Image Classification from Small Sample, with Distance Learning and Feature Selection

Daphna Weinshall and Lior Zamir

School of Computer Science and Engineering, The Hebrew University of Jerusalem, Jerusalem, Israel 91904

Abstract. Small sample is an acute problem in many application domains, which may be partially addressed by feature selection or dimensionality reduction. For the purpose of distance learning, we describe a method for feature selection using equivalence constraints between pairs of datapoints. The method is based on $L1$ regularization and optimization. Feature selection is then incorporated into an existing non-parametric method for distance learning, which is based on the boosting of constrained generative models. Thus the final algorithm employs dynamical feature selection, where features are selected anew in each boosting iteration based on the weighted training data. We tested our algorithm on the classification of facial images, using two public domain databases. We show the results of extensive experiments where our method performed much better than a number of competing methods, including the original boosting-based distance learning method and two commonly used Mahalanobis metrics.

Keyword: Feature Selection, Distance Learning, Small Sample, L1 regularization.

1 Introduction

A distance (or inverse similarity) function, defined for every pair of datapoints, is a useful way to describe data. It is also a useful way to transfer knowledge between related classes, and thus address the problems of small sample and even one-shot learning (with only one example per new class). Distances can be directly used for unsupervised clustering - as do spectral methods for example, or for supervised classification - as in nearest neighbor classification. An important special case is the family of kernel functions, which can also be used to enhance a variety of kernel-based classification algorithms (such as kernel-SVM). Here we are interested in the problem of learning a distance function from small sample, when the training data is a set of equivalence constraints on pairs of datapoints, indicating whether the pair originated from the same or different sources.

The problem of small sample is ubiquitous in application domains such as computer vision and bioinformatics, where data points may be initially represented in some high dimensional feature space, while the number of training examples is typically much smaller than the feature space dimensionality. It may lead to

G. Bebis et al. (Eds.): ISVC 2007, Part II, LNCS 4842, pp. 106–115, 2007.

problems of over-fitting and poor generalization. One common way to address this problem is to reduce dimensionality, or dramatically prune the dataspace via feature selection. We focus here on the second avenue of feature selection. We describe below a method for feature selection based on equivalence constraints, and incorporate the method into an existing non-parametric method for distance learning [10].

Our feature selection method, described in Section 2.1, relies on the use of the $L1$ norm in the evaluation of the cost function. There has been much recent work on the use of concave cost functions in order to achieve sparse signal decomposition [6], and $L1$ is probably the simplest such norm. Feature selection with $L1$ regularization has been studied before, as in the lasso method [17], but see also [19,13]. Unlike these methods here we use both $L1$ optimization and regularization, i.e., the loss function defined over the training data of equivalence constraints is also defined in terms of the $L1$ norm. This choice leads to a sparse solution over the set of constraints. Thus, in a typical solution some constraints are fully satisfied, while others may deviate largely from the target value; this seems desirable given the discrete nature of equivalence constraints.

Distance learning from equivalence constraints has been studied extensively in recent years. Much of this work focused on the learning of the (linear) Mahalanobis metric, as in [15,5,4,3,9], where feature selection is often done implicitly or explicitly as part of the learning procedure. We use here a more powerful non-parametric distance function learning algorithm [10] based on boosting. The weak learner in this algorithm computes in a semi-supervised manner a generative constrained Gaussian Mixture Model [14] to describe the data. But here lies the problem with small sample: in each iteration, the GMM algorithm can only work in a rather low dimensional space as it must estimate a number of covariance matrices defined over this space. Currently, the problem is solved by projecting the data initially into a low-dimensional space where the weak learner has a chance of working properly. With a very small sample, this one-time dimensionality reduction may be catastrophical, and lead to poor results. We therefore propose to compute the dimensionality reduction afresh in each iteration of the boosting method, see Section 2.2.

In Section 3 we describe extensive experimental results on facial image classification, where very significant improvement is obtained. In our experiments we used a single pair from each class of objects (each individual face), which is an instance of 'one shot learning' - how to learn a classifier from one example of a new class. Distance learning offers one way to approach this problem. Our results show that our algorithm performs better than a number of alternative distance learning methods. Other approaches have been developed recently in the context of object and class recognition, see for example [12,2,7,16,11]. Also note that in our method, we use feature selection to enhance the performance of the weak learner in each boosting iteration as in [18]. In a very different approach, boosting is used in [1] to select embeddings in the construction of a discriminative classifier.

2 Distance Learning Algorithm

As noted in the introduction, we use a non-parametric distance learning method [10] based on boosting, where in each iteration a generative Gaussian Mixture Model is constructed using a set of weighted equivalence constraints. This weak learner estimates a number of covariance matrices from the training sample. Thus, with small sample it must be used in a low dimensional space, which can be obtained via dimensionality reduction or feature selection (or both), see Section 2.1. Our final algorithm selects features dynamically - different features in each boosting iteration, as described in Section 2.2.

2.1 Feature Selection with $L1$ Optimization and Regularization

Notations. Given a set of data points, equivalence constraints over pairs of points consist of *positive constraints* - denoting points which come from the same source, and *negative constraints* - denoting points from different sources. Let p_1 and p_2 denote two $1 \times n$ data points. Let $\Delta^+ = p_1 - p_2$ denote the vector difference between two positively constrained points, $\Delta^- = p_1 - p_2$ denote the difference between two negatively constrained points, and W_{Δ^+}, W_{Δ^-} denote the weight of constraint Δ^+ and Δ^- respectively. Let A denote an $n \times n$ diagonal matrix whose $i-$th diagonal element is denoted A_i, N denote the number of features to be selected, and μ_N denote a regularization parameter which controls the sparseness of A as determined by N.

Problem Formulation. Feature selection is obtained using $L1$ regularization and optimization, which favors sparsity in both the features selection matrix A, and the set of satisfied constraints. The latter property implies a certain "slack" in the constraint satisfaction, where some constraints are fully satisfied while others behave like outliers; this seems desirable, given the discrete nature of equivalence constraints. Thus we define an optimization problem which will be solved (in its most general form) by linear programming. We define two variants of the optimization problem, with some different characteristics (as will be discussed shortly):

LP1: Linear Program version 1

$$F = \min_A \Big(\sum_{\Delta^+} W_{\Delta^+} \parallel \Delta^+ \cdot A \parallel_1 - \sum_{\Delta^-} W_{\Delta^-} \parallel \Delta^- \cdot A \parallel_1 + \mu_N |\sum_i A_i| \Big)$$
$$\text{s.t.} \quad A = diag[A_i], \quad 0 \le A_i \le 1 \tag{1}$$

Given the constraints on A_i, the following derivation holds:

$$F = \min_A \Big(\sum_{\Delta^+} W_{\Delta^+} \sum_{i=1}^{n} |\Delta_i^+| A_i - \sum_{\Delta^-} W_{\Delta^-} \sum_{i=1}^{n} |\Delta_i^-| A_i + \mu_N \sum_i A_i \Big)$$
$$= \min_A \Big(\sum_{i=1}^{n} A_i \Big(\sum_{\Delta^+} W_{\Delta^+} \Delta_i^+ - \sum_{\Delta^-} W_{\Delta^-} \Delta_i^- + \mu_N \Big) \Big)$$

$$= \min_A \sum_{i=1}^{n} A_i w_i, \qquad \text{s.t. } 0 \le A_i \le 1 \tag{2}$$

$$\text{where } w_i = \sum_{\Delta^+} W_{\Delta^+} \Delta_i^+ - \sum_{\Delta^-} W_{\Delta^-} \Delta_i^- + \mu_N$$

Clearly minimizing (2) gives a solution where $A_i = 0$ if $w_i > 0$, and $A_i = 1$ if $w_i < 0$. The regularization parameter μ_N thus determines exactly how many coordinates A_i will have the solution 1, or in other words, exactly how many features will be selected. Given that we want to select N features, the optimal solution is obtained by sorting the coefficients w_i, and then setting $A_i = 1$ for the N smallest coordinates, and $A_i = 0$ otherwise.

LP2: Linear Program version 2

$$F = \min_A \Big(\sum_{\Delta^+} W_{\Delta^+} \parallel \Delta^+ \cdot A \parallel_1 + \sum_{\Delta^-} W_{\Delta^-} \mid 1 - \parallel \Delta^- \cdot A \parallel_1 \mid + \mu_N \mid N - \sum_i A_i \mid \Big)$$

$$\text{s.t. } A = diag[A_i], \quad 0 \le A_i \le 1 \tag{3}$$

This defines a linear programming optimization problem, where $A_i > 0$ implies that feature i is selected (possibly with weight A_i).

Parameters: The only free parameter in both versions is the number of features to be selected. Weights are given by the boosting mechanism, and μ_N is either determined uniquely by N (in version 1, following the above derivation), or is set to a very large constant (in version 2).

RND: Random feature selection. To evaluate the significance of the way features are selected, we tested a third method, whereby N features are randomly selected. This would typically lead to very poor performance in the original high dimensional feature space, and it was therefore preceded by PCA dimensionality reduction, followed by projecting the data onto the M most significant principal components. This gave reasonable performance, which critically depended on the value of M (see Fig. 6). In later comparisons this value was chosen empirically, so as to allow optimal performance for this method.

2.2 The Final Distance Learning Algorithm

The final algorithm is shown in Alg. 1, where modifications from the original distBoost algorithm [10] are highlighted in boldface. In the algorithm's description, we denote by $W_{i_1 i_2}^t$ the weight $W_{\Delta^{+/-}}$ at iteration t of constraint $\Delta^{+/-} = (p_{i_1} - p_{i_2})$.

3 Experimental Results

3.1 Methods

Data and methodology: We used two public domain datasets: (i) The class of faces from the Caltech dataset (Faces 1999 from

Algorithm 1. Distance learning with feature selection

Input:

Data points: $(p_1, ..., p_n)$, $p_k \in \mathcal{R}^n$
A set of equivalence constraints: (p_{i_1}, p_{i_2}, y_i), where $y_i \in \{-1, 1\}$
Unlabeled pairs of points: $(p_{i_1}, p_{i_2}, y_i = *)$, implicitly defined by all unconstrained pairs of points

- Initialize $W^1_{i_1 i_2} = 1/(n^2)$ $i_1, i_2 = 1, \ldots, n$ (weights over pairs of points)
  $w_k = 1/n$ $k = 1, \ldots, n$ (weights over data points)
- For $t = 1, .., T$
  1. **Given the original data vectors in $\mathcal{R}^n$, obtain a lower dimensional description in $\mathcal{R}^D$ as follows:**
     - **To balance the effect of positive and negative constraints, normalize the weights such that:** $\sum_{\Delta^+ = (p_{i_1} - p_{i_2})} W^t_{i_1 i_2} = 1, \quad \sum_{\Delta^- = (p_{i_1} - p_{i_2})} W^t_{i_1 i_2} = 1.$
     - **Select $N \geq D$ features using one of the methods described in Section 2.1.**
     - **Given only the selected features, reduce the data dimensionality to D with PCA, and obtain $x_i = G_t(p_i)$.**
  2. As in [14], fit a constrained GMM (weak learner) on weighted data points x_i in $\mathcal{X} = \mathcal{R}^D$ using the equivalence constraints.
  3. As in [10], generate a weak hypothesis function $\tilde{h}_t : \mathcal{X} \times \mathcal{X} \to [0, 1]$ and define a weak distance function as $h_t(x_i, x_j) = \frac{1}{2}(1 - \tilde{h}_t(x_i, x_j)) \in [0, 1]$.
  4. Compute $r_t = \sum_{(x_{i_1}, x_{i_2}, y_i = \pm 1)} W^t_{i_1 i_2} y_i \tilde{h}_t(x_{i_1}, x_{i_2})$, only over **labeled** pairs.
     Accept the current hypothesis only if $r_t > 0$.
  5. Choose the hypothesis weight $\alpha_t = \frac{1}{2} \ln(\frac{1+r_t}{1-r_t})$.
  6. Update the weights of **all** points in $\mathcal{X} \times \mathcal{X}$ as follows:

$$W^{t+1}_{i_1 i_2} = \begin{cases} W^t_{i_1 i_2} \exp(-\alpha_t y_i \tilde{h}_t(x_{i_1}, x_{i_2})) & y_i \in \{-1, 1\} \\ W^t_{i_1 i_2} \exp(-\lambda * \alpha_t) & y_i = * \end{cases}$$

     where λ is a tradeoff parameter that determines the decay rate of the unlabeled points in the boosting process.
  7. Normalize: $W^{t+1}_{i_1 i_2} = \frac{W^{t+1}_{i_1 i_2}}{\sum_{i_1, i_2 = 1}^{n} W^{t+1}_{i_1 i_2}}$
  8. Translate the weights from $\mathcal{X} \times \mathcal{X}$ to $\mathcal{R}^n$: $w^{t+1}_k = \sum_j W^{t+1}_{kj}$

Output: A final distance function $D(p_i, p_j) = \sum_{t=1}^{T} \alpha_t h_t(G_t(p_i), G_t(p_j))$

`http://www.vision.caltech.edu/archive.html`), which contains images of individuals with different lighting, expressions and backgrounds. (ii) The YaleB dataset [8], which contains images of individuals under different illumination conditions. Examples are shown in Fig. 1. Each image was represented by its pixel gray-values, 28×28 in the Caltech dataset and 128×112 in the YaleB dataset. 19 classes (or different individuals) were used in both datasets, with

Fig. 1. Left: three pictures of the same individual from the YaleB database, which contains 1920 frontal face images of 30 individuals taken under different lighting conditions. Right: two images from the Caltech dataset.

training data composed of two examples randomly sampled from each class in each experiment. This generated 19 positive equivalence constraints, and a larger number of negative constraints. Our experiments indicated that it was sufficient to use a subset of roughly 19 negative constraints (equal to the number of positive constraints) to achieve the best performance. The test set included 20 new random images from each of the 19 classes.

Performance - evaluation and measures: Performance of each algorithm was evaluated on the test set only, using the Equal Error Rate (EER) of the ROC curve. This curve was obtained by parameterizing the distance threshold t: points at distance lower than t were declared 'positive' (or same class), while others were declared 'negative' (or different class). The EER is the point where the miss (false negative) rate equals the false positive rate. In some experiments we used the learnt distance to perform clustering of the test datapoints with the Ward agglomerative clustering algorithm. We then measured performance using the $F_{\frac{1}{2}} = \frac{2PR}{P+R}$ score, where P denotes precision rate and R denotes recall rate. Another performance measure used the k-nearest-neighbor classification, where for each point - the k nearest points are retrieved, and classification is done in agreement with the majority.

Algorithms: We evaluated three variants of Algorithm 1, using one of the three selection methods described in Section 2.1; accordingly, they are denoted below as *LP1*, *LP2* and *RND*. For comparison we used the following distance measures:

1. *Euclidean* metric in the original feature space.
2. *RCA* [15] - a Mahalanobis metric learning algorithm.
3. *DMC* [5] - a Mahalanobis metric learning algorithm.
4. *Euclidean FS* - the Euclidean metric in the lower dimensional space, obtained using feature selection as described in Section 2.1.
5. *DB-PCA* - the original distBoost algorithm, where dimensionality is initially reduced by projecting the data onto the D largest principal components.

Algorithm parameters: We used 150 boosting rounds for all methods (*LP1*, *LP2*, *RND*, *DB-PCA*). The number of features N to be selected in each iteration was 80 for the Caltech dataset, and 20 for the YaleB dataset. The dimension D subsequently used by the weak learner was chosen to be the maximal possible.

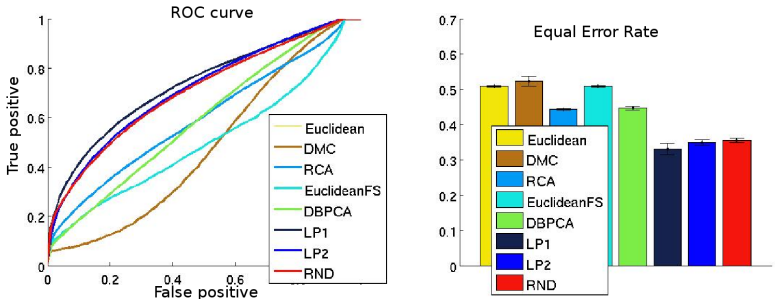

Fig. 2. Results using the Caltech facial image dataset. Left: the ROC curve of eight distance measures. Right: summary of the Equal Error Rate of the ROC curves (left) with standard error (ste) bars, for all algorithms.

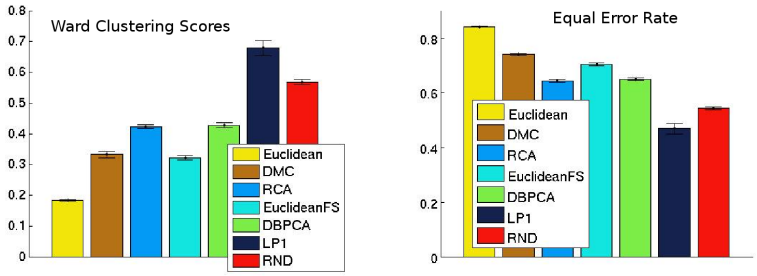

Fig. 3. Results using the YaleB facial image dataset, including clustering results (with the Ward algorithm) measured by $F_{\frac{1}{2}}$ on the left, and the summary of the Equal Error Rate of the ROC curves for all algorithms on the right. Ste bars are also shown.

3.2 Results

The performance of the various algorithms in the different experiments is summarized in Figs. 2, 3. Fig. 4 provides visualization of the features (or pixels) selected by the algorithm, while Fig. 5 shows the dependency of the results on the number of features which have been selected. Fig. 6 shows the behavior of the different algorithms as a function of the boosting iteration. Finally, Fig. 7 shows performance evaluation on totally new classes, including faces of individuals that the algorithms have never seen before.

3.3 Discussion

The results in Figs. 2, 3 clearly show that feature selection improves performance significantly. (Note that although the errors may appear relatively high, given the difficulty of the task - with so few training examples - these are state-of-the-art results.) Moreover, Fig. 7 shows that this advantage is maintained even when

Fig. 4. Visualization of the features selection process using the YaleB dataset. All the features selected by the algorithm up to a certain iteration are shown in white. From left to right, we show iteration 4, 20, and 150 respectively.

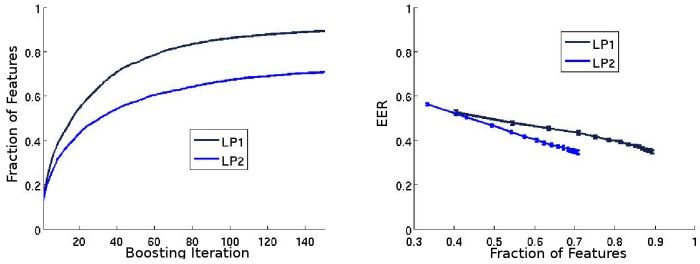

Fig. 5. Caltech facial image dataset. Left: the fraction of features (out of 784 image pixels) used by the hypotheses of the *LP1* and *LP2* methods as a function of the boosting iteration number. Right: EER scores as a function of the fraction of the total number of features selected by the *LP1* and *LP2* methods.

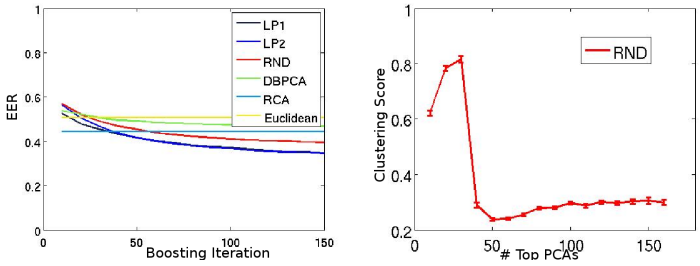

Fig. 6. Caltech facial image dataset: Left: EER scores as a function of the number of iterations. Right: Clustering performance of method RND as a function of the number of top principal components chosen for representation of the data (M).

presented with completely novel classes (a learning to learn scenraio), which is one of the important motivations for learning distance functions instead of classifiers. When comparing the two main feature selection variants - *LP1* and *LP2* (see definition in Section 2.1), we see comparable performance. But recall that *LP1* has a tremendous computational advantage, since its only computation involves sorting and it does not need to solve a linear programming problem. Thus *LP1*, with slightly better performance, is clearly preferable. The third

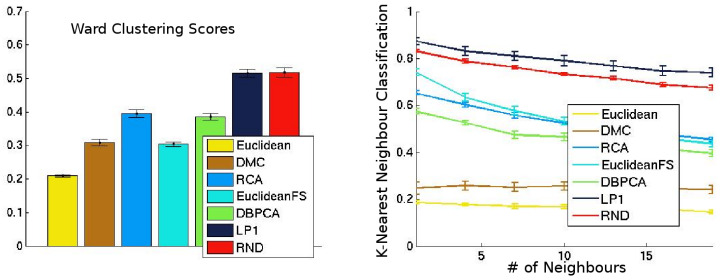

Fig. 7. Results on YaleB when tested on 10 totally novel classes - individuals that the algorithms have never seen before. Left: $F_{\frac{1}{2}}$ clustering scores. Right: K-nearest-neighbor classification scores.

variant - *RND* - usually performs overall less well. More importantly, its good performance critically depends on the number of features used for the random sampling (see Fig. 6-right), an unknown parameter apriori, which makes this variant even less appealing.

Both *LP1* and *LP2* do not use all the original features. This is most clearly seen in the results with YaleB dataset, where only 12% of the initial features are used, as is visualized in Fig. 4. This may give them yet another significant advantage over alternative distance learning methods in some applications, where features are expensive to compute. In this respect, the advantage of the *LP2* variant is more pronounced, as shown in Fig. 5. Finally, all the results above are reported for very small samples - with 2 examples from each of 19 classes. The relative advantage of the feature selection variants disappears when the sample increases.

4 Summary

We described a distance learning method which is based on boosting combined with feature selection, using equivalence constraints on pairs of datapoints. In a facial image classification task and with very few training examples, the new method performs much better than the alternative algorithms we have tried. The underlying reason may be that feature selection combined with boosting allows the distance learning algorithm to look at more features in the data, while being able to estimate only a small number of parameters in each round. Thus, within this very difficult domain of image classification from very small sample (or learning to learn), our algorithm achieves the goal of advancing the state-of-the-art.

References

1. Athitsos, V., Alon, J., Sclaroff, S., Kollios, G.: BoostMap: A method for efficient approximate similarity rankings. In: Proc. CVPR (2004)
2. Bart, E., Ullman, S.: Cross-generalization: learning novel classes from a single example by feature replacement. In: Proc. CVPR, pp. 672–679 (2005)

3. Bilenko, M., Basu, S., Mooney, R.J.: Integrating constraints and metric learning in semi-supervised clustering. In: ACM International Conference Proceeding Series (2004)
4. Chang, H., Yeung, D.Y.: Locally linear metric adaptation for semi-supervised clustering. In: ACM International Conference Proceeding Series (2004)
5. De Bie, T., Momma, M., Cristianini, N.: Efficiently learning the metric with side-information. In: Gavaldá, R., Jantke, K.P., Takimoto, E. (eds.) ALT 2003. LNCS (LNAI), vol. 2842, pp. 175–189. Springer, Heidelberg (2003)
6. Donoho, D.L., Elad, M.: Optimally sparse representation in general (nonorthogonal) dictionaries via l1 minimization. Proceedings of the National Academy of Sciences 100(5), 2197–2202 (2003)
7. Ferencz, A., Learned-Miller, E., Malik, J.: Building a classification cascade for visual identification from one example. In: Proc. ICCV, pp. 286–293 (2005)
8. Georghiades, A.S., Belhumeur, P.N., Kriegman, D.J.: From few to many: generative models for recognition under variable pose and illumination. In: IEEE Int. Conf. on Automatic Face and Gesture Recognition, pp. 277–284 (2000)
9. Goldberger, J., Roweis, S., Hinton, G., Salakhutdinov, R.: Neighbourhood components analysis. Advances in Neural Information Processing Systems, 17 (2005)
10. Hertz, T., Bar-Hillel, A., Weinshall, D.: Boosting margin based distance functions for clustering. In: ICML (2004)
11. Hertz, T., Bar-Hillel, A., Weinshall, D.: Learning a kernel function for classification with small training samples. In: ICML (2006)
12. Li, F.F., Fergus, R., Perona, P.: One-shot learning of object categories. IEEE PAMI 28(4), 594–611 (2006)
13. Ng, A.Y.: Feature selection, L1 vs. L2 regularization, and rotational invariance. In: ACM International Conference Proceeding Series (2004)
14. Shental, N., Bar-Hillel, A., Hertz, T., Weinshall, D.: Computing gaussian mixture models with em using equivalence constraints. In: NIPS (2003)
15. Shental, N., Hertz, T., Weinshall, D., Pavel, M.: Adjustment learning and relevant component analysis. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002. LNCS, vol. 2353, Springer, Heidelberg (2002)
16. Sudderth, E.B., Torralba, A., Freeman, W.T., Willsky, A.S.: Learning hierarchical models of scenes, objects, and parts. In: Proc. ICCV (2005)
17. Tibshirani, R.: Regression Shrinkage and Selection via the Lasso. Journal of the Royal Statistical Society. Series B (Methodological) 58(1), 267–288 (1996)
18. Tsymbal, A., Puuronen, S., Skrypnyk, I.: Ensemble feature selection with dynamic integration of classifiers. In: Int. ICSC-CIMA (2001)
19. Zheng, A.X., Jordan, M.I., Liblit, B., Aiken, A.: Statistical debugging of sampled programs. Advances in Neural Information Processing Systems 17 (2003)