A New Discriminative Kernel from Probabilistic Models

Koji Tsuda
koji.tsuda@aist.go.jp
AIST Computational Biology Research Center, Koto-ku, Tokyo, 135-0064, Japan, and Fraunhofer FIRST, 12489 Berlin, Germany

Motoaki Kawanabe
nabe@first.fraunhofer.de
Fraunhofer FIRST, 12489 Berlin, Germany

Gunnar Rätsch
Gunnar.Raetsch@anu.edu.au
Australian National University, Research School for Information Sciences and Engineering, Canberra, ACT 0200, Australia, and Fraunhofer FIRST, 12489 Berlin, Germany

Sören Sonnenburg
sonne@first.fraunhofer.de
Fraunhofer FIRST, 12489 Berlin, Germany

Klaus-Robert Müller
klaus@first.fraunhofer.de
Fraunhofer FIRST, 12489 Berlin, Germany, and University of Potsdam, 14469 Potsdam, Germany

Recently, Jaakkola and Haussler (1999) proposed a method for constructing kernel functions from probabilistic models. Their so-called Fisher kernel has been combined with discriminative classifiers such as support vector machines and applied successfully in, for example, DNA and protein analysis. Whereas the Fisher kernel is calculated from the marginal log-likelihood, we propose the TOP kernel derived from tangent vectors of posterior log-odds. Furthermore, we develop a theoretical framework on feature extractors from probabilistic models and use it for analyzing the TOP kernel. In experiments, our new discriminative TOP kernel compares favorably to the Fisher kernel.

1 Introduction

In classification tasks, the purpose of learning is to predict the output $y \in \{-1, +1\}$ of some unknown system given the input $x \in \mathcal{X}$ based on the

Neural Computation **14, 2397–2414** (2002)

training samples $\{\boldsymbol{x}_i, y_i\}_{i=1}^n$. A feature extractor is a vector-valued function $\boldsymbol{f}\colon \mathcal{X} \rightarrow \mathbb{R}^D$ designed for converting the representation of data without losing the information necessary for discrimination.

When $\mathcal{X}$ is a vector space like $\mathbb{R}^d$, many feature extraction methods have been proposed (Fukunaga, 1990). However, they are typically not applicable when $\mathcal{X}$ is a set of sequences of symbols and does not have the structure of a vector space as in DNA or protein analysis (Durbin, Eddy, Krogh, & Mitchison, 1998). In such cases, the similarity (or proximity) between two samples plays an important role (Cox & Ferry, 1993; Graepel, Herbrich, Bollmann-Sdorra, & Obermayer, 1999; Hofmann & Buhmann, 1997). The simplest method is to prepare several prototype samples and compose a feature vector from the similarities to these samples (Graepel et al., 1999). Alternatively in multidimensional scaling (MDS; Cox & Ferry, 1993), the samples are mapped such that the given dissimilarity is approximated by the Euclidean distance in feature space. However, similarities are often not available, and to define a "good" similarity measure in terms of the classification task in feature space is therefore difficult and requires a fair amount of prior knowledge.

Recently, the Fisher kernel (FK; Jaakkola & Haussler, 1999) was proposed, which allows measuring distances between symbols by computing features from probabilistic models $p(\boldsymbol{x} \mid \boldsymbol{\theta})$. At first, a parameter estimate $\hat{\boldsymbol{\theta}}$ is obtained from the training examples. Then the tangent vector of the log marginal likelihood $\log p(\boldsymbol{x} \mid \hat{\boldsymbol{\theta}})$ is used as a feature vector. *Fisher kernel* refers to the inner product in this feature space, but the method is effectively a feature extractor (also since the features are computed explicitly). The FK can be combined with discriminative classifiers such as support vector machines (SVMs), and it has achieved excellent classification results in several fields, for example in DNA and protein analysis (Jaakkola & Haussler, 1999; Jaakkola, Diekhans, & Haussler, 2000). Empirically, it is reported that the FK-SVM system often outperforms the classification performance of a plug-in estimate, that is, the pure probabilistic approach.[1] Note that the FK is only one possible member in the family of feature extractors $\boldsymbol{f}_{\hat{\boldsymbol{\theta}}}(\boldsymbol{x})\colon \mathcal{X} \rightarrow \mathbb{R}^D$ that can be derived from a probabilistic model. We call this family *model-dependent feature extractors* because different probabilistic models lead to different feature vectors. Exploring this family is a very important and interesting subject.

Since model-dependent feature extractors are newly developed, performance measures for them have not yet been established. In this article, we therefore propose two performance measures. Then we define a new kernel (or, equivalently, a feature extractor) derived from the tangent vector of posterior log-odds, which we denote as the TOP kernel. We will analyze

[1] In classification by plug-in estimate, $\boldsymbol{x}$ is classified by thresholding the posterior probability $\hat{y} = \mathrm{sign}(P(y = +1 \mid \boldsymbol{x}, \hat{\boldsymbol{\theta}}) - \frac{1}{2})$ (Devroye, Györfi, & Lugosi, 1996).

the performance of the TOP kernel in terms of our performance measures. Finally, the TOP kernel is compared—favorably—to the FK in experiments with artificial data and protein data.

2 Performance Measures

To begin, let us describe the notations. Let $\boldsymbol{x} \in \mathcal{X}$ be the input point and $y \in \{-1, +1\}$ be the class label. $\mathcal{X}$ may be a finite set or an infinite set like $\mathbb{R}^d$. Let us assume that we know the parametric model of the joint probability $p(\boldsymbol{x}, y \mid \boldsymbol{\theta})$, where $\boldsymbol{\theta} \in \mathbb{R}^p$ is the parameter vector. Assume that the model $p(\boldsymbol{x}, y \mid \boldsymbol{\theta})$ is regular (Murata, Yoshizawa, & Amari, 1994) and contains the true distribution. Then the true parameter $\boldsymbol{\theta}^*$ is uniquely determined. Let $\hat{\boldsymbol{\theta}}$ be a consistent estimator (Devroye et al., 1996) of $\boldsymbol{\theta}^*$, which is obtained by n training examples drawn independent and identically distributed (i.i.d.) from $p(\boldsymbol{x}, y \mid \boldsymbol{\theta}^*)$. Let $\partial_{\theta_i} f = \partial f / \partial \theta_i$, $\nabla_{\boldsymbol{\theta}} f = (\partial_{\theta_1} f, \ldots, \partial_{\theta_p} f)^\top$, and $\nabla^2_{\boldsymbol{\theta}} f$ denote the $p \times p$ matrix, the Hessian, whose (i, j)th element is $\partial^2 f / (\partial \theta_i \partial \theta_j)$.

As the FK is commonly used in combination with linear classifiers such as SVMs, one reasonable performance measure is the classification error of a linear classifier $\boldsymbol{w}^T \boldsymbol{f}_{\hat{\boldsymbol{\theta}}}(\boldsymbol{x}) + b$ in the feature space $\mathbb{R}^D$, where $\boldsymbol{w} \in \mathbb{R}^D$ and $b \in \mathbb{R}$. Usually $\boldsymbol{w}$ and b are determined by a learning algorithm, so the optimal feature extractor is different with regard to each learning algorithm. To cancel out this ambiguity and make a theoretical analysis possible, we assume the optimal learning algorithm is used. When $\boldsymbol{w}$ and b are optimally chosen, the classification error is

$$R(\boldsymbol{f}_{\hat{\boldsymbol{\theta}}}) = \min_{\boldsymbol{w} \in \mathcal{S}, b \in \mathbb{R}} E_{\boldsymbol{x}, y} \Phi[-y(\boldsymbol{w}^\top \boldsymbol{f}_{\hat{\boldsymbol{\theta}}}(\boldsymbol{x}) + b)], \tag{2.1}$$

where $\mathcal{S} = \{\boldsymbol{w} \mid \|\boldsymbol{w}\| = 1, \boldsymbol{w} \in \mathbb{R}^D\}$, $\Phi[a]$ is the step function, which is 1 if $a > 0$ and 0 otherwise, and $E_{\boldsymbol{x}, y}$ denotes the expectation with respect to the true distribution $p(\boldsymbol{x}, y \mid \boldsymbol{\theta}^*)$. $R(\boldsymbol{f}_{\hat{\boldsymbol{\theta}}})$ is at least as large as the Bayes error L^* (Fukunaga, 1990) and $R(\boldsymbol{f}_{\hat{\boldsymbol{\theta}}}) = L^*$ only if the linear classifier implements the same decision rule as the Bayes optimal rule. From a geometrical point of view, $R(\boldsymbol{f}_{\hat{\boldsymbol{\theta}}}) - L^*$ describes how linear the optimal boundary is in the feature space.

Now that we have a performance measure, it is natural to design a feature extractor that minimizes $R(\boldsymbol{f}_{\hat{\boldsymbol{\theta}}})$. This task is difficult because of the nondifferentiable function Φ. So we construct another measure, which upper-bounds $R(\boldsymbol{f}_{\hat{\boldsymbol{\theta}}})$: we consider the estimation error of the posterior probability by a logistic regressor $F(\boldsymbol{w}^\top \boldsymbol{f}_{\hat{\boldsymbol{\theta}}}(\boldsymbol{x}) + b)$, with $F(t) = 1/(1 + \exp(-t))$:

$$D(\boldsymbol{f}_{\hat{\boldsymbol{\theta}}}) = \min_{\boldsymbol{w} \in \mathbb{R}^D, b \in \mathbb{R}} E_{\boldsymbol{x}} |F(\boldsymbol{w}^\top \boldsymbol{f}_{\hat{\boldsymbol{\theta}}}(\boldsymbol{x}) + b) - P(y = +1 \mid \boldsymbol{x}, \boldsymbol{\theta}^*)|. \tag{2.2}$$

The relationship between $D(\boldsymbol{f}_{\hat{\boldsymbol{\theta}}})$ and $R(\boldsymbol{f}_{\hat{\boldsymbol{\theta}}})$ is illustrated as follows: Let $\hat{L}$ be the classification error rate of an arbitrary posterior probability estimator

$\hat{P}(y = +1 \mid \boldsymbol{x})$. The following inequality is known (Devroye et al., 1996):

$$\hat{L} - L^* \leq 2E_x|\hat{P}(y = +1 \mid \boldsymbol{x}) - P(y = +1 \mid \boldsymbol{x}, \boldsymbol{\theta}^*)|. \tag{2.3}$$

When we use $\hat{P}(y = +1 \mid \boldsymbol{x}) := F(\boldsymbol{w}^\top \boldsymbol{f}_{\hat{\boldsymbol{\theta}}}(\boldsymbol{x}) + b)$, this inequality leads to the following relationship between the two measures:

$$R(\boldsymbol{f}_{\hat{\boldsymbol{\theta}}}) - L^* \leq 2D(\boldsymbol{f}_{\hat{\boldsymbol{\theta}}}). \tag{2.4}$$

By this bound, it is useful to derive a new feature extractor that minimizes $D(\boldsymbol{f}_{\hat{\boldsymbol{\theta}}})$, as will be done in section 4.

3 The Fisher Kernel

The Fisher kernel (FK) is defined[2] as $K(\boldsymbol{x}, \boldsymbol{x}') = \boldsymbol{s}(\boldsymbol{x}, \hat{\boldsymbol{\theta}})^\top Z^{-1}(\hat{\boldsymbol{\theta}}) \boldsymbol{s}(\boldsymbol{x}', \hat{\boldsymbol{\theta}})$, where $\boldsymbol{s}$ is the Fisher score,

$$\boldsymbol{s}(\boldsymbol{x}, \hat{\boldsymbol{\theta}}) = (\partial_{\theta_1} \log p(\boldsymbol{x} \mid \hat{\boldsymbol{\theta}}), \ldots, \partial_{\theta_p} \log p(\boldsymbol{x} \mid \hat{\boldsymbol{\theta}}))^\top = \nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{x}, \hat{\boldsymbol{\theta}}),$$

and Z is the Fisher information matrix: $Z(\boldsymbol{\theta}) = E_{\boldsymbol{x}}[\boldsymbol{s}(\boldsymbol{x}, \boldsymbol{\theta})\boldsymbol{s}(\boldsymbol{x}, \boldsymbol{\theta})^\top \mid \boldsymbol{\theta}]$. The theoretical foundation of FK is described in the following theorem (Jaakkola & Haussler, 1999): "A kernel classifier employed the Fisher kernel derived from a model that contains the label as a latent variable is, asymptotically, at least as good a classifier as the MAP labeling based on the model" (p. 491). The theorem says that the FK can perform at least as well as the plug-in estimate, if the parameters of linear classifier are properly determined (cf. appendix A of Jaakkola & Haussler, 1999). With our performance measure, this theorem can be represented more concisely: $R(\boldsymbol{f}_{\hat{\boldsymbol{\theta}}})$ is bounded by the classification error of the plug-in estimate $R_\pi(\hat{\boldsymbol{\theta}})$:

$$R(\boldsymbol{f}_{\hat{\boldsymbol{\theta}}}) \leq R_\pi(\hat{\boldsymbol{\theta}}) = E_{x,y}\Phi[-y(P(y = +1 \mid \boldsymbol{x}, \hat{\boldsymbol{\theta}}) - \tfrac{1}{2})]. \tag{3.1}$$

Note that the classification rule constructed by the plug-in estimate $P(y = +1 \mid \boldsymbol{x}, \hat{\boldsymbol{\theta}})$ can also be realized by a linear classifier in feature space. Property 3.1 is important since it guarantees that the FK performs better when the optimal $\boldsymbol{w}$ and b are available. In the next section, we present a new kernel that also satisfies property 3.1 and has a more appealing theoretical property as well.

[2] In practice, some variants of the FK are used. For example, if the derivative of each class distribution, not marginal, is taken, the feature vector of FK is quite similar to that of our kernel. However, these variants should be deliberately discriminated from the FK in theoretical discussions. Throughout this article, including experiments, we adopt the original definition of the Fisher kernel from Jaakkola and Haussler (1999).

4 The TOP Kernel

4.1 Definition. Now we proceed to propose a new kernel. Our aim is to obtain a feature extractor that achieves small $D(\boldsymbol{f}_{\hat{\boldsymbol{\theta}}})$. When a feature extractor $\boldsymbol{f}_{\hat{\boldsymbol{\theta}}}(\boldsymbol{x})$ satisfies[3]

$$\boldsymbol{w}^\top \boldsymbol{f}_{\hat{\boldsymbol{\theta}}}(\boldsymbol{x}) + b = F^{-1}(P(y = +1 \mid \boldsymbol{x}, \boldsymbol{\theta}^*)) \;\; \text{for all } \boldsymbol{x} \in \mathcal{X}, \tag{4.1}$$

with certain values of $\boldsymbol{w}$ and b, we have $D(\boldsymbol{f}_{\hat{\boldsymbol{\theta}}}) = 0$. However, since the true parameter $\boldsymbol{\theta}^*$ is unknown, all we can do is to construct $\boldsymbol{f}_{\hat{\boldsymbol{\theta}}}$, which approximately satisfies equation 4.1. Let us define[4]

$$\begin{aligned} v(\boldsymbol{x}, \boldsymbol{\theta}) &= F^{-1}(P(y = +1 \mid \boldsymbol{x}, \boldsymbol{\theta})) \\ &= \log(P(y = +1 \mid \boldsymbol{x}, \boldsymbol{\theta})) - \log(P(y = -1 \mid \boldsymbol{x}, \boldsymbol{\theta})), \end{aligned} \tag{4.2}$$

which is called the posterior log-odds of a probabilistic model (Devroye et al., 1996). By Taylor expansion around the estimate $\hat{\boldsymbol{\theta}}$ up to the first order, we can approximate $v(\boldsymbol{x}, \boldsymbol{\theta}^*)$ as

$$v(\boldsymbol{x}, \boldsymbol{\theta}^*) \approx v(\boldsymbol{x}, \hat{\boldsymbol{\theta}}) + \sum_{i=1}^{p} \partial_{\theta_i} v(\boldsymbol{x}, \hat{\boldsymbol{\theta}})(\theta_i^* - \hat{\theta}_i). \tag{4.3}$$

Thus, by setting

$$\boldsymbol{f}_{\hat{\boldsymbol{\theta}}}(\boldsymbol{x}) := (v(\boldsymbol{x}, \hat{\boldsymbol{\theta}}), \partial_{\theta_1} v(\boldsymbol{x}, \hat{\boldsymbol{\theta}}), \ldots, \partial_{\theta_p} v(\boldsymbol{x}, \hat{\boldsymbol{\theta}}))^\top \tag{4.4}$$

and

$$\boldsymbol{w} := \boldsymbol{w}^* = (1, \theta_1^* - \hat{\theta}_1, \ldots, \theta_p^* - \hat{\theta}_p)^\top, \;\; b = 0, \tag{4.5}$$

equation 4.1 is approximately satisfied. Since a tangent vector of the posterior log-odds constitutes the main part of the feature vector, we call the inner product of the two feature vectors the TOP kernel:

$$K(\boldsymbol{x}, \boldsymbol{x}') = \boldsymbol{f}_{\hat{\boldsymbol{\theta}}}(\boldsymbol{x})^\top \boldsymbol{f}_{\hat{\boldsymbol{\theta}}}(\boldsymbol{x}'). \tag{4.6}$$

It is easy to verify that the TOP kernel satisfies equation 3.1, because we can construct the same decision rule as the plug-in estimate by using the first element only ($\boldsymbol{w} = (1, 0, \ldots, 0)$, $b = 0$).

[3] Notice that $F^{-1}(t) = \log t - \log(1 - t)$.

[4] One can easily derive TOP kernels from higher-order Taylor expansions. However, we will deal only with the first-order expansion here, because higher-order expansions would induce extremely high-dimensional feature vectors in practical cases.

4.2 A Theoretical Analysis. In this section, we compare the TOP kernel with the plug-in estimate in terms of performance measures. Later, we assume that $0 < P(y = +1 \mid \boldsymbol{x}, \boldsymbol{\theta}) < 1$ to prevent $|v(\boldsymbol{x}, \boldsymbol{\theta})|$ from going to infinity. Also, it is assumed that $\nabla_{\boldsymbol{\theta}} P(y = +1 \mid \boldsymbol{x}, \boldsymbol{\theta})$ and $\nabla^2_{\boldsymbol{\theta}} P(y = +1 \mid \boldsymbol{x}, \boldsymbol{\theta})$ are bounded. Substituting the plug-in estimate (denoted by the subscript π) into $D(\boldsymbol{f}_{\hat{\boldsymbol{\theta}}})$, we have

$$D_\pi(\hat{\boldsymbol{\theta}}) = E_{\boldsymbol{x}} |P(y = +1 \mid \boldsymbol{x}, \hat{\boldsymbol{\theta}}) - P(y = +1 \mid \boldsymbol{x}, \boldsymbol{\theta}^*)|.$$

Define $\Delta\boldsymbol{\theta} = \hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^*$. By Taylor expansion around $\boldsymbol{\theta}^*$, we have

$$\begin{aligned} D_\pi(\hat{\boldsymbol{\theta}}) &= E_{\boldsymbol{x}} |(\Delta\boldsymbol{\theta})^\top \nabla_{\boldsymbol{\theta}} P(y = +1 \mid \boldsymbol{x}, \boldsymbol{\theta}^*) \\ &\qquad + \frac{1}{2} (\Delta\boldsymbol{\theta})^\top \nabla^2_{\boldsymbol{\theta}} P(y = +1 \mid \boldsymbol{x}, \boldsymbol{\theta}_0)(\Delta\boldsymbol{\theta})| \\ &= O(\|\Delta\boldsymbol{\theta}\|), \end{aligned} \tag{4.7}$$

where $\boldsymbol{\theta}_0 = \boldsymbol{\theta}^* + \gamma\Delta\boldsymbol{\theta}\ \ (0 \le \gamma \le 1)$. When the TOP kernel is used,

$$D(\boldsymbol{f}_{\hat{\boldsymbol{\theta}}}) \le E_{\boldsymbol{x}} |F((\boldsymbol{w}^*)^\top \boldsymbol{f}_{\hat{\boldsymbol{\theta}}}(\boldsymbol{x})) - P(\boldsymbol{y} = +1 \mid \boldsymbol{x}, \boldsymbol{\theta}^*)|, \tag{4.8}$$

where $\boldsymbol{w}^*$ is defined as in equation 4.5. Since F is Lipschitz continuous, there is a finite positive constant M such that $|F(a) - F(b)| \le M|a - b|$. Thus,

$$D(\boldsymbol{f}_{\hat{\boldsymbol{\theta}}}) \le M E_{\boldsymbol{x}} |(\boldsymbol{w}^*)^\top \boldsymbol{f}_{\hat{\boldsymbol{\theta}}}(\boldsymbol{x}) - F^{-1}(P(\boldsymbol{y} = +1 \mid \boldsymbol{x}, \boldsymbol{\theta}^*))|. \tag{4.9}$$

Since $(\boldsymbol{w}^*)^\top \boldsymbol{f}_{\hat{\boldsymbol{\theta}}}(\boldsymbol{x})$ is the Taylor expansion of $F^{-1}(P(\boldsymbol{y} = +1 \mid \boldsymbol{x}, \boldsymbol{\theta}^*))$ up to the first order, equation 4.3, the first-order terms of $\Delta\boldsymbol{\theta}$ are excluded from the right side of equation 4.9; thus, $D(\boldsymbol{f}_{\hat{\boldsymbol{\theta}}}) = O(\|\Delta\boldsymbol{\theta}\|^2)$. Since both the plug-in and the TOP kernel depend on the parameter estimate $\hat{\boldsymbol{\theta}}$, the errors $D_\pi(\hat{\boldsymbol{\theta}})$ and $D(\boldsymbol{f}_{\hat{\boldsymbol{\theta}}})$ become smaller as $\|\Delta\boldsymbol{\theta}\|$ decreases. However,the rate of convergence of the TOP kernel is much faster than that of the plug-in estimate if $\boldsymbol{w}$ and b are optimally chosen.

This result is closely related to large sample performances. Assuming that $\hat{\boldsymbol{\theta}}$ is a $n^{1/2}$ consistent estimator with asymptotic normality (e.g., the maximum likelihood estimator), we have $\|\Delta\boldsymbol{\theta}\| = O_p(n^{-1/2})$ (Murata et al., 1994), where O_p denotes stochastic order (Barndorff-Nielsen & Cox, 1989). We can directly derive the convergence order as $D_\pi(\hat{\boldsymbol{\theta}}) = O_p(n^{-1/2})$ and $D(\boldsymbol{f}_{\hat{\boldsymbol{\theta}}}) = O_p(n^{-1})$. By using the relation 2.4, it follows that $R_\pi(\hat{\boldsymbol{\theta}}) - L^* = O_p(n^{-1/2})$ and $R(\boldsymbol{f}_{\hat{\boldsymbol{\theta}}}) - L^* = O_p(n^{-1})$.[5] Therefore, the TOP kernel has a much better convergence rate in $R(\boldsymbol{f}_{\hat{\boldsymbol{\theta}}})$. However, we must note that this fast rate is

[5] For detailed discussion about the convergence orders of classification error, see Devroye et al. (1996).

possible only when the optimal linear classifier is combined with the TOP kernel. Since nonoptimal linear classifiers typically have the rate $O_p(n^{-1/2})$ (Devroye et al., 1996), the overall rate is dominated by the slower rate and turns out to be $O_p(n^{-1/2})$. But this theoretical analysis is still meaningful, because it shows the existence of a very efficient linear boundary in the TOP feature space. This result encourages practical efforts to improve linear boundaries by engineering loss functions and regularization terms with cross validation, bootstrapping, or other model selection criteria (Devroye et al., 1996).

4.3 Exponential Family: A Special Case. When the distributions of the two classes belong to the exponential family, the TOP kernel can achieve an even better result than shown above. Distributions of the exponential family can be written as $q(\boldsymbol{x}, \boldsymbol{\eta}) = \exp(\boldsymbol{\eta}^\top \boldsymbol{t}(\boldsymbol{x}) + \psi(\boldsymbol{\eta}))$, where $\boldsymbol{t}(\boldsymbol{x})$ is a vector-valued function called sufficient statistics and $\psi(\boldsymbol{\eta})$ is a normalization factor (Geiger & Meek, 1998). Let α denote the parameter for class prior probability of the positive model $P(y = +1)$. Then the probabilistic model is described as

$$\begin{aligned} p(\boldsymbol{x}, y = +1 \mid \boldsymbol{\theta}) &= \alpha q_{+1}(\boldsymbol{x}, \boldsymbol{\eta}_{+1}), \\ p(\boldsymbol{x}, y = -1 \mid \boldsymbol{\theta}) &= (1-\alpha) q_{-1}(\boldsymbol{x}, \boldsymbol{\eta}_{-1}), \end{aligned}$$

where $\boldsymbol{\theta} = \{\alpha, \boldsymbol{\eta}_{+1}, \boldsymbol{\eta}_{-1}\}$. The posterior log-odds reads

$$\begin{aligned} v(\boldsymbol{x}, \boldsymbol{\theta}) = {} & \boldsymbol{\eta}_{+1}^\top \boldsymbol{t}_{+1}(\boldsymbol{x}) + \psi_{+1}(\boldsymbol{\eta}_{+1}) - \boldsymbol{\eta}_{-1}^\top \boldsymbol{t}_{-1}(\boldsymbol{x}) \\ & - \psi_{-1}(\boldsymbol{\eta}_{-1}) + \log \frac{\alpha}{1-\alpha}. \end{aligned} \tag{4.10}$$

The TOP feature vector is described as

$$f_{\hat{\boldsymbol{\theta}}}(\boldsymbol{x}) = (v(\boldsymbol{x}, \hat{\boldsymbol{\theta}}), \partial_\alpha v(\boldsymbol{x}, \hat{\boldsymbol{\theta}}), \nabla_{\eta_{+1}} v(\boldsymbol{x}, \hat{\boldsymbol{\theta}})^\top, \nabla_{\eta_{-1}} v(\boldsymbol{x}, \hat{\boldsymbol{\theta}})^\top)^\top,$$

where $\nabla_{\eta_s} v(\boldsymbol{x}, \hat{\boldsymbol{\theta}}) = s(\boldsymbol{t}_s(\boldsymbol{x}) + \nabla_{\eta_s} \psi_s(\hat{\boldsymbol{\eta}}_s))$ for $s = \{+1, -1\}$. So, when

$$\boldsymbol{w} = (1, 0, {\boldsymbol{\eta}_{+1}^*}^\top - \hat{\boldsymbol{\eta}}_{+1}^\top, {\boldsymbol{\eta}_{-1}^*}^\top - \hat{\boldsymbol{\eta}}_{-1}^\top)^\top$$

and b is properly set, the true log-odds $F^{-1}(P(y = +1 \mid \boldsymbol{x}, \boldsymbol{\theta}^*))$ can be constructed as a linear function in the feature space 4.1. Thus, $D(\boldsymbol{f}_{\hat{\boldsymbol{\theta}}}) = 0$ and $R(\boldsymbol{f}_{\hat{\boldsymbol{\theta}}}) = L^*$. Furthermore, since each feature is represented as a linear function of sufficient statistics $\boldsymbol{t}_{+1}(\boldsymbol{x})$ and $\boldsymbol{t}_{-1}(\boldsymbol{x})$, one can construct an equivalent feature space as $(\boldsymbol{t}_{+1}(\boldsymbol{x})^\top, \boldsymbol{t}_{-1}(\boldsymbol{x})^\top)^\top$ without knowing $\hat{\boldsymbol{\theta}}$.[6] This result has some importance because all graphical models without hidden states can be represented as members of the exponential family, for example, Markov models (Geiger & Meek, 1998).

[6] It is well known that the Bayes optimal boundary of exponential family distributions forms a hyperplane in the space of sufficient statistics (Devroye et al., 1996).

5 Experiment with Artificial Data

In this section, we present a classification experiment with artificial data. Here, the probabilistic model of each class is a mixture of two gaussians,

$$q(\boldsymbol{x}, \boldsymbol{\xi}) = \beta g(\boldsymbol{x}, \boldsymbol{\eta}_1) + (1 - \beta) g(\boldsymbol{x}, \boldsymbol{\eta}_2), \tag{5.1}$$

where $\boldsymbol{\xi} = [\beta, \boldsymbol{\eta}_1, \boldsymbol{\eta}_2]$ and g is the natural parameter representation of a gaussian distribution

$$g(\boldsymbol{x}, \boldsymbol{\eta}) = \exp\left\{ \eta_1 \|\boldsymbol{x}\|^2 + \sum_{i=1}^{d} x_i \eta_{i+1} \right. \\ \left. + \sum_{i=1}^{d} \eta_{i+1}^2 / (4\eta_1) - \frac{d}{2} \log(-\pi/\eta_1) \right\}. \tag{5.2}$$

Notice that the natural parameter $\boldsymbol{\eta}$ corresponds to the conventional parameters (mean $\boldsymbol{\mu}$ and standard deviation σ) as $\eta_1 = -1/(2\sigma^2)$, $\eta_i = \mu_{i-1}/\sigma^2$ $(i \geq 2)$. The true parameters of the two classes are defined as

$$\boldsymbol{\mu}_1 = \boldsymbol{\mu}_2 = (0, \ldots, 0), \sigma_1 = 1, \sigma_2 = \tfrac{1}{2}, \beta = \tfrac{1}{2} \qquad \text{(first class)}$$

$$\boldsymbol{\mu}_1 = \boldsymbol{\mu}_2 = (\tfrac{1}{10}, \ldots, \tfrac{1}{10}), \sigma_1 = \tfrac{4}{5}, \sigma_2 = \tfrac{2}{5}, \beta = \tfrac{1}{2} \quad \text{(second class)}.$$

Also, the true class prior probability is defined as $\alpha = \frac{1}{2}$. The derivatives in TOP and FKs are calculated with respect to the parameters $\boldsymbol{\xi}_{+1}, \boldsymbol{\xi}_{-1}$ in both classes and the class prior probability α as well.

In this experiment, the dimensionality of the input space is set to 100. The number of training samples is 30 in the first and 240 in the second experiment. The performance is measured by the error rate on a test set with 1000 samples. We compared the TOP kernel with the FK. As subsequent classifier, an SVM is chosen, which has a regularization parameter C. As candidate values of C, 10 equally spaced points in the log scale are taken from $[10^{-6}, 10^{-1}]$. The value of the parameter C is chosen from the candidate values according to the error rate on a validation set (100 samples).[7] The parameters $\boldsymbol{\xi}_{+1}$ and $\boldsymbol{\xi}_{-1}$ are estimated by the expectation-maximization algorithm (Bishop, 1995), and α is estimated by the ratio of training samples of two classes.

The test errors over 30 different samplings of training, validation, and test sets are shown in Figure 1. For reference, we also show the test errors of the Bayes optimal classifier and the test errors of the plug-in estimate. To illustrate the difference of FK and TOP in a detailed way, Figure 2 shows

[7] See Rätsch, Onoda, and Müller (2001) for details on how model selection is conducted in this type of experiment.

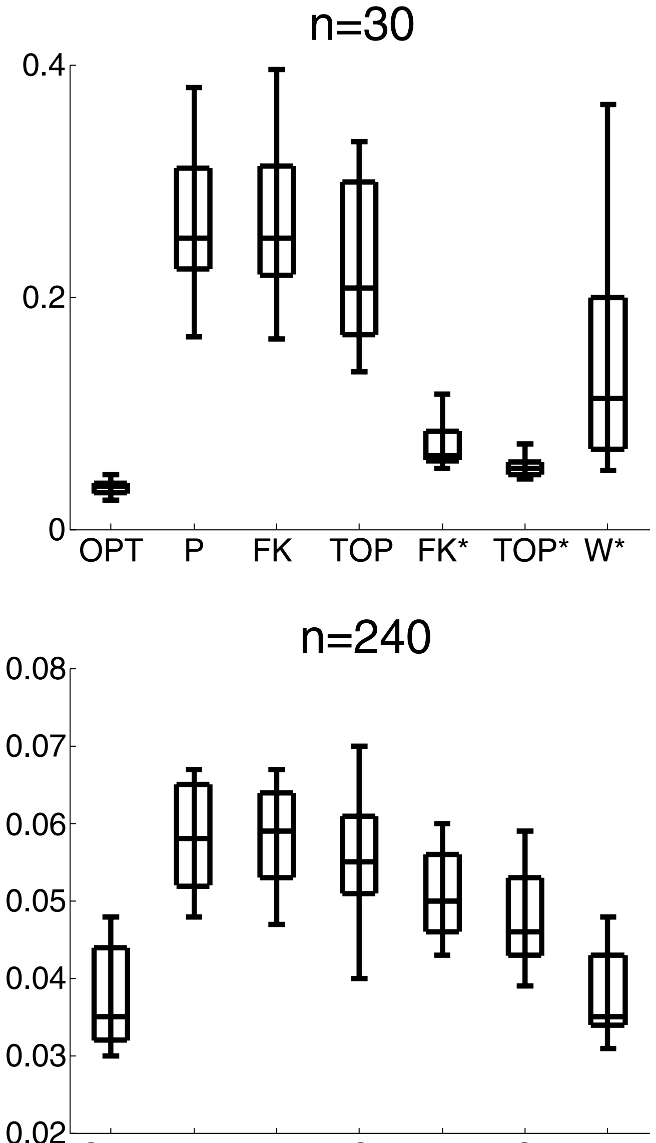

Figure 1: The error rates of various classifiers in the artificial data experiment. The top and bottom figures correspond to the results when $n = 30$ and $n = 240$, respectively. OPT: test errors of the Bayes optimal classifier; P: probabilistic models only; FK: the Fisher kernel with SVM; TOP: the TOP kernel with SVM; FK*: the Fisher kernel with the nearly optimal linear boundary constructed by a sufficient number of additional samples; TOP*: the TOP kernel with the nearly optimal linear boundary; W*: the TOP kernel with the weight vector $\boldsymbol{w}^*$.

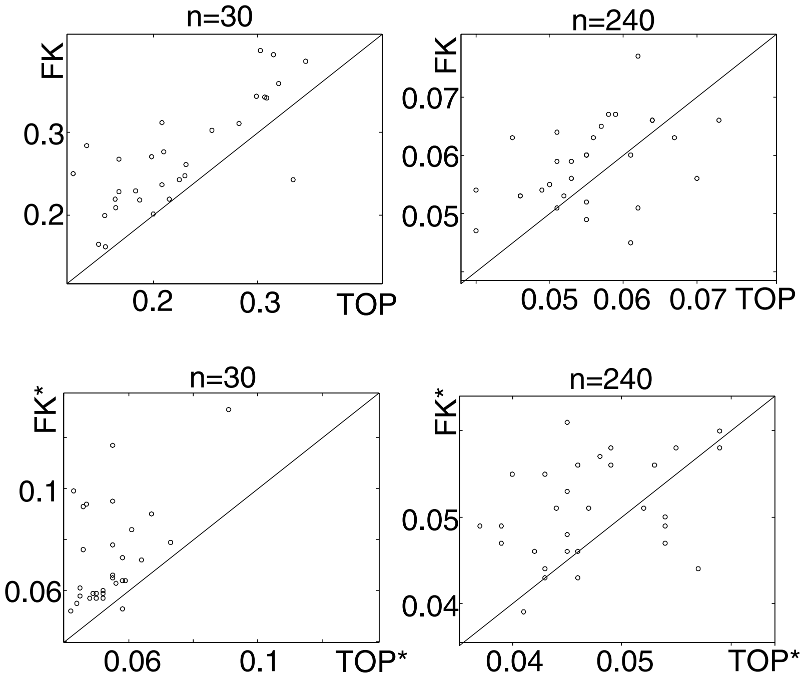

Figure 2: Comparison of error rates of SVMs with the Fisher kernel and the TOP kernel in the artificial data experiment. Every point corresponds to one of the 30 different training, validation, and test sets. The upper two figures correspond to FK and TOP in Figure 1, and the lower two correspond to FK* and TOP*.

comparative plots of test errors. Clearly, the TOP kernel has the smaller error rates in many cases. In order to investigate whether the differences in error rates are significant, two kinds of statistical tests are applied (see Table 1). One is the t-test (T), which compares the averages of error rates under the assumption that both distributions are gaussian. The other is the Wilcoxson signed rank test (WX), which uses the rank of differences of error rates. This is a nonparametric test, so it can be applied to any distribution. Because the distributions of error rates seem to be skewed (see Figure 1), we favor this test more than the t-test. When $n = 30$, both tests judged that the difference between average error rates of TOP and FK is significant, whereas when $n = 240$, only the Wilcoxson signed rank test judged that the difference is significant. So it is observed that the TOP kernel is especially effective in small sample cases.

So far, we have shown the performance of the combination of a feature extractor (FK or TOP) and SVM. In order to focus on feature extraction performance itself, it is desirable to observe the minimum achievable error by

Table 1: P Values of Statistical Tests in the Artificial Data Experiment.

Methods	Test	$n = 30$	$n = 240$
TOP, FK	T	0.0090**	0.12
	WX	$1.97 \times 10^{-5**}$	0.031*
TOP*, FK*	T	$2.2 \times 10^{-5**}$	0.024*
	WX	$2.1 \times 10^{-6**}$	0.0082**

Notes: The t-test is denoted as T and the Wilcoxson signed ranks test as WX.
$^*p < 0.05$. $^{**}p < 0.01$.

the optimal linear boundary. However, since it is difficult to derive the optimal boundaries analytically, a nearly optimal one is constructed by means of 3000 additional training samples, which are used not in constructing features but in determining linear boundaries. As the learning algorithm in feature space, we used the linear discriminant analysis (Fukunaga, 1990) without regularization. In Figures 1 and 2, these nearly optimal results are shown as FK* and TOP*. For reference, we also show the results of TOP when the Taylor coefficient $\boldsymbol{w}^*$ (see equation 4.5) is used as the weight vector (W^* in Figure 1).[8]

As seen in the test result (see Table 1), the difference between average error rates of TOP and FK is significant in both cases ($n = 30$ and 240). Thus, we are led to the conclusion that the TOP kernel extracts better discriminative features—at least in this experiment.

6 Experiments on Protein Data

In order to illustrate that the TOP kernel also works well for real-world problems, we present results on protein classification. The protein sequence data is obtained from the Superfamily web site, which provides sequence files with different degrees of redundancy filtering.[9] We used the one with 10% redundancy filtering. Here, 4541 sequences are hierarchically labeled into 7 classes, 558 folds, 845 superfamilies, and 1343 families according to the SCOP(1.53) scheme. In our experiment, we determine the top category classes as the learning target. The numbers of sequences in the classes are listed as 791, 1277, 1015, 915, 84, 76, and 383. We use only the first four classes, and six two-class problems are generated from all pairs among the

[8] When $n = 30$, the error rates of W^* are often larger than TOP* and have very large variance, because the Taylor expansion, equation 4.3, has large higher-order terms when the number of samples is small.

[9] Available on-line at: http://stash.mrc-lmb.cam.ac.uk/SUPERFAMILY/.

four classes.[10] The fifth and sixth classes are not used because the number of examples is too small. Also, the seventh class is omitted because it is quite different from the others and too easy to classify. In each two-class problem, the examples are randomly divided into 25% training set, 25% validation set, and 50% test set. The validation set is used for model selection.

As a probabilistic model for protein sequences, we train hidden Markov models (HMMs; Durbin et al., 1998) with fully connected states[11] by the Baum-Welch algorithm. [12] To construct FK and TOP kernels, the derivatives with respect to all parameters of the HMMs from both classes are included. The derivative with respect to the class prior probability is included as well. Let $q(\boldsymbol{x}, \boldsymbol{\xi})$ be the probability density function of an HMM. Then the marginal distribution is written as $p(\boldsymbol{x} \mid \hat{\boldsymbol{\theta}}) = \hat{\alpha} q(\boldsymbol{x}, \hat{\boldsymbol{\xi}}_{+1}) + (1 - \hat{\alpha}) q(\boldsymbol{x}, \hat{\boldsymbol{\xi}}_{-1})$, where α is a parameter corresponding to the class prior. The feature vector of FK consists of the following:

$$\nabla_{\boldsymbol{\xi}_s} \log p(\boldsymbol{x} \mid \hat{\boldsymbol{\theta}}) = P(y = s \mid \boldsymbol{x}, \hat{\boldsymbol{\theta}}) \nabla_{\boldsymbol{\xi}_s} \log q(\boldsymbol{x}, \hat{\boldsymbol{\xi}}_s) \qquad s \in \{-1, +1\} \tag{6.1}$$

$$\partial_\alpha \log p(\boldsymbol{x} \mid \hat{\boldsymbol{\theta}}) = \frac{1}{\hat{\alpha}} P(y = +1 \mid \boldsymbol{x}, \hat{\boldsymbol{\theta}}) - \frac{1}{1 - \hat{\alpha}} P(y = -1 \mid \boldsymbol{x}, \hat{\boldsymbol{\theta}}), \tag{6.2}$$

while the feature vector of TOP includes $\nabla_{\boldsymbol{\xi}_s} v(\boldsymbol{x}, \hat{\boldsymbol{\theta}}) = s \nabla_{\boldsymbol{\xi}_s} \log q(\boldsymbol{x}, \hat{\boldsymbol{\xi}}_s)$ for $s = \pm 1$.[13] We obtained $\hat{\boldsymbol{\xi}}_{+1}$ and $\hat{\boldsymbol{\xi}}_{-1}$ from the training examples of the respective classes and set $\hat{\alpha} = \frac{1}{2}$. In the definition of the TOP kernel, equation 4.6, we did not include any normalization of feature vectors. However, in practical situations, it is effective to normalize features for improving classification performance. Although it is difficult to determine a fair way of normalization, we chose a simple way: each feature of the TOP and the FK is normalized to have mean 0 and variance 1. Both the TOP kernel and FK are combined with SVMs using a bias term. For calculation of feature vectors from HMMs, see the appendix.

When classifying with HMMs, one observes the difference of the log-likelihoods for the two classes and discriminates by thresholding at an ap-

[10] When the TOP kernel is applied for separating two classes, the class-conditional models of both classes need to be known. In contrast, the FK is often used in filtering problems where the model of only one class is known (Jaakkola & Haussler, 1999).

[11] Several HMM models have been engineered for protein classification (Durbin et al., 1998). However, we do not use such HMMs because the main purpose of the experiment is to compare FK and TOP. Furthermore, the performances achieved with plain HMM models are lower than the ones presented here using discriminative training, which is well in line with results by Jaakkola, Diekhans, et al. (2000).

[12] We mainly followed the implementation of Durbin et al. (1998). For implementation details, see Sonnenburg (2001).

[13] Note that $\partial_\alpha v(\boldsymbol{x}, \hat{\theta})$ is a constant, which does not depend on $\boldsymbol{x}$, so it is not included in the feature vector.

Table 2: P Values of Statistical Tests in the Protein Classification Experiments.

Methods	Test	1-2	1-3	1-4	2-3	2-4	3-4
P, FK	T	0.95	0.14	0.78	0.0032^{**}	0.79	0.12
	WX	0.85	0.041^{*}	0.24	0.0040^{**}	0.80	0.026^{*}
P, TOP	T	0.015^{*}	$1.7 \times 10^{-8**}$	0.11	$3.0 \times 10^{-12**}$	0.059	$5.3 \times 10^{-5**}$
	WX	$4.3 \times 10^{-4**}$	$6.1 \times 10^{-5**}$	0.030^{*}	$6.1 \times 10^{-5**}$	0.035^{*}	$3.1 \times 10^{-4**}$
FK, TOP	T	0.0093^{**}	$2.2 \times 10^{-4**}$	0.21	$2.6 \times 10^{-8**}$	0.079	0.0031^{**}
	WX	$8.5 \times 10^{-4**}$	$6.1 \times 10^{-5**}$	0.048^{*}	$6.1 \times 10^{-5**}$	0.0034^{**}	$1.8 \times 10^{-4**}$

Notes: The t-test is denoted as T and the Wilcoxson signed ranks test (as WX).
$^{*}p < 0.05$. $^{**}p < 0.01$.

propriate value. Theoretically, this threshold should be determined by the (true) class prior probability, which is typically unavailable. Furthermore, the estimation of the prior probability from training data often leads to poor results (Durbin et al., 1998). To avoid this problem, the threshold is determined such that the false-positive rate and the false-negative rate are equal on the test set. This threshold is determined in the same way for FK-SVMs and TOP-SVMs.

The hybrid HMM-TOP-SVM system has several model parameters: the number of HMM states, the pseudo count value (Durbin et al., 1998), and the regularization parameter C of the SVM. We determine these parameters as follows. First, the number of states and the pseudo count value are determined such that the error of the HMM on the validation set (i.e., the validation error) is minimized. Based on the chosen HMM model, the parameter C is determined such that the validation error of the TOP-SVM is minimized. Here, the number of states and the pseudo count value are chosen from $\{3, 5, 7, 10, 15, 20, 30, 40, 60\}$ and $\{10^{-10}, 10^{-7}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}\}$, respectively. For C, 15 equally spaced points on the log scale are taken from $[10^{-4}, 10^{1}]$. Note that the model selection is performed in the same manner for the FK as well.

The error rates over 15 different training, validation, and test divisions are shown in Figures 3 and 4. The results of statistical tests are shown in Table 2 as well. In several settings (i.e. 1-3, 2-3, 3-4), the FK performed better than the plug-in estimate with significant difference in average error rates. This result partially agrees with observations in Jaakkola and Haussler (1999). However, our TOP approach outperforms the FK. According to the Wilcoxson signed ranks test, the TOP kernel was better in all settings with significant difference. Also, the t-test judged that the difference is significant except for 1-4 and 2-4. This indicates that the TOP kernel was able to capture discriminative information better than the FK could.

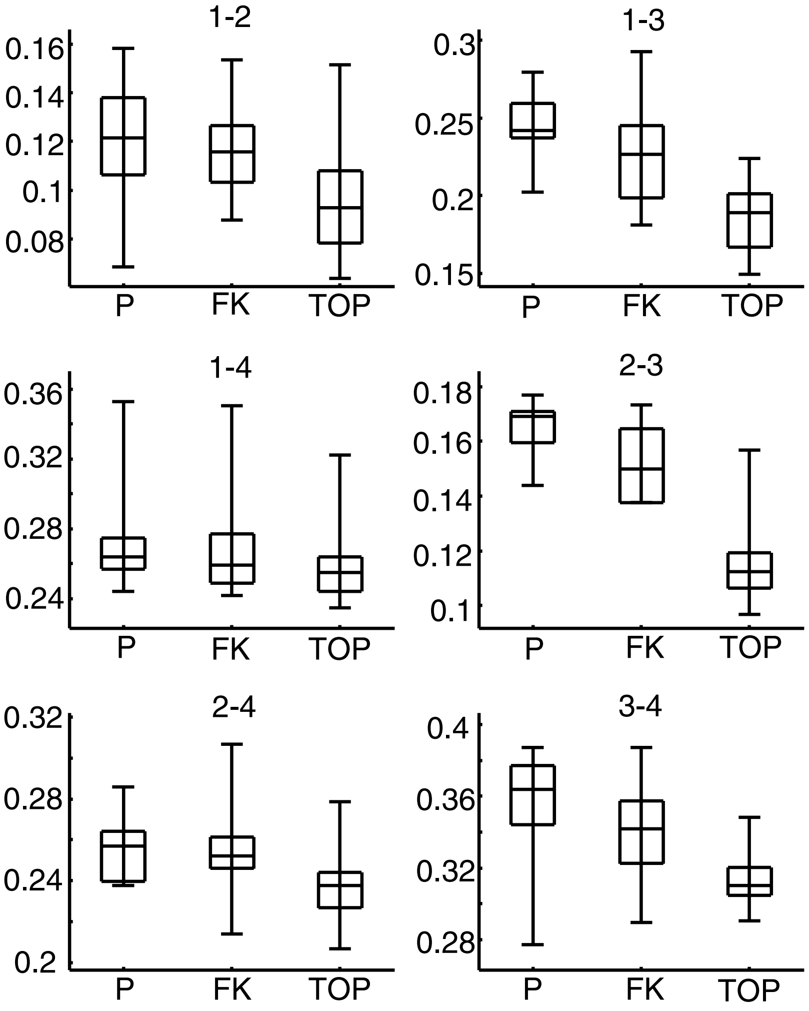

Figure 3: The error rates of SVMs with two feature extractors in the protein classification experiments. The labels P, FK, and TOP denote the plug-in estimate, the Fisher kernel, and the TOP kernel, respectively. Each graph is labeled with the two protein classes used for the experiment.

7 Conclusion

In this study, we presented the new discriminative TOP kernel derived from probabilistic models. We proposed two performance measures to analyze such kernels and gave bounds and rates to gain a better insight into model-dependent feature extractors from probabilistic models. Experimentally, we

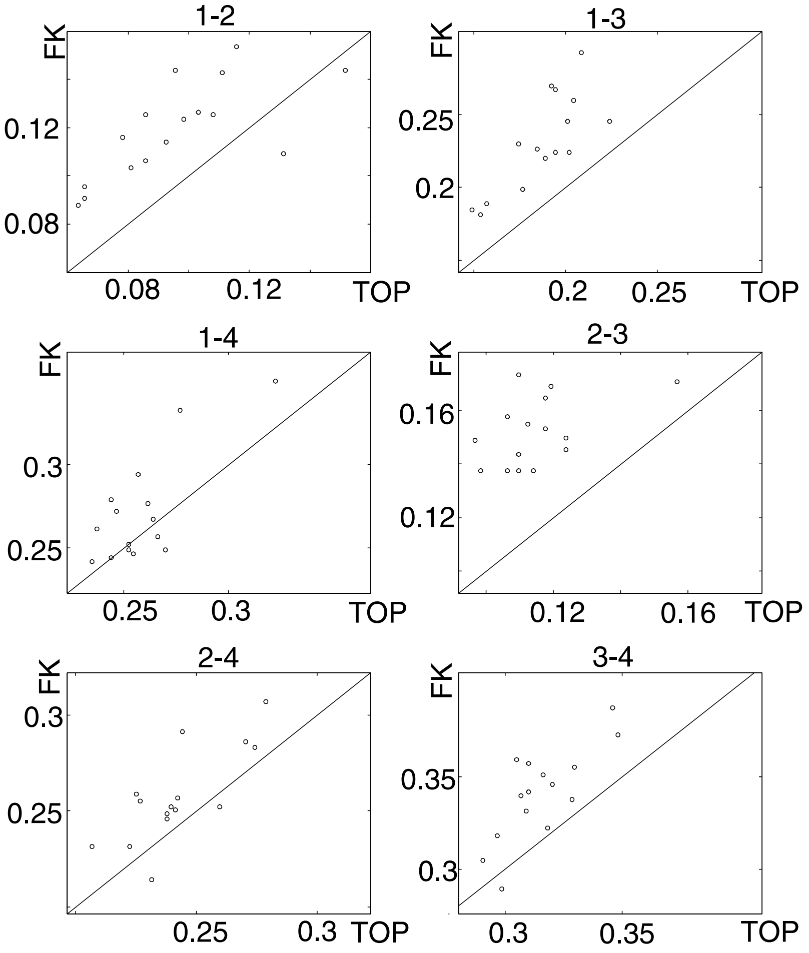

Figure 4: Comparison of the error rates of the SVMs with two feature extractors in protein classification experiments. Every point corresponds to one of the 15 different training, validation, and test set splits. Each graph is labeled with the two protein classes used for the experiment.

showed that the TOP kernel compares favorably to FK and the plug-in estimator on toy data and in a realistic protein classification experiment. Future research will focus on constructing small sample bounds for the TOP kernel to extend the validity of this work. Since other nonlinear transformations F are available for obtaining different and possibly even better features, we will consider learning the nonlinear transformation F from training samples. An interesting point is that so far, TOP kernels perform local linear approximations; it would be interesting to move in the direction of local or even global nonlinear expansions. Recently, it was reported that the FK-based classifiers can be understood in the Bayesian framework of maximum entropy discrimination (Jaakkola, Meila, & Jebara, 2000; Jaakkola, Meila, & Jebara, 1999) when the prior distribution of parameters is chosen in an appropriate way. It is therefore interesting to explore the relationship between the techniques established in this work for the TOP kernel and such Bayesian inference methods.

Appendix: Derivatives with Respect to HMM Parameters

We will illustrate how to compute derivatives of the likelihood with respect to HMM parameters (Rabiner, 1989; Durbin et al., 1998). Let n and m be the number of states and the number of symbols in the alphabet of HMM, respectively. Typically, HMM has the following parameters:

- $\boldsymbol{a} \in \mathbb{R}^{n\times n}$: the transition matrix (a_{ij} denotes the probability of a transition from state i to j).
- $\boldsymbol{b} \in \mathbb{R}^{n\times m}$: the emission matrix ($b_{ik}$ denotes the probability of emitting symbol k in state i)
- $\boldsymbol{p} \in \mathbb{R}^{n}$: the initial state distribution (p_i denotes the probability of the HMM to start in state i)
- $\boldsymbol{q} \in \mathbb{R}^{n}$: the terminal or end state distribution (q_i denotes the probability of the HMM to terminate in state i)

Let us define $\boldsymbol{\lambda} = \{\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{p}, \boldsymbol{q}\}$ for convenience. Let $\mathbf{o}$ denote an observed sequence of length T:

$$\mathbf{o} = (o_0, \ldots, o_{T-1}),\ 1 \leq o_i \leq m.$$

Then the probability that the sequence $\mathbf{o}$ is generated by the HMM is described as

$$\Pr[\mathbf{o} \mid \boldsymbol{\lambda}] = \sum_{\boldsymbol{s}} p_{s_0} b_{s_0 o_0} \left(\prod_{t=0}^{T-2} a_{s_t s_{t+1}} b_{s_{t+1} o_{t+1}} \right) q_{s_{T-1}}, \tag{A.1}$$

where $\sum_{\boldsymbol{s}}$ denotes the sum over all possible state sequences $s_0, \ldots, s_{T-1}$: $\sum_s = \sum_{s_0=1}^{n} \cdots \sum_{s_{T-1}=1}^{n}$. We will describe the derivative of $\Pr[\mathbf{o} \mid \boldsymbol{\lambda}]$ with respect to $\boldsymbol{\lambda}$ using forward and backward variables, where the forward variable α_t^i is defined as

$$\alpha_t^i := \Pr[o_0, o_1, \ldots, o_t, s_t = i \mid \boldsymbol{\lambda}],$$

and the backward variable β_t^i is described as

$$\beta_t^i := \Pr[o_{t+1}, o_{t+2}, \ldots, o_{T-1} \mid s_t = i, \boldsymbol{\lambda}].$$

These variables are efficiently computed by the standard forward-backward algorithm (Durbin et al., 1998). Then the derivatives with respect to parameters are obtained as follows:

$$\frac{\partial \Pr[\mathbf{o} \mid \boldsymbol{\lambda}]}{\partial p_k} = b_{k\,o_0}\beta_0^k \tag{A.2}$$

$$\frac{\partial \Pr[\mathbf{o} \mid \boldsymbol{\lambda}]}{\partial q_k} = \alpha_{T-1}^k \tag{A.3}$$

$$\frac{\partial \Pr[\mathbf{o} \mid \boldsymbol{\lambda}]}{\partial a_{kl}} = \sum_{t=0}^{T-2} \alpha_t^k b_{l\,o_{t+1}} \beta_{t+1}^l \tag{A.4}$$

$$\frac{\partial \Pr[\mathbf{o} \mid \boldsymbol{\lambda}]}{\partial b_{kl}} = \sum_{t=0}^{T-1} I(o_t = l)\frac{\alpha_t^k \beta_t^k}{b_{ko_t}}, \tag{A.5}$$

where $I(o_t = l) = 1$ if $o_t = l$ and 0 otherwise. The parameters in standard HMMs must satisfy the stochasticity conditions:

$$\sum_{j=1}^{n} a_{ij} = 1,\ \sum_{j=1}^{m} b_{ij} = 1,\ \sum_{j=1}^{n} p_j = 1,\ \sum_{j=1}^{n} q_j = 1.$$

For computations of FK and TOP, we use the derivatives with respect to unconstrained parameters $\boldsymbol{\lambda}'$ as in Jaakkola, Diekhans, et al., 2000. These unconstrained parameters are related to the original ones as

$$p_i = p_i' / \sum_{j=1}^{n} p_j',$$

where other parameters a_{ij}', b_{ij}', q_i' have the same relations (the formulas are not shown for brevity). By the chain rule, the derivative with respect to p_i' at the point $p_i' = p_i$ is obtained as

$$\frac{\partial \Pr[\mathbf{o} \mid \boldsymbol{\lambda}]}{\partial p_i'} = \frac{\partial \Pr[\mathbf{o} \mid \boldsymbol{\lambda}]}{\partial p_i} - \sum_{j=1}^{n} p_j \frac{\partial \Pr[\mathbf{o} \mid \boldsymbol{\lambda}]}{\partial p_j}. \tag{A.6}$$

The derivatives with respect to other unconstrained parameters can be obtained in the same way.

Acknowledgments

We thank T. Tanaka, M. Sugiyama, S.-I. Amari, K. Karplus, R. Karchin, F. Sohler, and A. Zien for valuable discussions. We gratefully acknowledge partial support from DFG (JA 379/9-1 and 9-2, MU 987/1-1) and travel grants from EU (Neurocolt II). Thanks also to the anonymous referees for their valuable comments and suggestions that led to a strong improvement of this article.

References

Barndorff-Nielsen, O., & Cox, D. (1989). *Asymptotic techniques for use in statistics*. London: Chapman and Hall.

Bishop, C. (1995). *Neural networks for pattern recognition*. New York: Oxford University Press.

Cox, T., & Ferry, G. (1993). Discriminant analysis using non-metric multidimensional scaling. *Pattern Recognition*, *26*, 145–153.

Devroye, L., Györfi, L., & Lugosi, G. (1996). *A probabilistic theory of pattern recognition*. New York: Springer-Verlag.

Durbin, R., Eddy, S., Krogh, A., & Mitchison, G. (1998). *Biological sequence analysis: Probabilistic models of proteins and nucleic acids*. Cambridge: Cambridge University Press.

Fukunaga, K. (1990). *Introduction to statistical pattern recognition* (2nd ed.). San Diego: Academic Press.

Geiger, D., & Meek, C. (1998). *Graphical models and exponential families* (Tech. Rep. No. MSR-TR-98-10). Redmond, WA: Microsoft Research.

Graepel, T., Herbrich, R., Bollmann-Sdorra, P., & Obermayer, K. (1999). Classification on pairwise proximity data. In M. Kearns, S. Solla, & D. Cohn (Eds.), *Advances in neural information processing systems*, 11 (pp. 438–444). Cambridge, MA: MIT Press.

Hofmann, T., & Buhmann, J. (1997). Pairwise data clustering by deterministic annealing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *19*, 1–14.

Jaakkola, T., Diekhans, M., & Haussler, D. (2000). A discriminative framework for detecting remote protein homologies. *J. Comp. Biol.*, *7*, 95–114.

Jaakkola, T., & Haussler, D. (1999). Exploiting generative models in discriminative classifiers. In M. Kearns, S. Solla, & D. Cohn (Eds.), *Advances in neural information processing systems, 11* (pp. 487–493). Cambridge, MA: MIT Press.

Jaakkola, T., Meila, M., & Jebara, T. (1999). *Maximum entropy discrimination* (Tech. Rep. No. AITR-1668). Cambridge, MA: MIT.

Jaakkola, T., Meila, M., & Jebara, T. (2000). Maximum entropy discrimination. In S. Solla, T. Leen, & K.-R. Müller (Eds.), *Advances in neural information processing systems, 12* (pp. 470–476). Cambridge, MA: MIT Press.

Murata, N., Yoshizawa, S., & Amari, S. (1994). Network information criterion—determining the number of hidden units for an artificial neural network model. *IEEE Trans. Neural Networks*, *5*, 865–872.

Rabiner, L. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, *77*, 257–285.

Rätsch, G., Onoda, T., & Müller, K.-R. (2001). Soft margins for AdaBoost. *Machine Learning*, *42*, 287–320.

Sonnenburg, S. (2001). *Hidden Markov model for genome analysis* [project report]. Berlin: Humboldt University.

Received March 22, 2001; accepted March 25, 2002.