Lecture 11:

Introduction to Probabilistic Graphical Models

Graphical Models

So far we have seen examples of two different types of graphical model representing the same inference problem:

Bayesian Networks:

These have the advantage of displaying causal relationships but may not converge correctly when multiply connected.

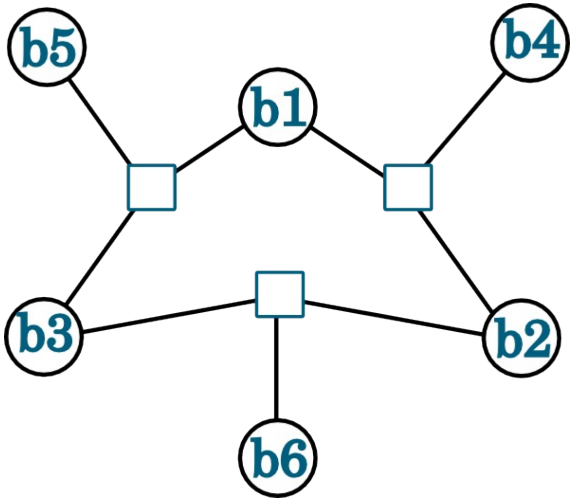
Join Trees:

These can always solve the problem (given sufficient time), but don't contain causal information.

The Tanner graph

The Tanner graph is a graphical model that was introduced as a model for error recovery in parity checking.

It is a bi-partite graph - ie it has two node types and each node connects only to its opposite type.



The circles represent bits and the squares represent parity checks. The squares evaluate to 1 if the sum of the bits is even.

Parity checking as an inference problem

Suppose the bits are transmitted down a noisy channel and let P_f be the probability that a bit is flipped during transmission.

Let Yi be the measured (received) bit values.

Let Xi be the true (transmitted) bit values which we want to estimate, then:

$$\begin{aligned} \mathrm{P}(X_i|Y_i) &= 1- P_f \quad \text{if } X_i = Y_i \\ &= P_f \quad \text{otherwise.} \end{aligned}$$

Parity checking as an inference problem

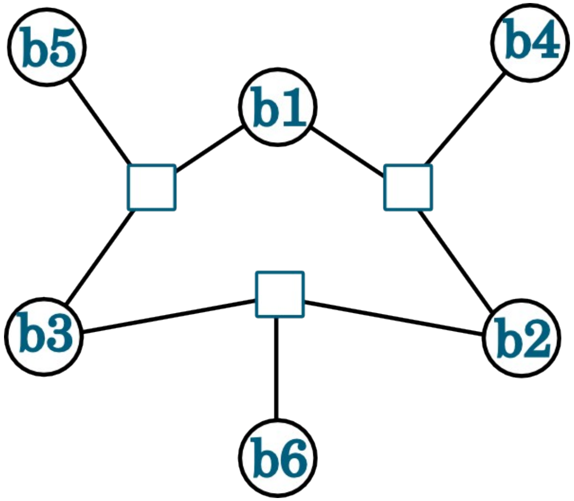
So given a possible bit string (X_1, X_2, X_3, ... X_n) we can calculate a probability of it being correct using:

$$P(X_1, X_2, ..X_N) = \prod_{j=1}^{N} P(X_j|Y_j)$$

If we did not have any parity constraints then the most probable bit string is the one for which $X_i = Y_i$ for all the bits - ie the string we received.

Parity checking as an inference problem

Now suppose that bits 1,2 and 3 are data bits and 4, 5 and 6 are parity bits.



The parity bits are set so that sum of the three bits in each group is even.

We check this by evaluating a constraint function for each square eg:

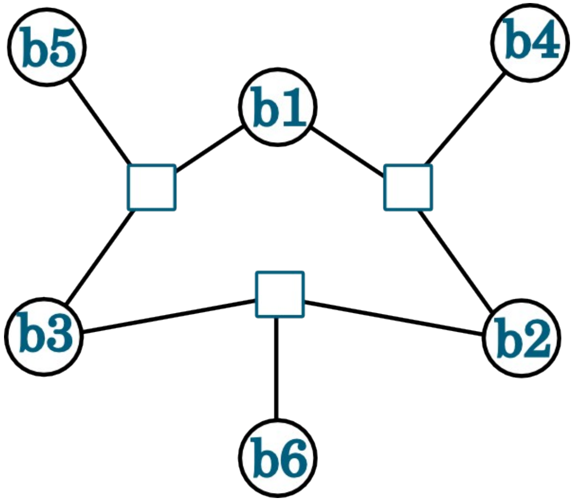
$$\Psi(X_1, X_3, X_5) = 1 - (X_1 + X_3 + X_5)\ mod\ 2$$

Parity checking as an inference problem

We can re-write our joint probability distribution as:

$$P(X_1, X_2, ..X_6) = \Psi(X_1, X_5, X_3)\Psi(X_1, X_2, X_4)\Psi(X_2, X_3, X_6)\prod_{j=1}^{N} P(X_j|Y_j)$$

The parity constraints ensure that any bit string failing a check receives a zero probability.



In the event that the received bit string fails a parity check the next most probable bit string is selected.

Factor Graphs

The factor graph is the most general graphical model to express probabilistic inference problems. It is a bi-partite graph representing a factorisation:

$$g(X_1, X_2, X_3, ...X_n) = \prod_{j=1}^{m} f_j(S_j)$$

The nodes are either variables (X_1, X_2, X_3, . . . X_n)

or factors (f_1, f_2, f_3, . . . f_n)

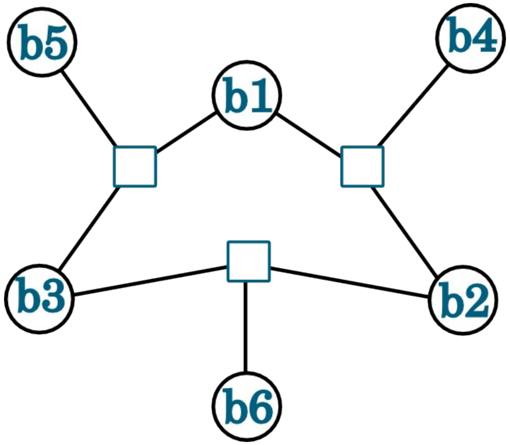
In probabilistic inference the function is usually a joint probability distribution.

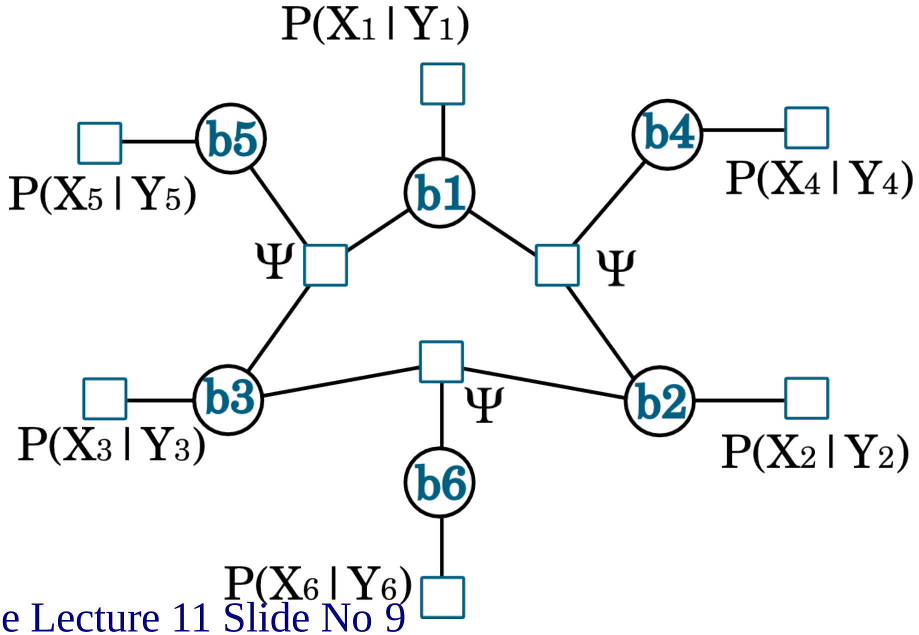
The Tanner graph as a factor graph

The Tanner graph represents a factorisation of a joint probability distribution.

$$P(X_1, X_2, ..X_6) = \Psi(X_1, X_5, X_6)\Psi(X_1, X_2, X_4)\Psi(X_2, X_3, X_6)\prod_{j=1}^{N} P(X_j|Y_j)$$

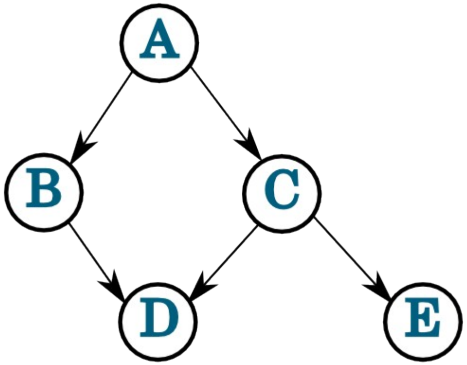
The factor graph is almost identical but includes factors of the form P(Xi|Yi).





Bayesian Nets as factorisation

Bayesian nets can be looked on as the factorisation of a joint probability distribution.



For example, for the metastatic cancer net:

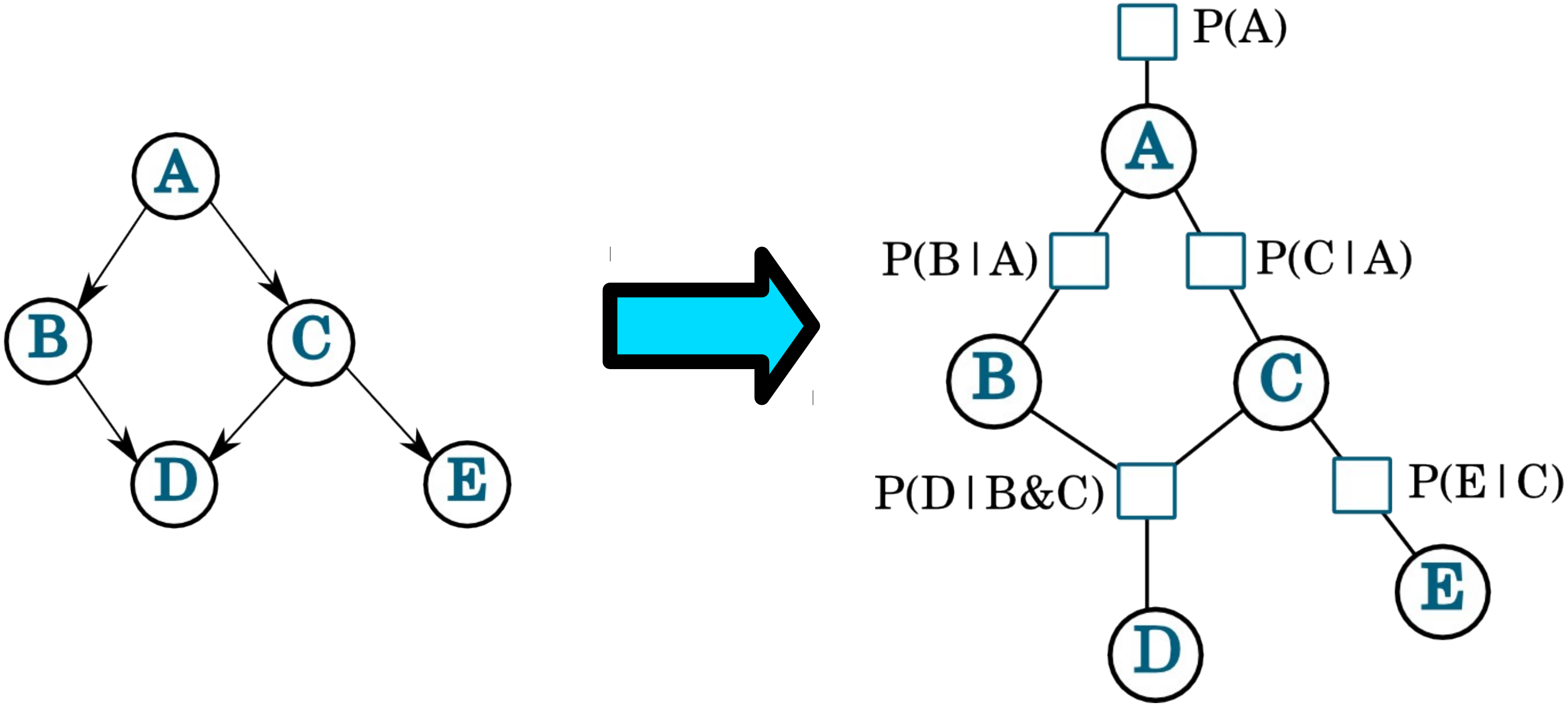
$$P(A,B,C,D,E) = P(A)P(B|A)P(C|A)P(D|B\&C)P(E|C)$$

or in general

$$P(X_1, X_2, X_3, ...X_n) = \prod_{i=1}^{n} P(X_i|Parents(X_i))$$

Bayesian Net factor graphs

Every Bayesian net can be represented as a factor graph. The factors are simply the conditional probability matrices. For the metastatic cancer we have:



Arguments in favour of the factor graph model

The factor graph model gives a uniform treatment to message passing.

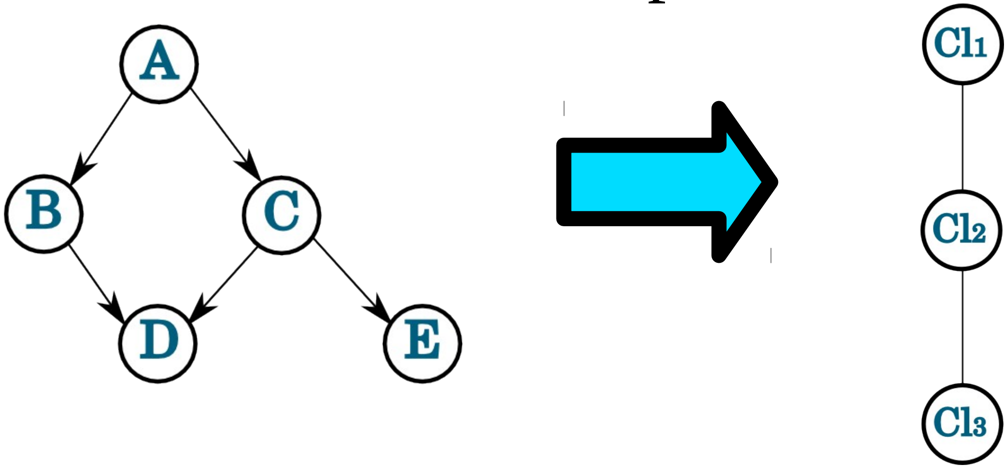
The message that a node sends to a neighbour is made up of the messages that it receives from all other neighbours.

The factors (conditional probability matrices) transform the messages into evidence for the receiving node.

Join (or Junction) Trees and factorisation

The join tree represents a different factorisation of the joint probability distribution of a Bayesian network. It is the product of potential functions.

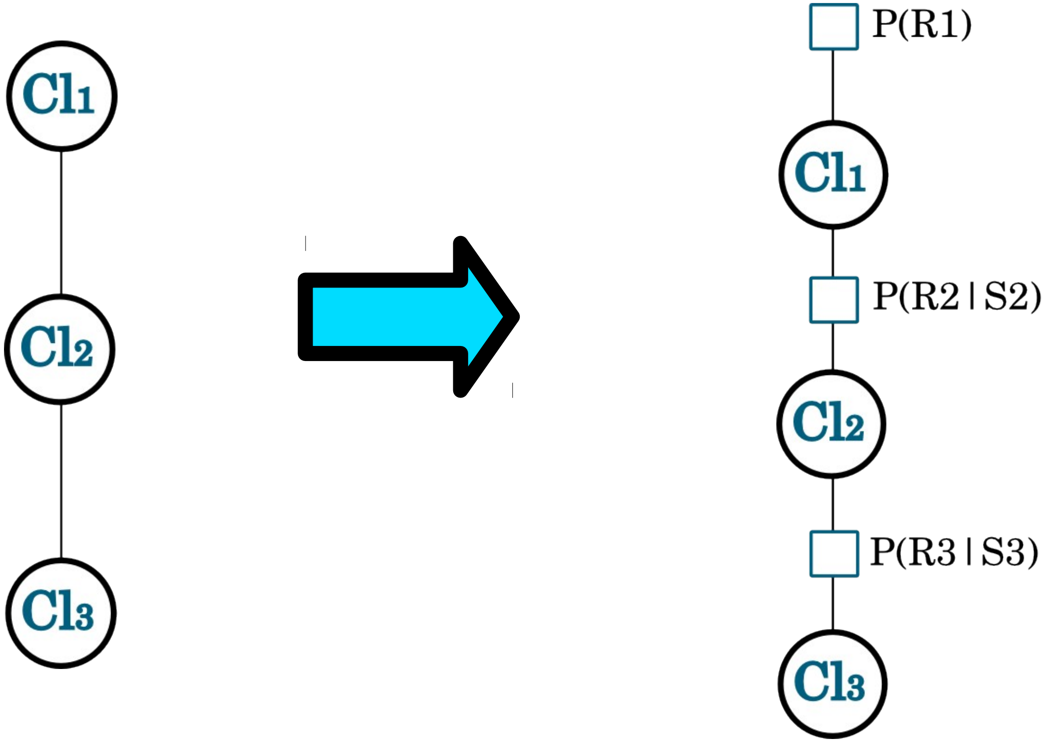
For the metastatic cancer example we have:



$$P(A,B,C,D,E) = \Psi(C_1)\Psi(C_2)\Psi(C_3) = \Psi(A,B,C)\Psi(B,C,D)\Psi(C,E)$$

Join Trees and factorisation

The potential functions in a join tree are of the form P(R|S). The factor graph representation is:



$$P(A,B,C,D,E) = \Psi(C_1)\Psi(C_2)\Psi(C_3) = \Psi(A,B,C)\Psi(B,C,D)\Psi(C,E)$$

Markov Random Fields

Join trees are an example of a graphical structure called Markov Random Fields. They are so-called since each node depends only on its immediate neighbours.

They are expressed as a factorisation into subsets Vj each with a potential function:

$$P(X_1, X_2, X_3, \ldots X_n) = \frac{1}{Z} \prod_{j=1}^{m} \Psi_j(V_j)$$

Where Z is a normalisation constant

$$Z = \sum_{X} \prod_{j=1}^{m} \Psi_j(V_j)$$

Normalisation in Markov Random Fields

The normalisation constant Z gives us flexibility in the definition the factorisation in a Markov Random Field.

However this comes at a computational cost, as for large inference problems Z is difficult to compute.

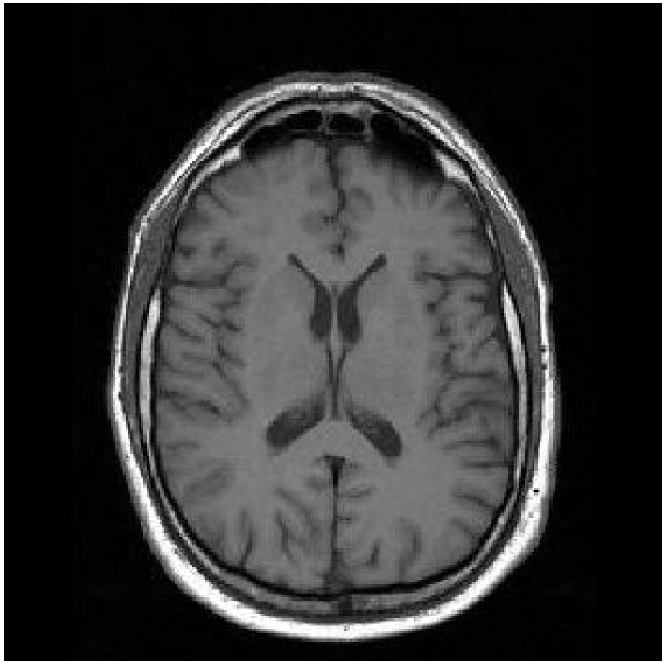
In the Join tree algorithm we carefully ensured that Z=1 whenever we updated the potential tables.

Pairwise Markov Random Fields

A pairwise Markov random field has all its variables joined in cliques of size 2, thus the corresponding factor graph has factors that join at most two variables.

Join trees are an example of pairwise Markov Random fields, but a more useful example is the grid structured network used in image processing.

Example - medical image segmentation/restoration

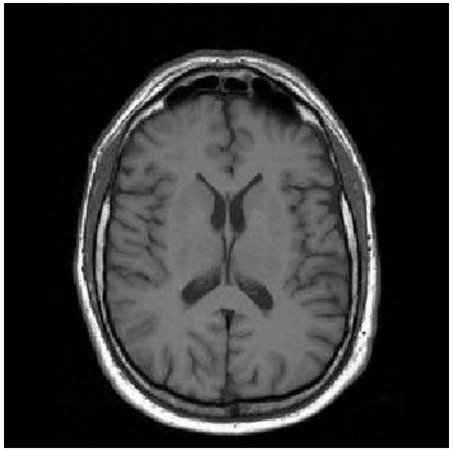


Pairwise Markov Random Fields have been used successfully in image processing for medical diagnosis:

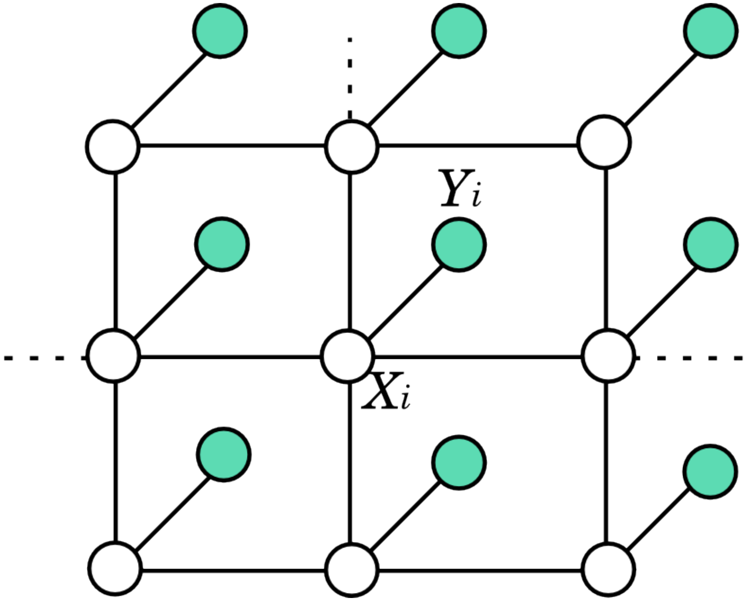
Segmentation - determine which class pixels belong to: Grey Matter (cortical neurones) White Matter (connective tissue) or Cerebral Spinal Fluid.

Restoration - correct pixels that are wrong due to imaging errors and artefacts.

Image Segmentation/Restoration using MRF



Each pixel in the image is a variable in the inference problem. The filled circles represent actual pixel values and are labelled Yi.



The model (segments or restored image values) is represented by the empty circles Xi.

The Xi values are calculated using the pixel measurements and the neighbour's messages.

Image Segmentation as Probabilistic Inference

We need to define two compatibility functions:

Φ(Xi,Yi) - relates the observed and hidden values and is rather like a conditional probability P(Xi|Yi). It expresses the probability of the pixel belonging to a particular class (WM, GM, CSF) give the measured pixel value Yi.

Ψ(Xi,Xj) - expresses the compatibility between adjacent pixels. Any pixels not connected will have a Y(Xi,Xj) value of 1 expressing no information. For connected pixels this compatibility function is like a joint probability of the adjacent states being neighbours.

Image segmentation as Probabilistic Inference

Given an image Y=(Y_1, Y_2, . . Y_n)

And a segmentation X = (X_1, X_2, . . X_n)

We can define a joint probability distribution:

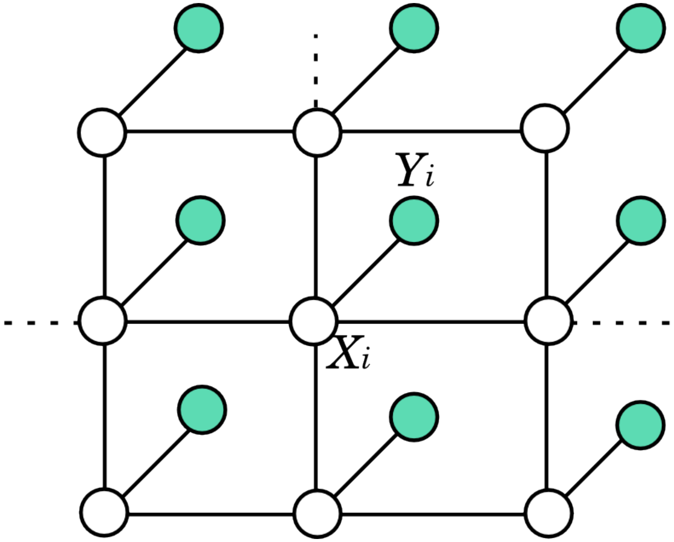
$$P(X,Y) = \frac{1}{Z} \prod_{i,j} \Psi(X_i, X_j) \prod_{i} \Phi(X_i, Y_i)$$

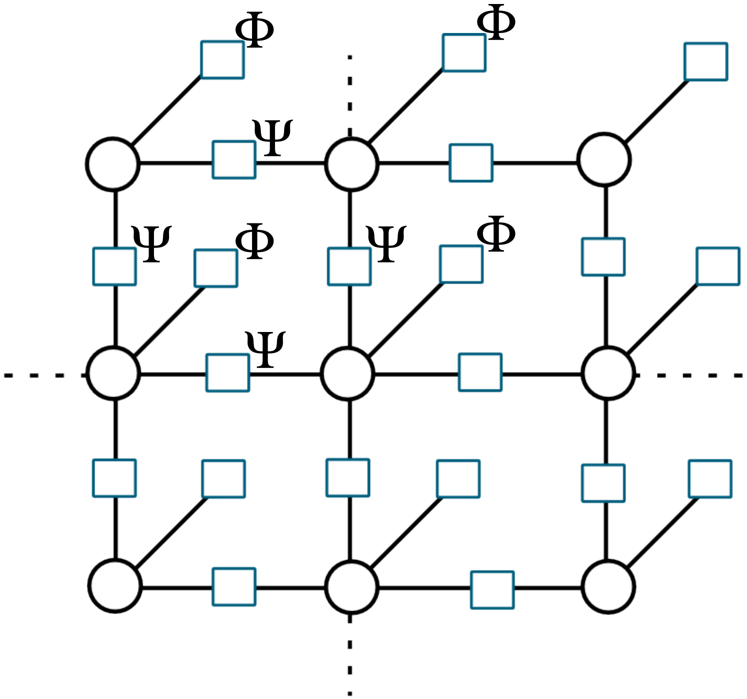
and find the values of Xi that give the maximum probability (ie the most likely segmentation)

Note that the high cost of computing Z makes this direct approach computationally infeasible.

Factor graph of the Pairwise MRF

The pairwise Markov random field, being just a factorisation of a probability distribution can be represented as a factor graph.





Belief Propagation

Belief propagation overcomes the computational difficulties of using a global joint probability distribution by making local computations.

We have already discussed belief propagation in Bayesian networks which is done using λ and π messages.

The belief in a variable is just its posterior probability distribution.

Belief propagation in MRFs

Belief propagation in Markov Random fields is simpler than in Bayesian Networks, because there is no notion of causality, and so all messages are of the same form (similar to the π message). We write the belief in a variable as:

$$b(X_i(s)) = \frac{1}{Z}\Phi(X_i(s), Y_i) \prod_{k \epsilon N(i)} m_k(X_i(s))$$

Where $X_i(s)$ means state s of node X_i and m_k means a message (or evidence) from a neighbour.

Belief propagation in MRFs

We write the un-normalised belief in a variable excluding the evidence from its neighbour *j* as:

$$b_{\setminus j}(X_i(s)) = \Phi(X_i(s), Y_i) \prod_{k \epsilon N(i) \setminus j} m_k(X_i(s))$$

Where \j means “excluding node j”. Notice the similarity of this equation to the definition of the π message.

Belief propagation in MRFs

Finally we can define the messages (equivalent to the π evidence) as:

$$\mathbf{m_i}(X_j) = \mathbf{b}_{\backslash\mathbf{j}}(X_i)\ \Psi(X_i, X_j)$$

Where

$\mathbf{m_i}(X_j)$ is a vector message for the states of X_j,

$\mathbf{b}_{\backslash\mathbf{j}}(X_i)$ is a vector of the beliefs in the states of X_i excluding the evidence from X_j and

$\Psi(X_i,X_j)$ is the compatibility matrix.

Loopy Propagation

In the case of Bayesian Networks we noted that loops in the network mean that the propagation would never terminate.

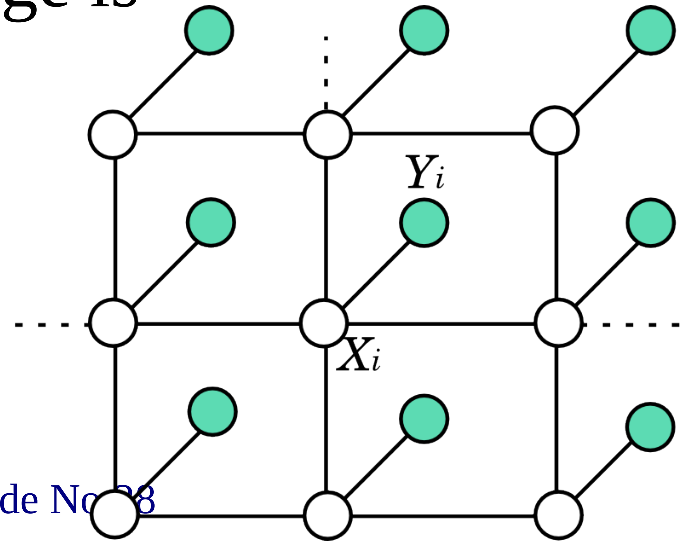
However, we could always find a correct probability distribution for any inference problem using the join tree method.

In fact Loopy propagation is an optimisation problem, and will normally converge though not necessarily to the optimum.

Loopy Propagation in Markov random fields

The pairwise MRFs used in computer vision usually work well because the optimisation is strongly constrained by the evidence variables (ie the pixel values from the image).

Hence belief propagation occurs as a series of epochs in which each pixel of the image is updated once. After a few epochs the segmentation will converge.



Summary on Graphical Models

In probabilistic inference graphical models express local relationships that allow us to carry out belief propagation in a reasonable time frame.

Without using a graphical model an inference may be expressed as a sum over the joint distribution:

$$P(X_n) = \sum_{X_1}\sum_{X_2}..\sum_{X_{n-1}} P(X_1, X_2, X_3...X_N)$$

but this is computationally infeasible for even moderate size problems.

Summary on Graphical Models

Bayesian Networks express conditional independence relationships and causality, and for the singly connected case (or specific instantiations in multiply connected networks) offer fast computation.

Join trees are always singly connected, but have larger factors and so become computationally infeasible if the number of cliques is small.

Pairwise MRFs, with multiple loops, can be solved effectively by using an iterative approach.

Summary on Graphical Models

Belief propagation in graphical models with loops is an optimisation problem.

Curiously the same equations are found to govern other physical properties (magnetism) where their solution is done by techniques of free energy minimisation.

Summary on Graphical Models

Graphical models occur in other fields as well, where they perform a similar role in expressing local properties. Examples are:

- Genetic regulatory metworks
- Neural Circuitry (spiking neurones)
- Sensor Networks

There is currently much active research in trying to relate the capabilities of theses different graphical models for making inferences.