Discriminative Training of Gaussian Mixtures for Image Object Recognition

J. Dahmen, R. Schlüter, H. Ney

Lehrstuhl für Informatik VI
RWTH Aachen - University of Technology
Ahornstraße 55, D-52056 Aachen,
{dahmen, schlueter, ney}@informatik.rwth-aachen.de

Abstract. In this paper we present a discriminative training procedure for Gaussian mixture densities. Conventional *maximum likelihood* (ML) training of such mixtures proved to be very efficient for object recognition, even though each class is treated separately in training. Discriminative criteria offer the advantage that they also use out-of-class data, that is they aim at optimizing class separability. We present results on the US Postal Service (USPS) handwritten digits database and compare the discriminative results to those obtained by ML training. We also compare our best results with those reported by other groups, proving them to be state-of-the-art.

1 Introduction

In the last few years, the use of Gaussian mixture densities for image object recognition proved to be very efficient [1]. On well known object recognition tasks such as the USPS handwritten digits database, we obtained results that are very well comparable or even superior to results reported using support vector machines, artificial neural nets or decision trees. A drawback of the conventional ML training of mixture densities is the fact, that each class is handled separately in training. In opposite to this, the dicriminative *maximum mutual information* criterion (MMI) optimizes the *a posteriori* probabilities of the training samples and hence the class separability. In the following, we will deal with the MMI criterion for Gaussian mixture densities. We will present results obtained on the USPS database and compare these with results obtained by using ML training. Although we could not yet improve our best ML result (3.6% error on USPS), we show that using discriminative criteria the number of model parameters needed to achieve good results can be drastically reduced. Thus, discriminative criteria are very efficient for realizing fast classifiers that can be used in real-time environments. In the next chapters we will shortly describe the USPS database used in our experiments as well as the feature reduction approach we make use of. In Chapter 4 we will present Gaussian mixtures densities in the context of the Bayesian decision rule as well as the ML training approach. The discriminative training procedure will be dealt with in Chapter 5. Before drawing conclusions in Chapter 7, we will present results in Chapter 6.

2 The US Postal Service Database

The USPS database (`ftp://ftp.mpik-tueb.mpg.de/pub/bs/data/`) is a well known handwritten digit recognition database. It contains 7291 training objects and 2007 test objects. The characters are isolated and represented by a 16×16 pixels sized grayscale image (see Figure 1). In our experiments, in order to lose no information, we use each pixel as a feature, yielding a 256-dimensional feature vector. The USPS recognition task is known to be very hard, with a human error rate of about 2.5% on the testing data [2]. For our experiments we created

Fig. 1. Example images taken from the USPS database

additional virtual training data by shifting each image by one pixel into eight directions. Doing so, we on the one hand get a more precise estimation for the model parameters and on the other hand we obviously incorporate an invariance to slight translations. This procedure leads to 65.619 training samples which are then used to train our system. Note that the translated images are of size 18×18 pixel, as we want to guarantee that no pixel belonging to a digit gets shifted out of the image.

3 Feature Reduction

To reduce the number of model parameters to be estimated in the following, transforming the data into some low dimensional feature space is advisable. To do this, we propose the following modified subspace method: In a first step, we use the training data to estimate a whitening transformation matrix W [3, pp.26-29]. The data is then transformed using W and is afterwards called *white*, that is the class conditional covariance matrix

$$\hat{\Sigma} = \frac{1}{N} \sum_{n=1}^{N} (x_n - \mu_{k_n})(x_n - \mu_{k_n})^t \tag{1}$$

is the matrix of identity, where N is the number of training samples, x_n is the observation of training sample n and μ_{k_n} is the mean vector of class k_n, to which x_n belongs. In a second step, we generate K prototype vectors of the form $\mu_k - \mu$, where K is the number of classes, μ_k is the mean vector of class k and μ is the overall mean vector. We now transform those vectors into an orthonormal basis. To avoid the numerical instabilities of the classical Gram-Schmidt approach (caused by rounding errors), this is done by using a singular

value decomposition [4, pp. 59-67]. This yields a maximum of $(K-1)$ base vectors, as the dimensionality of the subspace spanned by the prototypes can be shown to be less or equal $(K-1)$ [3, pp.451]. By projecting the original feature vectors into that subspace we obtain the reduced feature vectors, which will be used in the following.

The proposed method proved to be more robust than a conventional *linear discriminant analysis* (LDA) [5, pp.114-123], but still gives the same results as compared to using $(K-1)$ LDA features. In case fewer than $(K-1)$ features are wanted, a LDA can be used in a second step, on the previously reduced features. If more features are needed, it is advisable to create so-called pseudoclasses by clustering the training data. In the case of the USPS database we obtained our best results by creating four pseudoclasses per class (the resulting feature vectors being 39-dimensional). Note that the pseudoclasses are created by clustering the data, which is done using the algorithms described below.

4 Gaussian Mixture Densities

To classify an observation $x \in \mathbb{R}^d$ we use the Bayesian decision rule [5, pp.10-39]

$$x \longmapsto r(x) = \underset{k}{\operatorname{argmax}} \{p(k)p_\lambda(x|k)\} \tag{2}$$

where $p(k)$ is the *a priori* probability of class k, $p_\lambda(x|k)$ is the *class conditional probability* for the observation x given class k and $r(x)$ is the classifier's decision. The parameter λ represents the set of all parameters of the class conditional probabilities $p_\lambda(x_n|k)$. As neither $p(k)$ nor $p_\lambda(x|k)$ are known, we have to choose models for them and estimate their parameters by using the training data. In our experiments we set $p(k) = \frac{1}{K}$ for each class k and modell $p_\lambda(x|k)$ by using Gaussian mixture densities. A Gaussian mixture is defined as a linear combination of Gaussian component densities $\mathcal{N}(x|\mu_{ki}, \Sigma_{ki})$ with $\lambda = \{c_{ki}, \mu_{ki}, \Sigma_{ki}\}$:

$$p_\lambda(x|k) = \sum_{i=1}^{I_k} c_{ki} \cdot \mathcal{N}(x|\mu_{ki}, \Sigma_{ki}) \tag{3}$$

where I_k is the number of component densities used to model class k, c_{ki} are weight coefficients (with $c_{ki} > 0$ and $\sum c_{ki} = 1$), μ_{ki} is the mean vector and Σ_{ki} is the covariance matrix of component density i of class k. To avoid the problems of estimating a covariance matrix in a high-dimensional feature space, i.e. to keep the number of parameters to be estimated as small as possible, we make use of pooled covariance matrices in our experiments:

- *class specific variance pooling* :
  estimate only a single Σ_k for each class k, i.e. $\Sigma_{ki} = \Sigma_k \ \forall\ i = 1, ..., I_k$
- *global variance pooling* :
  estimate only a single Σ, i.e. $\Sigma_{ki} = \Sigma \ \forall\ k = 1, ..., K$ and $\forall\ i = 1, ..., I_k$

Furthermore, we will only use a diagonal covariance matrix, i.e. a variance vector. This does not mean a loss of information, as on the one hand a mixture density of that form can still (arbitrarily precise) approximate any density function and on the other hand the covariance matrix of our previously whitened data is known to be diagonal. ML parameter estimation is now done using the Expectation Maximization (EM) algorithm [6] combined with a Linde-Buzo-Gray based clustering procedure [7]. Note that we used global variance pooling and a maximum approximation of the EM-algorithm in our experiments. For more information on ML parameter estimation the reader is referred to [1].

5 Discriminative Training

Assume that the training data is given by 2-tupels of the form (x_n, k_n) with x_n being the observation of training sample $n \in \{1, ..., N\}$ and k_n the corresponding class label, $k_n = 1, ..., K$. The *a posteriori* probability for the class k given the observation x_n shall be denoted by $p_\lambda(k|x_n)$. Similarly, $p_\lambda(x_n|k)$ and $p(k)$ represent the according class conditional and a priori probabilities. In the following, the a priori probabilities are supposed to be given (see Chapter 4). The *maximum mutual information* criterion [8] can then be defined by the expression

$$F_{MMI}(\lambda) = \sum_{n=1}^{N} \log \frac{p(k_n) p_\lambda(x_n|k_n)}{\sum_{k=1}^{K} p(k) p_\lambda(x_n|k)}. \tag{4}$$

That is, the MMI criterion aims to maximize the sum of logarithms of the *a posteriori* probabilities $p_\lambda(k_n|x_n)$. A maximization of the MMI criterion defined above therefore tries to simultaneously maximize the class conditional probabilities of the given training samples and to minimize a weighted sum over the class conditional probabilities of all competing classes. Thus, the MMI criterion optimizes the class separability. In the following, we will present MMI reestimation formulae for the mixture density parameters, using global variance pooling.

5.1 MMI Parameter Optimization

In the following, mixture density parameters will be calculated in maximum approximation, that is we approximate sums of probabilities by the maximum addend. Performing *extended Baum-Welch* parameter optimization on the MMI criterion yields the following reestimation formulae for the means μ_{ki}, global diagonal variances σ^2 and mixture weights c_{ki} of Gaussian mixture densities (for more details on that topic, the reader is referred to [9]). Note that for ease of representation we skip the dimension index d in the following formulae.

$$\hat{\mu}_{ki} = \frac{\Gamma_{ki}(x) + D c_{ki} \mu_{ki}}{\Gamma_{ki}(1) + D c_{ki}} \tag{5}$$

$$\hat{\sigma}^2 = \frac{\sum_k D(\sigma^2 + \sum_i c_{ki}\mu_{ki}^2)}{KD} - \sum_{ki} \frac{\Gamma_{ki}(1) + Dc_{ki}}{KD}\ \hat{\mu}_{ki}^2 \tag{6}$$

$$\hat{c}_{ki} = \frac{\Gamma_{ki}(1) + Dc_{ki}}{\Gamma_k(1) + D} \tag{7}$$

with iteration constant D. $\Gamma_{ki}(g(x))$ and $\Gamma_k(g(x))$ are discriminative averages of functions $g(x)$ of the training observations, defined by

$$\Gamma_{ki}(g(x)) = \sum_n \delta_{i,i_{k,n}} \left[\delta_{k,k_n} - p_\lambda(k|x_n)\right] g(x_n) \tag{8}$$

$$\Gamma_k(g(x)) = \sum_i \Gamma_{ki}(g(x)) \tag{9}$$

$\delta_{i,j}$ is the *Kronecker delta*, i.e. given a training observation x_n of class k_n, $\delta_{i,i_{k,n}} = 1$ only if i is the 'best-fitting' component density $i_{k,n}$ given class k and $\delta_{k,k_n} = 1$ only if $k = k_n$. For fast but reliable convergence of the MMI criterion, the choice of the iteration constant D is crucial. Although there exists a proof of convergence [10], the size of the iteration constant guaranteeing convergence yields impractical small stepsizes, i.e. very slow convergence. In practice, fastest convergence is obtained if the iteration constants are chosen such that the denominators in the reestimation equations (5)-(7) and the according variances are kept positive:

$$D = h \cdot \max_{k,i} \left\{ D_{min}, \frac{1}{c_{ki}} \left(\frac{1}{\beta_k} - \Gamma_{ki}(1)\right)\right\} \tag{10}$$

$$\begin{aligned} D_{min} = \max_d & \frac{-\Gamma(x^2) + \alpha\Gamma(1) + \sum_{k,i} \left[2\Gamma_{ki}(x) - \Gamma_{ki}(1)\mu_{ki}\right]\mu_{ki}}{K(\sigma^2 - \alpha)} \\ & + \frac{\sum_{ki} \beta_k \left(\Gamma_{ki}(x) - \Gamma_{ki}(1)\mu_{ki}\right)^2}{K(\sigma^2 - \alpha)} \end{aligned} \tag{11}$$

Here, D_{min} denotes an estimation for the minimal iteration constant guaranteeing the positivity of variances and the *iteration factor* $h > 1$ controls the convergence of the iteration process, high values leading to low step sizes. The constants $\beta_k > 0$ are chosen to prevent overflow caused by low-valued denominators. In our experiments, parameter initialization is done using ML training and we chose

$$\frac{1}{\beta_k} = \max_i(|\Gamma_{ki}(1)|) + 1. \tag{12}$$

6 Results

In this chapter we will present results for the proposed classifier on the USPS database and compare these to the results obtained by the ML approach. Furthermore, we will compare our best results with those obtained by other state of the art classifiers such as support vector machines, artificial neural nets or

decision trees. For our experiments, the dimensionality of the feature space was reduced as described in Chapter 3, yielding a feature space of dimension 39. A comparison of the results obtained by ML and MMI respectively is shown in Table 1.

Table 1. Comparison of ML/ MMI (h=5, 50 iterations) results for global variance pooling with respect to total number of component densities used

#component densities	ML Error Rate [%]		MMI Error Rate [%]	
	Train	Test	Train	Test
10	17.0	13.9	11.4	10.2
20	13.1	12.0	6.4	8.1
40	10.3	9.9	3.9	6.8
80	8.2	9.2	2.2	5.8
160	6.4	8.5	1.2	6.3
320	4.6	6.8	0.34	5.9
640	3.3	6.2	0.02	5.7
1280	2.2	5.6	0.02	5.4
4965	0.66	5.2	0.01	4.7
8266	0.38	4.5	0.01	4.5
10360	0.38	4.6	0.01	4.6

We can draw the conclusion that discrimative training procedures work well for models with few parameters. Although the improvements get smaller with the number of model parameters increasing, it becomes clear that using MMI training drastically reduces the number of parameters needed to obtain good results. For instance, the error rate using a total of 80 component densities goes down from 9.2% (ML) to 5.8% (MMI), i.e. a relative improvement of nearly 40%. To obtain a similar error rate using ML, more than 1000 component densities are needed. Therefore, discriminative training criteria are very efficient for realizing fast recognizers, which can be used in real-time environments.

Our best results so far are obtained by ML training combined with the creation of virtual test samples. That is, each test sample is multiplied by shifting it into eight directions. This yields nine instances of the same test sample, which are classified separately. We then use classifier combination schemes, in this case the product rule [11], to come to a final decision for the original test sample. The basic idea behind this method is that we are able to use classifier combination rules (and their benefits) without having to create multiple classifiers. Instead, we simply create virtual test samples. Using that approach, the ML error rate goes down from 4.5% to 3.6%.

A comparison of our results with that reported by other state-of-the-art methods can be found in Table 2. Note that we only considered research groups that used exactly the same training and test sets. Without that constraint, a comparison of the training and classification methods used is not possible. Other groups for instance improved the recognition performance by adding 2.500 machine printed digits to the training set [2, 12].

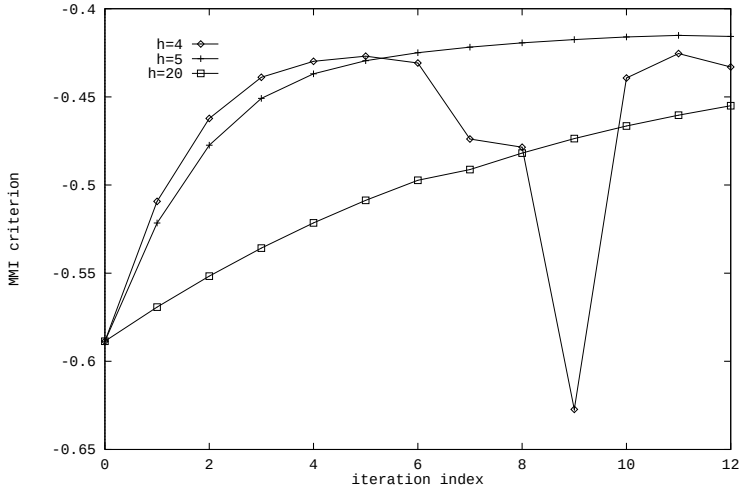

Fig. 2. MMI convergence behaviour for different h (single densities)

Table 2. Results reported on the USPS database

Method	Error Rate [%]
Human Performance [2]	2.5
Decision Tree C4.5 [13]	16.2
Two-Layer Neural Net [13]	5.9
5-Layer Neural Net (LeNet1) [13]	5.1
Support Vectors [14]	4.0
Invariant Support Vectors [15]	3.0
This work: MMI-Mixtures	4.5
ML-Mixtures	4.5
MMI-Mixtures, Product Rule	3.8
ML-Mixtures, Product Rule	3.6

Since discriminative training methods cannot guarantee convergence under realistic conditions, it is interesting to investigate the convergence behaviour. Figure 2 shows MMI convergence behaviour for single densities and different choices of the iteration factor h. As can be seen, the choice of $h = 4$ yields very fast, but unstable convergence. $h = 5$ as well as $h = 20$ lead to smooth convergence, yet the former (used in our experiments) leads to significantly faster convergence.

7 Conclusions

In this paper, we presented a discriminative training criterion for Gaussian mixture densities in image object recognition. Although we could not improve our best ML result of 3.6% on the USPS database yet, the MMI criterion is able to produce good results using only very few parameters. Furthermore it should

be noted that we have only just begun to use discriminative criteria in object recognition. Experience from speech recognition [9] raises hope to being able to improve our best results in the near future, too. For instance, the reestimation formula (7) for mixture weights c_{ki} is known to converge very slowly. We are currently implementing modified reestimation formulae which are known to give better convergence [8, 9]. Future work also includes realizing other discriminative criteria such as the *minimum classification error* criterion.

References

1. J. Dahmen, K. Beulen, H. Ney, "Objektklassifikation mit Mischverteilungen," P. Levi, R.-J. Ahlers, F. May, M. Schanz (eds.): *20. DAGM Symposium Mustererkennung 1998*, pp.167-174, Stuttgart, Germany, 1998.
2. P. Simard, Y. Le Cun, J. Denker, "Efficient Pattern Recognition Using a New Transformation Distance," S.J. Hanson, J.D. Cowan, C.L. Giles (eds.): *Advances in Neural Information Processing Systems 5*, Morgan Kaufmann, San Mateo CA, pp. 50-58, 1993.
3. K. Fukunaga, *Introduction to Statistical Pattern Recognition*, Academic Press, San Diego CA, 1990.
4. W. H. Press, S. A. Teukolsky, W. T. Vetterling, B. P. Flannery, *Numerical Recipes in C*, University Press, Cambridge, 1992.
5. R. O. Duda, P. E. Hart, *Pattern Classification and Scene Analysis*, John Wiley & Sons, 1973.
6. A.P. Dempster, N.M. Laird, D.B. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm," *Journal of the Royal Statistical Society*, 39(B), pp. 1-38, 1977.
7. Y. Linde, A. Buzo und R. M. Gray, "An algorithm for vector quantizer design," *IEEE Transactions on Communications*, Vol. 28, No. 1, pp.84-95,1980.
8. Y. Normandin, "Maximum Mutual Information Estimation of Hidden Markov Models," *Automatic Speech and Speaker Recognition*, C.-H. Lee, F.K. Soong, K.K. Paliwal (eds.), Kluwer Academic Publishers, Norwell, MA, pp.57-81, 1996.
9. R. Schlüter, W. Macherey, "Comparison of Discriminative Training Criteria," *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, Seattle, Washington, pp.493-496, May 1998.
10. L. E. Baum, J. A. Eagon, "An Inequality with Applications to Statistical Estimation for Probabilistic Functions of Markov Processes and to a Model for Ecology," *Bulletin of the American Mathematical Society*, Vol.73, pp.360-363, 1967.
11. J. Kittler, "On Combining Classifiers," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 20, No. 3, pp. 226-239, March 1998.
12. H. Drucker, R. Schapire, P. Simard, "Boosting Performance in Neural Networks," *International Journal of Pattern Recognition and Artificial Intelligence*, Vol.7, No.4, pp. 705-719, 1993.
13. V. Vapnik, *The Nature of Statistical Learning Theory*, Springer, New York, pp.142-143, 1995.
14. B. Schölkopf, *Support Vector Learning*, Oldenbourg Verlag, Munich, 1997.
15. B. Schölkopf, P. Simard, A. Smola, V. Vapnik, "Prior Knowledge in Support Vector Kernels," M. Jordan, M. Kearns, S. Solla (eds.): *Advances in Neural Information Processing Systems 10*, MIT Press, pp. 640-646, 1998.

This article was processed using the LaTeX macro package with LLNCS style