PGM_2012_Fall – Assignment 2 ---- Part1

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

Deadline: Hand-written section: 10/30 (Tue) before class. (Hand in hard copy)
Programming section: 11/04 (Sun) 23:59.

A. Hand-Written Section: Textbook exercises

This section is for individual work, and **EVERYONE** should do the following two problems.

Problem 1: see PGM textbook exercise 5.2. (the answer is yes, and you have to explain why)

Exercise 5.2. *Consider a pairwise Markov network defined on binary variables:*

$$p(x) = \phi(x_1, x_{100}) \prod_{i=1}^{99} \phi(x_i, x_{i+1}) \tag{5.7.2}$$

Is it possible to compute $\underset{x_1,\ldots,x_{100}}{\operatorname{argmax}}\, p(x)$ *efficiently?*

Problem 2: see PGM textbook exercise 5.4.

Exercise 5.4. *Consider the hidden Markov model:*

$$p(v_1, \ldots, v_T, h_1, \ldots, h_T) = p(h_1)p(v_1|h_1) \prod_{t=2}^{T} p(v_t|h_t)p(h_t|h_{t-1}) \tag{5.7.3}$$

in which $dom(h_t) = \{1, \ldots, H\}$ *and* $dom(v_t) = \{1, \ldots, V\}$ *for all* $t = 1, \ldots, T$.

1. *Draw a belief network representation of the above distribution.*
2. *Draw a factor graph representation of the above distribution.*
3. *Use the factor graph to derive a Sum-Product algorithm to compute marginals* $p(h_t|v_1, \ldots, v_T)$. *Explain the sequence order of messages passed on your factor graph.*
4. *Explain how to compute* $p(h_t, h_{t+1}|v_1, \ldots, v_T)$.
5. *Show that the belief network for* $p(h_1, \ldots, h_T)$ *is a simple linear chain, whilst* $p(v_1, \ldots, v_T)$ *is a fully connected cascade belief network.*

B. Programming Section: Inference for Topic Model

This section is for a **GROUP**. Students should form a group of three and one group needs to turn in one result. You can use whatever programming language you want, but you can't use toolkits like GMTK to generate the results.

In the following problems, you are introduced to the world of Topic Model. Topic model is used to model the relationship between words and documents. In the real world, a document can talk about several topics. Also the same word can appear with different probabilities under different topics. Since the topics are considered as hidden variables (i.e. not given), a topic model is used to learn such hidden topics to explain the relationship of documents and words. However, since we haven't taught you how to learn the hidden topics and parameters (ex: distribution of words under different topics), we will explicitly provide the parameters for you to **infer the marginal distribution** of topics in this assignment.

Submission instruction:

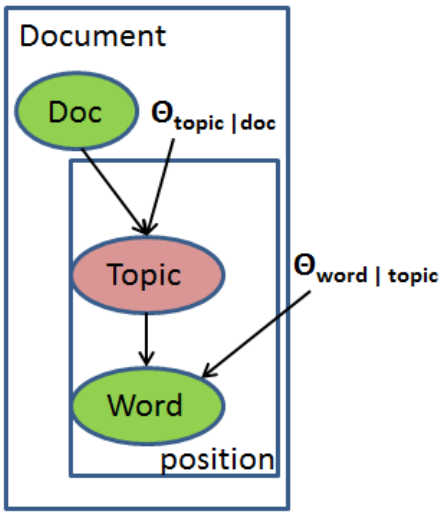
1. Write a report for two problems: (Be clear but concise)
   - Write the list of your members, and how you divide the work
   - In problem 1, write the Bayes rule for P(topic | word, document) and how you implement you model.
   - In problem 2, briefly show (a) (b) (c) and explain how you implement the model.
2. The directory tree of your submission should be:

```
[Student ID]
    | -- [your report for two problems]
    | -- prob1
    |       | -- src
    |       |       ` [your source code for problem 1]
    |       | -- ap.[K].plsa.pdf
    |       ` -- ap.topic.[K].plsa
    ` -- prob2
            | -- src
            |       ` [your source code for problem 2]
            | -- ap.[K].seqPlsa.pdf
            | -- ap.topic.[K].seqPlsa
            ` -- ap.logLike.[K]
```

3. Zip the above directory before uploading it onto the CEIBA.

Evaluation: We will read your report and compare your answer with the correct answer. (EX: calculate the accuracy of ap.[K].plsa and ap.[K].seqPlsa)

Problem 1: Probabilistic Latent Semantic Analysis (pLSA) .



In this problem, you are going to infer the topic of each word under its context (the document). You will be given the word distribution under topics $\boldsymbol{\theta}_{\mathbf{word|topic}}$ and the topic distribution under documents $\boldsymbol{\theta}_{\mathbf{topic|doc}}$.

- We collected 2250 documents and put them into the corpus file **"ap.txt"**. Each line is a document and there are totally 2250 lines.

- We parsed the documents for you. That is, we filtered out some unimportant words in the documents, and then mapped every remaining keyword into an integer (word ID).

  The mapping is stored in the vocabulary file **"ap.voc"**: the line number of the word is its ID. For example, the word *treason* is in the first line and its ID is 1.

  "ap.fea" is transformed from **"ap.txt"** using the mapping. Each line in the file again represents a document. Assume there are *W* keywords in this article, and the line will be:

  $$\mathbf{[W]\ [ID_1]\ [ID_2]\ \ldots\ldots\ [ID_W]}$$

- The two parameters $\boldsymbol{\theta}_{\mathbf{word|topic}}$、$\boldsymbol{\theta}_{\mathbf{topic|doc}}$ are given in files **"ap.word_topic.[K]"** and **"ap.topic_doc.[K]"**, of the format:

$$\begin{bmatrix} \ln(P(w_1 \mid topic_1)).....\ln(P(w_V \mid topic_1)) \\ ... \\ \ln(P(w_1 \mid topic_K)).....\ln(P(w_V \mid topic_K)) \end{bmatrix} \text{ and } \begin{bmatrix} \ln(P(topic_1 \mid d_1)).....\ln(P(topic_K \mid d_1)) \\ ... \\ \ln(P(topic_1 \mid d_D)).....\ln(P(topic_K \mid d_D)) \end{bmatrix}$$

  where **K** is the number of topics.

Please use Bayes Rule to derive the formula for

P(topic | word, document),

and then generate the conditional probability table. Now for each word in a document, we can calculate the marginal distribution of topics

P(topic | the word, the document the word belongs to)

Please output the topic with largest marginal probability for every single word in every document. Output to the file "**ap.topics.[K].plsa**" for K={10, 100}, where each line is:

[W(number of keywords)] [WordID$_1$]/[topicID$_1$] [WordID$_2$]/[topicID$_2$] … [WordID$_W$]/[topicID$_W$]

(Note that all IDs: docID, topicID, wordID start from 1.)

To help you check your program and visualize your result, we provide a tool "**ColorText**":

It generates a Latex file with first 10 documents colored using following command:

java ColorText ap.txt ap.voc ap.topics.[K].plsa [K] ap.color 10

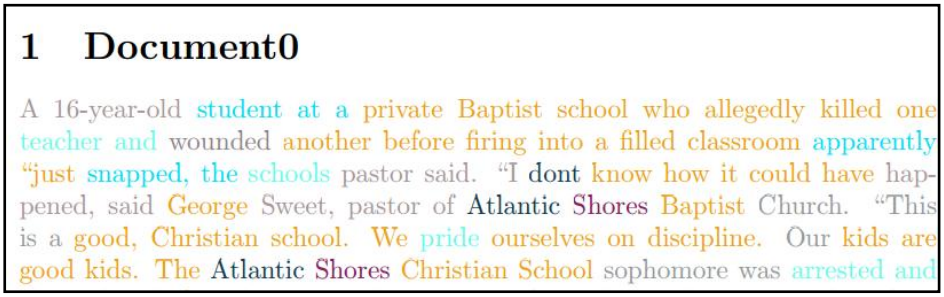
The **ap.color** is the output file.

Please then paste the latex file onto an Online Latex Compiler:

(Please upload only 10 documents; otherwise, it may crashes.)

http://latex.informatik.uni-halle.de/latex-online/latex.php (click the English flag)

You will get a PDF output like:

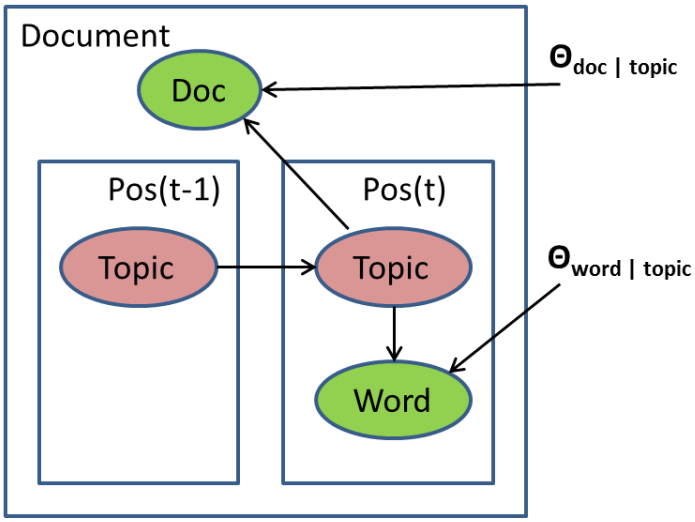


1 Document0

A 16-year-old student at a private Baptist school who allegedly killed one teacher and wounded another before firing into a filled classroom apparently "just snapped, the schools pastor said. "I dont know how it could have happened, said George Sweet, pastor of Atlantic Shores Baptist Church. "This is a good, Christian school. We pride ourselves on discipline. Our kids are good kids. The Atlantic Shores Christian School sophomore was arrested and

Finally, please submit the PDF output "**ap.[K].plsa.pdf**", the file "**ap.topic.[K].plsa**" and your source code.

Problem 2: Sequential pLSA



The pLSA doesn't look into the order of words in the document. However, the order information is important because words nearby are usually related. Therefore, here is a new model as shown in the figure, where the **topic[t]** of a given **word[t]** is related to its previous **topic[t-1]** by the conditional distribution. This is essentially to connect sequential topics in pLSA, but note that we use the symmetric form of pLSA in which the document-topic link is reversed.

Assume K is the number of topics. $P(\text{topic}[t] \mid \text{topic}[t\text{-}1])$ is defined by a parameter $\delta > 1/K$:

$$P(topic[t] \mid topic[t-1]) = \begin{cases} \delta, & if\ topic[t] = topic[t-1] \\ \dfrac{(1-\delta)}{(K-1)}, & otherwise \end{cases}$$

(a) Define $\mathbf{Z_t}$**=topic[t]**, $\mathbf{w_t}$**=word[t]**, **d=document ID**. Show that the marginal distribution $\mathbf{P(Z_t|w_1, \ldots, w_T,d)}$ can be factorized as:

$$P(Z_t = z \mid w_1,...,w_T,d) = \frac{M_L(Z_t = z)P(d \mid Z_t = z)P(w_t \mid Z_t = z)M_R(Z_t = z)}{C},$$

$$where\ C = \sum_{z=1}^{K} M_L(Z_t = z)P(d \mid Z_t = z)P(w_t \mid Z_t = z)M_R(Z_t = z)$$

$$\begin{cases} M_L(Z_t) = \sum_{Z_1 \ldots Z_{t-1}} \prod_{i=1}^{t-1} P(w_i \mid Z_i)P(d \mid Z_i)P(Z_{i+1} \mid Z_i) \\ M_R(Z_t) = \sum_{Z_{t+1} \ldots Z_T} \prod_{i=t+1}^{T} P(w_i \mid Z_i)P(d \mid Z_i)P(Z_i \mid Z_{i-1}) \end{cases}$$

Also, **C** is the data likelihood $\mathbf{P(w_1, \ldots, w_T,d)}$.

(b) Define K*K matrix **A:**

$$\mathbf{A[i,j]=P(Z_{t+1}=j \mid Z_t=i)},$$

and two K*1 vectors $\mathbf{p_{w(t)}}$, $\mathbf{p_d}$ with $\mathbf{p_{w(t)}[i]=P(w_t|Z_t=i)}$, $\mathbf{p_d[i]=P(d|Z_t=i)}$. Show the $\mathbf{M_L(Z_t)}$ and $\mathbf{M_R(Z_t)}$ can be recursively computed:

$$\begin{cases} M_L(Z_t)^T = (p_{w(t-1)} \otimes p_d \otimes M_L(Z_{t-1}))^T * A = \sum_{Z_{t-1}} P(w_{t-1} \mid Z_{t-1}) P(d \mid Z_{t-1}) P(Z_t \mid Z_{t-1}) M_L(Z_{t-1}) \\ M_R(Z_t) = A^*(p_{w(t+1)} \otimes p_d \otimes M_R(Z_{t+1})) = \sum_{Z_{t+1}} P(w_{t+1} \mid Z_{t+1}) P(d \mid Z_{t+1}) P(Z_{t+1} \mid Z_t) M_R(Z_{t+1}) \end{cases}$$

where $c = a \otimes b$ means $c[i] = a[i] * b[i]$.

(c) A naïve matrix multiplication **A*x** takes $O(K^2)$ time. However, the matrix **A** here possesses special structure:

$$A = \begin{bmatrix} \delta & \varepsilon & \dots & \varepsilon \\ \varepsilon & \delta & \dots & \varepsilon \\ \dots & \dots & \dots & \dots \\ \varepsilon & \varepsilon & \dots & \delta \end{bmatrix}, \quad where\ \varepsilon = \frac{1-\delta}{K-1}$$

Show there is an O(K) matrix multiplication algorithm computing **Ax**.
(Hint: Note **A** can be separated into a **diagonal** matrix and a **rank-1** matrix.)

(d) Use the same dataset **"ap.fea"**, **"ap.voc"** as Problem 1. Parameters $\boldsymbol{\theta}_{\mathbf{word|topic}}$ 、 $\boldsymbol{\theta}_{\mathbf{doc|topic}}$ are in files **"ap.word_topic.[K]"** and **"ap.doc_topic.[K]"**:

$$\begin{bmatrix} \ln(P(w_1 \mid topic_1)) \dots \dots \ln(P(w_V \mid topic_1)) \\ \dots \\ \ln(P(w_1 \mid topic_K)) \dots \dots \ln(P(w_V \mid topic_K)) \end{bmatrix} \quad \begin{bmatrix} \ln(P(d_1 \mid topic_1)) \dots \dots \ln(P(d_D \mid topic_1)) \\ \dots \\ \ln(P(d_1 \mid topic_K)) \dots \dots \ln(P(d_D \mid topic_K)) \end{bmatrix}$$

Note the former is the same as problem 1 but the latter is not. Create sequential pLSA model described in (a)-(c), to infer the marginal topic distribution $\mathbf{P(Z_t|w_1 \sim w_T)}$ for each word in each document.

1. Please submit the marginally most likely topic of each word in file "**ap.topic.[K].seqPlsa**" using the same format as in Problem 1, with **K={10, 100}** and δ **=0.9**. Use "ColorText" tool and online Latex compiler to generate a PDF file "**ap.[K].seqPlsa.pdf**" containing the first 10 documents as in Problem 1.

2. For **K={10, 100}**, report the data log-likelihood $\sum_{\mathbf{d}} \mathbf{ln\ P(w_1 \sim w_T, d)}$ in file **"ap.logLike.[K]"** (only 1 number in it), which measures how well your model explains the training corpus.