In image analysis problems, there is uncertainty about the state of the system being imaged due to the inherent ambiguities of the imaging process. We can use a probabilistic model to represent the imaging process, giving a joint probability distribution over the image and the state of the system being imaged. Determining this hidden state from the image is therefore another example of Bayesian inference. Once again, the complexity of the model tends to render exact inference intractable and therefore approximate solutions can be obtained, for example, using variational inference.

In this chapter, a particular image analysis problem, the problem of analysing scanned images of DNA microarrays, is investigated using this Bayesian approach. Variational inference is carried out using an extended form of Variational Message Passing which uses importance sampling to handle a conditional distribution that is not in the exponential family. Whilst such variational importance sampling was developed by Lawrence et al. [2002], its inclusion within the message passing framework is my own work.

4.1 DNA Microarrays

DNA microarray technology allows rapid identification of the level of expression of thousands of genes in a tissue or organism, all on a single slide.

Gene expression microarrays are typically produced by transferring cDNA\(^1\) or oligonucleotides\(^2\) in high salt solutions onto chemically modified glass microscope slides using a contact-printing instrument [Eisen and Brown 1999; Hegde et al. 2000]. These cDNA probes are exposed to target cDNA which has been reverse-transcribed and labelled with a fluorescent dye. The target cDNA then binds with just those probes that have complementary

---

\(^1\)Complementary DNA (cDNA) has a base sequence which is the complement of an original DNA sequence. The complement of a sequence is one with each base replaced by its complementary base: A by T, C by G, and vice versa.

\(^2\)An oligonucleotide is a short stretch (usually 2-50 bases) of single-stranded DNA.
4.2 MICROARRAY IMAGES

base sequences, in a process known as hybridisation. The resultant hybridisation patterns are detected by fluorescent imaging of the slide. The image must then be processed to identify the presence and levels of gene expression in the target.

There are a number of existing software tools for analysing microarray images and extracting the gene expression data. For example ScanAlyze\textsuperscript{3} allows a user to mark by hand the size and shape of each spot in the image. It would be desirable to automate this process, as this would both reduce the time taken to analyse the images and improve the reliability of the resultant gene expression data. Although there have been several attempts at automating this process using semi-empirical approaches, such as Dapple [Buhler et al. 2000] and Spot [Dudoit et al. 2000], these tools tend to be tuned for a particular type of image. Adopting any one such tool is often a long process of trial and error as the interplay of effects of algorithm parameters is difficult to anticipate.

In this chapter, I present a system for automatic analysis of microarray images using Bayesian methodology and variational inference. Whilst the system does have a small number of parameters, these directly model our knowledge of the images to be analysed and so any necessary initialisation is straightforward. Most image parameters are automatically inferred from the image or taken from the configuration file of the contact-printing instrument that was used to print the microarray.

In Section 4.2, I describe the experimental setup used to obtain test images. A probabilistic model of microarray images is developed in Section 4.3 and a method of performing inference in this model using an extended form of Variational Message Passing is described, with results on real images, in Sections 4.4 and 4.5. A solution to the problem of locating grids of spots is given in Section 4.6 and the entire system discussed in Section 4.7. Finally, in Section 4.8, the question of how to analyse the resultant gene expression data is addressed and a brief example given which uses Variational Message Passing.

4.2 Microarray Images

A typical microarray slide consists of a rectangular array of sub-grids, each sub-grid printed by one pin of the contact-printer. A sub-grid consists of an array of spots, each spot containing a single cDNA probe. The hybridised arrays are imaged using a scanner, such as a laser scanning confocal microscope, and the output stored as 16-bit image files. Where a number of dyes are used, one image is produced for each.

Microarray images typically have significant background noise and can also have other noise artefacts, some of which are introduced during the scanning process. For example, when using a wide-field CCD scanner, dust particles on the slide can cause scatter flares (large, bright circular artefacts) which may obscure one or more of the spots. Alternatively, when using a laser scanner, reflections within the optical subsystem of the scanner can result in the introduction of additional false spot images. In addition, the spots themselves vary in

\textsuperscript{3}ScanAlyze software is available from http://rana.lbl.gov/EisenSoftware.htm.
4.2. MICROARRAY IMAGES

Figure 4.1: (a) Section of an image from a laser scanner showing false spots (the two very faint spots above each top corner of the grid), noise artefacts and a high level of background noise. (b) Section of an image from a wide-field scanner showing the reduced background noise which is an advantage of this scanner. Unfortunately, this type of scanner also causes scatter flares – the one seen here obscures four of the spots. Note the variation in spot shape and size in each image. In both images, the intensities have been mapped so as to make dim spots visible.

size and shape, even within a single sub-grid. When there are sets of images for different dyes on the same slide, some noise artefacts will be common to all of the dye images, whilst there are often systematic variations in background noise from dye to dye.

Figure 4.1 shows two sub-grids extracted from actual microarray images which include examples of many of these noise artefacts.

4.2.1 Experimental methodology

The microarray images used throughout this chapter were created in the Ashburner Laboratory at Cambridge University Genetics Department by Gos Micklem, David Kreil et al., who have kindly made them available for this research. The variations between microarray images are partly due to the different experimental methodologies and equipment used in different laboratories. For this reason, the methodology used to create the test images for this system will now be described in some detail.

The microarrays used were printed using the BioRobotics MicroGrid II Total Array System 4 contact-printing instrument and 48 BioRobotics MicroSpot 2500 split-pins. Drosophila Gene Collection PCR-amplified cDNA inserts from the Berkeley Drosophila Genome Project 5 were printed on in-house coated Poly-L-Lysine slides. To minimise variations due to environ-

4 http://www.biorobotics.co.uk/
5 http://www.fruitfly.org/
mental effects, a BioRobotics Humidity Control Unit was used and printing took place in a temperature controlled room.

Printed slides were heated, and cross-linked using UV light. Unspecific binding of DNA to the slides was blocked using a solution of succinic anhydride in 1-methyl-2-pyrrolidinone and boric acid. Double stranded DNA was denatured by further heat treatment.

Tissue from the fruit fly Drosophila was homogenised in TRIzol, and RNA was extracted and precipitated with chloroform and isopropanol. Samples were then directly labelled by reverse transcription incorporating nucleotides with a covalently bound dye (either Cy3-dCTP or Cy5-dCTP). Samples labelled with different dyes were then jointly hybridised on a microarray slide using a Genomic Solutions GeneTAC hybridisation station.

To provide a variety of test images, the slides were scanned with one of two different scanners: a GenomicSolutions GeneTAC LS-IV confocal laser scanner or an Applied Precision ArrayWoRx wide-field CCD scanner.

4.3 A Probabilistic Model for Microarray Images

As in any inference problem, we start by defining our probabilistic model. The observed variables within this model are the grey levels of the image pixels. The model must also include latent variables representing the information that we are trying to extract from the image: the shape and location of the individual spots. Any assumptions that we make about the imaging process will be explicitly encoded in the model. The model therefore defines, by its assumptions, what types of images are suitable for analysis using this system (i.e. those where these assumptions hold).

Rather than working with the entire slide image, we assume that we have extracted a section of the image which contains a single sub-grid, like those of Figure 4.1. The number of rows and columns in the sub-grid can be found from the configuration file for the array printer. This file also tells us the approximate size of each spot and their approximate separations. Finally, we assume, at this stage, that we have a rough estimate of the location of each spot. This could be provided through user input (such as by specifying the location of three corner spots and interpolating using a regular grid) or by automatic means, as will be discussed later.

4.3.1 Latent variables and their prior distributions

We now define our latent variables. The actual location of each spot will be represented by a two-dimensional vector variable \( \mathbf{c} = (c_x, c_y) \), which is the location in pixels specified relative to the initial estimated location. The spot is assumed to be an axis-aligned ellipse and so the shape is encoded by \( \mathbf{r} = (r_x, r_y) \) where \( r_x \) is the radius in the \( x \)-direction and \( r_y \) is the radius in the \( y \)-direction. The assumption that spots are axis-aligned ellipses is a good assumption in the vast majority of cases where the spots are nearly circular and distortions are due to slight differences in scanning resolution on the \( x \) and \( y \) axes. However, extending the model
4.3. A PROBABILISTIC MODEL FOR MICROARRAY IMAGES

To allow for rotated ellipses or other shapes is also possible provided one is willing to accept the additional computation required to learn the extra parameters.

The prior distribution over the position vector \( \mathbf{c} \) is defined to be a Gaussian distribution

\[
P(\mathbf{c} | \mu_c, \gamma_c) = \mathcal{N}(\mathbf{c} | \mu_c, \gamma_c^{-1}),
\]

where \( \gamma_c \) is a diagonal inverse covariance matrix. The parameters \( \mu_c \) and \( \gamma_c \) are governed by conjugate hyper-priors

\[
P(\mu_c) = \mathcal{N}(\mu_c | \mathbf{m}_c, \beta_c^{-1} \mathbf{I}) \tag{4.2}
\]
\[
P(\gamma_c) = \text{Gamma}(\gamma_{c00} | a_c, b_c) \text{Gamma}(\gamma_{c11} | a_c, b_c). \tag{4.3}
\]

The parameter \( \mathbf{m}_c \) is set to the supplied rough location of the spot and the precision \( \beta_c \) is set to give a corresponding standard deviation of one quarter of the distance between the centres of adjacent spots. The parameters \( a_c \) and \( b_c \) were set to 0.05 and 0.1 respectively.

We define a similar prior distribution over the size vector \( \mathbf{r} \) with parameters \( \{ \mu_r, \gamma_r \} \) and hyper-parameters \( \{ \mathbf{m}_r, \beta_r, a_r, b_r \} \):

\[
P(\mu_r) = \mathcal{N}(\mu_r | \mathbf{m}_r, \beta_r^{-1} \mathbf{I}) \tag{4.4}
\]
\[
P(\gamma_r) = \text{Gamma}(\gamma_{r00} | a_r, b_r) \text{Gamma}(\gamma_{r11} | a_r, b_r). \tag{4.5}
\]

In this case, \( \mathbf{m}_r \) is set to the expected radius determined from the configuration file; all other parameters are the same as for the centre prior. To distinguish between the location and size variable for different spots, we define the \( j \)th spot to have location \( \mathbf{c}_j \) and size \( \mathbf{r}_j \).

4.3.2 The likelihood function

The likelihood function defines the probability of a particular image given a particular setting of all the latent variables \( \{ \mathbf{c}_j, \mathbf{r}_j \}_{j=1}^J \). In order to simplify the inference problem, we separate this likelihood function into a product of functions each corresponding to a small area of the image containing a spot. Thus, the likelihood function for the \( j \)th spot gives the probability of the rectangular subimage \( I_j \) centred on the approximate location of the \( j \)th spot given the \( j \)th set of parameters \( \{ \mathbf{c}_j, \mathbf{r}_j \} \). This independence assumption is valid provided that the amount of ‘wobble’ on the array printer is not so great that the spots actually overlap – if this is not the case and there are overlapping spots, there would be great difficulty determining their individual intensities anyway. In practice, this assumption holds as the spots are typically well separated. In the future, higher density arrayers may try to fit more spots on a single slide and image analysis may then require a model which does not make this independence assumption.

It follows that, within any given subimage \( I_j \), we expect to find a single spot. A setting of the parameters \( \{ \mathbf{c}_j, \mathbf{r}_j \} \) partitions the pixels of \( I_j \) into two disjoint sets: the spot pixels \( S \)
which lie inside an ellipse with centre $c_j$ and radii $r_j$ and the remaining background pixels $B$ which are outside the ellipse. In defining our likelihood function, we now make the further assumption that the probability distribution over the intensity of a particular pixel depends only on whether it is in $S$ or $B$. This pixel independence assumption allows us to write the likelihood function as

$$P(I_j | c_j, r_j) = \prod_{b \in B} P_B(I_b) \prod_{s \in S} P_S(I_s), \quad (4.6)$$

where $P_B(I_b)$ is the likelihood function for background pixel intensity and $P_S(I_s)$ is the likelihood function for spot pixel intensity.

This raises the question of how to define $P_B$ and $P_S$. One approach would be to use the rough spot positions to divide the entire sub-grid image into (approximately) spot and background pixels and to use the statistics of these two sets of pixels to define $P_B$ and $P_S$. The difficulty with this approach is that the distribution over pixel intensities varies from spot to spot and, in most images, varying background noise means that the distribution over background pixel intensities also changes significantly, even within a single sub-grid. Hence, if we were to fix $P_B$ and $P_S$ for the entire sub-grid, we would suffer from problems like background noise in one part of the image masking dim spots in other areas of the image, even if there were little background noise there.

We can avoid these problems by inferring $P_B$ and $P_S$ separately for each spot. To achieve this, we quantise the pixel intensities into one of $K$ bins. Each of $P_B$ and $P_S$ is then a discrete distribution which defines the probability of a pixel intensity being in each bin. The parameters of $P_S$ are the $K$ probabilities $\{p_1, p_2, \ldots, p_K\}$ where each $p_i$ is the probability that a spot pixel will lie in the $i$th intensity bin. Similarly, the parameters of $P_B$ are $\{q_1, q_2, \ldots, q_K\}$.

The likelihood function may now be rewritten as

$$P(I_j | c_j, r_j, \{p_k\}, \{q_k\}) = \prod_{k=1}^{K} p_k^{n_k} \prod_{j=1}^{K} q_k^{m_k}, \quad (4.7)$$

where $n_k$ is the number of pixels in $S$ that lie in the $k$th bin and $m_k$ is the number of pixels in $B$ that lie in the $k$th bin. We then define a Dirichlet prior over these parameters so that

$$P(\{p_k\}_{k=1}^{K}) = \text{Dirichlet}\left(\{p_k\}_{k=1}^{K} | \{u_k\}_{k=1}^{K}\right) \quad (4.8)$$

$$P(\{q_k\}_{k=1}^{K}) = \text{Dirichlet}\left(\{q_k\}_{k=1}^{K} | \{v_k\}_{k=1}^{K}\right). \quad (4.9)$$

Consider just the spot pixels $S$. We can now marginalise out $\{p_k\}$ and write the likelihood
in terms of the Dirichlet parameters \( \{u_k\} \) only,

\[
P(S \mid \{u_k\}) = \int \left( \prod_{k=1}^{K} p_k^{n_k} \right) \text{Dir} (\{p_k\} \mid \{u_k\}) \, dp_1 \ldots dp_K
\]  
(4.10)

\[
= \int \text{Dir} (\{p_k\} \mid \{u_k + n_k\}) \, dp_1 \ldots dp_K \frac{\prod_k \Gamma(u_k + n_k)}{\prod_k \Gamma(u_k + n_k)} \frac{\Gamma(\sum_k u_k)}{\Gamma(\sum_k u_k)} 
\]
(4.11)

where \( \Gamma() \) is the gamma function. A similar marginalisation for the background pixels \( B \) gives us our final likelihood function

\[
P(I_j \mid c_j, r_j, \{u_k\}, \{v_k\}) = \frac{\prod_k \Gamma(u_k + n_k)}{\Gamma(\sum_k u_k + n_k)} \frac{\Gamma(\sum_k u_k)}{\prod_k \Gamma(u_k)} \times \frac{\prod_k \Gamma(v_k + m_k)}{\Gamma(\sum_k v_k + m_k)} \frac{\Gamma(\sum_k v_k)}{\prod_k \Gamma(v_k)}. \]  
(4.12)

The prior parameters \( \{u_k\} \) and \( \{v_k\} \) can be thought of as pseudo-counts and can be set to be proportional to histograms of spot and background pixels over the entire image plus a constant value of 1 (to allow for previously unobserved intensities). The sums of these pseudo-counts dictate the strength of the Dirichlet priors. Good results were achieved when the sum of pseudo-counts was set to be equal to the number of pixels in each subimage and \( K \) was set to 300. Our entire probabilistic model can now be expressed as a Bayesian network, as shown in Figure 4.2.

---

**Figure 4.2:** The Bayesian network for a probabilistic model of microarray sub-grid images. The sub-grid contains \( J \) spots each of which has a centre \( c_j \) and radii \( r_j \) which we wish to infer. The subimage \( I_j \) contains the \( j \)th spot and consists of \( N \) pixel intensity values, quantised to one of \( K \) states. Given a spot location and size, these pixels are divided into two disjoint sets: spot pixels and background pixels. The discrete distributions over pixel intensities for each set have been marginalised out and so are not shown, but are instead governed by Dirichlet priors, whose parameters \( u, v \) are common to all spots. In this model, these parameters are fixed to constant values, indicated by the use of square nodes in the graph.
4.4 Variational Message Passing with Importance Sampling

The Bayesian network defined in the previous section does not allow for the direct application of Variational Message Passing in order to find the posterior over spot sizes and positions. The problem arises due to the form of the conditional \( P(I_{j,n} | \mathbf{c_j}, \mathbf{r}_j) \). This function is nonlinear and not an exponential family distribution. When using variational message passing, this prevents us from finding an analytical form for the child-to-parent messages from \( I_{j,n} \) to \( \mathbf{c}_j \) and to \( \mathbf{r}_j \), which in turn prevents us from finding the updated variational posteriors \( Q(\mathbf{c}_j) \) and \( Q(\mathbf{r}_j) \). Instead, we turn to sampling methods to approximate the posterior variational distribution of \( Q(\mathbf{c}_j) \) and \( Q(\mathbf{r}_j) \), whilst continuing to use standard VMP for the rest of the graph. Effectively, the sampling method will be used as a subroutine within the VMP algorithm. A range of sampling methods are available; for simplicity, we follow Lawrence et al. [2002] and use importance sampling.

Importance sampling is a technique which allows the calculation of approximate expectations under a posterior distribution \( P(x) \). For example, suppose we wish to find the expectation of a function \( f(x) \), we would aim to evaluate,

\[
\langle f(x) \rangle_p = \int f(x) P(x) \, dx.
\]  

This integral is intractable and so we introduce a proposal distribution \( q(x) \) (not to be confused with a variational distribution \( Q(x) \)),

\[
\langle f(x) \rangle_p = \int f(x) \frac{P(x)}{q(x)} q(x) \, dx. \tag{4.14}
\]

The proposal distribution is selected so that it is easy to sample from and we use \( L \) samples from it to approximate our desired expectation:

\[
\langle f(x) \rangle_p \approx \frac{1}{L} \sum_{i=1}^{L} \frac{f(x_i) P(x_i)}{q(x_i)} \tag{4.15}
\]

where the ratios \( P(x_i)/q(x_i) \) used to weight each sample are known as importance weights.

Importance sampling only allows us to estimate the expectations of functions under a posterior \( P \), rather than generate samples from \( P(x) \). The calculation of expectations is sufficient in this case because we only need to compute expectations of natural statistic vectors (which are just functions of individual variables).

When used in conjunction with variational message passing, it makes sense to perform all importance sampling within one node \( Z \): the node whose conditional distribution is non-exponential family (in this case, \( Z \) corresponds to the \( I_{j,n} \) node). The posterior distribution we aim to approximate is therefore the joint variational distribution over all parents of \( Z \) (which in this case is \( Q(\mathbf{c}_j)Q(\mathbf{r}_j) \)). We make the assumption that \( Z \) is observed, as is the case in the microarray model.
Now we must choose a proposal distribution which is as similar as possible to our posterior. Consider if we ignore the effect of the $Z$ node from the calculation of the variational posterior of one of its parents $X_j$. The parameters of such an (incorrect) posterior can then be found analytically using

$$\phi_j^* = \theta_j \left( \{m_{X_i \rightarrow X_j}\}_{i \in \text{pa}_j} \right) + \sum_{k \in \text{ch}_j \setminus Z} m_{X_k \rightarrow X_j}$$

(4.16)

where we are ignoring the message from $Z$. We shall call this posterior $R_j(X_j)$. Whilst this posterior is clearly not equal to $Q_j^*$, it will be similar to it and so usable as a proposal distribution for importance sampling. We therefore define the message from the parent $X_j$ of an importance sampling node $Z$ to be

$$m_{X_j \rightarrow Z} = \langle u_j \rangle_{R_j}$$

(4.17)

where we are sending a natural statistic vector rather than a parameter vector for consistency with other parent-to-child messages only (as either vector is sufficient to parameterise $R_j$).

At $Z$, we define a proposal distribution over the parents which is the product of the $R$ distributions for each parent. We then draw $S$ samples $\{\text{pa}_Z^{(s)}\}_{s=1}^S$ from this proposal distribution, which is a straightforward operation as we can sample for each parent variable independently. Following the importance sampling methodology, we find the importance weight of each sample from the ratio of the variational distribution to the proposal distribution evaluated for that sample

$$w_s = \frac{1}{K} \frac{Q(\text{pa}_Z^{(s)})}{\prod_{j \in \text{pa}_Z} R_j(X_j^{(s)})} = \frac{1}{K} P(Z | \text{pa}_Z^{(s)})$$

(4.18)

(4.19)

where the normalising constant $K$ is chosen to be $\sum_{s=1}^S w_s$, so that the sum of all the importance weights is one. The fact that these weights are calculated from $Z$ and the samples means that the calculation can be performed locally. All that remains is to use these weights to estimate the required expectations of natural statistic vectors for each parent

$$\langle u_j(X_j) \rangle_Q \approx \sum_{s=1}^S w_s u_j(X_j^{(s)})$$

(4.20)

and to send these as the message from $Z$ to that parent. The parent then adopts this message as the new expectation of its natural statistic vector. The corresponding distribution can be thought of as an exponential family approximation to the variational posterior.

One problem with importance sampling is that the sampling estimate can be dominated by a few samples with very high weights. This occurs when there is a mismatch between the proposal distribution $R$ and the distribution of interest (in this case, the variational posterior
4.5. INFERENCE IN THE MICROARRAY IMAGE MODEL

The claim here is that the proposal distribution $R$ is adaptive, adjusting in line with observed data, thereby improving the match between $R$ and $Q$.

The quality of the samples obtained during importance sampling can be summarised by $S_{\text{eff}} = \frac{1}{\sum_{s=1}^{S} w_s^2}$ where $1 \leq S_{\text{eff}} \leq S$ which is known as the effective number of samples. This quantity is used to determine the quality of the sampling approximation and also as a convergence criterion. The contribution of $Z$ to the lower bound $\mathcal{L}$ can also be estimated using

$$\mathcal{L}_Z \approx \sum_{s=1}^{S} w_s P(Z \mid \text{pa}_Z^{(s)}) = \frac{K}{S_{\text{eff}}}.$$  \hfill (4.21)

4.5 Inference in the Microarray Image Model

The hybrid variational/sampling algorithm described above was applied to the microarray image model, using $S = 100$ samples. The order of the updates for each spot was that the sampling node $I_{j,n}$ was updated first, followed by the remaining nodes. Due to the high computational expense of updating the sampling node, this node was only updated one iteration in ten.

As only a (noisy) estimate of the lower bound was available, it could not be used in the normal way as a convergence criterion. Instead, the algorithm was deemed to have converged when $S_{\text{eff}}$ became greater than $S/4$ or a fixed maximum number of iterations was reached.

4.5.1 Handling missing and obscured spots

Microarray images frequently have gaps where no spots appear, corresponding to cDNA probes where little or no hybridisation has occurred. There are also occasions where noise artefacts are sufficient to obscure or heavily mask the spot. The sub-grids of Figure 4.1 showed examples of each of these situations. In both cases, the image model used above provides a poor model of the resultant spot image; its assumption of an elliptical boundary between two areas with differing intensity distributions simply does not hold. In the case of missing spots, the model assumes a spot exists with the same intensity as the background which leads to significant uncertainty in the inferred spot location and size. In the case of obscured spots, the inferred spot size and position can be incorrect; indeed, it may not be possible to determine the actual position of such spots from the image.

For an image analysis algorithm to be useful, it must identify these two special cases and flag the spots so as to avoid outputting false or inaccurate data. The identification of these cases can be achieved by introducing new image models for each case and performing model comparison. The image model for a missing spot is simply an image whose pixels are all background pixels. As there are no latent variables in this model, we can write the image probability directly as

$$P(I_j \mid \mathcal{H}_1) = \left[ \frac{\prod_k \Gamma(v_k + m_k)}{\Gamma(\sum_k v_k + m_k) \prod_k \Gamma(v_k)} \right]^{\Gamma(\sum_k v_k)}$$  \hfill (4.22)
where the Dirichlet parameters \( \{ v_k \} \) are as defined for the standard model and \( m_k \) is the number of pixels in the subimage \( I_j \) whose intensities lie in the \( k \)th intensity bin.

We can define a similar model for obscured spots. When a spot is badly obscured by noise, then the image will contain non-background pixels due to this noise as well as due to the spot. These pixels will not lie in an elliptical region. In fact, the shape of the region will be unpredictable as we cannot make assumptions about what form the noise may take. Instead, we assume that any pixel is equally likely to be background or non-background and so its intensity distribution is an equal mixture of the background and foreground intensity distributions. The image probability under this model is therefore similar to that of the missing spot model except that the Dirichlet parameters \( \{ w_k \} \) are set to be the average of \( \{ u_k \} \) and \( \{ v_k \} \),

\[
P(I_j | \mathcal{H}_2) = \frac{\prod_k \Gamma(w_k + m_k) \Gamma(\sum_k w_k)}{\Gamma(\sum_k w_k + m) \prod_k \Gamma(w_k)}. \tag{4.23}
\]

If we refer to the image model described in Section 4.3 as \( \mathcal{H}_0 \), the approximate evidence for this model can be written as

\[
P(I_j | \mathcal{H}_0) \approx \exp(\mathcal{L}(Q)). \tag{4.24}
\]

If we assume that each subimage \( I_j \) was generated from one of these three models, then the posterior probability for the \( i \)th model is

\[
P(\mathcal{H}_i | I_j) = \frac{P(I_j | \mathcal{H}_i)P(\mathcal{H}_i)}{\sum_{k=0}^3 P(I_j | \mathcal{H}_k)P(\mathcal{H}_k)}. \tag{4.25}
\]

For simplicity, \( P(\mathcal{H}_i) \) was chosen to be uniform and each spot was flagged as NORMAL, MISSING or BAD (i.e. obscured) based on the model that had the highest posterior probability. The uncertainty in this flag state is not currently maintained as it is difficult for further processing stages to make use of it. Certainly, no existing tools are capable of maintaining many hypotheses about spot states during further processing.

### 4.5.2 Updating the prior parameters of the model

Because each sub-grid is printed by one pin of the arrayer, it is reasonable to assume that all the spots in a particular sub-grid are of a similar size and have similar deviations in position from a regular array. This assumption could be encoded in our model by the addition of shared hyper-hyper-priors over the parameters \( m_c, \beta_c, m_r, \beta_r \) and suchlike. Posterior distributions over these parameters could then be inferred using variational message passing. However, the addition of these latent variable nodes in the graph would prevent inference being carried out separately for each spot. It was decided, for the sake of simplicity, just to update the parameters \( m_c, \beta_c, m_r, \beta_r \) from the results of one pass of the algorithm and then reapply using these new parameter settings. These new prior parameters provide a much stronger learned prior and give an algorithm that is more robust to noise than one based on any fixed setting of these parameters.
The purpose of image analysis is not to find spot locations and sizes but to determine their intensities. The intensity $E_j$ of the $j$th spot will be a function of the spot parameters $\theta_j = \{c_j, r_j\}$ and the subimage $I_j$

$$E_j = f(I_j, \theta_j). \quad (4.26)$$

For each spot, there is uncertainty in the parameters $\theta_j$. We cannot therefore solve Equation 4.26 directly, but can only compute the expectation of $E_j$ under the approximate posterior distribution over $\theta_j$,

$$\langle E_j \rangle_{Q(\theta_j | I_j)} = \int f(I_j, \theta_j)Q(\theta_j | I_j) \, d\theta_j. \quad (4.27)$$

Our estimate of the posterior distribution over $\theta_j$ is available as a set of samples from this distribution $\{\theta_j^{(1)}, \theta_j^{(2)}, \ldots, \theta_j^{(S)}\}$ with corresponding importance weights $\{w_1, w_2, \ldots, w_S\}$. Importance sampling dictates that the above expectation is approximated by

$$\langle E \rangle_{P(\theta_j | I)} \approx \sum_{i=1}^{S} w_i f(I_j, \theta_j^{(i)}). \quad (4.28)$$

The function $f$ is typically chosen to be the mean or median intensity of all the spot pixels. To give an indication of the accuracy of this intensity value, its variance can be found using $\text{var}(E_j) = \langle E_j^2 \rangle - \langle E_j \rangle^2$.

### 4.5.4 Spot-finding results

The results of the microarray image analysis algorithm on two test sub-grid images are shown in Figure 4.3. The ellipses drawn over the image show the expected spot size and shape under the approximate posterior distribution given by the inference algorithm. The ellipses are coloured according to the spot states: NORMAL spots are green, BAD spots are red and MISSING spots are yellow. Spots which were found to be missing are marked using ellipses which are the average shape and size of all non-missing spots in the sub-grid.

The two images have very different noise characteristics. In particular, the right hand sub-grid has a high level of background noise including scatter flares. Nonetheless, the algorithm has located the spots with good accuracy given the level of noise.

### 4.6 Automatic Sub-grid Location

The spot-finding algorithm described above requires a set of approximate spot positions as a starting point. These can be obtained, for example, by requiring the user to locate a regular array of circles over each entire sub-grid. Whilst this is clearly much quicker than locating each spot individually, it is still time-consuming given that each slide image typically contains tens of sub-grids. If our goal is to automate the analysis of these images, then we should certainly aim to be able to find this approximate initialisation automatically.
4.6. AUTOMATIC SUB-GRID LOCATION

Figure 4.3: Results of the microarray image analysis algorithm on two test sub-images with different noise characteristics. The ellipses show the expected spot size and position under the approximate posterior distribution. The ellipses are coloured green for NORMAL spots, red for BAD (obscured) spots and yellow for MISSING spots. The results show that the algorithm is robust even to high levels of background noise.

As before, let us assume that we have an image \( I \) that contains only one entire sub-grid of spots (there may be other partial sub-grids). The printing and scanning process will introduce distortions in the image so that each sub-grid is not an exactly axis-aligned, rectangular array of spots. Indeed, the array may be translated, scaled, rotated, sheared or distorted in a non-linear fashion. However, we shall ignore non-linear effects and assume that the distortion can be modelled by an affine (linear) transform well enough to give a good approximation of spot locations. This assumption holds in the test images used because the non-linear distortions are not significant over the scale of individual sub-grids. The aim of automatic sub-grid location will therefore be to learn the affine transform which gives the best approximation of spot locations.

4.6.1 The sub-grid transform and its prior

The affine transform \( T \) gives a mapping from the physical slide co-ordinates \((x, y)\) in millimetres to image co-ordinates \((u, v)\) in pixels, defined as follows:

\[
\begin{pmatrix}
    u \\
    v \\
    1
\end{pmatrix} =
\begin{pmatrix}
    m_{00} & m_{01} & c_x \\
    m_{10} & m_{11} & c_y \\
    0 & 0 & 1
\end{pmatrix}
\begin{pmatrix}
    x \\
    y \\
    1
\end{pmatrix}
\overset{\text{def}}{=} T
\begin{pmatrix}
    x \\
    y \\
    1
\end{pmatrix}.
\]

The vector \( c = (c_x, c_y) \) contains the image co-ordinates of the centre of the sub-grid. The matrix \( M = (m_{00}, m_{01}; m_{10}, m_{11}) \) represents any rotation, scale or skew introduced by the printing and scanning process.
The inference task is to find the posterior distribution over the latent variable \( T \), so we must define a prior over \( T \). We assume independence between all the parameters

\[
P(T \mid \mathcal{H}) = P(c \mid \mathcal{H})P(M \mid \mathcal{H})
\]

\[
P(c \mid \mathcal{H}) = \mathcal{N}(c_x \mid 0, \sigma^2_x)\mathcal{N}(c_y \mid 0, \sigma^2_y)
\]

\[
P(M \mid \mathcal{H}) = \text{Gamma}(m_{00} \mid a_0, b_0)\text{Gamma}(m_{11} \mid a_1, b_1) \times \mathcal{N}(m_{01} \mid 0, \sigma^2_0)\mathcal{N}(m_{10} \mid 0, \sigma^2_1)
\]

where the standard deviations \( \sigma_x \) and \( \sigma_y \) in the sub-grid centre co-ordinates were set to be equal to half the distance between the sub-grids in the \( x \) and \( y \) directions. Suitable values for the other parameters were found to be \( a_0 = b_0 = a_1 = b_1 = 10, \sigma_0 = \sigma_1 = 0.02 \).

### 4.6.2 Inferring the sub-grid transform

For any sensible choice of likelihood function \( P(I \mid T) \), the posterior distribution over \( T \) will have a number of local maxima corresponding to translations of the sub-grid by one or more rows or columns from the true position. This means that gradient-based or local inference methods cannot be used initially as they would almost certainly get caught in one of these local maxima. Instead, we must use a procedure that explores all of the posterior modes sufficiently well to find the one containing the global maximum. Once this has been achieved, we can return to a gradient-based or local inference method to find the maximum of this mode and so find a MAP solution for \( T \). The posterior distribution is fairly tightly peaked around each local maximum and hence there is little point in retaining the uncertainty in \( T \) for a given mode, especially given that we are only looking for a rough initialisation. Note that as we do not require a posterior distribution over \( T \) and we cannot use local inference methods, Variational Message Passing will not be used in this case. Instead, the approach presented here is to start by exhaustively searching the space of \( T \) using a likelihood function that is extremely rapid to compute. This gives a solution which should be in the same region as the MAP solution. Then, conjugate gradient ascent with a more computationally expensive (but higher quality) likelihood function is used to find the MAP solution.

In each case, the transform \( T \) is used to divide the image into spot pixels \( S \) and background pixels \( B \). As in spot-finding, we assume independence between pixels and therefore the likelihood function for the entire image can be written as the product of each pixel’s likelihood

\[
P(I \mid T, \mathcal{H}) = \prod_{b \in B} P_B(I_b) \prod_{s \in S} P_S(I_s),
\]

where \( P_B(I_b) \) is the likelihood function for background pixel intensity and \( P_S(I_s) \) is the likelihood function for spot pixel intensity. It is convenient to work with the log-likelihood as
4.6. AUTOMATIC SUB-GRID LOCATION

Figure 4.4: (a) The sum of pixel intensities within the shaded rectangle D can be found from the values of the integral image at each corner. The value at corner 1 is A, at corner 2 it is A + B, at corner 3 it is A + C and at corner 4 it is A + B + C + D. The sum within D is thus 4 + 1 − 2 − 3. (b) Enlargement of part of a transformed sub-grid showing how the spot ellipses can be reasonably well approximated by rectangular regions. Note that the sub-grid has been both rotated and sheared.

then the pixel log-likelihoods can simply be added together

$$\log P(I \mid T, \mathcal{H}) = \sum_{b \in B} \log P_B(I_b) + \sum_{s \in S} \log P_S(I_s).$$

(4.34)

4.6.3 Searching through transform space

The first step of our inference procedure requires us to search through the space of possible transforms. To evaluate a likelihood function directly on the individual pixel intensities would involve at least as many operations as the number of pixels in the image (typically in excess of 200,000 operations). Even if we limited the operation to a simple addition, this would not be an efficient function to evaluate and so not suitable for use with a search.

Instead, inspired by the work of Viola and Jones [2001], the image is first transformed into an intermediate representation known as an integral image. The value of the integral image at \((x, y)\) is equal to the sum of all pixel intensities above and to the left of that location in the original image:

$$i(x, y) = \sum_{x' \leq x} \sum_{y' \leq y} I(x', y').$$

(4.35)

The sum of pixel intensities in any axis-aligned rectangular region with corners \((x_1, y_1)\) and \((x_2, y_2)\) can then be found using just four values of the integral image (see Figure 4.4a)

$$i_{\text{rect}}(x_1, y_1, x_2, y_2) = i(x_2, y_2) + i(x_1, y_1) - i(x_1, y_2) - i(x_2, y_1).$$

(4.36)
A likelihood function using the integral image

We can exploit the speed of calculating these rectangular area sums by choosing a likelihood function based on intensity sums and approximating our transformed spots with rectangles. Despite the fact that transformed spots are ellipses, their approximation by rectangles can be quite good if the transformation is not too extreme, as shown in Figure 4.4b. Each rectangle is centred on the spot centre and set to have the same area and aspect ratio as the transformed spot. The (approximate) sum of all spot pixels $\sum_s I_s$ is then found by adding the intensity sums for each spot rectangle, given by the integral image. The sum of all background pixel intensities $\sum_B I_b$ is found by subtracting this from the sum of all pixel intensities (which is the value in the lower-right corner of the intensity image):

$$\sum_{b \in B} I_b = i(x_{\text{max}}, y_{\text{max}}) - \sum_{j=1}^J i_{\text{rect}}(x_{j,1}, y_{j,1}, x_{j,2}, y_{j,2}),$$

where the corners of the rectangle approximating the $j$th spot are $(x_{j,1}, y_{j,1})$ and $(x_{j,2}, y_{j,2})$.

Now we assume that the background pixels have an intensity distribution with a peak close to zero and which decreases monotonically with intensity. This sort of distribution can be modelled using a truncated exponential distribution

$$P_B(I_b) = \frac{1}{\beta} \exp\left(-\frac{I_b}{\beta}\right) 0 \leq I_b \leq I_{\text{max}}$$

where $\beta$ is a scale parameter and $I_{\text{max}}$ is the maximum intensity value. The log-likelihood of all background pixels is then

$$\log P(B \mid \mathcal{H}) = -\frac{1}{\beta} \sum_{b \in B} I_b - |B| [\log \beta - \log(1 - \exp(-I_{\text{max}}/\beta))].$$

In Equation 4.39, the only dependence on the image is the sum of the background pixel intensities and the likelihood can be readily calculated from the integral image.

As spot pixels can have any intensity, they are modelled by a uniform distribution between 0 and $I_{\text{max}}$, giving $P_S(I_s) = 1/I_{\text{max}}$.

Search by regular sampling

Bayes’s Theorem gives the posterior distribution over $\mathbf{T}$ to be

$$P(\mathbf{T} \mid I, \mathcal{H}) \propto P(I \mid \mathbf{T}, \mathcal{H})P(\mathbf{T} \mid \mathcal{H}),$$

where the proportionality has been introduced as we are ignoring the evidence term $P(I \mid \mathcal{H})$ which does not depend on $\mathbf{T}$. We now need to search through the space of $\mathbf{T}$ to find a transform $\mathbf{T}$ that maximises this posterior probability. Even though we have an extremely efficient likelihood function, this multi-dimensional search is only possible because we need
4.6. AUTOMATIC SUB-GRID LOCATION

to search through only those transforms which are close to the identity (as image distortions are relatively small). As only small shears and rotations occur, just four dimensions were considered: the two location and two scale parameters. This four-dimensional space is divided into a regular grid and the posterior evaluated at each point in the grid, a procedure known as regular sampling. In all, the posterior is evaluated at \( \sim 64,000 \) values of \( T \) and the one that gives the maximum value is taken to be the approximate solution \( \hat{T} \).

4.6.4 Finding the MAP solution

The second step of the inference procedure involves finding a MAP solution by refining \( \hat{T} \) using conjugate gradient ascent.\(^6\) This method will find the local maximum of the posterior in the region of the approximate solution \( \hat{T} \), which should be the overall Maximum A Posteriori solution. To do this, new background and spot intensity distributions are used whose gradient can be computed

\[
P_B(I_b) = \mathcal{N}(I_b | 0, \sigma^2) \tag{4.41}
\]
\[
P_S(I_s) = \mathcal{N}(I_s | I_{\text{max}}, \sigma^2). \tag{4.42}
\]

If we consider the pixel at image location \( x \), the corresponding point on the physical slide is \( u = T^{-1}x \), for a particular transform \( T \). We define a function \( \mu(u) \) to be equal to \( I_{\text{max}} \) if \( u \) is inside a spot (according to the original configuration of the printing device) and zero elsewhere. At this stage we no longer approximate the spot ellipse by a rectangle. The log likelihood for any pixel is then written as

\[
\log P(I(x) | T) = \log \mathcal{N}(I(x) | \mu(u), \sigma^2) = -\frac{(I(x) - \mu(u))^2}{2\sigma^2} + \text{const.} \tag{4.43}
\]

The gradient of this pixel log likelihood function w.r.t. \( T \) is

\[
\frac{d}{dT} \log P(I(x) | T) = -\frac{(I(x) - \mu(u))}{\sigma^2} \nabla I(x) \ u^T. \tag{4.44}
\]

The function \( \nabla I(x) \) is the two-dimensional gradient of the image intensity at \( x \) which can be approximated using vertical and horizontal Sobel filters [Nalwa 1993]. The gradient of the entire log likelihood function is found by simply summing over all pixels,

\[
\frac{d}{dT} \log P(I | T) = \sum_x \frac{d}{dT} \log P(I(x) | T). \tag{4.45}
\]

This gradient can then be used with a conjugate gradient method to find the MAP solution \( T_{\text{MAP}} \). Conjugate gradient methods are efficient in that they only require a few evaluations of the gradient. This efficiency means that, although calculating the gradient requires a

\(^6\)Conjugate gradient descent is described in Bishop [1995].
4.6. AUTOMATIC SUB-GRID LOCATION

Figure 4.5: (a) The result of using the sub-grid location algorithm on a scaled version of a test slide image. Individual sub-grids are found, which allows initial sub-grid images to be extracted (such as the image on the right). Unfortunately, the assumption of an affine imaging transform combined with extreme noise effects at the edge of many slide images prevents this algorithm from working well in general and it can be necessary to perform this initial step manually. (b) A section of a scanned image which contains a single sub-grid and the sub-grid outline corresponding to the MAP transform found by the inference algorithm. The algorithm has correctly located the sub-grid within the image. The algorithm can be applied to image sections which have been extracted automatically, as in (a), or manually.

calculation for each pixel and is not particularly rapid, the overall process is still very quick. When extremely noisy images are used, an additional step can be added to ensure that the MAP solution has been found and to avoid off-by-one-row/column errors. The four transforms corresponding to shifting $\mathbf{T}_{\text{MAP}}$ one column to the left or right or one row up or down are used as initial points for the conjugate gradient algorithm. If any of these leads to a solution with higher posterior probability than $\mathbf{T}_{\text{MAP}}$ then it is chosen to be the new $\mathbf{T}_{\text{MAP}}$ and the procedure is repeated. Otherwise, the existing $\mathbf{T}_{\text{MAP}}$ is used.

4.6.5 Results for sub-grid finding

Figure 4.5b shows the outline of a transformed sub-grid found using this method. Unfortunately, it is difficult to assess this algorithm’s performance quantitively as I have been unable
to find any competing algorithms that perform the identical task. In addition, as the output is only approximate, it is difficult to compare the results of two algorithms, except qualitatively (i.e. whether the result coincided with the sub-grid or was incorrectly placed). In practice, the above algorithm was able to locate sub-grids correctly in the vast majority of images. The only images it failed on were test images where entire edge rows or columns were extremely faint. The presence of calibration spots in standard sub-grid images would normally prevent this from occurring.

### 4.6.6 Overall sub-grid location

The sub-grid location method described above relies on having an image with only one entire sub-grid in it. To extract such an image automatically requires an initial step which determines the approximate location of all the sub-grids within the entire slide image.

Once again, we would like to automate this process. One possibility is to re-apply the method of finding sub-grids to an image of the entire slide. We assume that the sub-grids themselves are arranged in a rectangular array which has been transformed by an unknown affine transform. Based on this assumption, the above algorithm can simply be applied to a scaled version of the slide image, so that the algorithm finds an array of sub-grids rather than spots. Figure 4.5a shows the output of the algorithm on an example slide image.

This method only gives an approximate sub-grid position (largely due to the assumption of an affine transform) and thus the extracted image used is the rectangle that contains the sub-grid enlarged in all directions by a small margin. This margin is currently set to twice the distance from spot centre to spot centre.

This method has been found to be effective on relatively undistorted slides with low background noise. Unfortunately, on other slide images it does not perform well. On an entire slide image, the assumption that the imaging transform is affine can be a poor one when there are large scale non-linear distortions. Additionally, there can be extreme noise artefacts around the edge of the slide image, which are not modelled well by the simple background model described earlier. To provide a reliable automatic solution for this step would involve learning a non-linear transform (i.e. a warp) and having a more complex intensity model. The alternative is to require the user to specify a linear transform and use a larger extracted image. As this requires just three mouse clicks per entire slide image, automating this step is not critical.

### 4.7 Discussion

In Sections 4.1–4.6, I have presented an algorithm for microarray image analysis that is based on an extension of Variational Message Passing which incorporates importance sampling to handle non-exponential family distributions. The algorithm is capable of identifying whether a spot is missing or obscured by noise artefacts and has been shown to perform well even in microarray images with a high level of background noise. In addition, I have shown that the
VMP algorithm can be initialised with a set of rough spot locations using a procedure that is either automatic for fairly clean slide images or requires minimal user interaction (three mouse clicks) for noisy, distorted slide images. This provides considerable time savings compared to using a procedure which requires locating each sub-grid by hand.

### 4.8 Gene Expression Data Analysis

The analysis of microarray images leads to a large quantity of gene expression data, along with appropriate measures of the certainty of those data (such as the standard deviation of the error in each measurement). The next step is to organise, analyse and visualise this data to reach conclusions about the biological processes being studied. The methods which can be used to achieve this are as open ended as the range of biological processes available to study. To date, a variety of methods have been used including:

- clustering by correlation/mutual information [Eisen et al. 1998; Spellman et al. 1998; Michaels et al. 1998]
- graph-based/hierarchical clustering [Ben-Dor et al. 1999; Bar-Joseph et al. 2001]
- Gaussian mixture model clustering [Yeung et al. 2001]
- self-organising maps [Tamayo et al. 1999]
- dimensionality reduction (PCA, ICA) [Raychaudhuri et al. 2000; Hori et al. 2001]
- latent variable modelling [Martoglio et al. 2002].

The majority of existing approaches do not take into account the uncertainty in the expression level data and do not provide a rigorous way of comparing different models for the data. It is an ongoing theme of this thesis that data analysis should be carried out by proposing probabilistic models, performing Bayesian inference to learn model parameters and comparing models using Bayesian model selection. It follows that analysis of gene expression data should also proceed along these lines and, indeed, this approach is starting to be used by some researchers. Hartemink et al. [2001] discuss the use of Bayesian networks as models of biological function which allow handling of uncertain expression data and rigorous comparison of different models whilst also permitting the introduction of latent variables (such as protein levels). Friedman et al. [2000] have used Bayesian networks to model the *S. cerevisiae* cell-cycle measurements of Spellman et al. [1998] and were able to capture much richer structure from the data than clustering methods, despite using models with no latent variables.

If probabilistic models in general, and Bayesian networks in particular, are to be used for gene expression data analysis, then it follows that the Variational Message Passing algorithm can be applied to rapidly perform approximate inference and model selection on novel models, provided they are conjugate-exponential or can be made so. To demonstrate the ease of use of this algorithm, I now present an example of using Variational Message Passing to perform
Independent Component Analysis on a small gene expression data set. The aim of this example is to provide a short illustration of how VMP allows complex models to be quickly constructed and applied to real data sets, rather than to break new ground in gene expression data analysis.

### 4.8.1 ICA of gene expression data using VMP

When applying Independent Component Analysis (ICA) to gene expression data, the pattern of gene expression for each tissue is represented as a linear superposition of a small number of underlying patterns or *signatures*. Unlike when using Principal Component Analysis, these signatures are not constrained to be orthogonal but are instead assumed to have amplitudes that are statistically independent of each other.

The core assumption of ICA, therefore, is that our gene expression data \( \mathbf{X} = (\mathbf{x}_1 \ldots \mathbf{x}_N)^\mathsf{T} \) can be modelled as a linear combination of signatures \( \mathbf{S} = (\mathbf{s}_1 \ldots \mathbf{s}_M)^\mathsf{T} \) plus some Gaussian noise \( \tau \), so that

\[
\mathbf{X} = \mathbf{W}^\mathsf{T} \mathbf{S} + \tau \tag{4.47}
\]

where each column of \( \mathbf{W} \) gives the amounts of each signature present in the corresponding tissue sample. The aim is to infer the signatures \( \mathbf{S} \), the amplitude matrix \( \mathbf{W} \) and the number of signatures \( M \).

Following Miskin [2000], the rows of \( \mathbf{W} \) are modelled using \( M \) Gaussian mixture models with \( C \) components, each of which has the form described in Section 1.8.7. The number of signatures is found by using an Automatic Relevance Determination prior \( \alpha_m \) on each signature (each row of \( \mathbf{S} \)) which allows signatures to be switched off if their presence is not supported by the data. The Bayesian network for this ICA model is shown in Figure 4.6.

![Bayesian network for ICA model](image)

**Figure 4.6:** The Bayesian network for the Independent Component Analysis model. Each gene expression vector \( \mathbf{x}_n \) is viewed as a linear superposition of signatures \( \mathbf{s}_m \). The hyperparameter \( \alpha_m \) controls which of the signatures are switched off and so allows the number of signatures to be determined. The elements in each row of the amplitude matrix \( \mathbf{W} \) are modelled using a mixture of \( C \) Gaussians with parameters \( \{\mu_{m,c}, \gamma_{m,c}\}_{c=1}^C \), where the means are set to be zero. The reconstruction error is modelled as being Gaussian with precision \( \tau \).
ICA model applied to ovarian tissue samples data

Now that the Bayesian network which we are using to model the data has been specified, Variational Message Passing allows inference to proceed automatically for any supplied data set. This will now be demonstrated on a small data set consisting of the gene expression levels of 175 genes in 17 tissue samples from Martoglio et al. [2000]. The tissue set consists of ovarian samples, some of which are tumourous, as described in Table 4.1.

<table>
<thead>
<tr>
<th>Tissue Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Normal (pre-menopausal)</td>
</tr>
<tr>
<td>2-5</td>
<td>Normal (post-menopausal)</td>
</tr>
<tr>
<td>6-10</td>
<td>Serous Papillary Adenocarcinoma (SPA)</td>
</tr>
<tr>
<td>11-14</td>
<td>Poorly Differentiated SPA (PD-SPA)</td>
</tr>
<tr>
<td>15</td>
<td>Benign Serous Carcinoma (BSC)</td>
</tr>
<tr>
<td>16-17</td>
<td>Benign Mucinous Carcinoma (BMC)</td>
</tr>
</tbody>
</table>

Table 4.1: Descriptions of the tissue samples in the ovarian tissue data set

The ICA model converged in about 200 iterations. A Hinton diagram showing the expected value of $W$ under the optimised variational posterior is shown in Figure 4.7. As can be seen from this diagram, only 7 out of a possible 17 signatures have been retained.

Figure 4.7: Hinton diagram of the expected $W$ amplitude matrix under the variational posterior when the ICA model is trained on a data set of ovarian tissue samples. The rows correspond to the 17 possible signatures – of which only 7 have been used. The columns show how much of the signatures are present in the each of the 17 tissue samples.
Biological interpretation of the inferred gene signatures

The ICA model assumes that the overall gene expression profile of each tissue is due to the superposition of the gene expressions of a number of independent biological processes. It follows that the amplitude matrix $W$ represents the level of activity of these processes in each tissue sample. By comparing the activity of a signature to the known characteristics of each tissue, it is possible to infer broadly which biological process the signature represents and therefore what genes are associated with that process.

Firstly, consider the fourth signature whose activity and gene expression levels are shown in Figure 4.8. This signature is present at a near-constant level in all of the samples. In addition, the signature contains only positive expression levels for all genes. This signature can therefore be interpreted as representing the genes which are expressed in all ovarian tissues at any time. Such genes are referred to as housekeeping genes as they are responsible for essential cell function such as the maintenance of cell cycle, metabolism and so on.

![Hinton diagram showing the level of activity of the 4th signature in each of the 17 tissue samples. As can be seen, this signature has almost constant activity in all the tissue samples.](image1)

![Bar chart showing the expression levels (in arbitrary units) of each of the 175 genes for the 4th signature. This signature expresses all of the genes and so can be regarded as representing the housekeeping genes for all ovarian tissue samples.](image2)

Figure 4.8: (a) Hinton diagram showing the level of activity of the 4th signature in each of the 17 tissue samples. As can be seen, this signature has almost constant activity in all the tissue samples. (b) Bar chart showing the expression levels (in arbitrary units) of each of the 175 genes for the 4th signature. This signature expresses all of the genes and so can be regarded as representing the housekeeping genes for all ovarian tissue samples.

Secondly, consider the 8th signature (Figure 4.9) which is only strongly present in the first tissue sample. This sample is the only pre-menopausal sample in the data set and it seems likely, therefore, that this signature differentiates pre-menopausal from post-menopausal gene expression.

Finally, consider the 15th signature (Figure 4.10) which is only strongly present in the 15th tissue sample. This sample is the one sample in the data set from a Benign Serous Carcinoma and hence this signature may be indicative of the presence of such a tumour. Clearly, a larger data set would be required before any strong conclusions could be drawn concerning the biological interpretation of particular signatures.
4.8. GENE EXPRESSION DATA ANALYSIS

Figure 4.9: (a) Hinton diagram showing the level of activity of the 8th signature in each of the 17 tissue samples. The signature is only strongly present in the first sample. (b) Bar chart of the gene expression levels for the 8th signature. This signature is dominated by the expression of the 43rd gene (which codes for an Endothelin-1 receptor).

Figure 4.10: (a) Hinton diagram showing the level of activity of the 15th signature in each of the 17 tissue samples. The signature is only strongly present in the 15th sample which was from a Benign Serous Carcinoma. (b) Bar chart of the gene expression levels for the 15th signature.

4.8.2 Conclusion

This brief example has shown that Variational Message Passing allows rapid application of a plausible probabilistic model of gene expression to a small data set. The resultant set of gene expression signatures has allowed some tentative interpretation of the independent biological processes involved.

Overall, it has been shown that Variational Message Passing can be applied successfully both to analyse scanned images of microarrays and to interpret the resultant gene expression levels to reach conclusions about the underlying biological systems.


