A Variational Bayesian Approach for Classification with Corrupted Inputs

Chao Yuan and Claus Neubauer
Siemens Corporate Research, 755 College Road East, Princeton, NJ 08540
{chao.yuan,claus.neubauer}@siemens.com

Abstract

Classification of corrupted images, for example due to occlusion or noise, is a challenging problem. Most existing methods tackled this problem using a two-step strategy: image reconstruction and classification of reconstructed images. However, their performances heavily relied on the accuracy of reconstruction and parameter estimation. We present a full Bayesian approach which infers the class label from the corrupted image by marginalizing the original image and parameters. Overfitting is effectively overcome through Bayesian integration. Our system consists of two models. The original image model, which specifies the original image generation process, is described by a Gaussian mixture model. The observation model, which relates the corrupted image to the original image, is depicted by an additive deviation model. Normal pixel and corrupted pixel values are elegantly handled by the covariance of the Gaussian deviation. We employ variational approximation to make the Bayesian integration tractable. The advantage of the proposed method is demonstrated by classification tests on the USPS digit database and PIE face database with pose and illumination variations.

1. Introduction

Object classification has received a large amount of attention over the last several decades and state-of-the-art performance has been achieved by classifiers such as support vector machines [1]. However, classification of corrupted images remains a challenging problem. A typical cause of image corruption is occlusion. For example, a face image can be occluded by sun glasses. Images can also be contaminated by noise such as impulse noise.

The problem of classifying corrupted images is difficult because a corrupted image can have tremendous number of possible variations. This is due to the fact that any pixel in an image can be corrupted and a corrupted pixel can take any value. This problem becomes relatively easier if the location of corrupted pixels [2] or the type of corruption is known [3]. However, such information is often unavailable; this paper addresses this more challenging situation.

Most prior work tackling this problem proceeded in two steps. In the first step, the original image $\mathbf{x}$ was reconstructed from the corrupted image $\mathbf{y}$. In the second step, classification was performed on the reconstructed image $\mathbf{x}$. Several studies have achieved good reconstruction results [4-6]. However, all these methods have to solve a difficult task: parameter estimation. Overfitting, which tends to occur either in image reconstruction or parameter estimation, will negatively affect the performance of the classifier.

Our major contribution is to propose a full Bayesian approach: we infer the class label c from the corrupted image $\mathbf{y}$ by marginalizing both original image $\mathbf{x}$ and parameters. By this Bayesian integration, the inference does not rely on particular point estimation of $\mathbf{x}$ and parameters, but a weighted combination of all possible settings of them as noted by MacKay [7] and Neal [8]. Bias, which happens in each individual setting, is expected to cancel out with each other and to be reduced through this integration.

Fig.1 shows the graphical model of the proposed system, which consists of two parts: the original image model and the observation model. The original image model employs a Gaussian mixture model (GMM) to describe how the original image $\mathbf{x}$ is generated. Each component of the GMM is denoted by a class c and a state variable s. Class c refers to an identity to be inferred such as a face in face recognition; state s represents a variation mode of a class, for example certain pose and illumination for face images. This GMM is learned using normal training images. The observation model describes how the corrupted image $\mathbf{y}$ relates to $\mathbf{x}$. We model $\mathbf{y}$ as the sum of $\mathbf{x}$ and a Gaussian deviation vector $\boldsymbol{\epsilon}$ whose covariance matrix $\boldsymbol{\Theta}^{-1}$ is adaptively adjusted for each pixel. A Gamma distribution is selected as the prior for $\boldsymbol{\Theta}$. With marginalization of state s, original image $\mathbf{x}$ and parameter $\boldsymbol{\Theta}$, our inference problem is formulated as

$$P(c|\mathbf{y}) = \sum_s \int_{\mathbf{x}} \int_{\boldsymbol{\Theta}} P(c, s, \mathbf{x}, \boldsymbol{\Theta}|\mathbf{y}) d\mathbf{x}\, d\boldsymbol{\Theta}. \qquad (1)$$

Since (1) does not have an analytic form, we resort to variational approximation [9-10] to make the integration tractable. Variational methods have been widely used in

1-4244-1180-7/07/$25.00 ©2007 IEEE

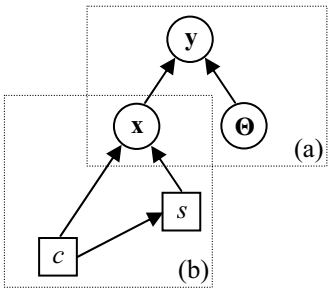

Figure 1. Graphical model for the proposed algorithm. (a) the observation model, in which the observed image $\mathbf{y}$ is modeled as the sum of the original image $\mathbf{x}$ and Gaussian deviation whose covariance is $\mathbf{\Theta}^{-1}$. (b) the original image model, where the original image $\mathbf{x}$ is modeled by a Gaussian mixture model governed by class c and state s. Class c is an identity to be inferred and state s indicates different variations of a class. Squares denote discrete variables and circles denote continuous variables. Our task is to infer the class label c from the observed image $\mathbf{y}$ by marginalizing s, $\mathbf{x}$ and $\mathbf{\Theta}$.

computer vision and have achieved great success in motion segmentation [11], visual tracking [12] and image deblurring [13].

The proposed framework is quite general without assuming any knowledge about corruption or application-specific physical models. We test our method using the USPS handwritten digit database [14] and the PIE face database with pose and illumination variations [15].

This paper is organized as follows. Section 2 surveys the related work. In Section 3, we describe the proposed variational inference algorithm. Test results are presented in Section 4. Section 5 summarizes this paper.

2. Related work

Most existing work focused on reconstructing the original image from the corrupted image, where robustness is the key. Simply speaking, robustness refers to the desirable property that original pixel values are recovered at corrupted pixels while normal pixels stay unchanged.

Some methods made special assumptions about their problems. For example, the location of corrupted pixels was assumed known by Hwang and Lee [2]. The original image was modeled as a linear combination of a set of prototype images and the linear combination coefficients were computed only from uncorrupted regions. Assuming that occlusion was due to eye glasses, Wu *et al*. [3] presented a sophisticated system to detect, localize and remove eye glasses. Both methods achieved impressive reconstruction results. However, their assumptions may not hold for other applications.

More general approaches include reconstruction using kernel PCA (KPCA) [16]. This is similar to a PCA based reconstruction, except that the reconstruction is done in a transformed feature space, which has a built-in ability to handle nonlinearity. However, as a projection-based method, the KPCA updated all pixels of a test image, which compromised its robustness performance [5].

Robustness was also pursued via a robust error function by Gross *et al*. [4]. The fitting of an active appearance model was thus less affected by the corrupted pixel values. Tsuda and Rätsch [5] addressed robustness using $L1$-norm distance and slack variables in the framework of linear programming. However, selection of scale parameters for the robust error function [4] or weight for slack variables [5] relied on applications and also the severity level of corruption. How to adaptively set these parameters for a test image is still a challenging problem.

Smet *et al*. [6] presented one of the best approaches for reconstructing occluded face images. A binary visibility map was introduced to govern the switching between normal pixel values and corrupted pixel values. The normal pixel values were generated from the powerful 3D morphable model [17] while the corrupted pixel values were modeled by a histogram. The Expectation-Maximization (EM) algorithm [18] was used to estimate the parameters of the 3D morphable model. The use of visibility map was shown to be a form of robust estimation (Fransens *et al*. [19]). However, reliably estimating the histogram of the occluded pixels from a single image is still an open issue.

Martínez [20] did not perform image reconstruction, but divided an image into fixed sub-regions. Matching was performed on each sub-region and combined to form the final classification result. This method can achieve excellent results as shown in our tests (Section 4.2). However, it is sensitive to the location of occlusion and may not perform well if occlusion happens to be present in most sub-regions.

3. Description of the proposed algorithm

Our goal is to infer the class label c given a test image $\mathbf{y}$. However, Eq.(1) does not have an analytic solution. We thus consider variational treatment. To proceed, we first describe the dependency relation of all variables in Fig.1. This is done for the observation model part (Section 3.1) and the original image model part (Section 3.2), separately. Then, the variational approximation and the actual inference are introduced in Section 3.3. Finally, algorithm speedup procedures are given in Section 3.4.

3.1. The observation model

We model the corrupted d-dimensional image $\mathbf{y}$ as the sum of the original image $\mathbf{x}$ and a deviation vector $\boldsymbol{\epsilon}$:

$$\mathbf{y} = \mathbf{x} + \boldsymbol{\epsilon}, \tag{2}$$

where $\boldsymbol{\epsilon}$ is independent of $\mathbf{x}$. $\boldsymbol{\epsilon}$ is assumed to be Gaussian with zero mean and a diagonal covariance matrix: $P(\boldsymbol{\epsilon}) = \mathcal{N}(\boldsymbol{\epsilon}|\mathbf{0}, \boldsymbol{\Theta}^{-1})$, where $\boldsymbol{\Theta} = diag(\theta_1, \theta_2, \ldots, \theta_d)$ is the inverse of the covariance matrix. This implies that pixel deviation ϵ_i is independent to each other. The conditional probability of $\mathbf{y}$ given $\mathbf{x}$ and $\boldsymbol{\Theta}$ (Fig.1a) can be expressed as

$$P(\mathbf{y}|\mathbf{x}, \boldsymbol{\Theta}) = \prod_{i=1}^{d} \mathcal{N}(y_i|x_i, \theta_i^{-1}). \tag{3}$$

With the Gaussian deviation vector $\boldsymbol{\epsilon}$, the observed image $\mathbf{y}$ can now be interpreted in a generative manner. On the one hand, if a pixel i is corrupted, the corresponding inverse variance θ_i can be set to a small value such that the observed value y_i is allowed to be different from its original value x_i. On the other hand, at a normal pixel, θ_i is set to a large value such that the deviation ϵ_i is almost deterministically zero and y_i is forced to be close to x_i.

Following Bishop [9], Ghahramani and Beal [10], we assume that each inverse variance θ_i has an independent Gamma distribution:

$$P(\boldsymbol{\Theta}) = \prod_{i=1}^{d} \Gamma(\theta_i|a, b), \tag{4}$$

where $\Gamma(\theta_i|a, b) \propto b^a \theta_i^{a-1} e^{-b\theta_i}$ denotes a Gamma distribution with hyper-parameters a and b. $P(x) \propto f(x)$ denotes that probability density function $P(x)$ is proportional to $f(x)$. Similarly to [9, 13], we prefer small a and b values and fix $a = b = 10^{-5}$ in this paper. The motivation for such choices will be given in Section 3.3.

3.2. The original image model

The original image model (Fig.1b) describes the prior distribution of the original image $\mathbf{x}$, which is modeled by a Gaussian mixture model (GMM). Each component is specified by a class c, a state s and has a probability of

$$P(c, s, \mathbf{x}) = p_{cs} \mathcal{N}(\mathbf{x}|\mathbf{m}_{cs}, \boldsymbol{\Sigma}_{cs}). \tag{5}$$

p_{cs} is the class and state prior $P(c, s)$; $\mathbf{m}_{cs}$ and $\boldsymbol{\Sigma}_{cs}$ are the mean and covariance of $\mathbf{x}$ given class c and state s, respectively.

The GMM is learned from a set of normal training images $\mathbf{X}$ and their corresponding class labels $\mathbf{C}$. The EM algorithm has been a standard way to estimate p_{cs}, $\mathbf{m}_{cs}$ and $\boldsymbol{\Sigma}_{cs}$ by maximizing $\log P(\mathbf{X}, \mathbf{C})$[18]. However, the following questions need to be addressed: how to cope with the high dimension of images, how to select the number of components and how to achieve high classification rate.

We apply the probabilistic principal component analysis (PPCA) from Tipping and Bishop [21] for dimension reduction of images. Suppose that there are L PPCAs. The lth PPCA is represented by a $d \times M$ eigenvector matrix $\mathbf{U}_l$, an $M \times M$ diagonal eigenvalue matrix $\boldsymbol{\Lambda}_l$, a d-dimensional mean vector $\mathbf{v}_l$ and a residue r_l^2. $M \ll d$ is the number of principal components. For the USPS database, we apply one PPCA to all training images; for the PIE face data, we use nine PPCAs, one PPCA for all training images at each pose. All training data are then projected onto their corresponding M-dimensional eigenspaces.

The EM algorithm is applied to the projected training images for every class in each eigenspace. There are thus a total of C (classes) $\times$ L (eigenspaces) EM computations and CL local GMMs. For simplicity, each mixture component is assumed to have an isotropic Gaussian distribution whose covariance is $\sigma^2 \mathbf{I}_M$. σ^2 is the isotropic variance and $\mathbf{I}_M$ is an $M \times M$ identity matrix.

A ten-fold cross validation is adopted to determine the number of components for each local GMM. To be specific, all projected training images of class c in eigenspace l are randomly divided into ten folds. With certain number of components choice K_{cl}, we apply the EM algorithm to nine folds of the data and compute the log likelihood of the remaining fold. The average log likelihood of ten-fold tests is computed. This cross-validation procedure selects the K_{cl} with the highest average log likelihood. Using this K_{cl} choice, we obtain q_{clk}, $\boldsymbol{\mu}_{clk}$ and $\sigma_{clk}^2 \mathbf{I}_M$, which are the prior, mean and covariance of the kth component of class c in the lth eigenspace, respectively.

To improve the classification performance of the GMM, we refine all local GMMs via discriminative training. Previous studies have shown that better classification rate can be achieved for a generative model if it is trained discriminatively [22-24]. Refs.[22, 23] proposed to maximize the following objective function

$$\log P(\mathbf{X}, \mathbf{C}) - \alpha \log P(\mathbf{X}). \tag{6}$$

The first term of (6) is what we optimize so far in the above EM algorithm. The second term considers the likelihood of $\mathbf{X}$ with respect to other classes, which should be small. α which takes value between 0 and 1 controls the balance between the first term and the second term. We use $\alpha = 1$, since it was shown to be a good choice by Holub and Perona [23]. Alternatively, one can obtain α using cross validation as proposed by Bouchard and Triggs [22]. Discriminative training is applied to each eigenspace to refine all $\boldsymbol{\mu}_{clk}$ while q_{clk} and σ_{clk}^2 are kept unchanged. Conjugate gradient is used to solve this optimization problem.

Finally, we convert all $C \times L$ local GMMs into one global GMM described by (5). The eigenspace label l and local GMM component label k are now combined and correspond to one state label s. The prior for the sth component of class c is

$$p_{cs} = \frac{1}{CL} q_{clk}. \tag{7}$$

Note that we assume the equal prior for all classes and all eigenspaces. The mean $\mathbf{m}_{cs}$ for each component in the original d-dimensional pixel space is reconstructed from the corresponding mean $\boldsymbol{\mu}_{clk}$ in the M-dimensional eigenspace:

$$\mathbf{m}_{cs} = \mathbf{U}_l \boldsymbol{\mu}_{clk} + \mathbf{v}_l. \tag{8}$$

The covariance $\boldsymbol{\Sigma}_{cs}$ for each component in the original pixel space is also reconstructed from the corresponding covariance $\sigma_{clk}^2 \mathbf{I}_M$ in the eigenspace:

$$\boldsymbol{\Sigma}_{cs} = \mathbf{U}_l(\sigma_{clk}^2 \mathbf{I}_M - r_l^2 \mathbf{I}_M)\mathbf{U}_l^T + r_l^2 \mathbf{I}_d. \tag{9}$$

Notice that if $\sigma_{clk}^2 \mathbf{I}_M$ is replaced with the eigenvalue matrix $\boldsymbol{\Lambda}_l$, Eq.(9) becomes the covariance for the single Gaussian distribution modeled by the PPCA [21].

3.3. Variational Bayesian inference

Given a test image $\mathbf{y}$, the objective is to infer its class label c using (1). Since Eq.(1) does not have an analytic solution, we consider variational treatment [9-13]. Specifically, the posterior distribution $P(c, s, \mathbf{x}, \boldsymbol{\Theta}|\mathbf{y})$ (or P) is approximated by a factorized distribution $Q(c, s, \mathbf{x}, \boldsymbol{\Theta})$ (or Q):

$$Q(c, s, \mathbf{x}, \boldsymbol{\Theta}) = Q(c, s, \mathbf{x})Q(\boldsymbol{\Theta}). \tag{10}$$

To ensure a good approximation, we require the Kullback-Leibler divergence $KL(Q||P)$ between Q and original P to be minimized. The great advantage of using Q instead of P is that $P(c|\mathbf{y})$ now has an analytic form.

The minimization of $KL(Q||P)$ with respect to Q involves iterations for estimation of two factorial distributions: $Q(c, s, \mathbf{x})$ and $Q(\boldsymbol{\Theta})$. It can be shown that $Q(c, s, \mathbf{x})$ describes a Gaussian mixture model. Each component has a probability of

$$Q(c, s, \mathbf{x}) = \widetilde{p}_{cs} \mathcal{N}(\mathbf{x}|\widetilde{\mathbf{m}}_{cs}, \widetilde{\boldsymbol{\Sigma}}_{cs}). \tag{11}$$

This GMM is similar to the GMM described by (5) except that new set of prior $\widetilde{p}_{cs}$, mean $\widetilde{\mathbf{m}}_{cs}$ and covariance $\widetilde{\boldsymbol{\Sigma}}_{cs}$ is needed for each component. $Q(\boldsymbol{\Theta})$ can be shown to depict a Gamma distribution:

$$Q(\boldsymbol{\Theta}) = \prod_{i=1}^{d} Q(\theta_i) = \prod_{i=1}^{d} \Gamma(\theta_i|\widetilde{a}_i, \widetilde{b}_i), \tag{12}$$

which is similar to the Gamma distribution in (4). However, different pixels now require different sets of hyperparameters $\widetilde{a}_i$ and $\widetilde{b}_i$.

Each iteration involves two steps. In the first step, we estimate $\widetilde{p}_{cs}$, $\widetilde{\mathbf{m}}_{cs}$ and $\widetilde{\boldsymbol{\Sigma}}_{cs}$, the parameters of (11). They all have analytic forms consisting of p_{cs}, $\mathbf{m}_{cs}$, $\boldsymbol{\Sigma}_{cs}$ and $\langle\boldsymbol{\Theta}\rangle$, where $\langle\boldsymbol{\Theta}\rangle$ denotes the expectation of $\boldsymbol{\Theta}$ with respect to the current estimate of $Q(\boldsymbol{\Theta})$. In the second step, $\widetilde{a}_i$ and $\widetilde{b}_i$, the parameters of $Q(\boldsymbol{\Theta})$ are estimated. Both $\widetilde{a}_i$ and $\widetilde{b}_i$ are ready to compute from a, b, $\langle x_i\rangle$ and $\langle x_i^2\rangle$, where $\langle x_i\rangle$ and $\langle x_i^2\rangle$ are expectation of x_i and x_i^2 with respect to $Q(c, s, \mathbf{x})$, respectively. The detailed forms of these parameters are provided in the appendix.

We now give insights to justify our choice of small a and b (vs. typically much larger values). Recall that the mean and variance of a Gamma distribution $\Gamma(\theta_i|a, b) \propto \theta_i^{a-1} e^{-b\theta_i}$ is a/b and a/b^2, respectively. The posterior mean and variance of θ_i with respect to $Q(\theta_i)$ can be shown to be

$$\langle\theta_i\rangle = \frac{\widetilde{a}_i}{\widetilde{b}_i} = \frac{\frac{1}{2} + a}{\frac{1}{2}\langle(x_i - y_i)^2\rangle + b}, \tag{13}$$

$$\langle(\theta_i - \langle\theta_i\rangle)^2\rangle = \frac{\widetilde{a}_i}{\widetilde{b}_i^2} = \frac{\frac{1}{2} + a}{\left(\frac{1}{2}\langle(x_i - y_i)^2\rangle + b\right)^2}. \tag{14}$$

For a corrupted pixel whose expected square of deviation $\langle\epsilon_i^2\rangle = \langle(x_i - y_i)^2\rangle$ is large, the posterior mean in (13) will be small, together with a small posterior variance in (14). This naturally restricts θ_i to a small value. On the other hand, for a normal pixel whose $\langle(x_i - y_i)^2\rangle$ is small, the posterior mean in (13) and posterior variance in (14) will both be large. This suggests that a broad range of large values be suitable for θ_i. Therefore, with small a and b, the observation model behaves just in the way as we have desired in Section 3.1. For the above reasons, we fix $a = b = 10^{-5}$ in our tests. Note that with these choices, a is negligible in the numerator of (13). Other small values can be used and they do not appreciably affect our results.

Our original inference problem (1) can now be analytically solved with $P(c, s, \mathbf{x}, \boldsymbol{\Theta}|\mathbf{y})$ replaced by its approximation $Q(c, s, \mathbf{x}, \boldsymbol{\Theta})$:

$$\begin{aligned} P(c|\mathbf{y}) &= \sum_s \int_{\mathbf{x}} \int_{\boldsymbol{\Theta}} Q(c, s, \mathbf{x}, \boldsymbol{\Theta}) d\mathbf{x}\, d\boldsymbol{\Theta} \\ &= \sum_s \int_{\mathbf{x}} Q(c, s, \mathbf{x}) d\mathbf{x} \int_{\boldsymbol{\Theta}} Q(\boldsymbol{\Theta}) d\boldsymbol{\Theta} \\ &= \sum_s \int_{\mathbf{x}} \widetilde{p}_{cs} \mathcal{N}(\mathbf{x}|\widetilde{\mathbf{m}}_{cs}, \widetilde{\boldsymbol{\Sigma}}_{cs}) d\mathbf{x} \\ &= \sum_s \widetilde{p}_{cs}. \end{aligned} \tag{15}$$

For every test image $\mathbf{y}$, we compute the approximate posterior probability of a class c given $\mathbf{y}$ or the last equation of (15). This is done for all classes and our algorithm outputs the class label with the highest posterior probability. The only quantities required in (15) are $\widetilde{p}_{cs}$ which are available after the $KL(Q||P)$ is minimized (see Appendix for details).

3.4. Algorithm complexity and speedup

The variational inference involves iterative estimation of the factorized distribution Q, which is a very time con-

suming process. In each iteration, to estimate $Q(c, s, \mathbf{x})$ we must calculate the inverse of $\mathbf{\Sigma}_{cs} + \langle \mathbf{\Theta} \rangle^{-1}$ for the sth component of class c. This requires time of $\mathcal{O}(d^3)$; for images, this can be very costly. In addition, such calculation must be done for every component. Thus, assuming a fixed number of iterations, the total time required is $\mathcal{O}(Kd^3)$, where K is the number of components in the GMM. This complexity is well illustrated by the PIE face test where $d = 40 \times 32 = 1,280$ and $K = 65 \times 9 \times 8 = 4,680$. We propose the following two ways to reduce the computational cost.

First, since $\mathbf{\Sigma}_{cs} + \langle \mathbf{\Theta} \rangle^{-1}$ can be expressed as a form of $\mathbf{ABA}^T + \mathbf{W}$ where $\mathbf{A}$ is a $d \times M$ matrix, $\mathbf{B}$ is an $M \times M$ diagonal matrix and $\mathbf{W}$ is a $d \times d$ diagonal matrix, we use the matrix inverse formula [25]:

$$(\mathbf{ABA}^T + \mathbf{W})^{-1} = \mathbf{W}^{-1} - \mathbf{W}^{-1}\mathbf{A}(\mathbf{B}^{-1} + \mathbf{A}^T\mathbf{W}^{-1}\mathbf{A})^{-1}\mathbf{A}^T\mathbf{W}^{-1}. \quad (16)$$

Recall that M, the number of eigenvectors, is usually much smaller than d. Therefore, the complexity for updating each component is reduced from $\mathcal{O}(d^3)$ to $\mathcal{O}(Md^2)$. This is another advantage of using PPCA and assuming isotropic Gaussian for each component in the eigenspace. Note that we also ignore the determinant of $\Sigma_{cs} + \langle \mathbf{\Theta} \rangle^{-1}$ in evaluating $\widetilde{p}_{cs}$, which is inspired by the success of widely used Mahalanobis distance [26].

Secondly, we perform component selection. The motivation is that usually a small portion of the K components contributes to each update. In other words, the probabilities $\widetilde{p}_{cs}$ of many components are very small and those components can be removed without much loss of precision.

Component selection is performed differently in different stages. During the initialization stage, we select κ components out of K components based on $L1$-norm distance from $\mathbf{y}$ to each component center $\mathbf{m}_{cs}$. Only the κ closest components to the input image $\mathbf{y}$ are kept for the following iterations. κ is empirically chosen to be 50 in this paper and this choice works well. A larger κ may achieve better precision but also increase complexity.

During the iteration stage, we rank the estimated probability $\widetilde{p}_{cs}$ for each component in a descending order after each iteration. We select components, starting with the one with the highest $\widetilde{p}_{cs}$ until the sum of $\widetilde{p}_{cs}$ of the selected components is larger than 95%. Only the selected components are used in the following iterations. In our experience, usually after several iterations, the number of remaining components is less than ten.

Through the matrix inverse formula (16) and component selection, we have managed to reduce the complexity from $\mathcal{O}(Kd^3)$ to a more tractable $\mathcal{O}(\kappa Md^2)$, where κ and M are typically much smaller than K and d, respectively. It takes 10 seconds to process a 40×32 pixel image using MATLAB 7 on a PC with a 2.16 GHz CPU.

Figure 2. Sample test images corrupted by impulse noise in the USPS data classification test. From left to right, each column represents digits $0, 1, \ldots, 9$. Three samples are shown for each digit.

4. Test results

4.1. Classification tests on the USPS database

We used the USPS database consisting of $9,298$ handwritten digits, $7,291$ of which were preset as the training set and the other $2,007$ were for testing [14]. Each image has 16×16 pixels, with each pixel taking value between -1 and 1. We used the original training images to train classifiers but corrupted all test images with impulse noise. Specifically, impulse noise was added to each pixel with probability T; for a corrupted pixel, its value was randomly set to -1 or 1. Fig.2 shows three test images for each of the ten digits with $T = 0.4$. This forms a very difficult classification task even for human.

The number of principal components M needs to be set for our method, PCA and kernel PCA. A good choice appeared to be between 20 and 40 for this database. We set this number to 30 for all methods.

For our variational Bayesian method, we applied a PPCA to all training images. Cross-validation in Section 3.2 automatically determined the number of components for each class. This number ranges from 14 to 55 with an average of 29 for all ten digits. The GMM was trained using the EM algorithm followed by discriminative training as in Section 3.2. The final GMM achieved a classification rate of 92.2% using original test images.

The proposed method (abbreviated as VB) was compared to the above GMM classifier and a SVM using Gaussian kernels, both directly applied to the corrupted test images. This SVM achieved a classification rate of 95.6% on the original test images, which is close to 95.8% [27] and 95.7% [28], reported by other SVM work. In addition, we applied the PCA and KPCA methods to reconstruct the original test image from the corrupted test image and then applied the same SVM to the reconstructed image. These methods are denoted by "PCA SVM" and "KPCA SVM", respectively.

Table 1 shows the classification rate of different methods with three T values (the probability of impulse noise). Our variational Bayesian method achieved the best results for all tests. The SVM's score was the lowest and can be significantly improved if it was applied to the reconstructed image either from the PCA or the KPCA. This implies that both PCA and KPCA can handle the impulse noise on this database reasonably well especially when noise level is low.

P_c	GMM	SVM	PCA SVM	KPCA SVM	VB
$T = 0.3$	71.2%	69.2%	84.5%	82.5%	$\mathbf{86.2}\%$
$T = 0.4$	61.2%	53.8%	72.4%	72.7%	$\mathbf{80.7}\%$
$T = 0.5$	45.2%	41.0%	60.8%	60.1%	$\mathbf{70.0}\%$

Table 1. Classification rate P_c of different methods on the USPS test images corrupted by impulse noise. T is the probability of adding impulse noise to each image pixel. Our proposed variational Bayesian (VB) method achieved the highest classification rate.

For example, at $T = 0.3$, the "PCA SVM" achieved 84.5% which is just 1.7% lower than our score. However, as T increased, the advantage of our method became clearer. For example, at $T = 0.5$, the variational Bayesian outperformed "PCA SVM" by 9.2%.

4.2. Classification tests on the PIE face database

We used a subset of the PIE face database with both pose and illumination variations [15]. This subset consists of 65 (subjects) $\times$ 9 (poses without elevation) $\times$ 21 (illuminations) $= 12,285$ images. Each face image was roughly cropped into a dimension of $40 \times 32 = 1,280$ based on the annotated locations of eyes and mouth. We linearly transformed each pixel value to the range between -1 and 1. For each subject at each pose, eight images with illumination number $2, 4, 6, 13, 15, 18, 19, 20$ were used in training; the other images were for testing. The $12,285$ images were thus split into a training set of 65 (subjects) $\times$ 9 (poses) $\times$ 8 (illuminations) $= 4,680$ images and a test set of 65 (subjects) $\times$ 9 (poses) $\times$ 13 (illuminations) $= 7,605$ images. The number of principal components M was set to 150.

Face images are often corrupted by occlusion, for example caused by sun glasses, hands and other objects. However, the original PIE face images do not have occlusion. We thus consider simulated occlusion. Two types of occlusion were used. In the block occlusion test, three candidate corrupted regions were pre-defined, each with a fixed size of 10×32 (25% of the whole image) and covering either the eye, nose or mouth part. For a test image, we randomly selected one of the above three regions, and set all pixel values within this region to a randomly picked constant. This constant choice is due to the fact that similar pixel values are often observed in real occlusion. Our algorithm will not be adversely affected if corrupted pixels take different values since we assume that pixel deviations are independent to each other as noted in Section 3.1. Fig.3 top shows the corrupted test images of one subject under one pose. In the mesh occlusion (Fig.3 bottom) test, a mesh-like occlusion was similarly added to cover about 25% of the original image, but the position of the occlusion was fixed for all images.

We used normal training images to train all classifiers.

Figure 3. Sample test images corrupted by simulated block occlusion (top row) and mesh occlusion (bottom row) for one subject at one pose in the PIE face database.

For our original image model, nine PPCAs were used, one for each pose. At each pose, all 65 (persons) $\times$ 8 (illuminations) $= 520$ training images were projected onto the corresponding eigenspace. Since there were only eight training images for each local GMM, we employed a kernel density estimator with a Gaussian kernel instead of the EM-based training. Leave-one-out procedure was used to determine the Gaussian kernel width for each local GMM. Discriminative training was then applied to these 65 local GMMs to refine the means of all 520 components. Combining all local GMMs from all nine poses, we obtained a global GMM with $4,680$ components.

The proposed variational Bayesian method was compared to a GMM classifier and a linear discriminant analysis (LDA) classifier ([29]). Using the LDA, a test image was projected onto a Fisherface subspace and classified by a nearest neighbor classifier. We also considered reconstructing an occluded image via a robust KPCA (RKPCA) ([30]) which employed robust estimation to downplay the effects of occluded pixels. Then the LDA was applied to the reconstructed image. We refer to this method as "RKPCA LDA". In addition, following [20], we divided an image into six sub-regions and applied an local LDA to each sub-region; the scores from all six local LDAs were fused through a naive Bayes classifier. This method is denoted by SLDA.

Note that our algorithm and the GMM classifier do not need pose estimation as each test image was matched against all $4,680$ mixture components. However, the LDA, RKPCA and SLDA do need pose estimation. For simplicity, we assumed that the pose of a test image was known only for these methods, which is expected to enhance their results. By using the original uncorrupted test image, the GMM achieved a classification rate of 94.8%. The LDA produced a nearly perfect 99.9%, which justified that the LDA is a top method to handle illumination variations ([29]).

Table 2 shows the classification rate P_c of different classifiers on the test images corrupted by two types of occlusion. The original GMM and LDA performed poorly on both tests. In the block occlusion test, the SLDA achieved a very impressive 98.2%, much higher than that of our VB method. This can be attributed to the fact that at least two of the six sub-regions were not affected by occlusion and faces can be classified based on uncorrupted partial images in this PIE data set. However, in the mesh occlusion test, the SLDA performed much worse because all six sub-regions

Tests	GMM	RKPCA LDA	LDA	SLDA	VB
Block	48.9%	64.8%	28.2%	**98.2%**	84.4%
Mesh	56.0%	56.2%	30.4%	63.8%	**87.5%**

Table 2. Classification rate P_c of different methods on the PIE face images corrupted by block occlusion (second row) and mesh occlusion (third row). Our proposed variational Bayesian (VB) method achieved good and consistent scores in both tests.

were occupted by occlusion and none of the local LDAs could produce a reliable score. This shows that the SLDA is sensitive to the location of occlusion as we noted in Section 2. In comparison, our VB method produced more consistent scores regardless of the occlusion location, because our observation model (Section 3.1) was able to locate occlusion by the adaptive deviation covariance. The RKPCA appeared to fail in image reconstruction because they did not help improving the LDA's performance. A possible reason is that RKPCA was confused between occlusion and normal pixel variations (due to illumination).

5. Summary

This paper presents a Bayesian approach for classification of corrupted images. We infer the class label from the corrupted test image by integrating over the original image and parameters. Our results do not depend on point estimation of the reconstructed image or parameters, but a weighted combination of them. This effectively reduces overfitting. Variational approximation is employed to carry out the Bayesian integration.

Our work can be extended in the following directions. First, despite of discriminative training, the Gaussian mixture model is still found inferior to the SVM or LDA in terms of classification performance. Possible improvements include another Bayesian treatment by combining multiple GMMs. Secondly, the proposed framework can be adapted to tackle multivariate regression problems. This requires the design of a new original image model suitable for regression problems.

Acknowledgement

We thank anonymous reviewers for helpful comments. We also thank Li Zhang (at Columbia Univ.) and Binglong Xie for useful discussions.

Appendix

We show how to estimate $Q(c, s, \mathbf{x})$ and $Q(\mathbf{\Theta})$ which minimizes the KL divergence. The log likelihood of the joint probability of all variables is

$$\begin{aligned}
&\log P(c, s, \mathbf{x}, \mathbf{y}, \mathbf{\Theta}) \\
=&\log P(\mathbf{y}|\mathbf{x}, \mathbf{\Theta}) + \log P(\mathbf{x}|c, s) + \log P(c, s) + \log P(\mathbf{\Theta}) \\
=&-\frac{1}{2}\sum_i (x_i - y_i)^2\theta_i + \frac{1}{2}\sum_i \log\theta_i \\
&-\frac{1}{2}(\mathbf{x} - \mathbf{m}_{cs})^T\mathbf{\Sigma}_{cs}^{-1}(\mathbf{x} - \mathbf{m}_{cs}) - \frac{1}{2}\log|\mathbf{\Sigma}_{cs}| + \log p_{cs} \\
&+ (a-1)\sum_i \log\theta_i - b\sum_i \theta_i + Constant. \qquad \text{(A-1)}
\end{aligned}$$

Step 1. Update $Q(c, s, \mathbf{x})$ with $Q(\mathbf{\Theta})$ fixed.
Using standard variational minimization, $Q(c, s, \mathbf{x})$ which minimizes the KL divergence has the following form

$$Q(c, s, \mathbf{x}) \propto e^{\int \log P(c,s,\mathbf{x},\mathbf{y},\mathbf{\Theta}) Q(\mathbf{\Theta}) d\mathbf{\Theta}}. \qquad \text{(A-2)}$$

After some matrix manipulation, $Q(c, s, \mathbf{x})$ can be shown to describe a Gaussian mixture model

$$Q(c, s, \mathbf{x}) = \widetilde{p}_{cs}\mathcal{N}(\mathbf{x}|\widetilde{\mathbf{m}}_{cs}, \widetilde{\mathbf{\Sigma}}_{cs}), \qquad \text{(A-3)}$$

whose parameters are

$$\widetilde{p}_{cs} \propto p_{cs}\frac{e^{-\frac{1}{2}(\mathbf{y}-\mathbf{m}_{cs})^T(\mathbf{\Sigma}_{cs}+\langle\mathbf{\Theta}\rangle^{-1})^{-1}(\mathbf{y}-\mathbf{m}_{cs})}}{|\mathbf{\Sigma}_{cs} + \langle\mathbf{\Theta}\rangle^{-1}|^{1/2}}, \qquad \text{(A-4)}$$

$$\widetilde{\mathbf{m}}_{cs} = \left(\mathbf{I}_d - \langle\mathbf{\Theta}\rangle^{-1}\left(\mathbf{\Sigma}_{cs} + \langle\mathbf{\Theta}\rangle\right)^{-1}\right)(\mathbf{y} - \mathbf{m}_{cs}) + \mathbf{m}_{cs}, \qquad \text{(A-5)}$$

$$\widetilde{\mathbf{\Sigma}}_{cs} = \langle\mathbf{\Theta}\rangle^{-1}\left(\mathbf{I}_d - \left(\mathbf{\Sigma}_{cs} + \langle\mathbf{\Theta}\rangle\right)^{-1}\langle\mathbf{\Theta}\rangle^{-1}\right). \qquad \text{(A-6)}$$

We additionally compute the following quantities which will be used in updating $Q(\mathbf{\Theta})$ in Step 2:

$$\langle\mathbf{x}\rangle = \sum_c\sum_s \widetilde{p}_{cs}\widetilde{\mathbf{m}}_{cs}, \qquad \text{(A-7)}$$

$$\langle\mathbf{x}\mathbf{x}^T\rangle = \sum_c\sum_s \widetilde{p}_{cs}(\widetilde{\mathbf{\Sigma}}_{cs} + \widetilde{\mathbf{m}}_{cs}\widetilde{\mathbf{m}}_{cs}^T). \qquad \text{(A-8)}$$

Step 2. Update $Q(\mathbf{\Theta})$ with $Q(c, s, \mathbf{x})$ fixed.
$Q(\mathbf{\Theta})$ which minimizes the KL divergence has the following form

$$Q(\mathbf{\Theta}) \propto e^{\sum_c\sum_s\int \log P(c,s,\mathbf{x},\mathbf{y},\mathbf{\Theta}) Q(c,s,\mathbf{x}) d\mathbf{x}}. \qquad \text{(A-9)}$$

$Q(\mathbf{\Theta})$ can be shown to be a Gamma distribution

$$Q(\mathbf{\Theta}) = \prod_{i=1}^d Q(\theta_i) = \prod_{i=1}^d \Gamma(\theta_i|\widetilde{a}_i, \widetilde{b}_i), \qquad \text{(A-10)}$$

whose parameters are determined by

$$\widetilde{a}_i = \frac{1}{2} + a, \qquad \text{(A-11)}$$

$$\widetilde{b}_i = \frac{1}{2}\langle(x_i - y_i)^2\rangle + b. \qquad \text{(A-12)}$$

The expectation of $\mathbf{\Theta}$ is computed to be used in Step 1:

$$\langle \theta_i \rangle = \frac{\widetilde{a}_i}{\widetilde{b}_i} = \frac{\frac{1}{2} + a}{\frac{1}{2}\langle (x_i - y_i)^2 \rangle + b}. \tag{A-13}$$

The above Step 1 and Step 2 are repeated. Upon convergence, we obtain the final forms of $Q(c, s, \mathbf{x})$ and $Q(\mathbf{\Theta})$.

References

[1] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

[2] B. W. Hwang and S. W. Lee. Reconstruction of partially damaged face images based on a morphable face model. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25(3):721–741, 2003.

[3] C. Wu, C. Liu, H. Y. Shum, Y. Q. Xu, and Z. Zhang. Automatic eyeglasses removal from face images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 26(3):322–336, 2004.

[4] R. Gross, I. Matthews, and S. Baker. Constructing and fitting active appearance models with occlusion. In *Proc. IEEE Workshop on Face Processing in Video*, pages 72–72, 2004.

[5] K. Tsuda and G. Rätsch. Image reconstruction by linear programming. *IEEE Trans. on Image Processing*, 14(6):737–744, 2005.

[6] M. De Smet, R. Fransens, and L. Van Gool. A generalized EM approach for 3D model based face recognition under occlusions. In *Proc. IEEE Computer Vision and Pattern Recognition*, pages 1423–1430, 2006.

[7] D. J. C. MacKay. A practical Bayesian framework for backprop networks. *Neural Computation*, 4(3):448–772, 1992.

[8] R. M. Neal. Bayesian learning via stochastic dynamics. In *Advances in Neural Information Processing Systems*, pages 475–482, 1993.

[9] C. M. Bishop. Variational principal components. In *Proc. Int'l Conf. on Artificial Neural Networks*, pages 509–514, 1999.

[10] Z. Ghahramani and M. J. Beal. Variational inference for Bayesian mixtures of factor analysers. In *Advances in Neural Information Processing Systems*, pages 449–455, 2000.

[11] N. Jojic and B. J. Frey. Learning flexible sprites in video layers. In *Proc. IEEE Computer Vision and Pattern Recognition*, pages 199–206, 2001.

[12] G. Hua and Y. Wu. Variational maximum a posteriori by annealed mean field analysis. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 27(11):1–15, 2005.

[13] R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, and W. T. Freeman. Removing camera shake from a single photograph. In *Proc. ACM SIGGRAPH*, pages 787–794, 2006.

[14] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541–551, 1989.

[15] T. Sim, S. Baker, and M. Bsat. The CMU pose, illumination, and expression (PIE) database. In *IEEE. Conf. on Automatic Face and Gesture Recognition*, pages 46–51, 2002.

[16] S. Mika, B. Schölkopf, A. J. Smola, K. R. Müller, M. Scholz, and G. Rätsch. Kernel PCA and de-noising in feature spaces. In *Advances in Neural Information Processing Systems*, pages 536–542, 1999.

[17] V. Blanz and T. Vetter. Face recognition based on fitting a 3D morphable model. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 25(9):1063–1074, 2003.

[18] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum-likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B*, 39(1):1–38, 1977.

[19] R. Fransens, C. Strecha, and L. Van Gool. Robust estimation in the presence of spatially coherent outliers. In *Proc. IEEE Computer Vision and Pattern Recognition Workshop*, pages 102–102, 2006.

[20] A. M. Martínez. Recognizing imprecisely localized, partially occluded, and expression variant faces from a single sample per class. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 24(6):748–763, 2002.

[21] M. E. Tipping and C. M. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society. Series B*, 61(3):611–622, 1999.

[22] G. Bouchard and B. Triggs. The trade-off between generative and discriminative classifiers. In *Proc. Int'l Symposium on Computational Statistics*, pages 721–728, 2004.

[23] A. Holub and P. Perona. A discriminative framework for modelling object classes. In *Proc. IEEE Computer Vision and Pattern Recognition*, pages 664–671, 2005.

[24] J. A. Lasserre, C. M. Bishop, and T. P. Minka. Principled hybrids of generative and discriminative models. In *Proc. IEEE Computer Vision and Pattern Recognition*, pages 87–94, 2006.

[25] K.V. Mardia, J.T. Kent, and J.M. Bibby. *Multivariate Analysis*. Academic Press, 1979.

[26] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification. 2nd Edition*. Wiley Interscience, 2001.

[27] B. Schölkopf, K. Sung, C. J. C. Burges, F. Girosi, P. Niyogi, T. Poggio, and V. Vapnik. Comparing support vector machines with Gaussian kernels to radial basis function classifiers. *IEEE Trans. on Signal Processing*, 45(11):2758–2765, 1997.

[28] H. Zhang, A. C. Berg, M. Maire, and J. Malik. SVM-KNN: Discriminative nearest neighbor classification for visual category recognition. In *Proc. IEEE Computer Vision and Pattern Recognition*, pages 2126–2136, 2006.

[29] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman. Eigenfaces vs. Fisherfaces: recognition using class specific linear projection. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19(7):711–720, 1997.

[30] T. Takahashi and T. Kurita. Robust de-noising by kernel PCA. In *Proc. Int'l Conf. on Artificial Neural Networks*, pages 739–744, 2002.