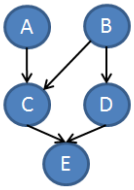
PGM_2012_Fall – Assignment 1 ---- Part1

Prof. Shou-de Lin, TA: TingWei Lin, ChungYi Li, EnHsu Yen

Deadline: Hand-written section: 10/9 (Tue) before class. (Hand in hard copy)
Programming section: 10/14 (Sun) 23:59.

A. Hand-Written Section

Problem 1: A, B, C, D and E are random variables and their values are denoted as a_i, b_j, c_k, d_m, e_n. Their relationship can be expressed as a Bayesian Network:



(a) Write down the factored form of the joint probability $p(a_i, b_j, c_k, d_m, e_n)$ using the Bayesian Network. (This should take you 1 min.)

(b) Use (a) to prove that the joint probability sums up to 1

$$\sum_{i,j,k,m,n} p(a_i, b_j, c_k, d_m, e_n) = 1,$$

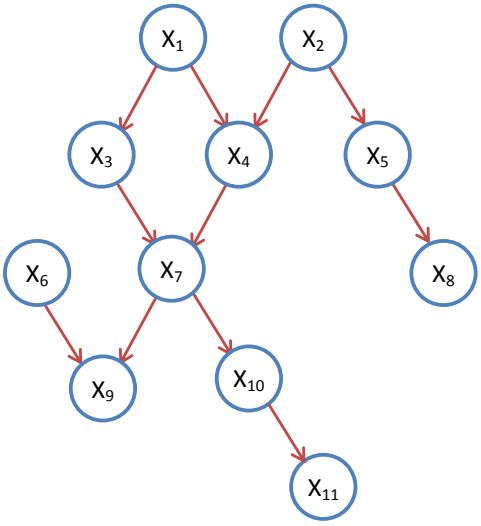
given that A, B, C, D and E are discrete random variables and each CPD in the Bayesian Network sums up to 1. EX: $\sum_i p(a_i) = 1$ and $\sum_k p(c_k|a_i) = 1$.

Note: please be clear in your proof. For example:

$\sum\sum xy$ is $\sum(\sum(xy))$ or $\sum(\sum(x)y)$?
$\sum x \sum y$ is $(\sum x)(\sum y)$ or $\sum(x(\sum y))$?
$\sum p(x_i|y_j)$ is $\sum_i p(x_i|y_j)$ or $\sum_{ij} p(x_i|y_j)$?

Problem 2: Look at the figure below. For each of the following assertions of (conditional) independence, state if they are True or False with brief explanation using d-separation.

1. $X_1 \perp X_6$
2. $X_8 \perp X_{11}$
3. $X_2 \perp X_8 \mid X_5$
4. $X_1 \perp X_9 \mid X_7$
5. $X_1 \perp X_7 \mid X_4$
6. $X_2 \perp X_3 \mid X_{10}$
7. $X_3 \perp X_{11} \mid X_9, X_{10}$
8. $X_3 \perp X_6 \mid X_9$
9. $X_2 \perp X_7 \mid X_3, X_4, X_6, X_9, X_{10}$



B. Programming Section: GMTK Toolkit

In the PGM class, we have made a brief introduction to the Graphical Models Toolkit (GMTK), and that should be enough for you to do this homework. If you still have problems when you use it, you may ask TAs or refer to the GMTK documentation files for more information.

For each problem in this section, we will provide you at least four files, which are

1. prob[N].str
2. prob[N].master
3. Makefile
4. prob[N].train.pfile

where N is problem id. The str file specifies the structure of your graphical model, the master file specifies the probability tables/distributions in your graphical model, and the pfile is the training data which is used to learn the parameters. To complete this homework, you should need to modify the str, and maybe the master file as well as the Makefile.

If you finish your model and want to evaluate it, you can type

```
make train
```

The result CPT(conditional probability table) will be save as prob[N].cpt.
And if you want to clean the output files, you can type “make clean”.

Your model will be evaluated based on a strategy called maximum likelihood estimation (MLE). That is, a good model should assign high probability to the observed data. Furthermore, here we will hide partial of the data as testing data to evaluate your model. The reason we do so is to prevent you from over-fitting the training data. That is, a proper model should have high likelihood on both training and the unseen testing data.

Submission format:

1. For each problem, put the following files in a directory named with prob[N]:
    1. prob[N].str
    2. prob[N].master
    3. Makefile
    4. prob[N].cpt
2. Write a short report about your idea and the graph you designed.
3. Put the report and the two problem directories in one single directory named with your student ID and zip it before uploading it to the CEIBA.

The quality of your model will be judged according to the following criteria:

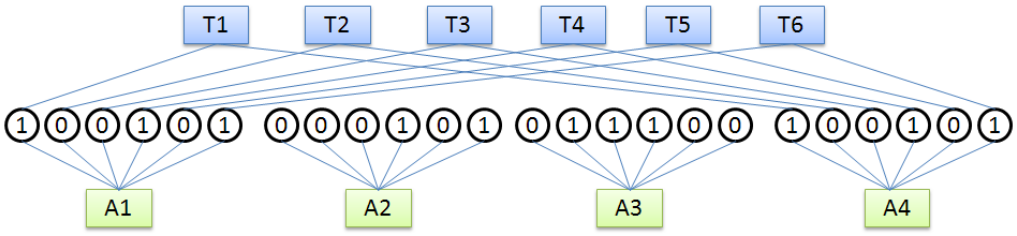
1. The complexity of your model. (i.e. #parameters). The fewer the better.
2. The likelihood of your model on training/testing data. The greater the better.
3. The quality/novelty/validity of your idea in the report.

Problem 1: Training with Multiple Annotators' Data

Generally for supervised learning, we are provided (usually by some domain experts) with a set of training pairs (x_i, y_i). In real world, it may be too expensive to generate the perfect label (gold standard) y_i for each training data. Instead, we may have multiple annotators to mark y_i based on their own judgments. Thus, they may mark the same y_i differently. Our overall goal is design a graphical model to pursue supervised learning using such multiple annotators' data as training inputs. In this problem, there are N annotators labeling M tasks. Each task is annotated by N annotators and each annotator provides labels for M tasks. The annotators can be roughly grouped into two categories: expert and rookie. The true label (T_i) and the category each annotator belongs (A_i) to are unknown. Try to design a model to explain this kind of data.

Note: the annotators are the same in the training/testing data, but the tasks are not.

Hint: when you are designing your str file, try to share the common parameters to reduce the model complexity



(All annotators & tasks have connections, not all plotted to keep diagram uncluttered.)

Data description:

Global information:

1. (#segment, #frame in a segment, dimension of one frame) = (1, 1, 100)
2. Each frame is $\{r_{1,1}, r_{1,2}, \ldots, r_{i,j}, \ldots r_{10,10}\}$, where $r_{i,j}$ is the label that annotator i marks task j.

Variable information:

Property\Variable	Annotator	Task	Label
#	10	10	100
Value	{rookie, expert}	{0, 1}	{0, 1}
Cardinality	2	2	2
Observed	F	F	T

Problem 2: HMM with variable duration of state

Hidden Markov Model has been used in many applications. But it is often criticized that the duration (or # of times) a Markov Chain stay in a state follows a geometric distribution, which is inconsistent with data in many applications. For example, here is a data sequence:

```
0000000 11111111 22222222 111111 0000000 22222222
```

If training a HMM with the data, we will get a degenerate model in which every state tends to not transiting to other states. To solve such problem, we have to modify the original HMM and make the duration of state follow any given discrete distribution.

There're several ways to deal with the problem, one of which is to add one "counter variable". The counter variable is used to count and decide whether the current state should transit to the next state. That is, assumed that the counter variable has initial value N_0, and we decrement its value in every timestamp of one state. When the counter variable reaches zero, it can tell the HMM to transit to the next state and reset itself to another initial value N_0'. By doing so, the duration of state follows the distribution of the counting length (N_0), which means that we can apply any discrete distribution on the duration of state transition on HMM.

To simplify the problem, we limit the duration of any state to be in the interval [6, 8] and the data sequences contain only three kinds of value {0, 1, 2}.

Hint: in this problem you may need to use decision tree and deterministic CPT to reduce the number of parameters.

Data description:

Global information:

1. (#segment, #frame in a segment, dimension of one frame)
   = (10, variable-length, 1)
2. Each frame is {x}, where x is the outcome.

Variable information:

Property\Variable	State	Outcome
Value	{s1, s2, s3}	{0,1,2}
Cardinality	3	3
Observed	F	T