Pairwise Coupling for Machine Recognition of Hand-Printed Japanese Characters

Volker Roth
Department of Computer Science III
University of Bonn
D-53117 Bonn, Germany

Koji Tsuda
AIST Computational Biology Research Center
135-0064 Tokyo, Japan

Abstract

Machine recognition of hand-printed Japanese characters has been an area of great interest for many years. The major problem with this classification task is the huge number of different characters. Applying standard "state-of-the-art" techniques, such as the SVM, to multi-class problems of this kind imposes severe problems, both of a conceptual and a technical nature: (i) separating one class from all others may be an unnecessarily hard problem; (ii) solving these subproblems can impose unacceptably high computational costs. In this paper, a new approach to Japanese character recognition is presented that successfully overcomes these shortcomings. It is based on a pairwise coupling procedure for probabilistic two-class kernel classifiers. Experimental results for Hiragana recognition effectively demonstrate that our method attains an excellent level of prediction accuracy while imposing very low computational costs.

1. Introduction

Offline machine recognition of hand-printed Japanese or Chinese characters is one of the most important classification problems with a very large number of different classes. Problems of this kind require us to put special emphasis on the multi-class aspect of deriving classification rules: simultaneously estimating a large number of class boundaries is usually a much harder task than solving standard two-class problems.

The usual way of handling multi-class problems is the following: a problem with c classes is treated as a collection of c "one-against-all-others" subproblems, together with some principle of combining the c outputs. This approach, however, bears two main disadvantages: (i) separating one class from all others may be an unnecessarily hard problem that often requires us to apply very complex classification models. For highly complex models it is often difficult to avoid overfitting phenomena in order to guarantee good generalization properties; (ii) all c subproblems are stated as optimization problems over the *full* learning set. For kernel classifiers applied to large-scale problems, this can impose unacceptably high computational costs.

The former problem is of a conceptual nature and can be overcome by a different approach to the multi-class problem: instead of solving c one-against-all problems, we might solve $c(c-1)/2$ pairwise classification problems, and try to couple the probabilities in a suitable way. Methods of this kind have been introduced in [1], [2] and are referred to as *pairwise coupling*. Learning such pairwise decision rules may be a much simpler problem than separating each class from the others.

In this paper special emphasis is put on nonlinear *Mercer kernel-based* classifiers. A recent overview over kernel methods can be found in [3]. Since for kernel methods the computational efficiency is mostly determined by the number of training samples, the pairwise coupling scheme also overcomes the numerical problems of the one-against-all strategy: it is much easier to solve $c(c-1)/2$ small problems that to solve c large problems. For the class of *probabilistic kernel classifiers* we have in mind, we show that this leads to a reduction of computational costs that scales linear in the number of classes. The availability of probabilistic outputs constitutes another advantage over the SVM: it allows us to quantify a confidence level for class assignments. For text recognition systems in particular, such posterior estimates of class membership are of great practical value: having available a weighted list of possible character assignments may improve the accuracy drastically when coupling optical character recognition techniques with semantic methods.

This paper is organized as follows: in section 2 *Nonlinear Kernel Discriminant Analysis* (NKDA) is introduced as a probabilistic kernel classifier. It can be used as a "building block" in the pairwise coupling procedure that is described

in section 3. Section 4.1 presents an intuitive toy example that demonstrates the differences between the pairwise coupling mechanism and conventional approaches to multiclass problems. We conclude this paper with performance studies for a large-scale dataset of handwritten Hiraganas in section 4.2. These experiments effectively demonstrate that the proposed method attains a level of accuracy even superior to the SVM, while additionally providing the user with posterior estimates of class membership. Moreover, concerning the computational costs, it significantly outperforms one of the best SVM optimization packages available.

2. Probabilistic kernel classifiers

The problem of classification formally consists of assigning observed vectors $\boldsymbol{x} \in \boldsymbol{R}^d$ into one of c classes. A *classifier* is a mapping that assigns labels to observations. In practice, a classifier is trained on a set of observed i.i.d. data-label pairs $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$, drawn from the unknown joint density

$$p(\boldsymbol{x}, y) = p(y|\boldsymbol{x})p(\boldsymbol{x}).$$

Classifiers can be partitioned into two main groups, namely *informative* and *discriminative* ones. In the informative approach, the classes are described by modeling their structure, i.e. their generative statistical model. Starting from these class models, the posterior distribution of the labels is derived via the Bayes formula. The most popular method of informative kind is classical *Linear Discriminant Analysis* (LDA). The availability of explicit class models has several advantages. For instance, it is easy to deal with incomplete or uncertain measurements. The inclusion of partially unlabeled data is an example of this kind, cf. [4]. However, the informative approach has a clear disadvantage: modeling the classes is usually a much harder problem than solving the classification problem directly.

In contrast to the informative approach, discriminative classifiers focus on modeling the decision boundaries or the class probabilities directly. No attempt is made to model the underlying class densities. In general, they are more robust as informative ones, since less assumptions about the classes are made.

Presumingly most popular discriminative method is the *Support Vector Machine* (SVM). Within a maximum entropy framework, it can be viewed as the classification model that makes the least assumptions about the estimated model parameters, cf. [5]. Thus, from a statistical viewpoint, it should be the most robust method. The main drawback of the SVM, however, is the absence of probabilistic outputs: in the SVM framework the classification task is considered readily solved by predicting the class labels. (Strategies for approximating SVM posterior estimates in a post-processing step have been reported in the literature, see e.g. [6, 7]. In this paper, however we restrict our attention to fully probabilistic models.)

In this paper we focus on strategies for coupling pairwise decision rules into into a joint posterior probability estimate for all c classes. A necessary condition for methods of this kind, however, is the availability of posterior estimates in each of the $c(c-1)/2$ subproblems. Thus, we are restricted to probabilistic classifiers like LDA. Since linear decision boundaries often do not adequately separate the classes, we are mainly interested in a generalized *kernel variant* of this classical method.

For convenience in what follows, in this section we restrict ourselves to the problem of separating only *two* classes. These two-class methods will then be "plugged" into a pairwise coupling scheme.

2.1. Nonlinear Kernel Discriminant Analysis

The central idea of *informative* classifiers is to model the conditional class densities $p(\boldsymbol{x}|y)$. Assuming a parameterized class conditional density $p_{\boldsymbol{\theta}_j}(\boldsymbol{x}|y=j)$ and collecting all model parameters in a vector $\boldsymbol{\theta}$, these parameters are estimated by maximizing the full log likelihood

$$\hat{\boldsymbol{\theta}}_{ML} = \operatorname{argmax}_{\boldsymbol{\theta}} \textstyle\sum_{i=1}^N \log p_{\boldsymbol{\theta}}(\boldsymbol{x}_i, y_i). \tag{1}$$

Classical *Linear Discriminant Analysis* (LDA) can be viewed as the simplest classifier of this kind. The classes are modeled by a parameterized multivariate Gaussian model, with the additional assumption that all classes share a common covariance matrix Σ:

$$p_{\boldsymbol{\theta}}(\boldsymbol{x}|y=j) = \mathcal{N}(\boldsymbol{x}; \mu_j, \Sigma). \tag{2}$$

Considering two-class problems, we define a *discriminant function* between the two classes with labels $\{+1, -1\}$ as

$$g(\boldsymbol{x}) = \log \tfrac{P(y=+1|\boldsymbol{x})}{P(y=-1|\boldsymbol{x})}. \tag{3}$$

It can be shown easily that the model assumptions (2) lead to a discriminant function that is linear in $\boldsymbol{x}$:[1]

$$g(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x}. \tag{4}$$

There are several approaches to restating classical LDA with Mercer kernels, see e.g. [4] and [8]. Since space here precludes a more detailed discussion, we only present a version that exploits the well-known analogy between LDA and linear indicator regression against the binary class labels (an early reference is [9], p. 152). The LDA solution can be found by regressing the input vectors $\boldsymbol{x}_i$ (summarized as rows of the "design" matrix X) against a binary target vector $\boldsymbol{y}$ with entries $\{+1, -1\}$:

$$\hat{\boldsymbol{w}} = \operatorname*{argmin}_{\boldsymbol{w}} \|\boldsymbol{y} - X\boldsymbol{w}\|^2. \tag{5}$$

[1] Throughout this paper we have dropped the constant b in the more general form $g(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x} + b$. This can be justified by assuming that the data vectors are augmented by an additional entry of one.

This corresponds to maximizing the full log likelihood as in (1). Direct optimization of the likelihood, however, often leads to severe overfitting problems, and in a Bayesian framework, a preference for smooth functions is usually encoded by introducing *priors* over the weights $\boldsymbol{w}$. In a regularization context, such prior information can be interpreted as adding some *bias* to maximum likelihood parameter estimates in order to reduce the estimator's variance. The common choice of a spherical Gaussian prior distribution with covariance $\Sigma_w \propto \lambda^{-1} I$ leads to a *ridge regression* model, [10]. The regularized version of (5) then reads

$$\hat{\boldsymbol{w}} = \underset{\boldsymbol{w}}{\operatorname{argmin}} \; \|\boldsymbol{y} - X\boldsymbol{w}\|^2 + \lambda \boldsymbol{w}^T \boldsymbol{w}. \tag{6}$$

It is known that the optimizing weight vector can be expanded in terms of the input vectors, $\boldsymbol{w} = \sum_{i=1}^{N} \boldsymbol{x}_i \alpha_i = X^T \boldsymbol{\alpha}$, cf. [4]. Substituting this expansion of $\boldsymbol{w}$ into (6) and introducing the dot product matrix $(K)_{ij} = (\boldsymbol{x}_i \cdot \boldsymbol{x}_j)$, $K = XX^T$, we can restate the minimization problem in terms of the vector of expansion coefficients $\boldsymbol{\alpha}$. Differentiating in $\boldsymbol{\alpha}$ leads us to the stationary condition:

$$\hat{\boldsymbol{\alpha}} = (K + \lambda I)^{-1} \boldsymbol{y}. \tag{7}$$

With the usual kernel trick, the dot products can be substituted by kernel functions satisfying Mercer's condition, in order to obtain a nonlinear variant of discriminant analysis. We refer to this model as *Nonlinear Kernel Discriminant Analysis* (NKDA). Given the optimal $\hat{\boldsymbol{\alpha}}$, we can make predictions for a new observation $\boldsymbol{x}_*$ by evaluating the discriminant function:

$$g(\boldsymbol{x}_*) = \sum\nolimits_{i=1}^{N} k(\boldsymbol{x}_*, \boldsymbol{x}_i) \hat{\alpha}_i, \tag{8}$$

where $k(\cdot, \cdot)$ denotes the kernel function. Note that (7) defines a system of linear equations: $K'\boldsymbol{\alpha} = \boldsymbol{y}$, with $K' = (K + \lambda I)$. For such systems there exist efficient approximation schemes: the matrix K' is symmetric, and the optimizing vector $\hat{\boldsymbol{\alpha}}$ can be computed in a highly efficient way by applying approximative *conjugate gradient* inversion techniques, cf. [11], p. 83. The key idea is to relate the problem of finding a solution to a system of equations to that of maximizing a function: the quadratic form

$$f(\boldsymbol{\alpha}) = \boldsymbol{y}\boldsymbol{\alpha} - \tfrac{1}{2}\boldsymbol{\alpha} K' \boldsymbol{\alpha} \tag{9}$$

is maximized when its gradient

$$\nabla f = K'\boldsymbol{\alpha} - \boldsymbol{y} \tag{10}$$

is zero, which is equivalent to finding the coefficient vector $\boldsymbol{\alpha}$ that solves (7). The optimization strategy can now be outlined as follows: starting with an initial vector $\boldsymbol{\alpha}^0$, a succession of search directions $\boldsymbol{h}^k$ and improved solution vectors $\boldsymbol{\alpha}^k$ is generated. After N iterations one is guaranteed to reach the solution of (7). However, in most practical problems, excellent approximations can be obtained after $k \ll N$ iterations. This property allows us to handle large-scale problems in a highly efficient way.

3. Pairwise coupling for multi-class problems

Typically two-class problems tend to be much easier to learn than multi-class problems. While for two-class problems only one decision boundary must be inferred, the general c-class setting requires us to apply a strategy for coupling decision rules. In the standard approach to this problem, c two-class classifiers are trained in order to separate each of the classes against all others. These decision rules are then coupled either in a probabilistic way (e.g. for LDA) or by some heuristic procedure (e.g. for the SVM).

A different approach to the multi-class problem was proposed in [2]. The central idea is to learn $c(c-1)/2$ pairwise decision rules and to couple the pairwise class probability estimates into a joint probability estimate for all c classes. It is obvious, that this strategy is only applicable for pairwise classifiers with probabilistic outputs.[2] From a theoretical viewpoint, pairwise coupling bears some advantages: (i) jointly optimizing over all c classes may impose unnecessary problems, pairwise separation might be much simpler; (ii) we can select a highly specific model for each of the pairwise subproblems.

Concerning *kernel classifiers* in particular, pairwise coupling is also attractive for practical reasons. For kernel methods, the computational cost are dominated by the size of the training set, N. For example, conjugate gradient approximations for NKDA scale as $\mathcal{O}(N^2 \cdot m)$, with m denoting the number of conjugate-gradient iterations (at each stage of iteration a $N \times N$-matrix-vector product must be computed). Keeping m fixed leads us to a $\mathcal{O}(N^2)$ dependency as a lower bound on the real costs.[3] Let us now consider c classes, each of which contains N_c training samples. For a one-against-all strategy, we have costs scaling as $\mathcal{O}(c(cN_c)^2) = \mathcal{O}(c^3 N_c^2)$. For the pairwise approach, this reduces to $\mathcal{O}(1/2\, c(c-1)(2N_c)^2) = \mathcal{O}(2(c^2 - c)N_c^2)$. Thus, we have a reduction of computational costs inverse proportional to the number of classes.

Pairwise coupling can be formalized as follows: considering a set of events $\{A_i\}_{i=1}^{c}$, suppose we are given pair-

[2] In a former version of [2], available as Tech. Rep. at the University of Toronto, it has been suggested to apply "approximative" pairwise coupling to the SVM. However, we feel that this approach is not very promising since it lacks a clear probabilistic interpretation.

[3] For the SVM, the situation is more difficult and heavily depends on implementation details. As an example, the popular LOQO package [12], has even $\mathcal{O}(N^3)$ complexity, due to a Cholesky decomposition of a $N \times N$ matrix. Working-set methods are usually much more efficient, but their performance is problem-dependent, and thus difficult to analyze.

wise probabilities $r_{ij} = \text{Prob}\,(A_i | A_i \text{ or } A_j)$.[4] Our goal is to couple the r_{ij}'s into a common set of probabilities $p_i = \text{Prob}\,(A_i)$. This problem has no general solution, but in [2] the following approximation is suggested: introducing a new set of auxiliary variables

$$\mu_{ij} = \frac{\hat{p}_i}{\hat{p}_i + \hat{p}_j}, \tag{11}$$

we wish to find $\hat{p}_i$'s such that the corresponding $\hat{\mu}_{ij}$'s are in some sense "close" to the observed r_{ij}'s. A suitable closeness measure is the Kullback-Leibler divergence between r_{ij} and $\hat{\mu}_{ij}$

$$\mathcal{D}^{KL} = \sum_{i<j} r_{ij} \log \frac{r_{ij}}{\hat{\mu}_{ij}} + (1 - r_{ij}) \log \frac{1 - r_{ij}}{1 - \hat{\mu}_{ij}}. \tag{12}$$

The associated stationary condition reads

$$\sum_{j \neq i} \hat{\mu}_{ij} = \sum_{j \neq i} r_{ij}\,, \quad i = 1, \ldots, c, \tag{13}$$

subject to $\sum_{i=1}^{c} p_i = 1$. Starting with an initial guess for the $\hat{p}_i$ and corresponding $\hat{\mu}_{ij}$'s, we can compute the $\hat{p}_i$'s that minimize (12) by iterating

- $\hat{p}_i \leftarrow \hat{p}_i \cdot \frac{\sum_{j \neq i} r_{ij}}{\sum_{j \neq i} \hat{\mu}_{ij}}$
- renormalize the $\hat{p}_i$'s and recompute the $\hat{\mu}_{ij}$.

Suppose, we have successfully trained all probabilistic pairwise classifiers. Then, we can predict the class membership of a new observation $\boldsymbol{x}_*$ as follows:

1. Evaluate the $c(c-1)/2$ classification rules to obtain $r_{ij}(\boldsymbol{x}_*) = 1/[1 + \exp\{g_{ij}(\boldsymbol{x}_*)\}]$, where $g_{ij}(\boldsymbol{x}_*)$ is the learned decision function between classes i and j, evaluated at point $\boldsymbol{x}_*$.
2. Compute the $\hat{\mu}_{ij}$'s based on the initial $\hat{p}_i$'s, and run the above iterations until some convergence criterion is met.
3. We finally obtain the estimated posterior probabilities for class membership of pattern $\boldsymbol{x}_*$.

4. Experiments

4.1. A simple toy example

Here we present a simple two-dimensional toy-examples that demonstrate the advantages of pairwise coupling over conventional multi-class approaches. Consider for example three classes as depicted in figure 1. In a pairwise approach, each of the three pairs can be separated by a linear decision boundary without errors. If, on the other hand, we try to separate each class from the two others, the classes are not linearly separable. In order to avoid training errors, we must use a nonlinear classifier that is able to produce decision boundaries as depicted in the graph. This example shows that the latter strategy may impose unnecessarily hard subproblems.

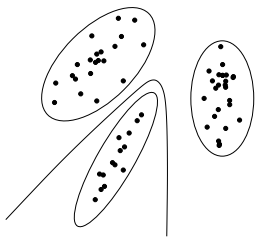

Figure 1. Separating the 3 classes with a "one-against-all-others" strategy requires us to use a nonlinear classifier.

4.2. Hiragana Recognition

In this section we compare the pairwise coupled NKDA classifier with other popular methods. We consider the problem of recognizing 71 classes of handwritten Hiraganas from the ETL9B database[5]. This database contains 200 black-and-white images of size 64×64 pixels per character. Some sample images from this dataset are depicted in figure 2.

Figure 2. Example images from the ETL9B database.

For the feature extraction, we used the *contour direction histogram features*, see [13]. This feature set represents each image as a 196 dimensional vector. Roughly speaking, the components of this feature vector encode the empirical distribution of black pixels under several directions.

In a first experiment, we compared the prediction accuracy of a 5-nearest-neighbor (5-NN) classifier and several kernel-based methods. Among the latter are a *Subspace*

[4] The $r_{ij}(\boldsymbol{x})$ can be interpreted as a conditional probability estimate for the membership of vector $\boldsymbol{x}$ in class i when separating its class only from class j, without considering any of the other classes, cf. [1].

[5] See `http://etl.go.jp/etlcdb`.

classifier in Hilbert Space, see [14], a SVM, and the proposed pairwise coupled NKDA method. In all kernel models, standard RBF kernels are used. Table 1 presents 5-fold cross validation estimates of misclassification rates.[6] In each of the 5 trials, the learning set consisted of 160 samples per character, the test set of 40 samples per character. Averaged misclassification rates and standard errors are presented.

Table 1. 5-fold cross validation estimates of misclassification rates [%], and standard errors (in brackets).

Classifier	**Error rate** [%]
5-NN	7.35
Subspace HS	3.64
SVM	2.87 (0.22)
Pairwise NKDA	2.25 (0.16)

From the above table we conclude that Pairwise NKDA significantly outperforms all other investigated techniques. Note, however, that Pairwise NKDA not only yields this excellent numerical accuracy level, but additionally provides us with probabilistic outputs. A closer look on the assignment probabilities for both the correctly classified and the misclassified samples gives considerable insight into the practical benefit of a probabilistic classification model. In figure 3 we have plotted histograms of estimated assignment probabilities, i.e. of the confidence level about the predicted class label.

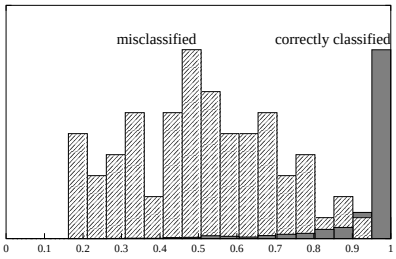

Figure 3. Histograms of estimated assignment probabilities, both for the misclassified samples and the correctly classified ones (rescaled for equal maximum value).

For the correctly classified samples, this empirical distribution is sharply peaked at probability values between 0.95 and 1.0, and rapidly decays with decreasing certainty of class assignments. This means that a correctly predicted label highly correlates with a a high confidence level. On the contrary, the histogram for the misclassified patterns is widely spread over a certainty range between 0.2 and 0.8. It is worth noticing that *only 4* misclassified patters attain a probability larger that 0.8. This observation suggests to introduce a *doubt class* that collects patters with uncertain labels. A full text recognition system could benefit from such a doubt class by assigning more weight to *semantic-based* prediction methods for elements of this class.

In a last experiment, we compared the *computational performance* of Pairwise NKDA and state-of-the-art SVM algorithms. We are aware that such comparisons are difficult in practice, since the performance of either algorithm heavily depends on certain tuning parameters. Nevertheless, we tried to make the experimental setup as fair as possible. All experiments are performed under the same hardware conditions, and the measured times are corrected for possible overheads for loading the samples into memory. The computation times for separating all 71 classes are summarized in table 2. We used two highly tuned SVM packages, SVM*light* V3.50 (see [15]) and *SVMTorch II* V1.07 (see [16]). As can bee seen, Pairwise NKDA is significantly faster than *SVMTorch*. The latter can be considered as one of the best SVM packages available. It outperforms interior-point algorithms, such as LOQO (see [12]), by several orders of magnitude: for the full sample size ($N = 11360$), a single Cholesky decomposition takes roughly 100 minutes. If we assume, that 10 interior-point iterations (and thus 10 such decompositions) are required for each of the 71 subproblems, this would accumulate to roughly $7 \cdot 10^4$ minutes (~ 50 days) for the whole learning procedure.

Table 2. Computation times for the 71-class Hiragana dataset.

Algorithm	**Computation time**
SVM*light* V3.50	65 min
SVMTorch II V1.07	23 min
Pairwise NKDA	8 min

For computing the pairwise classification rules, the two-class NKDA method was iteratively approximated by a conjugate gradient method, as explained in section 2.1. The iteration number was fixed to 20. Further iterations did not improve the prediction accuracy. Concerning the actual training times, the reader should also notice that we are comparing highly tuned SVM optimization packages with a straight-forward Pairwise NKDA implementation that basically uses standard routines from *Numerical Recipes*, [11]. We consider our implementation to yet possess ample opportunities for further optimization.

[6] The results for *5-NN* and *Subspace HS* are taken from [14]. The $k-NN$ result with $k = 5$ is the best one among $k = 1, 3, 5, 7$.

5. Conclusion

In this paper we have presented a new approach to multi-class classification with kernel methods. In particular, we have focused on a kernelized variant of classical linear discriminant analysis, which we call *NKDA*. Compared to the SVM, the main advantage of NKDA is its clear *probabilistic interpretation* within a Bayesian framework.

For multi-class problems, we can use the NKDA classifier as a building block in a *pairwise coupling* procedure. The main idea of pairwise coupling is to couple all pairwise decision rules into an estimate for the posterior probability of class membership. Besides of conceptual advantages over classical ways of handling multi-class problems, this technique additionally has a clear numerical advantage: for a fixed number of training patterns, the computational costs reduce linearly in the number of classes. Experiments for the large-scale problem of handwritten Hiragana recognition with 71 classes have effectively demonstrated that Pairwise NKDA attains a level of accuracy even superior to the SVM. Moreover, the user is provided not only with a predicted class label, but also with an explicite posterior probability. We thus not only have a single "hard" class assignment, but we are given a weighted list of possible labels.

Besides attaining an excellent level of prediction accuracy, Pairwise NKDA also imposes significantly lower computational costs as the SVM: even our straight-forward implementation outperformed one of the best SVM packages available. We thus conclude that Pairwise NKDA in general is a highly suited method for dealing with multi-class problems, in which it is advantageous to quantify the uncertainty about the predicted class labels.

Acknowledgments. We wish to thank M. Braun and J. Buhmann for fruitful discussions.

References

[1] J. Friedman, "Another approach to polychotomous classification," Tech. Rep., Stanford University, 1996.

[2] Trevor Hastie and Robert Tibshirani, "Classification by pairwise coupling," in *Advances in Neural Information Processing Systems*, Michael I. Jordan, Michael J. Kearns, and Sara A. Solla, Eds. 1998, vol. 10, The MIT Press.

[3] K.-R. Müller, S. Mika, G. Rätsch, K. Tsuda, and B. Schölkopf, "An introduction to kernel-based learning algorithms," *IEEE Transactions on Neural Networks*, vol. 12, no. 2, pp. 181–201, March 2001.

[4] V. Roth and V. Steinhage, "Nonlinear discriminant analysis using kernel functions," in *Advances in Neural Information Processing Systems*, S.A. Solla, T.K. Leen, and K.-R. Müller, Eds. 1999, vol. 12, pp. 568–574, MIT Press.

[5] T. Jaakkola, M. Meila, and T. Jebara, "Maximum entropy discrimination," in *Neural Information Processing Systems*, S. Solla, T. Leen, and K. Müller, Eds. 1999, vol. 12, pp. 470–476, MIT Press.

[6] P. Sollich, "Probabilistic methods for support vector machines," in *Advances in Neural Information Processing Systems*, S.A. Solla, T.K. Leen, and K.-R. Müller, Eds. 1999, vol. 12, pp. 349–355, MIT Press.

[7] L. Hermes, D. Frieauff, J. Puzicha, and J. Buhmann, "Support vector machines for land usage classification in Landsat TM imagery," in *Proc. of the IEEE 1999 International Geoscience and Remote Sensing Symposium*, 1999, vol. 1, pp. 348–350.

[8] S. Mika, G. Rätsch, J. Weston, B. Schölkopf, and K.-R. Müller, "Fisher discriminant analysis with kernels," in *Neural Networks for Signal Processing IX*, Y.-H. Hu, J. Larsen, E. Wilson, and S. Douglas, Eds. 1999, pp. 41–48, IEEE.

[9] R.O. Duda and P.E. Hart, *Pattern Classification and Scene Analysis*, Wiley, 1973.

[10] A.E. Hoerl and R.W. Kennard, "Ridge regression: Biased estimation for nonorthogonal problems," *Technometrics*, vol. 12, pp. 55–67, 1970.

[11] W.H. Press, S.A. Teukolsky, W.T Vetterling, and B.P. Flannery, *Numerical Recipes in C*, Cambridge University Press, 1992.

[12] R. J. Vanderbei, "LOQO: An interior point code for quadratic programming," *Optimization Methods and Software*, vol. 11, pp. 451–484, 1999.

[13] F. Kimura, K. Takashina, S. Tsuruoka, and Y. Miyake, "Modified quadratic discriminant functions and the application to chinese character recognition," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 9, no. 1, pp. 149–153, 1987.

[14] K. Tsuda, "Subspace classifier in the hilbert space," *Pattern Recognition Letters*, vol. 20, pp. 513–519, 1999.

[15] T. Joachims, "Making large-scale svm learning practical," in *Advances in Kernel Methods – Support Vector Learning*, B. Schölkopf, C. Burges, and A. Smola, Eds. MIT Press, 1999.

[16] Ronan Collobert and Samy Bengio, "Support vector machines for large-scale regression problems," Tech. Rep. IDIAP-RR-00-17, IDIAP, Martigny, Switzerland, 2000.