SVMs, Gaussian Mixtures, and their Generative/Discriminative Fusion

Thomas Deselaers, Georg Heigold, and Hermann Ney

[1] Computer Science Department – RWTH Aachen University, Aachen, Germany
E-mail: lastname@cs.rwth-aachen.de

Abstract

We present a new technique that employs support vector machines and Gaussian mixture densities to create a generative/discriminative joint classifier. In the past, several approaches to fuse the advantages of generative and discriminative approaches were presented, often leading to improved robustness and recognition accuracy. The presented method directly fuses both approaches, effectively allowing to fully exploit the advantages of both. The fusion of SVMs and GMDs is done by representing SVMs in the framework of GMDs without changing the training and without changing the decision boundary. The new classifier is evaluated on four tasks from the UCI machine learning repository. It is shown that for the relatively rare cases where SVMs have problems, the combined method outperforms both individual ones.

1. Introduction

Two major approaches to the classification of patterns are known: generative and discriminative approaches. Both have been successfully applied to many different problems and both have their own advantages and disadvantages.

Generative approaches try to find an optimal representation of the original data by keeping as much information as possible. They can be trained from partly labelled data and normally allow for a reconstruction of the most likely prototype for each modelled class. Generative methods can be built very robustly. Discriminative methods require fully labelled training data, can be applied very quickly and often show better recognition accuracy than their generative counterparts. The biggest problem of many discriminative approaches is that they are prone to overfitting, which requires significant extra effort to be overcome, e.g. the max-margin concept in SVMs is all about reducing overfitting.

Clearly, both approaches have their advantages and several authors have tried to combine the approaches to benefit from both. One common approach is followed in the object recognition literature, the two worlds are fused in a two-stage method: a generative model is used to create a fixed length representation of the image, which then is classified using a discriminative technique (e.g. [3, 4]).

A direct approach to joining the two principles is proposed in [10] which allows to seamlessly blend from a fully discriminative model to a fully generative model. In [5], a discriminative, boosted model is modified to account for reconstruction in addition to the discriminatory performance and a clear performance boost for noisy data was observed. In [9], the opposite approach is taken, where boosting is performed with Gaussians as weak classifiers. In many areas, such as speech recognition, discriminatively trained Gaussian Mixtures are frequently used [11].

Among the discriminative models, support vector machines (SVMs) are popular in many domains. They are easy to use and often obtain good results [13]. SVMs do not model a probability distribution, and are thus not open to the ideas presented in [10].

Despite the fact that SVMs are in general among the most successful and best understood methods, where finding a good set of parameters is relatively easy and standard procedures are known (i.e. cross validation on the training data), in some cases tuning the parameters of an SVM to obtain optimal performance turns an SVM into "*little more than a glorified template matcher*" [8]. This is in accordance to the observation, that an SVM (with radial basis function (RBF) kernel, which is probably the most commonly used kernel) in some cases has a large portion of the training data as support vectors (SVs) and thus it degenerates to a *discriminatively weighted* kernel densities classifier. This degeneration can be interpreted as effectively overfitting to the training data.

We present an approach that fuses an SVM with a generatively trained Gaussian Mixture Density (GMD) classifier and thereby profits from the advantages of both techniques. A close connection between Gaussian mixtures and SVMs was already discussed in [14], but to the best of our knowledge, the direct fusion of SVMs and GMDs has not yet been investigated. To fuse the two approaches, we first convert the SVM into a GMD with identical decision boundary and then blend this GMD with another (generatively trained) GMD to obtain a joint classifier.

Another way to fuse SVMs and GMDs is to compute their individual posterior probabilities and combine these. To obtain probabilities from an SVM, other approaches have been proposed e.g. in [12, 15, 16]. The method proposed here is not a late combination of two different classifiers, but a unified framework, to fuse the two classification methods into a single joint classifier.

2. Support Vector Machines

SVMs being a modern, well understood and widely used classifier, directly predict the label of an observation. An

SVM commonly discriminates between two classes: -1 and 1 using the decision rule

$$X \mapsto \operatorname{sgn}\left(\sum_{v_i \in \mathcal{S}} \alpha_i K(X, v_i) + \alpha_0 \right) \tag{1}$$

to classify the observation X where K is a kernel function, the v_i are the support vectors (SVs) and the α_i are the corresponding weights, α_0 is a bias term.

We consider the distance to the decision hyper-plane to be proportional to a class-conditional emission probability, i.e., we assume that given a class, the confidence for an observation vector X which is far away from the hyper plane is high, and conversely, that for each vector which is close to the hyper plane, the confidence that this vector comes from the class is low. Thus, we write

$$p(X|k) \propto \sum_{v_i \in \mathcal{S}_k} k\alpha_i K(X, v_i) + \alpha_0 \tag{2}$$

where S_k is the set of SVs for class k, i.e., those SVs with positive α_i for class $k = +1$ and those with negative α_i for class $k = -1$.

3. Gaussian Mixture Densities

GMDs are a *generative* model, Bayes decision rule is used for classification:

$$X \mapsto \arg\max_k \{p(k|X)\} \tag{3}$$

$$= \arg\max_k \left\{ p(k) \sum_i p(i|k) \mathcal{N}(X|\mu_{ki}, \Sigma_{ki}) \right\} \tag{4}$$

where class k is represented by I_k clusters, $p(i|k)$ are the cluster weights and $\mathcal{N}(X|\mu_{ki}, \Sigma_{ki})$ is the Gaussian representing the i-th cluster of class k with mean μ_{ki} and covariance matrix Σ_{ki}.

GMDs are trained using the EM algorithm to maximise the likelihood $\prod_{n=1}^{N} p(X_n|k_n)$ [2] by starting with an initial Gaussian over all observations which is iteratively split and reestimated until a certain number of densities is obtained. Densities with too few observations are deleted to ensure stable estimation.

4. Fusing SVMs and GMDs

As described above, SVMs are a discriminative classifier and GMDs are a generative classifier. In the following, we first describe how SVMs with RBF kernel can be represented in the form of GMDs without changing the decision boundary and then describe how two GMDs can be fused to profit from their individual advantages.

4.1. Approximating SVMs Using GMDs

Originally SVMs are designed to discriminate two classes. We describe the transformation for the two-class case first and then we extend this transformation to the multi-class case.

Two-Class Case. It is known that SVMs and GMDs can in principle model arbitrary decision boundaries and thus, can theoretically represent the respective other without any loss of accuracy or generalisation ability. This theoretical feature, however does not pose any advantage as normally the most difficult thing for a classifier is to find the model parameters, and thus it is not clear how to benefit from these theoretical equivalence here.

For SVMs with an exponential RBF kernel, a close similarity between SVMs and GMDs can be observed. Starting from the general form of the decision function, we give straight-forward rules to transform one into the respective other without changing the decision boundary.

The decision rule of a standard SVM (Eq. (1)), can be rewritten, by inserting the RBF kernel, as

$$X \mapsto \arg\max_{k \in \{-1,1\}} \left\{ \sum_{v_i \in \mathcal{S}_k} k\alpha_i e^{\left(-\gamma ||X - v_i||^2\right)} + \alpha_0 \right\}. \tag{5}$$

For comparison we give the decision rule of a GMM classifier, which is independent of the number of classes considered. Here, we use Gaussians with a globally pooled, diagonal covariance σ^2 and means μ_{ki}.

$$X \mapsto \arg\max_k \left\{ \sum_i \frac{p(k)p(i|k)}{(2\pi\sigma^2)^{D/2}} e^{\left(-\frac{1}{2}\frac{||x-\mu_{ki}||^2}{\sigma^2}\right)} \right\} \tag{6}$$

Now it can be seen, that Eq. (5) and (6) are identical except for the α_0 if the means μ_{ki} and the SVs v_i correspond. In fact, a GMD can be transformed into an SVM (and vice versa) by setting

$$k\alpha_i = \frac{p(k)p(i|k)}{(2\pi\sigma^2)^{D/2}} \qquad \gamma = \frac{1}{2\sigma^2} \qquad \mu_{ki} \quad = v_i \tag{7}$$

and α_0 can be sufficiently well approximated by an additional density with arbitrary mean and very high variance and a cluster weight proportional to α_0.

Thus, the main difference between a GMD and an SVM with RBF kernel is the optimisation criterion and the training method.

Multi-Class Case. The earliest used implementation for SVM multi-class classification is probably the "*one-against-the-rest*" (also known as "*one-against-all*") method, which has been used to extent other binary classifiers to multi-class problems before [6]. Therefore, not a single classifier is trained to discriminate between all classes at once but a classifier is trained for each class to discriminate it from all other classes and the decision is drawn according to the scores from these individual decisions. The decision rule in this case is:

$$X \mapsto \arg\max_k \left\{ \sum_{v_i \in \mathcal{S}_k} \alpha_{ki} K(x, v_i) + \alpha_{k0} \right\}, \tag{8}$$

where the parameters for each class k are optimised individually considering the two-class problem where all competing classes are considered to be from class -1 and class k is considered to be class 1.

Here, the relationship to the GMD classifier is similar to the two-class case, if this SVM is converted into a GMD, each SV becomes a mixture mean, we assume a pooled, diagonal covariance matrix with identical entries for each dimension inversely proportional to γ and the cluster weights are given through the weights α_i of the SVs. The transfor-

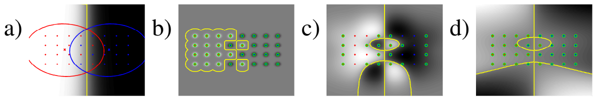

Figure 1. (a) a single density Gaussian classifier, the variance is given by the ellipse and the mean is denoted by a small star (b)-(d) support vector machines with (b) $\gamma = 100$, (c) $\gamma = 10$, (d) $\gamma = 2$. White areas denote high probabilities for the red class and dark areas denote high probabilities for the blue class, the decision boundary is yellow and SVs are denoted with green circles.

mation rules can be given analogously to those presented in Eqs. (7). It is necessary to address the class-wise constant bias terms α_{k0} which can be substituted by very diffuse Gaussians (one per class) with an arbitrary mean and a weight proportional to α_{k0}. Negative weights α_{ki} are compensated by adding respective densities to all other classes.

The same transformation can be applied if the SVM is trained to jointly discriminate all classes as described in [17] because the same decision rule is applied there and only the parameter estimation is done differently.

4.2. Fusing SVMs and GMDs

Given two GMDs $\mathcal{G}_1$ and $\mathcal{G}_2$ (let $t = 1, 2$)

$$\mathcal{G}_t = ((\mu_{t1} \dots \mu_{tI}), (\sigma_{t1} \dots \sigma_{tI}), (p_t(1) \dots p_t(I)) \qquad (9)$$

one trained using the EM algorithm for GMDs and the other obtained by transforming an SVM, it is possible to fuse both GMDs into one and arbitrarily fade between the two. The new, joint GMD $\mathcal{G}'$ is obtained as

$$\mathcal{G}' = \big((\mu_{11} \dots \mu_{1I}, \mu_{21} \dots \mu_{2J}), (\sigma_{11} \dots \sigma_{1I}, \sigma_{21} \dots \sigma_{2J}), \\ (wp_1(1) \dots wp_1(I), (1-w)p_2(1) \dots (1-w)p_2(J))\big)$$

where w is a weighting factor allowing to smoothly blend between $\mathcal{G}_1$ (for $w = 1$) and $\mathcal{G}_2$ (for $w = 0$).

Since the cluster weights of $\mathcal{G}_1$ and $\mathcal{G}_2$ are normalised, for $0 \leq w \leq 1$ the cluster weights of the resulting GMD $\mathcal{G}'$ are also normalised.

The resulting decision boundary, now is chosen according to a combination of the optimisation criteria of the SVM, which optimises classification performance, and the GMD, which optimises data representation. Thus, the resulting decision boundary is not-optimal wrt. either of these criteria, but according to some compromise of these.

In Fig. 1 an example GMD (1 density per class) and three differently parametrised SVMs are visualised for two-dimensional data. It can be seen that the SVMs have, depending on the scale of the kernel γ, many SVs, which is an indicator for possible overfitting. As will be experimentally shown, overfitting of SVMs to the training data is a problem in cases where the data is difficult to separate, which commonly goes along with a very high numbers of SVs. For GMDs, the number of parameters estimated can be fixed by the user (i.e. fix number of densities), thus by forcing the number of parameters to be small, overfitting can easily be avoided.

In Fig. 2, the GMD from Fig. 1a is fused with the three different SVMs from Fig. 1b-d with different weights w (w is the weight for the GMD obtained from the SVM). The smoothing of the probability distribution and thereby of the decision boundary can clearly be observed. The effect is best observed in the top row of Fig. 2, which shows a combination of the SVM with $\gamma = 0.01$ (cp. Fig. 1b) with the GMD (Fig. 1a). The before extremely bumpy decision boundary of the SVM is strongly smoothed and only when the SVM gets relatively high weight a tendency to overfitting can be observed. Similarly, the decision boundaries for the combinations with the other two SVMs are smoothed when combined with the GMD.

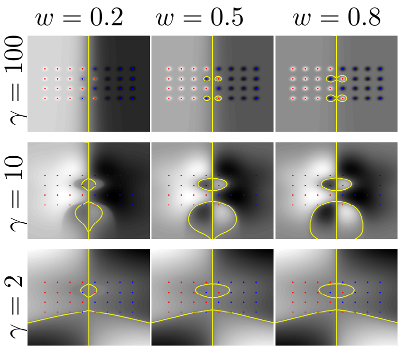

Figure 2. Fusing the Gaussian classifier from Fig. 1a with the SVMs from Fig. 1b-d using different weights. The decision boundary is plotted as a yellow line.

Table 1. Overview of the datasets used, C number of classes; N total number of vectors; D dimensionality of the vectors.

Dataset	C	N	D
Diabetes	2	768	8
German	2	1,000	24
Heart[†]	2	270	25
Vehicle	4	846	18

[†] categorical features were expanded (original dim. 13)

5. Experiments

Experimentally, we evaluate the proposed method on four datasets from the UCI machine learning repository[1] [1]. An overview over the datasets used is given in Table 1. These datasets were selected from the UCI repository by selecting those where classification is difficult, i.e. those where reported error rates are rather high. For all experiments we normalised the mean and the variance of all features to 0 and 1, respectively as recommended for the use with SVMs.

First, we present the experimental results using only SVMs and using only GMDs. We used libSVM[2] with one-against the rest training [7] and the common grid search on 5-fold cross validation (11 values for C, 10 values for γ) to determine the parameters γ and C for the SVM. The results for the SVMs and the GMDs (with 1, 2, and 32

[1] http://archive.ics.uci.edu/ml/index.html
[2] http://www.csie.ntu.edu.tw/~cjlin/libsvm/

Table 2. Results using SVMs and GMDs. We give the result for the SVM using the parameters determined on the data in 5-fold cross validation. For the SVM we also give the number of SVs in percentage. For GMD classifiers, we give three results for each database, using 1, 2, and 32 densities per class, respectively.

	SVM		GMD ER [%]		
Dataset	ER [%]	SVs [%]	1d	2d	32d
Diabetes	29.9	50.0	28.6	30.5	24.7
German	24.5	54.4	24.0	26.5	30.0
Heart	25.9	56.0	22.2	22.2	27.8
Vehicle	60.2	50.7	53.8	49.1	35.1

Table 3. Results of fusing SVMs and GMDs with $w = 0.5$.

	ER [%]	
Dataset	1 dens.	32 dens.
Diabetes	30.5	27.3
German	22.5	33.0
Heart	22.2	18.5
Vehicle	55.0	35.7

densities/class) are reported in Table 2.

It can be observed that the error rates are in general quite high which shows that the selected tasks can be considered difficult. As expected, the SVMs decided to choose a significant part of the training data as SVs and thus the SVM is on the best way to overfitting. The GMDs mostly have better results (on the test data) than the SVMs, although the SVMs have far better error rates on the training data (not reported here), which is an indicator for overfitting effects.

The results of fusing the classifiers using the SVM and GMDs with 1 and 32 densities are given in Table 3. For these experiments, we set $w = 0.5$. For the german-task and the heart task, the fused classifiers outperform their individual components, for the diabetes-task and for the vehicle-task, only the SVM is outperformed and the performance is similar to the GMD alone. Not surprisingly, for the vehicle- and diabetes-tasks the combination has better results if more densities are used, because here the GMDs were better with more densities. We assume that thus effectively the overfitting of the SVM is smoothed away by mixing with the GMD model. Informal experiments showed that for each of these tasks, improvements are possible by using different numbers of densities in the GMD and by using different weights w in the fusion, these results are omitted due to brevity constraints.

6. Conclusion

We presented a novel generative/discriminative classifier consisting of fusing a generative GMD classifier and an SVM with RBF kernel. We have shown that the combined method is able to overcome overfitting problems of the standard training procedure for SVMs on some tasks.

Concluding, we do not generally recommend to use the presented technique for arbitrary problems but rather only when the SVM alone suffers badly from overfitting problems (which may happen in strongly overlapping problems) or has a high number of SVs. For most tasks this is not the case and SVMs are known to be a well-understood and easily usable classification technique. However, the tasks presented here are different from most tasks in that respect as the SVMs here tend to overfit, i.e. choose a huge amount of training samples as SVs.

References

[1] A. Asuncion and D. J. Newman. UCI machine learning repository, 2007.

[2] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algoritm. *J Royal Statistical Society, Series B*, 39(1):1–38, 1977.

[3] T. Deselaers, D. Keysers, and H. Ney. Discriminative training for object recognition using image patches. In *CVPR*, pp. 157–162, 2005.

[4] G. Dorkó and C. Schmid. Object class recognition using discriminative local features. Rapport de recherche RR-5497, INRIA - Rhone-Alpes, 2005.

[5] H. Grabner, P. M. Roth, and H. Bischof. Eigenboosting: Combining discriminative and generative information. In *CVPR*, 2007.

[6] C.-W. Hsu and C.-J. Lin. A comparison of methods for multi-class support vector machines. *Trans. Neural Networks*, 13(2):415–425, 2002.

[7] T. K. Huang, R. Weng, and L. C. J. Generalized bradley-terry models and multi-class probability estimates. *JMLR*, 7:85–15, 2006.

[8] Y. LeCun, S. Chopra, M. A. Ranzato, and F.-J. Huang. Energy-based models in document recognition and computer vision. In *ICDAR*, 2007.

[9] B. Lin, X. Wang, R. Zhong, and Z. Zhuang. Continuous optimization based-on boosting gaussian mixture model. In *ICPR*, pp. 1192–1195, 2006.

[10] T. Minka. Discriminative models, not discriminative training. Technical Report TR-2005-144, Microsoft Research Cambridge, Cambridge, UK, 2005.

[11] Y. Normandin, R. Lacouture, and R. Cardin. MMIE training for large vocabulary continuous speech recognition. In *ICSLP*, pages 1367–1370, 1994.

[12] J. C. Platt. Probabilities for support vector machines. In *Advances in Large Margin Classifiers*, MIT Press, pp. 61–74, 1999.

[13] B. Schölkopf and A. J. Smola. *Learning with Kernels - Support Vector Machines, Regularisation, Optimization, and Beyond.* MIT Press, 2002.

[14] B. Schölkopf, K.-K. Sung, C. Burges, F. Girosi, P. Niyogi, T. Poggio, and V. Vapnik. Comparing support vector machines with gaussian kernels to radial basis function classifiers. *Trans. Signal Processing*, 45(11):2758–2765, 1997.

[15] M. Seeger. Probabilistic interpretations of support vector machines and other spline smoothing models. In *EuroCOLT*, 1999.

[16] P. Sollich. Probabilistic interpretation and bayesian methods for support vector machines. In *ICANN*, pp. 91–96, 1999.

[17] J. Weston and C. Watkins. Support vector machines for multiclass pattern recognition. In *Seventh European Symposium On Artificial Neural Networks*, 1999.