FAST SEMI-SUPERVISED DISCRIMINATIVE COMPONENT ANALYSIS

Jaakko Peltonen[1], *Jacob Goldberger*[2], *Samuel Kaski*[1]

[1]Helsinki University of Technology, Laboratory of Computer and Information Science,
Helsinki Institute for Information Technology and Adaptive Informatics Research Centre,
P.O. Box 5400, FI-02015 TKK, Finland
[2]Bar-Ilan University, School of Engineering, Ramat-Gan 52900, Israel

ABSTRACT

We introduce a method that learns a class-discriminative subspace or *discriminative components* of data. Such a subspace is useful for visualization, dimensionality reduction, feature extraction, and for learning a regularized distance metric. We learn the subspace by optimizing a probabilistic semiparametric model, a mixture of Gaussians, of classes in the subspace. The semiparametric modeling leads to fast computation ($O(N)$ for N samples) in each iteration of optimization, in contrast to recent nonparametric methods that take $O(N^2)$ time, but with equal accuracy. Moreover, we learn the subspace in a semi-supervised manner from three kinds of data: labeled and unlabeled samples, and unlabeled samples with pairwise constraints, with a unified objective.

1. INTRODUCTION

Optimization of the distance metric for data analysis has been intensively studied in recent years. We focus on classification tasks, where the metric is typically used to compare samples to each other or to prototypes; then the criterion of learning the metric is better discriminability of the classes. This task can be called *discriminative component analysis*.

For classification, methods have been introduced to learn global ([1, 2, 3, 4, 5] and many others) and local (e.g. [6]) metrics. We concentrate on global metrics, corresponding to linear transformations of the data. Nonlinear transformations correspond in some cases to local metrics but in general they lose the original topology. Linear transformations are also easier to interpret and more robust to overfitting.

Below we discuss three separate choices for learning global metrics: (1) whether to use a parametric or nonparametric model, (2) whether to use a generative model of classes and features or a discriminative (or conditional) model, and (3) whether to use semi-supervised information.

Linear Discriminant Analysis (LDA) and some extensions (e.g. [7]) can be seen as joint generative models of data and their classes. LDA is optimal if data are from the assumed family (restricted Gaussian mixture) and the projection has enough dimensions to contain all class variation. Otherwise joint generative modeling is suboptimal since only the conditional class distribution is of interest.

Discriminative methods have been proposed, based on approximations of Shannon entropy or some variant (see [5, 8]), conditional covariance operators [1], and others [2].

An especially intuitive probabilistic approach is to optimize a conditional model for the classes. This has been done in Neighborhood Components Analysis (NCA; [3]) and Informative Discriminant Analysis (IDA; [4]), which optimize a nonparametric class predictor and hence do not require distributional assumptions. The downside is the computational complexity; each iteration in the optimization is $O(N^2)$ where N is the number of data points.

When only few labeled samples are available, supervised methods may overfit, yielding poor performance on test data. *Semi-supervised learning* (e.g. [9]) can reduce overfitting by using additional information, typically unlabeled samples or samples with pairwise constraints. It has been widely applied outside dimensionality reduction, to e.g. mixture modeling [10, 11]. For dimensionality reduction, semi-supervised learning based on joint density estimation in the original space has been used in [12], and based on discriminative modeling of pairwise constraints but no labeled or unlabeled samples in [3]. Metrics have been learned based on pairwise constraints in [13].

In this paper we introduce a fast method for linear dimensionality reduction. It finds a class-discriminative subspace by optimizing a semiparametric mixture of Gaussians for density estimation within the subspace. Our method is discriminative for the labeled data, in contrast to generative methods like LDA.

Our method has two main contributions. Firstly, it *reduces computational complexity*: the semiparametric predictor improves speed to $O(N)$ compared to $O(N^2)$ for IDA and NCA, and adds robustness compared to nonparametric prediction. Secondly, our method can *learn from unlabeled data when it is available*; when labeled data is scarce, such semi-supervised learning can improve results. More specifically, in addition to the labeled data, our method

is able to learn from pairwise constrained and fully unlabeled data as well, if they are available.

We will use two abbreviations for our method: when we use labeled data but not unlabeled data, we denote the method DCA-GM (for "discriminative component analysis by Gaussian mixtures"); when we also use unlabeled data, we denote the method SDCA-GM (for "semi-supervised discriminative component analysis by Gaussian mixtures").

2. THE METHOD

2.1. Problem Setting

We are given a labeled data set consisting of N real-valued input vectors $\mathbf{x}_i$ in $\mathbb{R}^D$ and corresponding class labels c_i (C classes in total). We have additionally an unlabeled data set where class labels are not given; our assumption is that each unlabeled data point comes from one of the C classes.

The unlabeled data may be partly grouped: some of the unlabeled input vectors are placed into groups G_i where each group of input vectors is known to come from a single unknown class as in [13]. If pairwise must-link constraints are given instead of point groups, the groups can be identified by a transitive closure of the constraints as in [11].

The above kinds of data are available in many real-world problems: in face recognition, labeled data are pictures of known persons; unlabeled data are pictures of unknown persons; pairwise constrained data are pictures from short video sequences (so the same unknown person is in each picture).

The task is to find a low-dimensional linear transformation $\mathbf{A} : \mathbb{R}^D \to \mathbb{R}^d$ that preserves as much information required for classification as possible. The three kinds of data (labeled data, grouped unlabeled data, and ungrouped unlabeled data) all provide information for the task.

2.2. Our Method: Semiparametric Model

We want to somehow measure how well the low-dimensional linear transformation $\mathbf{A}$ discriminates classes, and optimize $\mathbf{A}$ to maximize this measure. The principle of our method is that we can measure how class-discriminative any transformation $\mathbf{A}$ is by simply building a class predictor for the transformed data $\mathbf{y} = \mathbf{A}\mathbf{x}$ and measuring its performance. Rather than using a slow nonparametric predictor as in [3, 4] we use a semiparametric predictor and optimize both the transformation and the predictor simultaneously. Note that finding the transformation is the main objective: optimizing the predictor is simply a means to that end.

We derive our semiparametric class predictor from a mixture of Gaussians representation for the transformed data $\mathbf{y} = \mathbf{A}\mathbf{x}$ and their classes c. Note that this representation is of the transformed data $\mathbf{y} = \mathbf{A}\mathbf{x}$ and their classes only; we do not make a generative model of the original features $\mathbf{x}$ as in, for instance, factor analysis.

We represent each class of the transformed data as a mixture of K Gaussian densities with a single covariance matrix for each class. The mixture generates the density

$$p(\mathbf{y}, c; \theta) = \sum_{k=1}^{K} \alpha_c \beta_{c,k} N(\mathbf{y}; \boldsymbol{\mu}_{c,k}, \boldsymbol{\Sigma}_c) \tag{1}$$

where the α_c are overall class weights, the $\beta_{c,k}$ are weights for the individual Gaussian components, and $N(\mathbf{y}; \boldsymbol{\mu}_{c,k}, \boldsymbol{\Sigma}_c)$ is the density of a Gaussian distribution with mean $\boldsymbol{\mu}_{c,k}$ and covariance matrix $\boldsymbol{\Sigma}_c$, computed at $\mathbf{y} = \mathbf{A}\mathbf{x}$. The α_c, $\beta_{c,k}$, $\boldsymbol{\mu}_{c,k}$, and $\boldsymbol{\Sigma}_c$ are parameters of the mixture, together denoted by θ. The α_c and $\beta_{c,k}$ are required to be nonnegative; the α_c sum to one, and the $\beta_{c,k}$ sum to one for each c.

If needed, the model can be easily generalized to have different numbers of Gaussians for the classes or different covariance matrices for each Gaussian. The equations would be very similar.

2.3. Objective Function

We seek a class-discriminative subspace: if we have much labeled data, the performance of the predictor (1) is naturally measured as log probability of correct classification. However, to increase the robustness of such a measure, we will do semi-supervised learning by also predicting properties of the unlabeled data: for the grouped unlabeled data we can naturally compute the log probability of the grouping information. For all unlabeled data (grouped or not) we also add a log probability of the transformed data. The interpretation of these terms is discussed further in Section 2.4.

The full objective function is

$$\begin{aligned} &\sum_{i \in labeled} \log p(c_i | \mathbf{A}\mathbf{x}_i; \theta) \\ &\quad + \lambda_1 \sum_{i \in groups} \log p(G_i; \mathbf{A}, \theta) \\ &\quad + \lambda_2 \sum_{i \in unlabeled} \log p(\mathbf{A}\mathbf{x}_i; \theta) \end{aligned} \tag{2}$$

where the λ_1 and λ_2 are weighting multipliers, $p(\mathbf{A}\mathbf{x}; \theta) = \sum_c p(\mathbf{A}\mathbf{x}, c; \theta)$ and $p(c|\mathbf{A}\mathbf{x}; \theta) = p(\mathbf{A}\mathbf{x}, c; \theta)/p(\mathbf{A}\mathbf{x}; \theta)$ are standard unlabeled and conditional probabilities computed from (1) with $\mathbf{A}\mathbf{x}$ in place of $\mathbf{y}$, and $p(G_i; \mathbf{A}, \theta)$ is the probability that the ith set of grouped transformed data comes from a single class, that is,

$$p(G_i; \mathbf{A}, \theta) = \frac{\sum_c \alpha_c \prod_{\mathbf{x} \in G_i} p(\mathbf{A}\mathbf{x}|c; \theta)}{\prod_{\mathbf{x} \in G_i} p(\mathbf{A}\mathbf{x}; \theta)} \tag{3}$$

where $p(\mathbf{A}\mathbf{x}|c; \theta) = p(\mathbf{A}\mathbf{x}, c; \theta)/\alpha_c$.

We define two versions our method. In DCA-GM only the first sum in (2) is used; that is, only labeled data are

used. In SDCA-GM all three sums are used; that is, any unlabeled data are also used. For both versions of our method, we maximize the objective function with respect to the linear transformation $\mathbf{A}$ (we also add a term that penalizes the matrix norm). For the mixture parameters θ we use a hybrid approach described in Section 3.

2.4. Interpretations for the objective function.

The first ('labeled') sum term in (2) is simply the log probability of correct classification of the labeled data. If all data is labeled, only this term remains and the objective function performs fully supervised conditional log-likelihood optimization in the same way as NCA and IDA but faster.

The second ('groups') sum term only affects the semi-supervised version SDCA-GM. It is a sum over groups, not individual points; it is the log probability that each group G_i is generated from a single unknown class. We assume the class is one of the C classes and samples are independent given the class. Note that information about grouping is weaker than known class labels: for example, two different groups might have the same underlying class. In the experiments we do not yet use grouping, but we provide the details to show our model can easily use such information.

The third ('unlabeled') sum term only affects the semi-supervised version SDCA-GM. The term is the unlabeled likelihood of all unlabeled transformed data (grouped or not). For a fixed subspace, a sum of conditional and unlabeled likelihoods would be a traditional way of performing semi-supervised learning of the mixture parameters. Here the crucial task is learning the subspace, so the term actually has a more subtle effect: under certain conditions the term prefers subspaces where the data is clustered. Suppose the covariance matrix of the transformed data $\mathbf{Ax}$ is fixed with respect to $\mathbf{A}$ (e.g., fix the norm of $\mathbf{A}$ and whiten the data beforehand). A Gaussian has the highest entropy of all distributions having the same covariance matrix; the third sum term in (2) is an empirical estimate of (negative) entropy, so maximizing it maximizes *nongaussianity of the transformed data*. Intuitively, well clustered data is more nongaussian than data that is mixed together.

A potential problem is that if the transformed data does not have a fixed covariance matrix, the 'unlabeled' term will also reward criteria unrelated to class discrimination: matrices with small norm and subspaces where the data has small scale along some directions. E.g. orthogonality constraints for $\mathbf{A}$ could potentially be used to improve the 'unlabeled' term. In experiments the simple version sufficed for good results, however, and we leave extensions to further work.

For SDCA-GM we use the scalar parameters λ_1 and λ_2 to control the relative scales of the three sum terms. In the experiments we did not optimize λ_2 but simply set it to a small value. We would do the same for λ_1 but we did not use grouped data in the experiments of this paper.

3. OPTIMIZATION

We first discuss how to learn the linear transformation, and then how to learn the mixture parameters.

Our main task is to optimize the linear transformation; we use standard conjugate gradient optimization for that. The gradient equations for each term in the objective function (2) are simple and are given in full in Appendix A.

We could in principle use conjugate gradients to learn both the linear transformation and the mixture parameters (reparameterization of α_c, $\beta_{c,k}$, and $\mathbf{\Sigma}_c$ would be necessary). Such very flexible discriminative learning might, however, be prone to overfitting, so we use the computationally more convenient expectation maximization (EM) to learn the centers $\boldsymbol{\mu}_{c,k}$, covariances Σ_c, and weights α_c and $\beta_{c,k}$ from the transformed data. See Appendix B for details.

We do a few steps of EM before each iteration of conjugate gradient that discriminatively optimizes $\mathbf{A}$. Since EM maximizes joint likelihood[1], the learning of $\mathbf{A}$ searches for a transformation *where a generative mixture model gives good discriminative performance*, which is a well-defined task.

This hybrid optimization is not a requirement of our model but a convenient simplification; we then only need to optimize the transformation $\mathbf{A}$ by conjugate gradients.

For both DCA-GM and SDCA-GM the computational complexities of gradient computation and EM estimation are $O(NCKdD + NCd^2 + Cd^3 + CKd^2)$ and $O(NDd + NCKd^2 + Cd^3 + CKd^2)$ respectively; both are linear with respect to the number of samples N. The total running time depends on the numbers of iterations and gradient computations and EM steps per iteration. In the experiments we used fixed small numbers of iterations and EM steps; potential performance improvement with more iterations and EM steps will be investigated in future work.

4. EXPERIMENTS

We evaluated our method on six data sets from the UC Irvine repository: Wine, Balance, Housing, Ionosphere, Iris, and Isolet.[2] Each data set was split 30 times into training (20%) and testing (80%) subsets. For each data set we sought linear projections to a two-dimensional subspace (except for Isolet where we sought five-dimensional projections since 2D projection does not yield satisfactory results).

We compare our method to two linear supervised dimensionality reduction methods: a combined method where Linear Discriminant Analysis (LDA) is followed by Relevant Component Analysis (RCA; [13]), which is known

[1] We could have used recent discriminative EM versions but they are much slower than normal joint EM.

[2] For the largest data set (Isolet), a subset of 3740 samples was used, and dimensionality was reduced to 30 by a Principal Component Analysis (PCA) projection, to reduce the computational load.

to be better than the standard LDA, and Neighborhood Components Analysis (NCA). As a baseline we use the unsupervised Principal Component Analysis (PCA).

We ran both versions of our method. The supervised DCA-GM was trained only on the labeled training subset. The semi-supervised SDCA-GM was also given the unlabeled test data during training; that is, a transductive setting was used. (We stress that SDCA-GM is applicable outside transductive tasks as well: it can learn in a semi-supervised manner from any unlabeled data, not just test data.) For SDCA-GM the weight of the 'unlabeled' term in (2) was set to $\lambda_2 = 0.01$. For both DCA-GM and SDCA-GM we used a mixture of two Gaussians to model each class; we used K-means and LDA+RCA to initialize the mixture and the linear transformation, respectively.

To do a fair comparison between all the linear dimensionality reduction methods, the performance of all the methods was evaluated by test accuracy of K nearest neighbor (KNN) classification (we used K=1) in the found subspace.

Example results of the semi-supervised SDCA-GM applied to transductive learning on the Wine data are shown in Figure 1. Notice how the bottomost Gaussian component of class 2 is not centered on the few labeled data, but is closer to the larger amount of unlabeled data; intuitively, relying on unlabeled data can be robust assuming that nearby unlabeled samples in reality come from the correct class. This is not always true; the upper Gaussian of class 3 has three nearby unlabeled samples that in reality come from class 1.

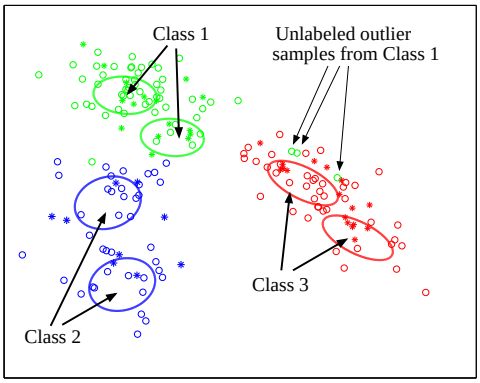

Fig. 1. SDCA-GM result on Wine data (178 points from 3 classes). A two-dimensional subspace was sought for originally 13-dimensional data. The labels of 20% of the points (denoted by '*') were known in the training. The labels of the rest of the points (denoted by 'o') were not revealed during training. The points are shown in the found two-dimensional space. Ellipses show the location and shape of the Gaussian components used to model each class.

The classification results on the test subsets are presented in Figure 2. SDCA-GM and DCA-GM are comparable with NCA which is considered to be the state-of-the-art; in some cases SDCA-GM is even slightly better. For these data sets both NCA, DCA-GM and SDCA-GM run fast and there is no significant difference in their running times; however, as stated in the previous sections, DCA-GM and SDCA-GM have a much smaller computational complexity than NCA. We will run tests on larger data and investigate optimization of the speed of our methods in future work.

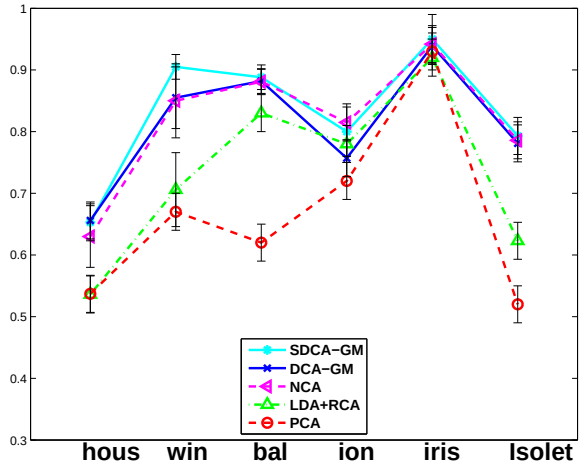

Fig. 2. Classification accuracy on UCI data sets Housing, Wine, Balance, Ionosphere, Iris, and Isolet. Results are averages of test data results over 30 realizations of splitting each data set into training (20%) and testing (80%) subsets. A linear dimensionality reduction down to $d = 2$ was sought in all cases except the Isolet data where $d = 5$.

5. DISCUSSION

In the experiments we measured performance with a KNN classifier for all methods. However, the natural classifier derived from the semiparametric method SDCA-GM is the (maximum) posterior probability of the class given the feature vector. Using this classifier yields a slight improvement over the results in Figure 2; another advantage is that training data are not needed in the classification step which saves memory and reduces computational complexity.

Initial investigation with smaller training set sizes (less than 20% of data used for training) indicate that performance of DCA-GM worsens when there is too little data, and that performance of SDCA-GM is affected by the weight λ_2 for the unlabeled term in the objective function (results not shown); too large λ_2 will incur poor performance. Potentially we could use e.g. cross-validation to select λ_2; such methods will be investigated in future work.

In the mixture modeling method of [11], class discovery is considered through cannot-link constraints. Such ideas could be used in our method, possibly allowing unlabeled samples to come from new classes.

SDCA-GM finds a class-discriminative subspace. In several applications such as visualization of class separability the subspace is the main result; restricting to a subspace is equivalent to regularizing the distance metric so that changes perpendicular to the subspace do not affect distances.

The linear transformation $\mathbf{A}$ is identifiable only with respect to the subspace it finds; within the subspace, it can be shown that changes in $\mathbf{A}$ can be exchanged with changes in θ without affecting the first two terms of (2).

If desired, a distance metric in the subspace can be sought by e.g. these possibilities: (1) If the topology is unimportant, compare points by their estimated class distributions. (2) For a global metric, run NCA inside the projection space, or (3) use the average covariance matrix of the classes. Here we simply used the Euclidean metric after the linear transformation; this sufficed to get good results.

We noticed only recently a related method that also optimizes a semiparametric conditional class predictor [14]. Compared to it, we use a more flexible parameterization and a robust combination of generative and discriminative optimization. Also, [14] does not use semi-supervised learning.

The objective function (2) looks similar to the multi-conditional loglikelihood in [15]; however, the model in [15] does (local) factor analysis whereas we model data only within the subspace. Moreover, the method in [15] would require nontrivial extension to do global dimensionality reduction.

6. CONCLUSIONS

We have presented a fast semi-supervised method for finding subspaces where classes of data can be well discriminated. The method optimizes a well-defined criterion: performance of a semiparametric mixture of Gaussians predictor for classes and pairwise-constrained data groups plus a regularization term for unlabeled data. The method has linear complexity with respect to the number of samples, and it performed as well as the state-of-the-art method Neighborhood Component Analysis (NCA) on benchmark data sets. Using unlabeled data was shown to improve results over NCA. Finally, we described how to use pairwise constraints in the method, and how to learn metrics for the found subspaces. These directions will be explored in future work.

7. APPENDIX A: GRADIENT EQUATIONS

The objective function (2) has three terms: it can be shown the gradient of the first (labeled) term with respect to $\mathbf{A}$ is

$$\sum_{i,c,k} (p(c,k|\mathbf{A}\mathbf{x}_i;\theta) - \delta_{c,c_i} p(k|\mathbf{A}\mathbf{x}_i,c;\theta)) \cdot \boldsymbol{\Sigma}_c^{-1}(\mathbf{A}\mathbf{x}_i - \boldsymbol{\mu}_{c,k})\mathbf{x}_i^T \quad (4)$$

where i goes over labeled points, $\delta_{c_i,c}$ is one if $c_i = c$ and zero otherwise, and

$$p(c,k|\mathbf{A}\mathbf{x};\theta) = \frac{\alpha_c \beta_{c,k} N(\mathbf{A}\mathbf{x};\boldsymbol{\mu}_{c,k},\boldsymbol{\Sigma}_c)}{\sum_{c',l} \alpha_{c'} \beta_{c',l} N(\mathbf{A}\mathbf{x};\boldsymbol{\mu}_{c',l},\boldsymbol{\Sigma}_{c'})} , \quad (5)$$

$$p(k|\mathbf{A}\mathbf{x},c;\theta) = \frac{\beta_{c,k} N(\mathbf{A}\mathbf{x};\boldsymbol{\mu}_{c,k},\boldsymbol{\Sigma}_c)}{\sum_{l} \beta_{c,l} N(\mathbf{A}\mathbf{x};\boldsymbol{\mu}_{c,l},\boldsymbol{\Sigma}_c)} . \quad (6)$$

The gradient of the second (groups) term is

$$\lambda_1 \sum_{i,x\in G_i,c,k} (p(c,k|\mathbf{A}\mathbf{x};\theta) - p(c|G_i;\mathbf{A},\theta) p(k|\mathbf{A}\mathbf{x},c;\theta)) \cdot \boldsymbol{\Sigma}_c^{-1}(\mathbf{A}\mathbf{x} - \boldsymbol{\mu}_{c,k})\mathbf{x}^T \quad (7)$$

where i goes over groups and $p(c|G_i;\mathbf{A},\theta)$ is given by (11). The gradient of the third (unlabeled) term is

$$-\lambda_2 \sum_{i,c,k} p(\mathbf{A}\mathbf{x}_i,c;\theta) \boldsymbol{\Sigma}_c^{-1}(\mathbf{A}\mathbf{x}_i - \boldsymbol{\mu}_{c,k})\mathbf{x}_i^T \quad (8)$$

where i goes over all unlabeled points.

8. APPENDIX B: DETAILS OF THE EM ALGORITHM

For DCA-GM the standard EM algorithm suffices. For SDCA-GM, we must also learn from unlabeled data and pairwise constrained data. To solve this subproblem, we extend the EM algorithm in [10] which learns a mixture model from pairwise constrained data. We extend that method in two ways. Firstly, our pairwise constraints (groupings) state that two points come from the same *class*, not necessarily the same *Gaussian* within the class. A similar extension has been used in [11]. Secondly, we also allow samples with known labels. We assume a fixed number of classes and components, and hence get simpler (faster) update rules than the considerably more complex ones of [11].

In the E step, for labeled points $\mathbf{x}_i$ we compute weights

$$p_i(c,k) = \delta_{c,c_i} p(k|\mathbf{A}\mathbf{x}_i,c;\theta) \quad (9)$$

where c_i is the known class label, and for grouped unlabeled points $\mathbf{x}_i$ we compute the weights

$$p_i(c,k) = p(c|G_i;\mathbf{A},\theta) p(k|\mathbf{A}\mathbf{x}_i,c;\theta) . \quad (10)$$

Here G_i denotes the group the ith point belongs to, and $p(c|G_i;\mathbf{A},\theta)$ is the probability that G_i comes from class c given that it comes from a single class:

$$p(c|G_i;\mathbf{A},\theta) = \frac{\alpha_c \prod_{\mathbf{x}\in G_i} p(\mathbf{A}\mathbf{x}|c;\theta)}{\sum_{c'} \alpha_{c'} \prod_{\mathbf{x}\in G_i} p(\mathbf{A}\mathbf{x}|c';\theta)} . \quad (11)$$

An unlabeled point $\mathbf{x}_i$ that is not grouped equals a group where $\mathbf{x}_i$ is the only member; then $p(c|G_i;\mathbf{A},\theta)$ reduces to $p(c|\mathbf{A}\mathbf{x}_i,\theta)$.

Given $p_i(c,k)$, the maximization (M step) proceeds as usual for EM-based optimization of a Gaussian mixture, using $p_i(c,k)$ as the weight of a particular point in the sums for the Gaussian k of class c. For example, we set the centers to $\boldsymbol{\mu}_{c,k} = \sum_i p_i(c,k)\mathbf{A}\mathbf{x}_i / \sum_i p_i(c,k)$, where i goes over all labeled and unlabeled (grouped or not) data points.

9. ACKNOWLEDGEMENTS

S. Kaski and J. Peltonen were supported by the Academy of Finland, decision number 108515. This work was also supported in part by the IST Programme of the European Community, under the PASCAL Network of Excellence, IST-2002-506778. This publication only reflects the authors' views. All rights are reserved because of other commitments.

10. REFERENCES

[1] Kenji Fukumizu, Francis R. Bach, and Michael I. Jordan, "Kernel dimensionality reduction for supervised learning," in *Advances in Neural Information Processing Systems 16*, Sebastian Thrun, Lawrence Saul, and Bernhard Schölkopf, Eds. MIT Press, Cambridge, MA, 2004.

[2] Amir Globerson and Sam Roweis, "Metric learning by collapsing classes," in *Advances in Neural Information Processing 18*, Y. Weiss, B. Schölkopf, and J. Platt, Eds., pp. 451–458. MIT Press, Cambridge, MA, 2006.

[3] Jacob Goldberger, Sam Roweis, Geoffrey Hinton, and Ruslan Salakhutdinov, "Neighbourhood components analysis," in *Advances in Neural Information Processing Systems 17*, Lawrence K. Saul, Yair Weiss, and Léon Bottou, Eds., pp. 513–520. MIT Press, Cambridge, MA, 2005.

[4] Samuel Kaski and Jaakko Peltonen, "Informative discriminant analysis," in *Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003)*, pp. 329–336. AAAI Press, Menlo Park, CA, 2003.

[5] Kari Torkkola, "Feature extraction by non-parametric mutual information maximization," *Journal of Machine Learning Research*, vol. 3, pp. 1415–1438, 2003.

[6] Carlotta Domeniconi, Jing Peng, and Dimitrios Gunopulos, "Locally adaptive metric nearest-neighbor classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 9, pp. 1281–1285, 2002.

[7] Nagendra Kumar and Andreas G. Andreou, "Heteroscedastic discriminant analysis and reduced rank HMMs for improved speech recognition," *Speech Communication*, vol. 26, pp. 283–297, 1998.

[8] José M. Leiva-Murillo and Antonio Artés-Rodríguez, "A Gaussian mixture based maximization of mutual information for supervised feature extraction," in *Proceedings of the Fifth International Conference on Independent Component Analysis and Blind Signal Separation (ICA 2004)*, C. G. Puntonet and A. Prieto, Eds., pp. 271–278. Springer-Verlag, Berlin Heidelberg, 2004.

[9] A. Blum and T. Mitchell, "Combining labeled and unlabeled data with co-training," in *Proceedings of the 11th Annual Conference on Computational Learning Theory*, pp. 92–100. 1998.

[10] Noam Shental, Aharon Bar-Hillel, Tomer Herz, and Daphna Weinshall, "Computing Gaussian mixture models with EM using equivalence constraints," in *Advances in Neural Information Processing Systems 16*, Sebastian Thrun, Lawrence Saul, and Bernhard Schölkopf, Eds. MIT Press, Cambridge, MA, 2003.

[11] Qi Zhao and David J. Miller, "Mixture modeling with pairwise, instance-level class constraints," *Neural Computation*, vol. 17, pp. 2482–2507, 2005.

[12] Shipeng Yu, Kai Yu, Volker Tresp, Hans-Peter Kriegel, and Mingrui Wu, "Supervised probabilistic principal component analysis," in *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD 2006)*, Lyle Ungar, Mark Craven, Dimitrios Gunopulos, and Tina Eliassi-Rad, Eds., pp. 464–473. ACM Press, New York, NY, 2006.

[13] Noam Shental, Tomer Hertz, Daphna Weinshall, and Misha Pavel, "Adjustment learning and relevant component analysis," in *ECCV '02: Proceedings of the 7th European Conference on Computer Vision-Part IV*, London, UK, 2002, pp. 776–792, Springer-Verlag.

[14] Sajama and Alon Orlitsky, "Supervised dimensionality reduction using mixture models," in *Proceedings of the 22nd International Machine Learning Conference (ICML 2005)*, Luc De Raedt and Stefan Wrobel, Eds. ACM Press, 2005.

[15] B. Michael Kelm, Chris Pal, and Andrew McCallum, "Combining generative and discriminative methods for pixel classification with multi-conditional learning," in *Proceedings of the 18th International Conference on Pattern Recognition (ICPR 2006)*, vol. 2, pp. 828–832. IEEE, 2006.