.electronics`, `sci.med, sci.space`	73*****	65	58	29	50	31	44

Table 1: *doc-doc* measure values (times 100) for embedding of seven newsgroups subsets. Average over neighborhood sizes $1,\ldots,1000$. Embedding dimension is $q = 2$. "MDS-e" stands for Euclidean distance, "MDS-c" for cosine distance, "SVD-l" preprocesses the data with $\log(count+1)$. The best method for each set is marked with an asterisk (*****).

To compare performance across several subsets, and since different subsets have different inherent "hardness", we define a normalized measure of purity that rescales the *doc-doc* measure performance for each of the 7 tasks. Results are scaled such that the best performing measure in a task has a normalized value of 1, and the one performing most poorly has a value of 0. As a result, any method that achieves the best performance consistently would achieve a normalized score of one. The normalized results are summarized in Figure 12a. CODE significantly outperforms other methods and IsoMap comes second.

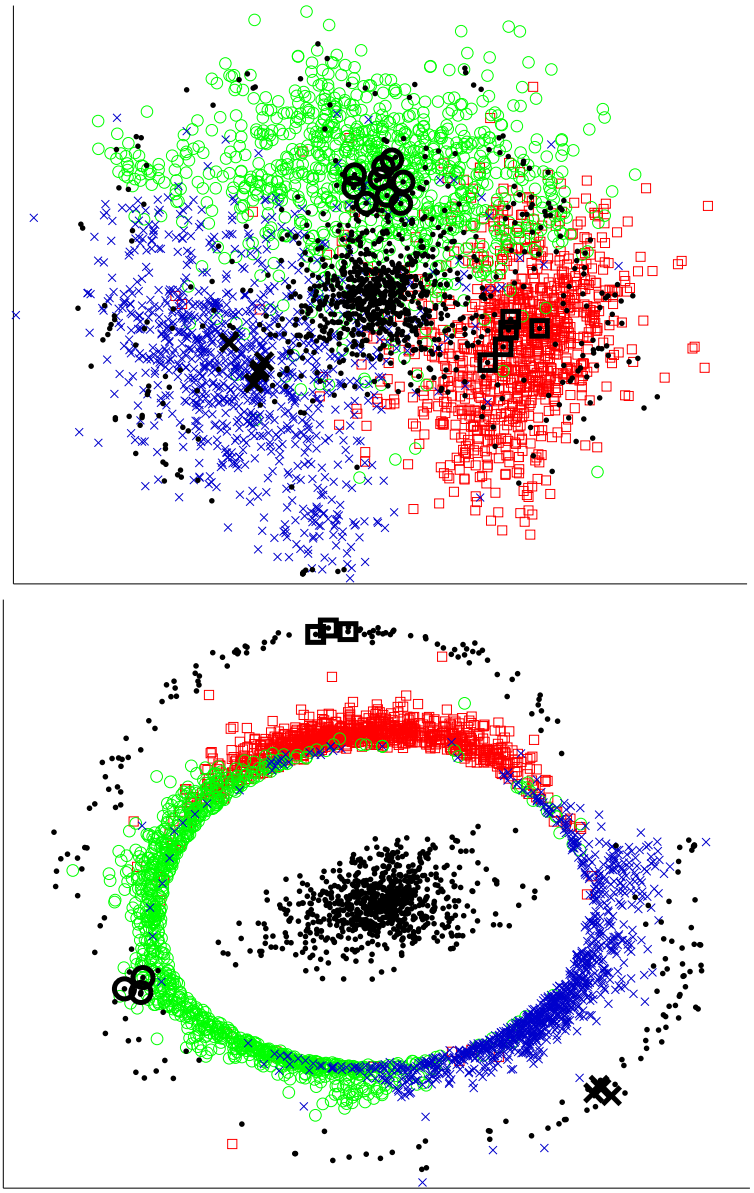

Figure 10: Visualization of two dimensional embeddings of the 20 newsgroups data under two different models. Three newsgroups are embedded: `sci.crypt` (red squares), `sci.electronics` (green circles) and `sci.med` (blue xs). **Top**: The embedding of documents and words using the conditional word-given-document model $p_{CM}(doc, word)$. Words are shown in black dots. Representative words around the median of each class are shown in black, with the marker shape corresponding to the class. They are $\{sick, hospital, study, clinical, diseases\}$ for `med`, $\{signal, filter, circuits, distance, remote, logic, frequency, video\}$ for `electronics`, and $\{legitimate, license, federal, court\}$ for `crypt`. **Bottom**: Embedding under the joint model $p_{MM}(doc, word)$. Representative words were chosen visually to be near the center of the arc corresponding to each class. Words are: $\{eat, AIDS, breast\}$ for `med`, $\{audio, noise, distance\}$ for `electronics`, and $\{classified, secure, scicrypt\}$ for `crypt`.

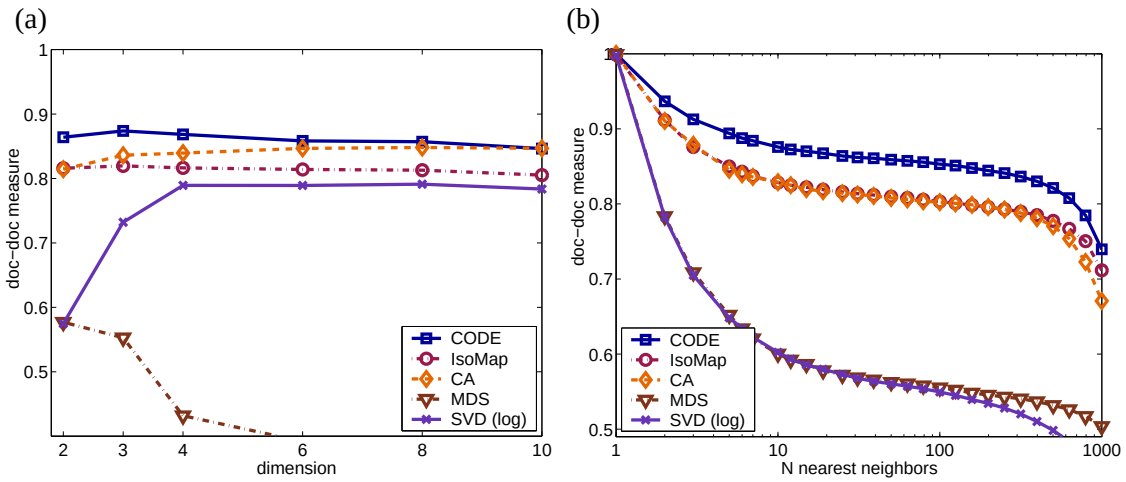

Figure 11: Parametric dependence of the *doc-doc* measure for different algorithms. Embeddings were obtained for the three newsgroups described in Figure 10. **(a)** *doc-doc* as a function of embedding dimensions. Average over neighborhood sizes $1,\ldots,100$. **(b)** *doc-doc* as a function of neighborhood size. Embedding dimension is $q = 2$

The performance of the heterogeneous embedding of words and documents was evaluated using the *doc-word* measure for the CA and CODE algorithms. Results for seven newsgroups are shown in Figure 12b, and CODE is seen to significantly outperform CA.

Finally, we compared the performance of the gradient optimization algorithm to the PSD-CODE method described in Section 5. Here we used a smaller data set because the number of the parameters in the PSD algorithm is quadratic in $|X|+|Y|$. Results for both the *doc-doc* and *doc-word* measures are shown in Figure 13, illustrating the effectiveness of the PSD algorithm, whose performance is similar the to non-convex gradient optimization scheme, and is sometimes even better.

8.5 Comparison Between Different Distribution Models

Section 2 introduced a class of possible probabilistic models for heterogeneous embedding. Here we compare the performance of these models on the 20 Newsgroup data set.

Figure 10 shows an embedding for the conditional model p_{CM} in Equation 3 and for the symmetric model p_{MM}. It can be seen that both models achieve a good embedding of both the relation between documents (different classes mapped to different regions) and document-word relation (words mapped near documents with relevant subjects). However, the p_{MM} model tends to map the documents to a circle. This can be explained by the fact that it also partially models the marginal distribution of documents, which is uniform in this case.

A more quantitative evaluation is shown in Figure 14. The figure compares various CODE models with respect to the *doc-doc* and *doc-word* measures. While all models perform similarly on the *doc-doc* measure, the *doc-word* measure is significantly higher for the two models $p_{MM}(doc,word)$ and $p_{CM}(doc,word)$. These models incorporate the marginals over words, and directly model the statistical dependence ratio $\overline{p}(x,y)$, as explained in Section 2. The model $p_{MC}(doc,word)$ does

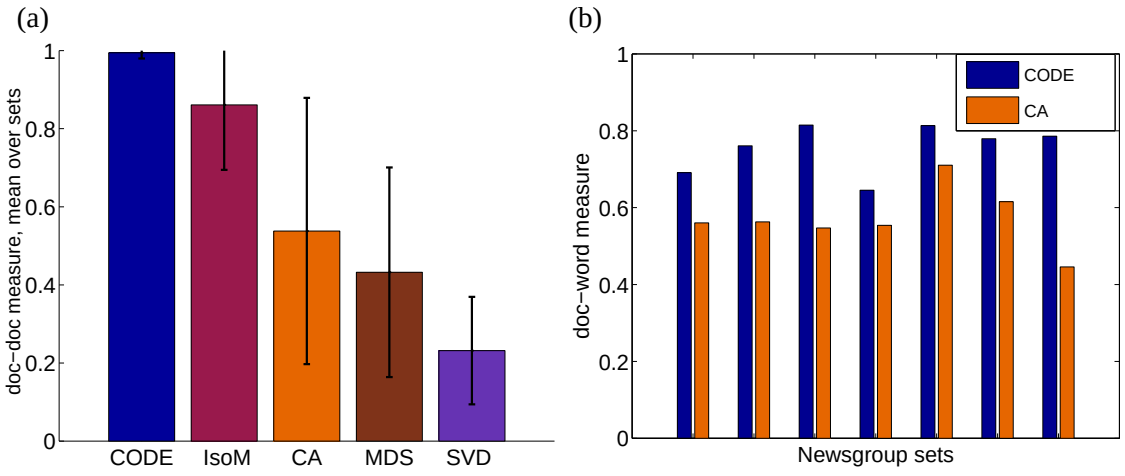

Figure 12: **(a)** *Normalized* doc-doc *measure* (see text) averaged over 7 newsgroup sets. Embedding dimension is $q = 2$. Sets are detailed in Table 1. Normalized *doc-doc* measure was calculated by rescaling at each data set, such that the poorest algorithm has score 0 and the best a score of 1. **b** The *doc-word* measure for the CODE and CA algorithms for the seven newsgroup sets. Embedding dimension is $q = 2$.

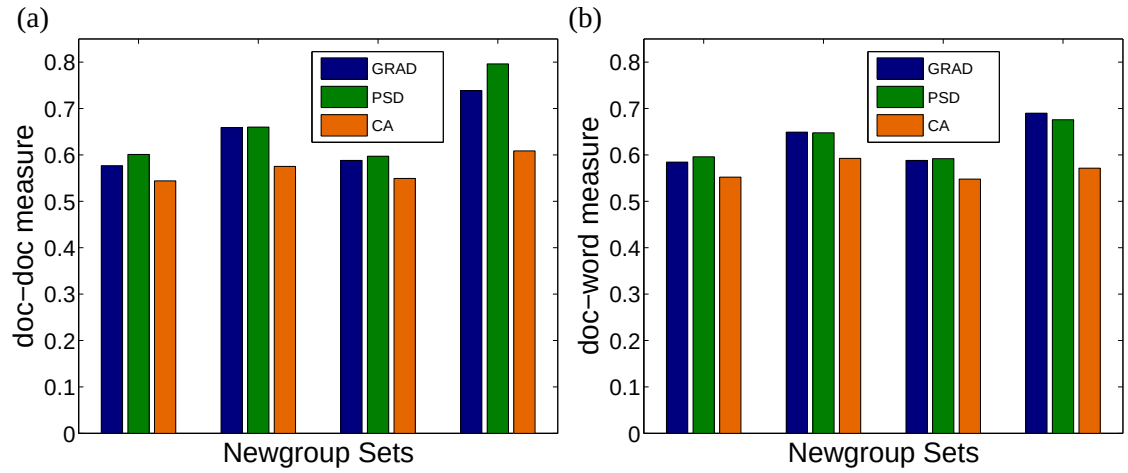

Figure 13: Comparison of the PSD-CODE algorithm with a gradient based maximization of the CODE likelihood (denoted by GRAD) and the correspondence analysis (CA) method. Both CODE methods used the $p_{CM}(doc, word)$ model. Results for $q = 2$ are shown for five newsgroup pairs (given by rows 1,2,4,5 in Table 1). Here 500 words were chosen, and 250 documents taken from each newsgroup. **a.** The *doc-doc* measure. **b.** The *doc-word* measure.

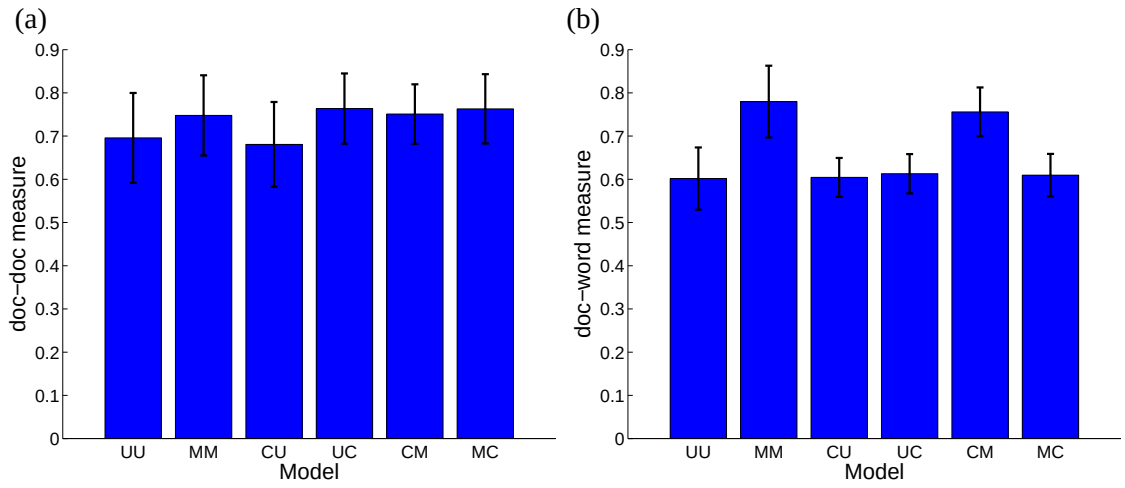

Figure 14: Comparison of different embedding models. Averaged results for the seven newsgroup subsets are shown for the *doc-doc* (left figure) and *doc-word* (right figure) measures. Model names are denoted by two letters (see Section 2.3), which reflect the treatment of the *document* variable (first letter) and *word* variable (second letter). Thus, for example *CM* indicates conditioning on the document variable, whereas *MC* indicates conditioning on the word variable.

not perform as well, presumably because it makes more sense to assume that the document is first chosen, and then a word is chosen given the document, as in the $p_{CM}(doc, word)$ model.

9. Discussion

We presented a method for embedding objects of different types into the same low dimension Euclidean space. This embedding can be used to reveal low dimensional structures when distance measures between objects are unknown or cannot be defined. Furthermore, once the embedding is performed, it induces a meaningful metric between objects of the same type. Such an approach may be used, for example, for embedding images based on accompanying text, and derive the semantic distance between images.

We showed that co-occurrence embedding relates statistical correlation to the local geometric structure of one object type with respect to the other. Thus the local geometry may be used for inferring properties of the data. An interesting open issue is the sensitivity of the solution to sample-to-sample fluctuation in the empirical counts. One approach to the analysis of this problem could be via the Fisher information matrix of the model.

The experimental results shown here focused mainly on the conditional based model of Equation 4. However, different models may be more suitable for data types that have no clear asymmetry.

An important question in embedding objects is whether the embedding is unique, namely, can there be two *different* optimal configurations of points. This question is related to the rigidity and uniqueness of graph embeddings, and in our particular case, complete bipartite graphs. A theorem of Bolker and Roth (1980) asserts that embeddings of complete bipartite graphs with at least 5

vertices on each side, are guaranteed to be rigid, that is they cannot be continuously transformed. This suggests that the CODE embeddings for $|X|, |Y| \geq 5$ are locally unique. However, a formal proof is still needed.

Co-occurrence embedding does not have to be restricted to distributions over pairs of variables, but can be extended to multivariate joint distributions. One such extension of CODE would be to replace the dependence on the pairwise distance $\|\phi(x) - \psi(x)\|$ with a measure of average pairwise distances between multiple objects. For example, given three variables X, Y, Z one can relate $p(x,y,z)$ to the average distance of $\phi(x), \psi(y), \xi(z)$ from their centroid $\frac{1}{3}(\phi(x) + \psi(y) + \xi(z))$. The method can also be augmented to use statistics of same-type objects when these are known, as discussed in Section 6.1.

An interesting problem in many embedding algorithms is generalization to new values. Here this would correspond to obtaining embeddings for values of X or Y such that $\bar{p}(x) = 0$ or $\bar{p}(y) = 0$, for instance because a word did not appear in the sample documents. When variables are purely categorical and there is no intrinsic similarity measure in either the X or Y domains, there is little hope for generalizing to new values. However, in some cases the X or Y variables may have such structure. For example, objects in X may be represented as vectors in $\mathbb{R}^p$. This information can help in generalizing embeddings, since if x_1 is close to x_2 in $\mathbb{R}^p$ it may be reasonable to assume that $\phi(x_1)$ should be close to $\phi(x_2)$. One strategy for applying this intuition is to model $\phi(x)$ as a continuous function of x, for instance a linear map Ax or a kernel-based map. Such an approach has been previously used to extend embedding methods such as LLE to unseen points (Zhang, 2007). This approach can also be used to extend CODE and it will be interesting to study it further. It is however important to stress that in many cases no good metric is known for the input objects, and it is a key advantage of CODE that it can produce meaningful embeddings in this setting.

These extensions and the results presented in this paper suggest that probability-based continuous embeddings of categorical objects could be applied efficiently and provide accurate models for complex high dimensional data.

Appendix A. A Short Review of Correspondence Analysis

Correspondence analysis (CA) is an exploratory data analysis method that embeds two variables X and Y into a low dimensional space such that the embedding reflects their statistical dependence (Greenacre, 1984). Statistical dependence is modeled by the ratio

$$q(x,y) = \frac{\bar{p}(x,y) - \bar{p}(x)\,\bar{p}(y)}{\sqrt{\bar{p}(x)\,\bar{p}(y)}}$$

Define the matrix Q such that $Q_{xy} = q(x,y)$. The CA algorithm computes an SVD of Q such that $Q = USV$ where S is diagonal and U, V are rectangular orthogonal matrices. We assume that the diagonal of S is sorted in descending order. To obtain the low dimensional embeddings, one takes the first q columns and rows of $P_x^{-0.5} S\sqrt{U}$ and $P_y^{-0.5} S\sqrt{U}$ respectively, where P_x, P_y are diagonal matrices with $\bar{p}(x)$, $\bar{p}(y)$ on the diagonal. It can be seen that this procedure corresponds to a least squares approximation of the matrix Q via a low dimensional decomposition. Thus, CA cannot be viewed as a statistical model of $\bar{p}(x,y)$, but is rather an L_2 approximation of empirically observed correlation values.

The ratio $q(x,y)$ is closely related to the chi-squared distance between distributions, and there indeed exist interpretations (Greenacre, 1984) of CA which relate it to approximating this distance.

Also, as mentioned in Section 4, it can be shown (Hill, 1974) that CA corresponds to Canonical Correlation Analysis when X and Y are represented via indicator vectors. For example, $X = 3$ is represented as a vector $\mathbf{e} \in \mathbb{R}^{|X|}$ such that $\mathbf{e}(3) = 1$ and all other elements are zero.

References

F. R. Bach and M. I. Jordan. Kernel independent component analysis. *Journal of Machine Learning Research*, 3:1–48, 2002.

A. L. Berger, S.A. Della Pietra, and V.J. Della Pietra. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–71, 1996.

D. P. Bertsekas. On the Goldstein-Levitin-Polyak gradient projection method. *IEEE Transactions on Automatic Control*, 21:174–184, 1976.

E.D. Bolker and B. Roth. When is a bipartite graph a rigid framework? *Pacific Journal of Mathematics*, 90:27–44, 1980.

S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge Univ. Press, 2004.

M. Chalmers and P. Chitson. Bead: explorations in information visualization. In *Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 330–337. ACM Press, New York, NY, 1992.

G. Chechik and N. Tishby. Extracting relevant structures with side information. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems 15*, pages 857–864. MIT Press, Cambridge, MA, 2003.

M. Chiang. Geometric programming for communication systems. *Foundations and Trends in Communications and Information Theory*, 2(1):1–154, 2005.

T.M. Cover and J.A Thomas. *Elements of Information Theory*. Wiley-Interscience, New York, 1991.

T. Cox and M. Cox. *Multidimensional Scaling*. Chapman and Hall, London, 1984.

M. Fazel, H. Hindi, and S. P. Boyd. A rank minimization heuristic with application to minimum order system approximation. In *Proceedings of the American Control Conference*, volume 6, pages 4734–4739. American Automatic Control Council, New York, 2001.

R.A. Fisher. The precision of discriminant functions. *Annals of Eugenics, London*, 10:422–429, 1940.

A. Globerson and N. Tishby. Sufficient dimensionality reduction. *Journal of Machine Learning Research*, 3:1307–1331, 2003.

A. Globerson, G. Chechik, F. Pereira, and N.Tishby. Euclidean embedding of co-occurrence data. In L. K. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 497–504. MIT Press, Cambridge, MA, 2005.

M.J. Greenacre. *Theory and Applications of Correspondence Analysis*. Academic Press, London, 1984.

J.H. Ham, D.D. Lee, and L.K. Saul. Learning high dimensional correspondences with low dimensional manifolds. In *Proceedings of the 20th International Conference on Machine Learning. Workshop on The Continuum from Labeled to Unlabeled Data in Machine Learning and Data Mining*, pages 34–41, 2003.

M.O. Hill. Correspondence analysis: A neglected multivariate method. *Applied Statistics*, 23(3): 340–354, 1974.

G. Hinton and S.T. Roweis. Stochastic neighbor embedding. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems 15*, pages 833–840. MIT Press, Cambridge, MA, 2003.

T. Hofmann. Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning*, 42(1):177–196, 2001.

H. Hotelling. The most predictable criterion. *Journal of Educational Psychology*, 26:139–142, 1935.

T. Iwata, K. Saito, N. Ueda, S. Stromsten, T. Griffiths, and J. Tenenbaum. Parametric embedding for class visualization. In L. K. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems 17*. MIT Press, Cambridge, MA, 2005.

P.L. Lai and C. Fyfe. Kernel and nonlinear canonical correlation analysis. In *International Joint Conference on Neural Networks*, pages 365–378. IEEE Computer Society, Los Alamitos, CA, 2000.

D. Lee and H. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999.

X. Lin. Map displays for information retrieval. *Journal of the American Society for Information Science*, 48(1):40–54, 1997.

G. Mei and C. R. Shelton. Visualization of collaborative data. In R. Dechter and T. Richardson, editors, *Proceedings of the Twenty-Second International Conference on Uncertainty in Artificial Intelligence*, pages 341–348. AUAI Press, Arlington, VA, 2006.

G. Michailidis and J. de Leeuw. The Gifi system of descriptive multivariate analysis. *Statistical Science*, 13(4):307–336, 1998.

R. B. Nelsen. *An Introduction to Copulas*. Springer, New York, 1999.

V.Y. Pan and Z.Q. Chen. The complexity of the matrix eigenproblem. In *Proceedings of the Thirty-First Annual ACM Symposium on Theory of Computing*, pages 507–516. ACM Press, New York, NY, 1999.

S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290:2323–2326, 2000.

S.T. Roweis. NIPS 0-12 data. http://www.cs.toronto.edu/∼roweis/data.html, 2000.

J.B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290:2319–2323, 2000.

L. Vandenberghe and S. Boyd. Semidefinite programming. *SIAM Review*, 38(1):49–95, 1996.

K. Q. Weinberger and L. K. Saul. Unsupervised learning of image manifolds by semidefinite programming. *International Journal of Computer Vision*, 70(1):77–90, 2006.

E. Xing, A. Ng, M. Jordan, and S. Russell. Distance metric learning, with application to clustering with side-information. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems 15*, pages 505–512. MIT Press, Cambridge, MA, 2002.

S. Yan D. Xu B. Zhang H.J. Zhang. Graph embedding: a general framework for dimensionality reduction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 40–51, 2007.

H. Zhong, J. Shi, and M. Visontai. Detecting unusual activity in video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 819–826. IEEE Computer Society, Los-Alamitos, CA, 2004.