10-708 Probabilistic Graphical Models

Homework 3

Due Apr 13, in class

Rules:

1. Homework is due on the due date in the class on April 13. Please see course website for policy on late submission.
2. We recommend that you typeset your homework using appropriate software such as LaTeX. If you are writing please make sure your homework is cleanly written and legible. The TAs will not invest undue effort to decrypt bad handwriting.
3. You must hand in a hard copy of the homework. The only exception is if you are out of town in which case you can email your homeworks to `10708-instructor@cs.cmu.edu`. If this is the case, your homework must be typeset using proper software. Please do not email written and scanned copies. Your email must be sent by the beginning of the class on the due date.
4. The submission procedure for the programming component is described along with the corresponding question.
5. You are allowed to collaborate on the homework, but you should write up your own solution and code. Please indicate your collaborators in your submission.
6. Please staple your homeworks.

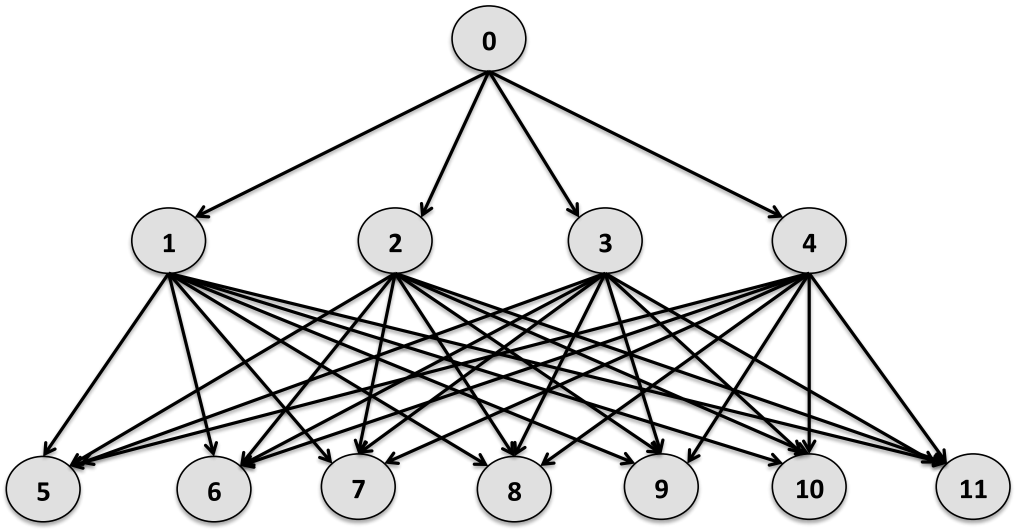

Figure 1: Directed GM for Problem 1. Here 0 . . . 11 are variables representing "IsSummer" . . . "HasFever" as described in Homework 1, Problem 2

1 Exact Inference (40 Points) (Mrinmaya)

In Homework1 (Question 2), you implemented a fully observed directed GM. We explored a host of queries on your model and compared them to the results obtained using the true joint probability distribution. Most of you implemented a brute-force algorithm and were baited by the nemesis of exponential computational complexity. In this homework, you will replace your brute-force algorithm with more principled algorithms you learned in class - Variable Elimination and Message Passing. Consider the Directed GM for the problem shown in Figure 1. Do not use your own GM proposed in homework 1. We will consider the same set of queries as before:

- What is the probability a patient has flu given they are coughing and have a high fever? (Observed Variables: HasFever=true, Coughs=true; Query Variable: HasFlu)
- What is the probability distribution over the symptoms (HasRash, Coughs, IsFatigued, Vomits, and HasFever) given the patient has pneumonia?
- What is the probability of vomiting in summer?

1.1 Variable Elimination (10 Points)

(i) For query 1, show the moralized graph. Choose a good elimination ordering and write down the variable elimination procedure. Show the intermediate factors produced after eliminating each variable. What is the time and space complexity of the variable elimination algorithm?
Note:- Do not substitute numbers for the marginal and conditional probabilities yet. You just have to describe the procedure for a good elimination ordering similar to slides 25-27 in lecture 7.

Note: The true distribution is generated using the model described in Figure 1 but some small noise has been added to it. So your answers will not exactly match the answers using the true distribution but will be pretty close.

1.2 Message Passing on Clique Trees/Factor Graphs (15 Points)

(i) Now, assume that you were instead doing variable elimination on clique trees. For the first query, use the same elimination ordering you chose before. Construct a clique tree. What do the messages here correspond to? Again, you need not substitute any numbers. What is the time and space complexity of the resulting algorithm?

(ii) Now consider, instead, the view of message passing on a factor graph for query 1. What do the factors here correspond to? Write down any 3 messages from variables to factors and any 3 messages from the factors to variables. Again, do not substitute any numbers. What is the time and space complexity of the resulting algorithm?

(iii) Explain what could be the potential benefit of message passing over variable elimination? How would you answer a query that has variables that span beyond any clique/factor in the graph using message passing?

1.3 Implement Variable Elimination or Message Passing (15 Points)

Now, implement one of the above algorithms: you can directly implement the variable elimination procedure or you could implement this using the message passing protocol. If you use message passing, you can use the clique tree view or the factor graph view. Your implementation should be general i.e. your implementation should be able to handle any valid query. Use the data provided in homework 1 to estimate the marginals $P(X_i)$ for nodes that do not have any parents (in our case node 0 only) and the conditionals $P(X_i|\pi(X_i))$ for all other nodes X_i ($\pi(X_i)$ represents the set of parents of X_i). You can validate your implementation using the three queries described above. Report the final answers for all the three queries and run times. Compare the answers and run times with the brute-force exponential computation that you performed in homework 1. Also compare the result and runtime to the case when you simply used the samples to answer the queries. Report all your results in your assignment document and mail your code to `pgm.asst.2015@gmail.com`.

2 Variational Inference (50 Points) (Pengtao)

In this section, through a defined model, we will learn how to perform posterior inference of latent variables and learning of model parameters.

2.1 HMM-LDA Model

The standard Latent Dirichlet Allocation model ignores the ordering of words in a document and assumes words are generated independently given the document-topic proportion vector. While this assumption facilitates computational efficiency, it is unrealistic in practice. The ordering of words carries important semantics and cannot be simply ignored. Oftentimes, the topic assignment of word w_i at position i strongly relies on the topic assignment of word w_{i-1} at position $i-1$. To solve this problem, we define a HMM-LDA model which integrates Hidden Markov Model (HMM) and Latent Dirichlet Allocation (LDA) to incorporate word ordering into topic modeling. The model is shown in Figure 2.

We assume there are D documents in total. Each document d is an ordered sequence of N_d words $w_1, w_2, \cdots, w_{N_d}$. We assume there are K topics in the document corpora. Each topic has a multinomial distribution β over the V words in the vocabulary. These topics can transit among each other with certain transition probabilities. A is the transition probability matrix where A_{ij} denotes the probability to transit from topic i to topic j. Each document has a topic proportion vector θ and each word in the document is assumed to be generated from a topic denoted by z. In HMM-LDA, the topic z_n at position n can be either generated from the topic proportion vector θ or transited from topic z_{n-1} at position $n-1$. To make this binary decision, we introduce a binary variable δ_n for each position n. If $\delta_n = 1$, z_n is generated from θ; if $\delta_n = 0$, z_n is transited from z_{n-1}. These binary variable are generated from a Bernoulli distribution parametrized by ω. For the first word w_1, we assume its topic is always generated from the topic proportion

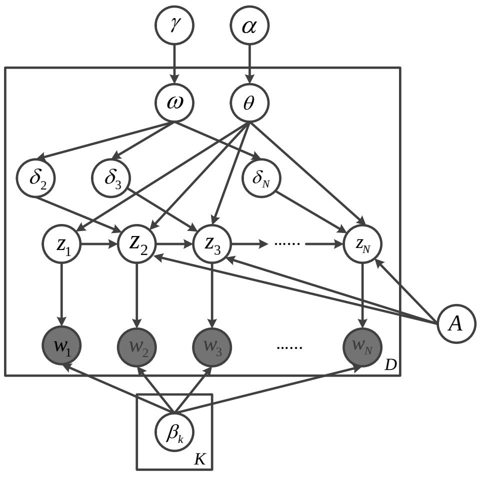

Figure 2: HMM-LDA Model

vector θ. Each document has a document-specific ω which is drawn from a Beta distribution parametrized by γ.

The generative process of this model is described as follows. For each document

- Sample $\omega \sim Beta(\gamma)$
- Sample $\theta \sim Dirichlet(\alpha)$
- For the first word w_1
  - Sample $z_1 \sim Multinomial(\theta)$
  - Sample $w_1 \sim Multimomial(\beta_{z_1})$
- For word w_i at positiion $i = 2, 3, \cdots, N$
  - Sample $\delta_i \sim Bernoulli(\omega)$
  - If $\delta_i = 1$
    * Sample $z_i \sim Multinomial(\theta)$
    * Sample $w_i \sim Multimomial(\beta_{z_i})$
  - If $\delta_i = 0$
    * Sample $z_i \sim Multinomial(A_{z_{i-1}})$
    * Sample $w_i \sim Multimomial(\beta_{z_i})$

We use variational EM to approximate the posterior of latent variables and learn model parameters. To do this, a mean field variational distribution needs to be defined, which is parameterized by some parameters called variational parameters. The variational EM algorithm iteratively performs two steps: 1) in the E step, variational parameters are updated; 2) in the M step, model parameters are optimized.

1. Latent variables and model parameters (2 Points)
   Identify the latent variables which you are going to define variational distribution over and the model parameters which are you going to optimize in the M step.

2. Variational distribution (3 Points)
   Define your variational distribution

3. Joint distribution (2 Points)
   Write down the joint distribution of all random variables (including the latent ones and observed ones)

4. Variational lower bound (8 Points)
   Derive the variational lower bound

5. Update variational parameters (8 points)
   Derive the update equations of variational parameters

6. Update model parameters (8 points)
   Derive the update equations of model parameters

7. Implement this algorithm (15 points)
   Any programming language is acceptable. A Wikipedia dataset is available on the class website. Please refer to the readme.txt file for detailed descriptions of the dataset. In the experiment, set the number of topics to 10. Mail the code to `pgm.asst.2015@gmail.com`.

8. Topic visualization (4 points)
   To visualize each learned topic β, select the top 10 words what correspond to the largest 10 values in β. Report these words in the writeup.

3 Markov Chain Monte Carlo (40 Points) (Xun)

3.1 Sampling Basics (10 Points)

1. A simple sampling method adopted by many of the standard math libraries is the *inverse probability transform*: draw $u \sim \text{Unif}\,(0,1)$, then draw $x \sim F^{-1}(u)$, where F^{-1} is the inverse of the cdf. Show that x generated by this procedure follows distribution F. What is the drawback of this method?

2. Show that if both transition kernels K_1 and K_2 have $p(\cdot)$ as stationary density, so do K_1K_2 and $\lambda K_1 + (1-\lambda)K_2$ for any $\lambda \in [0,1]$. In practice, the former corresponds to sampling from K_1 and K_2 *cyclically* and the latter draws either K_1 with probability λ or K_2 otherwise. Although it is not required to show, extension to more than 2 kernels should be straightforward.

   In the continuous case, the cyclic kernel can be defined as composition of functions:

$$(K_1 \circ K_2)(x,z) = \int K_2(x,y)K_1(y,z)\,\mathrm{d}y. \tag{1}$$

3. Recall MH sampling for target distribution $p(x)$ using proposal $q(x|y)$: at state s, first draw $t \sim q(t|s)$, then accept t with probability

$$A = \min\left(1, \frac{\widetilde{p}(t)q(s|t)}{\widetilde{p}(s)q(t|s)}\right), \tag{2}$$

   where $\widetilde{p}(x)$ is the unnormalized target distribution. Show that $p(x)$ is the stationary distribution of the Markov chain defined by this procedure.

   Consider both continuous and discrete cases.

4. Recall Gibbs sampling for target distribution $p(\mathbf{x}) = p(x_1,\ldots,x_d)$: for each $j \in \{1,\ldots,d\}$, draw $t \sim p(x_j|\text{rest})$ and set $x_j = t$. Show that $p(\mathbf{x})$ is the stationary distribution of the Markov chain defined by this procedure.

3.2 Gibbs Sampling for SVM (30 Points)

Consider the binary SVM for $(\mathbf{x}_i, y_i), i = 1, \ldots, n$, $\mathbf{x}_i \in \mathbb{R}^d$, $y_i \in \{+1, -1\}$:

$$\begin{aligned} \underset{\mathbf{w},\boldsymbol{\xi}}{\text{minimize}} \quad & \frac{\lambda}{2}\|\mathbf{w}\|_2^2 + 2\sum_{i=1}^n \xi_i && (3) \\ \text{subject to} \quad & y_i \mathbf{w}^\top \mathbf{x}_i \geq 1 - \xi_i, \quad \forall i && (4) \\ & \xi_i \geq 0, \quad \forall i. && (5) \end{aligned}$$

Note that the constant 2 on the slack variable does not alter the problem since it can be absorbed into the regularization parameter λ.

1. Rewrite (3) as MAP estimate of a probabilistic model.
2. Instead of MAP, we can estimate the full posterior with the help of auxiliary variables. Although it might sound counterintuitive, augmenting the posterior to a higher dimension leads to an easier problem in this case. In particular, derive the posterior over augmented variables $(\mathbf{w}, \boldsymbol{\gamma})$, $\boldsymbol{\gamma} = \{\gamma_1, \ldots, \gamma_n\}$, using the following integral identity

   $$e^{-2\max(0,u)} = \int_0^\infty \phi(u; -\gamma, \gamma)\,\mathrm{d}\gamma, \tag{6}$$

   where $\phi(x; \mu, \sigma^2)$ denotes the Gaussian density with mean μ and variance σ^2.
3. Derive the conditional distribution $p(\gamma_i|\text{rest})$.
   A useful fact: If $X \sim \mathcal{GIG}(a, b, \rho)$[1] with $\rho = \frac{1}{2}, a = \lambda$, and $b = \frac{\lambda}{\mu^2}$, then $X^{-1} \sim \mathcal{IG}(\mu, \lambda)$.
4. Derive the conditional distribution $p(\mathbf{w}|\text{rest})$. Notice that this corresponds to *block* Gibbs sampling.
5. Implement the Gibbs sampler and run it on `svmdata.mat`.
   - The dataset contains a training set `(X, y)` and a test set `(XX, yy)`. Data points are stored column-wise, *e.g.*, `X` is a $d \times n$ matrix.
   - For reproducibility, please set the random seed to zero.
   - A simple sampling algorithm for inverse Gaussian can be found in the Wikipedia article[2].
   - Many standard library routines (*e.g.*, `mvnrnd`) take covariance matrices as input, however you may find that only precision matrix is available. Try to avoid inverting the precision matrix by reinventing `mvnrnd`. Make use of the fact that if $\mathbf{z} = (z_1, \ldots, z_d)^\top$ with each $z_j \sim \mathcal{N}(0, 1)$, then $\boldsymbol{\mu} + \mathbf{A}\mathbf{z} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, where $\mathbf{A}$ is any real matrix that satisfies $\mathbf{A}\mathbf{A}^\top = \boldsymbol{\Sigma}$. Write down your solution in the hard copy.
   - Discard samples from the first B iterations and average samples from $B + 1$ to T to approximate the posterior mean.
   - Set $\lambda = 1, B = 400, T = 600$. Initialize $\gamma_i = 0$ and $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \lambda\mathbf{I})$. Plot the objective function value (computed using the last sample) on the y-axis and iteration on the x-axis. Also plot the objective function computed using the averaged samples[3] after burn-in.
   - Report the test accuracy, using 1) the last sample at $T = 600$; and 2) the average of all samples after burn-in.
   - Attach the figures and results in the hard copy. Mail the code to `pgm.asst.2015@gmail.com` as usual.

[1] http://en.wikipedia.org/wiki/Generalized_inverse_Gaussian_distribution
[2] http://en.wikipedia.org/wiki/Inverse_Gaussian_distribution
[3] http://en.wikipedia.org/wiki/Moving_average