

A Comparative Study on the Use of Labeled and Unlabeled Data for Large Margin Classifiers

Hiroya Takamura and **Manabu Okumura**

Precision and Intelligence Laboratory, Tokyo Institute of Technology

4259 Nagatsuta Midori-ku Yokohama, 226-8503

takamura@pi.titech.ac.jp

oku@pi.titech.ac.jp

Abstract

We propose to use both labeled and unlabeled data with the Expectation-Maximization (EM) algorithm in order to estimate the generative model and use this model to construct a Fisher kernel. The Naive Bayes generative probability is used to model a document. Through the experiments of text categorization, we empirically show that, (a) the Fisher kernel with labeled and unlabeled data outperforms Naive Bayes classifiers with EM and other methods for a sufficient amount of labeled data, (b) the value of additional unlabeled data diminishes when the labeled data size is large enough for estimating a reliable model, (c) the use of categories as latent variables is effective, and (d) larger unlabeled training datasets yield better results.

1 Introduction

One general trend in recent developments for statistical learning approaches in Natural Language Processing (NLP) is the incorporation of labeled and unlabeled data. For example, Naive Bayes (NB) classifiers can be enhanced with the Expectation-Maximization (EM) algorithm (Nigam et al., 2000). However, for large-margin classifiers (Smola et al., 2000) including Support Vector Machines (SVMs), which show a high categorization performance in many tasks of NLP and other fields (Vapnik, 1998; Joachims,

1998; Kudo and Matsumoto, 2001), the question of how to combine labeled and unlabeled data for those classifiers has not been completely answered. In this paper, we propose a solution to this question. The high ability of SVMs makes this question worth to be tackled. Therefore we select SVMs as an example of a large-margin classifier in this research.

One possibility, which we focus on in this paper, is the use of the Fisher kernel (Jaakkola and Haussler, 1998). This kernel function is based on a probabilistic generative model of data and the unlabeled data can be used to estimate the model. Since inputs to SVMs are only labeled data, this way of using unlabeled data is rather indirect. Although the estimation of models can be done with only unlabeled data, information from labeled data is too precious to be vainly discarded, as suggested by Tsuda (2002). For this reason, we propose to use both labeled and unlabeled data with the EM algorithm in order to estimate the generative model. Specifically, we take a generative model proposed by Nigam et al. (2000) and conduct experiments of text categorization.

The objective of this research is to give answers to questions such as “in which situation does EM reinforce the effectiveness of the Fisher kernel?” and “the Fisher kernel using EM is better than NB with EM?”

Several methods have been proposed for the same or similar purpose. Transductive SVMs (TSVMs) proposed by Joachims (1999) use unlabeled examples as labeled, by iteratively relabeling unlabeled examples. This method pursued the same purpose by adopting a different

learning procedure from usual SVMs, while our method adopts a different feature representation. The Fisher kernel based on the Probabilistic Latent Semantic Indexing (Hofmann, 2000) is similar to ours in the sense that it is a combination of the Fisher kernel and the EM algorithm, but the information of labels is not used to estimate the model. Tsuda et al. (Tsuda et al., 2002) proposed, what they call, the Tangent vector Of Posterior log-odds (TOP) kernel. While Fisher kernels use generative probabilities, TOP kernels use posterior probabilities. Although TOP kernels yield high categorization performance, they are designed for binary classification tasks. Therefore we focus on the Fisher kernel here. However, as Tsuda et al (2002) pointed out, if we take categories as latent variables as we do later in our model, the Fisher kernel becomes similar to the TOP kernel.

The rest of the paper is organized as follows. In Section 2, we explain Naive Bayes classifiers and its combination with the EM algorithm. They are used as the base model of our Fisher kernel. In Section 3, SVMs and the Fisher kernel are explained. SVMs are used as classifiers in the experiments. In Section 4, we propose a Fisher kernel constructed on the EM-enhanced Naive Bayes model. Experiments and conclusions are described respectively in Section 5 and Section 6.

2 Naive Bayes Classifiers and EM

This section introduces the multinomial NB model, which we later use as a generative model for the Fisher kernel because the multinomial model reportedly yields better results than other NB models (McCallum and Nigam, 1998).

2.1 Multinomial Model of NB Classifiers

This model has been successfully applied to text categorization and its generative probability of example \mathbf{x} given a category c has the form :

$$P(\mathbf{x}|c, \theta) = P(|\mathbf{x}|) |\mathbf{x}|! \prod_w \frac{P(w|c)^{N(w, \mathbf{x})}}{N(w, \mathbf{x})!} \quad (1)$$

where $P(|\mathbf{x}|)$ denotes the probability that a text of length $|\mathbf{x}|$ occurs, and $N(w, \mathbf{x})$ denotes the number of occurrences of w in text \mathbf{x} . The occurrence of a text is modeled as a set of events, in which a word is drawn from the whole vocabulary.

2.2 Enhancing NB with EM

The EM algorithm is a method to estimate a model which has the maximal likelihood of the data, when some variables can not be observed (those variables are called *latent variables*) (Dempster et al., 1977). Nigam et al. (2000) proposed a combination of Naive Bayes classifiers and the EM algorithm, which we use as a base for constructing a Fisher kernel.

First, we ignore the unrelated factors of Equation (1), and obtain the followings :

$$P(\mathbf{x}|c, \theta) \propto \prod_w P(w|c)^{N(w, \mathbf{x})}, \quad (2)$$

$$P(\mathbf{x}|\theta) \propto \sum_c P(c) \prod_w P(w|c)^{N(w, \mathbf{x})}. \quad (3)$$

In the following, we use θ to denote all the parameters of the model.

If we regard c as a latent variable and introduce a Dirichlet distribution as the prior distribution for the parameters, the Q-function (i.e., the expected log-likelihood) of this models is defined as :

$$Q(\theta|\bar{\theta}) = \log(P(\theta)) + \sum_{\mathbf{x} \in D} \sum_c P(c|\mathbf{x}, \bar{\theta}) \times \log \left(P(c) \prod_w P(w|c)^{N(w, \mathbf{x})} \right), \quad (4)$$

where $P(\theta) \propto \prod_c (P(c)^{\alpha-1} \prod_w (P(w|c)^{\alpha-1}))$; a Dirichlet distribution. α is a hyper-parameter and D is the set of examples used for model estimation.

Then we obtain the following EM steps:
E-step:

$$P(c|\mathbf{x}, \bar{\theta}) = \frac{P(c|\bar{\theta}) P(\mathbf{x}|c, \bar{\theta})}{\sum_c P(c|\bar{\theta}) P(\mathbf{x}|c, \bar{\theta})}, \quad (5)$$

M-step:

$$P(c) = \frac{(\alpha - 1) + \sum_{\mathbf{x} \in D} P(c|\mathbf{x}, \bar{\theta})}{(\alpha - 1)|C| + |D|}, \quad (6)$$

$$P(w|c) = \frac{(\alpha - 1) + \sum_{\mathbf{x} \in D} P(c|\mathbf{x}, \bar{\theta}) N(w, \mathbf{x})}{(\alpha - 1)|W| + \sum_w \sum_{\mathbf{x} \in D} P(c|\mathbf{x}, \bar{\theta}) N(w, \mathbf{x})}, \quad (7)$$

where $|C|$ is the number of categories and $|W|$ is the number of words. For a labeled example, Equation (5) is not used. Instead of that, $P(c|\mathbf{x}, \bar{\theta})$

is set as 1.0 if c is the category of example \mathbf{x} , otherwise 0.

As we can see from Equations (6) and (7), the larger α is, the more uniform the distribution becomes. In the actual application, α is treated as a user-given hyper-parameter.

3 Support Vector Machines and Fisher Kernel

In this section, we briefly explain SVMs and the Fisher kernel.

3.1 Support Vector Machines and Kernel Method

Suppose a set of ordered pairs consisting of a feature vector and its label $\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_l, y_l)\}$, ($\forall i, \mathbf{x}_i \in \mathbf{R}^d, y_i \in \{-1, 1\}$) is given. 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 numbers β_i^* ($\forall i$) and b^* , the optimal hyperplane is expressed as follows:

$$f(\mathbf{x}) = \sum_i \beta_i^* y_i \mathbf{x}_i \cdot \mathbf{x} - b^*. \quad (8)$$

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 (8) 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.

3.2 Fisher kernel

The above kernel functions are not dependent of data distribution. However, Jaakkola and Hausler (1998) proposed a data-dependent kernel,

which is based on the theory of information geometry.

Suppose we have a probabilistic generative model $P(\mathbf{x}|\theta)$ of the data (we denote a sample 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 $K(\mathbf{x}^1, \mathbf{x}^2)$ at an estimate $\hat{\theta}$ is given by :

$$(\nabla_\theta \log P(\mathbf{x}^1|\hat{\theta}))^t I^{-1}(\hat{\theta}) (\nabla_\theta \log P(\mathbf{x}^2|\hat{\theta})). \quad (9)$$

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

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

4 Fisher Kernel on NB model

Following the definition of the Fisher kernel, we construct our version of the Fisher kernel based on the NB model. In order to empirically investigate how the kernel works, we consider some variants with minor differences.

4.1 Derivation of the Fisher Kernel

As in the Fisher kernel proposed by Hofmann (Hofmann, 2000), we use *spherical parameterization* (Kass and Vos, 1997) $\rho_{wc} = 2\sqrt{P(w|c)}$ and $\rho_c = 2\sqrt{P(c)}$ instead of the original parameters $P(w|c)$ and $P(c)$, because this parameterization is supposed to provide a reasonable approximation of $I(\theta)$ by the identity matrix. The features for our Fisher kernel can be obtained by differentiating the Fisher score of each example with respect to each parameter :

$$\begin{aligned} \frac{\partial \log P(\mathbf{x}|\theta)}{\partial(\rho_{wc})} &= \frac{P(c)N(w, \mathbf{x}) \prod_w P(w|c)^{N(w, \mathbf{x})}}{P(\mathbf{x}|\theta)P(w|c)} \\ &\quad \times \frac{\partial P(w|c)}{\partial \rho_{wc}} \\ &= \frac{N(w, \mathbf{x})P(c|\mathbf{x}, \theta)}{P(w|c)} \times \frac{\rho_{wc}}{2} \end{aligned}$$

$$\begin{aligned}
&= \frac{N(w, \mathbf{x})P(c|\mathbf{x}, \theta)}{\sqrt{P(w|c)}}, \quad (10) \\
\frac{\partial \log P(\mathbf{x}|\theta)}{\partial(\rho_c)} &= \frac{P(c|\mathbf{x}, \theta)}{P(c)} \frac{\partial P(c)}{\partial \rho_c} \\
&= \frac{P(c|\mathbf{x}, \theta)}{P(c)} \times \frac{\rho_c}{2} \\
&= \frac{P(c|\mathbf{x})}{\sqrt{P(c)}}. \quad (11)
\end{aligned}$$

Matrix $I(\theta)$ is replaced by the identity matrix.

4.2 Some variants to the proposed kernel

With slight changes, we can construct several variants to our Fisher kernel.

Sometimes we can use additional unlabeled examples other than labeled examples during the model estimation, but sometimes not. We also have an alternative to regard c as a given category or an unknown cluster. When unknown clusters are to be used, the initial values of the parameters are randomly set. The EM algorithm is supposed to form clusters consisting of similar examples.

We name each case to clarify the experimental settings to be described later :
ul-cat : with unlabeled data, category for c ,
ul-cl : with unlabeled data, clusters for c ,
n-cat : without unlabeled data, category for c .
No EM computation was performed for n-cat in our experiments.

5 Experiments

To evaluate our methods, we conducted experiments of multiclass text categorization. The dataset we used is 20 Newsgroups¹, which has 20 categories and consists of 18828 newsgroup articles (we deleted invalid articles which have no text part). The size of each category is in the range of 600 to 1000. Each article has exactly one category label. The words that occur less than ten times in the whole dataset are not used as features in our experiments.

The whole dataset is split into a labeled training set consisting of 100 to 5000 articles, an un-

¹Available from <http://kdd.ics.uci.edu/>.

Another frequently-used dataset is Reuters-21578, which is available from <http://www.daviddlewis.com/resources/>. However, Reuters-21578 is mainly for the binary text categorization task, whereas 20 Newsgroups is for the multiclass task. This is why we use 20 Newsgroups here.

labeled training set consisting of 10000 articles², and a test set consisting of 3765 articles. The two training sets are used to estimate the probabilistic model with the EM algorithm. The unlabeled training set is not used as input to SVMs. The test set is not used for the estimation of the model. We performed experiments with 5 different data-splits, which produce 5 non-overlapping test sets.

Hyper-parameter α for the Dirichlet prior of the parameters is set as 2.0, which is a frequently-used value. When unknown clusters are used as latent variables, the number of the unknown clusters is fixed to 20, which is the same as the number of the given categories.

We used SVMs for a classifier. Although soft-margin parameter C was fixed as 1.0 in our experiments, other values of this parameter did not produce any significant change in results. The one-versus-rest method (Kressel, 1999) was employed for multiclass classification. All the feature vectors are normalized to the same length (Herbrich and Graepel, 2000).

Deterministic annealing EM algorithm (Ueda and Nakano, 1998) is one standard way to avoid local maximum. We adopt this algorithm, which can be implemented with a slight change in Equation (5).

We evaluate results with classification accuracy, which is here defined as the number of correctly classified articles divided by the number of all the articles.

5.1 Comparison of the methods

We compare the proposed methods (i.e., SVM (n-cat, ul-cat, ul-cl)), the linear kernel SVMs and TSVMs³. The linear kernel uses only labeled data for training. The result is summarized in Table 1.

For most of the labeled data sizes, SVM (ul-cat) yields the best accuracy. The exception is the labeled data size 100, where the linear kernel SVMs and TSVMs perform better. This is probably because the small labeled dataset cannot provide sufficient information enough for constructing a model which effectively represents the

²Nigam et al. (2000) also used 10000 additional unlabeled training samples.

³We use the packages, TinySVM (<http://cl.aist-nara.ac.jp/~taku-ku/software/TinySVM/>) and SVM-light (<http://svmlight.joachims.org/>).

Table 1: Categorization accuracy : comparison of the methods.

| #labeled | SVM (n-cat) | SVM (ul-cat) | SVM (ul-cl) | SVM (linear) | TSVM (linear) | NB+EM (ul-cat) | NB |
|----------|----------------|-----------------|----------------|-----------------|------------------|-------------------|------|
| 100 | 23.7 | 30.0 | 29.2 | 31.8 | 33.5 | 26.7 | 13.5 |
| 200 | 41.3 | 46.9 | 43.2 | 44.7 | 46.0 | 37.4 | 24.1 |
| 300 | 45.5 | 53.3 | 51.2 | 51.4 | 52.5 | 41.2 | 32.0 |
| 500 | 54.2 | 61.1 | 57.2 | 60.0 | 60.8 | 50.9 | 43.1 |
| 1000 | 68.0 | 71.0 | 67.8 | 68.9 | 69.3 | 65.0 | 59.3 |
| 2000 | 78.7 | 79.0 | 74.6 | 75.8 | 76.1 | 76.2 | 74.1 |
| 5000 | 85.1 | 85.4 | 82.0 | 82.1 | 82.0 | 83.6 | 82.8 |

category structure. For large labeled data sizes, SVM (n-cat) is comparable to SVM (ul-cat), although the former is slightly worse than the latter. This fact shows that large labeled data can estimate a reliable model without help of unlabeled data. From these observations, we conclude that the Fisher kernel with labeled and unlabeled data works well for a sufficient amount of labeled data, although the value of using unlabeled data diminishes for very large labeled data.

The accuracies of SVM (n-cat) are worse than those of the linear kernel for up to 1000 labeled examples. This is presumably caused by the inaccurate estimation of the model, since the data size used in the model estimation is quite small for those cases. Similar phenomena are seen for NB with EM in the experiment by Nigam et al. (2000), where they have drawn the same conclusion for the decrease of accuracy. The difference between SVM (ul-cat) and SVM (ul-cl) shows that the use of categories as latent variables is effective. On the other hand, the use of unknown clusters as latent variables only decreases the accuracy.

5.2 Different sizes of unlabeled data

In the experiment above, we have shown that when the given categories are used as latent variables, the SVMs with EM-enhanced Fisher kernel perform well with 10000 unlabeled training examples.

A natural question that arises next is what happens if only fewer unlabeled training examples are available. To answer this question, we conduct experiments for different numbers of unlabeled training examples (from 1000 to 10000),

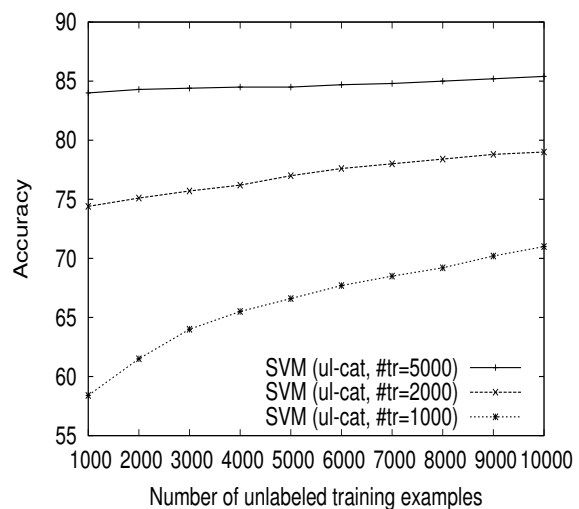


Figure 1: Categorization accuracy : different sizes of unlabeled data, SVM(ul-cat).

while keeping the number of the labeled examples constant (we take 1000, 2000 and 5000).

The result is shown in Figure 1. For every number of labeled training examples, the accuracy monotonically increases as the number of unlabeled training examples increases. This result shows that larger unlabeled training datasets yield better results and that small unlabeled data only decrease accuracy.

The improvement for 5000 labeled examples is not so large as the ones for 1000 and 2000. One possible reason is that an accurate estimation has been done only with 5000 labeled examples and unlabeled training examples do not provide much additional information.

6 Conclusion

We proposed to use both labeled and unlabeled data with the EM algorithm in the estimation of the generative model, and construct the Fisher kernel on the model. We conducted several comparative experiments.

Our conclusions are, (a) SVMs using the Fisher kernel with labeled and unlabeled data work well for a sufficient amount of labeled and unlabeled data, (b) the value of additional unlabeled data diminishes when the labeled data size is large enough for estimating a reliable model, (c) the use of categories as latent variables is effective, (d) larger unlabeled training datasets yield better results.

Future work includes the following.

All these conclusions are based on the NB model. To validate those conclusions in a more general level, experiments with other models and datasets will be required.

In this paper, word clustering was not used. Hofmann and Puzicha (1998) described a two-dimensional EM clustering, which can be used to model co-occurrence of word and document. Using such models for the Fisher kernel has the possibility of increasing accuracy for small labeled datasets, because features will be generalized in the model and the sparseness will be alleviated.

References

- Arthur P. Dempster, Nan M. Laird, and Donald B. Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society Series B*, 39(1):1–38.
- Ralf Herbrich and Thore Graepel. 2000. A PAC-bayesian margin bound for linear classifiers: Why SVMs work. In *Advances in Neural Information Processing Systems, 12*, pages 224–230.
- Thomas Hofmann and Jan Puzicha. 1998. Statistical models for co-occurrence data. Technical Report AIM-1625, Artificial Intelligence Laboratory, Massachusetts Institute of Technology.
- 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 European Conference on Machine Learning*, pages 137–142.
- Thorsten Joachims. 1999. Transductive inference for text classification using support vector machines. In *Proceedings of 16th International Conference on Machine Learning (ICML '99)*, pages 200–209.
- Robert E. Kass and Paul W. Vos. 1997. *Geometrical foundations of asymptotic inference*. New York : Wiley.
- Ulrich Kressel. 1999. Pairwise classification and support vector machines. In Bernhard Schölkopf, Christopher J. C. Burgesa, and Alexander J. Smola, editors, *Advances in Kernel Methods *Support Vector Learning*, pages 255–268. The MIT Press.
- Taku Kudo and Yuji Matsumoto. 2001. Chunking with support vector machines. In *Proceedings of Second Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL 2001)*, pages 192–199.
- Andrew McCallum and Kamal Nigam. 1998. A comparison of event models for naive bayes text classification. In *Proceedings of AAAI-98 Workshop on Learning for Text Categorization*, pages 41–48.
- Kamal Nigam, Andrew Mccallum, Sebastian Thrun, and Tom Mitchell. 2000. Text classification from labeled and unlabeled documents using EM. *Machine Learning*, 39(2/3):103–134.
- Alexander J. Smola, Peter J. Bartlett, Bernhard Schölkopf, and Dale Schuurmans. 2000. *Advances in Large Margin Classifiers*. MIT Press.
- 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.
- Naonori Ueda and Ryohei Nakano. 1998. Deterministic annealing EM algorithm. *Neural Networks*, 11(2):271–282.
- Vladimir Vapnik. 1998. *Statistical Learning Theory*. John Wiley, New York.