Hybrid Generative-Discriminative Classification using Posterior Divergence

Xiong Li
Shanghai Jiao Tong University
Shanghai, 200240, China
lixiong@sjtu.edu.cn

Tai Sing Lee
Carnegie Mellon University
Pittsburgh, PA 15213, USA
tai@cs.cmu.edu

Yuncai Liu
Shanghai Jiao Tong University
Shanghai, 200240, China
whomliu@sjtu.edu.cn

Abstract

Integrating generative models and discriminative models in a hybrid scheme has shown some success in recognition tasks. In such scheme, generative models are used to derive feature maps for outputting a set of fixed length features that are used by discriminative models to perform classification. In this paper, we present a method, called posterior divergence, to derive feature maps from the log likelihood function implied in the incremental expectation-maximization algorithm. These feature maps evaluate a sample in three complementary measures: (1) how much the sample affects the model; (2) how well the sample fits the model; (3) how uncertain the fit is. We prove that the linear classification error rate using the outputs of the derived feature maps is at least as low as that of plug-in estimation. We present efficient algorithms for computing these feature maps for semi-supervised learning and supervised learning. We evaluate the proposed method on three typical applications, i.e. scene recognition, face and non-face classification and protein sequence analysis, and demonstrate improvements over related methods.

1. Introduction

Generative and discriminative models are two complimentary paradigms of machine learning. Generative models are particularly useful in dealing with missing data and discovering latent structures from given data in an unsupervised manner, situated somewhere between clustering and semi-supervised learning. They are also good at representing data such as images and variable-length sequences (e.g. natural language sentences and protein sequences) with fixed length features, for their flexibility. However, the classification performance of generative models using *plug-in estimation* (i.e. $\hat{y} = \text{sign}(P(y = +1 \mid x, \theta) - 1/2)$ is generally inferior to discriminative models which are more powerful in capturing decision boundaries among different classes and more widely used in recognition tasks. At present, several hybrid generative-discriminative schemes have been proposed to combine the strengths of these two classes of models in a number of applications, from scene classification [3], object recognition [5], speech recognition [19] to biological sequence analysis [6, 21], resulting in state-of-the-art performance.

These hybrid schemes sought to integrate the intra-class information from generative models and the complementary inter-class information from discriminative methods. Typically, a feature detector or a kernel similarity is derived from the given generative model. That is, given a learned model $P(\mathbf{x} \mid \theta)$, we find a fixed number of *feature maps* (or mapping functions) $\phi_i(\mathbf{x}, \theta) : \mathbf{x} \to \mathbb{R}$ for $i = 1, \cdots, K$. Then we obtain the *feature detector* $\Phi(\mathbf{x}, \theta) = (\phi_1(\mathbf{x}, \theta), \cdots, \phi_K(\mathbf{x}, \theta))^T$, and the *kernel similarity* $K(\mathbf{x}, \mathbf{x}'; \theta) = \Phi(\mathbf{x}, \theta)^T \Phi(\mathbf{x}', \theta)$. The resulting features here are not visual features in the normal sense (e.g. SIFT [13]) but are abstract ones with dimensions defined by the feature maps and the number of dimensions K determined by the generative model structure. There are roughly two classes of hybrid methods: parameter based methods and random variable based methods.

Parameter based methods were represented by Fisher kernel (FK) [7] and Tangent vector of posterior log-odds kernel (TK) [20]. These methods derive feature maps based on differential operation of the log likelihood function of generative models, i.e. $\phi_i(\mathbf{x}, \theta) = \nabla_{\theta_i} \log P(\mathbf{x} \mid \theta_i)$, and then construct kernel based on these features and the Fisher information matrix I: $K(\mathbf{x}, \mathbf{x}'; \theta) = \Phi(\mathbf{x}, \theta)^T \mathrm{I} \Phi(\mathbf{x}', \theta)$. As discussed in [7], embedding the kernel into the classifier is almost equivalent to using the feature maps directly in the classifier because I is close to identity. Thus, these kernels can effectively be treated as feature maps. These methods, however, greatly depend on the parametrization of the generative models. In the case that the number of free model parameters is less than the number of dimensions of samples, several samples may map to the same feature, resulting in an ambiguous and less discriminative representation.

Random variable based methods start from considerations in the free energy score space (FESS) [15]. These methods also seek to derive feature maps based on the log likelihood function of a model, as the parameter based

methods. But they focus on the random variables, rather than on the parameters in their derivation. The lower bound of log likelihood (see Equation 1), according to the random variables, is expanded and each resulting term becomes a feature map. The feature map measures how well a sample fits a random variable. This method overcomes the difficulty of the parameter based methods mentioned above, and could produce informative features even when the model is imperfect, or the parameters are less than the dimensions of the samples. However, these methods are still fragile because its feature maps may degenerate, particularly when some unorthodox EM algorithms are used. For example, some hidden variables in [18] are shared by all the samples and thus their distributions cannot be factorized using the samples. In this case, feature maps derived from variables using FESS could produce the same response for multiple samples and have no discriminative power. Section 5.2 provided details of one such example. Nevertheless, there might still be useful information that can be extracted from these random variables, using the method we now propose.

Here, we propose a new hybrid scheme that combine criteria implicit in the random variable based methods and in the parameter based methods. We motivate our approaches by three measures to capture more discriminative information in samples: (1) how much a sample affects the model; (2) how well a sample fits the model; (3) how uncertain the fitting is. The first measure, *posterior divergence*, assesses the change in model parameters brought on by the input sample $\mathbf{x}^c$, is also characteristic of the parameter based methods. The second and third measures are addressed in the inference step in the EM algorithm, i.e. during the inference of hidden variables conditioned on every sample, and are related to random variable based methods. We will show that the three measures can be derived from an unified formulation, and prove that the performance of proposed method is at least as good as that of plug-in estimation. Then, the method is evaluated on scene recognition with PLSA [4], face and non-face classification with MCVQ [18] and protein sequence analysis with HMM [16].

The remainder of this paper is organized as follows. We introduce the background and state the problem in Section 2. The formulation of the method is given in Section 3. We discuss the properties of the proposed method in Section 4. Section 5 presents three validation experiments. Section 6 draws a conclusion.

2. Background

Current strategies [7, 20, 15] to derive feature maps are based on the variational EM algorithm [9] that is developed for learning those generative models whose log likelihood functions are intractable to be integrated. It derives a tractable lower bound for the intractable likelihood function so that we can perform the learning and inference on the lower bound instead of the log likelihood.

For a generative model θ, let $\mathbf{x} \in \mathbb{R}^D$ be the observed random variable; $H = (h_1, \cdots, h_M)$ be the set of hidden random variables; i and m index samples and hidden variables respectively; $Q^c(h_m)$ denotes the approximate distribution of the posterior distribution $P(h_m \mid \mathbf{x}^c)$. The variational method derives a lower bound from Jensen's inequality to approximate the log likelihood:

$$\log P(\mathbf{x} \mid \theta) \geq -\mathbb{KL}(Q(H) \parallel P(\mathbf{x}, H)) = -\mathcal{F}(Q, \theta) \qquad (1)$$

where $\mathbb{KL}$ denotes Kullback-Leibler divergence, $\mathcal{F}$ denotes the variational free energy. $Q(H)$ could be factorized according to variables $Q(H) = \prod_m Q(h_m)$. $Q(h)$ can be further factorized according to samples $Q(h) = \prod_i Q^i(h)$ since samples are assumed to be i.i.d. Using these factorizations then:

$$\begin{aligned} \mathcal{F}(Q, \theta) &= \sum\nolimits_i \mathcal{F}(Q^i, \theta) \qquad (2) \\ &= \sum\nolimits_i E_{Q^i}[\log Q^i(H) - \log P(\mathbf{x}, H \mid \theta)] \end{aligned}$$

Substitute Equation 2 into Equation 1, then the log likelihood of a sample set is expressed as the summation of the sample log likelihood. So far we could perform EM algorithm on lower bound $-\mathcal{F}(Q, \theta)$ instead of the log likelihood $\log P(\mathbf{x} \mid \theta)$, by alternatively maximizing the lower bound of the sample set with respect to Q^i and θ.

On the other hand, the log likelihood function, i.e. the lower bound here, implies a group of measures on samples. Such measures (e.g. $E[Q(h_m \mid \mathbf{x}^i)]$) provide a probabilistic perspective to look at samples and to identify samples. For brevity, we do not distinguish measure and feature map in notation. On the basis of the lower bound, FK and TK derive feature maps using differential operation with respect to parameters $\{\nabla_{\theta_m} \log -\mathcal{F}(Q, \theta)\}_m$. FESS expands the low bound and uses the resulting terms as feature maps. However, these methods either directly or implicitly evaluate how much a sample affect the model, or how well a sample fits the model, but not both simultaneously, thus suffering from the problems discussed in Section 1.

3. Posterior Divergence

To overcome the degeneration issue, we propose to derive an alternative set of feature maps from the perspective of incremental EM algorithm [14]. The derived feature maps address all three measures.

3.1. Formulation

Different from regular EM algorithm that looks at all samples in each iteration, the incremental EM algorithm only looks at one or few selected samples to update the model in each iteration. Let $\mathbf{x}^c$ be the sample to be looked at the t-th iteration; $\mathcal{X} = (\mathbf{x}^1, \cdots, \mathbf{x}^N)$ be set of samples containing $\mathbf{x}^c$; $\mathcal{X}_{-c}$ be the resulting set of removing $\mathbf{x}^c$ from $\mathcal{X}$. i indexes samples and m indexes hidden variables. Let $P(\mathbf{x} \mid \theta)$

be the model estimated from the sample set $\mathcal{X}$ and $\{Q^i\}_i$ be the approximations of posterior distributions $\{P(H \mid \mathbf{x}^i, \theta)\}_i$ where $\mathbf{x}^i \in \mathcal{X}$; $P(\mathbf{x} \mid \theta_{-c})$ be the model estimated from the sample set $\mathcal{X}_{-c}$ and $\{Q^i_{-c}\}_i$ be the approximations of posterior distributions $\{P(H \mid \mathbf{x}^i, \theta_{-c})\}_i$ where $\mathbf{x}^i \in \mathcal{X}_{-c}$.

The E step of incremental EN algorithm computes the approximate distribution $Q^{c,t}$ of $P(H \mid \mathbf{x}^c)$, and M step combines $\{Q^{i,t-1}\}_{i \neq c}$ and $Q^{c,t}$ to update the model θ. Therefore the *implied log likelihood* of the input sample $\mathbf{x}^c$ in incremental EM algorithm could be written as the contribution of $\mathbf{x}^c$ to the log likelihood for the entire sample set:

$$\mathcal{L}(\mathbf{x}^c) = \sum_{i=1}^{N}[-\mathcal{F}(Q^i, \theta)] - \sum_{i \neq c}^{N}[-\mathcal{F}(Q^i_{-c}, \theta_{-c})] \tag{3}$$

This log likelihood encodes the contributions of the input sample $\mathbf{x}^c$ to the model (i.e. $\theta_{-c} \rightarrow \theta$) and the approximate distributions (i.e. $Q^i_{-c} \rightarrow Q^i$). Note that it differs from the previous log likelihood (i.e. lower bound $-\mathcal{F}(Q^c, \theta)$) derived by variational EM algorithm.

Substitute Equation 2 into Equation 3, we obtain the expansion of the implied log likelihood. To derive feature maps, we factorize the terms of the resulting expansion, i.e. $Q^i(H)$ and $P(\mathbf{x}, H \mid \theta)$, as follows:

$$Q^i(H) = \prod_{m=1}^{M} Q^i(h_m) \tag{4}$$

$$P(\mathbf{x}, H|\theta) = P(\mathbf{x} \mid \mathrm{pa}_x, \theta) \prod_{m=1}^{M} P(h_m \mid \mathrm{pa}_m, \theta) \tag{5}$$

where pa_x and pa_m are the parent variable sets of $\mathbf{x}$ and h_m respectively. pa_m will be null when h_m has no parent variables. Substitute Equation 5 and Equation 4 into Equation 2, and further substitute the resulting expression into Equation 3, and rearrange it according to random variables:

$$\begin{aligned}
\mathcal{L} = & \underbrace{[\sum_{i=1}^{N} E_{Q^i} \log P(\mathbf{x} \mid \mathrm{pa}_x, \theta) - \sum_{i \neq c}^{N} E_{Q^i_{-c}} \log P(\mathbf{x} \mid \mathrm{pa}_x, \theta_{-c})]}_{\mathbf{x}-crossentropy} \\
& + \underbrace{[\sum_{i=1}^{N} E_{Q^i} \log P(h_1 \mid \mathrm{pa}_1, \theta) - \sum_{i \neq c}^{N} E_{Q^i_{-c}} \log P(h_1 \mid \mathrm{pa}_1, \theta_{-c})]}_{h_1-crossentropy} \\
& - \underbrace{[\sum_{i=1}^{N} E_{Q^i} \log Q^i(h_1) - \sum_{i \neq c}^{N} E_{Q^i_{-c}} \log Q^i_{-c}(h_1)]}_{h_1-entropy} \\
& + \underbrace{\cdots}_{h_2 \cdots h_{M-1}} + \underbrace{\cdots}_{h_M-crossentropy} + \underbrace{\cdots}_{h_M-entropy}
\end{aligned} \tag{6}$$

where $\mathcal{L} \triangleq \mathcal{L}(\mathbf{x}^c)$. The terms are in the form of entropy or cross entropy functions, which measure the fitness of a sample to random variables and the uncertainty in the fitness.

Here we make an assumption to formulate a more interpretable expression. If the size of sample set $\mathcal{X}_{-c}$, i.e. n is relative large, the difference between Q^i and Q^i_{-c} is so little that we could assume that it could be ignored. Hence we have approximations for any sample $\mathbf{x}^i$ and variable h_m:

$$Q^i(h_m) \approx Q^i_{-c}(h_m) \tag{7}$$

Applying the approximation to Equation 6 and arranging the resulting expression, we have:

$$\begin{aligned}
\mathcal{L} \approx & \underbrace{[\sum_{i \neq c}^{N} E_{Q^i_{-c}} \log \frac{P(\mathbf{x} \mid \mathrm{pa}_x, \theta)}{P(\mathbf{x} \mid \mathrm{pa}_x, \theta_{-c})}}_{\phi^{\mathbf{x}}_{pd}} + \underbrace{E_{Q^c_{-c}} \log P(\mathbf{x} \mid \mathrm{pa}_x, \theta)]}_{\phi^{\mathbf{x}}_{fit}} \\
& + \underbrace{[\sum_{i \neq c}^{N} E_{Q^i_{-c}} \log \frac{P(h_1 \mid \mathrm{pa}_1, \theta)}{P(h_1 \mid \mathrm{pa}_1, \theta_{-c})}}_{\phi^{h_1}_{pd}} + \underbrace{E_{Q^c_{-c}} \log P(h_1 \mid \mathrm{pa}_1, \theta)}_{\phi^{h_1}_{fit}} \\
& - \underbrace{E_{Q^c_{-c}} \log Q^c_{-c}(h_1)]}_{\phi^{h_1}_{ent}} + \underbrace{\cdots}_{h_2 \cdots h_{M-1}} + [\underbrace{\cdots}_{\phi^{h_M}_{pd}} + \underbrace{\cdots}_{\phi^{h_M}_{fit}} - \underbrace{\cdots}_{\phi^{h_M}_{ent}}]
\end{aligned} \tag{8}$$

where $\phi^v_{pd}(\mathbf{x}^c)$ denotes the *posterior divergence* function which is the difference of cross-entropy and measures how much the sample $\mathbf{x}^c$ affects the posterior distribution of the random variable v (measure (1)), and $\phi^v_{fit}(\mathbf{x}^c)$ denotes the fitness function that measures how well the sample $\mathbf{x}^c$ fits the distribution of the random variable v (measure (2)), and $\phi^v_{ent}(\mathbf{x}^c)$ denotes entropy function that measures the uncertainty of an approximation distribution of the random variable v (measure (3)). We refer to the three components as 'PD-PD','PD-FIT','PD-ENT' respectively, and refer to all three components as 'PD' in the following section. For a generative model and the input sample $\mathbf{x}^c$, we have a set of feature maps:

$$\Phi^c : \mathbf{x}^c \rightarrow [\phi^{\mathbf{x}}_{pd}, \phi^{\mathbf{x}}_{fit}; \phi^{h_1}_{pd}, \phi^{h_1}_{fit}, \phi^{h_1}_{ent}; \cdots; \phi^{h_M}_{pd}, \phi^{h_M}_{fit}, \phi^{h_M}_{ent}]$$

Note that ϕ^c is specified for sample $\mathbf{x}^c$ because the approximate distribution Q^c only relates to the sample. Since the number of feature maps is determined by the model structure, the sample features given by a model share the same number of dimensions and could straightforwardly work with discriminative classifiers (e.g. SVM [22]).

3.2. Algorithms

This section focuses how to estimate the approximation distributions $\{Q^i_{-c}\}_i$, the prior model θ_{-c} and the posterior model θ and so that we could construct feature maps for a given sample $\mathbf{x}^c$ using Equation 8. Here we present two algorithms to treat semi-supervised learning (see example in Section 5.1) and supervised learning (see example in Section 5.2) where we assume the extracted features are used for supervised learning in this paper.

In the standard semi-supervised learning, a generative model is trained from unlabeled samples $\mathcal{X}_0$ and then used to extract the features of samples $\mathcal{X}$. The procedure is summarized in Algorithm 1. For the supervised learning, a generative model is trained from samples $\mathcal{X}$ and used to extract its features. We describe the procedure in Algorithm 2. Note that in both algorithms, the E step (estimate Q) and M step (estimate θ) have analytical solutions and need no iterations. For N input samples, the two algorithms run E step and M step for N rounds respectively.

Algorithm 1 For semi-supervised learning

Input: Sample set $\mathcal{X} = (\mathbf{x}^1, ..., \mathbf{x}^N)$
Given pre-trained θ_{-c} and $\{Q^i_{-c}\}_{i=1}^N$ from $\mathcal{X}_{-c}$
for $c = 1$ to N **do**
  $Q^c_{-c} \leftarrow \arg\max_{Q^c_{-c}} -\mathcal{F}(Q^c_{-c}, \theta_{-c})$
  $\theta \leftarrow \arg\max_\theta \sum_{i=1}^N E_{Q^i_{-c}} \log P(\mathbf{x} \mid H, \theta_{-c})$
  Construct Φ^c with θ, θ_{-c}, $\{Q^i_{-c}\}_{i=1}^N$ using Equation 8
end for
Output: feature map set $\{\Phi^i\}_{i=1}^N$

Algorithm 2 For supervised learning

Input: Sample set $\mathcal{X} = (\mathbf{x}^1, ..., \mathbf{x}^N)$
Estimate θ and $\{Q^i\}_{i=1}^N$ from $\mathcal{X}$ using variational EM
Use approximation $Q^i_{-c} \leftarrow Q^i$
for $c = 1$ to N **do**
  $\theta_{-c} \leftarrow \arg\max_{\theta_{-c}} \sum_{i \neq c} E_{Q^i_{-c}} \log P(\mathbf{x} \mid H, \theta_{-c})$
  Construct Φ^c with θ, θ_{-c}, $\{Q^i_{-c}\}_{i=1}^N$ using Equation 8
end for
Output: feature map set $\{\Phi^i\}_{i=1}^N$

Though posterior divergence is developed on incremental EM algorithm, it could work with kinds of EM algorithms, such as variational EM algorithm [9], incremental EM algorithm [14] and Monte Carlo EM algorithm [23] etc. Further, for inference and learning methods designed for specific generative models, we always could estimate Q^i_{-c} at the inference step and θ, θ_{-c} at the learning step. Hidden Markov Models with Baum-Welch algorithm [1] will be presented in Section 5.2 as an illustration example.

4. Properties

This section compares the error rate of posterior divergence with that of plug-in estimation, and investigates its relationship to previous works [7, 15].

4.1. Error rate comparison with plug-in estimation

The feature maps of posterior divergence define a feature space whose number of dimensions is fixed for a given generative model and hence could straightforwardly work with discriminative classifiers. The following part will show that the features given by PD working with linear classifier perform at least as good as plug-in estimation.

Let $\mathbf{x} \in \mathcal{X}$ be the input sample and $y(\mathbf{x}) \in \{-1, +1\}$ be its label. Assuming the sample set $\mathcal{X}$ is modeled by distribution $P(\mathbf{x} \mid \theta)$. In the plug-in estimation, the model parameter θ_{+1} is learned from samples of a single class labeled as $+1$ and is a consistent estimation of true parameter θ^*_{+1}. For an input sample $\mathbf{x}^c$, it is assigned to $+1$ for $P(\mathbf{x}^c \mid \theta_{+1}) > 1/2$ and -1 for otherwise. Then we consider the linear classifier which the derived features work on. A linear classifier takes the form of $\mathbf{w}^T\Phi(\mathbf{x}) + b$ where $\mathbf{w} \in (\mathbf{w} \mid \mathbf{w} \in \mathbb{R}^d, \|\mathbf{w}\| = 1)$, $b \in \mathbb{R}$ and its classification error can be shown as [20]:

$$R(\Phi) = \min_{\mathbf{w},b} E_{\mathbf{x},y} \Psi[-y(\mathbf{w}^T\Phi(\mathbf{x}) + b)] \tag{9}$$

where $\Phi(\cdot)$ is the feature map; $E_{\mathbf{x},y}$ denotes the expectation with respect to the true distribution $P(\mathbf{x}, y \mid \theta^*)$; and $\Psi[a]$ is an indicator function that takes 1 for $a > 0$ and 0 for others.

Using the error rate measure defined in Equation 9, we can show that posterior divergence, when used with linear classifier, is superior to plug-in estimation, as shown in the following proposition.

Proposition 4.1. *In the posterior divergence feature space derived from a trained generative model, the error rate of a linear classifier is at least as low as that of the plug-in estimation:*

$$R(\Phi) \leq E_{\mathbf{x},y} \Psi[-y(P(y = +1 \mid \mathbf{x}, \hat{\theta}) - \frac{1}{2})] = R(\lambda)$$

Proof. $\forall\, \mathbf{w} \in R^N$, $b \in R$, they always satisfies an inequality $R(\Phi) = \min_{\mathbf{w},b} E_{\mathbf{x},y} \Psi[-y(\mathbf{w}^T\Phi(\mathbf{x}) + b)] \leq E_{\mathbf{x},y} \Psi(\mathbf{w}, b)$. With the inequality and let $\mathbf{w} = \mathbf{1}, b = -\log 1/2$, we have:

$$\begin{aligned}
R(\Phi) &= \min_{\mathbf{w},b} E_{\mathbf{x},y} \Psi[-y(\mathbf{w}^T\Phi(\mathbf{x}) + b)] \\
&\leq E_{\mathbf{x},y} \Psi[-y(\mathbf{1}^T\Phi(\mathbf{x}) - \frac{1}{2})] \\
&= E_{\mathbf{x},y} \Psi[-y(\log P(\mathbf{x} \mid \hat{\theta}_{+1}) - \log \frac{1}{2})] \\
&= E_{\mathbf{x},y} \Psi[-y(\log P(y = +1 \mid \mathbf{x}, \hat{\theta}) - \log \frac{1}{2})] \\
&= E_{\mathbf{x},y} \Psi[-y(P(y = +1 \mid \mathbf{x}, \hat{\theta}) - \frac{1}{2})] = R(\lambda)
\end{aligned}$$

The last equality holds because log is an increasing function while $\Psi[\cdot]$ is an indicator function. □

For some generative models whose the log likelihood is intractable, both posterior divergence and plug-in estimation work on the variational approximation of the log likelihood, hence the theorem holds. When models are tractable, posterior divergence could be straightforwardly extended, and the above proposition and proof still holds.

For previous methods [7, 20, 15] that work on the lower bound of the log likelihood, it always has $\mathbf{w}, b$ to satisfy $\mathbf{w}^T\Phi(\mathbf{x}) + b = -\mathcal{F} \approx \log P(\mathbf{x})$ where $-\mathcal{F}$ is the lower bound of $\log P(\mathbf{x})$. Let $\mathbf{w} = \theta_s - \theta, b = \log P(\mathbf{x}\,|\,\theta) - \log 1/2$ for FK, $\mathbf{w} = [1, (\theta_+ - \theta)^T, \mathbf{0}]^T, b = -\log 1/2$ for TK, and $\mathbf{w} = \mathbf{1}, b = -\log 1/2$ for FESS, then we could validate that they perform at least as well as plug-in estimation.

4.2. Relationship to previous methods

It can be shown that if FESS [15] uses the approximation of Equation 7 and expands the lower bound according to random variables, its resulting feature maps are equivalent to PD-FT and PD-ENT of posterior divergence. A main difference between the two methods is PD-PD that encodes characteristic information of samples. Again, the posterior divergence factorizes log likelihood according to random variables and results an appropriate number of dimensions, while FESS may produces a high-dimensional feature space but with trivial and less informative dimensions.

The proposed method is also related to FK [7]. Working on the variational lower bound and using Taylor expansion, the feature map of FK could be formulated as:

$$\nabla_{\theta_j} \log P(\mathbf{x}|\theta) \approx \nabla_{\theta_j}(-\mathcal{F}(Q,\theta)) = \nabla_{\theta_j} E_Q \log(v_j|\,\mathrm{pa}_j, \theta_j)$$

where v_j is the set of observed or hidden variables parameterized by θ_j. On the other hand, we can linearize the posterior divergence function using Taylor expansion like FK:

$$\begin{aligned}\phi_{pd}^{v_j} &= \sum_{i\neq c}^{N} E_{Q^i_{-c}}[\log P(v_j\,|\,\mathrm{pa}_j, \theta_j) - \log P(v_j\,|\,\mathrm{pa}_j, \theta_{j,-c})] \\ &\approx \sum_{i\neq c}^{N} (\theta_j - \theta_{j,-c}) \cdot \nabla_{\theta_j} E_{Q^i_{-c}} \log P(v_j\,|\,\mathrm{pa}_j, \theta_j)\end{aligned}$$

Note the right term is the feature map derived by FK. It suggests that the posterior divergence function $\phi_{pd}^{v_j}$ is a linear combination of FK functions on samples.

5. Experiments

We evaluate the proposed approach on three typical applications of generative models: scene recognition, face and non-face recognition, and protein sequence analysis. In these experiments, FK [7], TK [20] and FESS [15] are used for comparison. These methods are used as model-dependent feature extractors whose outputs are delivered to the linear SVM [22] for classification. We ignore the plug-in estimation for comparison purpose as its inferiority to above methods that has been theoretically proved in Proposition 4.1 and experimentally validated in previous works [7, 15, 20].

5.1. Scene recognition

Several generative models (e.g. PLSA [4] and LDA [2]) have been used in this problem and shown some attractive characteristics (e.g. discovering topics unsupervisedly). Here we use PLSA model to learn the feature maps because it is slightly superior to LDA in scene recognition [3]. The output features are delivered to SVM for classification.

The CVCL scene database[1] is used to test all methods. It is composed of 4 typical natural scenes (coast, open country, forest and mountain) and 4 urban scenes (highway, street, inside city and tall building). We treat the scene recognition task as 8 two-class problems, each of which classifies a scene from other 7 ones.

For each image, we extract 200 SIFT descriptors [13] from 12×12 squares located by the DOG interest point detector. The number of interest points for each image is fixed through adaptively adjusting the threshold of DOG. With a code book formed from all descriptors by clustering, descriptors are quantized to visual words and then each image is further represented by its word histogram.

PLSA [4] is used to model the relationship of visual words and scenes. Let random variables w, z and d denote the term, topic and image respectively, and $m(w,d)$ denotes the number of term w in image d. The joint distribution of PLSA is $P(w,z,d) = [P(w\,|\,z)P(d\,|\,z)P(z)]^{m(w,d)}$, and its free energy is given by:

$$\begin{aligned}\mathcal{F} \;=\; & \sum_{d,w} m(d,w) \sum_{z} Q(z\,|\,d,w)[\log Q(z\,|\,d,w) \\ & - \log P(d\,|\,z)P(w\,|\,z)P(z)]\end{aligned}$$

where Q is the approximation distribution. With this expression, one could obtain FESS by directly expanding $\mathcal{F}$ according to the terms in the square bracket. The posterior divergence feature map for the input document d^c, as formulated in Equation 8, can be shown in following form:

$$\begin{aligned}\phi_w^c &: \sum_{i\neq c} E_{mQ^i_{-c}} \log \frac{P(w\,|\,z,\theta)}{P(w\,|\,z,\theta_{-c})},\; E_{mQ^c_{-c}} \log P(w\,|\,z,\theta) \\ \phi_d^c &: \sum_{i\neq c} E_{mQ^i_{-c}} \log \frac{P(d^c\,|\,z,\theta)}{P(d^c\,|\,z,\theta_{-c})},\; E_{mQ^c_{-c}} \log P(d^c\,|\,z,\theta) \\ \phi_z^c &: \sum_{i\neq c}\sum_{w} mQ^i_{-c} \log \frac{P(z\,|\,\theta)}{P(z\,|\,\theta_{-c})},\; \sum_{w} mQ^c_{-c} \log P(z\,|\,\theta), \\ & \sum_{w} -mQ^c_{-c} \log Q^c_{-c}\end{aligned}$$

where mQ^c is not a real distribution but E_{mQ^i} takes the expectation form for brevity. If the number of terms and topics are K, M respectively, the posterior divergence will have $2\times K + 2 + 3\times M$ feature maps.

[1] http://cvcl.mit.edu/database.htm

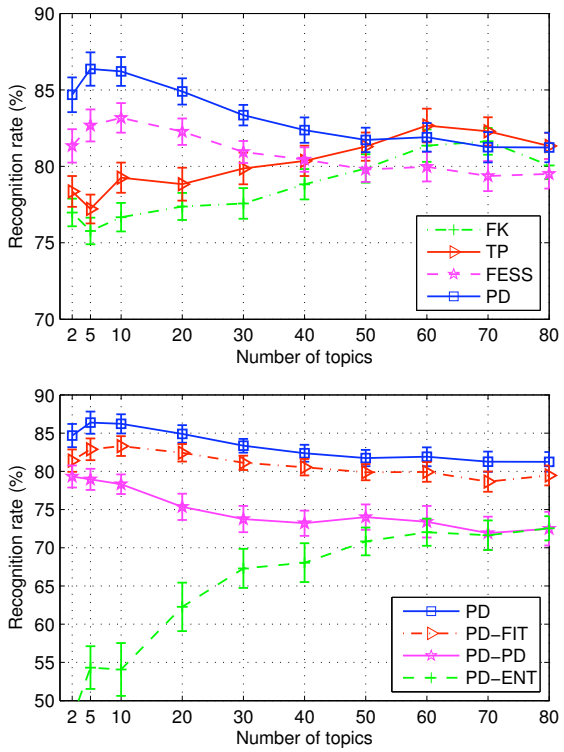

Figure 1. Performance comparisons on the PLSA model for scene recognition. The number of topics is variable. Four methods (Top) and the three components of PD (Bottom) are evaluated.

For each round of test, 30% positive samples are randomly chosen to form training set and another 30% to form testing set. Same number of samples are for the negative category. We test FK, TK, FESS and PD as well as its components, each for 20 rounds, and report the average results in Figure 1. As shown in the top figure, PD outperforms other methods when the topic number $K \leq 60$, and takes the peak performance of all methods when $K = 5$. We also found that PD and FESS share similar trend that works well for small K and then decreases along it, while FK and TK generally follow an opposite trend. These observations indicate that the two classes of methods have distinct performance trends, and that the performance of PD in this case is closer to the random variable based methods. The bottom figure presents the comparison of three components of PD, where PD-FIT outperforms the other two components and is the key determinant of the trend of PD, confirming that it is similar to FESS in this case. The other two components likely capture some non-redundant information as they help to improve the overall performance.

5.2. Face and non-face classification

To validate the effectiveness of posterior divergence in unorthodox EM algorithms, we uses MCVQ [18] for face and non-face classification in the semi-supervised manner. MCVQ is a generative model developed for learning parts-based representation. This model is especially suited for face representation for it works well on registered data. Here we use the CBCL face database[2] for experiment. It contains 2429 registered faces and all are in form of 19×19 gray images. The CBCL database also has number of non-face images that could be used as negative samples in test.

We learn a MCVQ model from the face database and use it to construct feature extractors. In order to learn a better representation, smoothness and symmetry priors are imposed using the technique of [11]. Let part number $K = 6$ and the state number $J = 10$. Then with the learned model, we are able to construct feature maps for FK, TK, FESS and PD. Here we present some feature maps of PD for demonstration. As shown in [18], the variational free energy of MCVQ is given by:

$$\mathcal{F}(Q,\theta) = E_Q[\sum_{d,k} r_{dk} \log \frac{g_{dk}}{a_{dk}} + \sum_{k,j} s_{kj} \log \frac{m_{kj}}{b_{kj}} - \sum_{d,k,j} r_{dk} s_{kj} \log \mathcal{N}(x_d)]$$

Since the parameter g_{dk} is shared by all samples in MCVQ, i.e. $E_{Q^c}[r^c_{dk}] = g_{dk}$ for any sample c, we could write the r_{dk} associated feature maps $\phi_{r_{dk}}$ as:

$$\phi^c_{r_{dk}} : \sum_{i \neq c}^{N} g_{dk,-c} \log \frac{a_{dk}}{a_{dk,-c}},\ \log a_{dk}^{g_{dk,-c}},\ \log g_{dk,-c}^{g_{dk,-c}}$$

where $a_{dk,-c}$ is the parameter of the previous model θ_{-c}. Note that the two functions on the right are independent with input sample and degenerate to constant. We can validate that in FESS all feature maps $\phi_{r_{dk}}$ suffer from this degeneration. In contrast, posterior divergence still works in this case for the first function.

With the trained MCVQ model, we extract the features of 400 face images and 400 non-face images of CBCL database using Algorithm 1. Of these 100 faces and 100 non-faces are randomly selected as training set and the rest selected as test set for the linear SVM in each test round. We report the average results of 20 round of tests in Figure 2, with different numbers of states J. The top figure shows that, in these configurations, all four methods share similar trends in performance, but our PD method outperforms the other three methods. The bottom figure shows that the PD-PD component outperforms the PD-FIT component in this case. The PD-ENT component shows the poor performance. This illustrates that PD-PD in our method can still extract useful discriminative features based on how much a sample affects the model, even when the FESS-like PD-FIT and PD-ENT components degenerate on random variables r_{dk}, i.e. $E_{Q^c}[r^c_{dk}] = g_{dk}$.

[2] http://cbcl.mit.edu/software-datasets/

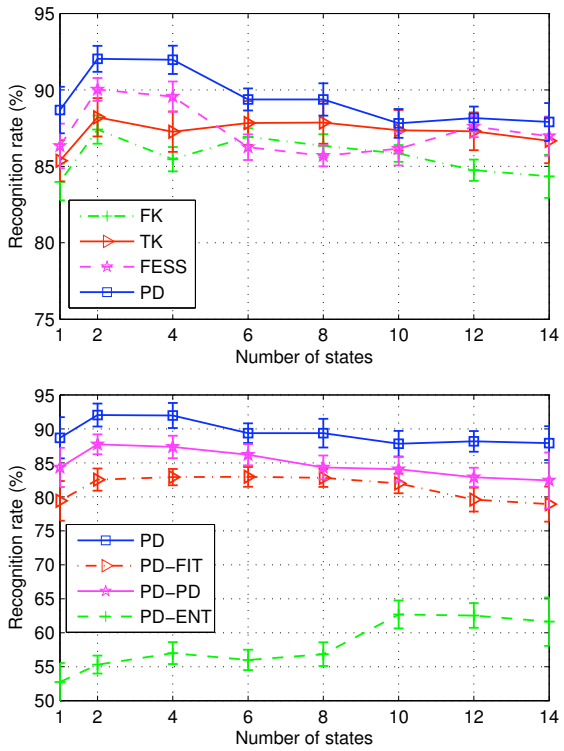

Figure 2. Performance comparisons of face and non-face classification. The number of states is variable. Four methods (Top) and the three components of PD (Bottom) are evaluated.

5.3. Remote homology recognition

In this experiment, we consider remote homology recognition that assigns protein sequences into classes defined in the SCOP (1.53)[3] taxonomy tree. The protein sequences data is obtained form ASTRAL database[4] with the E-value threshold of 10^{-25} to reduce similar sequences. All 4352 protein sequences are hierarchically labeled according to SCOP specification, which results 7 classes, 509 folds, 801 superfamilies and 1294 families. In this experiment, 804, 950, 694, 737, 54, 121 and 992 sequences of 7 classes respectively are available. Following [20], we use the first 4 classes for validation so that 6 two-class problems are formed. For each two-class problem, 30% samples are randomly selected for training and 40% for testing.

Here we employ hidden Markov models (HMM) [16] for protein classification because of its ability to model variable length sequences and its state-of-the-art performance. Let random binary vector $\mathbf{q}^t_{1\times M}$ and $\mathbf{y}^t_{1\times N}$ indicate the hidden state (M probable states) and output state (N probable states) at time t, parameters $\pi, A_{M\times M}, B_{M\times N}$ be the initial probability, state changing probability and output probability. The Baum-Welch algorithm [1] is used to estimate model parameters $\theta = (\pi, A, B)$. The free energy function of HMM is given by:

$$\mathcal{F}(Q,\theta) = E_Q[\sum_{i=1}^{M} q_i^0 \log\frac{\tau_i}{\pi_i} + \sum_{t=0}^{T_c-1}\sum_{i,j=1}^{M} q_i^t q_j^{t+1}\log\frac{g_{ij}}{a_{ij}} - \sum_{t=0}^{T_c}\sum_{i,j=1}^{M,N} q_i^t y_j^t \log b_{ij}]$$

where $q_i q_j$ and $q_i y_j$ could be viewed as two set of variables. Based on the model θ, we estimate the approximate distributions $\{Q^i(q_i^0, q_i q_j, q_i y_j \,|\, \tau, G)\}_i$ through maximizing the variational lower bound $-\mathcal{F}(Q,\theta)$ with respect to Q^i and then θ_{-c} with respect to θ. Then the feature maps can be derived through Equation 8. For example:

$$\phi^c_{q_i q_j} : \sum_{k\neq c} g^k_{ij,-c}\log\frac{a_{ij}}{a_{ij,-c}},\ g^c_{ij,-c}\log a_{ij},\ g^c_{ij,-c}\log g^c_{ij,-c}$$

$$\phi^c_{q_i y_j} : \sum_{k\neq c} h^k_{ij,-c}\log\frac{b_{ij}}{b_{ij,-c}},\ h^c_{ij,-c}\log b_{ij,-c}$$

Note that the difference between FK and TK is that FK takes single model but TK takes two models of samples classes into account, although their feature maps share similar form (differential operator). Feature vectors of both FK and TK are normalized to 1, which will improve the performance to some extent. As for FESS and PD, it is worth noting that the length of feature vector depends on how to expand the log likelihood into feature maps. In order to get features (or feature maps) with fixed length from HMM, we use a standard approach [15] that normalizes the likelihood by the sequence length.

For each two-class problem, we perform the experiment on randomly selected training and test sets for 20 rounds. The average recognition rates are reported in Table 1. We found that PD outperforms other methods on most data sets except for the set '2-3'. Even for set '2-3', PD's performance is very close to the top performance. In particular, the fitness component derived from feature maps PD-FIT share approximate performance with FESS for their similar definition, which has been previously stated in Section 3.1.

6. Conclusions

In the paper, we present a method to construct feature maps from generative models, so that one can learn feature spaces using Bayesian statistical methods, hereby bridging generative and discriminative models. Feature maps based on the posterior divergence of of the log likelihood function implied in the incremental EM algorithm are found to capture discriminative information that are more complete and robust than existing parameter based or random variable based methods. The three measures can each be related to FK and FESS respectively. Our method is able to

[3] http://scop.mrc-lmb.cam.ac.uk/scop/
[4] http://astral.berkeley.edu/

Feature	1 - 2	1 - 3	1 - 4	2 - 3	2 - 4	3 - 4
FK	81.52	82.23	72.30	83.61	68.88	70.36
TK	82.73	82.18	73.94	**83.89**	70.13	71.08
FESS	84.39	79.60	74.48	81.17	69.06	69.31
PD	**87.54**	**83.60**	**76.78**	83.02	**73.34**	**74.45**
PD-PD	84.54	77.61	72.61	79.04	65.88	64.41
PD-FIT	84.34	79.70	74.63	81.18	68.84	69.87
PD-ENT	75.59	71.18	65.91	70.03	59.37	58.76

Table 1. Recognition rates (%) of seven kinds of features. The data name such as '1-2' indicates that classes 1 and 2 are specified as the positive and negative categories respectively.

work with generalized EM algorithm, unorthodox EM algorithm, Monte Carlo EM algorithm and specifically designed learning algorithms on a variety of generative models with an efficient computation scheme. The method depends on adopted generative models and the approximation of posterior distribution as Equation 1. Therefore, it requires generative models themselves being able to model the given data and that they converge to a local maximum.

Beyond the three applications in the paper, the proposed method should be easily adoptable for other computer vision or pattern analysis tasks as long as the data could be modeled by some generative models. There are other attempts to integrate generative and discriminative models. For example, [8, 17, 10] use generative models as priors over discriminative models and [24, 12] learn generative models with the help of discriminative constraints. These methods are theoretically distinct from our method (as well as FESS and FK/TK methods). It would however be interesting to compare their performance with ours in a variety of classification applications.

Acknowledgment

This work was supported by National Basic Research Program of China 2011CB302203, NSFC 60833009 and 60975012 and Microsoft Research Asia Fellowship. Lee is supported by NSF CISE 0713206, AFOSR FA9550-091-0678 and Pennsylvania Department of Health through the commonwealth university research enhancement program.

References

[1] L. Baum, T. Petrie, G. Soules, and N. Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *The Annals of Mathematical Statistics*, 41(1):164–171, 1970.

[2] D. Blei, A. Ng, and M. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[3] A. Cristani, U. Castellani, V. Murino, and N. Jojic. A hybrid generative/discriminative classification framework based on free energy terms. In *ICCV*, 2009.

[4] T. Hofmann. Probabilistic latent semantic analysis. In *UAI*, pages 289–296, 1999.

[5] A. Holub, M. Welling, and P. Perona. Hybrid generative-discriminative visual categorization. *International Journal of Computer Vision*, 77(1):239–258, 2008.

[6] T. Jaakkola, M. Diekhans, and D. Haussler. Using the Fisher kernel method to detect remote protein homologies. In *International Conference on Intelligent Systems for Molecular Biology*, pages 149–158, 1999.

[7] T. Jaakkola and D. Haussler. Exploiting generative models in discriminative classifiers. In *NIPS*, pages 487–493, 1999.

[8] T. Jaakkola, M. Meila, and T. Jebara. Maximum entropy discrimination. In *NIPS*, 1999.

[9] M. Jordan, Z. Ghahramani, J. T., and S. L. Introduction to variational methods for graphical models. *Machine Learning*, 37:183–233, 1999.

[10] J. Lasserre, C. Bishop, and T. Minka. Principled hybrids of generative and discriminative models. In *CVPR*, volume 1, pages 87–94. IEEE, 2006.

[11] X. Li, L. Wang, H. Liu, and Y. Liu. Learning parts-based representation for face transition. In *ACM Multimedia*, 2010.

[12] X. Li, X. Zhao, Y. Fu, and Y. Liu. Bimodal gender recognition from face and fingerprint. In *CVPR*, 2010.

[13] D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[14] R. Neal and G. Hinton. A view of the EM algorithm that justifies incremental, sparse, and other variants. *Learning in Graphical Models*, 89:355–368, 1998.

[15] A. Perina, M. Cristani, U. Castellani, V. Murino, and N. Jojic. Free energy score space. In *NIPS*, pages 1428–1436, 2009.

[16] L. Rabiner. A tutorial on hidden Markov models and selected applications inspeech recognition. *Proceeding of the IEEE*, 77(2):257–286, 1989.

[17] R. Raina, Y. Shen, A. Ng, and A. McCallum. Classification with hybrid generative/discriminative models. In *NIPS*, volume 16, 2004.

[18] D. Ross and R. Zemel. Multiple cause vector quantization. In *NIPS*, pages 1041–1048, 2003.

[19] N. Smith and M. Gales. Speech recognition using SVMs. In *NIPS*, volume 25, 2002.

[20] K. Tsuda, M. Kawanabe, G. Ratsch, S. Sonnenburg, and K. Muller. A new discriminative kernel from probabilistic models. *Neural Computation*, 14(10):2397–2414, 2002.

[21] K. Tsuda, T. Kin, and K. Asai. Marginalized kernels for biological sequences. *Bioinformatics*, 18(Suppl 1):S268, 2002.

[22] V. Vapnik. *The nature of statistical learning theory*. Springer Verlag, 2000.

[23] G. Wei and M. Tanner. A Monte Carlo implementation of the EM algorithm and the poor man's data augmentation algorithms. *Journal of the American Statistical Association*, 85(411):699–704, 1990.

[24] J. Zhu, A. Ahmed, and E. Xing. Maximum Margin Supervised Topic Models for Regression and Classification. In *ICML*, volume 382. ACM, 2009.