Learning Discriminative Fisher Kernels

Laurens van der Maaten LVDMAATEN@GMAIL.COM
Pattern Recognition & Bio-informatics Laboratory, Delft University of Technology, THE NETHERLANDS

Abstract

Fisher kernels provide a commonly used vectorial representation of structured objects. The paper presents a technique that exploits label information to improve the object representation of Fisher kernels by employing ideas from metric learning. In particular, the new technique trains a generative model in such a way that the distance between the log-likelihood gradients induced by two objects with the same label is as small as possible, and the distance between the gradients induced by two objects with different labels is as large as possible. We illustrate the strong performance of classifiers trained on the resulting object representations on problems in handwriting recognition, speech recognition, facial expression analysis, and bio-informatics.

1. Introduction

Classification of structured objects, i.e., automatically assigning a single label to a time series or graph, is an important problem in many domains. A traditional approach to the classification of a structured object $\mathbf{X}_n = \{\mathbf{x}_{n1}, \ldots, \mathbf{x}_{nT}\}$ with $\mathbf{x}_{nt} \in \mathbb{R}^d$ is to train a generative model for each of the classes $y \in \{1, 2, \ldots, C\}$, and to use the resulting generative models in a Bayes classifier through $p(y_n = c|\mathbf{X}_n) \propto p(\mathbf{X}_n|y_n = c)p(y_n = c)$. The main drawback of such an approach to structured object classification is that it cannot make use of powerful discriminative learning techniques that have been developed for the classification of vectorial data, such as kernel machines or metric learning. To address this drawback, various studies have proposed kernel or dissimilarity functions that capture some measure of similarity between structured objects, and that can be used in the training of (kernel) classifiers (Gärtner, 2003; Bicego et al., 2009). Key examples of such kernels and dissimilarity measures are the Fisher kernel (Jaakkola & Haussler, 1998), the TOP kernel (Tsuda et al., 2002a), the probability product kernel (Jebara et al., 2004), marginalized kernels (Tsuda et al., 2002b), and graph edit distances (Bunke & Allermann, 1983) or dynamic time warping (Sakoe & Chiba, 1978).

In general, computing graph edit distances is NP-hard, which limits practical applicability. Probability product kernels and marginalized kernels have the disadvantage that they cannot readily be used in learning settings in which the training objects do not have the same underlying graph structure (such as the learning settings we consider in our experiments); Fisher kernels and TOP kernels do not have such a limitation. The key intuition behind the Fisher kernel (and the TOP kernel) is that similar objects induce similar log-likelihood gradients in the parameters of a generative model $p(\mathbf{X})$. To construct a Fisher kernel for structured objects, one thus computes the log-likelihood gradient induced by each of the objects in the parameters of a generative model. The Fisher kernel function is then defined as a weighted inner product between the gradients of two structured objects. Herein, the weighting is performed using the *Fisher information metric*; this weighting is necessary because different types of model parameters have different scales. In practice, however, the Fisher information metric is often ignored and a (normalized) kernel is used that simply embeds objects in a Euclidean space by using the gradients induced by the objects as features.

Clearly, the embedding computed by the Fisher kernel depends on how the parameters of the generative model $p(\mathbf{X})$ are set. When constructing a Fisher kernel, the generative model is usually trained to maximize the likelihood of the data. However, there is no guarantee that maximum likelihood training leads to an object representation that is well suited for discrimination, i.e., to an embedding in which objects with similar labels are embedded close together and in which objects with different labels are embedded far apart. In this paper, we argue that maximum

Appearing in *Proceedings of the 28th International Conference on Machine Learning*, Bellevue, WA, USA, 2011.

likelihood training of generative models indeed may lead to suboptimal Fisher kernels, because a generative model that models the data well (in terms of data likelihood) leads to gradient representations that are (nearly) zero. To address this problem of Fisher kernels, we propose a technique that learns the model parameters in such a way that the resulting embedding has a low nearest-neighbor error. We evaluate the new technique, called "Fisher kernel learning" (FKL), on data sets for (1) online handwritten character recognition, (2) recognition of spoken Arabic words, (3) recognition of facial expressions from videos, and (4) recognition of mutagen molecules. The results of our experiments reveal the potential benefits of FKL compared to traditional Fisher kernels.

The outline of the remainder of this paper is as follows. In Section 2, we discuss the Fisher kernel in more detail, and we argue that the use of models that are trained to maximize data likelihood as a basis for the Fisher kernel may be suboptimal. Section 3 presents our new technique for learning the model parameters, called Fisher kernel learning (FKL), which aims to set the parameters in such a way as to minimize the nearest-neighbor error of the object embedding. Section 4 presents the results of experiments in which we compare FKL with two other structured object representations (using four different classifiers). Section 5 discusses the results of these experiments. Section 6 concludes the paper, and presents potential directions for future work.

2. Fisher Kernel

Fisher kernels have been proposed as a principled way to use the power of probabilistic generative models in kernel methods (Jaakkola & Haussler, 1998), and have since been successfully used in numerous applications, e.g., in protein homology detection (Jaakkola et al., 2000) and speaker recognition (Campbell et al., 2006).

Suppose we are given a collection of labeled structured objects $\mathcal{D} = \{(\mathbf{X}_1, y_1), (\mathbf{X}_2, y_2), \ldots, (\mathbf{X}_N, y_N)\}$, where the structured object $\mathbf{X}_n = \{\mathbf{x}_{n1}, \mathbf{x}_{n2}, \ldots, \mathbf{x}_{nT}\}$ and the object label $y_n \in \{1, 2, \ldots, C\}$. Also, suppose that the underlying structure of object $\mathbf{X}_n$ can be represented using a graph $G_n = (V_n, E_n)$ in which the i-th vertex corresponds to $\mathbf{x}_{ni}$ and in which an edge (i, j) indicates a pairwise relation between $\mathbf{x}_{ni}$ and $\mathbf{x}_{nj}$. A straightforward way to model the distribution $p(\mathbf{X})$ is using a pairwise Markov Random Field (MRF) over multinomial hidden variables $\mathbf{z} = \{z_1, z_2, \ldots, z_T\}$ with arbitrary emission distributions as follows

$$p(\mathbf{X}) \propto \sum_{\mathbf{z}} \left[\prod_{i \in V} p_\omega(\mathbf{x}_i | z_i) \right] \exp \left[\sum_{(i,j) \in E} A_{z_i z_j} \right]. \quad (1)$$

Herein, $\mathbf{A}$ is a matrix of log-transition probabilities, and the emission distribution $p_\omega(\mathbf{x}_i|z_i)$ is, e.g., a Gaussian or a multinomial distribution with parameters ω. The parameters of the pairwise MRF, $\Theta = \{\omega, \mathbf{A}\}$, are typically trained in such a way as to maximize the log-likelihood $\mathcal{L}(\mathcal{D}) = \sum_{n=1}^{N} \log p(\mathbf{X}_n)$. This training can be performed using a (variational) expectation-mazimization algorithm or using gradient ascent.

The rationale behind the Fisher kernel is that two similar objects induce similar gradients in the parameters of the generative model (Jaakkola & Haussler, 1998). In other words, the Fisher kernel assumes that two similar structured objects $\mathbf{X}_n$ and $\mathbf{X}_m$ have similar partial derivatives $\frac{\partial \mathcal{L}(\mathbf{X}_n)}{\partial \theta}$ and $\frac{\partial \mathcal{L}(\mathbf{X}_m)}{\partial \theta}$ for all $\theta \in \Theta$. To simplify the notation, we denote the gradient of the log-likelihood $\mathcal{L}(\mathbf{X}_n)$ of a single structured object $\mathbf{X}_n$ with respect to the model parameters as $\mathbf{g}_n = \left[\forall \theta \in \Theta : \frac{\partial \mathcal{L}(\mathbf{X}_n)}{\partial \theta} \right]$.

Using the gradients $\mathbf{g}_n$ as *features* that represent the structured object $\mathbf{X}_n$, the Fisher kernel function κ is defined as

$$\kappa(\mathbf{X}_i, \mathbf{X}_j) = \mathbf{g}_i^T \mathbf{U}^{-1} \mathbf{g}_j.$$

Herein, the matrix $\mathbf{U}$ is the *Fisher information metric*, which corrects the similarity measurement for the fact that generative models generally lie on a non-linear Riemannian manifold[1]. In other words, the Fisher kernel is defined as the inner product of the directions of gradient ascent over the manifold, i.e., the inner product of the *natural* gradients (Amari, 1998). The Fisher information metric can be computed as

$$\mathbf{U} = \mathbb{E} \left[\left(\frac{\partial \mathcal{L}(\mathbf{X})}{\partial \Theta} \right)^T \left(\frac{\partial \mathcal{L}(\mathbf{X})}{\partial \Theta} \right) \right]_{p(\mathbf{X})}, \quad (2)$$

where the expectation is over the distribution defined by the generative model. Asymptotically, however, the information metric is immaterial (Jaakkola & Haussler, 1998), which is why the Fisher information metric is often ignored in practice, i.e., it is assumed that $\mathbf{U} = \mathbf{I}$. The resulting *practical* Fisher kernel (Shawe-Taylor & Christianini, 2004) thus simply uses the gradients $\mathbf{g}_n$ as features, without any further rescalings or normalizations.

[1] The Fisher information metric accounts for the fact that a change of, say, 0.1 in the transition parameters has a different effect on the log-likelihoods than a change of 0.1 in the emission parameters.

The key problem of the feature representation used in the Fisher kernel is that maximum likelihood training does not necessarily give rise to a representation that is well suited for classification tasks. In fact, if the distribution represented by the trained model closely resembles the data distribution, the Fisher kernel representation is presumably very bad: the gradients for all objects in the data will be nearly zero. In practice, the Fisher kernel representation often comprises a large number of very small gradients (for objects that have high probability under the model) and a few very large ones (for objects that have low probability under the model). It is unlikely that such a representation forms a good basis for the classification of structured objects. Indeed, it is possible to partly overcome scaling problems by using kernel normalization or by plugging the gradients into a Gaussian kernel with adaptive bandwidth. However, this does not address the problem that the generative model was trained to maximize a different objective than the objective we have in mind: maximizing separation between classes in the embedding.

3. Fisher Kernel Learning

Fisher kernel learning (FKL) aims to address the problem of Fisher kernel representations that maximum likelihood learning may lead to poor object representations. It does so by employing the label information that is available for the objects in the training data. Instead of training the generative model $p(\mathbf{X})$ using maximum likelihood, FKL trains the model in such a way that objects with the same class induce gradients that are similar, whereas objects with different classes induce log-likelihood gradients that are dissimilar. Effectively, FKL thus applies ideas from metric learning (Goldberger et al., 2005; Globerson & Roweis, 2006; Weinberger et al., 2007) to the Fisher representation.

FKL defines the similarity between two structured objects $\mathbf{X}_i$ and $\mathbf{X}_j$ using a *stochastic selection rule*

$$p_{ij} = \frac{\exp(-(\mathbf{g}_i - \mathbf{g}_j)^T \mathbf{W}^T \mathbf{W} (\mathbf{g}_i - \mathbf{g}_j))}{\sum_{j' \neq i} \exp(-(\mathbf{g}_i - \mathbf{g}_{j'})^T \mathbf{W}^T \mathbf{W} (\mathbf{g}_i - \mathbf{g}_{j'}))}, \quad (3)$$

where $\mathbf{W}$ is a diagonal matrix.

The product $\mathbf{M} = \mathbf{W}^T \mathbf{W}$ is a diagonal metric ($\mathbf{M} \succeq 0$) that weighs the partial derivatives; it serves a similar purpose as the Fisher information metric. The probability p_{ij} may be interpreted as the probability that object i picks object j as its nearest neighbor in the log-likelihood gradient embedding. Assuming we perform classifications using a 1-nearest neighbor classifier, p_{ij} may thus be interpreted as the probability that object $\mathbf{X}_i$ inherits the class label of object $\mathbf{X}_j$.

Motivated by neighborhood components analysis (Goldberger et al., 2005), FKL maximizes the *expected number of correctly classified objects* by a 1-nearest neighbor classifier that operates under the above stochastic selection rule, i.e., it maximizes

$$O(\Theta, \mathbf{W}; \mathcal{D}) = \sum_i \sum_{j \neq i} \delta_{y_i y_j} p_{ij}, \quad (4)$$

where $\delta_{y_i y_j}$ is the Kronecker delta. The maximization of Equation 4 is performed with respect to the model parameters Θ and the matrix $\mathbf{W}$. In preliminary experiments, we also investigated maximizing the log-probability of a correct classification by maximizing $O(\Theta, \mathbf{W}; \mathcal{D}) = \sum_i \sum_{j \neq i} \delta_{y_i y_j} \log p_{ij}$. Such an objective function is motivated by MCML (Globerson & Roweis, 2006), but we found it to work less well.

Intuitively, maximizing Equation 4 trains the model in such a way that objects with the same class induce similar gradients in the model parameters (have large p_{ij}), and that objects of a different class induce dissimilar gradients (have small p_{ij}). Here, similarity is defined under the Mahalanobis metric $\mathbf{M}$; the metric $\mathbf{M}$ is learned together with the model parameters Θ. We expect that FKL produces an object representation that is better suited for classification than the Fisher kernel representation discussed in Section 2.

The objective function in Equation 4 contains two types of unknown variables: (1) the posterior probabilities over vertices $\forall i \in V_n : \gamma_{nik} = p(z_{ni} = k | \mathbf{X}_n)$ and edges $\forall (i,j) \in E_n : \xi_{nijkm} = p(z_{ni} = k, z_{nj} = m | \mathbf{X}_n)$, and (2) the parameters Θ and $\mathbf{W}$. This suggests the use of an EM-like algorithm that alternates between (1) computing the posterior probabilities $\boldsymbol{\gamma}_{ni}$ and $\boldsymbol{\xi}_{nij}$ using an (approximate) inference algorithm and (2) increasing the value of Equation 4 by updating the parameters Θ and $\mathbf{W}$ using a (projected) gradient step, keeping the posteriors fixed. In general, such an alternating optimization of Equation 4 is not guaranteed to converge to a local maximum. Nonetheless, we opt to use the alternating optimization algorithm, because we found it to converge in practice. Moreover, alternating optimization is significantly less cumbersome than doing gradient ascent.

Fixing the posteriors $\boldsymbol{\gamma}_{ni}$ and $\boldsymbol{\xi}_{nij}$, the gradient of $O(\Theta, \mathbf{W}; \mathcal{D})$ with respect to a parameter $\theta \in \Theta$ is

$$\frac{\partial O}{\partial \theta} = 2 M_\theta \sum_i \left[\sum_{j \neq i} \left[\delta_{y_i y_j} p_{ij} (\mathbf{g}_i - \mathbf{g}_j) \left(\frac{\partial \mathbf{g}_i}{\partial \theta} - \frac{\partial \mathbf{g}_j}{\partial \theta} \right) \right] - p_i \sum_{j \neq i} \left[p_{ij} (\mathbf{g}_i - \mathbf{g}_j) \left(\frac{\partial \mathbf{g}_i}{\partial \theta} - \frac{\partial \mathbf{g}_j}{\partial \theta} \right) \right] \right],$$

where M_θ represents the element from the diagonal of $\mathbf{M}$ that corresponds to parameter θ, and $p_i = \sum_j p_{ij}$. After each gradient update of the model parameters, the solution may need to be projected back onto the manifold of valid models: e.g., if the emission distributions are multinomial, the parameters of these distributions need to be normalized to sum up to one.

The gradient of Equation 4 (with the posteriors $\boldsymbol{\gamma}_{ni}$ and $\boldsymbol{\xi}_{nij}$ fixed) with respect to $\mathbf{W}$ is given by

$$\frac{\partial O}{\partial \mathbf{W}} = 2\mathbf{W} \circ \frac{\partial O}{\partial \mathbf{M}},$$

where $\circ$ indicates an element-wise multiplication, and where $\frac{\partial O}{\partial \mathbf{M}}$ is given by

$$\frac{\partial O}{\partial \mathbf{M}} = -\sum_i \sum_{j \neq i} \delta_{y_i y_j} p_{ij} \|\mathbf{g}_i - \mathbf{g}_j\|^2 - p_i \sum_{j \neq i} p_{ij} \|\mathbf{g}_i - \mathbf{g}_j\|^2.$$

The metric $\mathbf{M}$ remains positive semidefinite without a projection step, since we defined $\mathbf{M} = \mathbf{W}^T\mathbf{W}$.

4. Experiments

To evaluate the performance of FKL, we performed classification experiments on four data sets: (1) an online handwritten character data set, (2) a data set of spoken Arabic digits, (3) a facial expression analysis data set, and (4) a data set of mutagen molecules. The first three data sets contain variable-length time series, whereas the fourth data set contains general (loopy) graphs. The four data sets are briefly introduced in 4.1. The setup of our experiments is presented in 4.2; the results of the experiments are presented in 4.3.

4.1. Data sets

The online handwritten character data set contains pen trajectory data that consists of three variables, viz., the pen movement in the x-direction and y-direction, and the pen pressure (Williams et al., 2008). The data set contains 2,858 time series with an average length of 120 frames. Each time series corresponds to a single handwritten character that has one of 20 labels. The data set is not completely balanced, but the variation in number of objects per class is small.

The Arabic spoken digit data set contains utterances of digits spoken in Arabic (Hammami & Bedda, 2010). In the collection of the data set, 88 speakers (44 males and 44 females) uttered each of the 10 digits ten times, leading to a total of 8,800 utterances. Each time series consists of 13-dimensional MFCCs that were sampled at 11,025Hz, 16-bits using a Hamming window.

The facial expression analysis data set we used is the second release of the Cohn-Kanade data set (Lucey et al., 2010). The data set contains 593 videos of subjects showing a (posed) facial expression; the average length of each video is 18.1 frames. A subset of 327 of the videos is labeled as corresponding to one of the seven basic emotions (anger, contempt, disgust, fear, happiness, sadness, and surprise); we used this labeled subset in our experiments. We used the project-out inverse compositional algorithm (Matthews & Baker, 2004) to fit active appearance models (Cootes et al., 1998) with 68 facial feature points on all frames of the 327 videos. We represent each frame by the variation[2] of the feature point locations with respect to the first frame (Lucey et al., 2010), leading to time series with a $2 \times 68 = 136$-dimensional feature representation.

The mutagenicity data set comprises 4,337 molecules, which are represented as graphs in which each node corresponds to one of fourteen atoms (i.e., there is a single discrete feature per node) and each edge corresponds to a covalent bond (Riesen & Bunke, 2008). Roughly half of the molecules are mutagen, whereas the other half is nonmutagen. Mutagenicity is an adverse property of a compound that hampers its potential to become a marketable drug. We aim to predict mutagenicity of a molecule based on its structure.

4.2. Experimental setup

In our experiments, we compared FKL to two other structured object embeddings: (1) a vector of the log-likelihoods $\log p(\mathbf{X}|y = c)$ under models that were trained on objects from a single class and (2) the gradients that the objects induce in the model parameters of a model that was trained using maximum likelihood (i.e., the practical Fisher kernel representation). We leave the comparison of FKL to discriminative models such as HCRFs (Quattoni et al., 2010) and discriminative mixtures/HMMs (Eddy et al., 1995; Kim & Pavlovic, 2006) to future work because of space limitations, and because these models do not provide an object embedding like FKL. We also do not compare with classifiers based on graph edit distances (Bunke & Allermann, 1983), as these are NP-hard to compute[3].

In all experiments, we used the pairwise MRFs described in Equation 1 as model. When the structured objects are time series, this model is similar to

[2]Before this variation is computed, variation due to translation, rotation, or rescaling of the face is projected out using a Procrustes alignment.

[3]We note here that for the special case of time series, edit distances and dynamic time warping actually perform very well (Xi et al., 2006; Ding, 2008). However, we are interested in the more general case of structured objects.

an HMM with undirected state chain. On the handwritten character, Arabic speech, and facial expression data sets, we used isotropic Gaussian emission distributions $p_\omega(\mathbf{x}_i|z_i)$ because these data sets have continuous features. On the mutagenicity data set, we used multinomial emission distributions $p_\omega(\mathbf{x}_i|z_i)$ because this data set has discrete features.

To perform inference in the model (i.e., to evaluate the posterior probabilities over vertices and edges), we use belief propagation when inference is tractable, and loopy belief propagation with a sequential message-passing scheme (Kschischang et al., 2001) when inference is intractable. The inference procedures were implemented using libDAI (Mooij, 2010).

To obtain the log-likelihood and Fisher object representations, we perform maximum likelihood learning of the model parameters Θ using an EM-algorithm in tractable models, and using gradient ascent in intractable models. We do not regularize the model parameters. The log-likelihood and Fisher object representations are computed using the model obtained after maximum likelihood learning has converged.

To obtain the FKL object representations, we train the model using an optimizer that alternates between (1) evaluating the posterior probabilities over vertices and edges and (2) maximizing $O(\Theta, \mathbf{W}; \mathcal{D})$ with respect to the models parameters Θ and the matrix $\mathbf{W}$ by performing a step in the direction of steepest ascent, keeping the posteriors fixed. In our implementation of FKL, the step size is determined using a line search that satisfies the Wolfe conditions (we used back-tracking and cubic inter/extrapolation).

For data sets with continuous features, we normalized the data to be zero-mean, unit-variance. In models that were trained using maximum likelihood, the emission distributions were initialized by training a mixture model using an EM-algorithm. In models that were trained using FKL, the parameters of the emission distributions were initialized randomly. In all models, the transition log-probabilities $\mathbf{A}$ were initialized to 0. The metric $\mathbf{M}$ was initialized as $\text{diag}\left(\frac{N}{\sum_{n=1}^{N} \mathbf{g}_n^2}\right)$, in which the gradients $\mathbf{g}_n$ were computed using the initial model.

The classification is performed by feeding the object representations into three classifiers: (1) a softmax classifier, (2) a linear SVM, and (3) a large-margin nearest neighbor classifier (LMNN; Weinberger et al. (2007)). The L2-regularization parameter of the softmax classifiers and the slack variable of the SVMs were determined based on cross-validation tests on a small held-out validation set. For the log-likelihood representation, we also investigated a Bayes classifier that computes $p(y = c|\mathbf{X}) \propto p(\mathbf{X}|y = c)p(y = c)$.

Table 1. Generalization errors (in %) on the handwritten character data set for four different classifiers (Bayes classifier, logistic regressor, SVM, and LMNN + k-NN) on three different object embeddings (log-likelihoods of class-specific models, Fisher representations, and FKL representations), using various numbers of hidden states. The table reports the generalization error over 10 folds. Best performance for each number of hidden states is typeset in boldface.

Classif.	K	Likelih.	Fisher	FKL
Bayes	2	17.68	–	–
	5	10.04	–	–
	10	6.98	–	–
Softmax	2	4.70	11.16	10.04
	5	4.84	5.58	4.25
	10	4.67	5.33	3.82
SVM	2	**3.86**	9.37	9.51
	5	3.65	4.46	3.72
	10	3.89	4.63	3.51
LMNN	2	4.88	17.75	8.60
	5	4.07	17.47	**3.26**
	10	4.11	10.32	**3.33**

On the handwritten character data set and on the Cohn-Kanade data set, the generalization performance of our classifiers is measured using 10-fold cross-validation. On the Arabic speech data set, we measured the classifier performances using the fixed division into training set (75% of the data) and test set (the remaining 25%) proposed by Hammami & Bedda (2010). On the mutagenicity data set, we used a fixed division into training set (90% of the data) and test set (the remaining 10%). Code that reproduces the results of our experiments is available from `http://homepage.tudelft.nl/19j49/fisher`.

4.3. Results

Below, we present the results of our experiments on the four data sets.

Character data set. The results of our experiments on the online handwritten character data are presented in Table 1. The table presents the average generalization errors over 10 folds for the three different representations and four different classifiers, using three different values[4] for the number of hidden states K. To the best to our knowledge, no previous results on the online handwritten character data set are presented

[4]The reader should note here that to obtain the log-likelihood representation, C models with K hidden states are trained; the other two representations only use a single model with K hidden states.

Table 2. Generalization errors (in %) on the Arabic speech data set for four different classifiers on three different embeddings. The table reports the generalization error on the fixed training/test division proposed by Hammami & Bedda (2010).

Classif.	K	Likelih.	Fisher	FKL
Bayes	2	14.23	–	–
	5	12.46	–	–
	10	14.46	–	–
Softmax	2	**8.86**	16.41	21.46
	5	8.14	10.46	10.09
	10	10.32	8.23	6.95
SVM	2	8.45	16.73	22.59
	5	**7.91**	9.41	10.36
	10	10.55	7.64	**6.91**
LMNN	2	10.27	16.28	25.32
	5	13.68	17.23	12.41
	10	11.00	36.77	7.23

Table 3. Generalization errors (in %) on the Cohn-Kanade data set for four different classifiers on three different embeddings. The table reports the mean generalization error over 10 folds.

Classif.	K	Likelih.	Fisher	FKL
Bayes	2	50.00	–	–
	5	55.63	–	–
	10	57.05	–	–
Softmax	2	28.13	30.63	**8.75**
	5	40.31	45.00	**10.63**
	10	45.31	65.00	**9.06**
SVM	2	36.56	54.69	10.94
	5	48.13	60.63	13.44
	10	42.19	77.19	14.06
LMNN	2	66.25	70.94	10.94
	5	66.56	66.56	15.94
	10	65.63	65.31	15.63

in the literature; this prevents us from comparing our results with those of other studies. From the results presented in Table 1, we make three main observations.

First, we observe that the log-likelihood representation outperforms the other two representations for small numbers of hidden states. This observation is presumably due to the fact that the log-likelihood representation is determined by C times more parameters for the same number of hidden states, because it trains a separate model for each of the C classes. This may provide the likelihood representation with more flexibility than the other two representations.

Second, we observe that the FKL representation has the potential of outperforming the other two representations. In particular, the lowest generalization error obtained using the FKL representation is 3.26%, whereas the likelihood and Fisher representations achieve errors of 3.86% and 4.46%, respectively. The strong performance of the FKL representations appears to be independent of the selected classifier.

Third, the results in the table reveal the relatively poor performance of the standard Fisher representation; in our experiments, classifiers trained on the Fisher representation are outperformed by FKL as well as by classifiers that are trained on per-class log-likelihoods. Presumably, this result is due to the fact that to obtain the Fisher representation, models are used that were trained to maximize the likelihood of the training data. As we argued in Section 2, this may lead to object representations that are suboptimal in terms of discrimination between classes.

Arabic spoken digits. The generalization errors achieved on the Arabic spoken digits data set are presented in Table 2. On the Arabic spoken digits data, again, the log-likelihood representation outperforms FKL when the number of hidden states is small. Curiously, on the Arabic spoken digits data, Fisher representations also lead to higher generalization accuracies than the FKL representation for small numbers of hidden states. So far, we have no good explanation for this result.

Having said that, the best performance on the Arabic spoken digits data is, again, obtained using the FKL representation: the log-likelihood representation achieves a lowest generalization error of 7.91%, the Fisher kernel representation achieves a lowest error of 7.64%, whereas the FKL representation achieves a lowest error of 6.91%. The performance of FKL on the Arabic spoken digits data is on par with the state-of-the-art performance on this data set of 6.88% (Hammami & Bedda, 2010).

Facial expression data. The results of the experiments on the Cohn-Kanade data set are presented in Table 3; we report generalization errors that are averaged over 10 folds. The results presented in Table 3 are in line with those on the previous two data sets: the FKL data representation leads to higher generalization accuracies than the Fisher kernel representation. In particular, the generalization errors achieved by the likelihood and Fisher kernel representations are 28.13% and 30.63%, respectively; the FKL representation attains a generalization error of 8.75%.

An interesting observation from Table 3 is the relatively poor performance of the log-likelihood representation. Presumably, this poor performance is the result of the nature of the facial expression data: because of the small changes in facial feature point locations in most expressions, very small variations in the features

Table 4. Generalization errors (in %) on the mutagenicity data set for four different classifiers on three different embeddings. The table reports the generalization error on a fixed 90% training/ 10% test division.

Classif.	K	Likelih.	Fisher	FKL
Bayes	2	51.96	–	–
	5	42.49	–	–
	10	45.73	–	–
Softmax	2	30.48	33.72	32.10
	5	33.49	33.03	**24.94**
	10	32.10	28.18	**22.17**
SVM	2	38.12	33.95	34.64
	5	36.72	33.72	34.41
	10	35.33	30.48	30.95
LMNN	2	40.42	33.72	**24.25**
	5	39.03	33.03	45.69
	10	43.08	36.72	33.03

determine which facial expression is present. These small variations presumably have a small effect on the likelihoods of the data under the models (the C models may model very similar distributions), which gives rise to the poor performance of the log-likelihood representation on the facial expression recognition task.

Mutagenicity data. In Table 4, we present the generalization errors of our classifiers on the mutagenicity data set. The reader should note that this data set contains structured objects which have arbitrary (loopy) underlying graphs, so not just time series.

The results on the mutagenicity data set are in line with those on the previous three data sets. In particular, the lowest error of 22.17% is obtained using an FKL object representation, whereas the log-likelihood and Fisher kernel representations give rise to lowest errors of 30.48% and 28.18%, respectively. By comparison, the state-of-the-art performance on this data set is 34.5% error (although this result was obtained using a somewhat different experimental setup; Riesen & Bunke (2008)). The results of the experiments on the mutagenicity data set suggest that the performance of FKL does generalize to models in which exact inference is intractable.

5. Discussion

Taken together, the results of our experiments reveal that, in discriminative tasks, training a model to minimize the nearest-neighbor error of a Fisher kernel embedding has the potential of outperforming embeddings of structured objects that are derived from models that maximize the likelihood of the data. Although our present experiments focus on the classification of graph-structured objects that can be appropriately modeled by the pairwise MRF model in Equation 1, the idea of training probabilistic generative models to minimize the nearest-neighbor error of an embedding of the data may be applied more generally. In particular, the log-likelihood gradient of any generative model may be plugged into Equation 3.

For instance, the log-likelihood gradients of probabilistic matrix factorization (Salakhutdinov & Mnih, 2008) may be plugged into Equation 3. When applied to a problem such as movie recommendation, the resulting technique may be able to use genre labels of movies to construct a movie embedding in which movies from different genres are better separated (as such a solution leads to log-likelihood gradients that have large between-class variation and small within-class variation). As another example, one may use the gradients of probabilistic latent semantic analysis (Hofmann, 1999) in Equation 3. The resulting technique would be able to embed documents in such a way, that documents with similar topic or genre labels are embedded close together (and documents with different labels are embedded far apart). We leave such extensions of FKL to future work.

6. Conclusion

We presented a technique, called Fisher kernel learning (FKL), that uses label information to improve the object embeddings that are used in Fisher kernels. The label information is exploited by learning the parameters of the underlying model in such a way as to minimize the nearest-neighbor error of the embedding. The results of our experiments with FKL (using pairwise MRFs as the underlying model) on four real-world data sets illustrate its potential.

Future work aims at investigating applications of FKL to other types of data, in particular, to user-movie ratings and to corpora of word-presence/tf-idf vectors. We also intend to explore unsupervised variants of FKL, for instance, by replacing the current metric-learning objective by a maximizing-variance objective (Weinberger et al., 2007). Such a variant may be used for, among others, the visualization of structured object similarities in two-dimensional scatter plots.

Acknowledgments

The author thanks Lawrence Saul, Fei Sha, David Tax, and Marco Loog for helpful discussions. Part of this work was performed while the author was at University of California, San Diego. This work was supported by NWO grant no. 680.50.0908, by EU-FP7 NoE on Social Signal Processing, and by NSF award no. 0812576.

References

Amari, S. Natural gradient works efficiently in learning. *Neural Computation*, 10(2):251–276, 1998.

Bicego, M., Pekalska, E., Tax, D.M.J., and Duin, R.P.W. Component-based discriminative classification for Hidden Markov Models. *Pattern Recognition*, 42(11):2637–2648, 2009.

Bunke, H. and Allermann, G. Inexact graph matching for structural pattern recognition. *Pattern Recognition*, 1 (4):245–253, 1983.

Campbell, W.M., Campbell, J.P., Reynolds, D.A., Singer, E., and Torres-Carrasquillo, P.A. Support vector machines for speaker and language recognition. *Computer Speech & Language*, 20(2–3):210–229, 2006.

Cootes, T.F., Edwards, G., and Taylor, C.J. Active appearance models. In *Proceedings of the European Conference on Computer Vision*, volume 2, pp. 484–498, 1998.

Ding, H. Querying and mining of time series data: experimental comparison of representations and distance measures. *Proceedings of the VLDB*, 1(2):1542–1552, 2008.

Eddy, S.R., Mitchison, G., and Durbin, R. Maximum discrimination hidden Markov models of sequence consensus. *Journal of Computational Biology*, 2(1):9–24, 1995.

Gärtner, T. A survey of kernels for structured data. *ACM SIGKDD Explorations Newsletter*, 5(1):49–58, 2003.

Globerson, A. and Roweis, S. Metric learning by collapsing classes. In *Advances in Neural Information Processing Systems*, volume 18, pp. 451–458, 2006.

Goldberger, J., Roweis, S., Hinton, G.E., and Salakhutdinov, R.R. Neighbourhood components analysis. In *Advances in Neural Information Processing Systems*, volume 17, pp. 513–520, 2005.

Hammami, N. and Bedda, M. Improved tree model for Arabic speech recognition. In *Proceedings of the* 3^{rd} *IEEE International Conference on Computer Science and Information Technology*, pp. 521–526, 2010.

Hofmann, T. Probabilistic latent semantic indexing. In *Proceedings of the* 22^{th} *Annual International SIGIR Conference*, pp. 50–57, 1999.

Jaakkola, T. and Haussler, D. Exploiting generative models in discriminative classifiers. In *Advances in Neural Information Processing Systems*, pp. 487–493, 1998.

Jaakkola, T., Diekhans, M., and Haussler, D. A discriminative framework for detecting remote protein homologies. *Journal of Computational Biology*, 7(1–2):95–114, 2000.

Jebara, T., Kondor, R., and Howard, A. Probability product kernels. *Journal of Machine Learning Research*, 5 (Jul):819–844, 2004.

Kim, M. and Pavlovic, V. Discriminative learning of mixture of bayesian network classifiers for sequence classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 268–275, 2006.

Kschischang, F.R., Frey, B.J., and Loeliger, H.-A. Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory*, 47(2):498–519, 2001.

Lucey, P., Cohn, J.F., Kanade, T., Saragih, J., Ambadar, Z., and Matthews, I. The extended Cohn-Kanade dataset (CK+): A complete dataset for action unit and emotion-specified expression. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 94–101, 2010.

Matthews, I. and Baker, S. Active appearance models revisited. *International Journal of Computer Vision*, 60 (2):135–164, 2004.

Mooij, J.M. libDAI: A free and open source C++ library for discrete approximate inference in graphical models. *Journal of Machine Learning Research*, 11(Aug):2169–2173, 2010.

Quattoni, A., Wang, S., Morency, L.-P., Collins, M., and Darrell, T. Hidden conditional random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(10), 2010.

Riesen, K. and Bunke, H. IAM graph database repository for graph based pattern recognition and machine learning. In *Lecture Notes in Computer Science*, volume 5342, pp. 287–297, 2008.

Sakoe, H. and Chiba, S. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 26 (1):43–49, 1978.

Salakhutdinov, R. and Mnih, A. Probabilistic matrix factorization. In *Advances in Neural Information Processing Systems*, volume 20, pp. 1257–1264, 2008.

Shawe-Taylor, J. and Christianini, N. *Kernel Methods for Pattern Analysis*. Cambridge University Press, Cambridge, UK, 2004.

Tsuda, K., Kawanabe, M., Rätsch, G., Sonnenborg, S., and Müller, K.-R. A new discriminative kernel from probabilistic models. *Neural Computation*, 14(10):2397–2414, 2002a.

Tsuda, K., Kin, T., and Asai, K. Marginalized kernels for biological sequences. *Bioinformatics*, 18:S268–S275, 2002b.

Weinberger, K.Q., Sha, F., Zhu, Q., and Saul, L.K. Graph Laplacian regularization for large-scale semidefinite programming. In *Advances in Neural Information Processing Systems*, volume 19, 2007.

Williams, B.H., Toussaint, M., and Storkey, A.J. Modelling motion primitives and their timing in biologically executed movements. In *Advances in Neural Information Processing Systems*, volume 20, pp. 1609–1616, 2008.

Xi, X., Keogh, E.J., Shelton, C.R., Wei, L., and Ratanamahatana, C.A. Fast time series classification using numerosity reduction. In *Proceedings of the International Conference on Machine Learning*, pp. 1033–1040, 2006.