

CHAPTER 5

STRUCTURED VARIATIONAL DISTRIBUTIONS

The fully factorised variational approximation has been widely used with great success in many applications; its use for image analysis and biological modelling has already been illustrated in this thesis. However, the fully factorised variational distribution does represent a somewhat restrictive approximation. It cannot, for instance, capture the posterior correlation between variables, since the approximation is fully separable.

The limitations associated with the fully factorised approximation can be overcome to a large extent by allowing a broader class of variational distributions which retain some of the structure of the original model. These will, in general, give a closer approximation to the true posterior distribution. However, it is essential that inference using this richer family of distributions remains computationally tractable.

In this chapter, the variational inference framework of Chapter 2 will be extended to allow the use of variational distributions in this richer family of structured distributions¹. This extension will enable the posterior dependencies between particular variables to be captured, improving the quality of the approximation and extending the number of applications where the framework can be applied.

5.1 Inference Using Structured Variational Distributions

The Variational Message Passing algorithm uses a variational distribution which is fully factorised, giving an approximate posterior which is separable with respect to individual variables. In general, an improved approximation will be achieved if a posterior distribution is used which retains some dependency structure. Whilst Wiegerinck [2000] has presented a general framework for such structured variational inference, he does not provide a general-purpose algorithm for applying this framework. In this chapter, I extend VMP to allow structured variational distributions and so provide such an algorithm.

¹The work presented in this chapter builds on a paper written in collaboration with Christopher M. Bishop [Bishop and Winn 2003].

Since this work was carried out, Xing et al. [2003] have presented a Generalised Mean Field (GMF) algorithm for structured variational inference. The GMF algorithm allows for the variational distribution to be factorised into clusters, giving an improved approximation over standard VMP. However, the GMF algorithm requires that the clusters do not overlap and so posterior dependencies cannot be captured between variables in different clusters. As will be described, Structured VMP does not have this constraint and allows for overlapping clusters, hence giving an improved approximation over that of the GMF algorithm.

5.1.1 Optimising structured variational distributions

To allow dependency structure to be retained, our variational distribution must now be defined to have a more general product-of-factors form (previously discussed in Section 1.4.2),

$$Q(\mathbf{H}) = \frac{1}{Z_Q} \prod_i \Phi_i(c_i) \quad (5.1)$$

where each c_i is a *cluster* (the small subset of variables that appear in a particular factor) and Φ_i the corresponding non-negative factor function, known as a *cluster potential*. The clusters are, in general, non-disjoint and their union contains all the latent variables, so $\bigcup_i c_i = \mathbf{H}$. Note that the joint probability distribution of any Bayesian network can be written in this form. The normalisation constant Z_Q is set to ensure that Q is a valid probability distribution:

$$Z_Q = \int \prod_i \Phi_i(c_i) d\mathbf{H}. \quad (5.2)$$

As in the fully factorised case, the goal is to maximise the lower bound $\mathcal{L}(Q)$, defined in Equation 1.40 as

$$\mathcal{L}(Q) = \langle \log P(\mathbf{H}, \mathbf{D}) - \log Q(\mathbf{H}) \rangle_Q. \quad (5.3)$$

Substituting our new form of Q and omitting the arguments of the cluster potentials gives

$$\mathcal{L}(Q) = \frac{1}{Z_Q} \int \prod_i \Phi_i \left\{ \log P(\mathbf{H}, \mathbf{D}) - \sum_i \log \Phi_i + \log Z_Q \right\} d\mathbf{H}. \quad (5.4)$$

The terms containing one cluster potential Φ_j can be separated out

$$\mathcal{L}(Q) = \frac{1}{Z_Q} \int_{c_j} \Phi_j \int_{\mathbf{H} \setminus c_j} \prod_{i \neq j} \Phi_i \left\{ \log P(\mathbf{H}, \mathbf{D}) - \sum_i \log \Phi_i + \log Z_Q \right\} d\mathbf{H}. \quad (5.5)$$

The aim is to maximise $\mathcal{L}(Q)$ subject to the normalisation constraint. This can be achieved by means of a Lagrange multiplier λ , so we seek instead to maximise

$$\bar{\mathcal{L}}(Q) = \mathcal{L}(Q) + \lambda \left(\frac{1}{Z_Q} \int \prod_i \Phi_i(c_i) d\mathbf{H} - 1 \right) \quad (5.6)$$

and so obtain the following stationarity condition

$$0 = \int \prod_{i \neq j} \Phi_i \left\{ \log P(\mathbf{H}, \mathbf{D}) - \sum_{i \neq j} \log \Phi_i - \log \Phi_j + \log Z_Q + 1 \right\} d\mathbf{H} \setminus c_j. \quad (5.7)$$

We can now solve to find the setting of the potential Φ_j^* that maximises $\mathcal{L}(Q)$ whilst all other potentials are held constant

$$\log \Phi_j^* = \left\langle \log P(\mathbf{H}, \mathbf{D}) - \sum_{i \neq j} \log \Phi_i \right\rangle_{\mathbf{H} \setminus c_j} + \text{const.} \quad (5.8)$$

As the original P distribution is defined as a Bayesian network, the joint probability can be written as

$$P(\mathbf{X}) = \prod_i P(X_k | \text{pa}_k) \quad (5.9)$$

where $\mathbf{X} = (\mathbf{H}, \mathbf{D})$ as before. This can be viewed as a product of potentials of the form $P(X_k | \text{pa}_k)$ with corresponding clusters $d_k = \{X_k, \text{pa}_k\}$. Substituting into Equation 5.8 and removing constant terms gives the final result (see also Wiegerinck [2000])

$$\log \Phi_j^* = \left\langle \sum_{k \in D_j} \log P(X_k | \text{pa}_k) - \sum_{i \in C_j} \log \Phi_i \right\rangle_{\mathbf{H} \setminus c_j} - z \quad (5.10)$$

where the summations have been restricted to be over only the sets of log potentials D_j and C_j , defined as follows. The set D_j contains all clusters d_k that have a non-zero intersection with the cluster c_j . Similarly, the set C_j contains all clusters c_i that intersect with c_j , excluding c_j itself. The constant z can be inferred from the global normalisation constraint of Equation 5.2. Thus, the update for a cluster depends only on overlapping clusters and can therefore be performed locally in the cluster graph, provided that the normalisation constant can also be found using local operations.

The expectation in Equation 5.10 is taken with respect to the conditional distribution defined by

$$Q(\mathbf{H} \setminus c_j | c_j) = \frac{\prod_{i \neq j} \Phi_i}{\int_{\mathbf{H} \setminus c_j} \prod_{k \neq j} \Phi_k}. \quad (5.11)$$

The new update equation is slightly more complex than the one derived for the fully factorised case (Equation 1.49) due to the addition of a second term in the expectation, relating to overlapping clusters in Q . If Q is fully factorised, all the clusters c_j are disjoint and so C_j is empty for all j . The second term then vanishes and we recover the original update equation for the fully factorised Q distribution.

5.1.2 Using a Bayesian network as a variational distribution

Consider the special case of Equation 5.1 where the Q distribution corresponds to a Bayesian network. In this case, the potentials have the form of conditional distributions $\Phi_i(c_i) = Q(X_i | \text{pa}_i)$ where $c_i = \{X_i, \text{pa}_i\}$. To ensure the potentials are valid conditional probability distributions, an additional local normalisation constraint must be introduced for each potential

$$\int \Phi_i(c_i) dX_i = 1. \quad (5.12)$$

Taking these additional constraints into account leads to a slightly different update equation

$$\log Q(X_j | \text{pa}_j)^* = \left\langle \sum_{k \in D_{X_j}} \log P(X_k | \text{pa}_k) - \sum_{i \in C_{X_j}} \log Q_i(X_i | \text{pa}_i) \right\rangle_{\mathbf{H} \setminus c_j} - z(\text{pa}_i) \quad (5.13)$$

where D_{X_j} is the set of all clusters d_k that depend on X_j and similarly C_{X_j} is the set of all clusters c_i that depend on X_j , excluding c_j itself. The local normalising factor $z(\text{pa}_i)$ can be inferred using Equation 5.12. In this case, the update for the j th cluster depends only on clusters in the Markov blanket of X_j in P and Q .

We must now consider how to choose a variational distribution that is tractable (in that it allows cluster potential updates to be computed analytically) whilst providing a good approximation to the true posterior distribution.

5.2 Choice of Structured Variational Distribution

There is a large range of possible structured variational distributions and so we will need to focus on tractable distributions that capture important dependencies in the P distribution. The approach that will be used here is to choose Q distributions whose dependency structure corresponds to sub-graphs of the original graphical model. It is anticipated that the dependency structure of the prior model will provide a good indication of the dependency structure of the posterior [Wiegerinck 2000; Saul and Jordan 1996]. The strategy, therefore, will be to take the original graphical model and to remove edges as required, in order to achieve tractability of inference.

To illustrate this approach, Figure 5.1 shows an example using the ASIA chest clinic Bayesian network from Lauritzen and Spiegelhalter [1988]. This example shows the improved approximation (in terms of KL divergence) given by using a structured Q distribution instead of a fully factorised one. The dependency structure of the original model has been partially retained by using a structured distribution of a tree whose edges are all present in the original graph.

In general, if we do not remove any of the edges of the original graph, then Q will simply equal P and inference will be intractable by assumption (we would not be using an approximate inference method if inference were tractable in P). Alternatively, removing all the edges

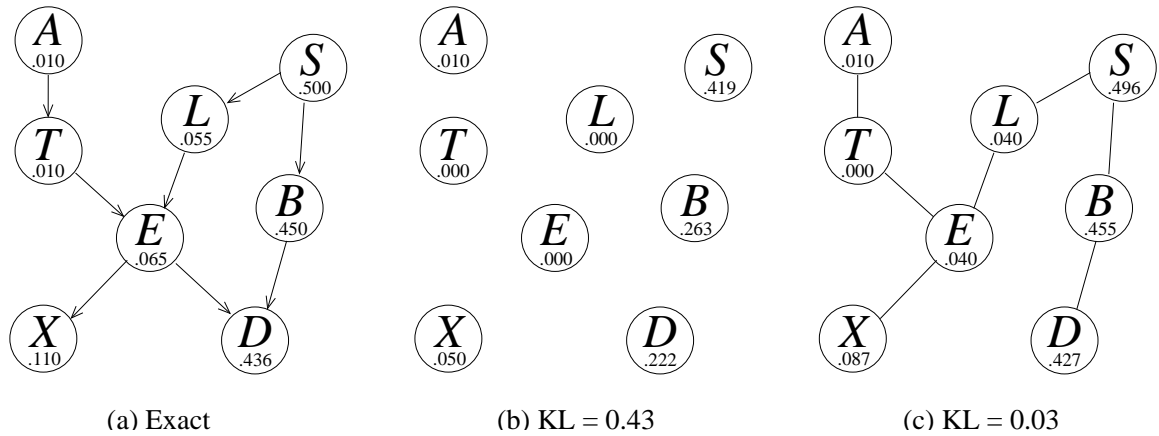


Figure 5.1: Example taken from Wiegierinck [2000] showing how a structured variational distribution gives a better approximation to the exact posterior than a fully factorised distribution. The model used is the ASIA chest clinic Bayesian network from Lauritzen and Spiegelhalter [1988]. (a) The Bayesian network for the model showing marginal probabilities under the exact posterior P . (b) The fully factorised approximation, with marginal probabilities under the variational distribution. (c) A structured approximation where Q is a tree, with marginal probabilities. For each approximation, the KL divergence between Q and P is given, showing the significant improvement in accuracy given by using the structured variational distribution.

of the original graph results in a fully-factorised Q distribution, which has been shown to lead to tractable inference for all conjugate-exponential models. At some point between these two extremes lies a variational distribution that retains the most structure possible whilst still remaining tractable. This distribution will give the best possible approximation to the true posterior distribution. This begs the question “which edges can be retained in Q without loss of tractability?” which will be addressed shortly. It is worth noting that, in practice, it may not always be preferable to use the distribution which gives the absolute best approximation. Retaining additional structure in Q comes with an associated penalty in terms of computational cost and memory requirements and so it makes more sense to use the Q distribution with the simplest structure that provides good results for a particular model and domain.

Given that we wish to use a Q distribution defined by a subgraph of a Bayesian network, it would seem natural to choose Q to be a Bayesian network also. In this case, the cluster potentials would be conditional probabilities $Q(X_i | \text{pa}_i)$, as described in Section 5.1.2. Unfortunately, in order to implement the variational update equations, it is necessary to compute appropriate expectations, which in turn requires that marginal distributions over small subsets of variables are available. In a Bayesian Network, message passing would be required to find such marginal distributions. What would be preferable is a form of Q where the cluster potentials themselves are marginal distributions. This can be achieved by using a Q distribution in the form of a *junction tree*.

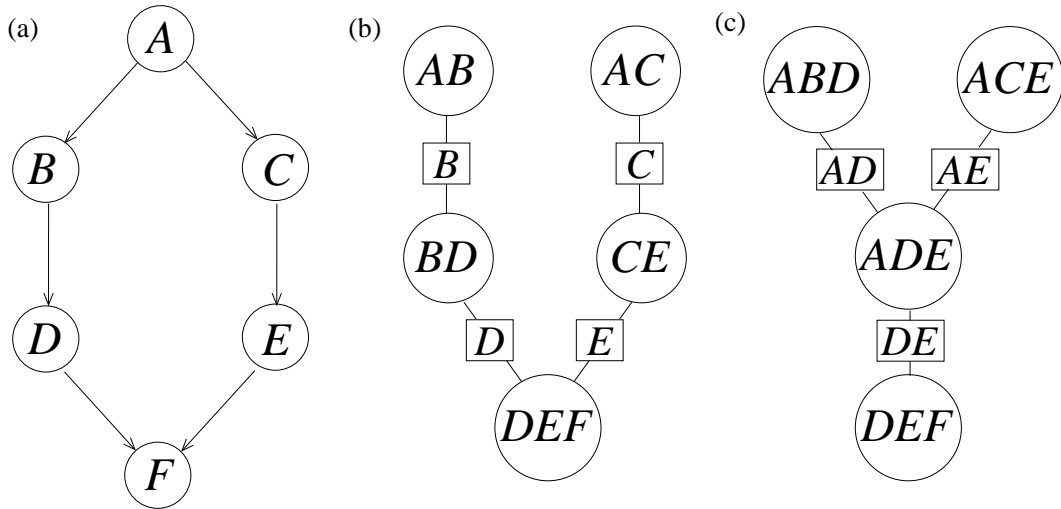


Figure 5.2: (a) A Bayesian network. (b) A cluster graph that captures the dependencies in the Bayesian network. The square nodes are separators, which contain the variables common to the two adjacent clusters. This cluster tree does not satisfy the running intersection property as the nodes on the path from AB to AC do not all contain the intersection variable A . (c) An alternative cluster tree that does obey the running intersection property and hence is a junction tree for the Bayesian network.

5.3 Variational Junction Trees

A junction tree [Jensen 1996; Cowell et al. 1999] is a tree-structured cluster graph that satisfies the *running intersection property*. This property states that for any pair of clusters c_i and c_j , all clusters on the path between c_i and c_j contain the intersection $c_i \cap c_j$. To illustrate the running intersection property, Figure 5.2b shows a cluster tree that does not satisfy this property, whilst Figure 5.2c shows one that does and which is therefore a junction tree. The clusters in a junction tree are connected via *separators* each of which contains the variables common to the two neighbouring clusters. The separator between two connected clusters c_i and c_j is referred to as s_{ij} .

A junction tree can be constructed from a Bayesian network by:

- *moralising* the graph by connecting all pairs of parents for every node and dropping the arrows on the edges (giving an undirected graph);
- *triangulating* the undirected graph by adding additional edges to ensure that every cycle of length four or more has a chord;
- forming a junction tree by using the cliques of the undirected graph as clusters.

As there are typically many possible ways of triangulating a graph, there can be many possible junction trees corresponding to a single Bayesian network. Figure 5.2a shows a Bayesian network and Figure 5.2c shows one possible corresponding junction tree.

A *consistent* junction tree is one where the potentials Φ_i and Φ_j of any two clusters c_i and c_j , which have a non-zero intersection $A = c_i \cap c_j$, satisfy

$$\int \Phi_i(c_i) dc_i \setminus A = \int \Phi_j(c_j) dc_j \setminus A. \quad (5.14)$$

This constraint means that the marginal potential over any subset of variables is the same no matter which cluster potential it is derived from. The running intersection property means that all nodes containing a particular set of variables must form a connected subgraph and so consistency can be enforced globally using local operations on the graph (i.e. message passing).

When using a junction tree as a variational distribution, Q has the form

$$Q(\mathbf{H}) = \frac{\prod_i \Phi_i(c_i)}{\prod_{j,k} \Phi_{jk}(s_{jk})} \quad (5.15)$$

in which the product in the numerator is over clusters and the product in the denominator is over separators. The separator potentials $\Phi_{ij}(s_{ij})$ are defined by (either) neighbouring cluster potential

$$\Phi_{ij}(s_{ij}) = \int \Phi_i(c_i) dc_i \setminus s_{ij}. \quad (5.16)$$

The end result of this form of Q and the consistency constraints is that the cluster potentials can be interpreted directly as marginal probability distributions:

$$\Phi_i(c_i) = Q(c_i). \quad (5.17)$$

This property of junction trees allows expectations to be calculated directly, provided that the variables involved can be found in a single cluster of the tree.

5.3.1 Inference using variational junction trees

It follows from Equation 5.10 that the cluster potential Φ_j^* can be optimised using

$$\log \Phi_j^* = \left\langle \sum_{k \in D_j} \log P(X_k | \text{pa}_k) - \sum_{i \in C_j} \log \Phi_i + \sum_{(l,m) \in S_j} \log \Phi_{lm} \right\rangle_{\mathbf{H} \setminus c_j} - z \quad (5.18)$$

where C_j and D_j are as defined previously and S_j is the set of separators which depend on c_j . The constant z is set to ensure that Φ_j^* is normalised. It is important to note, however, that having modified Φ_j^* , the junction tree is no longer consistent. To recover consistency, a `DistributeEvidence` procedure must be used with c_j as the root. In the junction tree, this involves sending messages from c_j to each neighbour by first updating the intervening separator

$$\Phi_{ij}^*(s_{ij}) = \int \Phi_j^*(c_j) dc_j \setminus s_{ij} \quad (5.19)$$

and then the neighbouring cluster using

$$\Phi_i^*(c_i) = \Phi_i(c_i) \frac{\Phi_{ij}^*(s_{ij})}{\Phi_{ij}(s_{ij})}. \quad (5.20)$$

Recursively, each neighbour then sends out messages to all its neighbours except the one from which the message came. When all clusters in the tree have received a message, the tree is consistent again.

The above analysis also applies in the case where Q consists of several disconnected junction trees. In this case, the Q distribution consists of a product of factors, one for each tree. Let \mathbf{T}_α be the set of nodes associated with a particular tree and $Q(\mathbf{T}_\alpha)$ the corresponding factor in the Q distribution so that

$$Q(\mathbf{X}) = \prod_{\alpha} Q(\mathbf{T}_\alpha). \quad (5.21)$$

Suppose that a cluster c_j is contained in \mathbf{T}_α . Its potential can be updated using

$$\log \Phi_j^* = \left\langle \sum_{k \in D_j} \langle \log P(X_k | \text{pa}_k) \rangle_{\beta \neq \alpha} - \sum_{i \in C_j} \log \Phi_i + \sum_{(l,m) \in S_j} \log \Phi_{lm} \right\rangle_{\mathbf{T}_\alpha \setminus c_j} - z \quad (5.22)$$

and the `DistributeEvidence` procedure need only be applied to clusters in \mathbf{T}_α . The expectation in the first term does not depend on $Q(\mathbf{T}_\alpha)$ and therefore this term will remain unchanged throughout the update of this tree.

It is, of course, also possible to update the separator potentials ϕ_{lm} using Equation 5.10. Consider that the set of separator variables s_{lm} is always a subset of each neighbouring cluster c_l and c_m . The optimisation of $Q(c_l)$ or $Q(c_m)$ is guaranteed to lead to at least as good an increase in \mathcal{L} as optimising $Q(s_{lm})$. This can be seen by writing $Q(c_l) = Q(s_{lm})Q(c_l \setminus s_{lm} | s_{lm})$ and hence optimisation of $Q(c_l)$ includes optimising just $Q(s_{lm})$ as a special case. Therefore any optimisation possible by updating separator potentials can be equalled or improved by updating neighbouring clusters and so we choose to update only cluster potentials. Furthermore, following changing a separator potential, it would be necessary to define a new algorithm for making the junction tree consistent.

5.4 Reducing Computation for Inference with VJTs

Optimising a potential using Equation 5.22 and performing `DistributeEvidence` throughout the entire tree after each update leads to a computationally expensive inference algorithm. For many common structures of P and Q , the resultant junction trees satisfy certain constraints which allow considerable savings to be made on the required computation. Where Q has been formed by deleting edges from P , two possible cases arise: where all deleted edges are between nodes in different junction trees of Q (Case I) and where at least one edge is between

nodes contained in the same junction tree (Case II). The modified update procedure and the resultant reduction in computation will now be discussed for these two cases.

5.4.1 Case I: Q junction tree with no internal deleted edges

I will now consider a special case of optimising $Q(\mathbf{T}_\alpha)$ for a junction tree \mathbf{T}_α . I start by defining $\hat{\psi}_k(d_k \cap \mathbf{T}_\alpha) = \langle \log P(X_k | \text{pa}_k) \rangle_{\beta \neq \alpha}$ for each d_k that intersects with \mathbf{T}_α . Now suppose that we have the case where, for each $\hat{\psi}_k$, there is at least one cluster c_j that contains the intersection $d_k \cap \mathbf{T}_\alpha$. When Q is formed by deleting edges of P , this is equivalent to stating that no edges have been deleted between nodes in \mathbf{T}_α . I then choose to assign each $\hat{\psi}_k$ potential to a cluster that contains $d_k \cap \mathbf{T}_\alpha$ and define A_i to be the set of such potentials assigned to the i th cluster. Thus, each $\hat{\psi}_k$ is assigned to exactly one A_i . Prior to the optimisation, the cluster and separator potentials are initialised using

$$\phi_i(c_i) \stackrel{\text{def}}{=} \log \Phi_i(c_i) = \sum_{k \in A_i} \hat{\psi}_k \quad (5.23)$$

$$\phi_{jk}(s_{jk}) \stackrel{\text{def}}{=} \log \Phi_{jk}(s_{jk}) = 0 \quad (5.24)$$

where the ϕ notation has been introduced as shorthand for the corresponding $\log \Phi$. The above initialisation means that the junction tree is not consistent. This can be rectified by performing `CollectEvidence` and then `DistributeEvidence` with respect to an arbitrarily chosen root cluster c_k . Following these two message passing procedures, the junction tree will be consistent. It is important to note, however, that neither procedure changes the overall distribution $Q(\mathbf{T}_\alpha)$ and so the potentials are related by

$$\sum_i \phi_i - \sum_{l,m} \phi_{lm} = \sum_k \hat{\psi}_k \quad (5.25)$$

where the first sum is over clusters in \mathbf{T}_α and the second is over separators in \mathbf{T}_α . Now consider applying Equation 5.22 to update the potential for the j th cluster

$$\phi_j^* = \left\langle \sum_{k \in D_j} \hat{\psi}_k - \sum_{i \in C_j} \phi_i + \sum_{(l,m) \in S_j} \phi_{lm} \right\rangle_{\mathbf{T}_\alpha \setminus c_j} - z, \quad (5.26)$$

and then extending the summations to include all potentials which intersect with \mathbf{T}_α , except ϕ_j itself. The additional potentials are all constant with respect to c_j and so only the normalisation constant z needs to be changed:

$$\phi_j^* = \left\langle \sum_k \hat{\psi}_k - \left(\sum_{i \neq j} \phi_i - \sum_{l,m} \phi_{lm} \right) \right\rangle_{\mathbf{T}_\alpha \setminus c_j} - z'. \quad (5.27)$$

Substituting in from Equation 5.25 gives

$$\phi_j^* = \left\langle \sum_k \hat{\psi}_k - \left(\sum_k \hat{\psi}_k - \phi_j \right) \right\rangle_{\mathbf{T}_\alpha \setminus c_j} - z' = \phi_j \quad (5.28)$$

and hence the update has left the potential ϕ_j unchanged.

It follows that, in the situation where there are no deleted edges in \mathbf{T}_α , the factor $Q(\mathbf{T}_\alpha)$ can be optimised using just the initialisation defined above and the `CollectEvidence` and `DistributeEvidence` message-passing procedures. It is therefore true that if Q consists of a single tree \mathbf{T} then the condition that there are no deleted edges is equivalent to stating that Q is a junction tree for P . The above initialisation and message-passing then corresponds to the procedure used for belief propagation. In short, belief propagation is a special case of this more general algorithm.

5.4.2 Case II: Q junction tree with internal deleted edges

The alternate case to the one considered above is when edges *are* deleted between nodes in a tree \mathbf{T}_α . This means that there is at least one potential $\hat{\psi}_k(d_k \cap \mathbf{T}_\alpha)$ where the intersection $d_k \cap \mathbf{T}_\alpha$ is not contained in any cluster. The set of such potentials shall be denoted B_α and the set of all other potentials A_α , so that each potential lies in either B_α or A_α .

We assign each potential $\hat{\psi}_k$ in A_α to a cluster that contains $d_k \cap \mathbf{T}_\alpha$, as before. Again, we define A_i to be the set of potentials in A_α assigned to cluster c_i . We also define a set B_i to be the set of potentials in B_α whose variables intersect with c_i . Note that each potential in A_α appears in only one A_i whilst each potential in B_α may appear in one or more B_i .

We initialise the cluster and separator potentials as before

$$\phi_i(c_i) = \sum_{k \in A_i} \hat{\psi}_k \quad (5.29)$$

$$\phi_{jk}(s_{jk}) = 0. \quad (5.30)$$

This initialisation leads to an inconsistent junction tree and so we perform `CollectEvidence` and then `DistributeEvidence` with respect to an arbitrarily chosen root cluster. At this stage, the tree is consistent and the overall distribution $Q(\mathbf{T}_\alpha)$ and the $\hat{\psi}$ potentials are related by

$$\sum_i \phi_i - \sum_{l,m} \phi_{lm} = \sum_{k \in A_\alpha} \hat{\psi}_k. \quad (5.31)$$

Now we consider updating a potential ϕ_j which has no intersection with any of the potentials in B_α (i.e. $B_j = \emptyset$). We apply Equation 5.22 and extend the summations as in the previous section except that the first summation is extended to be only over A_α .

$$\phi_j^* = \left\langle \sum_{k \in A_\alpha} \hat{\psi}_k - \sum_{i \neq j} \phi_i + \sum_{l,m} \phi_{lm} \right\rangle_{\mathbf{T}_\alpha \setminus c_j} - z'. \quad (5.32)$$

Substituting in Equation 5.31 gives

$$\phi_j^* = \left\langle \sum_{k \in A_\alpha} \hat{\psi}_k - \left(\sum_{k \in A_\alpha} \hat{\psi}_k - \phi_j \right) \right\rangle_{\mathbf{T}_\alpha \setminus c_j} - z' = \phi_j \quad (5.33)$$

and so, once again, the update has left the potential ϕ_j unchanged.

Unfortunately, updating a potential which does intersect with at least one of the potentials in B_α is not as straightforward. Suppose ϕ_j is now such a potential (and so $B_j \neq \emptyset$), the update equation becomes

$$\phi_j^* = \left\langle \sum_{k \in A_\alpha} \hat{\psi}_k + \sum_{k \in B_j} \hat{\psi}_k - \sum_{i \neq j} \phi_i + \sum_{l,m} \phi_{lm} \right\rangle_{\mathbf{T}_\alpha \setminus c_j} - z'. \quad (5.34)$$

Now let us apply this to a particular potential ϕ_m which we wish to update first. As we have just initialised all potentials, Equation 5.34 becomes

$$\phi_m^* = \phi_m + \sum_{k \in B_m} \langle \hat{\psi}_k \rangle_{\mathbf{T}_\alpha \setminus c_m} - z \quad (5.35)$$

where the expectation is with respect to the conditional distribution defined by

$$Q(\mathbf{T}_\alpha \setminus c_m | c_m) = \frac{Q(\mathbf{T}_\alpha)}{Q(c_m)} = \frac{\prod_{i \neq m} e^{\phi_i}}{\prod_{k,l} e^{\phi_{kl}}}. \quad (5.36)$$

As a potential has been changed, the junction tree is now inconsistent and we must perform `DistributeEvidence`(c_m). This procedure will modify all the other cluster potentials and all separator potentials, whose new states will be denoted ϕ_i^* and ϕ_{kl}^* . It is important to note that the conditional distribution of Equation 5.36 refers to the potentials *prior* to performing `DistributeEvidence`. The expectations in Equation 5.34 will therefore no longer be available after the tree has been made consistent and so we define

$$\lambda_{mk}(c_m \cap d_k) = \langle \hat{\psi}_k \rangle_{\mathbf{T}_\alpha \setminus c_m} \quad (5.37)$$

and store each λ_{mk} before `DistributeEvidence` is performed. Each λ_{mk} can be thought of as a pseudo-separator potential between the cluster c_m and each d_k .

Now suppose we want to update a general potential ϕ_j^* which has $B_j \neq \emptyset$. We use

$$\phi_j^{**} = \phi_j^* + \sum_{k \in B_j} \left\langle \hat{\psi}_k - \sum_{i \in C_k} \lambda_{ik} \right\rangle_{\mathbf{T}_\alpha \setminus c_j}^* - z \quad (5.38)$$

where the * above the expectation reminds us that it refers to the new $Q^*(\mathbf{T}_\alpha)$ defined by the updated potentials ϕ_i^* and ϕ_{kl}^* . The set C_k is the set of clusters that intersect with d_k , excluding c_j itself. The potentials λ_{ik} are set to zero if the cluster c_i has not yet been updated.

Following this update, the potential λ_{jk} is set to be

$$\lambda_{jk}(c_j \cap d_k) = \left\langle \hat{\psi}_k - \sum_{i \in \mathcal{C}_k} \lambda_{ik} \right\rangle_{\mathbf{T}_\alpha \setminus c_j}^* \quad (5.39)$$

As the potential for cluster c_j has been modified, we must perform `DistributeEvidence`(c_j) to regain a consistent junction tree. As we go on to update the next potential, the required expectations must be with reference to the newly calculated $Q^{**}(\mathbf{T}_\alpha)$ and so on for all remaining potentials.

It follows that $Q(\mathbf{T}_\alpha)$ can be optimised following a consistent initialisation by updating only the potentials for clusters c_j where $B_j \neq \emptyset$. It is, however, necessary to perform `DistributeEvidence` after updating each of these potentials. Depending on the graphs of P and Q , this may well provide significant saving in computation over the naive approach. A further saving can be made by noting that when updating all but the last potential, a reduced `DistributeEvidence` scheme can be used which stops once messages have been received by all clusters c_j where $B_j \neq \emptyset$.

5.5 An Algorithm for Structured Variational Inference

I will now present Structured Variational Message Passing (SVMP) – a general-purpose algorithm which allows variational inference using a structured Q distribution. To keep the explanation of SVMP as straightforward as possible, I will start by dealing only with Case I and then extend the algorithm to handle Case II as well.

5.5.1 Allowable models

In standard Variational Message Passing (Chapter 2), it was required that all conditional probabilities $P(X_k | \text{pa}_k)$ were members of the exponential family, so that natural parameter vectors (or equivalent) could be used as messages. In order to be able to use a similar form of messages for SVMP, this constraint is extended to require that all junction tree potentials can be expressed in (multivariate) exponential family form as well. The result of this restriction is that we can only retain edges of P in our Q distribution if they lie between nodes whose conditional probabilities have the same functional form. Thus, a subgraph of Q will be tractable if it contains either (1) only discrete nodes in which the conditional distribution of a node given its parents is given by a pick function over the states of the parents, or (2) it comprises Gaussian nodes each of whose mean is a multi-linear function of its parents. In fact, a subgraph of mixed discrete and Gaussian nodes will also be tractable provided there are no edges representing a Gaussian parent with a discrete child – however, I do not consider this more complex case in this analysis.

I shall therefore define our Q distribution by deleting edges of P that do not lie between two discrete or linear-Gaussian nodes. Whilst the user may choose to delete further edges to

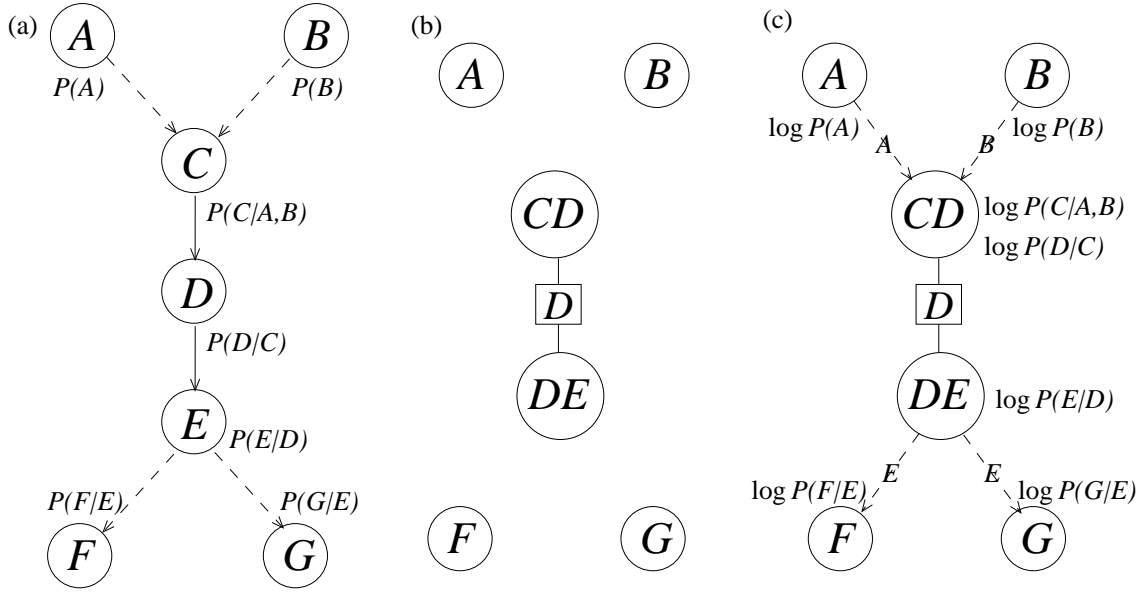


Figure 5.3: (a) The Bayesian network for P showing conditional probabilities. The dashed lines indicate which edges will be deleted in Q . (b) The set of five disjoint junction trees, four of which consist of single nodes, which are created when constructing the junction tree representation of Q . (c) The modified cluster graph for SVMP showing how the log conditionals have been assigned to clusters. The dashed arrows show where messages will be passed between the trees and what variable those messages will depend on.

improve the speed of inference at the expense of some further restriction on the form of the Q distribution, such deletions are not necessary for tractability. To help choose which further edges to delete (if any), it would be straightforward for an implementation of this algorithm to assist the user by providing guidance on the expected computation time of each iteration based on resultant clique sizes.

5.5.2 Structured Variational Message Passing algorithm

The first stage of the Structured VMP algorithm is to construct the junction tree (or set of trees) for the Q distribution. Starting with the directed graph for the Q distribution, we follow the three steps (moralisation, triangulation and construction) outlined in Section 5.3 which results in a set of one or more disjoint junction trees. For example, suppose P takes the form shown in Figure 5.3a where the dashed edges are those which will be deleted in Q . The initial step of SVMP will lead to the set of junction trees shown in Figure 5.3b. Note that all but one of these trees consist of a single node.

The second stage of SVMP involves associating each log conditional $\log P(X_k | \text{pa}_k)$ with a cluster c_j that contains both X_k and all of the parents pa_k which lie in the same tree as c_j . As Case I applies, this procedure is guaranteed to be possible. I then define A_j to be the set of log conditionals associated with cluster c_j . The only variables in each log conditional that are not in c_j must lie in other junction trees and this is marked graphically using a dashed

arrow pointing to c_j from the cluster c_i containing the log conditional for each such ‘external’ variable. The arrow is labelled with the names of the external variable or variables and this set of variables is denoted d_{ij} . These directed edges will be used to allow messages (which are functions of d_{ij}) to be passed between junction trees. The result of applying this process to our example is shown in Figure 5.3c. Note that the dashed arrows have no separators. Separators are only required when two cluster potentials are being made jointly consistent – which is not the case here as, being in different junction trees, the connected clusters have no variables in common.

The next stage is to initialise the cluster potentials to give a consistent Q distribution. This can be achieved in a simple fashion by setting each cluster potential to a suitably broad setting (e.g. uniform for discrete potentials) and then applying `CollectEvidence` and `DistributeEvidence` in each tree.

Given this consistent starting point, optimisation of the Q distribution factor for an individual junction tree \mathbf{T}_α can begin. Following the procedure described in Section 5.4.1, the separators are initialised to zero and each cluster potential c_i is initialised using

$$\phi_i(c_i) = \sum_{k \in A_i} \langle \log P(X_k | \text{pa}_k) \rangle_{\text{pa}_k \setminus c_i} + \sum_{j \in \text{ch}_i} \sum_{k \in A_j} \langle \log P(X_k | \text{pa}_k) \rangle_{c_j \setminus c_i} \quad (5.40)$$

where ch_i is the set of clusters which are connected by dashed edges pointing away from c_i (which will *not* include any clusters in \mathbf{T}_α). Thus, in Case I, we need only define an inter-tree parent-to-child message from parent cluster c_m to c_i

$$m_{c_m \rightarrow c_i} = \text{Moments}(Q(d_{mi})) \quad (5.41)$$

and an inter-tree child-to-parent message from child c_j to c_i

$$m_{c_j \rightarrow c_i} = \sum_{k \in A_j} \text{Natural}(\langle \log P(X_k | \text{pa}_k) \rangle_{Q(c_j \setminus d_{ji})}) \quad (5.42)$$

where the functionals `Moments()` and `Natural()` are as defined in Section 2.5.2. These messages allow the initialisation of each cluster potential using

$$\text{Natural}(\phi_i(c_i)) = \sum_{k \in A_i} \text{Natural}(\langle \log P(X_k | \text{pa}_k) \rangle_{Q(\text{pa}_k \setminus c_i)}) + \sum_{j \in \text{ch}_i} m_{c_j \rightarrow c_i}. \quad (5.43)$$

Following this initialisation, the optimisation of $Q(\mathbf{T}_\alpha)$ can be completed by performing `CollectEvidence` and `DistributeEvidence`.

To illustrate this process, consider initialising the centre tree in the graph of Figure 5.3c. For cluster DE , the incoming messages from F and G are

$$m_{F \rightarrow DE} = \text{Natural}(\langle \log P(F | E) \rangle_F) \quad (5.44)$$

$$m_{G \rightarrow DE} = \text{Natural}(\langle \log P(G | E) \rangle_G) \quad (5.45)$$

and so ϕ_{DE} is initialised to

$$\text{Natural}(\phi_{DE}) = \text{Natural}(\log P(E | D)) + m_{F \rightarrow DE} + m_{G \rightarrow DE}. \quad (5.46)$$

For cluster CD , the incoming messages are:

$$m_{A \rightarrow CD} = \text{Moments}(Q(A)) \quad (5.47)$$

$$m_{B \rightarrow CD} = \text{Moments}(Q(B)) \quad (5.48)$$

and therefore ϕ_{CD} is initialised to

$$\text{Natural}(\phi_{CD}) = \text{Natural}(\log P(D | C)) + \text{Natural}(\langle \log P(C | A, B) \rangle_{Q(A)Q(B)}). \quad (5.49)$$

Applying `CollectEvidence` and `DistributeEvidence` will result in the optimal distribution for $Q(C, D, E)$, given that all other factors of Q are held fixed. A summary of the entire SVMP algorithm is given in Algorithm 5.1.

Algorithm 5.1 The Structured Variational Message Passing (SVMP) Algorithm

1. Moralise the graph of Q , triangulate and construct junction tree(s).
 2. Associate each log conditional $\log P(X_k | \text{pa}_k)$ with a cluster c_j in a tree \mathbf{T}_α that contains $\mathbf{T}_\alpha \cap (X_k \cup \text{pa}_k)$ and create dashed edges to clusters containing variables in $(X_k \cup \text{pa}_k) \setminus c_j$.
 3. Initialise each junction tree to a consistent distribution.
 4. For each junction tree in turn:
 - (a) Initialise all separator potentials to zero;
 - (b) For each cluster, set its potential using Equation 5.43 and messages received from all parent and children clusters (which will lie in other junction trees);
 - (c) Perform `CollectEvidence` and `DistributeEvidence` on the tree.
 5. Repeat from step 4.
-

The SVMP algorithm is guaranteed to optimise one factor $Q(\mathbf{T}_\alpha)$ given the remaining factors, except in the (avoidable) circumstances which will now be described. Consider the graph shown in Figure 5.4a and the corresponding Q junction trees of Figure 5.4b. In updating the factor $Q(Y)$ we need to compute the expectation of $P(Y | A, E)$ with respect to the variational posterior distribution $Q(A, E)$. As it stands, this will be represented by a product of marginals $Q(A)Q(E)$ given by the two message to Y from A and E . We can extend the formalism to capture correctly the correlation between A and E by adding an edge connecting these two nodes, thereby ensuring that they are in the same cluster of the junction tree. This can be achieved in general by moralising nodes such as Y before removing the edges.

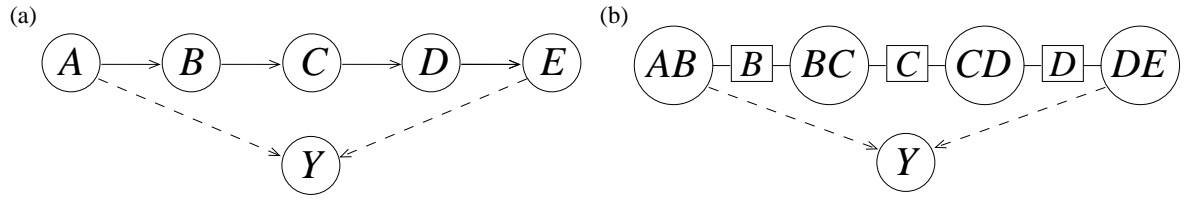


Figure 5.4: (a) Example graph showing the need for additional moralisation in order to find optimal variational marginals. The Q distribution structure for this example is defined by the subgraph comprising the Markov chain at the top, with the dashed links removed. (b) The junction trees of the Q distribution, with dashed arrows showing where messages will pass to (and from) node Y .

5.5.3 Extending the algorithm to allow internal deleted edges (Case II)

As stated earlier, Case II is the situation where there is (at least) one pair of variables in a junction tree \mathbf{T}_α of Q that are connected by an edge in P , but do not lie in the same cluster in Q (for example nodes C and E in Figure 5.5a). It follows that there is at least one conditional distribution $P(X_j | \text{pa}_j)$ where the intersection $\mathbf{T}_\alpha \cap (X_j \cup \text{pa}_j)$ cannot be found in any cluster of \mathbf{T}_α .

The existing Algorithm 5.1 will fail at step 2 when it tries to associate these conditionals with clusters, because no cluster will contain the required variables. We can extend the algorithm to handle these ‘uncontained’ conditionals $\{\hat{\psi}_j\}$, by creating a new cluster Ω_j

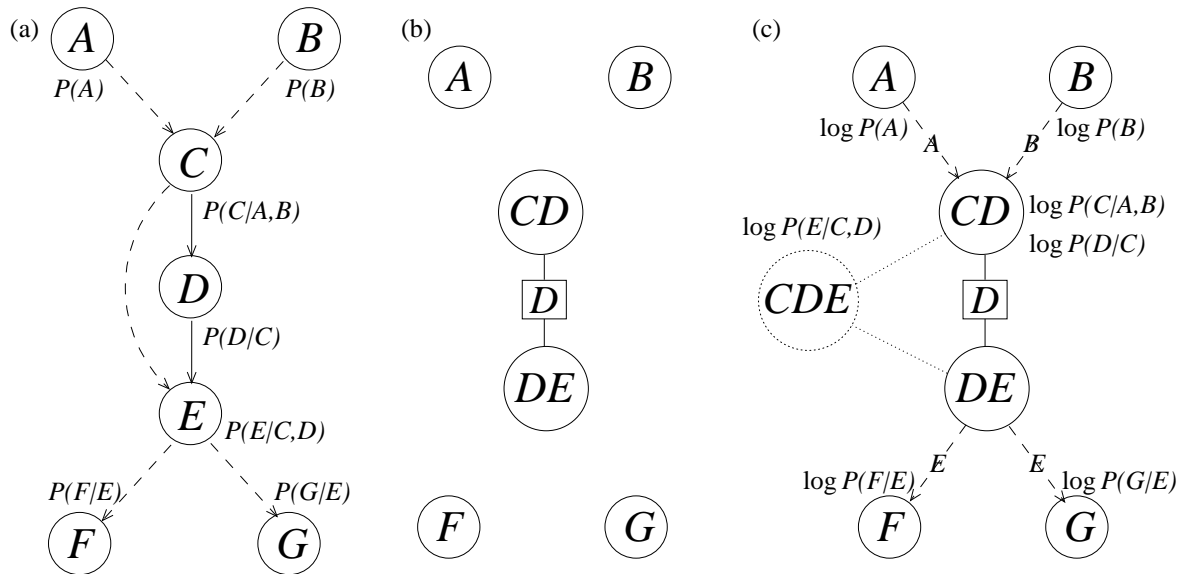


Figure 5.5: (a) The Bayesian network for P where dashed lines indicate which edges will be deleted in Q . Note that the addition of dashed edge between C and E , compared to the model of 5.3a. (b) The set of junction trees of Q . These are identical to 5.3b as the structure of Q is unchanged. (c) The modified cluster graph for SVMP showing how a P -cluster (shown dotted) has been created for the conditional $P(E | C, D)$ which is not contained in any cluster of Q . This P -cluster is connected to clusters of Q with which it has any variables in common.

containing the variables $\mathbf{T}_\alpha \cap (X_j \cup \text{pa}_j)$ for each such conditional (see for example, the *CDE* cluster shown dotted in Figure 5.5c). These new clusters will be denoted *P*-clusters (as they correspond to factors of *P*). Each *P*-cluster has a potential ω_j and is connected to all clusters in \mathbf{T}_α which have any variables in common with the *P*-cluster. We define B_i to be the set of *P*-clusters connected to a standard cluster c_i .

The algorithm then proceeds as before, ignoring the *P*-clusters and their corresponding conditionals until after the `CollectEvidence` and `DistributeEvidence` procedures at the end of step 4. At this point, the extended algorithm applies the changes required to take into account the effect of the ignored conditionals. Firstly, each ω_j potential is initialised to the corresponding $\hat{\psi}_j$ potential. Secondly, we take each standard cluster c_i which is connected to at least one *P*-cluster (so $B_i \neq \emptyset$) and update its potential using,

$$\phi_i^* = \phi_i + \sum_{j \in B_i} \langle \omega_j \rangle_{\Omega_j \setminus c_i} - z, \quad (5.50)$$

and immediately modify each connected *P*-cluster potential using

$$\omega_j \leftarrow \omega_j - \langle \omega_j \rangle_{\Omega_j \setminus c_i}. \quad (5.51)$$

This can be achieved using message passing by sending a message from the *P*-cluster Ω_j to the cluster being updated c_i of the form

$$m_{\Omega_j \rightarrow c_i} = \text{Natural}(\langle \omega_j \rangle_{\Omega_j \setminus c_i}) \quad (5.52)$$

and subtracting it locally from ω_j . This message can be computed in the *P*-cluster Ω_j only if it has first received messages from all other clusters $c_k, k \neq i$ of the form

$$m_{c_k \rightarrow \Omega_j} = \text{Moments}(Q(c_k \cap \Omega_j)). \quad (5.53)$$

The cluster potential is then updated using

$$\text{Natural}(\phi_i^*) = \text{Natural}(\phi_i) + \sum_{j \in B_i} m_{\Omega_j \rightarrow c_i}. \quad (5.54)$$

Following this update, the variational junction tree will no longer be consistent and so `DistributeEvidence`(c_i) must be performed. This update procedure is then repeated for all other clusters where $B_i \neq \emptyset$. Algorithm 5.2 on the next page gives a summary of this extended SVMP algorithm.

Algorithm 5.2 Extended SVMP Algorithm

1. Moralise the graph of Q , triangulate and construct junction tree(s).
 2. Associate each log conditional $\log P(X_k | \text{pa}_k)$ with a cluster c_j in a tree \mathbf{T}_α that contains $\mathbf{T}_\alpha \cap (X_k \cup \text{pa}_k)$ and create dashed edges to clusters containing variables in $(X_k \cup \text{pa}_k) \setminus c_j$. Where a conditional cannot be associated a cluster, create a new P -cluster and connect it to all clusters with any variables in common.
 3. Initialise each junction tree to a consistent distribution.
 4. For each junction tree in turn:
 - (a) Initialise all separator potentials to zero;
 - (b) For each cluster, set its potential using Equation 5.43 and messages received from all parent and children clusters (which will lie in other junction trees);
 - (c) Perform `CollectEvidence` and `DistributeEvidence` on the tree;
 - (d) Initialise each ω_j to $\hat{\psi}_j$;
 - (e) For each cluster c_i where $B_i \neq \emptyset$, update its potential and all connected ω_j potentials using Equations 5.52–5.54. Then perform `DistributeEvidence`(c_i).
 5. Repeat from step 4.
-

5.6 Structured VIBES: A Partial Implementation of SVMP

A full implementation of the Structured Variational Message Passing algorithm would be a significant undertaking. Instead, the VIBES software described in Section 2.3 has been extended to implement a partial form of SVMP with the following constraints:

- retained edges in Q had to be between discrete nodes (leading to discrete multivariate potentials). Nodes with no connections in Q could have any distribution supported by standard VIBES;
- the retained structure of Q could not have cycles (i.e. had to be a tree), thus removing the need for moralisation and triangulation steps;
- edges between nodes connected by another path in Q could not be deleted (so only Case I was allowed).

5.6.1 Example: Hidden Markov Model

The extended VIBES software will be illustrated using a Bayesian hidden Markov model in which prior distributions are defined over the probabilities for the initial state of the hidden variables as well as over the transition and emission matrices. This model was described, and also solved variationally, in MacKay [1997]. In order to highlight the comparison against the structured framework we have allowed all of the variables to be unobserved. The screen shot

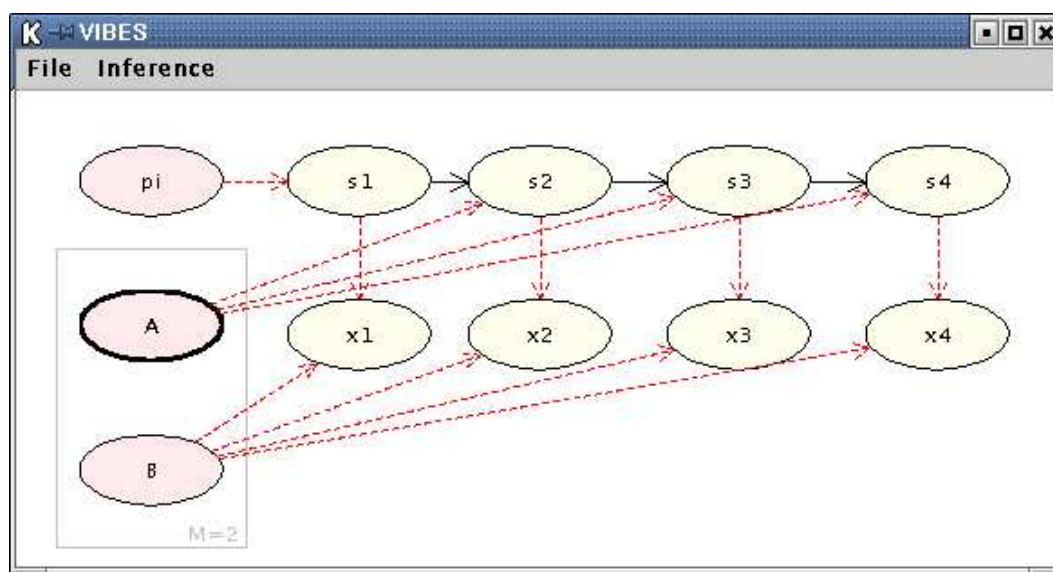


Figure 5.6: VIBES screenshot showing the graphical model defining the distribution for a Bayesian hidden Markov model. Links shown in dashed red are those that will be removed in defining the structured Q distribution, while those shown in black will remain.

from VIBES for the directed graph defining the $P(X)$ distribution is shown in Figure 5.6. As a point of comparison I first solve this model using the fully factorised variational approximation using standard VIBES. Next I apply a structured variational approximation as shown in Figure 5.7 using SVMP. Here the links along the hidden Markov chain are retained, leading to a more flexible class of Q distributions. On this tiny model, the converged value of the lower bound \mathcal{L} for the structured distribution of Figure 5.7 is 3.873 nats compared to 3.631

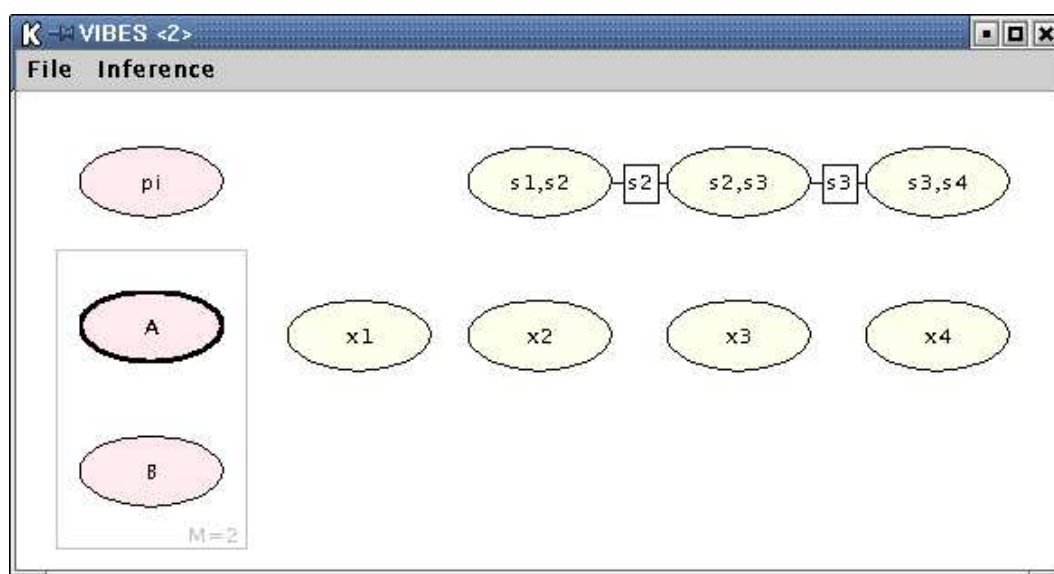


Figure 5.7: VIBES screenshot of the HMM structured variational distribution, showing the junction tree.

nats for the fully factorised distribution, showing that the use of a structured Q distribution leads to an improved approximation.

5.7 Discussion

I have demonstrated an algorithm, Structured Variational Message Passing, which allows variational inference to be performed automatically using a Q distribution that retains some of the structure of the original P distribution. In addition, a partial implementation of this algorithm has been shown to give an improved approximation over standard VMP for a Hidden Markov model. As belief propagation is a special case of SVMP, a full implementation of SVMP would be a powerful tool, able to:

- perform exact inference (on pure discrete or linear Gaussian graphs) where clique sizes would make this practicable;
- perform approximate inference on all other conjugate-exponential graphs (including mixed continuous and discrete graphs) whilst allowing the approximating Q distribution to retain some of the structure of the P distribution;
- allow the accuracy of the approximation to be adjusted downwards to reduce computation time, until the limiting case of a fully-factorised Q is reached.

It is also possible to consider Q distributions represented by a graph that is not a sub-graph of the original P distribution. In fact, my assumption that the Q distribution is created by deleting edges of P is not required by SVMP, which allows any structure to be used in the Q distribution, provided that all resultant Q potentials are tractable.

BIBLIOGRAPHY

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

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

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

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

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

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

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

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