CSE474/574: Introduction to Machine Learning(Fall 2012)

Instructor: Sargur N. Srihari
Teaching Assistants: Yu Liu, Yingbo Zhou

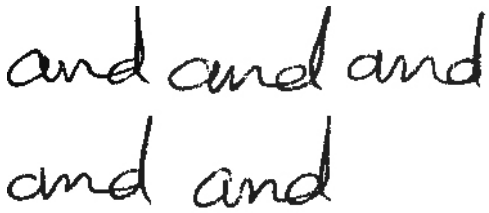
Project 2: Probabilistic Graphical Models: Model Evaluation, Inference and Sampling

Due Date: Thursday, Dec. 15, 2012.

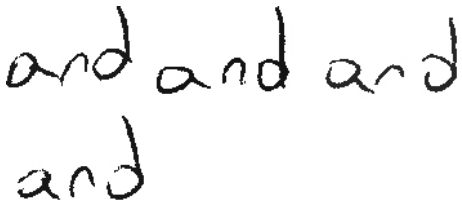
1 Objective

The goal of this project is to determine probabilistic graphical models (PGMs) from given data, evaluate them for the purpose of learning the best model, perform probabilistic inferences using them and sample data from them. Please read this project description in its entirety before beginning the project.

You will work with data that consists of discrete-valued variables which are characteristics of handwriting provided by humans. Given several examples of the handwritten word *and* provided by a writer, a document examiner assigns values to several characteristic features. These characteristics are different depending on whether the writing is *cursive* or *hand-print* (Table 1). Some examples and the values assigned to nine characteristics for two writers are shown in Figure 1. The images are provided only for interest and you will only be working with discrete data values that are provided to you.



(a)



(b)

Figure 1: Examples of handwritten *and* from two writers: (a) Five examples from Writer 0001 who writes cursively (where the image files from top-left to bottom-right correspond to 0001a_69.png, 0001a_89.png, 0001a_99.png, 0001a_126.png, and 0001a_152.png); the nine cursive characteristics for this writer are: $x_1 = 1, x_2 = 1, x_3 = 1, x_4 = 1, x_5 = 2, x_6 = 2, x_7 = 1, x_8 = 2$, and $x_9 = 2$; and (b) four examples from writer 0002, which is considered to be hand-print (the image files are 0002a_83.png, 0002a_93.png, 0002a_116.png, and 0002a_120.png) with nine hand-print characteristics $x_1 = 1, x_2 = 4, x_3 = 0, x_4 = 3, x_5 = 1, x_6 = 1, x_7 = 3, x_8 = 1$, and $x_9 = 0$.

We would like to create PGMs so that we can evaluate the probability of any given handwriting sample. The characteristics of *and*, as given by document examiners, are in Table 1– which depend on whether the handwriting is 'cursive' or 'hand-printed'. In both cases there

(a) Cursive Characteristics

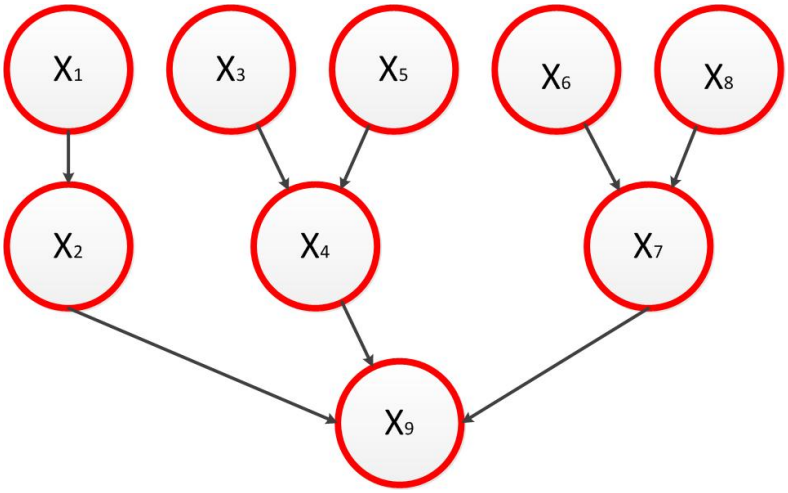
Initial stroke of formation of 'a' (x_1)	Formation of staff of 'a' (x_2)	Number of arches of 'n' (x_3)	Shape of arches of 'n' (x_4)	Location of mid-point of 'n' (x_5)	Formation of staff of 'd' (x_6)	Formation of initial stroke of 'd' (x_7)	Formation of terminal stroke of 'd' (x_8)	Symbol in place of the word 'and' (x_9)
Right of staff (0)	Tented (0)	One (0)	Pointed (0)	Above baseline (0)	Tented (0)	Over-hand (0)	Curved up (0)	Formation (0)
Left of staff (1)	Retraced (1)	Two (1)	Rounded (1)	Below baseline (1)	Retraced (1)	Under-hand (1)	Straight across (1)	Symbol (1)
Center of staff (2)	Looped (2)	No fixed pattern (2)	Retraced (2)	At baseline (2)	Looped (2)	Straight across (2)	Curved down (2)	None (2)
No fixed pattern (3)	No staff (3)		Combination (3)	No fixed pattern (3)	No fixed pattern (3)	No fixed pattern (3)	No obvious ending stroke (3)	
	No fixed pattern (4)		No fixed pattern (4)				No fixed pattern (4)	

(b) Hand-print Characteristics

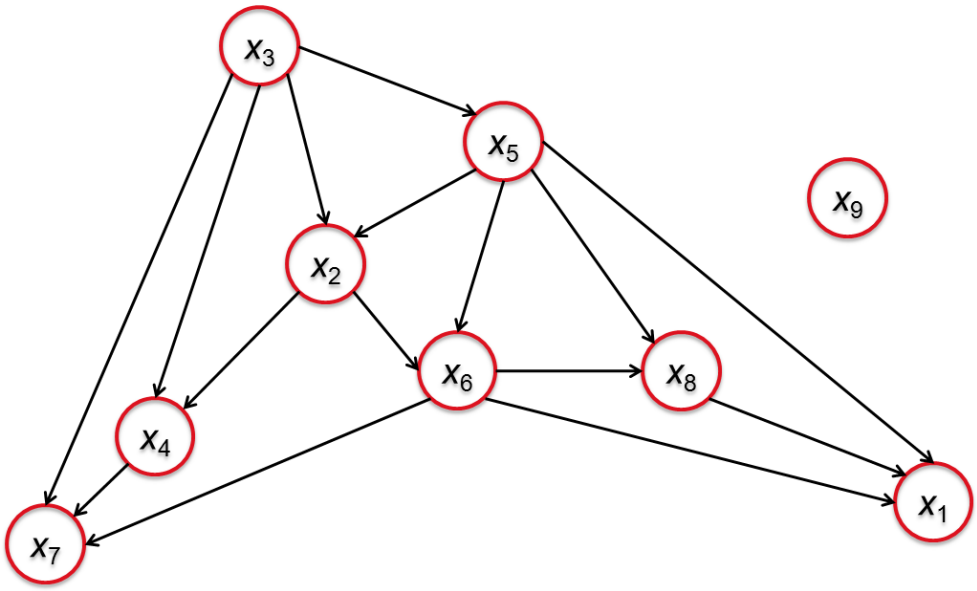
Number of strokes for formation of 'a' (x_1)	Formation of staff of 'a' (x_2)	Number of strokes for formation of 'n' (x_3)	Formation of staff of 'n' (x_4)	Shape of arch of 'n' (x_5)	Number of strokes for formation of 'd' (x_6)	Formation of staff of 'd' (x_7)	Initial stroke of 'd' (x_8)	Unusual formation (x_9)
One continuous (0)	Tented (0)	One continuous (0)	Tented (0)	Pointed (0)	One continuous (0)	Tented (0)	Top of staff (0)	Formation (0)
Two strokes (1)	Retraced (1)	Two strokes (1)	Retraced (1)	Rounded (1)	Two strokes (1)	Retraced (1)	Bulb (1)	Symbol (1)
Three strokes (2)	Looped (2)	Three strokes (2)	Looped (2)	No fixed pattern (2)	Three strokes (2)	Looped (2)	No fixed pattern (2)	None (2)
Upper case (3)	No staff (3)	Upper case (3)	No staff (3)		Upper case (3)	Single down (3)	Undetermined (3)	
No fixed pattern (4)	Single line down (4)	No fixed pattern (4)	No fixed pattern (4)		No fixed pattern (4)	Single up (4)		
	No fixed pattern (5)					No fixed pattern (5)		

Table 1: Characteristics of handwritten *and*. There are nine characteristics, which are different depending on whether the writing is: (a) *cursive* or (b) *hand-print*.

are nine random variables x_1-x_9 which take on five different values for cursive (0-4) and six values for hand-print (0-5) as indicated in parentheses. Four possible PGM structures for the characteristics are shown as Bayesian Networks (BNs) in Figure 2 and as Markov Networks (MNs) in Figure 3.



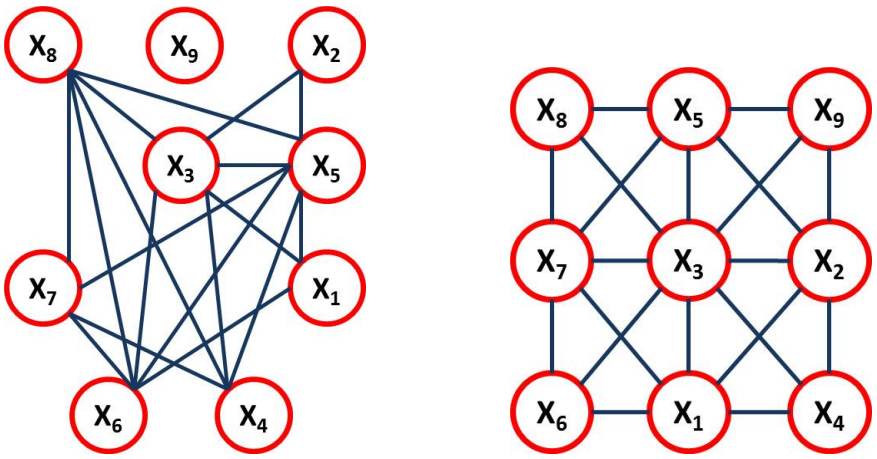
(a)



(b)

Figure 2: Possible Bayesian Networks for the characteristics *and*: (a) cursive (nine variables), and (b) hand-print (nine variables).

In this project, you will need to learn the parameters of these models, evaluate the models and perform inferences. In addition, you will also have a chance to design you own PGM, or learn the structure of the model from the dataset and evaluate and compare the models.



(a) (b)

Figure 3: Possible Markov Networks for *and*: (a) cursive and (b) hand-print.

2 Tasks

2.1 Bayesian Networks

1. **Parameters**: Determine all the parameters for the two BNs shown in Figure 2 using the data files as described in Section 4. Also determine the parameters for BNs where all variables are assumed to be independent.

2. **Evaluations**: Which of the BNs (given in the figure or the alternative one that assumes independence) represents the data set the best? Use the definition of log-loss given in Appendix A.

3. **Inferences**: Using these BNs answer the different probabilistic queries given in Section 3. Give examples of common and rare *and* for both cursive and hand-print.

4. **Sampling**: Generate samples from one or more BNs using a method such as ancestral or Gibbs sampling, and show the images associated with the characteristics generated. If you estimate the parameters using maximum likelihood, then you can definitely find the sample from your dataset. If you applied priors (smoothing) for estimating the parameters, you can restrict your sample to be inside the dataset by restricting the features take on values that only exist in the dataset.

5. **Optional**: See if you can construct a better BN (lower log-loss) for this data. For example, you can learn the structure from the dataset by determining the dependencies between two variables by computing the χ^2 value or evaluate the mutual information followed by running a maximal spanning tree algorithm to get the structure (this is essentially the Chow-Liu tree algorithm).

2.2 Markov Networks

The MN structures represented in Figure 3 can correspond to many different MNs. We can define clique potentials as described in Section 5.1 of Appendix B. Alternatively in order to reduce

number of parameters we may assume that some values of some potentials are set to be 1 as described in Section 5.2 of Appendix B. Hence, consider using feature-based description of MNs that can be found in files 'MNfeatures-handprint AND.txt' and 'MNfeatures-cursive AND.txt'. Determine the parameters and evaluate the models using log-loss shown in Appendix B.

Use a MN representation (cliques or log-linear model) and perform the evaluations and inferences as described above for BNs. Include an MN that assumes all features are independent.

3 Inference Tasks

We first define several probabilities that deal with comparing two samples of handwriting. Then we pose some probabilistic queries.

3.1 Definitions

For a given measurement $\mathbf{x}$ with known distribution $p(\mathbf{x})$, the following related measures can be defined: PRC or the Probability of Random Correspondence, nPRC or the PRC of at least one pair among n, and conditional nPRC which is the PRC of a given (or specific) characteristics $\mathbf{x}$ among n. These definitions are formalized below.

- PRC: probability that two randomly chosen samples $\mathbf{x}$ and $\mathbf{y}$ have the same characteristics within specified tolerance, ϵ, is

$$\rho = p(z = 1) = \sum_{\mathbf{x}} \sum_{\mathbf{y}} p(z = 1|\mathbf{x}, \mathbf{y}) p(\mathbf{x}) p(\mathbf{y}) \tag{1}$$

  where $p(\mathbf{x}) = p(\mathbf{y})$ is the distribution of $\mathbf{x}$ and z is a indicator variable such that $p(z = 1/\mathbf{x}, \mathbf{y}) = 1$ if $\mathbf{x} = \mathbf{y} \pm \epsilon$ and is 0 otherwise.

- nPRC: the probability that among a set of n samples $\mathbf{y}_1, .., \mathbf{y}_n$, some pair have the same characteristics, within specified tolerance (ϵ), where $n \geq 2$. The nPRC, can be written in terms of the PRC as

$$p(z' = 1) = \rho[n] = 1 - (1 - \rho)^{\frac{n(n-1)}{2}} \tag{2}$$

  where $z' = 1$ if there is a matching pair and 0 otherwise. Note that when $n = 2$, PRC=nPRC. Since there are $\binom{n}{2}$ pairs involved this probability can be much higher than PRC. For instance, in the famous birthday paradox, while the probability of a birthday (PRC) is 1/365, the value of nPRC for $n = 24$ is 0.5.

- Conditional nPRC: the probability that in a set of n samples, a specific one with value $\mathbf{x}$ coincides with another sample, within specified tolerance. Since we are trying to match a specific value $\mathbf{x}$, this probability depends on the probability of $\mathbf{x}$. It is smaller than nPRC and can be lower than the PRC. The exact relationship with respect to PRC depends on the distribution of $\mathbf{x}$.

  The conditional nPRC is given by $p(x = x_s|x_s, n) = 1 - (1 - p(x_s))^{n-1}$ where x_s is the given specific evidence, x is a random sample from a data set (size of n), and $p(x_s)$ is the probability of x_s.

3.2 Probabilistic Queries

For each of the PGMs considered, determine the following:

1. The PRC for the PGM.
2. The conditional nPRC for some samples.
3. The probability of finding a particular example of *and* among n writers (conditional nPRC). For each example make a plot with respect to n.

4 Data Files

The files provided to you are as follows:

1. and-dataset.zip - 15,500 automatically extracted images of 'and'
2. truthCursive.txt - ground truth labels of 3,075 writers have cursive writing 'and'
3. truthHandprint.txt - ground truth labels of 1,135 writers have handprint writing 'and'
4. Files for MN features: 'MNfeatures-andprint AND.txt' and 'MNfeatures-cursive AND.txt'.

The ground truth label files all have the same format as follows:

writer_id, document_id, value of x_1, value of x_2, ..., value of x_9

for example if you get a ground truth label as '1, a, 1, 1, 1, 1, 2, 2, 1, 2, 2', that means this label is for writer one, document a, and the features from x_1 to x_9 take on the values 1, 1, 1, 1, 2, 2, 1, 2, 2 respectively. Note that since there are multiple images from the same writer on the same document, the ground truth label is summarized from all the images of the same writer on that document. For instance, for writer 1's document A she got five different images of 'and' (i.e. 0001a_69.png, 0001a_89.png, 0001a_99.png, 0001a_126.png, and 0001a_152.png), the above label is a summarization from all these five images.

5 Submission

1. **Project Report:** Write a project report where you explain your models and tabulate your different results clearly. Your report can be in pdf or word format (you can also submit latex source packages). Use the following guidelines in writing your report:
   (a) Title of Project with your name
   (b) Section 1– BN models: List the models evaluated. If you develop new BNs explain the intuition of the model structure. Which of the models performed best? Did the model that assumed independence of characteristics perform any worse than the mre complex models? Also answer the probabilities queries, with a plot of nPRC against n.
   (c) Section 2– MN models: compare clique-based and feature-based models.

(d) Section 3– Conclusions based on all experiments. Compare the results obtained using BN and MN models. If you tried both types of MN (factor and feature) how do the results compare?

(e) Additional grading considerations will include creativity in choice of models, completeness in interpreting your statistics, and the clarity and flow of your report.

2. **Matlab code:** Submit whatever code you have written in this project. It includes the codes for data preparation, parameter learning, inference, and maybe evaluation, etc. You should include short usage/information about arguments, return values of your functions in your report or as comment in your code in order for us to verify that your code produces the results you submitted.

   All the files mentioned above should be submitted via the CSE submit script, e.g.
   Submit for undergraduates: `submit_cse474 train.m infer.m report.pdf`
   Submit for graduates: `submit_cse574 train.m infer.m report.pdf`

   It is your responsibility to make sure that your submission is received before deadline (e.g. you should complete your project and submit it before hand to avoid connection issues on the exact time of deadline).

3. **Additional Copy of Project Report:** Email an additional copy of your completed project report (pdf or latex folder) to Professor Srihari (Srihari@buffalo.edu).

Appendix A: Log-Loss of Bayesian Networks

The evaluation of the likelihood of a BN is straight-forward. If $\mathbf{x} = \{x_1, x_2, \ldots, x_m\}$ are variables in the BN, then the joint probability is given by:

$$p(\mathbf{x}) = \prod_{i=1}^{m} p(x_i|x_{\pi_i})$$

where x_{π_i} denotes the set of nodes that is parent of node x_i. Then the likelihood of the BN model on a particular dataset $\mathcal{D}$ follows immediately,

$$L(\mathcal{D}) = \prod_{\mathbf{x}\in\mathcal{D}} p(\mathbf{x}) = \prod_{\mathbf{x}\in\mathcal{D}} \prod_{i=1}^{m} p(x_i|x_{\pi_i})$$

and the log-likelihood is::

$$\log L(\mathcal{D}) = \sum_{\mathbf{x}\in\mathcal{D}} \log p(\mathbf{x}) = \sum_{\mathbf{x}\in\mathcal{D}} \sum_{i=1}^{m} \log p(x_i|x_{\pi_i}).$$

The log-loss is given by simply negating the log-likelihood as $l(\mathcal{D}) = -\log L(\mathcal{D})$.

You will need to estimate the conditional probabilities $p(x_i|x_{\pi_i})$ in the BN from the given data to calculate the likelihood.

You may want to use Dirichlet priors to deal with insufficient data.

Appendix B: Log-Loss of Markov Networks

MNs can be parameterized in several ways: one based on clique potentials and the other, called a *log-linear representation* based on features.

5.1 Using Factors

In the case of a Markov network with variables $\mathbf{x} = \{x_1, x_2, \ldots, x_m\}$, let us denote a clique by C and the set of variables in that clique by $\mathbf{x}_C$. Then the joint distribution is written as a product of *potential functions* $\psi_C(\mathbf{x}_C)$ over the maximal cliques of the graph

$$p(\mathbf{x}) = \frac{1}{Z} \prod_{C} \psi_C(\mathbf{x}_C)$$

where Z is the partition function:

$$Z = \sum_{\mathbf{x}} \prod_{C} \psi_C(\mathbf{x}_C)$$

obtained by summing over all possible values of $\mathbf{x}$. By considering only potential functions which satisfy $\psi_C(\mathbf{x}_C) \geq 0$ we ensure that $p(\mathbf{x}) \geq 0$. Then the log-likelihood of a Markov model on a dataset $\mathcal{D}$ is:

$$\log L(\mathcal{D}) = \sum_{\mathbf{x}\in\mathcal{D}} \sum_{C} \log \psi_C(\mathbf{x}_C) - |\mathcal{D}| \log Z.$$

where $|\cdot|$ returns the cardinality. As before, log-loss is given by $l(\mathcal{D}) = -\log L(\mathcal{D})$.

You will need to estimate the potential functions $\psi_C(\mathbf{x}_C)$ for all the cliques. Count the number of times of the feature values that co-occur in the same clique. For example if x_1 and x_2 are in the same clique, to estimate the potential function on this clique, you need to count the number of times that x_1 taken on value 0 and x_2 taken on value 0, x_1 taken on value 0 and x_2 taken on value 1, and so on, till all possible combinations are exhausted.

In principle we can enumerate all possible values of the variables in factor ψ to get the partition function Z. However, since the number of possible combinations is large, a simplistic approach is to only use only pairwise factors. It approximates the original MN and uses fewer parameters (but it is not as accurate) Alternatively, you can use the *log-linear* representation of MNs using features described in the next section.

5.2 Log-linear model using Features

Given a set of features $F = \{f_i(C_i)\}_{i=1}^k$, where $f_i(C_i)$ is a feature function defined over the variables C_i, a log-linear model for Markov Network is as follows:

$$p(\mathbf{x} : \boldsymbol{\theta}) = \frac{1}{Z(\boldsymbol{\theta})} \exp\left(\sum_{i=1}^{k} \theta_i f_i\left(\mathbf{x}\langle C_i\rangle\right)\right)$$

where $\mathbf{x}\langle C_i\rangle$ is a set of values in $\mathbf{x}$ for variables in C_i; $\boldsymbol{\theta}$= $\{\theta_i\}_{i=1}^k$ is the set of weight parameters, and $Z(\boldsymbol{\theta})$ is the partition function:

$$Z(\boldsymbol{\theta}) = \sum_{\mathbf{x}} \exp\left(\sum_{i=1}^{k} \theta_i f_i\left(\mathbf{x}\langle C_i\rangle\right)\right)$$

Feature functions f_i in the form of binary-valued indicator functions are provided as files to you. For learning θ_i from the data use gradient ascent with the goal to maximize log-likelihood (you can read about this procedure in section 20.3.1 in the PGM book).

Then the log-likelihood of a Markov Network on a dataset $\mathcal{D}$ is as follows:

$$\log L(\mathcal{D}) = \sum_{i=1}^{k} \theta_i \cdot \left(\sum_{\mathbf{x}\in\mathcal{D}} f_i\left(\mathbf{x}\langle C_i\rangle\right)\right) - |\mathcal{D}| \cdot \ln Z(\theta)).$$

From the expression it is clear that in order to calculate log-likelihood one needs to estimate the feature functions $F = \{f_i(\mathbf{x}\langle C_i\rangle)\}_{i=1}^k$ for all samples $\mathbf{x} \in \mathcal{D}$ and run inference to get a value of the partition function $Z(\theta)$.