Available online at www.sciencedirect.com
SCIENCE DIRECT®
Pattern Recognition Letters 27 (2006) 618–626
Pattern Recognition Letters
www.elsevier.com/locate/patrec

Appearance-based gender classification with Gaussian processes

Hyun-Chul Kim [a,*], Daijin Kim [b], Zoubin Ghahramani [c], Sung Yang Bang [b]

[a] Department of Industrial and Management Engineering, POSTECH, Pohang University of Science and Technology, San 31, Hyoja Dong, Nam Gu, Pohang 790-784, South Korea
[b] Department of Computer Science and Engineering, POSTECH, Pohang 790-784, South Korea
[c] Gatsby Computational Neuroscience Unit, UCL, London WC1N 3AR, UK

Received 11 August 2004; received in revised form 15 September 2005
Available online 18 November 2005

Communicated by Y.J. Zhang

Abstract

This paper concerns 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, support vector machines (SVMs) which are popular kernel classifiers have been applied to gender classification and have shown excellent performance. SVMs have difficulty in determining the hyperparameters in kernels (using cross-validation). We propose to use Gaussian process classifiers (GPCs) which are Bayesian kernel classifiers. The main advantage of GPCs over SVMs is that they determine the hyperparameters of the kernel based on Bayesian model selection criterion. The experimental results show that our methods outperformed SVMs with cross-validation in most of data sets. Moreover, the kernel hyperparameters found by GPCs using Bayesian methods can be used to improve SVM performance.

Keywords: Gender classification; Appearance-based gender classification; Kernel machines; Gaussian process classifiers; Support vector machines

1. 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. Cottrell and Metcalfe (1991) reduced the dimension of whole face images by autoencoder network and classified gender based on the reduced input features. Golomb et al. (1991) used a two-layer neural network (called SexNet) without dimensionality reduction. Tamura et al. (1996) used a neural network and showed that even very low resolution image such as 8×8 can be used for gender classification. Gutta et al. (2000) used the mixture of experts with ensembles of radial basis functions (RBF) networks and a decision tree as a gating network. Moghaddam and Yang (2002) showed that support vector machines (SVMs) worked better than other classifiers such as ensemble of RBF networks, classical RBF networks, Fisher linear discriminant, nearest neighbor etc. Jain and Huang (2004) extracted wholistic features by independent component analysis (ICA) and classified it with linear discriminant analysis (LDA). Costen et al. (2004) used the exploratory basis pursuit classification which is a sparse kernel classifier.

* Corresponding author. Tel.: +82 562 279 8075; fax: +82 562 279 2299.
E-mail address: grass@postech.ac.kr (H.-C. Kim).

0167-8655/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.patrec.2005.09.027

The second approach is the geometrical feature based approach. Burton et al. (1993) extracted point-to-point distances from 73 points on face images and used discriminant analysis as a classifier. Brunelli and Poggio (1992) 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 (Moghaddam and Yang, 2002). In their experiments the Gaussian 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 performance, 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 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 (O'Hagan, 1978; Neal, 1997; Williams and Barber, 1998; Gibbs and MacKay, 2000). 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 (Williams and Barber, 1998), Markov Chain Monte Carlo (Neal, 1997), and variational approximations (Gibbs and MacKay, 2000). Opper and Winther (2000) used the TAP[2] approach originally proposed in statistical physics of disordered systems to integrate over the latent values. The TAP approach for this model is equivalent to the more general expectation propagation (EP) algorithm for approximate inference (Minka, 2001). The expectation maximization–expectation propagation (EM–EP) algorithm has been proposed to learn the hyperparameters based on EP (Kim and Ghahramani, 2003). 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 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.

2. 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.

Fig. 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 (pixel-level features) 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. The benefit in being natural is that it could make it easier to do what the natural being does. 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. The disadvantages of the appearance-based

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

[2] TAP is an abbreviation of its developers' names such as Thouless, Anderson and Palmer.

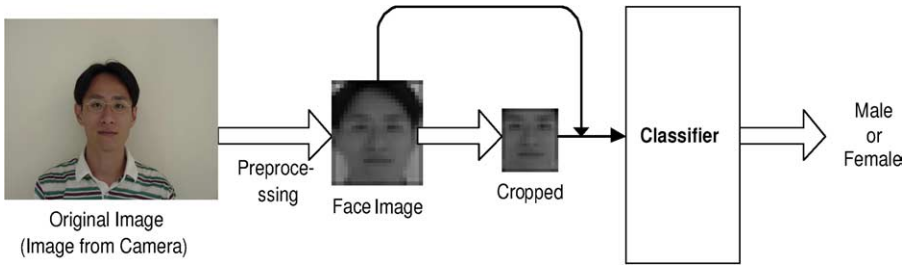

Fig. 1. The process of appearance-based gender classification.

approach is that it has more features than the geometrical feature based approach and that it does not provide a good explanation why a facial image is classified as a male or female.

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

3. Gaussian process classifiers

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

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 $\tilde{\boldsymbol{x}}$. 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 Williams and Rasmussen (1995) 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

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

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

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

$$\begin{aligned} p(\tilde{f}|D, \tilde{\boldsymbol{x}}, \Theta) &= \int p(\boldsymbol{f}, \tilde{f}|D, \tilde{\boldsymbol{x}}, \Theta)\mathrm{d}\boldsymbol{f} \\ &= \int p(\tilde{f}|\tilde{\boldsymbol{x}}, \boldsymbol{f}, \Theta)p(\boldsymbol{f}|D, \Theta)\mathrm{d}\boldsymbol{f}, \end{aligned} \tag{2}$$

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

$$\begin{aligned} p(\boldsymbol{f}|D, \Theta) &\propto p(Y|\boldsymbol{f}, X, \Theta)p(\boldsymbol{f}|X, \Theta) \\ &= \left\{\prod_{i=1}^{n} p(y_i|f_i, \Theta)\right\} p(\boldsymbol{f}|X, \Theta). \end{aligned} \tag{3}$$

The first term in Eq. (3) 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 $\boldsymbol{f}$ on $\boldsymbol{x}$ implicitly, the GP prior over functions can be written

$$\begin{aligned} p(\boldsymbol{f}|X, \Theta) = &\frac{1}{(2\pi)^{N/2}|\boldsymbol{C}_\Theta|^{1/2}} \\ &\times \exp\left(-\frac{1}{2}(\boldsymbol{f} - \boldsymbol{\mu})^\top \boldsymbol{C}_\Theta^{-1}(\boldsymbol{f} - \boldsymbol{\mu})\right), \end{aligned} \tag{4}$$

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

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

$$p(y_i|f_i, \Theta) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{y_i f(\boldsymbol{x}_i)} \exp\left(-\frac{z^2}{2}\right)\mathrm{d}z = \mathrm{erf}(y_i f(\boldsymbol{x}_i)). \tag{5}$$

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

Since $p(\boldsymbol{f}|D, \Theta)$ in Eq. (3) 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 (Williams and Barber, 1998; Gibbs and MacKay, 2000; Neal, 1997), respectively. Expectation propagation, which is described in the next section, was used in (Opper and Winther, 2000; Minka, 2001).

4. The EM–EP algorithm for GPCs

4.1. Expectation propagation for GPCs

The expectation-propagation (EP) algorithm is an approximate Bayesian inference method (Minka, 2001). 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

$$p(\phi|y_1, \ldots, y_n) \propto p(\phi)\prod_{i=1}^{n} p(y_i|\phi). \tag{6}$$

We approximate this by

$$q(\phi) \propto \tilde{t}_0(\phi)\prod_{i=1}^{n} \tilde{t}_i(\phi), \tag{7}$$

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

$$\tilde{t}_i^{\text{new}}(\phi) = \arg\min_{\tilde{t}_i(\phi)} \mathrm{KL}\left(\frac{q(\phi)}{\tilde{t}_i^{\text{old}}(\phi)} p(y_i|\phi) \middle\| \frac{q(\phi)}{\tilde{t}_i^{\text{old}}(\phi)} \tilde{t}_i(\phi)\right), \tag{8}$$

where KL is the Kullback–Leibler divergence and

$$\mathrm{KL}(p(x)\|q(x)) = \int p(x) \log \frac{p(x)}{q(x)} \mathrm{d}x. \tag{9}$$

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 (Minka, 2001; Opper and Winther, 2000). The latent function $\boldsymbol{f}$ plays the role of the parameter ϕ above. The form of the likelihood we use in the GPC is

$$p(y_i|f_i) = \epsilon + (1 - 2\epsilon) H(y_i f_i), \tag{10}$$

where $H(x) = 1$ if $x > 0$, and otherwise 0. The hyperparameter, ϵ in Eq. (10) models labeling error outliers. The EP algorithm approximates the posterior $p(\boldsymbol{f}|D) = p(\boldsymbol{f})p(D|\boldsymbol{f})/p(D)$ as a Gaussian having the form $q(\boldsymbol{f}) \sim \mathcal{N}(\boldsymbol{m}_{\boldsymbol{f}}, \boldsymbol{V}_{\boldsymbol{f}})$, where the GP prior $p(\boldsymbol{f}) \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{C})$ has covariance matrix $\boldsymbol{C}$ with elements C_{ij} defined by the covariance function

$$\begin{aligned} C_{ij} &= c(\boldsymbol{x}_i, \boldsymbol{x}_j) \\ &= v_0 \exp\left\{-\frac{1}{2}\sum_{m=1}^{d} l_m d_m(x_i^m, x_j^m)\right\} + v_1 + v_2 \delta(i,j), \end{aligned} \tag{11}$$

where x_i^m is the mth element of $\boldsymbol{x}_i$, and $d_m(x_i^m, x_j^m) = (x_i^m - x_j^m)^2$ if x^m is continuous; $1 - \delta(x_i^m, x_j^m)$ 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. (5) is equivalent to using the threshold function in Eq. (10) with $\epsilon = 0$ and non-zero latent noise v_2.

EP tries to approximate $p(\boldsymbol{f}|D) = p(\boldsymbol{f})/p(D)\prod_{i=1}^{n} p(y_i|\boldsymbol{f})$, where $p(\boldsymbol{f}) \sim \mathcal{N}(0, \boldsymbol{C})$. $p(y_i|\boldsymbol{f}) = t_i(\boldsymbol{f})$ is approximated by $\tilde{t}_i(\boldsymbol{f}) = s_i \exp(-\frac{1}{2v_i}(f_i - m_i)^2)$. 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 (Minka, 2001). We define the following notation[3]: $\Lambda = \mathrm{diag}(v_1, \ldots, v_n)$; $h_i = E[f_i]$; $h_i^{\backslash i} = E[f_i^{\backslash i}]$, where $h_i^{\backslash i}$ and $f_i^{\backslash i}$ are quantities obtained from a whole set except for $\boldsymbol{x}_i$. The EP algorithm is as follows which we repeat for completeness—please refer to Minka (2001) 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: $h_i^{\backslash i} = h_i + \lambda_i v_i^{-1}(h_i - m_i)$.
(2) Recompute part of the new posterior: $z = \frac{y_i h_i^{\backslash i}}{\sqrt{\lambda_i}}$; $Z_i = \epsilon + (1 - 2\epsilon)\mathrm{erf}(z)$ $\alpha_i = \frac{1}{\sqrt{\lambda_i}} \frac{(1-2\epsilon)\cdot\mathcal{N}(z;0,1)}{\epsilon + (1-2\epsilon)\mathrm{erf}(z)}$; $h_i = h_i^{\backslash i} + \lambda_i \alpha_i$, 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: $\boldsymbol{A} = (\boldsymbol{C}^{-1} + \boldsymbol{\Lambda}^{-1})^{-1}$; 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:

$$q(\boldsymbol{f}) \sim \mathcal{N}(\widetilde{\boldsymbol{C}}\boldsymbol{\alpha}, \boldsymbol{A}), \tag{12}$$

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

The approximate evidence can be obtained as

$$p(Y|X, \Theta) \approx \frac{|\Lambda|^{1/2}}{|\boldsymbol{C} + \Lambda|^{1/2}} \exp(B/2) \prod_{i=1}^{n} s_i, \tag{13}$$

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. (13) 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. (13). In the following section, we derive the algorithm to find the hyperparameters automatically based not in Eq. (13) but a variational lower bound of the evidence.

4.2. The EM–EP algorithm

EP for GPCs propose a method to estimate latent values but not hyperparameters. We put $\Theta = \Theta_{\text{cov}} \cup \{\epsilon\}$, and $\Theta_{\text{cov}} = \{v_0, v_1, v_2\} \cup \{l_p | p = 1, 2, \ldots, d\}$ for the hyperparameters. Here we present the EM–EP algorithm based on EP to estimate both latent values and hyerparameters (Kim and Ghahramani, 2003). 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 approximate EM (expectation maximization) algorithm. In the E-step, we infer the approximate (Gaussian) density for latent function values $q(\boldsymbol{f})$ using EP. In the M-step, using $q(\boldsymbol{f})$ 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.

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

E-step: EP iterations are performed given the hyperparameters. $p(\boldsymbol{f}|D)$ is approximated as a Gaussian density $q(\boldsymbol{f})$ given by Eq. (12).

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

$$\log p(Y|X,\Theta) = \log \int p(Y|\boldsymbol{f},X,\epsilon)p(\boldsymbol{f}|X,\Theta_{\text{cov}})\mathrm{d}\boldsymbol{f} \geqslant \int q(\boldsymbol{f}) \log \frac{p(Y|\boldsymbol{f},X,\epsilon)p(\boldsymbol{f}|X,\Theta_{\text{cov}})}{q(\boldsymbol{f})}\mathrm{d}\boldsymbol{f} = F. \tag{14}$$

Using the E-step result Eq. (12) and the definition of $\tilde{\boldsymbol{C}}$, we obtain the following gradient update rule with respect to the covariance hyperparameters

$$\frac{\partial F}{\partial \Theta_{\text{cov}}} = \frac{1}{2}\boldsymbol{\alpha}^{\top}\mathrm{diag}(\boldsymbol{y})\frac{\partial \boldsymbol{C}}{\partial \Theta_{\text{cov}}}\mathrm{diag}(\boldsymbol{y})\boldsymbol{\alpha} - \frac{1}{2}\mathrm{tr}\left(\boldsymbol{C}^{-1}\frac{\partial \boldsymbol{C}}{\partial \Theta_{\text{cov}}}\right) + \frac{1}{2}\mathrm{tr}\left(\boldsymbol{C}^{-1}\frac{\partial \boldsymbol{C}}{\partial \Theta_{\text{cov}}}\boldsymbol{C}^{-1}\boldsymbol{A}\right). \tag{15}$$

(See Kim and Ghahramani, 2003 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.

5. Experimental results

We performed experiments on appearance-based gender classification with Gaussian processes using the database PF01 (Postech Faces 2001) (Kim et al., 2001) and Aleix database (Martinez and Benavente, 1998). The database PF01 has color face images of 103 Asian people, 53 men and 50 women, where for each person there are 17 images under various conditions (one normal, four illumination-varying ones, eight pose-varying ones, four expression-varying ones). The Aleix database has over 4000 color images of 126 people's faces (70 men and 56 women), where images are frontal view faces with different facial expressions, illumination conditions, and occlusions (sun glasses and scarf).

We performed gender classification on four partial data sets which are only normal face images (103 images, Faceset PF-I) in PF01, normal and expression-varying face images ($5 \times 103 = 515$ images, Faceset PF-II) in PF01, only normal face images (126 images, Faceset AL-I) in PF01, and normal and expression-varying face images ($4 \times 126 = 504$ images, Faceset AL-II) in PF01. Figs. 2 and 3 show the normal and expression-varying images of three men and three women in the database PF01 and the Aleix database, respectively. 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 of preprocessing further cropped the face images to exclude hair and background. Fig. 4 shows the example of a normalized image (256×256) and a cropped face image

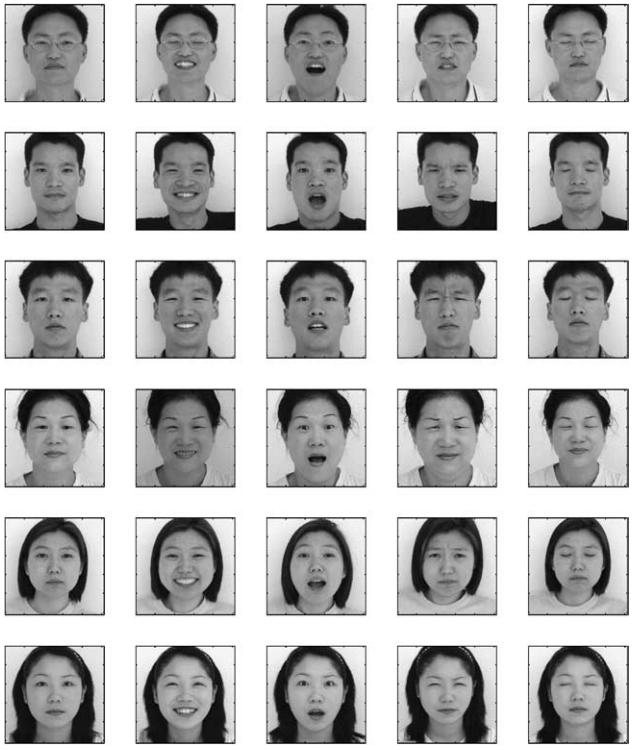

Fig. 2. Some images in the database PF01.

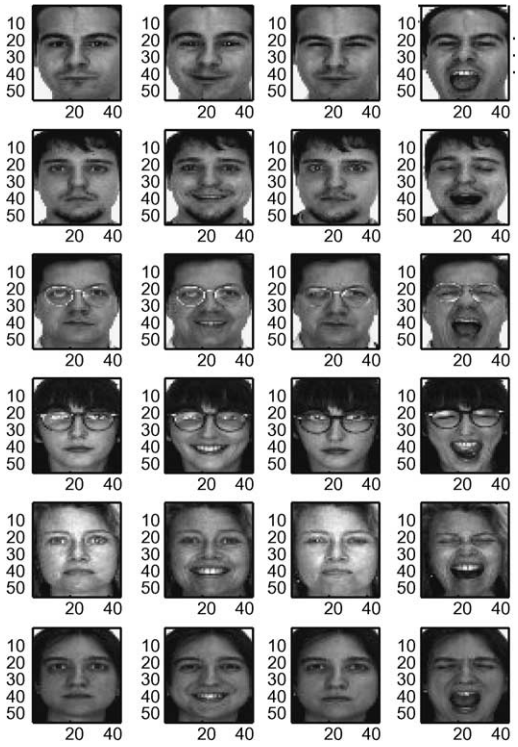

Fig. 3. Some images in the Aleix database.

(28×23, cropped type A) and a more cropped face image (20×16, 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. If images are not aligned well, the appearance-based method would not work well.

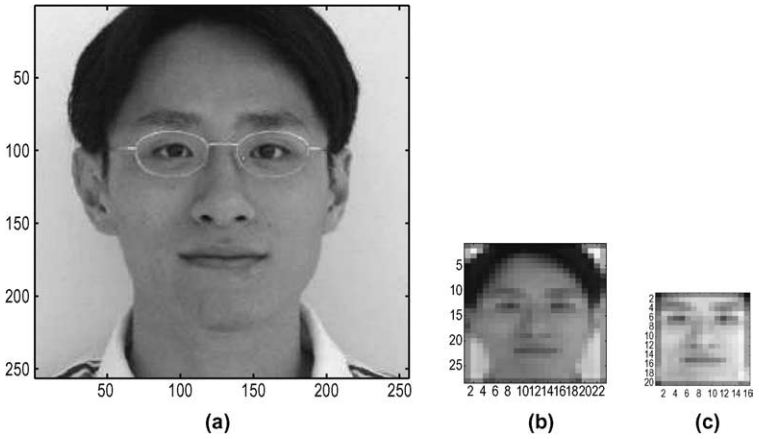

Fig. 4. Preprocessed images: (a) Normalized image, (b) downsampled and cropped image (28×23), (c) downsampled and more cropped image (20×16).

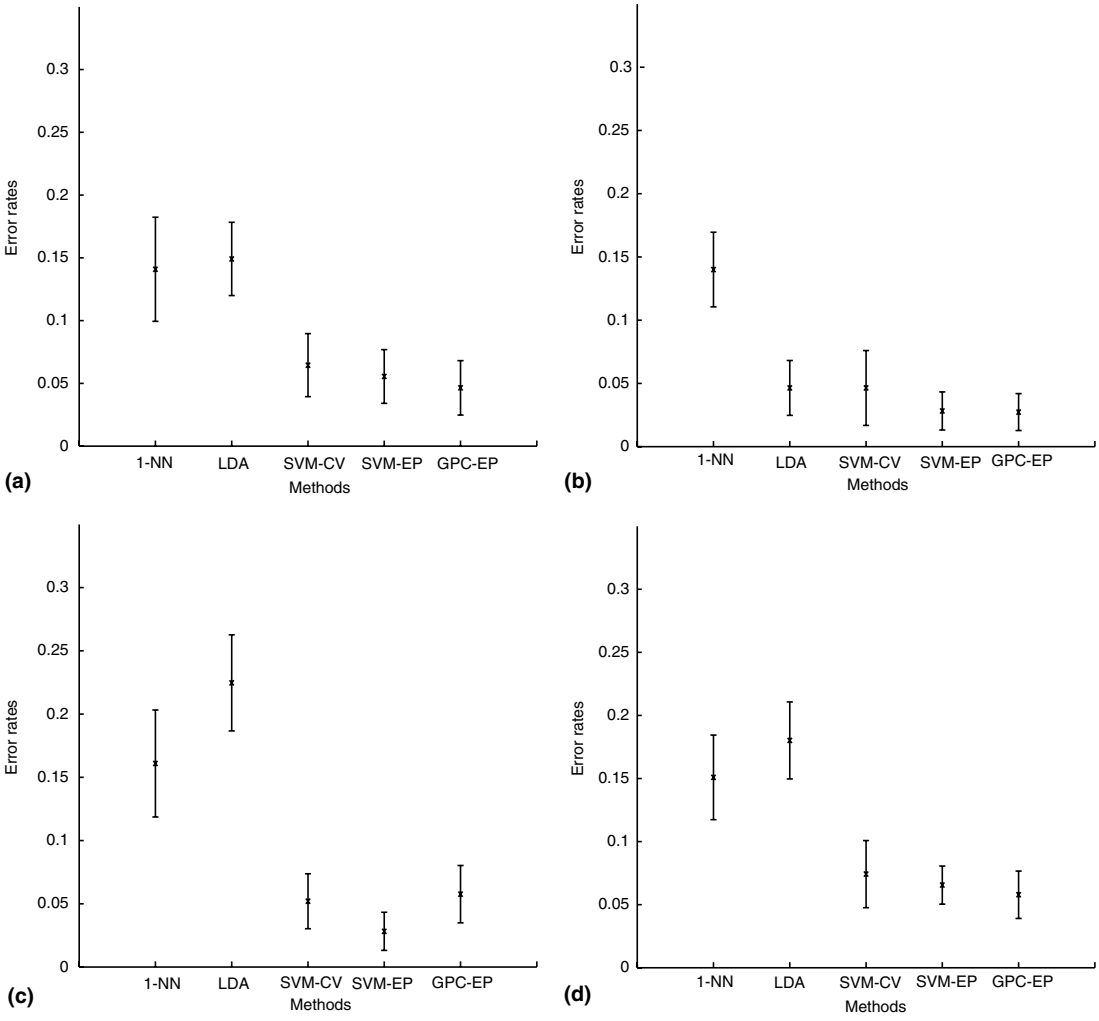

Fig. 5. Classification error rates of various methods for four kinds of gender classification data sets (PF'01 DB): (a) data set P-I, (b) data set P-II, (c) data set P-III, (d) data set P-IV.

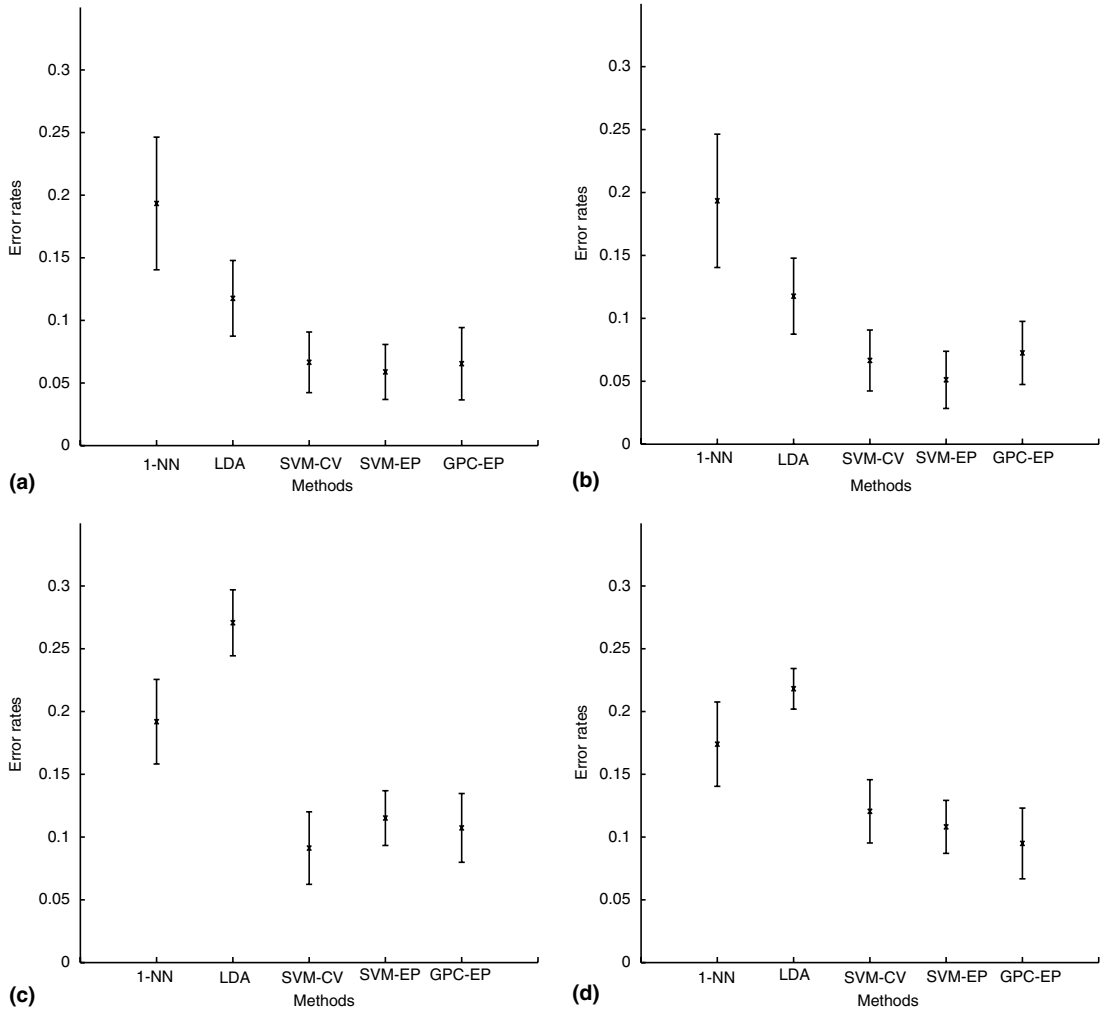

Fig. 6. Classification error rates of various methods for four kinds of gender classification data sets (Aleix DB): (a) data set A-I, (b) data set A-II, (c) data set A-III, (d) data set A-IV.

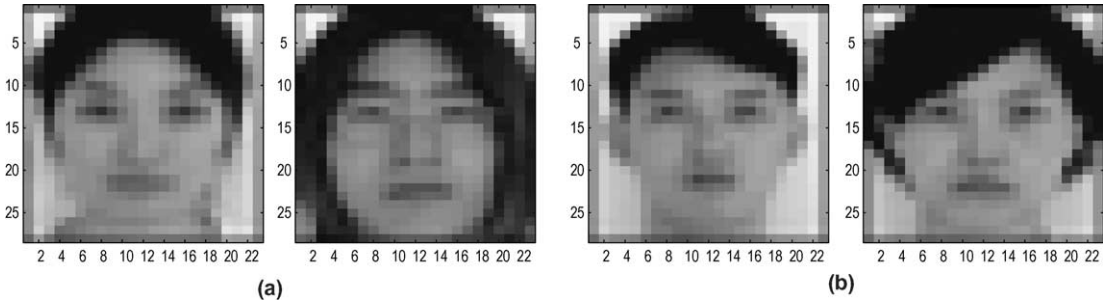

Fig. 7. Misclassified facial images: (a) data set P-I, (b) data set P-II.

We have eight different data sets: data set P-I (Faceset PF-I, cropped type A), data set P-II (Faceset PF-I, cropped type B), data set P-III (Faceset PF-II, cropped type A), and data set P-IV (Faceset PF-II, cropped type B), data set A-I (Faceset AL-I, cropped type A), data set A-II (Faceset AL-I, cropped type B), data set A-III (Faceset AL-II, cropped type A), and data set A-IV (Faceset AL-II, cropped type B). Data set P-I, P-II, P-III and P-IV are the data sets which include normal faces, more cropped normal faces, expression-varying faces, and more cropped expression-varying faces from the database PF01, respectively. Data set A-I, A-II, A-III and A-IV are the data sets which include normal faces, more cropped normal faces, expression-varying faces, and more cropped expression-varying faces from the Aleix database, 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 (GPC-EP). Figs. 5 and 6 show the classification error rates of these methods over four different data sets in the database PF01 and Aleix database. Each data set was divided into 10 folds. Each fold was subsequently used as a test set, while the other nine folds were used as a training set. Before GPC or SVM are applied, all feature values are normalized based on the training set so that their means are zero and their variances are one. The points '×' in Figs. 5 and 6 are means of 10 trials and error bars are from standard deviations of the mean estimators. In Fig. 7, we show misclassified images. In Fig. 7(a), we can know that long hair is not always a key feature of women. The classification rates of data sets of cropped type A are not always better than ones of cropped type B.

GPC-EP used a single lengthscale hyperparameter (i.e. $l_m = l$) for all feature dimensions[4]. 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[5]. Instead, the penalty parameter C allowing training errors (i.e. penalizing the SVM slack variables) was selected by 5-fold cross-validation.[6] 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 five-fold cross-validation.[7] 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)[8].

In the data set P-I, P-II, P-IV, and A-IV, GPC-EP is the best, in the data set P-III, A-I, and A-II, SVM-EP is the best, and in the data set A-III SVM-CV is the best. In all the data sets except for one, GPC-EP or SVM-EP is the best. Also, in all the data sets except for one, 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 (Kim and Ghahramani, 2003).

6. Conclusion

We have proposed the appearance-based gender classification method with Gaussian processes. 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. In most of the data sets, EM–EP algorithms worked better than SVMs with cross-validation 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.

Acknowledgments

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

Brunelli, R., Poggio, T., 1992. Hyperbf networks for gender classification. In: DARPA Image Understanding Workshop.

Burton, A., Bruce, V., Dench, N., 1993. What's the difference between men and women? Evidence from facial measurements. Perception 22, 153–176.

Costen, N., Brown, M., Akamatsu, S., 2004. Sparse models for gender classification. In: Proceedings of the 6th International Conference on Automatic Face and Gesture Recognition.

Cottrell, G., Metcalfe, J., 1991. EMPATH: Face, emotion, and gender recognition using holonsAdvances in Neural Information Processing Systems 3, vol. 3. MIT Press.

Gibbs, M., MacKay, D.J.C., 2000. Variational Gaussian process classifiers. IEEE Trans. NN 11 (6), 1458.

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

Gutta, S., Huang, J.R.J., Jonathon, P., Wechsler, H., 2000. Mixture of experts for classification of gender, ethnic origin, and pose of human faces. IEEE Trans. Neural Networks 11 (4), 948–960.

Jain, A., Huang, J., 2004. Integrating independent components and linear discriminant analysis. In: Proceedings of the 6th International Conference on Automatic Face and Gesture Recognition.

Kim, H.-C., Ghahramani, Z., 2003. The EM–EP algorithm for Gaussian process classification. In: Proceedings of the Workshop on Probabilistic Graphical Models for Classification (ECML), pp. 37–48.

[4] 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.

[5] 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.

[6] 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 $\{C|\log_{10}C = -0.4 + \log C_1, -0.3 + \log C_1, \ldots, 0.4 + \log C_1\}$.

[7] 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 $\{l|\log_{10}l = -0.4 + \log_{10}l_1, -0.3 + \log l_1, \ldots, 0.4 + \log l_1\}$.

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

Kim, H.-C., Sung, J.-W., Je, H.-M., Kim, S.-K., Jun, B.-J., Kim, D., Bang, S.-Y., 2001. Asian face image database PF01. Technical Report, Intelligent Multimedia Lab, Department of CSE, POSTECH.

Martinez, A., Benavente, R., 1998. The ar face database. CVC Technical Report #24.

Minka, T., 2001. A family of algorithms for approximate Bayesian inference. Ph.D. Thesis, MIT.

Moghaddam, B., Yang, M., 2002. Learning gender with support faces. IEEE Trans. Pattern Anal. Mach. Intell. 24 (5), 707–711.

Neal, R., 1997. Monte Carlo implementation of Gaussian process models for Bayesian regression and classification. Technical Report CRG-TR-97-2, Department of Computer Science, University of Toronto.

O'Hagan, A., 1978. On curve fitting and optimal design for regression. J. Royal Stat. Soc. B 40, 1–32.

Opper, M., Winther, O., 2000. Gaussian processes for classification: Mean field algorithms. Neural Comput. 12, 2655–2684.

Tamura, S., Kawai, Mitsumoto, H., 1996. Male/female identification from 8×6 very low resolution face images by neural networks. Pattern Recogn. 29 (2), 331–335.

Williams, C.K.I., Barber, D., 1998. Bayesian classification with Gaussian processes. IEEE Trans. PAMI 20, 1342–1351.

Williams, C.K.I., Rasmussen, C.E., 1995. Gaussian processes for regression. In: NIPS 8, vol. 8. MIT Press.