# CALD 10–702
# Statistical Approaches to Learning and Discovery

*Assignment 4*

April 14, 2003

*Due in class on Monday, April 23.*

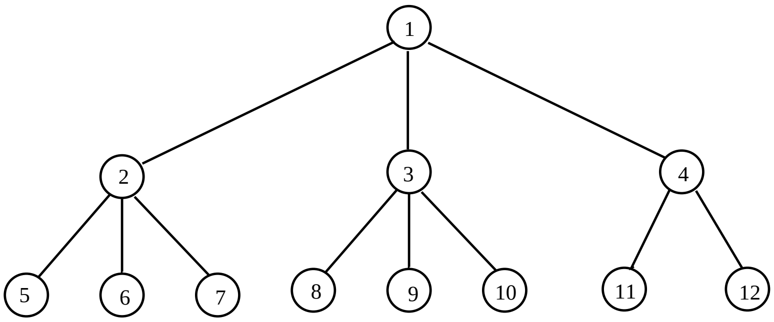
**Instructions:** This assignment includes three problems, involving belief propagation, discriminative vs. generative models, and learning with labeled and unlabeled data. Please submit written answers to each question. For the problem involving coding, you email your Matlab source code to Tianjiao Chu (`tchu@andrew`).

**Problem 1**. *Belief Propagation*

In this problem you will prove the correctness of belief propagation on trees. The strategy will be to first define an algorithm that is apparently more complex as it works with entire subtrees, and to then show how the computations can be carried out recursively, deriving the message passing updates of the sum-product algorithm.

Consider a graphical model $p(z_1, z_2, \ldots, z_n)$ where the underlying graph is an undirected tree, as shown in Figure 1. We assume that each node is represented as $z_i = (x_i, y_i)$ where $x_i$ is unobserved, and $y_i$ is observed. Given evidence $\boldsymbol{y} = (y_1, \ldots, y_n)$, our task is to compute the marginal probabilities $p(x_i \,|\, \boldsymbol{y})$ for each $i = 1, 2, \ldots, n$.

It will be convenient to use a new piece of notation. If $(i, j)$ is an edge in the tree, let $ij{\downarrow}$ denote the set of nodes $k$ such that there is a path from $i$ to $k$ passing through $j$. For example, in the tree below we have that $12{\downarrow} = \{5, 6, 7\}$, $38{\downarrow} = \emptyset$, and $14{\downarrow} = \{11, 12\}$.




To begin, establish the following elementary facts:

1.1 In case $j \neq k$ then $ij{\downarrow} \cap ik{\downarrow} = \emptyset$.

1.2 For every node $k$, there is a node $i$ and a node $j \in \mathcal{N}(i)$ (one of the neighbors of $i$), so that $x_k \in ij\!\downarrow$.

Now, let $\Phi_{ij\downarrow}(x_{ij\downarrow})$ denote the product of all clique potentials involving the nodes in $ij\!\downarrow$. Thus,

$$\Phi_{ij\downarrow}(x_{ij\downarrow}) = \prod_{k,l \in ij\downarrow} \Psi_{kl}(x_k, x_l) \prod_{k \in ij\downarrow} \Psi_k(x_k, y_k) \tag{1}$$

Next, define the "message" $m_{ji}(x_i)$ sent from $j$ to $i$ as

$$m_{ji}(x_i) = \sum_{x_{ij\downarrow}} \Psi_{ij}(x_i, x_j)\, \Phi_{ij\downarrow}(x_{ij\downarrow}) \tag{2}$$

1.3 Using the definition of $m_{ji}$, show that

$$p(x_i \,|\, \boldsymbol{y}) \;\propto\; \Psi_i(x_i, y_i) \prod_{j \in \mathcal{N}(i)} m_{ji}(x_i) \tag{3}$$

1.4 Now show that the messages $m_{ji}$ satisfy the recursion

$$m_{ji}(x_i) = \sum_{x_j} \Psi_{ij}(x_i, x_j)\, \Psi_j(x_j, y_j) \prod_{k \in \mathcal{N}(j) \backslash i} m_{kj}(x_j) \tag{4}$$

Together, these steps show the correctness of the sum-product algorithm on trees.

**Problem 2**. *Comparing Belief Propagation to Gibbs Sampling*

In this problem[1] you will write code to implement two approximate inference strategies, one using loopy belief propagation, the other using Gibbs sampling. Each of these will be compared to exact inference.

The underlying graph will be a "toroidal grid," that is, a grid that is wrapped around so that all nodes have the same degree. Both the unobserved variable $x_i$ and the observed variable $y_i$ at each node $i$ will be binary. The graph can be represented by an adjacency matrix; for example, the $3 \times 3$ grid has 9 nodes and adjacency matrix

$$\begin{pmatrix} 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 \end{pmatrix} \tag{5}$$

[1] Due to Yair Weiss.

The graphical model on the $k \times k$ toroidal grid is defined in terms of a $2 \times 2$ matrix for each edge $(i, j)$, representing the potential $\Psi_{ij}$, and a $2 \times N$ matrix of local evidence potentials $\Psi_i$, where $N = k^2$. The potentials $\Psi_{ij}$ are defined in the following canonical form:

$$\psi_{ij} = \begin{pmatrix} e^{\frac{\lambda_{ij}}{2}} & e^{-\frac{\lambda_{ij}}{2}} \\ e^{-\frac{\lambda_{ij}}{2}} & e^{\frac{\lambda_{ij}}{2}} \end{pmatrix} \tag{6}$$

where $\Lambda = [\lambda_{ij}]$ is a given $N \times N$ matrix of weights.

The task is to compue the marginal conditional probabilities $p(x_i \mid \boldsymbol{y})$. To do this, you should implement the loopy belief propagation and Gibbs sampling algorithms in Matlab as follows.

1. Gibbs sampling is to be implemented as a function

   ```
   [beliefs] = Gibbs(grid, lambda, local, startX, burn, interval, S)
   ```

   where `grid` is the $N \times N$ adjacency matrix of the grid, `lambda` is the $N \times N$ matrix of $\lambda_{ij}$ values, which defines the edge potentials according to equation (6), and `local` is the $2 \times N$ matrix of evidence potentials.

   You should implement Gibbs sampling by starting with $x_i$ as specified by the vector `startX`, and then sequentially going over the variables and sampling each one, thus generating a new vector $\boldsymbol{x}$. This is repeated for `burn` iterations, after which a single sample is chosen each `interval` iterations. A total of `S` samples is generated in this way to estimate the marginal probabilities, which are returned as the $2 \times N$ matrix `beliefs`.

2. Loopy BP is to be implemented as a function

   ```
   [beliefs] = Loopy(grid, lambda, local, strategy)
   ```

   where `grid`, `lambda`, and `local` are as above. If `strategy` $= 1$, then *sequential updating* should be used, where the beliefs are updated one at a time, and the new beliefs are used for subsequent updates. Otherwise, if `strategy` $= 2$ then *parallel updating* should be used, where all of the beliefs are updated using the current beliefs.

To investigate the performance of these two approximate inference strategies, you should carry out the following experiments, using the appropriate Matlab files in the directory `http://www.cs.cmu.edu/~tom/10-702/assignment4`.

2.1 Compare Gibbs and loopy BP to exact inference for the $3 \times 3$ grid using the "easy" setting of potentials with $\lambda_{ij} = 0.05$ and the evidence matrix `3x3local.mat`. The exact inference results are contained in the matrix `3x3easy.mat`. To judge the algorithms, use the average and maximum Kullback-Leibler divergence of the marginal distributions $p(x_i \mid \boldsymbol{y})$ for the correct and estimated beliefs. Summarize the results and give appropriate plots.

2.2 Repeat the above procedure for the "medium" problem where $\lambda_{ij} = 2$ and the "hard" problem where the $\lambda$s are given in the matrix `3x3lambda.mat`.

2.3 Carry out the equivalent calculations for the $5 \times 5$ and $7 \times 7$ grids.

In addition to handing in a write-up of your results on the above computations, you should pass in your two Matlab functions `Gibbs.m` and `Loopy.m`. It is important that your implementation matches the above specification.

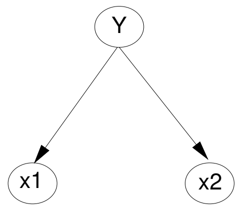
**Problem 3**. *Learning from Labeled and Unlabeled Data*

This question asks you to examine the use of EM and graphical models to learn from labeled and unlabeled data. We are particularly interested in the question of when unlabeled data can help to learn $P(y|x)$ more accurately.

Consider a setting in which we have three random Boolean variables: $y, x_1$ and $x_2$. We are interested in learning $P(y|x_1, x_2)$. We are given a set of training data drawn from $P(y, x_1, x_2)$ where the data can be partitioned into three subsets: $L, U_y$, and $U_{x2}$, as follows:

- $L$ = examples of the form $\langle y, x_1, x_2 \rangle$ (fully labeled data)
- $U_y$ = examples of the form $\langle x_1, x_2 \rangle$ ($y$ labels are unobserved)
- $U_{x2}$ = examples of the form $\langle y, x_1 \rangle$ ( $y$ labels observed, variable $x_2$ unobserved)

Consider the following graphical model (note this corresponds to a naive Bayes classifier):

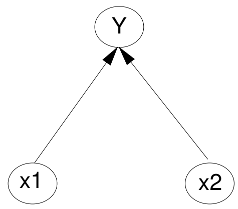



3.1.A Design an EM algorithm to learn the parameters of this Graphical Model from the union of data sets $L$ and $U_y$ (i.e., a combination of labeled and unlabeled data). Initialize your model parameters with MLE's based on $L$ only. Write down the E and M steps for refining these initial parameter estimates.

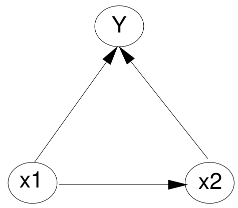
3.1.B Can the unlabeled data $U_y$ make any difference in this case in the learned $P(y|x_1, x_2)$? In other words, can you devise data sets $L$ and $U_y$ such that EM employing the union of $L$ and $U_y$ produces a different result than MLE's based on $L$? If yes, give a simple example data set and show the different results. In no, explain why not.

3.1.C Consider instead training using the union of data sets $L$ and $U_{x2}$. Can the unlabeled data $U_{x2}$ make a difference in this case in the learned $P(y|x_1, x_2)$? If yes, give a simple example data set and show the different results. In no, explain why not.

3.2.A,B,C Answer questions A, B, and C above for the following graphical model.




3.3.A,B,C Answer questions A, B, and C above for the following graphical model.




3.4 Would you like to make any conjectures in general about classes of graphical models for which unlabeled data can help to learn $P(y|x_1, \ldots, x_n)$?