Modeling Category Structures with a Kernel Function

Hiroya Takamura
Precision and Intelligence Laboratory
Tokyo Institute of Technology
4259 Nagatsuta Midori-ku Yokohama,
226-8503 Japan
takamura@pi.titech.ac.jp

Yuji Matsumoto
Department of Information Technology
Nara Institute of Science and Technology
8516-9 Takayama Ikoma Nara,
630-0101 Japan
matsu@is.aist-nara.ac.jp

Hiroyasu Yamada
School of Information Science
Japan Advanced Institute of Science and Technology
1-1 Asahidai Tatsunokuchi Ishikawa, 923-1292 Japan
h-yamada@jaist.ac.jp

Abstract

We propose one type of TOP (Tangent vector Of the Posterior log-odds) kernel and apply it to text categorization. In a number of categorization tasks including text categorization, negative examples are usually more common than positive examples and there may be several different types of negative examples. Therefore, we construct a TOP kernel, regarding the probabilistic model of negative examples as a mixture of several component models respectively corresponding to given categories. Since each component model of our mixture model is expressed using a one-dimensional Gaussian-type function, the proposed kernel has an advantage in computational time. We also show that the computational advantage is shared by a more general class of models. In our experiments, the proposed kernel used with Support Vector Machines outperformed the linear kernel and the Fisher kernel based on the Probabilistic Latent Semantic Indexing model.

1 Introduction

Recently, Support Vector Machines (SVMs) have been actively studied because of their high generalization ability (Vapnik, 1998). In the formulation of SVMs, functions which measure the similarity of two examples take an important role. These functions are called *kernel functions*. The usual dot-product of two vectors respectively corresponding to two examples is often used. Although some variants to the usual dot-product are sometimes used (for example, higher-order polynomial kernels and RBF kernels), the distribution of examples is not taken into account in such kernels.

However, new types of kernels have more recently been proposed; they are based on the probability distribution of examples. One is Fisher kernels (Jaakkola and Haussler, 1998). The other is TOP (Tangent vector Of the Posterior log-odds) kernels (Tsuda et al., 2002). While Fisher kernels are constructed on the basis of a generative model of data, TOP kernels are based on the class-posterior probability, that is, the probability that the positive class occurs given an example. However, in order to use those kernels, we have to select a probabilistic model of data. The selection of a model will affect categorization result. The present paper provides one solution to this issue. Specifically, we proposed one type of TOP kernel, because it has been reported that TOP kernels perform better than Fisher kernels in terms of categorization accuracy.

We briefly explain our kernel. We focus on negative examples in binary classification. Negative examples are usually more common than positive examples. There may be several different types of negative examples. Furthermore, the categories of negative examples are sometimes explicitly given (for example, the situation where we are given documents, each of which has one of three categories “sports”,”politics” and “economics”, and we are to extract documents with “politics”). In such a situation, the probabilistic model of negative examples can be regarded as a mixture of several component models. We effectively use this property. Although many other models can be used, we propose a model based on the separating hyperplanes in the original feature space. Specifically, a one-dimensional Gaussian-type function normal to a hyperplane corresponds to a category. The negative class is then expressed as a kind of Gaussian mixture. The reason for the selection of this model is that the resulting kernel has an advantage in computational time. The kernel based on this mixture model, what we call

Hyperplane-based TOP (HP-TOP) kernel, can be computed efficiently in spite of its high dimensionality. We later show that the computational advantage is shared by a more general class of models.

In the experiments of text categorization, in which SVMs are used as classifiers, our kernel outperformed the linear kernel and the Fisher kernel based on the Probabilistic Latent Semantic Indexing model proposed by Hofmann (2000) in terms of categorization accuracy.

2 SVMs and Kernel Method

In this section, we explain SVMs and the kernel method, which are the basis of our research. SVMs have achieved high accuracy in various tasks including text categorization (Joachims, 1998; Dumais et al., 1998).

Suppose a set D^l of ordered pairs consisting of a feature vector and its label

$$\begin{aligned} D^l = \quad & \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \cdots, (\mathbf{x}_l, y_l)\}, \\ & (\forall i,\ \mathbf{x}_i \in \mathbf{R}^{|I|}, y_i \in \{-1, 1\}) \end{aligned} \tag{1}$$

is given. D^l is called *training data*. I is the set of feature indices. In SVMs, a separating hyperplane ($f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} - b$) with the largest margin (the distance between the hyperplane and its nearest vectors) is constructed.

Skipping the details of SVMs' formulation, here we just show the conclusion that, using some real numbers α_i^* $(\forall i)$ and b^*, the optimal hyperplane is expressed as follows:

$$f(\mathbf{x}) = \sum_i \alpha_i^* y_i \mathbf{x}_i \cdot \mathbf{x} - b^*. \tag{2}$$

We should note that only dot-products of examples are used in the above expression.

Since SVMs are linear classifiers, their separating ability is limited. To compensate for this limitation, the *kernel method* is usually combined with SVMs (Vapnik, 1998).

In the kernel method, the dot-products in (2) are replaced with more general inner-products $K(\mathbf{x}_i, \mathbf{x})$ (*kernel functions*). The polynomial kernel $(\mathbf{x}_i \cdot \mathbf{x}_j + 1)^d$ $(d \in \mathbf{N}_+)$ and the RBF kernel $\exp\{-\|\mathbf{x}_i - \mathbf{x}_j\|^2 / 2\sigma^2\}$ are often used. Using the kernel method means that feature vectors are mapped into a (higher dimensional) Hilbert space and linearly separated there. This mapping structure makes non-linear separation possible, although SVMs are basically linear classifiers.

Another advantage of the kernel method is that although it deals with a high dimensional (possibly infinite) space, explicit computation of high dimensional vectors is not required. Only the general inner-products of two vectors need to be computed. This advantage leads to a relatively small computational overhead.

3 Kernels from Probabilistic Models

Recently new type of kernels which connect generative models of data and discriminative classifiers such as SVMs, have been proposed: the Fisher kernel (Jaakkola and Haussler, 1998) and the TOP (Tangent vector Of the Posterior log-odds) kernel (Tsuda et al., 2002).

3.1 Fisher Kernel

Suppose we have a probabilistic generative model $p(\mathbf{x}|\theta)$ of the data (we denote an example by $\mathbf{x}$). The Fisher score of $\mathbf{x}$ is defined as $\nabla_\theta \log p(\mathbf{x}|\theta)$, where ∇_θ means partial differentiation with respect to the parameters θ. The Fisher information matrix is denoted by $I(\theta)$ (this matrix defines the geometric structure of the model space). Then, the Fisher kernel at an estimate $\hat{\theta}$ is given by:

$$\begin{aligned} & K(\mathbf{x}^1, \mathbf{x}^2) \\ = \quad & (\nabla_\theta \log p(\mathbf{x}^1|\hat{\theta}))^t I^{-1}(\hat{\theta})(\nabla_\theta \log p(\mathbf{x}^2|\hat{\theta})) \end{aligned} \tag{3}$$

The Fisher score of an example approximately indicates how the model will change if the example is added to the training data used in the estimation of the model. That means, the Fisher kernel between two examples will be large, if the influences of the two examples to the model are similar and large (Tsuda and Kawanabe, 2002).

The matrix $I(\theta)$ is often approximated by the identity matrix to avoid large computational overhead.

3.2 TOP Kernel

On the basis of a probabilistic model of the data, TOP kernels are designed to extract feature vectors $\mathbf{f}_{\hat{\theta}}$ which are considered to be useful for categorization with a separating hyperplane.

We begin with the proposition that, between the generalization error $R(\mathbf{f}_{\hat{\theta}})$ and the expected error of the posterior probability $D(\mathbf{f}_{\hat{\theta}})$, the relation $R(\mathbf{f}_{\hat{\theta}}) - L^* \leq 2D(\mathbf{f}_{\hat{\theta}})$ holds, where L^* is the Bayes error. This inequality means that minimizing $D(\mathbf{f}_{\hat{\theta}})$ leads to reducing the generalization error $R(\mathbf{f}_{\hat{\theta}})$. $D(\mathbf{f}_{\hat{\theta}})$ is expressed, using a logistic function $F(t) = 1/(1 + \exp(-t))$, as

$$\begin{aligned} & D(\mathbf{f}_{\hat{\theta}}) \\ = & \min_{\mathbf{w}, b} E_{\mathbf{X}} |F(\mathbf{w} \cdot \mathbf{f}_{\hat{\theta}} - b) - P(y = +1|\mathbf{x}, \theta^*)|, \end{aligned} \tag{4}$$

where θ^* denotes the actual parameters of the model. The TOP kernel consists of features which *can* minimize $D(\mathbf{f}_{\hat{\theta}})$. In other words, we would like to have feature vectors $\mathbf{f}_{\hat{\theta}}$ that satisfy the following:

$$\forall \mathbf{x}, \quad \mathbf{w} \cdot \mathbf{f}_{\hat{\theta}}(\mathbf{x}) - b = F^{-1}(P(y = +1|\mathbf{x}, \theta^*)). \tag{5}$$

for certain values of $\mathbf{w}$ and b.

For that purpose, we first define a function $v(\mathbf{x}, \theta)$:

$$\begin{aligned} & v(\mathbf{x}, \theta) \equiv F^{-1}(P(y = +1|\mathbf{x}, \theta)) \\ & = \log P(y = +1|\mathbf{x}, \theta) - \log P(y = -1|\mathbf{x}, \theta). \end{aligned} \tag{6}$$

The first-order Taylor expansion of $v(\mathbf{x}, \theta^*)$ around the estimate $\hat{\theta}$ is

$$v(\mathbf{x}, \theta^*) \approx v(\mathbf{x}, \hat{\theta}) + \sum_i (\theta_i^* - \hat{\theta}_i) \frac{\partial v(\mathbf{x}, \hat{\theta})}{\partial \theta_i}. \quad (7)$$

If $\mathbf{f}_{\hat{\theta}}$ is of the following form:

$$\mathbf{f}_{\hat{\theta}}(\mathbf{x}) = \big(v(\mathbf{x}, \hat{\theta}), \partial v(\mathbf{x}, \hat{\theta})/\partial\theta_1, \cdots, \partial v(\mathbf{x}, \hat{\theta})/\partial\theta_p\big), \quad (8)$$

and if $\mathbf{w}$ and b are properly chosen as

$$\mathbf{w} = (1, \theta_1^* - \hat{\theta}_1, \cdots, \theta_p^* - \hat{\theta}_p), \quad b = 0, \quad (9)$$

then (5) is approximately satisfied. Thus, the TOP kernel is defined as

$$K(\mathbf{x}_1, \mathbf{x}_2) = \mathbf{f}_{\hat{\theta}}(\mathbf{x}_1) \cdot \mathbf{f}_{\hat{\theta}}(\mathbf{x}_2). \quad (10)$$

A detailed discussion of the TOP kernel and its theoretical analysis have been given by Tsuda et al (Tsuda et al., 2002).

4 Related Work

Hofmann (2000) applied Fisher kernels to text categorization under the Probabilistic Latent Semantic Indexing (PLSI) model (Hofmann, 1999).

In PLSI, the joint probability of document $\mathbf{d}$ and word w is :

$$P(\mathbf{d}, w) = \sum_k P(z_k) P(\mathbf{d}|z_k) P(w|z_k), \quad (11)$$

where variables z_k correspond to latent classes. After the estimation of the model using the EM algorithm, the Fisher kernel for this model is computed. The average log-likelihood of document $\mathbf{d}$ normalized by the document length is given by

$$l(\mathbf{d}) = \sum_j \hat{P}(w_j|\mathbf{d}) \log \sum_k P(w_j|z_k) P(z_k|\mathbf{d}), \quad (12)$$

where

$$\hat{P}(w_j|\mathbf{d}) = \frac{freq(w_j, \mathbf{d})}{\sum_m freq(w_m, \mathbf{d})}. \quad (13)$$

They use *spherical parameterization* (Kass and Vos, 1997) instead of the original parameters in the model. They define parameters $\rho_{jk} = 2\sqrt{P(w_j|z_k)}$ and $\rho_k = 2\sqrt{P(z_k)}$, and obtained

$$\frac{\partial l(\mathbf{d})}{\partial \rho_{jk}} = \frac{\hat{P}(w_j|\mathbf{d}) P(z_k|\mathbf{d}, w_j)}{\sqrt{P(w_j|z_k)}}, \quad (14)$$

$$\frac{\partial l(\mathbf{d})}{\partial \rho_k} \approx \frac{P(z_k|\mathbf{d})}{\sqrt{P(z_k)}}. \quad (15)$$

Thus, the Fisher kernel for this model is obtained as described in Appendix A.

The first term of (31) corresponds to the similarity through latent spaces. The second term corresponds to the similarity through the distribution of each word. The number of latent classes z_k can affect the value of the kernel function. In the experiment of (Hofmann, 2000), they computed the kernels with the different numbers (1 to 64) of z_k and added them together to make a robust kernel instead of deciding one specific number of latent classes z_k.

They concluded that the Fisher kernel based on PLSI is effective when a large amount of unlabeled examples are available for the estimation of the PLSI model.

5 Hyperplane-based TOP Kernel

In this section, we explain our TOP kernel.

5.1 Derivation of HP-TOP kernel

Suppose we have obtained the parameters $\mathbf{w}_c$ and b_c of the separating hyperplane for each category $c \in C_{category}$ in the original feature space, where $C_{category}$ denotes the set of categories.

We assume that the class-posteriors $P_c(+1|\mathbf{d})$ and $P_c(-1|\mathbf{d})$ are expressed as[1]

$$P_c(+1|\mathbf{d}) = \frac{P(c) q(\mathbf{d}|c)}{\sum_{c'} P(c') q(\mathbf{d}|c')}, \quad (16)$$

$$P_c(-1|\mathbf{d}) = \frac{\sum_{e \neq c} P(e) q(\mathbf{d}|e)}{\sum_{c'} P(c') q(\mathbf{d}|c')} \quad (17)$$

where, for any category x, component function $q(\mathbf{d}|x)$ is of Gaussian-type:

$$q(\mathbf{d}|x) = \frac{1}{\sqrt{2\pi\sigma_x^2}} \exp\{-\frac{((\mathbf{w}_x \cdot \mathbf{d} - b_x) - \mu_x)^2}{2\sigma_x^2}\}, \quad (18)$$

with the mean μ_x of a random variable $\mathbf{w}_x \cdot \mathbf{d} - b_x$ and the variance σ_x. Those parameters are estimated with the maximum likelihood estimation, as follows:

$$\mu_x = \frac{\sum_{(\mathbf{d},y) \in D^l, y=x} \{\mathbf{w}_x \cdot \mathbf{d} - b_x\}}{|\{(\mathbf{d},y) \in D^l | y=x\}|}, \quad (19)$$

$$\sigma_x = \frac{\sum_{(\mathbf{d},y) \in D^l, y=x} \{\mathbf{w}_x \cdot \mathbf{d} - b_x - \mu_x\}^2}{|\{(\mathbf{d},y) \in D^l | y=x\}|}. \quad (20)$$

We choose the Gaussian-type function as an example.However, this choice is open to argument, since some other models also have the same computational advantage as described in Section 5.4.

We set $\theta_{x1} = \mu_x/\sigma_x^2$, $\theta_{x2} = -1/2\sigma_x^2$. Although θ_{x1} and θ_{x2} are not the natural parameters of this model,

[1] We cannot say $q(\mathbf{d}|x)$ is a generative probability of $\mathbf{d}$ given class x, because it is one-dimensional and not valid as a probability density in the original feature space.

we parameterize this model using the parameters θ_{x1}, θ_{x2}, $\mathbf{w}_x$, b_x and $P(x)$ ($\forall x \in C_{category}$) for simplicity. Using this probabilistic model,we compute function $v(\mathbf{d}, \theta)$ as described in Appendix B (θ denotes $\{\mathbf{w}_x, b_x, \theta_{x1}, \theta_{x2} | x \in C_{category}\}$ and w_{xi} denotes the i-th element of the weight vector $\mathbf{w}_x$).

The partial derivatives of this function with respect to the parameters are in Appendix C.

Then we can follow the definition (10) to obtain our version of the TOP kernel. We call this new kernel a *hyperplane-based TOP (HP-TOP)* kernel.

5.2 Properties of HP-TOP kernel

In the derivatives (39), which provide the largest number of features, original features d_i are accompanied by other factors computed from probability distributions. This form suggests that two vectors are considered to be more similar, if they have similar distributions over categories. In other words, an occurrence of a word can have different contribution to the classification result, depending on the context (i.e., the other words in the document). This property of the HP-TOP kernel can lead to the effect of word sense disambiguation, because "bank" in a financial document is treated differently from "bank" in a document related to a river-side park.

The derivatives (34) and (35) correspond to the first-order differences, respectively for the positive class and the negative class. Similarly, the derivatives (36) and (37) for the second-order differences. The derivatives (40) and (41) are for the first-order differences normalized by the variances.

The derivatives other than (38) and (38) directly depend on the distance from a hyperplane, rather than on the value of each feature. These derivatives enrich the feature set, when there are few *active words*, by which we mean the words that do not occur in the training data. For this reason, we expect that the HP-TOP kernel works well for a small training dataset.

5.3 Computational issue

Computing the kernel in this form is time-consuming, because the number of components of type (39) can be very large:

$$O(|I| \times |C_{category}|), \tag{21}$$

where I denotes the set of indices for original features.

However, we can avoid this heavy computational cost as follows. Let us compute the dot-product of derivatives (39) of two vectors $\mathbf{d}^1$ and $\mathbf{d}^2$, which is shown in Appendix D. The last expression (45) is regarded as the scalar product of two dot-products. Thus, by preserving vectors $\mathbf{d}$ and

$$\left(-\frac{P(e)q(\mathbf{d}|e)}{P_{-c}(\mathbf{d})} \frac{\mu_e - (\mathbf{w}_e \cdot \mathbf{d} - b_e)}{\sigma_e^2} \right)_{e \neq c, e \in C_{category}}, \tag{22}$$

we can efficiently compute the dot-product in (39); the computational complexity of a kernel function is

$$O(|I|), \tag{23}$$

on the condition that the original dimension is larger than the number of categories. Thus, from the viewpoint of computational time, our kernel has an advantage over some other kernels such as the PLSI-based Fisher kernel in Section 4, which requires the computational complexity of $O(|I| \times |C_{cluster}|)$, where $C_{cluster}$ denotes the set of clusters.

In the PLSI-based Fisher kernel, each word has a probability distribution over latent classes. In this sense, the PLSI-based Fisher kernel is more detailed, but detailed models are sometimes suffer overfitting to the training data and have the computational disadvantage as mentioned above.

The PLSI-based Fisher kernel can be extended to a TOP kernel by using given categories as latent classes. However, the problem of computational time still remains.

5.4 General statement about the computational advantage

So far, we have discussed the computational time for the kernel constructed on the Gaussian mixture. However, the computational advantage of the kernel, in fact, is shared by a more general class of models.

We examine the required conditions for the computational advantage. Suppose the class-posteriors have the mixture form as Equations (16) and (17), but function $q(\mathbf{d}|x)$ does not have to be a Gaussian-type function. Instead, function $q(\mathbf{d}|x)$ is supposed to be represented using some function r parametrized by $\mathbf{w}_e$ and b, as:

$$q(\mathbf{d}|x) \quad = \quad r(f_x(\mathbf{d})|x), \tag{24}$$

where f_x is a scalar function. Then, let us obtain the derivative of $v(\mathbf{d}, \theta)$ with respect to w_{ei}, which is the bottleneck of kernel computation:

$$\begin{aligned} & \frac{\partial v(\mathbf{d}, \theta)}{\partial w_{ei}} \\ = \quad & \frac{-P(e)q(\mathbf{d}|e)}{P_{-c}(\mathbf{d})} \frac{\partial r(f_e(\mathbf{d})|e)}{\partial w_{ei}} \\ = \quad & \frac{-P(e)q(\mathbf{d}|e)}{P_{-c}(\mathbf{d})} \frac{\partial r(f_e(\mathbf{d})|e)}{\partial f_e(\mathbf{d})} \frac{\partial f_e(\mathbf{d})}{\partial w_{ei}}. \end{aligned} \tag{25}$$

The first two factors of (25) do not depend on i. Therefore, if the last factor of (25) is variable-separable with respect to e and i:

$$\frac{\partial f_e(\mathbf{d})}{\partial w_{ei}} \quad = \quad S(e)T(i), \tag{26}$$

where S and T are some function, then the derivative (25) is also variable-separable. In such cases, the efficient computation described in Section 5.3 is possible by preserving the vectors:

$$\big(T(i)\big)_{i\in I}\,, \tag{27}$$

$$\left(-\frac{P(e)q(\mathbf{d}|e)}{P_{-c}(\mathbf{d})}\frac{\partial r(f_e(\mathbf{d})|e)}{\partial f_e(\mathbf{d})}S(e)\right)_{e\neq c, e\in C_{category}}. \tag{28}$$

We have now obtained the required conditions for the efficient computation: Equation (24) and the variable-separability.

In case of Gaussian-type functions, function f_e and its derivative with respect to w_{ei} are

$$f_e(\mathbf{d}) = \mathbf{w}_e \cdot \mathbf{d} - b_e, \tag{29}$$

$$\frac{\partial f_e(\mathbf{d})}{\partial w_{ei}} = d_i. \tag{30}$$

Thus, the conditions are satisfied.

6 Experiments

Through experiments of text categorization, we empirically compare the HP-TOP kernel with the linear kernel and the PLSI-based Fisher kernel. We use Reuters-21578 dataset[2] with ModApte-split (Dumais et al., 1998). In addition, we delete some texts from the result of ModApte-split, because those texts have no text body. After the deletion, we obtain 8815 training examples and 3023 test examples. The words that occur less than five times in the whole training set are excluded from the original feature set.

We do not use all the 8815 training examples. The size of the actual training data ranges from 1000 to 8000. For each dataset size, experiments are executed 10 times with different training sets.The result is evaluated with F-measures for the most frequent 10 categories (Table 1). The total number of categories is actually 116. However, for small categories, reliable statistics cannot be obtained. For this reason, we regard the remaining categories other than the 10 most frequent categories as one category. Therefore, the model for negative examples is a mixture of 10 component models (9 out of the 10 most frequent categories and the new category consisting of the remaining categories).

We assume uniform priors for categories as in (Tsuda et al., 2002). We computed the Fisher kernels with different numbers (10, 20 and 30) of latent classes and added them together to make a robust kernel (Hofmann, 2000). After the learning in the original feature space, the parameters for the probability distributions are estimated with maximum likelihood estimation as in Equations (19) and (20), followed by the learning with the proposed kernel.

Table 1: The categories and their sizes of Reuters-21578

category	training texts	test texts
earn	2725	1051
acq	1490	644
money-fx	464	141
grain	399	135
crude	353	164
trade	339	133
interest	291	100
ship	197	87
wheat	199	66
corn	161	48

We used an SVM package, TinySVM[3], for SVM computation. The soft-margin parameter C was set to 1.0 (other values of C showed no significant changes in results).

The result is shown in Figure 1 (for macro-average) and Figure 2 (for micro-average). The HP-TOP kernel outperforms the linear kernel and the PLSI-based Fisher kernel for every number of examples.

At each number of examples, we conducted a Wilcoxon Signed Rank test with 5% significance-level, for the HP-TOP kernel and the linear kernel, since these two are better than the other. The test shows that the difference between the two methods is significant for the training data sizes 1000 to 5000. The superiority of the HP-TOP kernel for small training datasets supports our expectation that the enrichment of feature set will lead to better performance for few active words. Although we also expected that the effect of word sense disambiguation would improve accuracy for large training datasets, the experiments do not provide us with an empirical evidence for the expectation. One possible reason is that Gaussian-type functions do not reflect the actual distribution of data. We leave its further investigation as future research.

In this experimental setting, the PLSI-based Fisher kernel did not work well in terms of categorization accuracy. However, this Fisher kernel will perform better when the number of labeled examples is small and a number of unlabeled examples are available, as reported by Hofmann (2000).

We also measured computational time of each method (Figure 3). The vertical axis indicates the average computational time over 100 runs of experiments (10 runs for each category). Please note that training time in this fig-

[2] Available from http://www.daviddlewis.com/resources/.

[3] Available from http://cl.aist-nara.ac.jp/~taku-ku/software/TinySVM/.

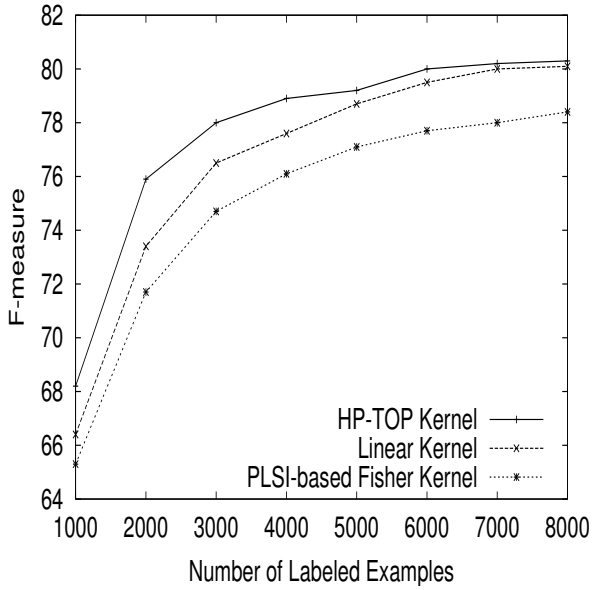

Figure 1: Macro-average of F-measure

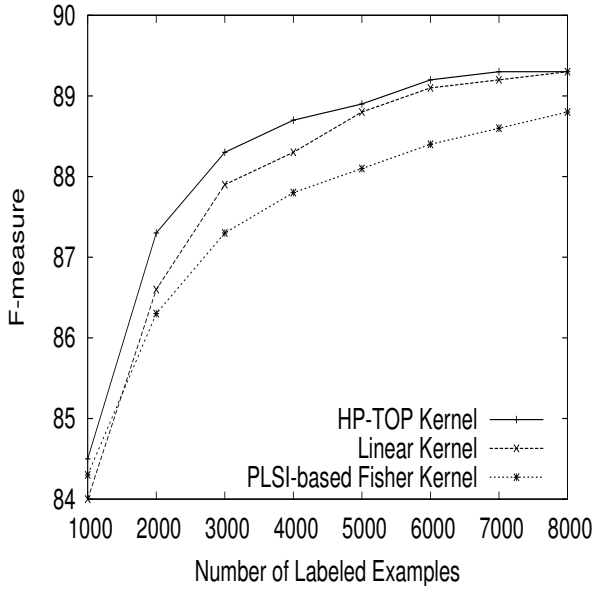

Figure 2: Micro-average of F-measure

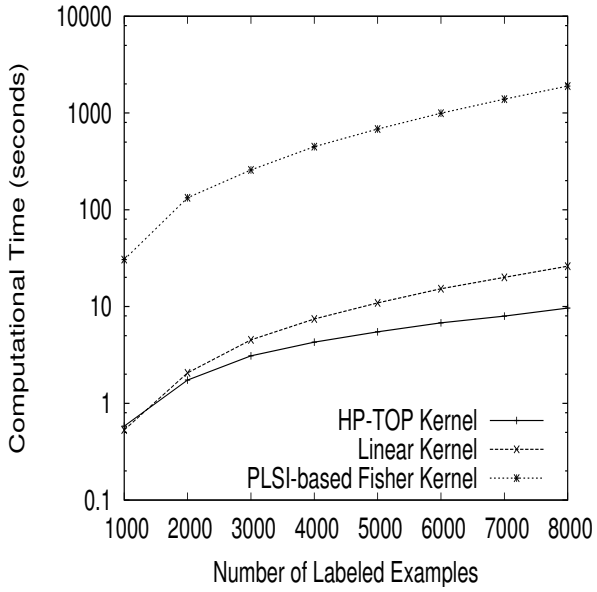

Figure 3: Computational time of each method

ure does not include the computational time required for feature extraction[4]. This result empirically shows that the HP-TOP kernel outperforms the PLSI-based Fisher kernel in terms of computational time as theoretically expected in Section 5.3.

7 Conclusion

We proposed a TOP kernel based on separating hyperplanes. The proposed kernel is created from one-dimensional Gaussians along the normal directions of the hyperplanes. We showed that the computational advantage that the proposed kernel has is shared by a more general class of models. We empirically showed that the proposed kernel outperforms the linear kernel in text categorization.

Although the superiority of the proposed method to the linear kernel was shown, the proposed method has to be further investigated. Firstly, for large data sizes (namely 7000 and 8000), the proposed method was not significantly better than the linear kernel. The effectiveness of the proposed method should be confirmed by more experiments and theoretical analysis. Secondly, we have to compare the proposed method with other kernels in order to check the effectiveness of the kernel function consisting of one-dimensional Gaussians normal to the hyperplanes. The use of Gaussians is open to argument, because their symmetric form is somewhat against our

[4]If the computational time required for feature extraction is included, the HP-TOP kernel cannot be faster than the linear kernel.

intuition.

This model can be extended to incorporate unlabeled examples, for example, using the EM algorithm. In that sense, the combination of PLSI and the semi-supervised EM algorithm is also one promising model. When the category structure of the negative examples is not given, the proposed method is not applicable. We should investigate whether unsupervised clustering can substitute for the category structure.

References

Susan T. Dumais, John Platt, David Heckerman, and Mehran Sahami. 1998. Inductive learning algorithms and representations for text categorization. In *Proceedings of the Seventh International Conference on Information and Knowledge Management (ACM-CIKM98)*, pages 148–155.

Thomas Hofmann. 1999. Probabilistic Latent Semantic Indexing. In *Proceedings of the 22nd Annual ACM Conference on Research and Development in Information Retrieval*, pages 50–57, Berkeley, California, August.

Thomas Hofmann. 2000. Learning the similarity of documents: An information geometric approach to document retrieval and categorization. In *Advances in Neural Information Processing Systems, 12*, pages 914–920.

Tommi Jaakkola and David Haussler. 1998. Exploiting generative models in discriminative classifiers. In *Advances in Neural Information Processing Systems 11*, pages 487–493.

Thorsten Joachims. 1998. Text categorization with support vector machines: Learning with many relevant features. In *Proceedings of the 10th European Conference on Machine Learning*, pages 137–142.

Robert E. Kass and Paul W. Vos. 1997. *Geometrical foundations of asymptotic inference*. New York : Wiley.

Koji Tsuda and Motoaki Kawanabe. 2002. The leave-one-out kernel. In *Proceedings of International Conference on Artificial Neural Networks*, pages 727–732.

Koji Tsuda, Motoaki Kawanabe, Gunnar Rätsch, Sören Sonnenburg, and Klaus-Robert Müller. 2002. A new discriminative kernel from probabilistic models. *Neural Computation*, 14(10):2397–2414.

Vladimir Vapnik. 1998. *Statistical Learning Theory*. John Wiley, New York.

A Fisher Kernel based on PLSI

$$K(\mathbf{d}^1,\mathbf{d}^2) = \sum_k \frac{P(z_k|\mathbf{d}^1)P(z_k|\mathbf{d}^2)}{P(z_k)} + \sum_j \hat{P}(w_j|\mathbf{d}^1)\hat{P}(w_j|\mathbf{d}^2)\sum_k \frac{P(z_k|\mathbf{d}^1,w_j)P(z_k|\mathbf{d}^2,w_j)}{P(w_j|z_k)}, \quad (31)$$

$$where \quad P(z_k|\mathbf{d},w_j) = \frac{P(z_k)P(\mathbf{d}|z_k)P(w_j|z_k)}{\sum_l P(z_l)P(\mathbf{d}|z_l)P(w_j|z_l)} \quad \left(= \frac{P(z_k)P(\mathbf{d}|z_k)P(w_j|z_k)}{P(\mathbf{d},w_j)}\right). \quad (32)$$

B Function v for HP-TOP Kernel

$$\begin{aligned} v(\mathbf{d},\alpha,\mathbf{w},b) &= \log P(+1|\mathbf{d}) - \log P(-1|\mathbf{d}) \\ &= \log \frac{P(c)q(\mathbf{d}|c)}{\sum_{c'} P(c')q(\mathbf{d}|c')} - \log \frac{\sum_{e\neq c} P(e)q(\mathbf{d}|e)}{\sum_{c'} P(c')q(\mathbf{d}|c')} \\ &= \log P(c)q(\mathbf{d}|c) - \log \sum_{e\neq c} P(e)q(\mathbf{d}|e) \\ &= \log P(c)\exp\{\theta_{c1}(\mathbf{w}_c\cdot\mathbf{d}) + \theta_{c2}(\mathbf{w}_c\cdot\mathbf{d})^2 + \frac{\theta_{c1}^2}{4\theta_{c2}} - \frac{1}{2}\log\frac{-\pi}{\theta_{c2}}\} \\ &\quad - \log \sum_{e\neq c} P(e)\exp\{\theta_{e1}(\mathbf{w}_e\cdot\mathbf{d}) + \theta_{e2}(\mathbf{w}_e\cdot\mathbf{d})^2 + \frac{\theta_{e1}^2}{4\theta_{e2}} - \frac{1}{2}\log\frac{-\pi}{\theta_{e2}}\}, \end{aligned} \quad (33)$$

where $\theta_{x1} = \mu_x/\sigma_x^2, \theta_{x2} = -1/2\sigma_x^2$.

C Partial Derivatives

$$\frac{\partial v(\mathbf{d},\theta)}{\partial \theta_{c1}} = \mathbf{w}_c\cdot\mathbf{d} - b_c - \mu_c, \quad (34)$$

$$\frac{\partial v(\mathbf{d},\theta)}{\partial \theta_{e1}} = -\frac{P(e)q(\mathbf{d}|e)}{\sum_{c'\neq c} P(c')q(\mathbf{d}|c')}(\mathbf{w}_e\cdot\mathbf{d} - b_e - \mu_e), \quad (35)$$

$$\frac{\partial v(\mathbf{d},\theta)}{\partial \theta_{c2}} = (\mathbf{w}_c\cdot\mathbf{d} - b_c)^2 - \mu_c^2 - \sigma_c^2, \quad (36)$$

$$\frac{\partial v(\mathbf{d},\theta)}{\partial \theta_{e2}} = -\frac{P(e)q(\mathbf{d}|e)}{\sum_{c'\neq c} P(c')q(\mathbf{d}|c')}\{(\mathbf{w}_e\cdot\mathbf{d} - b_e)^2 - \mu_e^2 - \sigma_e^2\}, \quad (37)$$

$$\frac{\partial v(\mathbf{d},\theta)}{\partial w_{ci}} = \frac{\mu_c - (\mathbf{w}_c\cdot\mathbf{d} - b_c)}{\sigma_c^2}d_i, \quad (38)$$

$$\frac{\partial v(\mathbf{d},\theta)}{\partial w_{ei}} = -\frac{P(e)q(\mathbf{d}|e)}{\sum_{c'\neq c} P(c')q(\mathbf{d}|c')}\frac{\mu_e - (\mathbf{w}_e\cdot\mathbf{d} - b_e)}{\sigma_e^2}d_i, \quad (39)$$

$$\frac{\partial v(\mathbf{d},\theta)}{\partial b_c} = \frac{\mathbf{w}_c\cdot\mathbf{d} - b_c - \mu_c}{\sigma_c^2}, \quad (40)$$

$$\frac{\partial v(\mathbf{d},\theta)}{\partial b_e} = -\frac{P(e)q(\mathbf{d}|e)}{\sum_{c'\neq c} P(c')q(\mathbf{d}|c')}\frac{\mathbf{w}_e\cdot\mathbf{d} - b_e - \mu_e}{\sigma_e^2}, \quad (41)$$

$$\frac{\partial v(\mathbf{d},\theta)}{P(c)} = \frac{1}{P(c)}, \quad (42)$$

$$\frac{\partial v(\mathbf{d},\theta)}{P(e)} = -\frac{P(\mathbf{d}|e)}{\sum_{c'\neq c} P(c')q(\mathbf{d}|c')}. \quad (43)$$

D Dot-product of Derivatives (39) in Appendix C

$$\sum_{e\neq c}\sum_i \frac{\partial v(\mathbf{d}^1,\theta)}{\partial w_{ei}}\frac{\partial v(\mathbf{d}^2,\theta)}{\partial w_{ei}} = \sum_{e\neq c}\sum_i \frac{P(e)^2 q(\mathbf{d}^1|e)q(\mathbf{d}^2|e)}{P_{-c}(\mathbf{d}^1)P_{-c}(\mathbf{d}^2)}\frac{\mu_e - (\mathbf{w}_e\cdot\mathbf{d} - b_e)}{\sigma_e^2}\frac{\mu_e - (\mathbf{w}_e\cdot\mathbf{d} - b_e)}{\sigma_e^2}d_i^1 d_i^2 \quad (44)$$

$$= \left(\sum_{e\neq c} \frac{P(e)^2 q(\mathbf{d}^1|e)q(\mathbf{d}^2|e)}{P_{-c}(\mathbf{d}^1)P_{-c}(\mathbf{d}^2)}\frac{\mu_e - (\mathbf{w}_e\cdot\mathbf{d} - b_e)}{\sigma_e^2}\frac{\mu_e - (\mathbf{w}_e\cdot\mathbf{d} - b_e)}{\sigma_e^2}\right)\mathbf{d}^1\cdot\mathbf{d}^2, \quad (45)$$

where $P_{-c}(\mathbf{d})$ denotes $\sum_{c'\neq c} P(c')q(\mathbf{d}|c')$.