10701/15781 Machine Learning, Spring 2007: Homework 5

Due: Wednesday, April 25, beginning of the class

Instructions

There are four questions on this assignment. Refer to the webpage for policies regarding collaboration, due dates, and extensions.

1 [25 points] Bayes Net structure learning [Brian]

1.1 Mutual information

Consider the following dataset with four binary variables A, B, C, D:

A	B	C	D
1	0	1	0
1	0	1	1
1	0	0	1
0	1	0	1
1	0	1	0

Your goal is to learn a Bayesian network structure for this data set.

1. Compute the mutual information $I(X, Y)$ for each pair of variables X, Y.
2. Using the computed mutual informations, find a *tree* Bayesian network that maximizes the likelihood of the training data.

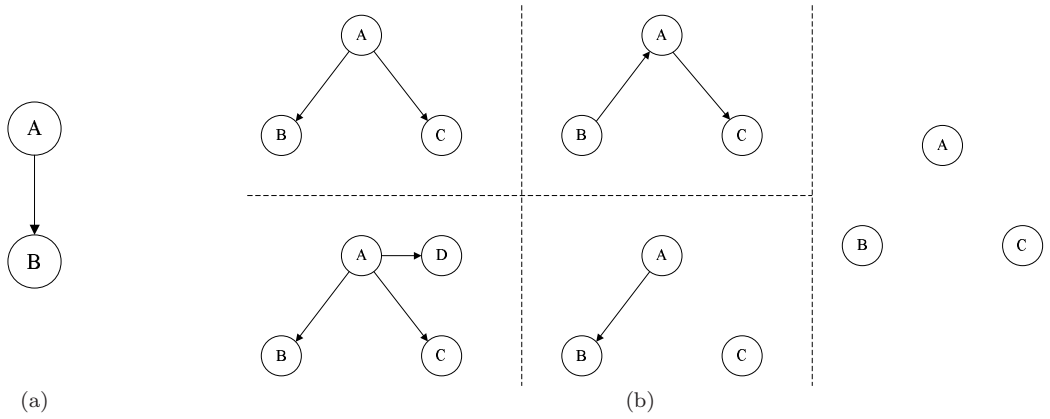

Figure 1: Bayesian networks for Question 1.2.1.

1.2 Learning forests with BIC score

In this part, you will derive a variation of the Chow-Liu algorithm for learning Bayesian networks with forest graph structure using the BIC score.

1. Recall that the BIC score is

$$\text{score}_{\text{BIC}}(\mathcal{G}; \mathcal{D}) = \log p(\mathcal{D} \mid \mathcal{G}, \theta_{\mathcal{G}}) - \frac{\text{NumberParams}(\mathcal{G})}{2} \log M,$$

   where $\mathcal{G}$ is a Bayesian network, $\mathcal{D}$ is the dataset, $\theta_{\mathcal{G}}$ are the parameters that represent the CPDs in $\mathcal{G}$, and NumberParams($\mathcal{G}$) is the total number of independent parameters in the CPDs of $\mathcal{G}$. For example, for the Bayesian network in Figure 1(a), the number of parameters is $N_A - 1 + N_A(N_B - 1) = N_A N_B - 1$, where $N_X = |\text{Val}[X]|$ is the number of possible values for variable X.

   Write down the number of independent parameters in the CPDs for each Bayesian network in Figure 1(b), in terms of N_A, N_B, N_C, and N_D. Show your work.

2. A forest is a graph that consists of one or more connected components, each of which is a tree (but not a poly-tree, i.e., each node has at most one parent). Prove that the number of independent parameters for any forest $\mathcal{G}$ can be written as

$$\text{NumberParams}(\mathcal{G}) = \sum_{\{X,Y\} \in \text{E}_{\mathcal{G}}} (N_X - 1)(N_Y - 1) + \left(\sum_{X \in \text{V}_{\mathcal{G}}} N_X \right) - |\text{V}_{\mathcal{G}}|. \quad (1)$$

   Here, $\text{E}_{\mathcal{G}}$ are the undirected edges of $\mathcal{G}$ and $\text{V}_{\mathcal{G}}$ are its vertices (variables).

   Hint: In a forest $\mathcal{G}$, $|V_{\mathcal{G}}| = |E_{\mathcal{G}}| + \#components_G$, where $\#components_G$ is the number of connected components in $\mathcal{G}$.

3. Using (1), describe an efficient algorithm for learning a forest Bayesian network that maximizes the BIC score. (*Hint: This algorithm is a small modification of Chow Liu.*)

2 [20 points] K-means and Gaussian Mixture Models [Andy]

1. Consider the data set in Figure 2. The '+' symbols indicate data points, and the (centers of the) squares A, B, and C indicate the starting cluster centers. Show the results of running the K-means algorithm on this data set. To do this, use the remaining figures, and for each iteration, indicate which data points will be associated with each of the clusters, and show the locations of the updated class centers. If a cluster center has no points associated with it during the cluster update step, it will not move. Use as many figures as you need until the algorithm converges.

2. Can the starting cluster assignments affect the final output of K-means? If so, draw a data set and two different starting cluster configurations such that running K-means (where k=3) until convergence yields two different results. If not, explain why the starting cluster assignments do not matter.

3. Draw a data set where a mixture of spherical Gaussians (where the covariance matrix is the identity matrix times some positive scalar) can model the data well, but K-means cannot.

4. Draw a data set where a mixture of diagonal Gaussians (where the covariance matrix can have non-zero values on the diagonal, and zeros elsewhere) can model the data well, but K-means and a mixture of spherical Gaussians cannot.

5. Draw a data set where a mixture of Gaussians with unrestricted covariance matrices can model the data well, but K-means and a mixture of diagonal Gaussians cannot.

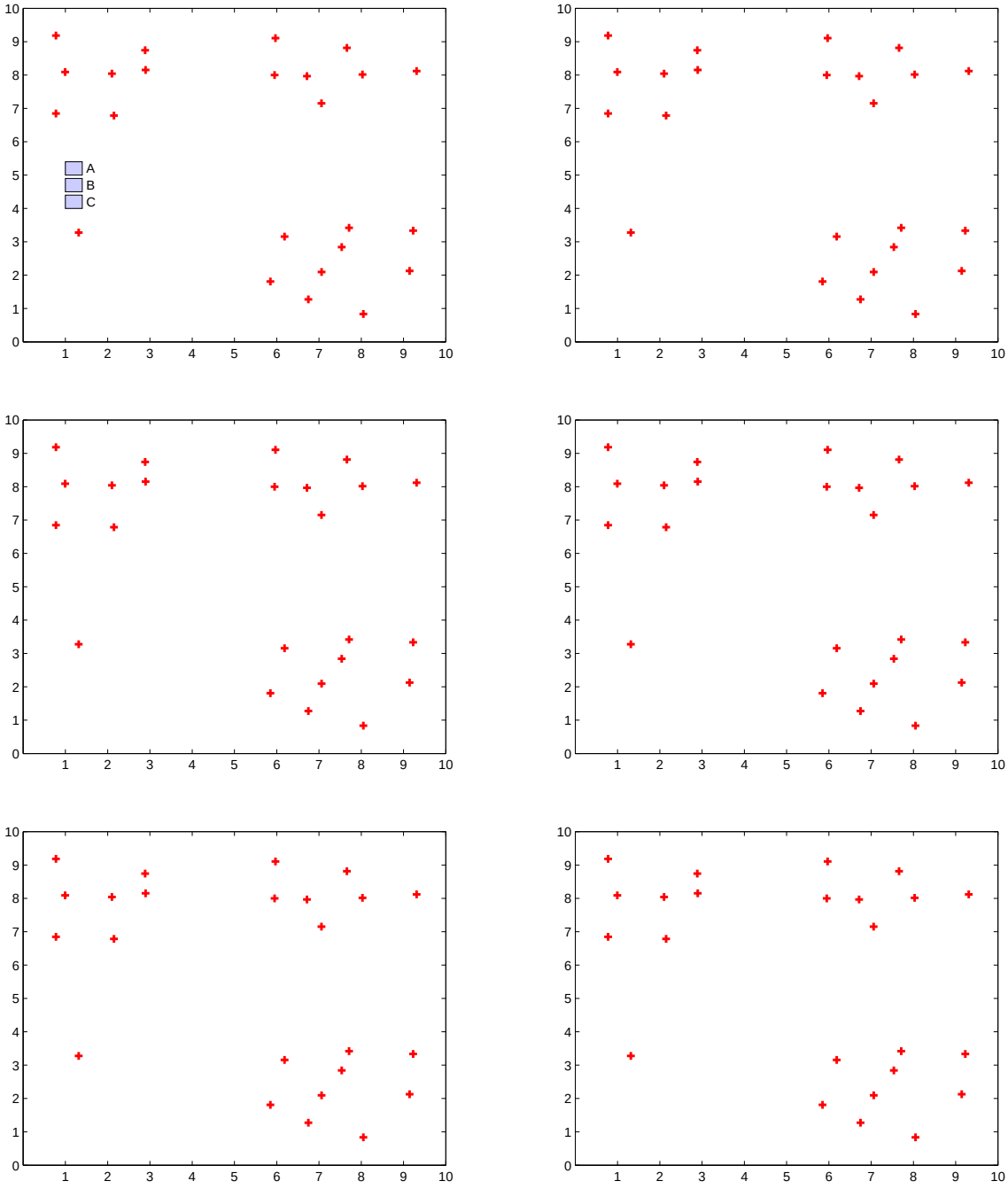

Figure 2: K-means data set

3 [20 points] Baum-Welch [Jon]

When fitting a model we generally want to choose parameters that will maximize the likelihood of our data. Expectation Maximization (EM) algorithm works by guessing initial parameter values, then estimating the likelihood of the data under the current parameters. These likelihoods can then be used to re-estimate the parameters, iteratively until a local maximum is reached.

In this problem, you will derive the EM algorithm for HMMs (also called the Baum-Welch algorithm). Unlike the problem from HW #4, the training set for this problem is assumed to be unlabeled. The objective here is to use EM to find the transition matrix (we will ignore the observation model for this problem, though it's not much harder to include it).

First we define two hidden variables. Given a series of observations $\mathbf{o}$, $\xi_t(i,j)$ is defined as the probability of being in state i at time index t and in state j at time index $t+1$:

$$\xi_t(i,j) = P(X_t = i, X_{t+1} = j|\mathbf{o})$$

and $\gamma_t(i)$ is the posterior probability

$$\gamma_t(i) = P(X_t = i|\mathbf{o})$$

where $\mathbf{o}$ is the sequence of observations.

1. (E-step)

   We saw in class that the E-step for Bayes Nets is just inference. For HMM, this means that we must calculate the distribution of hidden variables $\xi_t(i,j)$ and $\gamma_t(i)$. In class we also derived the expression of $\gamma_t(i)$ as

   $$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{\sum_i \alpha_t(i)\beta_t(i)}$$

   Now show that $\xi_t(i,j)$ can be computed as:

   $$\xi_t(i,j) = \frac{\alpha_t(i)a_{i,j}b_j(o_{t+1})\beta_{t+1}(j)}{P(\mathbf{o})}$$

   where α_t and β_t are the factors from Forward-Backward algorithm. $a_{i,j}$ and $b_j(o_{t+1})$ are the parameters of the model: $a_{i,j}$ is the probability of making a transition from state i to state j; o_{t+1} is the $t+1$-th element of the observation sequence $\mathbf{o}$; and $b_j(o_{t+1})$ is the probability of emitting o_{t+1} if we are in state j.

2. (M-step)

   The M-step for Bayes Nets sets:

   $$P(X_i = x_i|\mathbf{Pa}_{X_i} = \mathbf{z}) = \frac{\mathbb{E}[\text{Count}(X_i = x_i, \mathbf{Pa}_{X_i} = \mathbf{z})]}{\mathbb{E}[\text{Count}(\mathbf{Pa}_{X_i} = \mathbf{z})]}$$

   Use this M-step update rule to show that for Baum-Welch, we can re-estimate the model parameters using the following formulas:

   $$\pi(i) = \frac{\sum_n^N \gamma_{n,1}(i)}{N} \tag{2}$$

   where $\pi(i) = P(X_1 = i)$ is the starting state distribution, and N is the number of training examples. and

   $$a_{i,j} = \frac{\sum_n^N \sum_{t=1}^{T-1} \xi_{n,t}(i,j)}{\sum_n^N \sum_{t=1}^{T-1} \gamma_{n,t}(i)} \tag{3}$$

   where the $a_{i,j}$ are the transition probabilities as in part (1).

4 [35 points] Spectral Clustering [Jon & Purna]

There is a class of clustering algorithms, called spectral clustering algorithms, which has recently become quite popular. Many of these algorithms are quite easy to implement and perform well on certain clustering problems compared to more traditional methods like k-means. In this problem, we will try to develop some intuition about why these approaches make sense and implement one of these algorithms.

Before beginning, we'll review a few basic linear algebra concepts you may find useful for some of the problems.

- If A is a matrix, it has an v with eigenvalue λ if $Av = \lambda v$.
- For any $m \times m$ symmetric matrix A, the *Singular Value Decomposition* of A yields a factorization of A into

$$A = USU^T$$

where U is an $m \times m$ orthogonal matrix (meaning that the columns are pairwise orthogonal). and $S = diag(|\lambda_1|, |\lambda_2|, \ldots, |\lambda_m|)$ where the λ_i are the eigenvalues of A.

Given a set of m datapoints $x_1, \ldots, x_m$, the input to a spectral clustering algorithm typically consists of a matrix, A, of pairwise similarities between datapoints. A is often called the *affinity matrix*. The choice of how to measure similarity between points is one which is often left to the practitioner. A very simple affinity matrix can be constructed as follows:

$$A(i,j) = A(j,i) = \begin{cases} 1 & \text{if } d(x_i, x_j) < \Theta \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

where $d(x_i, x_j)$ denotes the Euclidean distance between points x_i and x_j.

The general idea of spectral clustering is to construct a mapping of the datapoints to an eigenspace of A with the hope that points are well separated in this eigenspace so that something simple like k-means applied to these new points will perform well.

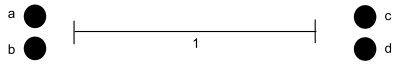

Figure 3: Simple dataset

As an example, consider forming the affinity matrix for the dataset in Figure 3 using Equation 4 with $\Theta = 1$. We have that

$$A = \left[\begin{array}{c|cccc} & a & b & c & d \\ \hline a & 1 & 1 & 0 & 0 \\ b & 1 & 1 & 0 & 0 \\ c & 0 & 0 & 1 & 1 \\ d & 0 & 0 & 1 & 1 \end{array}\right]$$

Now for this particular example, the clusters $\{a, b\}$ and $\{c, d\}$ show up as nonzero blocks in the affinity matrix. This is, of course, artificial, since we could have constructed the matrix A using any ordering of $\{a, b, c, d\}$. For example, another possibile affinity matrix for A could have been:

$$\tilde{A} = \left[\begin{array}{c|cccc} & a & c & b & d \\ \hline a & 1 & 0 & 1 & 0 \\ c & 0 & 1 & 0 & 1 \\ b & 1 & 0 & 1 & 0 \\ d & 0 & 1 & 0 & 1 \end{array}\right]$$

The key insight here is that the eigenvectors of matrices A and $\tilde{A}$ have the same entries (just permuted). The eigenvectors with nonzero eigenvalue of A are: $e_1 = (.7, .7, 0, 0)^T$, $e_2 = (0, 0, .7, .7)^T$. And the nonzero eigenvectors of $\tilde{A}$ are: $\tilde{e}_1 = (.7, 0, .7, 0)^T$, $\tilde{e}_2 = (0, .7, 0, .7)^T$. Spectral clustering embeds the original datapoints in a new space by using the coordinates of these eigenvectors. Specifically, it maps the point x_i to the point $(e_1(i), e_2(i), \ldots, e_k(i))$ where $e_1, \ldots, e_k$ are the top k eigenvectors of A. We refer to this mapping as the *spectral embedding*. See Figure 4 for an example.

1. For the dataset in Figure 5 assume that the first cluster has m_1 points and the second one has m_2 points. If we use equation 4 to compute affinity matrix A, is there a value of Θ for which you can analytically compute the first two eigenvalues and eigenvectors? If not, explain why not. If yes, what are they? What are the other eigenvalues ?

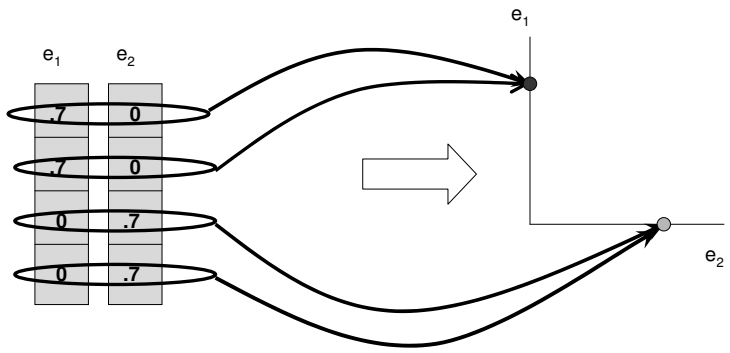

Figure 4: Using the eigenvectors of A to embed the datapoints. Notice that the points $\{a, b, c, d\}$ are tightly clustered in this space

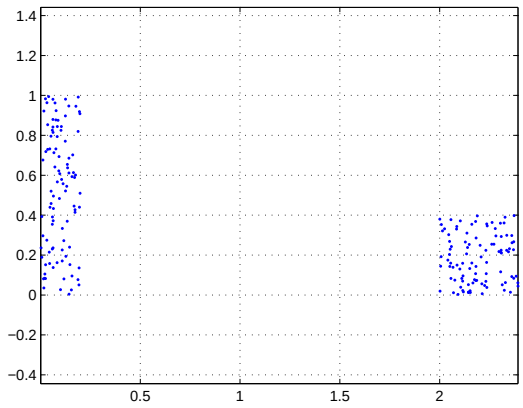

Figure 5: Dataset with rectangles

2. As in Figure 4 we can now compute the spectral embedding of the datapoints using the k top eigenvectors. For the dataset in figure 5 write down your best guess for the coordinates of the $k = 2$ cluster centers using the Θ that you picked in part 1.

3. A desirable property for any clustering algorithm is that its output should be be invariant with respect to the ordering of the datapoints. Let A be the affinity matrix constructed using the ordering $x_1, \ldots, x_m$. And let B be the affinity matrix constructed using the ordering $x_{\pi(1)}, \ldots, x_{\pi(m)}$ where π is a permutation of $\{1, \ldots, m\}$.

   A permutation matrix P is a matrix obtained by permuting the columns of an $m \times m$ identity matrix according to some permutation of the numbers 1 to m. Every row and column therefore contains precisely a single 1 with 0s everywhere else, and every permutation corresponds to a unique permutation matrix. For example, one possible permutation matrix is:

$$P = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

   (a) Show that there exists a permutation matrix P such that $AP = PB$. *Hint: Work a few examples to see how left multiplication by a permutation matrix compares to right multiplication by a permutation matrix.*

(b) Show that $P^T P = I$.

4. Show that A and B have the same eigenvectors up to a permutation of the entries. *Hint: Use the Singular Value Decomposition and the fact that both A and B are symmetric.*

4.1 Algorithm description

Frequently, the affinity matrix is constructed as

$$A_{ij} = e^{-d(x_i,x_j)^2/\sigma} \tag{5}$$

where σ is some user-specified parameter. The best that we can hope for in practice is a near block-diagonal affinity matrix. It can be shown in this case, that after projecting to the space spanned by the top k eigenvectors, points which belong to the same block are close to each other in a euclidean sense. We won't try to prove this, but using this intuition, you will implement one (of many) possible spectral clustering algorithms. This particular algorithm is described in

`On Spectral Clustering: Analysis and an algorithm`
`Andrew Y. Ng, Michael I. Jordan, Yair Weiss (2001)`

We won't try to justify every step, but see the paper if you are interested. The steps are as follows:

- Construct an affinity matrix A using Equation 5.
- Symmetrically 'normalize' the rows and columns of A to get a matrix N: such that $N(i,j) = \frac{A(i,j)}{\sqrt{d(i)d(j)}}$, where $d(i) = \sum_k A(i,k)$.
- Construct a matrix Y whose columns are the first k eigenvectors of N.
- Normalize each row of Y such that it is of unit length.
- Cluster the dataset by running k-means on the set of spectrally embedded points, where each row of Y is a datapoint.

1. Run k-means on the datasets provided on the class webpage and provide plots of the results. For text.mat, take $k = 6$. For all others use $k = 2$.
2. Implement the above spectral clustering algorithm and run it on the four provided datasets using the same k. Plot your clustering results using $\sigma = .025, .05, .2, .5$. *Hints: You may find the MATLAB functions pdist and eig to be helpful. A function plotClusters.m has been provided to help visualize clustering results.*
3. Plot the first 10 eigenvalues for the rectangles.mat and text.mat datasets when $\sigma = .05$. What do you notice?
4. How do k-means and spectral clustering compare?

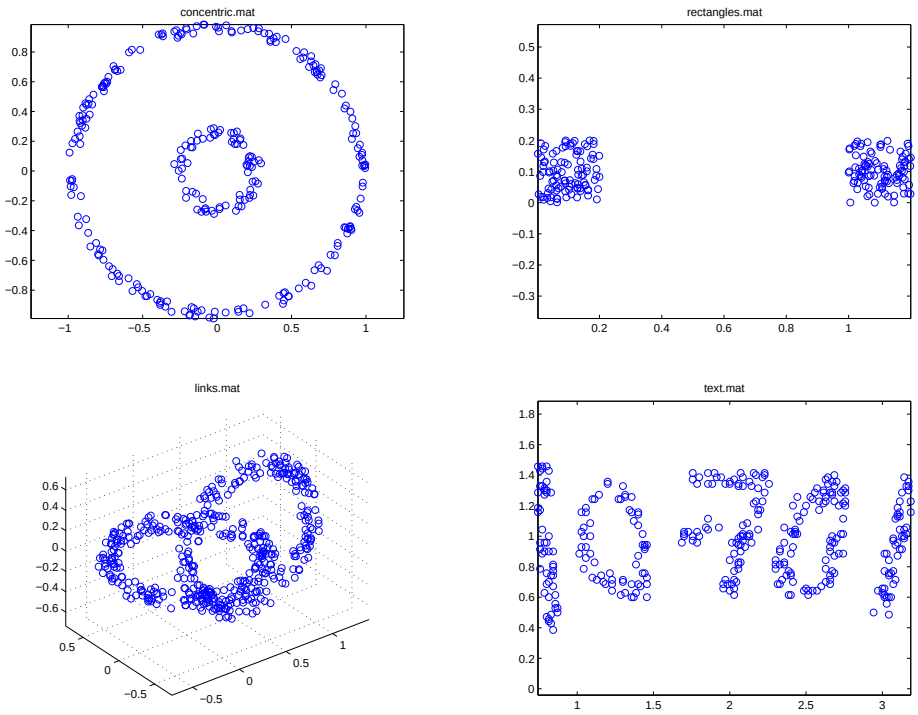

Figure 6: Synthetic Datasets.