2006 International Joint Conference on Neural Networks
Sheraton Vancouver Wall Centre Hotel, Vancouver, BC, Canada
July 16-21, 2006

Gender Classification with Bayesian Kernel Methods

Hyun-Chul Kim, Daijin Kim, *Senior Member, IEEE*, Zoubin Ghahramani,
Sung Yang Bang, *Senior Member, IEEE*

***Abstract*— We consider the gender classification task of discriminating between images of faces of men and women from face images. In appearance-based approaches, the initial images are preprocessed (e.g. normalized) and input into classifiers. Recently, SVMs which are popular kernel classifiers have been applied to gender classification and have shown excellent performance. We propose to use one of Bayesian kernel methods which is Gaussian Process Classifiers (GPCs) for gender classification. The main advantage of Bayesian kernel methods such as GPCs over SVMs is that they determine the hyperparameters of the kernel based on Bayesian model selection criterion. Our results show that GPCs outperformed SVMs with cross validation.**

I. Introduction

The face is a characteristic feature of human beings which contains identity and emotion. It is possible to identify a person and her/his characteristics such as emotion (or expression) and gender from her/his face. Recognizing human gender is important since lots of social interactions and services depend on the gender. People respond differently according to gender. Human computer interaction system can be more user-friendly and more human-like when it considers the user's gender.

There are two main approaches for gender classification. The first approach is the appearance-based approach which uses a whole face image. [1] reduced the dimension of whole face images by autoencoder network and classified gender based on the reduced input features. [2] used a 2-layer neural network (called SexNet) without dimensionality reduction. [3] used a neural network and showed that even very low resolution image such as 8x8 can be used for gender classification. [4] used the mixture of experts with ensembles of RBF networks and a decision tree as a gating network. [5] showed that SVMs worked better than other classifiers such as ensemble of RBF networks, classical RBF networks, Fisher linear discriminant, nearest neighbor etc. [6] extracted wholistic features by ICA and classified it with LDA. [7] used the exploratory basis pursuit classification which is a sparse kernel classifier .

The second approach is the geometrical feature based approach. [8] extracted point-to-point distances from 73 points on face images and used discriminant analysis as a classifier. [9] extracted 16 geometric features such as eyebrow thickness and pupil-to-eyebrow distance and used HyperBF networks as a classifier.

As mentioned above, the appearance-based approach with SVM showed excellent performance [5]. In their experiments the Guassian kernel worked better than linear or polynomial kernels. They did not mention how to set the hyperparameters[1] for Gaussian kernel which have an influence on performancem, but just showed the test results with several different hyperparameters. Learning the hyperparameters should be included in the training process. A standard way to determine the hyperparameters is by cross validation. Alternatively we could use Bayesian kernel classifiers such as Gaussian process classifiers which automatically incorporate method to determine the hyperparameters. In this paper we propose to use Gaussian process classifiers (GPCs) for appearance-based gender classification.

GPCs are a Bayesian kernel classifier derived from Gaussian process priors over functions which were developed originally for regression [10], [11], [12], [13]. In classification, the target values are discrete class labels. To use Gaussian processes for binary classification, the Gaussian process regression model can be modified so that the sign of the continuous latent function it outputs determines the class label. Observing the class label at some data point constrains the function value to be positive or negative at that point, but leaves it otherwise unknown. To compute predictive quantities of interest we therefore need to integrate over the possible unknown values of this function at the data points.

Exact evaluation of this integral is computationally intractable. However, several successful methods have been proposed for approximately integrating over the latent function values, such as the Laplace approximation [12], Markov Chain Monte Carlo [11], and variational approximations [13]. Opper and Winther (2000) used the TAP approach originally proposed in statistical physics of disordered systems to integrate over the latent values [14]. The TAP approach for this model is equivalent to the more general Expectation Propagation (EP) algorithm for approximate inference [15]. The EM-EP algorithm has been proposed to learn the hyperparameters based on EP [16]. GPCs with the hyperparameters obtained by the EM-EP algorithm have shown better performance than SVMs which had the hyperparameters set by cross validation, on most of data sets tested. In many cases the hyperparameters determined by the EM-EP algorithm

Hyun-Chul Kim is with the Department of Industrial and Management Engineering, POSTECH, Pohang, 790-784, South Korea (email: grass@postech.ac.kr).

Daijin Kim is with the Department of Computer Science and Engineering, POSTECH, Pohang, 790-784, South Korea (email: dkim@postech.ac.kr).

Zoubin Ghahramani is with the Department of Engineering, University of Cambridge, UK (email: zoubin@gatsby.ucl.ac.uk).

Sung Yang Bang is with the Department of Computer Science and Engineering, POSTECH, Pohang, 790-784, South Korea (email: sybang@postech.ac.kr).

[1]Hyperparameters control properties of the kernel and the amount of classification noise

0-7803-9490-9/06/$20.00/©2006 IEEE

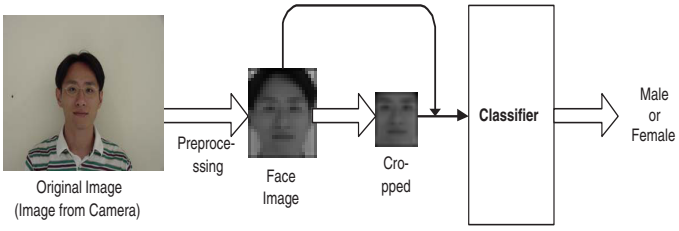


Fig. 1. The process of appearance-based gender classification

were more suitable for SVMs than the ones determined by cross validation technique. In this paper we use the EM-EP algorithm to learn Gaussian process classifiers for gender classification. We expect that GPCs with the EM-EP algorithm work better than SVMs with the cross validation and provide better hyperparameters for the kernels of SVMs.

The paper is organized as follows. Section 2 introduces appearance-based gender classification. In Section 3, we introduce Gaussian process classification. In section 4, we describe the EP method and the EM-EP algorithm for Gaussian process classification. In section 5, we show experimental results on the PF01 database and compared with other classification methods including SVMs. In section 6, we draw conclusions and remark on future work.

II. Appearance-based Gender Classification

The appearance-based approach to gender classification discriminates between male and female classes from face images without first explicitly extracting any geometrical features. A typical way to do this is to train a classifier with training images and to classify new images by the trained classifier. Face images should be well-aligned so that facial features are in the same positions. Since gender classification is a two-class classification problem, any kind of binary classifier can be deployed.

Figure 1 shows the process of appearance-based gender classification. Assume that a classifier has been already trained with some images in advance. The whole process of gender classification can be explained by the following. First, images are captured. Then, the captured images are preprocessed by face detection and facial feature extraction algorithms and cropped by an appropriate cropping technique. The preprocessed face images can include a whole outline of faces with hair or can include only inner face parts with only facial features. Then, the preprocessed image is applied to the classifier and the classifier determines the gender of the input image.

The appearance-based approach has two main advantages. First, it preserves appearance of face images which can be considered to be naive features. It is difficult to determine what kind of geometrical features we should use and to tell the meaning of those features. In contrast to this, appearance-based approach is more natural since it uses face images themselves. Second, it does not need to extract facial features or points very accurately. To get good geometrical features, we need to know quite accurate facial feature or point locations which requires accurate facial feature extraction. In contrast to this, we need to know relatively small number of facial features for alignment in the appearance-based approach.

We follow the above process for appearance-based gender classification and use Gaussian process classifiers.

III. Gaussian Process Classifiers

Let us assume that we have a data set D of data points $\vec{x}_i$ with binary class labels $y_i \in \{-1, 1\}$: $D = \{(\vec{x}_i, y_i)|i = 1, 2, \ldots, n\}, X = \{\vec{x}_i|i = 1, 2, \ldots, n\}, Y = \{y_i|i = 1, 2, \ldots, n\}$. Given this data set, we wish to find the correct class label for a new data point $\tilde{\vec{x}}$. We do this by computing the class probability $p(\tilde{y}|\tilde{\vec{x}}, D)$.

We assume that the class label is obtained by transforming some real valued latent variable $\tilde{f}$, which is the value of some latent function $f(\cdot)$ evaluated at $\tilde{\vec{x}}$. We put a Gaussian process prior on this function, meaning that any number of points evaluated from the function have a multivariate Gaussian density (see [17] for a review of GPs). Assume that this GP prior is parameterized by Θ which we will call the hyperparameters. We can write the probability of interest given Θ as:

$$p(\tilde{y}|\tilde{\vec{x}}, D, \Theta) = \int p(\tilde{y}|\tilde{f}, \Theta) p(\tilde{f}|D, \tilde{\vec{x}}, \Theta)\, d\tilde{f} \tag{1}$$

This is the probability of the class label $\tilde{y}$ at a new data point $\tilde{\vec{x}}$ given data D and hyperparameters Θ

The second part of Eq 1 is obtained by further integration over $\vec{f} = [f_1 f_2 \cdots f_n]$, the values of the latent function at the data points.

$$p(\tilde{f}|D, \tilde{\vec{x}}, \Theta) = \int p(\vec{f}, \tilde{f}|D, \tilde{\vec{x}}, \Theta)\, d\vec{f} \tag{2}$$

$$= \int p(\tilde{f}|\tilde{\vec{x}}, \vec{f}, \Theta) p(\vec{f}|D, \Theta) d\vec{f} \tag{3}$$

where $p(\tilde{f}|\tilde{\vec{x}}, \vec{f}, \Theta) = p(\tilde{f}, \vec{f}|\tilde{\vec{x}}, X, \Theta)/p(\vec{f}|X, \Theta)$ and

$$p(\vec{f}|D, \Theta) \propto p(Y|\vec{f}, X, \Theta) p(\vec{f}|X, \Theta) \tag{4}$$

$$= \{\prod_{i=1}^{n} p(y_i|f_i, \Theta)\} p(\vec{f}|X, \Theta). \tag{5}$$

The first term is the likelihood : the probability for each observed class given the latent function value, while the second term is the GP prior over functions evaluated at the data. Writing the dependence of $\vec{f}$ on $\vec{x}$ implicitly, the GP prior over functions can be written

$$p(\vec{f}|X, \Theta) = \frac{1}{(2\pi)^{N/2}|C_\Theta|^{1/2}} \exp(-\frac{1}{2}(\vec{f} - \boldsymbol{\mu})^\top C_\Theta^{-1}(\vec{f} - \boldsymbol{\mu})), \tag{6}$$

where the mean $\boldsymbol{\mu}$ is usually assumed to be the zero vector $\vec{0}$ and each term of a covariance matrix C_{ij} is a function of $\vec{x}_i$ and $\vec{x}_j$, i.e. $c(\vec{x}_i, \vec{x}_j)$.

One form for the likelihood term $p(y_i|f_i, \Theta)$, which relates $f(\vec{x}_i)$ monotonically to probability of $y_i = +1$, is

$$p(y_i|f_i, \Theta) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{y_i f(\vec{x}_i)} \exp(-\frac{z^2}{2})\, dz = \mathrm{erf}(y_i f(\vec{x}_i)). \tag{7}$$

Other possible forms for the likelihood are a sigmoid function $1/(1+\exp(-y_i f(\vec{x}_i)))$, a step function $H(y_i f(\vec{x}_i))$, and a step function with a labelling error $\epsilon+(1-2\epsilon)H(y_i f(\vec{x}_i))$.

Since $p(\vec{f}|D,\Theta)$ in Eq 5 is intractable due to the non-linearity in the likelihood terms, we use an approximate method. Laplace approximation, variational methods and Markov Chain Monte Carlo method were used in [12], [13], and [11], respectively. Expectation propagation, which is described in the next section, was used in [14] and [15]

IV. The EM-EP Algorithm for GPCs

A. EP for GPCs

The Expectation-Propagation (EP) algorithm is an approximate Bayesian inference method [15]. We review EP in its general form before describing its application to GPCs.

Consider a Bayesian inference problem where the posterior over some parameter ϕ is proportional to the prior times likelihood terms for an i.i.d. data set

$$p(\phi|y_1,\ldots,y_n) \propto p(\phi)\prod_{i=1}^{n} p(y_i|\phi) \tag{8}$$

We approximate this by

$$q(\phi) \propto \tilde{t}_0(\phi)\prod_{i=1}^{n} \tilde{t}_i(\phi) \tag{9}$$

where each term (and therefore q) is assumed to be in the exponential family. EP successively solves the following optimization problem for each i

$$\tilde{t}_i(\phi) = \arg\min_{\tilde{t}_i(\phi)} \mathrm{KL}\left(\frac{q(\phi)}{\tilde{t}_i^{\mathrm{old}}(\phi)}p(y_i|\phi)\middle\|\frac{q(\phi)}{\tilde{t}_i^{\mathrm{old}}(\phi)}\tilde{t}_i(\phi)\right) \tag{10}$$

Where KL is the Kullback-Leibler divergence and

$$\mathrm{KL}(p(x)||q(x)) = \int p(x)\log\frac{p(x)}{q(x)}\,dx. \tag{11}$$

Since q is in the exponential family, this minimization is solved by matching moments of the approximated distribution. EP iterates over i until convergence. The algorithm is not guaranteed to converge although it did in practice in all our examples and has worked well for many other authors. Assumed Density Filtering (ADF) is a special online form of EP where only one pass through the data is performed ($i = 1,\ldots n$).

We describe EP for GPC referring to [15] and [14]. The latent function $\vec{f}$ plays the role of the parameter ϕ above. The form of the likelihood we use in the GPC is

$$p(y_i|f_i) = \epsilon + (1-2\epsilon)H(y_i f_i), \tag{12}$$

where $H(x) = 1$ if $x > 0$, and otherwise 0. The hyperparameter, ϵ in Eq 12 models labeling error outliers. The EP algorithm approximates the posterior $p(\vec{f}|D) = p(\vec{f})p(D|\vec{f})/p(D)$ as a Gaussian having the form $q(\vec{f}) \sim \mathcal{N}(\vec{m}_{\vec{f}}, V_{\vec{f}})$, where the GP prior $p(\vec{f}) \sim \mathcal{N}(\vec{0}, C)$ has covariance matrix C with elements C_{ij} defined by the covariance function

$$C_{ij} = c(\vec{x}_i,\vec{x}_j) = v_0\exp\{-\frac{1}{2}\sum_{m=1}^{d} l_m d_m(x_i^m, x_j^m)\} + v_1 + v_2\delta(i,j), \tag{13}$$

where x_i^m is the mth element of $\vec{x}_i$, and $d_m(x_i^m, x_j^m) = (x_i^m - x_j^m)^2$ if x^m is continous; $1-\delta(x_i^m, x_j^m)$ if x is discrete, where $\delta(x_i^m, x_j^m)$ is 1 if $x_i^m = x_j^m$ and 0 if $x_i^m \neq x_j^m$. The hyperparameter v_0 specifies the overall vertical scale of variation of the latent values, v_1 the overall bias of the latent values from zero mean, v_2 the latent noise variance, and l_m the (inverse) lengthscale for feature dimension m. The erf likelihood term in Eq 7 is equivalent to using the threshold function in Eq 12 with $\epsilon = 0$ and non-zero latent noise v_2.

EP tries to approximate $p(\vec{f}|D) = p(\vec{f})/p(D)\prod_{i=1}^{n} p(y_i|\vec{f})$, where $p(\vec{f}) \sim \mathcal{N}(0, C)$. $p(y_i|\vec{f}) = t_i(\vec{f})$ is approximated by $\tilde{t}_i(\vec{f}) = s_i\exp(-\frac{1}{2v_i}(f_i - m_i)^2)$. From this initial setting, we can derive EP for GPC by applying the general idea described above. The resulting EP procedure is virtually identical to the one derived in [15]. We define the following notation[2]: $\Lambda = \mathrm{diag}(v_1,\ldots,v_n)$; $h_i = E[f_i]$; $h_i^{\setminus i} = E[f_i^{\setminus i}]$, where $h_i^{\setminus i}$ and $f_i^{\setminus i}$ are quantities obtained from a whole set except for $\vec{x}_i$. The EP algorithm is as follows which we repeat for completeness—please refer to [15] for the details of the derivation. After the initialization $v_i = \infty, m_i = 0, s_i = 1, h_i = 0, \lambda_i = C_{ii}$, the following process is performed until all (m_i, v_i, s_i) converge:

Loop $i = 1, 2, \ldots, n$:

1) Remove the approximate density $\tilde{t}_i$ (for ith data point) from the posterior to get an 'old' posterior: $h_i^{\setminus i} = h_i + \lambda_i v_i^{-1}(h_i - m_i)$
2) Recompute part of the new posterior: $z = \frac{y_i h_i^{\setminus i}}{\sqrt{\lambda_i}}$; $Z_i = \epsilon + (1-2\epsilon)\mathrm{erf}(z)$ $\alpha_i = \frac{1}{\sqrt{\lambda_i}}\frac{(1-2\epsilon)\mathcal{N}(z;0,1)}{\epsilon+(1-2\epsilon)\mathrm{erf}(z)}$; $h_i = h_i^{\setminus i} + \lambda_i\alpha_i$, where $\mathrm{erf}(z)$ is a cumulative normal density function.
3) Get a new $\tilde{t}_i$: $v_i = \lambda_i(\frac{1}{\alpha_i h_i} - 1)$; $m_i = h_i + v_i\alpha_i$; $s_i = Z_i\sqrt{1 + v_i^{-1}\lambda_i}\exp(\frac{\lambda_i\alpha_i}{2h_i})$
4) Now that v_i is updated, finish recomputing the new posterior: $A = (C^{-1} + \mathbf{\Lambda}^{-1})^{-1}$; For all i, $h_i = \sum_j A_{ij}\frac{m_j}{v_j}$; $\lambda_i = (\frac{1}{A_{ii}} - \frac{1}{v_i})^{-1}$

Our approximated posterior over the latent values is:

$$q(\vec{f}) \sim \mathcal{N}(\tilde{C}\boldsymbol{\alpha}, A), \tag{14}$$

where $\tilde{C}_{ij} = y_j c(\vec{x}_i,\vec{x}_j)$ (or $\tilde{C} = C\mathrm{diag}(\vec{y})$). Classification of a new data point $\tilde{\vec{x}}$ can be done according to $\underset{\tilde{y}}{\mathrm{argmax}}\ p(\tilde{y}|\tilde{\vec{x}}) = \mathrm{sgn}(E[\tilde{f}]) = \mathrm{sgn}(\sum_{i=1}^{n}\alpha_i y_i c(\vec{x}_i,\tilde{\vec{x}}))$.

The approximate evidence can be obtained as:

$$p(Y|X,\Theta) \approx \frac{|\Lambda|^{1/2}}{|C+\Lambda|^{1/2}}\exp(B/2)\prod_{i=1}^{n} s_i \tag{15}$$

[2] $\mathrm{diag}(v_1,\ldots,v_n)$ means a diagonal matrix whose diagonal elements are $v_1,\ldots,v_n$. Similarly for $\mathrm{diag}(\vec{v})$.

where $B = \sum_{ij} A_{ij} \frac{m_i m_j}{v_i v_j} - \sum_i \frac{m_i^2}{v_i}$. The approximate evidence in Eq 15 can be used to evaluate the feasibility of kernels or their hyperparameters to the data. But, it is tricky to get a hyperparameter updating rule from Eq 15. In the following section, we derive the algorithm to find the hyperparameters automatically based not on Eq 15 but a variational lower bound of the evidence.

B. The EM-EP algorithm

EP for GPCs propose a method to estimate latent values but not hyperparameters. We put $\Theta = \Theta \cup \{\epsilon\}$, and $\Theta = \{v_0, v_1, v_2\} \cup \{l_p | p = 1, 2, \ldots, d\}$. for the hyperparameters. Here we present the EM-EP algorithm based on EP to estimate both latent values and hyerparameters [16]. We tackle the problem of learning the classifier hyperparameters as one of optimizing hyperparameters for Gaussian process regression with hidden target values. This idea makes it possible to apply an EM-like algorithm. In the E-step, we infer the approximate (Gaussian) density for latent function values $q(\vec{f})$ using EP. In the M-step, using $q(\vec{f})$ obtained in the E-step, we maximize the variational lower bound of $p(Y|X, \Theta)$. The E-step and M-step are alternated until convergence.

E-step EP iterations are performed given the hyperparameters. $p(\vec{f}|D)$ is approximated as a Gaussian density $q(\vec{f})$ given by Eq 14.

M-step Given $q(\vec{f})$ obtained from the E-step, find the hyperparameters which maximize the variational lower bound of $p(Y|X, \Theta) = \int p(Y|\vec{f}, X, \epsilon) p(\vec{f}|X, \Theta)\, d\vec{f}$. Since the above integral is intractable, we take a variational lower bound F as follows.

$$\begin{aligned} \log p(Y|X,\Theta) &= \log \int p(Y|\vec{f}, X, \epsilon) p(\vec{f}|X, \Theta)\, d\vec{f} \\ &\geq \int q(\vec{f}) \log \frac{p(Y|\vec{f}, X, \epsilon) p(\vec{f}|X, \Theta)}{q(\vec{f})}\, d\vec{f} \\ &= F \end{aligned} \tag{16}$$

Using the E-step result Eq 14 and the definition of $\tilde{C}$, we obtain the following gradient update rule with respect to the covariance hyperparameters

$$\begin{aligned} \frac{\partial F}{\partial \Theta} &= \frac{1}{2} \boldsymbol{\alpha}^\top \mathrm{diag}(\vec{y}) \frac{\partial C}{\partial \Theta} \mathrm{diag}(\vec{y}) \boldsymbol{\alpha} \\ &- \frac{1}{2} tr(C^{-1} \frac{\partial C}{\partial \Theta}) + \frac{1}{2} tr(C^{-1} \frac{\partial C}{\partial \Theta} C^{-1} A). \end{aligned} \tag{17}$$

(See [16] for the derivation of the M-step.)

We found that in practice EM-EP always converged and the local maxima were good solutions. EM-EP has a complexity of $O(n^3)$ due to the matrix inversion in EP.

V. Experimental Results

We performed experiments on appearance-based gender classification with Gaussian processes using the database PF01 (Postech Faces 2001) [18]. This database has color face images of 103 Asian people, 53 men and 50 women, where

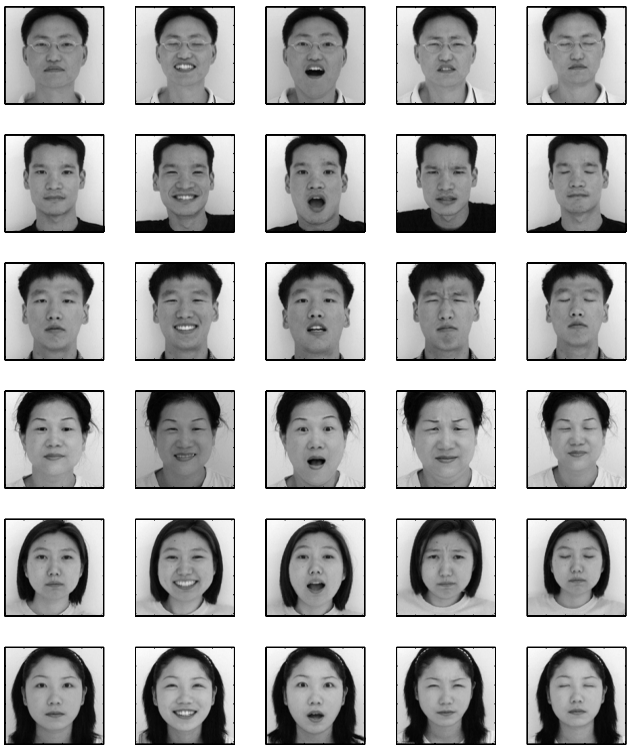
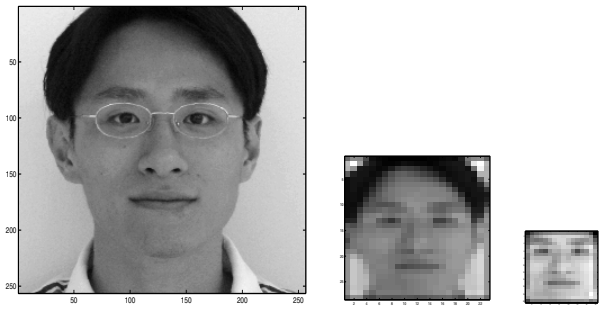


Fig. 2. Some images in the database PF01

for each person there are 17 images under various conditions (1 normal, 4 illumination-varying ones, 8 pose-varying ones, 4 expression-varying ones).

We performed gender classification on two partial data sets where one includes only normal face images (103 images, Faceset I) and another includes normal and expression ($5 \times 103 = 515$ images, Faceset II). Figure 2 shows the normal and expression-varying images of 3 men and 3 women in the database. For each partial data set, we preprocessed face images in two ways. The first from of preprocessing downsampled and cropped face images including hairs and contour of faces and the second form o further cropped the face images to exclude hair and background. Figure 3 shows the example of a normalized image (256x256) and a cropped face image (56x46, cropped type A) and a more cropped face image (20x16, cropped type B). All images are aligned so that eyes are placed in the same positions, which can be done with eye detection algorithms in practice.

We have 4 different data sets: data set I (Faceset I, cropped type A), data set II (Faceset I, cropped type B), data set III (Faceset II, cropped type A), and data set IV (Faceset II, cropped type B). Data set I, II, III and IV are the data sets which include normal faces, more cropped normal faces, expression-varying faces, and more cropped expression-varying faces, respectively. On these data sets, we applied many different classifiers including one nearest neighbor (1-NN), linear discriminant analysis (LDA), SVM with cross-validation (SVM-CV), SVM with EM-EP hyperparameters (SVM-EP), and GPC with the EM-EP algorithm



(a) (b) (c)

Fig. 3. Preprocessed Images: (a) Normalized image, (b) Downsampled and cropped image (56x46), (c) Downsampled and more cropped image (20x16)

	Nor.	Crop.
1-NN	0.1409±0.0415	0.1400±0.0295
LDA	0.1491±0.0292	0.0464±0.0217
CV-SVM	0.0645±0.0251	0.0464±0.0296
EP-SVM	0.0555±0.0214	0.0282±0.0151
EP-GPC	**0.0464±0.0217**	**0.0273±0.0146**
	Nor. E.	Crop. E.
1-NN	0.1609±0.0423	0.1509±0.0335
LDA	0.2247±0.0380	0.1802±0.0305
CV-SVM	0.0520±0.0217	0.0742±0.0266
EP-SVM	**0.0282±0.0151**	0.0655±0.0151
EP-GPC	0.0576±0.0227	**0.0578±0.0188**

TABLE I

CLASSIFICATION ERROR RATES OF VARIOUS METHODS FOR 4 KINDS OF GENDER CLASSIFICATION DATA SETS

(GPC-EP). Table I shows the classification error rates of these various methods for the 4 data sets. The numbers in Table I are the means of those 10 error rates and standard deviations of the mean estimators.

GPC-EP used a single lengthscale hyperparameter (i.e. $l_m = l$) for all feature dimensions [3]. In all GPC models the hyperparameter ϵ was not updated but fixed to zero. In SVM-EP the kernel (i.e. covariance function) had the same hyperparameters as the corresponding GPC-EP that were trained using EM-EP except for the latent noise variance v_2 which was omitted because it caused degradation in SVM performance[4]. Instead, the penalty parameter C allowing training errors (i.e. penalizing the SVM slack variables) was selected by 5-fold cross-validation.[5] In SVM-CV we applied SVMs with a Gaussian kernel with a single lengthscale hyperparameter (without v_0, v_1 and v_2) selected by 5-fold cross-validation.[6] We also had to determine the penalty parameter C, so we performed a 2-level grid search over a 2-dimensional parameter space (C, l)[7].

In the data set I, II and IV, GPC-EP is the best, and in the data set III SVM-EP is the best. In all the data sets. SVM-EP is better than SVM-CV. Therefore, for the data sets tested the hyperparameters found by the EM-EP algorithm seem to be also more suitable hyperparameters for SVMs than the ones obtained by cross-validation. This shows that the EM-EP algorithm finds suitable hyperparameters successfully and those hyperparameters are also suitable for SVMs. This result is consistent with the result on the benchmark data sets in [16]. In all the data sets GPC-EP gives excellent performance.

[3]The initial values of hyperparameters for the first fold were as follows: $v_0^0 = 1$, $v_1^0 = 0.0001$, $v_2^0 = 0.001$, $l_m^0 = l^0 = 1/(2 \times d), \forall m$, and those for subsequent folds are the results for the former fold.

[4]For SVMs, we used the MATLAB Support Vector Machine Toolbox available from `http://theoval.sys.uea.ac.uk/~gcc/svm/toolbox` with modified kernel functions.

[5]Firstly, we did a coarse grid search over $\{C|\log_{10} C = 0, 0.5, 1, 1.5, 2, 2.5, 3\}$ to obtain C_1. Then did a finer grid search over $\{C|\log_{10} C = -0.4 + \log C_1, -0.3 + \log C_1, \ldots, 0.4 + \log C_1\}$.

[6]Similarly to the selection of C, we did a 2-level grid search over $\{l|\log_{10} l = -3, -2.5, -2, -1.5, -1, -0.5, 0\}$ and $\{l|\log_{10} l = -0.4 + \log_{10} l_1, -0.3 + \log l_1, \ldots, 0.4 + \log l_1\}$.

[7]The same grids as above for parameters C, l were used.

VI. CONCLUSION

We have proposed the gender classification technique with one of Bayesian kernel methods which is GPCs. GPCs incorporate the Bayesian model selection framework to determine the kernel hyperparameters, which is an important advantage over SVMs. In the experiments the hyperparameters obtained by GPC with the EM-EP algorithm were even more suitable for SVMs than the ones obtained by cross validation. GPCs worked better than SVMs and provided kernel hyperparameters to make SVMs work better.

We used Gaussian kernels in this paper. Gaussian kernels do not seem to be ideal for image data since they do not capture correlations between pixels. If we invent more proper kernels for face images, we might improve the performance. It would also be interesting to perform experiments on a larger face data set.

ACKNOWLEDGMENT

Hyun-Chul Kim, Daijin Kim and Sung-Yang Bang would like to thank the Ministry of Education of Korea for its financial support toward the Division of Mechanical and Industrial Engineering, and the Division of Electrical and Computer Engineering at POSTECH through BK21 program.

REFERENCES

[1] G. Cottrell and J. Metcalfe, "EMPATH: Face, emotion, and gender recognition using holons," in *Advances in neural information processing systems 3*, vol. 3. MIT Press, 1991.

[2] B. Golomb, D. Lawrence, and T. Sejnowski, "SEXNET: A neural network identifies sex from human faces," in *Advances in neural information processing systems 3*, vol. 3. MIT Press, 1991.

[3] S. Tamura, Kawai, and H. Mitsumoto, "Male/female identification from 8x6 very low resolution face images by neural networks," *Pattern Recognition*, vol. 29, no. 2, pp. 331–335, 1996.

[4] S. Gutta, J. R. J. Huang, P. Jonathon, and H. Wechsler, "Mixture of experts for classification of gender, ethnic origin, and pose of human faces," *IEEE Transactions on Neural Networks*, vol. 11, no. 4, pp. 948–960, 2000.

[5] B. Moghaddam and M. Yang, "Learning gender with support faces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 707–711, 2002.

[6] A. Jain and J. Huang, "Integrating independent components and linear discriminant analysis," in *Proceedings of the 6th International Conference on Automatic Face and Gesture Recognition*, 2004.

[7] N. Costen, M. Brown, and S. Akamatsu, "Sparse models for gender classification," in *Proceedings of the 6th International Conference on Automatic Face and Gesture Recognition*, 2004.
[8] A. Burton, V. Bruce, and N. Dench, "What's the difference between men and wemen? evidence from facial measurements," *Perception*, vol. 22, pp. 153–176, 1993.
[9] R. Brunelli and T. Poggio, "Hyperbf networks for gender classification," in *DARPA Image Understanding Workshop*, 1992.
[10] A. O'Hagan, "On curve fitting and optimal design for regression," *Journal of the Royal Statistical Society B*, vol. 40, pp. 1–32, 1978.
[11] R. Neal, "Monte Carlo implementation of Gaussian process models for Bayesian regression and classification," *Technical Report CRG–TR–97–2, Dept. of Computer Science, University of Toronto*, 1997.
[12] C. K. I. Williams and D. Barber, "Bayesian classification with Gaussian processes," *IEEE Transactions on PAMI*, vol. 20, pp. 1342–1351, 1998.
[13] M. Gibbs and D. J. C. MacKay, "Variational Gaussian process classifiers," *IEEE Transcations on NN*, vol. 11, no. 6, p. 1458, 2000.
[14] M. Opper and O. Winther, "Gaussian processes for classification: Mean field algorithms," *Neural Computation*, vol. 12, pp. 2655–2684, 2000.
[15] T. Minka, *A family of algorithms for approximate Bayesian inference*. PhD thesis, MIT, 2001.
[16] H.-C. Kim and Z. Ghahramani, "The EM-EP algorithm for Gaussian process classification," in *Proceedings of the Workshop on Probabilistic Graphical Models for Classification (ECML)*, 2003, pp. 37–48.
[17] C. K. I. Williams and C. E. Rasmussen, "Gaussian processes for regression," in *NIPS 8*, vol. 8. MIT Press, 1995.
[18] H.-C. Kim, J.-W. Sung, H.-M. Je, S.-K. Kim, B.-J. Jun, D. Kim, and S.-Y. Bang, "Asian face image database PF01," *Technical Report, Intelligent Multimedia Lab, Dept. of CSE, POSTECH*, 2001.