Bayesian Feature Weighting for Unsupervised Learning, with Application to Object Recognition

Peter Carbonetto* **Nando de Freitas**

Department of Computer Science
University of British Columbia
Vancouver, BC, Canada

Paul Gustafson **Natalie Thompson**

Department of Statistics
University of British Columbia
Vancouver, BC, Canada

Abstract

We present a method for variable selection/weighting in an unsupervised learning context using Bayesian shrinkage. The basis for the model is a finite mixture of multivariate Gaussian distributions. We demonstrate how the model parameters and cluster assignments can be computed simultaneously using an efficient EM algorithm. Applying our Bayesian shrinkage model to a complex problem in object recognition (Duygulu, Barnard, de Freitas and Forsyth 2002), our experiments yield good results.

1 INTRODUCTION

The importance of unsupervised learning is growing in many research areas, including bioinformatics, data mining, object recognition, database clustering, browsing tools for digital libraries, collaborative filtering and information retrieval (Blei, Ng and Jordan 2002, Barnard and Forsyth 2001, Barnard, Duygulu and Forsyth 2001, Duygulu et al. 2002, Golub, Slonim and Tamayo 1999, Hofmann and Puzicha 1999). In most of these settings, one typically begins data analysis by weighting or selecting a subset of informative features. In computer vision, for example, investigators often assign weights to features such as colour, stereo, motion, texture and shape. Gene expression analysis with DNA micro-arrays, where one clusters vectors (patients) with thousands of entries (genes), is another instance where feature weighting plays a significant role. Since the number of patients tends to be small, knowing which genes provide more discriminatory information would be of great benefit. In these applications, variable selection is at the discretion of the investigator and may be based on prior information. In general, however, making informed decisions on the utility of variables may not be feasible so we would like to rely on schemes to select variables automatically.

Variable selection techniques for supervised learning have been widely studied. For examples, see (George and Foster 1997, Ng and Jordan 2001, Smith and Kohn 1996). Surprisingly, the same cannot be said about unsupervised learning. Aside from naive search methods, the work of (Liu, Zhang, Palumbo and Lawrence 2002) seems to be the only principled approach to this problem. Liu *et al* project the data to a lower dimension with PCA, and apply Markov chain Monte Carlo to mixture models of the data. By analytically integrating out the other parameters, they sample only the mixture allocation variables and the number of PCA components. The projection of the data to a lower dimension is needed to construct efficient Markov chains. However, variable selection has to be done in the spectral domain and, hence, one loses the intuitive notion of what actual variables are being selected.

In this paper, we present a Bayesian shrinkage model for mixture models and explain how it yields a transparent method for variable selection/weighting. The relevant variables and model parameters are computed with an efficient EM algorithm.

*Authorship in alphabetical order.

2 A MOTIVATING EXAMPLE

We begin with a simple example which illustrates the utility in variable selection for unsupervised learning.

Let the data x be distributed over a mixture of multivariate gaussians, $x \overset{iid}{\sim} \sum_{i=1}^{n_c} \lambda_c \mathcal{N}_{n_x}(\mu_c, \Sigma_c)$ where n_x is the dimension of the data, n_c is the number of mixture components, μ_c and Σ_c are the mean and covariance for mixture component c, and the cluster assignment probabilities λ_c sum to 1. For this example, we assume a total of three mixture components, diagonal covariance matrices and dimension $n_x = 6$. In addition, we set the means to

$$\mu_1 = \begin{bmatrix} 1.2 \\ 2.3 \\ 5.3 \\ 0.0 \\ 0.1 \\ 0.0 \end{bmatrix} \quad \mu_2 = \begin{bmatrix} 5.0 \\ 4.9 \\ 4.7 \\ 0.2 \\ 0.3 \\ 0.2 \end{bmatrix} \quad \mu_3 = \begin{bmatrix} 9.5 \\ 5.6 \\ 3.8 \\ 0.0 \\ 0.0 \\ 0.5 \end{bmatrix}$$

and assign the variance $\sigma^2 = [1\ 1\ 1\ 2\ 2\ 2]'$ and the mixture probablities $\lambda = [0.36\ 0.26\ 0.38]'$.

Given a training set of data points $\{x_1, x_2, ..., x_{200}\}$ sampled from the true mixture distribution, we estimate the model parameters λ and μ using EM. We assume the σ_i's are known, but the argument extends naturally to a scenario with unknown variances. From above, it is apparent that the first three features of data x are useful in varying degrees for classification. Conversely, the last three features have only a marginal effect on determining to which component in the true distribution a data point belongs.

As illustrated by the contour plots in Figure 1, including the last three irrelevant features in training exacerbates classification on the mixture components because we may detect idiosyncracies in samples x. Pruning the irrelevant features holds more promise for a generalized estimate because we will be less prone to training on artifacts in the sample data. When we verified this experimentally, classification of the 200 training examples was correct on 99.5 percent of the instances for variable selection, but only 37.5 percent of the time when trained on all the variables.

This example suggests that if anything, variable selection has a great potential benefit in classification and clustering problems, and even more so when dealing with high-dimensional data. From some simple experiments we performed, it may also be the case that variable selection is of greater help than in supervised learning. Surprisingly, almost all the literature on variable selection and shrinkage is in the regression context.

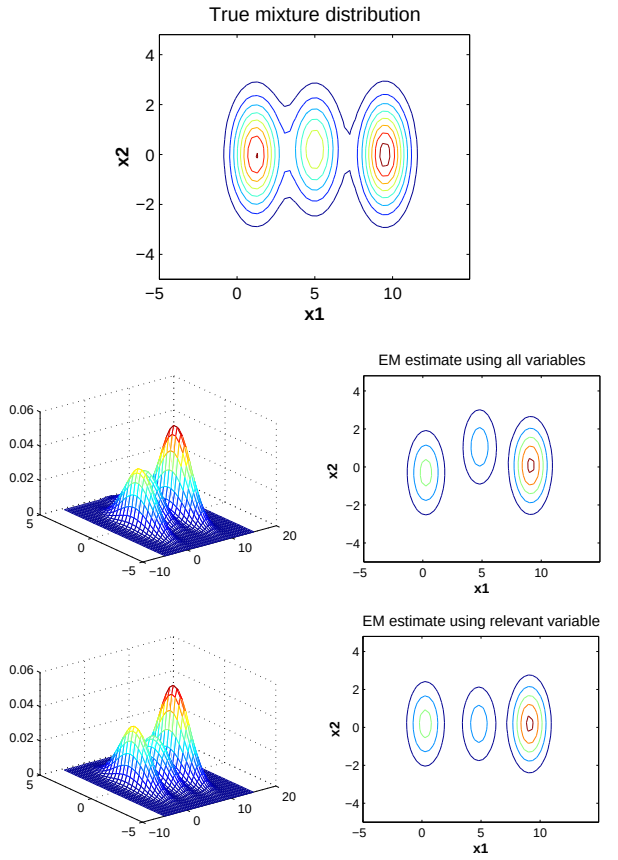

Figure 1: *A demonstration of the advantage of variable selection in two dimensions. The mixture component means in the true distribution from which we sample the data have different values over the* x1 *component. In constrast, the means are the approximately the same in the* x2 *component, which suggests that pruning the second variable would be benefitial to clustering. The distribution with variable selection better approximates the true distribution. Using a sample test set, we got errors of 0.03 and 0.23 for the training with and without variable selection. Note the probability distributions shown in this figure are unnormalized.*

3 MODEL SPECIFICATION

The model is a standard finite mixture of multivariate Gaussian distributions:

$$x_i \mid \theta \overset{iid}{\sim} \sum_{c=1}^{n_c} \lambda_c \mathcal{N}(x_i; \mu_c, \Sigma_c) \tag{1}$$

where $\{x_i \in \mathbb{R}^{n_x};\ i = 1 \ldots N\}$ denotes the observations, $\theta \triangleq (\lambda, \mu, \Sigma)$ encompasses all the model parameters, λ denotes the mixing weights, μ and Σ denote the mean and covariance of the mixture component densities, and n_c denotes the number of components.

The mixture model is defined on the standard probability simplex $\{\lambda : \lambda_c \geq 0 \text{ for all } c \text{ and } \sum_{c=1}^{n_c} \lambda_c = 1\}$.

We concentrate on this simple model for presentation purposes. However, extensions to hierarchical mixtures (Barnard and Forsyth 2001, Hofmann 1999) and variational mixtures (Blei et al. 2002) are straightforward. Later, in Section 5.1, we present a slight modification of this model for our object recognition and multimedia translation applications.

We introduce the latent allocation variables $z_i \in \{1, \ldots, n_c\}$ to indicate that a particular sequence x_i belongs to a specific cluster c. These indicator variables $\{z_i; i = 1, \ldots, N\}$ correspond to an i.i.d. sample from the distribution $p(z_i = c) = \lambda_c$.

3.1 PRIOR SPECIFICATION

We follow a hierarchical Bayesian strategy where the unknown parameters θ and the allocation variables z are regarded as being drawn from appropriate prior distributions. We acknowledge our uncertainty about the exact form of the prior by specifying it in terms of some unknown parameters (hyperparameters). We place hyperpriors on some of the hyperparameters so that irrelevant features are shrunk to a common value. That is, the estimation of the hyperparameters will allow us to eliminate the effect of irrelevant features in the mixture model.

Our hierarchical Bayesian model is depicted as a graphical model in Figure 2. The allocation variables z_i are assumed to be drawn from a multinomial distribution, $z_i \sim \mathcal{M}_{n_c}(1; \lambda)$, which admits the density

$$p(z_i|\lambda) = \prod_{c=1}^{n_c} \lambda_c^{\mathbb{I}_c(z_i)},$$

where $\mathbb{I}_c(z_i) = 1$ if x_i belongs to group c and $\mathbb{I}_c(z_i) = 0$ otherwise. We place a conjugate Dirichlet prior on the mixing coefficients $\lambda \sim \mathcal{D}_{n_c}(\nu)$, having the following density

$$p(\lambda|\nu) = \frac{\Gamma(\nu_0)}{\Gamma(\nu_1) \cdots \Gamma(\nu_{n_c})} \prod_{c=1}^{n_c} \lambda_c^{\nu_c - 1}, \qquad (2)$$

where $\Gamma(\cdot)$ denotes the Gamma function and $\nu_0 = \sum_c \nu_c$.

We assign a Gaussian prior to the component means $\mu_c \sim \mathcal{N}(\mu^*, T)$, where T is diagonal with elements $\tau_1^2, \ldots, \tau_{n_x}^2$. We take μ^* to be sample mean of the observed data. The idea is to estimate the τ's such that the irrelevant features have τ's near zero and the corresponding components of μ_c across groups are shrunk towards sensible common values.

We place a conjugate inverse-Wishart prior on the covariances $\Sigma_c \sim IW_{n_x}(\alpha, \alpha\Sigma^*)$, admitting the density

$$p(\Sigma|\alpha, \Sigma^*) \propto |\Sigma|^{-(\alpha+n_x+1)/2} \exp\{-(1/2)tr(\alpha\Sigma^*\Sigma^{-1})\}$$

This prior is centered at the covariance of the data Σ^*. We place an inverse Gamma distribution on each $\tau_k^2 \sim \mathcal{I}g(a, b)$, for $k = 1, \ldots, n_x$. This distribution admits the density

$$p(\tau_k^2|a, b) \propto \left(\tau_k^2\right)^{-a-1} \exp\left(-\frac{b}{\tau_k^2}\right).$$

The τ_k^2 are assumed to be independent:

$$p(T|a, b) = \prod_{k=1}^{n_x} p(\tau_k^2|a, b)$$

Since all the priors are conjugate, obtaining the posterior distributions of the parameters by conditioning on the allocation variables is straightforward.

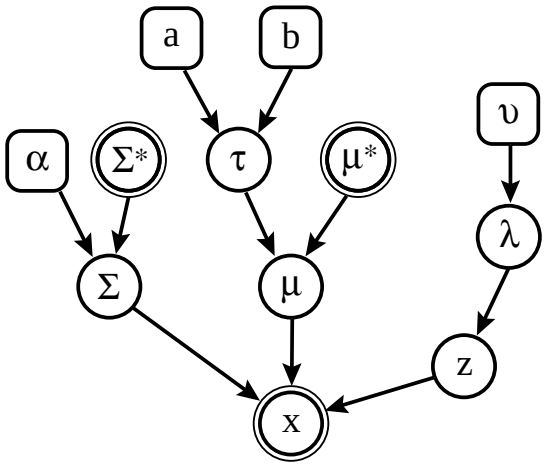

Figure 2: *DAG for the Bayesian mixture model with shrinkage priors. The fixed hyper-parameters are represented with boxes, the variables determined by the data as double circles, and the unknown random variables with empty circles.*

4 COMPUTATION

The parameters of the mixture model cannot be computed analytically unless one knows the mixture indicator variables. We have to resort to numerical methods. One can implement a Gibbs sampler to compute the parameters and allocation variables. This is done by sampling the parameters and allocation variables from their respective posteriors. However, this algorithm can be computationally intensive for the applications we have in mind. Instead we opt for an expectation maximization (EM) algorithm to compute the *maximum a posteriori* (MAP) point estimates of the mixture model.

4.1 EM ALGORITHM

After initialization, the EM algorithm for MAP estimation iterates between the following two steps:

1. **E step:** Compute the expectation of the complete log-posterior with respect to the distribution of the allocation variables $\mathcal{Q}(\theta) = \mathbb{E}_{p(z|x,\theta^{(\text{old})})}[\log p(z, x, \theta)]$, where $\theta^{(\text{old})}$ represents the value of the parameters at the previous time step.
2. **M step:** Maximize over the parameters: $\theta^{(\text{new})} = \arg\max_{\theta} \mathcal{Q}(\theta)$

For the E step, we compute the posterior of the allocation variables:

$$\xi_{ic} \triangleq p(z_i = c|x_i, \theta^{(\text{old})}) = \frac{p(z_i = c, x_i, \theta^{(\text{old})})}{p(x_i|\theta^{(\text{old})})},$$

which, in our case, is given by:

$$\xi_{ic} = \frac{\lambda_c \mathcal{N}(x_i; \mu_c, \Sigma_c)}{\sum_{c'} \lambda_{c'} \mathcal{N}(x_i; \mu_{c'}, \Sigma_{c'})} \tag{3}$$

For the M step, we note that the function $\mathcal{Q}(\theta)$ expands to

$$\begin{aligned}
&\sum_i \sum_c \xi_{ic} \left\{ -\frac{1}{2} log|\Sigma_c| - \frac{1}{2}(x_i - \mu_c)' \Sigma_c^{-1}(x_i - \mu_c) \right\} \\
&\quad + \sum_c \left\{ -\frac{1}{2} \log|T| - \frac{1}{2}(\mu_c - \mu^*)' T^{-1}(\mu_c - \mu^*) \right\} \\
&\quad + \sum_c \left\{ -\frac{1}{2}(\alpha + n_x + 1) \log|\Sigma_c| - \frac{1}{2} tr(\alpha \Sigma^* \Sigma_c^{-1}) \right\} \\
&\quad + \sum_k (-a - 1) \log\left(\tau_k^2\right) - \frac{b}{\tau_k^2}.
\end{aligned}$$

Taking derivatives, we obtain the following update equations for the model parameters:

$$\begin{aligned}
\widehat{\lambda}_c &\leftarrow \frac{\sum_{i=1}^N \xi_{ic} + \nu_c - 1}{N + \sum_{c'} \nu_{c'} - n_c} \\
\widehat{\mu}_c &\leftarrow (\xi_{\cdot c} \Sigma_c^{-1} + T^{-1})^{-1} \left[\Sigma_c^{-1} (\sum_i \xi_{ic} x_i) + T^{-1} \mu^* \right] \\
\widehat{\Sigma}_c &\leftarrow \frac{[\sum_i \xi_{ic}(x_i - \mu_c)(x_i - \mu_c)'] + \alpha \Sigma^*}{\xi_{\cdot c} + \alpha + (n_x + 1)} \\
\widehat{\tau}_k^2 &\leftarrow \frac{b}{a + \frac{n_c}{2} + 1} + \frac{1}{2a + n_c + 2} \sum_{c=1}^{n_c} (\mu_{ck} - \mu_k^*)^2
\end{aligned}$$

where $\xi_{\cdot c} \triangleq \sum_{i=1}^N \xi_{ic}$. These updates, together with equation (3), constitute our EM algorithm.

Using the matrix inversion lemma, we can rewrite the update step for μ_c as

$$\widehat{\mu}_c \leftarrow T(\Sigma_c + \xi_{\cdot c} T)^{-1} (\sum_i \xi_{ic} x_i) + \Sigma_c (\Sigma_c + \xi_{\cdot c} T)^{-1} \mu^*$$

In this expression, we don't have to invert the diagonal matrix T. This permits elements of T to go to zero, allowing for feature selection.

From the EM update equations, one can readily observe the shrinkage in operation. As $\tau \to 0$, the estimate of the means shrinks to the sample mean and the covariance shrinks to the sample covariance. Typically we normalise the data so that $\mu^* = 0$ and $\Sigma^* = I_{n_x}$. Hence, the term $\alpha\Sigma^*$ in the update of the covariances stabilises the algorithm. That is, it prevents the ill-conditioning phenomenon arising from having an unbounded likelihood.

5 OBJECT RECOGNITION APPLICATION

Over the last decade, we have experienced an explosion of multimedia in digital libraries and the world-wide-web. In particular, we have witnessed the emergence of large collections of annotated images. Examples of these include the Corel dataset (see Figure 3), most museum image collections (`http://www.thinker.org/fam/thinker.html`), the web archive (`http://www.archive.org`), news photographs, private photo albums and videos. Multimedia robotics, where robots acquire images, sounds and text as they navigate through a particular environment also results in this type of data. Typically, these annotations refer to the content of the annotated image, but may vary in their accuracy and conciseness. For instance, the Corel annotations describe specific image content, but not all of it. Museum collections are often annotated with some specific material such as the artist and date of acquisition, and often the annotations deal with some rather abstract material.

The annotated image data allows us to formulate object recognition as a statistical machine translation problem (Duygulu et al. 2002). We segment images into regions (blobs) and then learn to predict words using regions. Each region is described by some set of features. In machine translation, a lexicon links words in one language to words in the other language. (Brown, Della Pietra, Della Pietra and Mercer 1993, Melamed 2001). In the domain of object recognition, the feature we associate with image regions do not naturally occupy a discrete space. The simplest solution to this problem is to use K-means to vector quantise the image region representation. This approach, adopted in (Duygulu et al. 2002), was well

lion mane grass sky jet plane sky

Figure 3: *Examples from the Corel data set (Duygulu et al. 2002). We have associated keywords and segments for each image, but we don't know which word corresponds to which segment. The number of words and segments can be different; even when they are the same, we may have more than one segment for a single word, or more than one word for a single blob.*

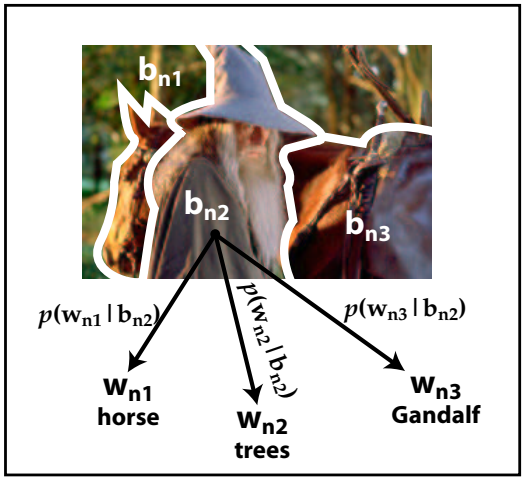

Figure 4: *The association probabilities provide the correspondences (assignments) between each word and the various image segments (blobs).*

received by the computer vision community (it won best paper prize in cognitive computer vision at ECCV 2002). We would like to surmount the problem of having to introduce noise by vector quantising the blobs, so instead we adopt a discrete-Gaussian translation model that allows us to translate Gaussian features directly to discrete features. The shrinkage component of our model allows us to carry out feature selection within this translation procedure.

5.1 TRANSLATION MODELS

Our approach to recognition is analogous to machine translation, as originally proposed by (Duygulu et al. 2002). We translate image regions (French) to words (English). In particular, our model acts as a *lexicon*, a devices that predicts one representation (words; English) given another representation (image regions; French). Learning a lexicon from data is a standard problem in machine translation literature (Brown et al. 1993, Melamed 2001). Typically, a lexicon is learned from a form of data set known as an *aligned bitext* — a text in two languages where rough correspondence, perhaps at the paragraph or sentence level, is known. The problem of lexicon acquisition involves determining precise correspondences between words of different languages. Analogously, we can consider a data set consisting of annotated images as an aligned bitext since we have an image consisting of regions, and a set of text. We know the text goes with the image, but we don't know which word goes with which region.

Computing the translation probabilities is not straightforward because we have a "chicken and egg" situation: to build the translation probabilities we need to know the correct correspondences between images and words, but to figure out the correspondences we need to know the translation probabilities. As explained in (Duygulu et al. 2002), we can overcome this problem by marginalising over the possible associations using a mixture model, an idea that was originally proposed for text in (Brown et al. 1993). This model assumes one-to-one probabilistic assignments as shown in Figure 4. For each blob $b_{nj} \in \mathbb{R}^{n_x}$ in the n-th image, there is a probability of it being associated with each of the words w_{ni} in the image. More precisely, the translation model is given by:

$$p(b|w) = \prod_{n=1}^{N} \prod_{j=1}^{M_n} \sum_{i=1}^{L_n} p(a_{nj} = i) t(b = b_{nj} | w = w_{ni})$$

where $p(a_{nj} = i)$ denotes the alignment probabilities such that $a_{nj} = i$ if b_{nj} translates to w_{ni}. The number of blobs and words in the n-th image are M_n and L_n, respectively. Since the words in the annonations are unordered, there is no reason for preferring specific alignments. Therefore, we set $p(a_{nj} = i) = 1/L_n$.

In (Duygulu et al. 2002), $t(b = b_{nj} | w = w_{ni})$ is a discrete translation table. In this approach, the translation probabilities $t(\cdot|\cdot)$ are Gaussian. More precisely, there is one Gaussian for each word in the vocabulary. Each word generates image segments from a corresponding Gaussian cluster. By clustering the words (say with K-means or SVD methods), it is also possible to have a Gaussian cluster for each concept. The discrete-Gaussian translation model is now

$$\begin{aligned} p(b|w) &= \prod_{n=1}^{N} L_n^{-M_n} \prod_{j=1}^{M_n} \sum_{i=1}^{L_n} t(b_{nj} | w_{ni}) \\ &= \prod_{n=1}^{N} L_n^{-M_n} \prod_{j=1}^{M_n} \sum_{i=1}^{L_n} \mathcal{N}(b_{nj}; \mu_c, \Sigma_c) \delta_{w_c^*}(w_{ni}) \end{aligned}$$

where c is in the range $(c = 1, \ldots, L_T)$, $w_c^\star$ denotes a particular word token, and $\delta_{w_c^\star}(w_{ni}) = 1$ if $w_c^\star$ appears in the n-th annotation, otherwise it is 0. The index c is the index for the L_T word tokens and accordingly the L_T clusters. In the end, the objective is to generate words from blobs. This is a straightforward application of Bayes rule, $p(w|b) \propto p(b|w)p(w) \propto p(b|w)$ since we currently assume a uniform prior for $p(w)$.

This model is a variation of the mixture model of equation 1. Using the notation

$$\xi_{nji} \triangleq p(a_{nj} = i | w_{ni}, b_{nj}, \theta^{(\text{old})}) = \frac{t(b_{nj}|w_{ni})}{\sum_{k=1}^{L_n} t(b_{nj}|w_{nk})},$$

the updates for the parameters are

$$\widehat{\mu}_c = \left(T^{-1} + \Sigma_c^{-1} \sum_n \sum_{j=1}^{M_n} \sum_{i=1}^{L_n} \xi_{nji} \delta_{w_c^\star}(w_{ni}) \right)^{-1} \left(T^{-1}\mu^* + \Sigma_c^{-1} \sum_n \sum_{j=1}^{M_n} \sum_{i=1}^{L_n} \xi_{nji} \;\; b_{nj} \;\; \delta_{w_c^\star}(w_{ni}) \right)$$

$$\widehat{\Sigma}_c = \frac{\alpha\Sigma^* + \sum_n \sum_{j=1}^{M_n} \sum_{i=1}^{L_n} \xi_{nji} (b_{nj} - \mu_k)(b_{nj} - \mu_k)' \delta_{w_c^\star}(w_{ni})}{\sum_n \sum_{j=1}^{M_n} \sum_{i=1}^{L_n} \xi_{nji} \delta_{w_c^\star}(w_{ni}) + \alpha + n_x + 1}$$

The updates for τ remain the same as before.

5.2 SIMPLE DATA ASSOCIATION

Our first experiment serves the role of a "proof of concept". We used the discrete-Gaussian translation model to generate synthetic data. In particular, the vocabulary consists of 3 word tokens and we assume that the blob components are normally distributed. The blobs have 9 features, but only the first 3 are relevant. That is, the first 3 features are sampled from 3 different Gaussians corresponding to the three word tokens, while the last 6 features are sampled from a Gaussian common to all the entries in the dataset. We added enough variance to provide some degree of overlap between blob corresponding to different words. We also created a similar dataset to test the models.

The results of using maximum likelihood (ML) and MAP point estimators are given in Figure 5. The box plots show the RMS error between the estimated means and the true means over the 19 trials. The hyper-prior's parameters were set to $a = -1$ and $b = 1/100000$ (any other small number would do). This choice allows for shrinkage with some regularisation. We consistently found that the lowest possible value of α gave the best results (in this case, $\alpha = 11$ is the lowest value).

From the plot, it is clear that ML performs the best when it is provided only the first three features since it is less prone to learning meaningless correspondences

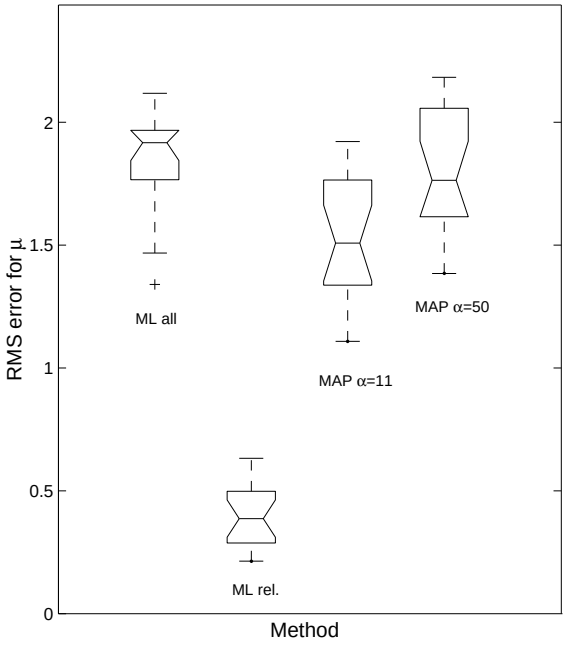

Figure 5: *Root mean squared errors between the estimated means and the true means. The box plots are generated from trials on 19 random data sets.*

in the data. According to this simple experiment, the MAP trials with different values of α tend to perform better than ML when given both the relevant and irrelevant blob features. The Bayesian priors do mitigate the effect of the irrelevant features. Further evidence is provided by the estimated τ:

$$\widehat{\tau} = [1.1 \;\; 1.1 \;\; 1.0 \;\; 0.3 \;\; 0.2 \;\; 0.2 \;\; 0.3 \;\; 0.3 \;\; 0.3]$$

Notice the last 6 irrelevant features are being shrunk to zero.

5.3 COREL DATA EXPERIMENT

We trained the models on annotated images from the Corel database. Each image, with 4-5 keywords, is segmented using Normalized Cuts (Shi and Malik 1997). Each of the regions provided by the segmenter are represented by 10 real-valued features: horizontal and vertical position in the image, area, perimeter divided by area, average (lab) colour and standard deviation of the (lab) colour.

We make a point of selecting features we intuitively believe are useful, such as colour, as well as features that may be less relevant (such as position). However, it is impossible to determine beforehand the relative utility of the individual features.

The training and test sets each contain 50 images. Over the entire dataset there are 49 different words. We consider four models. The first is a maximum likelihood model that takes advantage of all the features. The second is similar, except that it is trained only on our choice of relevant features, mean and standard deviation of colour. We diagonalize the covariances in both ML methods to maintain stability of the EM algorithm, although this move comes at the cost of loss of valuable information. The last two models are variants of the MAP, where we diagonalize the covariances in one model and keep the updated covariances untouched in the other. Both Bayesian methods use parameter values of $\alpha = 13$, $a = -1$ and $b = 1/100000$ as before.

To measure the performance of the models, we manually annotated the images in the test set. In some cases, there might be more than one annotation deemed correct because the blob consisted of several subjects. We used two simple but effective error measures:

- **PR-n**: This measure reports an error of 1 if none of the n most likely predicted words results in a correct blob annotation.
- **PR-sampled**: This measure reports the probability of an incorrect annotation given that we sample from the estimated word probabilities for a particular blob.

Both error measures are averaged over the number of blobs in each image, and then over the number of documents in the data set. We also compare the models' predictions to the predictions obtained using the empirical word frequency of the training set. Matching the performance of the empirical density is required to demonstrate non-trivial learning. Doing substantially better than this on the Corel data is difficult. The annotators typically provide several common words (*e.g. sky, water, people*), and fewer less common words (*e.g. tiger*). This means that annotating all images with, say, *sky, water*, and *people* is a moderately successful strategy. Performance using the empirical word frequency would be reduced if the empirical density was more flat. Thus, the increment of performance over the empirical density is a sensible indicator.

Figures 6 through 7 show the error measure plots for PR-1, PR-3 and PR-sampled on both the training and test sets for 12 independent trials using the same data set. The variance in the box plots is due to random initialization. The MAP tends to show less variance due to the stability the priors place on the model. This extra stability allows us to adopt full-covariance models and hence obtain better results.

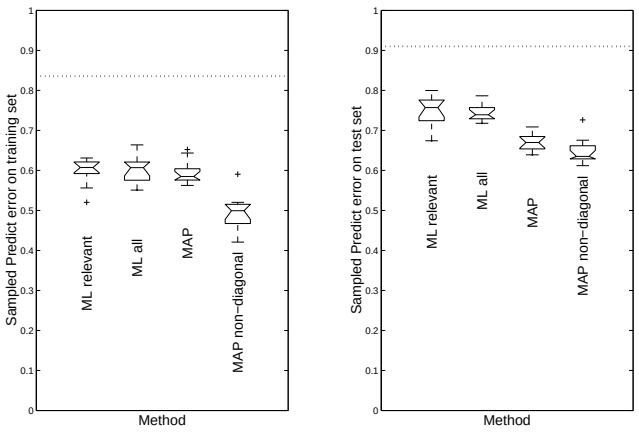

Figure 6: *Recognition errors using the PR-sampled measure. The horizontal line corresponds to the empirical distribution predictions. The MAP approach with full-covariances can perform even better than the ML approach using only the relevant features. This is the result of the stability brought in by the priors in the update equations for the means and covariances.*

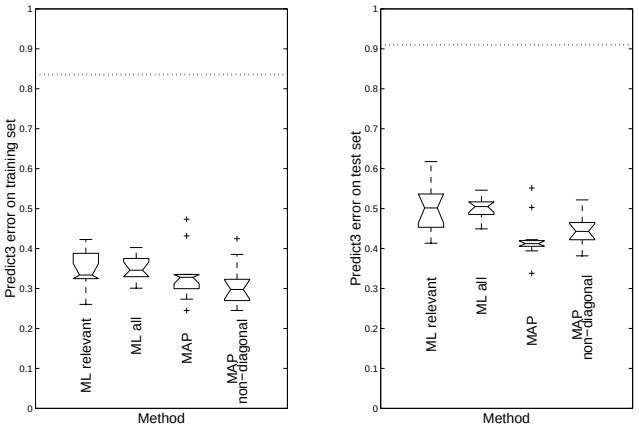

Figure 7: *Recognition errors using the PR-3 measure. The horizontal line corresponds to the empirical distribution predictions.*

Encouragingly, the Bayes methods tend to place low importance on the features we suspect having no relevance to the concepts they describe. The estimated diagonal elements of T were:

$$\widehat{\tau} = [0.4 \;\; 0.1 \;\; 0.0 \;\; 0.9 \;\; 1.8 \;\; 4.9 \;\; 2.4 \;\; 2.2 \;\; 3.2 \;\; 3.4]$$

In this situation, the first four components of the means refer to the area, x, y and boundary/area features and the last six correspond to the mean and standard deviation (lab) colour.

Some sample recognition results on arbitrary test set images are show in Figures 8 and 9.

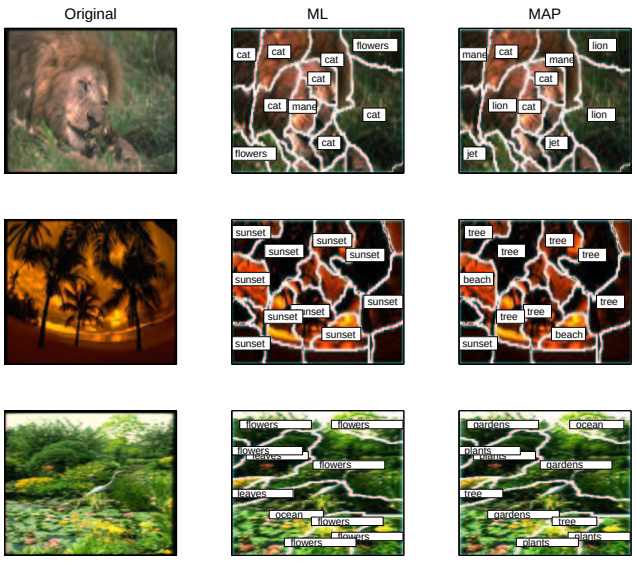

Figure 8: *Good recognition results on test set images.*

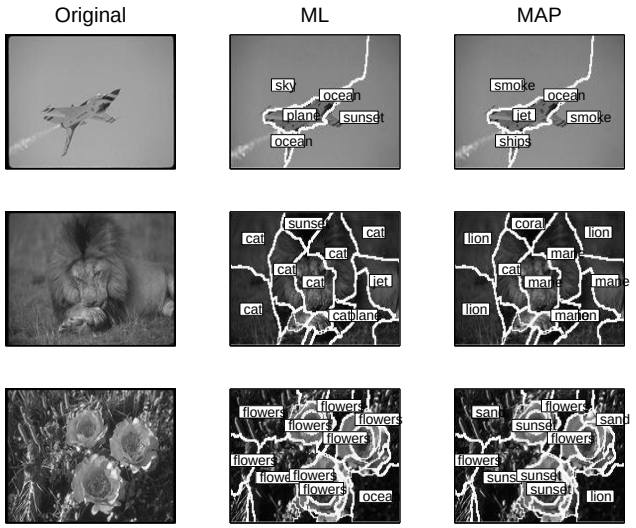

Figure 9: *Randomly chosen results on test set images.*

6 CONCLUSIONS

We presented a simple yet very powerful and practical method for weighting or selecting features in unsupervised learning scenarios. The results on a complex object recognition problem show that our Bayesian models not only weight the features appropriately, but also increase the stability of the algorithms.

ACKNOWLEDGMENTS

Much of this work originated from collaborations with Kobus Barnard, Pınar Duygulu and David Forsyth. The Corel data and its segmentations were kindly provided by Kobus Barnard and Pınar Duygulu. We would like to thank Eric Brochu and Kejie Bao for proof-reading the paper.

References

Barnard, K. and Forsyth, D. (2001). Learning the semantics of words and pictures, *International Conference on Computer Vision*, Vol. 2, pp. 408– 415.

Barnard, K., Duygulu, P. and Forsyth, D. (2001). Clustering art, *Computer Vision and Pattern Recognition*, Vol. 2, pp. 434–439.

Blei, D. M., Ng, A. Y. and Jordan, M. I. (2002). Latent dirichlet allocation, *in* T. G. Dietterich, S. Becker and Z. Ghahramani (eds), *Advances in Neural Information Processing Systems 14*, MIT Press, Cambridge, MA.

Brown, P., Della Pietra, S. A., Della Pietra, V. J. and Mercer, R. L. (1993). The mathematics of statistical machine translation: Parameter estimation, *Computational Linguistics* **19**(2): 263–311.

Duygulu, P., Barnard, K., de Freitas, N. and Forsyth, D. (2002). Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary, *ECCV*.

George, E. I. and Foster, D. P. (1997). Calibration and empirical Bayes variable selection, Unpublished. Department of Management Science and Information Systems, University of Texas.

Golub, T. R., Slonim, D. K. and Tamayo, P. (1999). Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring, *Science* **286**: 531–537.

Hofmann, T. (1999). Probabilistic latent semantic analysis, *Uncertainty in Artificial Intelligence*.

Hofmann, T. and Puzicha, J. (1999). Latent class models for collaborative filtering, *Sixteenth International Joint Conference on Artificial Intelligence*, pp. 688–693.

Liu, J. S., Zhang, J. L., Palumbo, M. L. and Lawrence, C. E. (2002). Bayesian clustering with variable and transformation selections, *in* J. M. Bernardo, J. O. Berger, A. P. Dawid and A. F. Smith (eds), *Bayesian Statistics*, Vol. 7, Oxford University Press.

Melamed, I. D. (2001). *Empirical Methods for Exploiting Parallel Texts*, MIT Press.

Ng, A. Y. and Jordan, M. I. (2001). Convergence rates of the Voting Gibbs classifier, with application to Bayesian feature selection, *International Conference on Machine Learning*.

Shi, J. and Malik, J. (1997). Normalized cuts and image segmentation, *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 731–737.

Smith, M. and Kohn, R. (1996). Nonparametric regression using Bayesian variable selection, *Journal of Econometrics* **75**(2): 317–343.