Fast Discriminative Component Analysis for Comparing Examples

Jaakko Peltonen[1,3]**, Jacob Goldberger**[2] **and Samuel Kaski**[1]
[1]Helsinki Institute for Information Technology, Laboratory of Computer and Information Science, Helsinki University of Technology, P.O. Box 5400, FI-02015 TKK, Finland
[2]School of Engineering, Bar-Ilan University, Ramat-Gan 52900, Israel
[3]Department of Computer Science, P.O. Box 68, FI-00014 University of Helsinki, Finland
goldbej@eng.biu.ac.il {jaakko.peltonen, samuel.kaski}@tkk.fi

Abstract

Two recent methods, Neighborhood Components Analysis (NCA) and Informative Discriminant Analysis (IDA), search for a class-discriminative subspace or *discriminative components* of data, equivalent to learning of distance metrics invariant to changes perpendicular to the subspace. Constraining metrics to a subspace is useful for regularizing the metrics, and for dimensionality reduction. We introduce a variant of NCA and IDA that reduces their computational complexity from quadratic to linear in the number of data samples, by replacing their purely non-parametric class density estimates with semiparametric mixtures of Gaussians. In terms of accuracy, the method is shown to perform as well as NCA on benchmark data sets, outperforming several popular linear dimensionality reduction methods.

1 Introduction

Optimizing the distance metric has been intensively studied in recent years. We focus on classification tasks, where algorithms typically use the metric to compare samples to each other or to prototypes; then the criterion of learning the metric is better classification, or more generally better discriminability of the classes. This task can be called *discriminative component analysis*. Another possible application domain is "supervised unsupervised learning," where the metric is learned in a supervised setting and used for unsupervised learning [1].

In classification settings methods have been introduced for learning both global ([2, 3, 4, 5, 6, 7] and many others) and local [8, 9, 10] metrics. Two essentially equivalent methods, Neighborhood Components Analysis (NCA; [4]) and Informative Discriminant Analysis (IDA; [6, 11]), search for subspaces which both regularizes the problem and helps visualize the results. The methods are non-parametric 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.

We introduce a faster method which still outperforms several linear dimensionality reduction methods on benchmark data sets. The method is $O(N)$, since it uses semiparametric mixtures of Gaussians for density estimation. We denote the method DCA-GM (short for "discriminative component analysis by Gaussian mixtures"). The method finds a class-discriminative subspace. In several applications such as visualization of class separability the subspace is the main result. If a metric is desired then it can be sought with several methods in the subspace; we introduce a fast method having linear computational complexity.

2 The method

We begin with a labelled 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). The task is to find a low-dimensional linear transformation $\mathbf{A} : \mathbb{R}^D \to \mathbb{R}^d$ such that the transformation preserves as much information required for classification as possible. The performance will be measured by a class predictor working on the transformed data. Combining the steps, both the transformation and the predictor can be optimized simultaneously.

We use a parametric class predictor derived from a mixture of Gaussians representation for the transformed data $\mathbf{y}$ and their classes. We represent each class as a mixture of K Gaussian densities with a single covariance matrix for each class.[1] The mixture generates the following 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 α_c are overall class weights, $\beta_{c,k}$ are weights for individual Gaussian components,[2] 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 $\mathbf{y}$. The α_c, $\beta_{c,k}$, $\boldsymbol{\mu}_{c,k}$, and $\boldsymbol{\Sigma}_c$ are parameters of the mixture, together denoted θ.

As our objective function we maximize the log probability of correct classification:

$$L(\mathbf{A}, \theta) = \sum_i \log p(c_i | \mathbf{A}\mathbf{x}_i; \theta) = \sum_i \log \frac{p(\mathbf{A}\mathbf{x}_i, c_i; \theta)}{\sum_c p(\mathbf{A}\mathbf{x}_i, c; \theta)} \tag{2}$$

We maximize this objective function with respect to the linear transformation $\mathbf{A}$ (we also add a term that penalizes the matrix norm). Note that the objective function only involves the conditional probabilities $p(c|\mathbf{A}\mathbf{x}_i; \theta)$; thus, although our model functionally generates a joint density, the linear transformation is trained discriminatively. For the mixture parameters θ we use a hybrid approach described in the next section.

2.1 Learning the model parameters

We first discuss how to learn the linear transformation, and then how to learn the mixture parameters. For the linear transformation we use standard conjugate gradient optimization. It can be shown that the gradient of the objective function (2) is

$$\frac{\partial L(\mathbf{A}, \theta)}{\partial \mathbf{A}} = \sum_{i,c,k} \Big(p(c, k | \mathbf{A}\mathbf{x}_i; \theta) - \delta_{c_i,c} p(k | \mathbf{A}\mathbf{x}_i, c_i; \theta) \Big) \boldsymbol{\Sigma}_c^{-1} (\mathbf{A}\mathbf{x}_i - \boldsymbol{\mu}_{c,k}) \mathbf{x}_i^T \tag{3}$$

where $\delta_{c_i,c}$ is one if $c_i = c$ and zero otherwise, and

$$p(c, k | \mathbf{A}\mathbf{x}; \theta) = \frac{\alpha_c \beta_{c,k} N(\mathbf{A}\mathbf{x}; \boldsymbol{\mu}_{c,k}, \boldsymbol{\Sigma}_c)}{\sum_{c',l} \alpha_{c'} \beta_{c',l} N(\mathbf{A}\mathbf{x}; \boldsymbol{\mu}_{c',l}, \boldsymbol{\Sigma}_{c'})} \ , \quad p(k | \mathbf{A}\mathbf{x}, c; \theta) = \frac{\beta_{c,k} N(\mathbf{A}\mathbf{x}; \boldsymbol{\mu}_{c,k}, \boldsymbol{\Sigma}_c)}{\sum_l \beta_{c,l} N(\mathbf{A}\mathbf{x}; \boldsymbol{\mu}_{c,l}, \boldsymbol{\Sigma}_c)} \ . \tag{4}$$

A hybrid optimization approach. We could in principle use conjugate gradient to learn both the linear transformation and the mixture parameters.[3] Instead, for convenience we use expectation maximization (EM) to learn the centers $\boldsymbol{\mu}_{c,k}$, the covariances Σ_c, and the weights α_c and $\beta_{c,k}$ from the transformed data. We do a few steps of this EM estimation before each iteration of conjugate gradient. The 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 gradient.

[1] Allowing different numbers of Gaussians for the classes or different covariance matrices for each Gaussian would yield very similar equations.

[2] The α_c and $\beta_{c,k}$ are nonnegative; the α_c sum to one, and the $\beta_{c,k}$ sum to one for each c.

[3] Reparameterizations would be necessary to make the α_c and $\beta_{c,k}$ stay multinomial distributions during optimization, and to make the $\boldsymbol{\Sigma}_c$ stay valid covariance matrices.

Improving optimization by reparameterization. In the hybrid approach described above, the θ do not change during the conjugate gradient iteration for $\mathbf{A}$. This can slow down convergence. We briefly mention that it is possible avoid the slowdown by making part of the mixture directly dependent on $\mathbf{A}$: reparameterize the centers $\boldsymbol{\mu}_{c,k} = \mathbf{A}\boldsymbol{\mu}'_{c,k}$ where $\boldsymbol{\mu}'_{c,k}$ are locations in $\mathbb{R}^D$. This changes the gradient (3) only slightly: the rightmost term changes from $\mathbf{x}_i^T$ to $(\mathbf{x}_i - \boldsymbol{\mu}'_{c,k})^T$. In the EM step, estimate $\boldsymbol{\mu}'_{c,k}$ given $p(k|\mathbf{A}\mathbf{x}_i, c_i; \theta)$ by $\boldsymbol{\mu}'_{c,k} = \sum_{i:c_i=c} p(k|\mathbf{A}\mathbf{x}_i, c_i; \theta)\mathbf{x}_i / \sum_{i:c_i=c} p(k|\mathbf{A}\mathbf{x}_i, c_i; \theta)$ which is equivalent to EM where the hidden variable distribution is computed from transformed data by $p(k|\mathbf{A}\mathbf{x}_i, c_i; \theta)$.[4] In the experiments we did not use the reparameterization.

2.2 Properties of the method

Computational complexity. The computational complexities of the 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 number of iterations and numbers of gradient computations and EM steps per iteration. In the experiments we ran the algorithm for fixed small numbers of iterations and EM steps.

Partial unidentifiability of the metric. The linear transformation is identifiable only with respect to the subspace it finds: within the subspace, changes in the linear transformation can be exchanged with changes in the mixture parameters.[5]

In some cases identifying the subspace is enough, but often we also wish to find a metric for it. Our method provides a well-defined estimate of conditional class probabilities; this estimate is unaffected by unidentifiability and can be used to derive a metric in the projection space. We briefly mention three possibilities: 1) If the topology in the projection space is unimportant, simply compare points by their estimated class distributions. 2) For a local, topology preserving metric, compute local Fisher matrices as in [1]. 3) For a global, topology preserving metric, run NCA inside the projection space, or average local Fisher matrices over data points (the latter is an $O(N)$ computation). In the experiments we did not use these possibilities but used the Euclidean metric after the linear transformation; this sufficed to get good results.

Relation to previous work. Linear Discriminant Analysis (LDA) and some of its extensions [13, 14] can be interpreted as generative models that use a restricted mixture of Gaussians to model data and their classes jointly, whereas we optimize the conditional likelihood of the classes. Discriminative methods have been proposed based on a computationally easier alternative to Shannon entropy (see [15]), approximations to Shannon entropy [16], likelihood ratios of class-specific and class-independent models [17], and conditional covariance operators on reproducing kernel Hilbert spaces [2]; advantages of our method are that it does not involve approximations, does not reduce to LDA even for simple models, and has an intuitive yet rigorous objective function. The most closely related earlier work are IDA [6, 11] and NCA [4] which optimize a nonparametric conditional class predictor; our use of a semiparametric predictor improves speed and robustness compared to IDA and NCA.

3 Experiments

We evaluated the performance of our method (DCA-GM) on four standard data sets from the UC Irvine repository (Wine, Balance, Ionosphere, and Iris). Each data set was split 30 times into training (70%) and testing (30%) subsets. We implemented three linear supervised dimensionality-reduction methods for comparison: LDA, LDA+Relevant Component Analysis (RCA; [18]) and NCA. In our method we used a mixture of three Gaussians to model each class; we used K-means and LDA+RCA to initialize the mixture and linear transformation, respectively. The performance of the methods was evaluated by test accuracy of K nearest neighbor (KNN) classification (we use K=1).

The classification results on the test subsets are presented in Figure 1. DCA-GM is comparable with NCA which is considered to be the state-of-the-art. For these small data sets both NCA and

[4] A similar back-projection approach was used in [12], but only at the start of optimization.

[5] For any invertible $d \times d$ matrix $\mathbf{B}$ we have $p(c|\mathbf{B}\mathbf{A}\mathbf{x}; \theta) = p(c|\mathbf{A}\mathbf{x}; \theta')$ where θ' uses covariance matrices $\mathbf{B}^{-1}\boldsymbol{\Sigma}_c\mathbf{B}^{-1,T}$ and the $\boldsymbol{\mu}'_{c,k}$, $\beta_{c,k}$ and α_c are the same as in θ.

DCA-GM run fast and there is no significant difference in their running times; however, as stated in the previous sections, DCA-GM has much smaller computational complexity than NCA. We plan to run larger data sets later to show the difference.

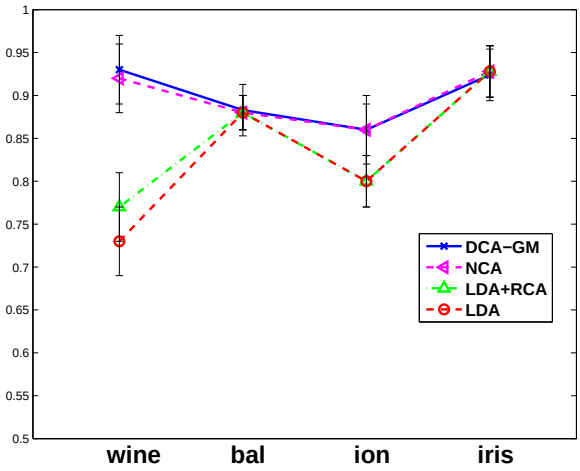

Figure 1: KNN classification accuracy on UCI data sets Wine, Balance (bal), Ionosphere (ion), and Iris. Results are averages of test data results over 30 realizations of splitting each data set into training (70%) and testing (30%) subsets. A linear dimensionality reduction down to $d = 2$ was applied in all cases.

4 Conclusions

We have presented a fast 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 the classes. The method has linear complexity with respect to the number of samples and performed as well as the state of the art NCA method on benchmark data sets. Here the method was used in a simple fashion to compare examples, but we also mentioned more advanced approaches for future work.

Acknowledgments

S. Kaski and J. Peltonen belong to the Adaptive Informatics Research Centre, a national centre of excellence of the Academy of Finland. They were supported by grant 108515, and by University of Helsinki's Research Funds. 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.

References

[1] Jaakko Peltonen, Arto Klami, and Samuel Kaski. Improved learning of Riemannian metrics for exploratory analysis. *Neural Networks*, 17:1087–1100, 2004.

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

[3] Amir Globerson and Sam Roweis. Metric learning by collapsing classes. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing 18*, pages 451–458. MIT Press, Cambridge, MDA, 2006.

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

[5] B. Hammer and T. Villmann. Generalized relevance learning vector quantization. *Neural Networks*, 15:1059–1068, 2002.

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

[7] Kari Torkkola and William Campbell. Mutual information in learning feature transformations. In *Proceedings of ICML-2000, the 17th International Conference on Machine Learning*, pages 1015–1022. Morgan Kaufmann, Stanford, CA, 2000.

[8] Carlotta Domeniconi, Jing Peng, and Dimitrios Gunopulos. Locally adaptive metric nearest-neighbor classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24:1281–1285, 2002.

[9] Trevor Hastie and Robert Tibshirani. Discriminant adaptive nearest neigbor classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18:607–616, 1996.

[10] Jing Peng, Douglas R. Heisterkamp, and H. K. Dai. LDA/SVM driven nearest neighbor classification. *IEEE Transactions on Neural Networks*, 14:940–942, 2003.

[11] Jaakko Peltonen and Samuel Kaski. Discriminative components of data. *IEEE Transactions on Neural Networks*, 16:68–83, 2005.

[12] K. Torkkola. Learning discriminative feature transforms to low dimensions in low dimentions. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, pages 969–976. MIT Press, Cambridge, MA, 2002.

[13] Trevor Hastie and Robert Tibshirani. Discriminant analysis by Gaussian mixtures. *Journal of the Royal Statistical Society B*, 58:155–176, 1996.

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

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

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

[17] Mu Zhu and Trevor Hastie. Feature extraction for non-parametric discriminant analysis. *Journal of Computational and Graphical Statistics*, 12:101–120, 2003.

[18] 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*, pages 776–792, London, UK, 2002. Springer-Verlag.