

CHAPTER 3

APPLICATION: NON-LINEAR IMAGE MODELLING

The variational inference framework can be used to perform Bayesian inference in a very wide variety of domains. In this chapter, it will be used in the domain of machine vision for modelling image subspaces. The approach involves constructing a probabilistically consistent density model which can capture essentially arbitrary non-linearities, and which can discover an appropriate dimensionality for modelling the manifold¹. A key feature is the use of a fully Bayesian formulation in which the appropriate model complexity, and indeed the dimensionality of the manifold itself, can be discovered *automatically* as part of the inference procedure [Bishop 1999a]. The model is based on a mixture of components each of which is a latent variable model whose dimensionality can be inferred from the data. It avoids a discrete model search over dimensionality, involving instead the use of continuous hyper-parameters to determine an effective dimensionality for the components in the mixture model.

I will start by summarising a number of previous approaches to modelling image subspaces. I will go on to describe the probabilistic model used in my approach and how the variational inference framework was used to fit it to the data. Thereafter, results on synthetic and real data sets will be presented.

3.1 Modelling Image Subspaces

Interest in image subspace modelling has grown considerably in recent years in contexts such as recognition, detection, verification and coding. Although an individual image can be considered as a point in a high-dimensional space described by the pixel values, an ensemble of related images, for example faces, lives on a (noisy) non-linear manifold having a much lower *intrinsic* dimensionality. One of the simplest approaches to modelling such manifolds involves finding the principal components of the ensemble of images, as used for example in ‘eigen-faces’ [Turk and Pentland 1991].

¹The work in this chapter is based on a paper written in collaboration with Christopher M. Bishop [Bishop and Winn 2000].

However, simple principal component analysis (PCA) suffers from two key limitations. Firstly, it does not directly define a probability distribution, and so it is difficult to use standard PCA as a component in a probabilistic solution to a computer vision problem. Secondly, the manifold defined by PCA is necessarily linear. Techniques which address the first of these problems by constructing a density model include Gaussians and mixtures of Gaussians [Moghaddam and Pentland 1997]. The second problem has been addressed by considering non-linear projective methods such as principal curves and auto-encoder neural networks [Moghaddam 1999]. Bregler and Omohundro [1995] and Heap and Hogg [1998] use mixture representations to try to capture the non-linearity of the manifold. However, their model fitting is based on simple clustering algorithms (related to K -means) and lacks the fully probabilistic approach, as discussed in this chapter.

A central problem in density modelling in high dimensional spaces concerns model complexity. Models fitted using maximum likelihood are particularly prone to severe over-fitting unless the number of free parameters is restricted to be much less than the number of data points. For example, it is clearly not feasible to fit an unconstrained mixture of Gaussians directly to the data in the original high-dimensional space using maximum likelihood due to the excessive number of parameters in the covariance matrices. Moghaddam and Pentland [1997] therefore project the data onto a PCA sub-space and then perform density estimation within this lower dimensional space using Gaussian mixtures. While this limits the number of free parameters in the model, the non-linearity of the manifold requires the PCA space to have a significantly higher dimensionality than that of the manifold itself, and so again the model is prone to over-parameterisation.

One important aspect of model complexity concerns the dimensionality of the manifold itself, which is typically not known in advance. Moghaddam [1999], for example, arbitrarily fixes the model dimensionality to be 20.

Several authors have explored the use of non-linear warping of the image, for example in the context of face recognition, in order to take account of changes of pose or of interpersonal variation [Black and Yacoob 1995; Cootes et al. 1995; Frey and Jojic 1999]. In so far as such distortions can be accurately represented, these transformations should be of significant benefit in tackling the subspace modelling problem, albeit at increased computational expense. Such approaches can be used to augment virtually any sub-space modelling algorithm, including those discussed in this chapter, and so they will not be considered further.

3.2 Models for Manifolds

This approach to modelling the manifolds of images builds upon recent developments in latent variable models and can be seen as a natural development of PCA and mixture modelling frameworks leading to a highly flexible, fully probabilistic framework. Firstly, I will show how conventional PCA can be reformulated probabilistically and hence used as the component distribution in a mixture model. Then I show how a Bayesian approach allows the

model complexity (including the number of components in the mixture as well as the effective dimensionality of the manifold) to be inferred from the data.

3.2.1 Maximum likelihood PCA

Principal component analysis (PCA) is a widely used technique for data analysis. It can be defined as the linear projection of a data set into a lower-dimensional space under which the retained variance is a maximum, or equivalently under which the sum-of-squares reconstruction cost is minimised.

Consider a data set D of observed d -dimensional vectors $D = \{\mathbf{t}_n\}$ where $n \in \{1, \dots, N\}$. Conventional PCA is obtained by first computing the sample covariance matrix given by

$$\mathbf{S} = \frac{1}{N} \sum_{n=1}^N (\mathbf{t}_n - \bar{\mathbf{t}})(\mathbf{t}_n - \bar{\mathbf{t}})^T \quad (3.1)$$

where $\bar{\mathbf{t}} = N^{-1} \sum_n \mathbf{t}_n$ is the sample mean. Next the eigenvectors \mathbf{u}_i and eigenvalues λ_i of \mathbf{S} are found, where $\mathbf{S}\mathbf{u}_i = \lambda_i\mathbf{u}_i$ and $i = 1, \dots, d$. The eigenvectors corresponding to the q largest eigenvalues (where $q < d$) are retained, and a reduced-dimensionality representation of the data set is defined by $\mathbf{x}_n = \mathbf{U}_q^T(\mathbf{t}_n - \bar{\mathbf{t}})$ where $\mathbf{U}_q = (\mathbf{u}_1, \dots, \mathbf{u}_q)$.

A significant limitation of conventional PCA is that it does not define a probability distribution. Recently, however, Tipping and Bishop [1999b] and Roweis [1998] have shown how PCA can be reformulated as the maximum likelihood solution of a specific latent variable model, as follows. Firstly, a q -dimensional latent variable \mathbf{x} is introduced whose prior distribution is a zero mean Gaussian $P(\mathbf{x}) = \mathcal{N}(\mathbf{0}, \mathbf{I}_q)$ and \mathbf{I}_q is the q -dimensional unit matrix. The observed variable \mathbf{t} is then defined as a linear transformation of \mathbf{x} with additive Gaussian noise $\mathbf{t} = \mathbf{W}\mathbf{x} + \boldsymbol{\mu} + \boldsymbol{\epsilon}$ where \mathbf{W} is a $d \times q$ matrix, $\boldsymbol{\mu}$ is a d -dimensional vector and $\boldsymbol{\epsilon}$ is a zero-mean Gaussian-distributed vector with covariance $\tau^{-1}\mathbf{I}_d$ (where τ is the precision). Thus, $P(\mathbf{t} | \mathbf{x}) = \mathcal{N}(\mathbf{W}\mathbf{x} + \boldsymbol{\mu}, \tau^{-1}\mathbf{I}_d)$. The marginal distribution of the observed variable is then given by the convolution of two Gaussians and is itself Gaussian

$$P(\mathbf{t}) = \int P(\mathbf{t} | \mathbf{x})P(\mathbf{x}) d\mathbf{x} = \mathcal{N}(\boldsymbol{\mu}, \mathbf{C}) \quad (3.2)$$

where the covariance matrix $\mathbf{C} = \mathbf{W}\mathbf{W}^T + \tau^{-1}\mathbf{I}_d$. The model of Equation 3.2 represents a constrained Gaussian distribution governed by the parameters $\boldsymbol{\mu}$, \mathbf{W} and τ .

It was shown by Tipping and Bishop [1999b] that the stationary points of the log likelihood with respect to \mathbf{W} satisfy

$$\mathbf{W}_{\text{ML}} = \mathbf{U}_q(\boldsymbol{\Lambda}_q - \tau^{-1}\mathbf{I}_q)^{1/2} \quad (3.3)$$

where the columns of \mathbf{U}_q are eigenvectors of \mathbf{S} , with corresponding eigenvalues in the diagonal matrix $\boldsymbol{\Lambda}_q$. It was also shown that the *maximum* of the likelihood is achieved when the q largest eigenvalues are chosen, so that the columns of \mathbf{U}_q correspond to the *principal* eigenvectors, with all other choices of eigenvalues corresponding to saddle points. The maximum

likelihood solution for τ is then given by

$$\frac{1}{\tau_{\text{ML}}} = \frac{1}{d-q} \sum_{i=q+1}^d \lambda_i \quad (3.4)$$

which has a natural interpretation as the average variance lost per discarded dimension. The density model (Equation 3.2) thus represents a probabilistic formulation of PCA. It is easily verified that conventional PCA is recovered by treating τ as a parameter and taking the limit $\tau \rightarrow \infty$.

Probabilistic PCA has been successfully applied to problems in data compression, density estimation and data visualisation, and has been extended to mixture and hierarchical mixture models [Bishop and Tipping 1998; Tipping and Bishop 1999a,b]. As with conventional PCA, however, the model itself provides no mechanism for determining the value of the latent-space dimensionality q . For $q = d - 1$ the model is equivalent to a full-covariance Gaussian distribution², while for $q < d - 1$ it represents a constrained Gaussian in which the variance in the remaining $d - q$ directions is modelled by the single parameter τ . Thus, the choice of q corresponds to a problem in model complexity optimisation. In principal *cross-validation*³ to compare all possible values of q offers a possible approach. However, maximum likelihood estimation is highly biased (leading to ‘overfitting’) and so in practice excessively large data sets would be required and the procedure would become computationally intractable.

3.2.2 Bayesian PCA

The issue of model complexity can be handled naturally within a Bayesian paradigm. Armed with the probabilistic reformulation of PCA defined in Section 3.2.1, a Bayesian treatment of PCA is obtained by first introducing prior distributions over the parameters $\boldsymbol{\mu}$, \mathbf{W} and τ . A key goal is to control the effective dimensionality of the latent space (corresponding to the number of retained principal components). Furthermore, it is desirable to avoid discrete model selection and hence continuous hyper-parameters are introduced to determine automatically an appropriate *effective* dimensionality for the latent space as part of the process of Bayesian inference. This is achieved by introducing a hierarchical prior $P(\mathbf{W} | \boldsymbol{\alpha})$ over the matrix \mathbf{W} , governed by a q -dimensional vector of hyper-parameters $\boldsymbol{\alpha} = \{\alpha_1, \dots, \alpha_q\}$. Each hyper-parameter controls one of the columns of the matrix \mathbf{W} through a conditional Gaussian distribution of the form

$$P(\mathbf{W} | \boldsymbol{\alpha}) = \prod_{i=1}^q \left(\frac{\alpha_i}{2\pi} \right)^{d/2} \exp \left\{ -\frac{1}{2} \alpha_i \|\mathbf{w}_i\|^2 \right\} \quad (3.5)$$

²This follows from the fact that the $q - 1$ linearly independent columns of \mathbf{W} have independent variances along $q - 1$ directions, while the variance along the remaining direction is controlled by τ .

³For a definition of cross-validation, see Bishop [1995, pp. 372–375].

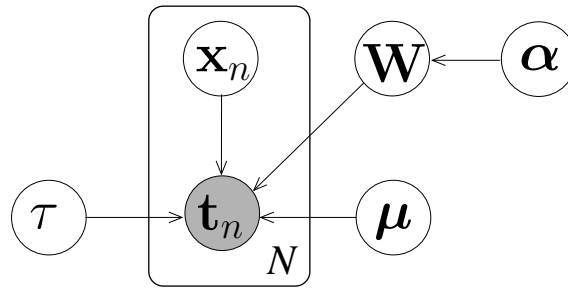


Figure 3.1: The Bayesian network for the Principal Component Analysis model. Each of the N observed data vectors \mathbf{t}_n has a corresponding low-dimensional representation \mathbf{x}_n . The columns of the matrix \mathbf{W} correspond to directions in the latent space and are equivalent to the principal components. The hyper-parameter α controls which of the principal components are ‘switched off’ and so provides a form of automatic relevance determination.

where $\{\mathbf{w}_i\}$ are the columns of \mathbf{W} . This form of prior is motivated by the framework of *automatic relevance determination* (ARD) introduced in the context of neural networks by Neal [1994] and MacKay [1995]. Each α_i controls the inverse variance of the corresponding \mathbf{w}_i , so that if a particular α_i has a posterior distribution concentrated at large values, the corresponding \mathbf{w}_i will tend to be small, and that direction in latent space will be effectively ‘switched off’. The dimensionality of the latent space is set to its maximum possible value $q = d - 1$.

The specification of the Bayesian model is completed by defining the remaining priors to have the form

$$P(\boldsymbol{\mu}) = \mathcal{N}(\boldsymbol{\mu} | \mathbf{0}, \beta^{-1}\mathbf{I}) \quad (3.6)$$

$$P(\boldsymbol{\alpha}) = \prod_{i=1}^q \Gamma(\alpha_i | a_\alpha, b_\alpha) \quad (3.7)$$

$$P(\tau) = \Gamma(\tau | a_\tau, b_\tau). \quad (3.8)$$

Here $\mathcal{N}(\mathbf{x} | \mathbf{m}, \boldsymbol{\Sigma})$ denotes a multivariate normal distribution over \mathbf{x} with mean \mathbf{m} and covariance matrix $\boldsymbol{\Sigma}$. Similarly, $\Gamma(x | a, b)$ denotes a Gamma distribution over x as defined in Appendix A.3. Broad priors are obtained by setting $a_\alpha = b_\alpha = a_\tau = b_\tau = 10^{-3}$ and $\beta = 10^{-3}$.

The Bayesian network for the complete model is shown in Figure 3.1. The model is conjugate-exponential and so the variational message passing framework can be applied immediately to train the model for a particular data set; for example, it can be applied to train the model on the toy data set of the following example.

Example 3.1: PCA on 3-dimensional manifold in 10-dimensions

As an illustration of the role of the hyperparameters in determining model complexity, consider a data set consisting of 300 points in 10 dimensions, in which the data is drawn from a Gaussian distribution having standard deviation 1.0 in 3 directions and standard deviation 0.5 in the remaining 7 directions. The result

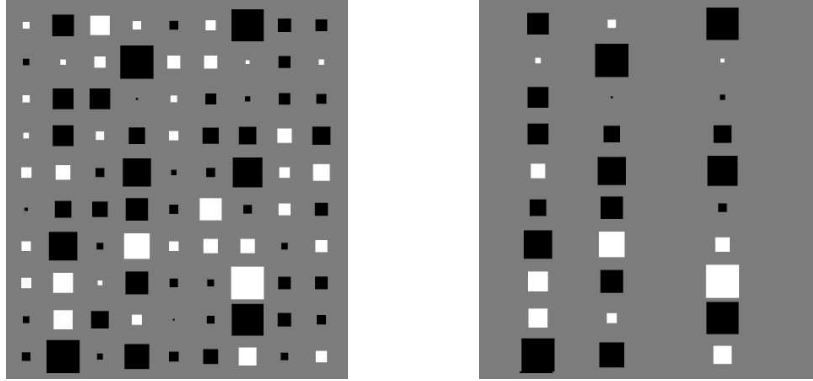


Figure 3.2: Hinton diagrams of the matrix \mathbf{W} for a data set in 10 dimensions having $m = 3$ directions with larger variance than the remaining 7 directions. The area of each square is proportional to the magnitude of the corresponding matrix element, and the squares are white for positive values and black for negative values. The left plot shows \mathbf{W}_{ML} from maximum likelihood PCA while the right plot shows the posterior mean $\langle \mathbf{W} \rangle$ from the Bayesian approach, showing how the model is able to discover the appropriate dimensionality by suppressing the 6 surplus degrees of freedom.

of fitting both maximum likelihood and Bayesian PCA models is shown in Figure 3.2. In this case the Bayesian model has an effective dimensionality of $q_{\text{eff}} = 3$, as expected, and an inferred noise standard deviation of roughly 0.5.

Effective dimensionality and data set size

Given a latent space of dimensionality q , the effective dimensionality of the space which will be inferred will depend on the number of data points available N . If the number of data points is too small, there may not be sufficient data to justify retaining all the latent space dimensions and some may be switched off. The effective dimensionality will therefore be lower than the actual dimensionality.

To investigate this relationship, consider data sets in 10 dimensions whose data are drawn from a multivariate Gaussian distribution with standard deviations of $\{1.0, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1\}$ in each of the 10 dimensions. Data set sizes from $N = 40$ to 700 were used with 50 data sets being generated of each size. A Bayesian PCA model was trained on each data set and the effective dimensionality q_{eff} recorded. This effective dimensionality corresponds to the number of dimensions with large α values (where ‘large’ was defined to be greater than a quarter of the maximum value of any α). The average effective dimensionality over the 50 data sets for each value of N was found and plotted against N . The results are shown in the graph of Figure 3.3. As can be seen, hardly any data points are required to find the first few dimensions of the latent space (which have large variance) but more and more data points are required to support each additional dimension.

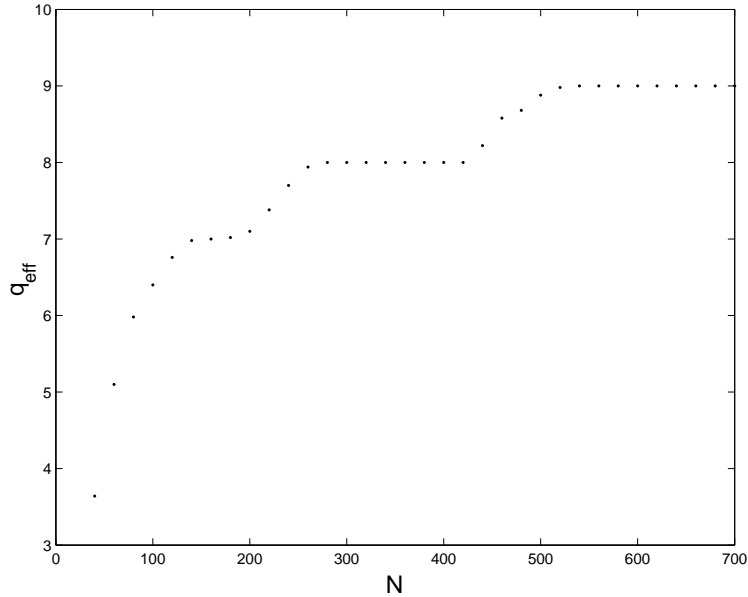


Figure 3.3: Effective dimensionality of a Bayesian PCA model for various sizes of data set. Each point represents the average inferred dimensionality for 50 data sets of size N . As the PCA model assumes that direction of smallest variance is noise, the learned dimensionality of a large data set is nine. However, smaller data sets do not provide sufficient evidence to retain all of these data dimensions; those with smaller variance are instead classified as noise.

3.2.3 Mixtures of Bayesian PCA models

Given a probabilistic formulation of PCA, it is now possible to construct a mixture distribution comprising a linear superposition of principal component analysers. If such a model were to be fitted to data using maximum likelihood we would have to choose both the number M of components and the latent space dimensionality q of the components. For moderate numbers of components and data spaces of several dimensions it quickly becomes computationally costly to use cross-validation.

Here Bayesian PCA offers an advantage in allowing the effective dimensionalities of the models to be determined automatically. Furthermore, we also wish to determine the appropriate number of components in the mixture. We do this by variational model selection (see Section 1.8.3) as an integral part of the learning procedure, as discussed in the next section.

To formulate the probabilistic model we introduce, for each data point \mathbf{t}_n , an additional M -dimensional binary latent variable \mathbf{s}_n which has one non-zero element denoting which of the M components in the mixture is responsible for generating \mathbf{t}_n . These discrete latent variables have distributions governed by hyperparameters $\boldsymbol{\pi} = \{\pi_m\}$ where $m = 1, \dots, M$,

$$P(\mathbf{s} = \delta_m | \boldsymbol{\pi}) = \pi_m \quad (3.9)$$

where δ_m denotes a vector with all elements zero except element m whose value is 1. The parameters $\boldsymbol{\pi}$ are given a Dirichlet distribution, as defined in Appendix A.5.

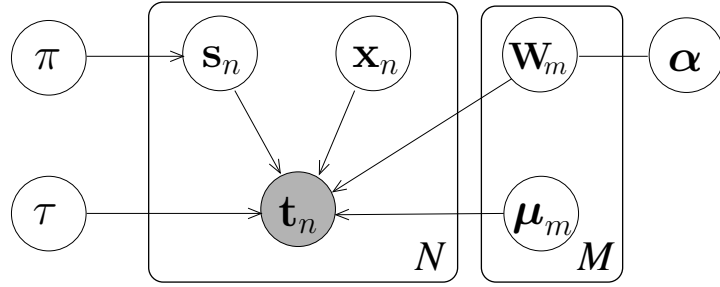


Figure 3.4: The Bayesian network for the mixture of PCA models showing the hierarchical prior over \mathbf{W} governed by the vector of shared hyper-parameters α . The left-hand plate contains N independent observations of the data vector \mathbf{t}_n (shown shaded) together with the corresponding hidden variables \mathbf{x}_n and \mathbf{s}_n , whilst the right-hand plate contains the M copies of the parameters associated with each component in the mixture.

In a simple mixture of Bayesian PCA models, each component would be free to determine its own dimensionality. A central goal of this work, however, is to model a continuous non-linear manifold. It follows that we may want the components in the mixture to have a common dimensionality whose value is a-priori unknown and which should be inferred from the data. This can be achieved within our framework by using a single set of α hyper-parameters which are *shared* by all of the components in the mixture. The probabilistic structure of the resulting model is displayed diagrammatically in the Bayesian network of Figure 3.4.

3.3 Variational Inference

Having expressed the probabilistic model as a Bayesian network, inference must be performed to determine the posterior distribution over the latent variables in the model, given a data set. As in the case of Bayesian PCA, the mixture of Bayesian PCA model of Figure 3.4 is conjugate-exponential. It follows that Variational Message Passing can be applied to find a variational approximation to the desired posterior distribution. In this initial application of VMP, the variational solution will also be derived by hand to provide both a check on the results given by VMP and an example of the amount of time that can be saved by using the variational inference framework.

3.3.1 Derivation of the variational solution

First, we assume a Q distribution of the form

$$Q(S, X, \pi, \mathbf{W}, \alpha, \boldsymbol{\mu}, \tau) = Q(S)Q(X | S)Q(\pi)Q(\mathbf{W})Q(\alpha)Q(\boldsymbol{\mu})Q(\tau) \quad (3.10)$$

where $X = \{\mathbf{x}_n\}$. The joint distribution of data and parameters is given by

$$\left[\prod_{n=1}^N P(\mathbf{t}_n | \mathbf{x}_n, \mathbf{W}, \boldsymbol{\mu}, \tau, S) \right] P(X)P(S | \pi)P(\pi)P(\mathbf{W} | \alpha)P(\alpha)P(\boldsymbol{\mu})P(\tau). \quad (3.11)$$

Using Equations 3.10 and 3.11 in Equation 1.52, and substituting for the various $P(\cdot)$ distributions, we obtain the following results for the component distributions of $Q(\cdot)$:

$$Q(X | S) = \prod_{n=1}^N Q(\mathbf{x}_n | \mathbf{s}_n) \quad (3.12)$$

$$Q(\mathbf{x}_n | \mathbf{s}_n = \delta_m) = \mathcal{N}(\mathbf{x}_n | \mathbf{m}_x^{(nm)}, \Sigma_x^{(m)}) \quad (3.13)$$

$$Q(\boldsymbol{\mu}) = \prod_{m=1}^M \mathcal{N}(\boldsymbol{\mu}_m | \mathbf{m}_\mu^{(m)}, \Sigma_\mu^{(m)}) \quad (3.14)$$

$$Q(\mathbf{W}) = \prod_{m=1}^M \prod_{k=1}^d \mathcal{N}(\tilde{\mathbf{w}}_{km} | \mathbf{m}_w^{(km)}, \Sigma_w^{(m)}) \quad (3.15)$$

$$Q(\boldsymbol{\alpha}) = \prod_{m=1}^M \prod_{i=1}^q \Gamma(\alpha_{mi} | \tilde{a}_\alpha, \tilde{b}_\alpha^{(mi)}) \quad (3.16)$$

$$Q(\tau) = \Gamma(\tau | \tilde{a}_\tau, \tilde{b}_\tau) \quad (3.17)$$

$$Q(\Pi) = \prod_{m=1}^M \text{Dir}(\pi_m | \tilde{u}^{(m)}) \quad (3.18)$$

$$Q(S) = \prod_{n=1}^N Q(\mathbf{s}_n) \quad (3.19)$$

where $\tilde{\mathbf{w}}_k$ denotes a column vector corresponding to the k th row of \mathbf{W} . Here I have defined:

$$\mathbf{m}_x^{(nm)} = \langle \tau \rangle \Sigma_x^{(m)} \langle \mathbf{W}_m^T \rangle (\mathbf{t}_n - \langle \boldsymbol{\mu}_m \rangle) \quad (3.20)$$

$$\Sigma_x^{(m)} = (\mathbf{I}_q + \langle \tau \rangle \langle \mathbf{W}_m^T \mathbf{W}_m \rangle)^{-1} \quad (3.21)$$

$$\mathbf{m}_\mu^{(m)} = \Sigma_\mu^{(m)} \langle \tau \rangle \sum_{n=1}^N \langle s_{nm} \rangle (\mathbf{t}_n - \langle \mathbf{W}_m \rangle \langle \mathbf{x}_n | m \rangle) \quad (3.22)$$

$$\Sigma_\mu^{(m)} = \left(\beta + \langle \tau \rangle \sum_{n=1}^N \langle s_{nm} \rangle \right)^{-1} \mathbf{I}_d \quad (3.23)$$

$$\mathbf{m}_w^{(km)} = \Sigma_w \langle \tau \rangle \sum_{n=1}^N \langle s_{nm} \rangle \langle \mathbf{x}_n | m \rangle (t_{nk} - \langle \mu_k \rangle) \quad (3.24)$$

$$\Sigma_w^{(m)} = \left(\text{diag} \langle \boldsymbol{\alpha}_m \rangle + \langle \tau \rangle \sum_{n=1}^N \langle s_{nm} \rangle \langle \mathbf{x}_n \mathbf{x}_n^T | m \rangle \right)^{-1} \quad (3.25)$$

$$\tilde{a}_\alpha = a_\alpha + \frac{d}{2} \quad \tilde{b}_\alpha^{(mj)} = b_\alpha + \frac{\langle \|\mathbf{w}_{mj}\|^2 \rangle}{2} \quad \tilde{a}_\tau = a_\tau + \frac{Nd}{2} \quad (3.26)$$

$$\begin{aligned} \tilde{b}_\tau &= b_\tau + \frac{1}{2} \sum_{n=1}^N \sum_{m=1}^M \langle s_{nm} \rangle \{ \|\mathbf{t}_n\|^2 + \langle \|\boldsymbol{\mu}_m\|^2 \rangle + \text{Tr}(\langle \mathbf{W}_m^T \mathbf{W}_m \rangle \langle \mathbf{x}_n \mathbf{x}_n^T | m \rangle) \\ &\quad + 2 \langle \boldsymbol{\mu}_m^T \rangle \langle \mathbf{W}_m \rangle \langle \mathbf{x}_n | m \rangle - 2 \mathbf{t}_n^T \langle \mathbf{W}_m \rangle \langle \mathbf{x}_n | m \rangle - 2 \mathbf{t}_n^T \langle \boldsymbol{\mu}_m \rangle \} \end{aligned} \quad (3.27)$$

$$\tilde{u}^{(m)} = u_m + \sum_{n=1}^N \langle s_{nm} \rangle \quad (3.28)$$

$$\begin{aligned} \log Q(\mathbf{s}_n = \delta_m) &= \langle \log \pi_m \rangle - \frac{1}{2} \langle \mathbf{x}_n^T \mathbf{x}_n | m \rangle - \frac{1}{2} \langle \tau \rangle \{ \|\mathbf{t}_n\|^2 + \langle \|\boldsymbol{\mu}_m\|^2 \rangle \\ &+ \text{Tr}(\langle \mathbf{W}_m^T \mathbf{W}_m \rangle \langle \mathbf{x}_n \mathbf{x}_n^T | m \rangle) + 2 \langle \boldsymbol{\mu}_m^T \rangle \langle \mathbf{W}_m \rangle \langle \mathbf{x}_n | m \rangle \end{aligned} \quad (3.29)$$

$$- 2 \mathbf{t}_n^T \langle \mathbf{W}_m \rangle \langle \mathbf{x}_n | m \rangle - 2 \mathbf{t}_n^T \langle \boldsymbol{\mu}_m \rangle \} + \frac{1}{2} \log |\boldsymbol{\Sigma}_{\mathbf{x}}^{(m)}| + \text{const.} \quad (3.30)$$

where $\text{diag}\langle \boldsymbol{\alpha} \rangle$ denotes a diagonal matrix whose diagonal elements are given by $\langle \alpha_i \rangle$. The constant in $\log Q(\mathbf{s}_n = \delta_m)$ is found simply by summing and normalising. Note also that $\langle \mathbf{x}_n | m \rangle$ denotes an average with respect to $Q(\mathbf{x}_n | \mathbf{s}_n = \delta_m)$.

The framework also permits a direct evaluation of the posterior distribution over the number M of components in the mixture (assuming a suitable prior distribution, for example a uniform distribution up to some maximum value). However, in order to reduce the computational complexity of the inference problem we adopt an alternative approach based on model comparison using the numerically evaluated lower bound $\mathcal{L}(Q)$ which approximates the log model probability $\log P(\mathbf{t})$. Our optimisation mechanism dynamically adapts the value of M through a scheme involving the addition and deletion of components (as in Ueda et al. [1999]; Ghahramani and Beal [1999]).

One of the limitations of fitting conventional Gaussian mixture models by maximum likelihood is that there are singularities in the likelihood function in which a component's mean coincides with one of the data points while its covariance shrinks to zero. Such singularities do not arise in the Bayesian framework due to the use of priors over model parameters.

3.4 Results

In order to demonstrate the operation of the algorithm, its behaviour has been explored using synthetic data before being applied to real data sets.

3.4.1 Synthetic data: noisy sinusoid

Figure 3.5 shows a non-linear one dimensional manifold embedded in two dimensions, together with the result of fitting a Bayesian PCA mixture model. The lines represent the non-zero principal directions of each component in the mixture. At convergence the model had eight components, having a common effective dimensionality of one.

3.4.2 Synthetic data: noisy sphere

Figure 3.6 shows synthetic data from a noisy two-dimensional sphere in three dimensions together with the converged model, which has 12 components with effective dimensionality of two. Similar results with synthetic data are robustly obtained when embedding low-dimensional non-linear manifolds in spaces of higher dimensionality.

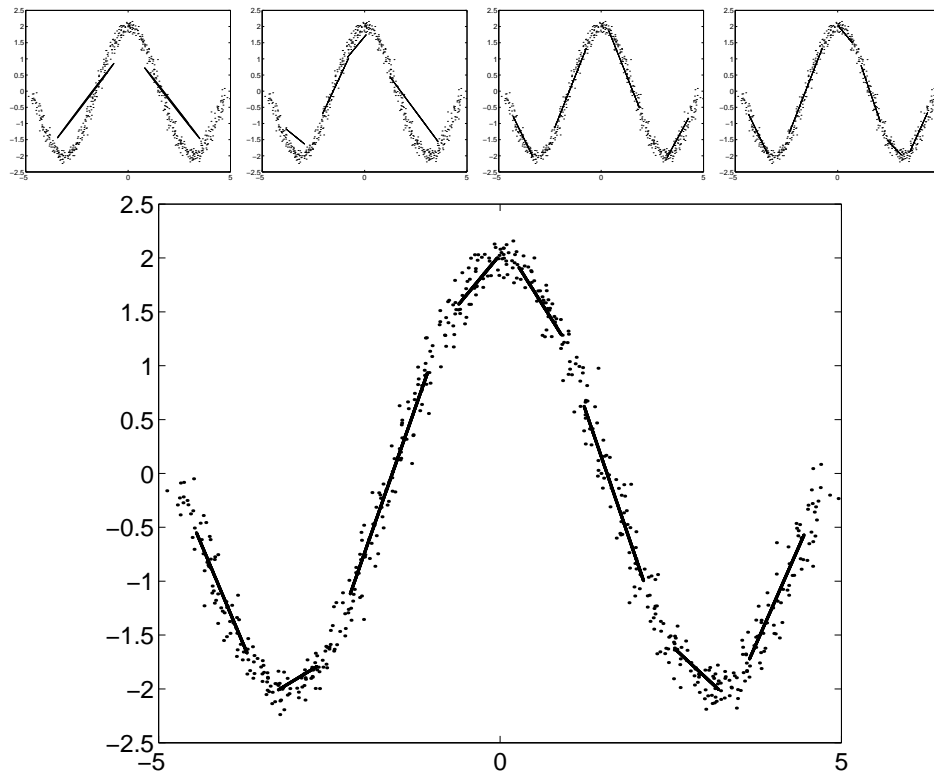


Figure 3.5: Example of a Bayesian PCA mixture model fitted to a highly non-linear one dimensional manifold. The top four graphs are snapshots of the model taken during the inference process. The bottom graph is the converged result.

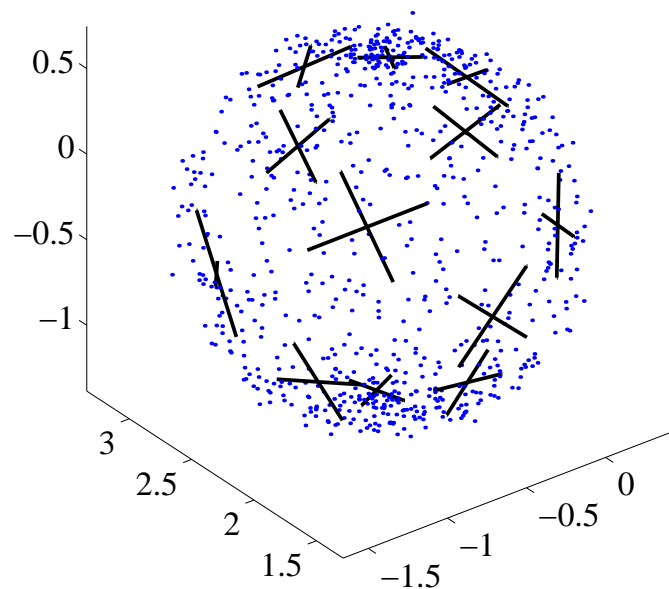


Figure 3.6: Example of a Bayesian PCA mixture model fitted to a two-dimensional manifold corresponding to the (noisy) surface of a sphere. The model has been fitted using twelve PCA components, each with an effective dimensionality of two, showing that the dimensionality of the manifold has been correctly determined.

3.4.3 Faces data set

Let us now apply the framework to the problem of modelling the manifold of a data set of face images. The data set used is a combination of images from the Yale face database and the University of Stirling database. The training set comprises 276 training images, which have been cropped, sub-sampled to 26×15 , and normalised pixel-wise to zero mean and unit variance. The test set consists of a further 100 face images, together with 200 non-face images, taken from the Corel database, all of which were pre-processed in the same way as the training data.

The converged Bayesian PCA mixture model has four components, having a common dimensionality of five, as emphasised by the Hinton diagram of the shared α hyper-parameters shown in Figure 3.7.



Figure 3.7: Hinton diagram showing the inverses of the α hyper-parameters (corresponding to the variances of the principal components) indicating a manifold with an intrinsic dimensionality of five.

The small number of components and low dimensionality of the resultant model suggests that the small size of the data set has restricted the complexity of the learned model. In order to see how well the model has captured the manifold, we first run the model generatively to give some sample synthetic images, as shown in Figure 3.8. The extent to which the model has succeeded in modelling the manifold of faces can be quantified by using the density model to classify the images in the test set as faces versus non-faces. To do this, the density under the model was evaluated for each test image and if this density exceeded some threshold the image was classified as a face. The threshold value determines the trade-off between false negatives and false positives, leading to a Receiver Operating Curve (ROC), as shown in Figure 3.9. For comparison the corresponding ROC curves for a single maximum likelihood PCA model for a range of different q values are also shown. It can be seen that moving from a linear model (PCA) to a non-linear model (a Bayesian PCA mixture) gives a significant improvement in

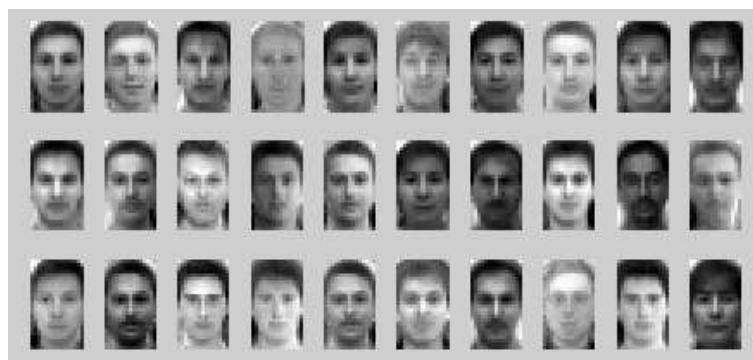


Figure 3.8: Synthetic faces obtained by running the learned mixture distribution generatively.

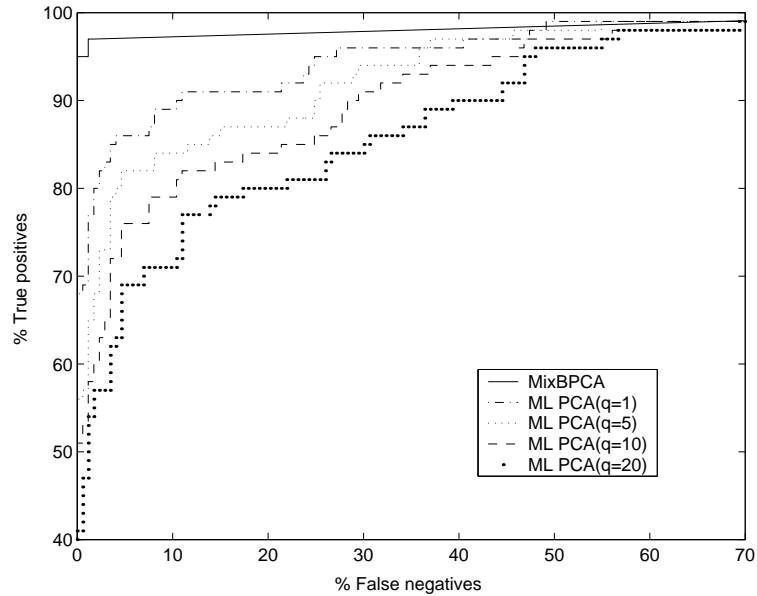


Figure 3.9: ROC curves for classifying images as faces versus non-faces for the Bayesian PCA mixture model (MixBPCA), together with the corresponding results for a maximum likelihood PCA model (ML PCA) with various values for q the number of retained principal components. This highlights the significant improvement in classification performance in going from a linear PCA model to a non-linear mixture model. In the maximum likelihood model it is necessary to repeat inference with a range of values of q ; the ability of the Bayesian approach to determine latent space dimensionality avoids this problem.

classification performance. This result also highlights the fact that the Bayesian approach avoids the need to set parameters such as q by exhaustive exploration.

3.4.4 Handwritten digits data set

As a second application of the framework, a model was constructed of the manifolds of images of hand-written digits. A data set taken from the CEDAR U.S. Postal Service database was used, comprising 11,000 images (equally distributed over the ten digits) each of which is 8×8 greyscale, together with a similar independent test set of 2711 images. Figure 3.10 shows synthetic images generated from a Bayesian PCA mixture model fitted to the training set.

The learned model achieved 4.83% error rate on the test set. For comparison we note that Tipping and Bishop [1999a] used the same training and test sets with a maximum likelihood mixture of probabilistic principal component analysers. The training set in this case was itself subdivided into training plus validation sets. For each of the ten digit models considerable computational effort was expended in finding the optimum values of M (the number of components in the mixture) and q (the dimensionality of the latent spaces) by evaluation of performance on the validation set. This approach achieved 4.61% error rate on the test set, which is comparable with the result obtained from the single run of the Bayesian PCA mixture model.



Figure 3.10: Digits synthesised from each of the ten trained Bayesian PCA mixture model by running the models generatively.

3.4.5 Image compression

As a final application, the model was used to perform image compression by block transform image coding. A 480×640 test image, shown in Figure 3.11a, was divided into 4800 8×8 blocks. The blocks were formatted into 64-dimensional vectors. Half of these vectors, corresponding to the left hand side of the image, were used to train the model. The remaining vectors were used to test the trained model. For comparison, a standard PCA model and a vector quantised PCA (VQPCA) model were also tested. Vector quantised PCA is described in Kambhatla and Leen [1997] and is summarised in Algorithm 3.1.

Algorithm 3.1 Vector Quantised PCA (VQPCA)

1. Select cluster centres at random from points in the data set and assign all data points to the nearest cluster centre.
 2. Set matrices W_i to the principal axes of the covariance matrix of cluster i .
 3. Assign data points to the cluster that best reconstructs them and set new cluster centres to the mean of points assigned to that cluster.
 4. Repeat from step 2 to until the cluster allocations are constant.
-

In each method, the reduced dimensionality representation of each block was quantised to give a final bit rate of 0.5 bits per pixel (which gives a compression ratio of 16 to 1). This includes the cost of coding the component label in the mixture model. The bits were allocated equally to each transform variable.

Figure 3.11b shows an enlarged part of original image and the corresponding reconstructions for standard PCA, mixture of Bayesian PCA and VQPCA along with the reconstruction errors of each. The mixture of Bayesian PCA model gives the smallest reconstruction error and has the fewest reconstruction artefacts (for example, examine the curve of the underside of the bridge in the reconstructed images).

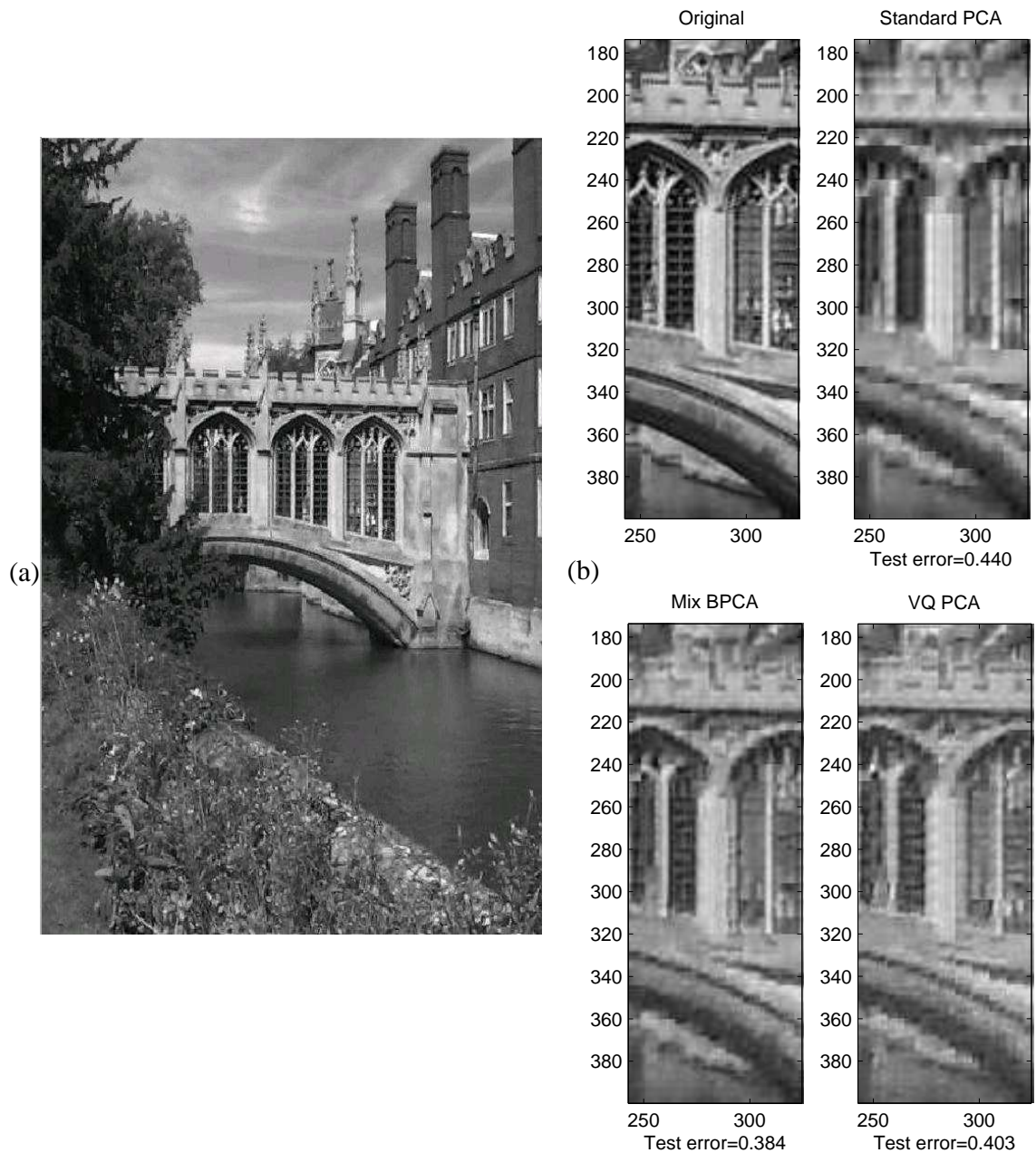


Figure 3.11: (a) The original image used to test the compression algorithm. (b) Detail of the original and reconstructed images for various compression methods. The normalised test errors shown are the reconstruction errors for the whole image.

3.5 Discussion

In this chapter, I have introduced a fully probabilistic approach to modelling manifolds of images, in which an appropriate model complexity, as well as the manifold's intrinsic dimensionality, can be inferred from the data. The use of Variational Message Passing has allowed inference in this model to be performed automatically. As a comparison, the variational update equations have also been derived by hand, showing the considerable additional time and effort required to do this. Results on data sets of images of faces and handwritten digits and on an image compression example demonstrate the practical feasibility of the system as well as improved performance compared to previous approaches.

An important advantage of the system is that there are no significant adjustable parameters in the model which need to be set by the user. The model complexity is inferred from the data and, since no model optimisation is required, the model can be trained just once on the data set, avoiding the need for computationally expensive cross-validation.

BIBLIOGRAPHY

- D. Ackley, G. Hinton, and T. Sejnowski. A learning algorithm for Boltzmann machines. *Cognitive Science*, 9:147–169, 1985.
- H. Attias. A variational Bayesian framework for graphical models. In S. Solla, T. K. Leen, and K-L Muller, editors, *Advances in Neural Information Processing Systems*, volume 12, pages 209–215, Cambridge MA, 2000. MIT Press.
- Z. Bar-Joseph, D. Gifford, and T. Jaakkola. Fast optimal leaf ordering for hierarchical clustering. *Bioinformatics*, 17:S22–29, 2001.
- D. Barber and C. M. Bishop. Variational learning in Bayesian neural networks. In C. M. Bishop, editor, *Generalization in Neural Networks and Machine Learning*. Springer Verlag, 1998.
- K. J. Bathe. *Finite Element Procedures*. Prentice-Hall, Englewood Cliffs, NJ, 1996.
- Rev. T. Bayes. An essay towards solving a problem in the doctrine of chances. In *Philosophical Transactions of the Royal Society*, volume 53, pages 370–418, 1763.
- A. Ben-Dor, R. Shamir, and Z. Yakhini. Clustering gene expression patterns. *Journal of Computational Biology*, 6(3/4):281–297, 1999.
- J. M. Bernardo and A.F.M. Smith. *Bayesian Theory*. John Wiley and Sons, New York, 1994.
- C. M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.
- C. M. Bishop. Bayesian PCA. In S. A. Solla M. S. Kearns and D. A. Cohn, editors, *Advances in Neural Information Processing Systems*, volume 11, pages 382–388. MIT Press, 1999a.
- C. M. Bishop. Variational principal components. In *Proceedings Ninth International Conference on Artificial Neural Networks, ICANN'99*, volume 1, pages 509–514. IEE, 1999b.
- C. M. Bishop and M. E. Tipping. A hierarchical latent variable model for data visualization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(3):281–293, 1998.
- C. M. Bishop and M. E. Tipping. Variational Relevance Vector Machines. In *Proceedings of 16th Conference in Uncertainty in Artificial Intelligence*, pages 46–53. Morgan Kaufmann, 2000.

- C. M. Bishop and J. M. Winn. Non-linear Bayesian image modelling. In *Proceedings Sixth European Conference on Computer Vision*, volume 1, pages 3–17. Springer-Verlag, 2000.
- C. M. Bishop and J. M. Winn. Structured variational distributions in VIBES. In *Proceedings Artificial Intelligence and Statistics*, Key West, Florida, 2003. Society for Artificial Intelligence and Statistics.
- C. M. Bishop, J. M. Winn, and D. Spiegelhalter. VIBES: A variational inference engine for Bayesian networks. In *Advances in Neural Information Processing Systems*, volume 15, 2002.
- M. J. Black and Y. Yacoob. Recognizing facial expressions under rigid and non-rigid facial motions. In *International Workshop on Automatic Face and Gesture Recognition, Zurich*, pages 12–17, 1995.
- C. Bregler and S.M. Omohundro. Nonlinear manifold learning for visual speech recognition. In *Fifth International Conference on Computer Vision*, pages 494–499, Boston, Jun 1995.
- J. Buhler, T. Ideker, and D. Haynor. Dapple: Improved techniques for finding spots on DNA microarrays. Technical report, University of Washington, 2000.
- R. Choudrey, W. Penny, and S. Roberts. An ensemble learning approach to independent component analysis. In *IEEE International Workshop on Neural Networks for Signal Processing*, 2000.
- G. F. Cooper. The computational complexity of probabilistic inference using Bayesian belief networks. *Artificial Intelligence*, 42:393–405, 1990.
- T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. Active shape models — their training and application. In *Computer vision, graphics and image understanding*, volume 61, pages 38–59, 1995.
- R. G. Cowell, A. P. Dawid, S. L. Lauritzen, and D. J. Spiegelhalter. *Probabilistic Networks and Expert Systems*. Statistics for Engineering and Information Science. Springer-Verlag, 1999.
- R. T. Cox. Probability, frequency and reasonable expectation. *American Journal of Physics*, 14(1):1–13, 1946.
- P. Dagum and M. Luby. Approximating probabilistic inference in Bayesian belief networks is NP-hard. *Artificial Intelligence*, 60:141–153, 1993.
- A. Darwiche. Conditioning methods for exact and approximate inference in causal networks. In *Eleventh Annual Conference on Uncertainty in Artificial Intelligence*. Morgan Kaufmann, August 1995.

- S. Dudoit, Y. H. Yang, Matthew J. Callow, and T. P. Speed. Statistical methods for identifying differentially expressed genes in replicated cDNA microarray experiments. Technical report, Department of Biochemistry, Stanford University School of Medicine, 2000.
- M. Eisen, P. Spellman, D. Botstein, and P. Brown. Cluster analysis and display of genome-wide expression patterns. In *Proceedings of National Academy of Science*, volume 95, pages 14863–14867, 1998.
- M.B. Eisen and P.O. Brown. DNA arrays for analysis of gene expression. *Methods in Enzymology*, 303:179–205, 1999.
- B.S. Everitt and D.J. Hand. *Finite Mixture Distributions*. Chapman and Hall, London, 1981.
- R.P. Feynman. *Statistical Mechanics*. W. A. Benjamin, Inc., MA, 1972.
- B. Frey. *Graphical models for machine learning and digital communications*. MIT Press, Cambridge, MA, 1998.
- B. Frey and N. Jojic. Transformed component analysis: joint estimation of spatial transformations and image components. In *Seventh International Conference on Computer Vision*, pages 1190–1196, 1999.
- B. Frey, F. Kschischang, H. Loeliger, and N. Wiberg. Factor graphs and algorithms. In *Proceedings of the 35th Allerton Conference on Communication, Control and Computing 1997*, 1998.
- N. Friedman, M. Linial, I. Nachman, and D. Pe’er. Using Bayesian networks to analyze expression data. In *RECOMB*, pages 127–135, 2000.
- R.G. Gallager. Low density parity check codes. *IRE Trans. Info. Theory*, IT-8:21–28, Jan 1962.
- R.G. Gallager. *Low density parity check codes*. Number 21 in Research monograph series. MIT Press, Cambridge, MA, 1963.
- S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(1): 721–741, 1984.
- Z. Ghahramani and M. J. Beal. Variational inference for Bayesian mixture of factor analysers. In *Advances in Neural Information Processing Systems*, volume 12, 1999.
- Z. Ghahramani and M. J. Beal. Propagation algorithms for variational Bayesian learning. In T. K. Leen, T. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems*, volume 13, Cambridge MA, 2001. MIT Press.

- W. R. Gilks and P. Wild. Adaptive rejection sampling for Gibbs sampling. *Applied Statistics*, 41(2):337–348, 1992.
- A. Hartemink, D. Gifford, T. Jaakkola, and R. Young. Using graphical models and genomic expression data to statistically validate models of genetic regulatory networks. In *Pacific Symposium on Biocomputing*, volume 6, pages 422–433, 2001.
- T. Heap and D. Hogg. Wormholes in shape space: Tracking through discontinuous changes in shape. In *Sixth International Conference on Computer Vision*, pages 344–349, 1998.
- P. Hegde, R. Qi, R. Abernathy, C. Gay, S. Dharap, R. Gaspard R, J. Earle-Hughes, E. Snesrud, N. H. Lee, and J. Quackenbush. A concise guide to cDNA microarray analysis. *Biotechniques*, 29(3):548–562, 2000.
- T. Heskes. Stable fixed points of loopy belief propagation are minima of the Bethe free energy, 2002.
- G. E. Hinton and D. van Camp. Keeping neural networks simple by minimizing the description length of the weights. In *Proceedings of the Sixth Annual Conference on Computational Learning Theory*, pages 5–13, 1993.
- G. E. Hinton and R. S. Zemel. Autoencoders, minimum description length, and Helmholtz free energy. In *Advances in Neural Information Processing Systems*, volume 6, 1994.
- G. Hori, M. Inoue, S. Nishimura, and H. Nakahara. Blind gene classification on ICA of microarray data. In *ICA 2001*, pages 332–336, 2001.
- T. Jaakkola. *Variational Methods for Inference and Estimation in Graphical Models*. PhD thesis, MIT, 1997.
- F. Jensen. *An introduction to Bayesian networks*. UCL Press, 1996.
- M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul. An introduction to variational methods for graphical models. In M. I. Jordan, editor, *Learning in Graphical Models*, pages 105–162. Kluwer, 1998.
- N. Kambhatla and T.K. Leen. Dimension reduction by local principal component analysis. *Neural Computation*, 9(7):1493–1516, 1997.
- R. Kinderman and J. L. Snell. Markov random fields and their applications. *American Mathematical Society*, 1:1–142, 1980.
- F. R. Kschischang, B. Frey, and H.-A. Loeliger. Factor graphs and the sum-product algorithm. *IEEE Trans. Inform. Theory*, 47(2):498–519, 2001.
- S. Kullback. *Information Theory and Statistics*. Dover Publications, New York, 1959.

- S. Kullback and R. A. Leibler. On information and sufficiency. *Annals of Mathematical Statistics*, 22:79–86, 1951.
- S. L. Lauritzen and D.J. Spiegelhalter. Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society*, 50:157–224, 1988.
- S. L. Lauritzen and N. Wermuth. Graphical models for associations between variables, some of which are qualitative and some quantitative. *Annals of Statistics*, 17:31–57, 1989.
- N. Lawrence, M. Milo, M. Niranjana, P. Rashbass, and S. Soullier. Reducing the variability in microarray image processing by Bayesian inference. Technical report, Department of Computer Science, University of Sheffield, 2002.
- D. B. Lenat. CYC: A large-scale investment in knowledge infrastructure. *Communications of the ACM*, 38(11):33–38, 1995.
- D. J. Lunn, A. Thomas, N. G. Best, and D. J. Spiegelhalter. WinBUGS – a Bayesian modelling framework: concepts, structure and extensibility. *Statistics and Computing*, 10:321–333, 2000. <http://www.mrc-bsu.cam.ac.uk/bugs/>.
- D. J. C. MacKay. Probable networks and plausible predictions – a review of practical Bayesian methods for supervised neural networks. *Network: Computation in Neural Systems*, 6(3): 469–505, 1995.
- D. J. C. MacKay. Ensemble learning for hidden Markov models, 1997. Unpublished manuscript, Department of Physics, University of Cambridge.
- D. J. C. MacKay. Introduction to Monte Carlo methods. In M. I. Jordan, editor, *Learning in Graphical Models*. Kluwer, 1998.
- D. J. C. MacKay. *Information theory, inference, and learning algorithms*. Cambridge University Press, Cambridge, UK, 2003.
- D. J. C. MacKay and R. M. Neal. Good codes based on very sparse matrices. In *IMA: IMA Conference on Cryptography and Coding, LNCS lately (earlier: Cryptography and Coding II, Edited by Chris Mitchell, Clarendon Press, 1992)*, 1995.
- A. Martoglio, J. W. Miskin, S. K. Smith, and D. J. C. MacKay. A decomposition model to track gene expression signatures: preview on observer-independent classification of ovarian cancer. *Bioinformatics*, 18:1617–1624, 2002.
- A. Martoglio, B. D. Tom, M. Starkey, A. N. Corps, S. Charnock-Jones, and S. K. Smith. Changes in tumorigenesis- and angiogenesis-related gene transcript abundance profiles in ovarian cancer detected by tailored high density cDNA arrays. *Molecular Medicine*, 6(9): 750–765, 2000.

- R. J. McEliece, D. J. C. MacKay, and J.-F. Cheng. Turbo decoding as an instance of Pearl's Belief Propagation algorithm. *IEEE Journal on selected areas in communication*, 1997.
- G. S. Michaels, D. B. Carr, M. Askenazi, S. Fuhrman, X. Wen, and R. Somogyi. Cluster analysis and data visualization of large-scale gene expression data. In *Pacific Symposium on Biocomputing*, volume 3, pages 42–53, 1998.
- T. P. Minka. Expectation propagation for approximate Bayesian inference. In *Proceedings of the 17th Annual Conference on Uncertainty in Artificial Intelligence*, pages 362–369. Morgan Kaufmann, 2001a.
- T. P. Minka. *A family of algorithms for approximate Bayesian inference*. PhD thesis, MIT, 2001b.
- J. W. Miskin. *Ensemble Learning for Independent Component Analysis*. PhD thesis, University of Cambridge, 2000.
- J. W. Miskin and D. J. C. MacKay. Ensemble learning for blind source separation. In S. J. Roberts and R. M. Everson, editors, *ICA: Principles and Practice*. Cambridge University Press, 2000.
- B. Moghaddam. Principal manifolds and Bayesian subspaces for visual recognition. In *Seventh International Conference on Computer Vision*, pages 1131–1136, 1999.
- B. Moghaddam and A. Pentland. Probabilistic visual learning for object recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):696–710, 1997.
- V.S. Nalwa. *A Guided Tour of Computer Vision*. Addison-Wesley, 1993.
- R. M. Neal. Connectionist learning of belief networks. *Artificial Intelligence*, 56:71–113, 1992.
- R. M. Neal. Probabilistic inference using Markov chain Monte Carlo methods. Technical Report CRG-TR-93-1, Department of Computer Science, University of Toronto, Canada, 1993.
- R. M. Neal. *Bayesian Learning for Neural Networks*. PhD thesis, University of Toronto, Canada, 1994.
- R. M. Neal and G. E. Hinton. A new view of the EM algorithm that justifies incremental and other variants. In M. I. Jordan, editor, *Learning in Graphical Models*, pages 355–368. Kluwer, 1998.
- J. Pearl. Fusion, propagation and structuring in belief networks. *Artificial Intelligence*, 29: 241–288, 1986.
- J. Pearl. Evidential reasoning using stochastic simulation of causal models. *Artificial Intelligence*, 32:245–257, 1987.

- J. Pearl. *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann, San Francisco, 1988.
- S. Raychaudhuri, J. Stuart, and R. Altman. Principal Components Analysis to summarize microarray experiments: application to sporulation time series. In *Pacific Symposium on Biocomputing*, volume 5, 2000.
- S. Roweis. EM algorithms for PCA and SPCA. In Michael I. Jordan, Michael J. Kearns, and Sara A. Solla, editors, *Advances in Neural Information Processing Systems*, volume 10. MIT Press, 1998.
- J. Rustagi. *Variational Methods in Statistics*. Academic Press, New York, 1976.
- J. Sakurai. *Modern Quantum Mechanics*. Addison-Wesley, Redwood City, CA, 1985.
- L. K. Saul and M. I. Jordan. Exploiting tractable substructures in intractable networks. In D. S. Touretzky, M. C. Mozer, and M. E. Hasselmo, editors, *Advances in Neural Information Processing Systems*, volume 8, pages 486–492. MIT Press, 1996.
- E. H. Shortcliffe. *Computer-Based Medical Consultations: MYCIN*. Elsevier Science, New York, 1976.
- P. T. Spellman, G. Sherlock, M. Q. Zhang, V. R. Iyer, K. Anders, M. B. Eisen, P. O. Brown, D. Botstein, and B. Futcher. Comprehensive identification of cell cycle-regulated genes of the yeast *saccharomyces cerevisiae* by microarray hybridization. In *Molecular Biology of the Cell*, volume 9, pages 3273–3297, 1998.
- D. J. Spiegelhalter. Probabilistic reasoning in predictive expert systems. In L. N. Kanal and J. F. Lemmer, editors, *Uncertainty in Artificial Intelligence*, pages 47–68, Amsterdam, 1986. North Holland.
- P. Tamayo, D. Slonim, J. Mesirov, Q. Zhu, S. Kitareewan, E. Dmitrovsky, E. Lander, and T. Golub. Interpreting patterns of gene expression with self-organizing maps: Methods and application to hematopoietic differentiation. In *Proceedings of the National Academy of Science*, volume 96, pages 2907–2912, 1999.
- A. Thomas, D. J. Spiegelhalter, and W. R. Gilks. BUGS: A program to perform Bayesian inference using Gibbs sampling. In J. M. Bernardo, J. O. Berger, A. P. Dawid, and A. F. M. Smith, editors, *Bayesian Statistics*, Oxford: Clarendon Press, 1992.
- M. E. Tipping and C. M. Bishop. Mixtures of probabilistic principal component analyzers. *Neural Computation*, 11(2):443–482, 1999a.
- M. E. Tipping and C. M. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society, Series B*, 21(3):611–622, 1999b.

- M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86, 1991.
- N. Ueda, R. Nakano, Z. Ghahramani, and G. E. Hinton. SMEM algorithm for mixture models. In *Advances in Neural Information Processing Systems*, volume 11, 1999.
- P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition*, 2001.
- Niclas Wiberg. *Codes and Decoding on General Graphs*. PhD thesis, Linköping University, 1996.
- W. Wiegnerinck. Variational approximations between mean field theory and the junction tree algorithm. In *Uncertainty in Artificial Intelligence*. Morgan Kaufmann, 2000.
- C. K. I. Williams and C. E. Rasmussen. Gaussian processes for regression. In D. S. Touretzky, M. C. Mozer, and M. E. Hasselmo, editors, *Advances in Neural Information Processing Systems*, volume 8, pages 514–520. MIT Press, 1996.
- P. H. Winston. *Artificial Intelligence*. Addison-Wesley, third edition, 1992.
- E. P. Xing, M. I. Jordan, and S. Russell. A generalized mean field algorithm for variational inference in exponential families. In *Uncertainty in Artificial Intelligence*. Morgan Kaufmann, 2003.
- J. Yedidia, W. Freeman, and Y. Weiss. Understanding belief propagation and its generalizations. In G. Lakemeyer and B. Nebel, editors, *Exploring Artificial Intelligence in the New Millennium*. Morgan Kaufmann, 2002.
- K. Yeung, C. Fraley, A. Murua, A. Raftery, and W. Ruzzo. Model-based clustering and data transformations for gene expression data. *Bioinformatics*, 17(10):977–987, 2001.