IDS576: Assignment 2

Due date: Mar 29 (5.00 PM CT)

Turn in solutions as pdf(s) (see template) and ipynb file(s) on the course website (for instance, as a single zip file).

Note: Answer the following questions in complete sentences and with full clarity. Since this is a graduate class, please use any assumptions necessary to answer the questions satisfactorily. As in real life, use your judgment to figure out what conditions to assume while answering. Answering too little or too much will garner point deduction. The answer to 'should I include this in my answer' is almost always a yes, especially if it completes your answer to the question asked. If in doubt, use the discussion board on the course website. Across group collaboration is not allowed. Cite all your sources (see the Syllabus sheet for these and other pointers).

1 Embeddings (25pt)

Instead of embedding words, we will embed movies. In particular, if we can embed movies, then similar movies will be close to each other and can be recommended. This line of reasoning is analogous to the *distributional hypothesis of word meanings* (see `https://en.wikipedia.org/wiki/Distributional_semantics`). For words, this roughly translates to words that appear in similar sentences should have similar vector representations. For movies, vectors for two movies should be similar if they are watched by similar people.

Let the total number of movies be M. Let $X_{i,j}$ be the number of users that liked both movies i and j. We want to obtain vectors $v_1, ..., v_i, ..., v_j, ..., v_M$ for all movies such that we minimize the cost $c(v_1, ..., v_M) = \sum_{i=1}^{M} \sum_{j=1}^{M} \mathbf{1}_{[i \neq j]} (v_i^T v_j - X_{i,j})^2$. Here $\mathbf{1}_{[i \neq j]}$ is a function that is 0 when $i = j$ and 1 otherwise.

1. Compute data $X_{i,j}$ from the attached csv files [1]. In movieratings.csv, we have 943 users rate 1682 movies generating 100000 observations. Each row in the csv file is movie_id, user_id and rating (1 implies user likes and 0 implies user dislikes). File movies.csv maps movie_ids to the actual names.

2. Optimize function $c(v_1, ..., v_M)$ over $v_1, ..., v_M$ using gradient descent. Do this for two different starting parameters: (a) when all the vectors are zeros, and (b) when each coordinate of each vector is i.i.d random uniform between $[-0.7, 0.7]$. Plot the loss as a function of iteration for both settings.

3. Recommend top 10 movies (not vectors or indices but movie names) given movie 'Aladdin'. Describe your recommendation strategy.

4. Recommend top 10 movies given movies 'Toy Story' and 'Home Alone'. Describe your recommendation strategy.

[1] courtesy Arora and Hazan, COS402.

2 RNNs (25pt)

Note: This problem is quite open-ended, so make any assumptions as necessary.

1. Pick a text corpus or use the provided one (text_corpus.txt[2]).
2. Follow the instructions from this page to set up a twitter bot (up to reading tweets from a file).
3. Train an RNN (e.g., LSTM) based character level language model (say using Keras/Tensorflow) from the corpus.
4. Output the character strings (≤ 140) generated by the RNN model to the twitter bot and make it tweet 20 times (1 per 6 minutes).
5. Remove your twitter credentials from the python code and submit your implementation and the name of the twitter bot you deployed.
6. Also describe the model and implementation details. Show training performance as a function of training iterations.
7. (Bonus) Train an n-gram character level language model and repeat the above steps.

3 Factorization I (6pt)

Let A be a random variable (RV) with support $\{0, 1, 2\}$. Similarly, let B, C and D be random variables with supports $\{0, 1\}$, $\{1, 2, 3\}$ and $\{10, 20\}$.

1. Write down the joint distribution of A, B, C and D in a factored form. How many numbers (parameters) are needed to fully specify this joint distribution? Write down all factorizations that are possible for $P(A, B, C)$ and $P(A, B)$.
2. If we know that $P(A|B, C, D) = P(A|B)$ and $P(C|D) = P(C)$, then what is the number of parameters needed to specify the join distribution $P(A, B, C, D)$?
3. If we know that $P(B, C, D)$ respects DAG $B \rightarrow C \rightarrow D$, then does it imply $P(C|B, D) = P(C|B)$?
4. If we know that $P(A, B, C, D) = P(A)P(B)P(C)P(D)$, how many parameters are needed to represent the joint distribution?

4 Factorization II (6pt)

Let X_i for $i = 1, 2, 3$ be an indicator random variable for the event that a coin toss comes up heads (happening with some probability p). Assume X_i are independent. Let $Z_4 = X_1 \oplus X_2$ and $Z_5 = X_2 \oplus X_3$ where $\oplus$ denotes the XOR (exclusive OR, see https://en.wikipedia.org/wiki/Exclusive_or) operation.

[2] From Andrej Karpathy's char-rnn repo https://github.com/karpathy/char-rnn

1. Show the computations of the following: $P(X_2, X_3|Z_5 = 0)$ and $P(X_2, X_3|Z_5 = 1)$.
2. Draw a DPGM and write down the corresponding conditional probability tables. What independence relationships are captured by the DPGM?
3. Draw a UPGM and write down the corresponding factors/potentials. What independence relationships are captured by the UPGM?
4. Under what conditions on p would $Z_5 \perp X_3$ and $Z_4 \perp X_1$? Are these independences captured by the above two graphs? Explain.

5 Naive Bayes (15pt)

Let $i = 1, ..., D$ index words in a dictionary. Let X_i be a random variable representing the presence (1) or absence (0) of the i^{th} word of the dictionary in a document (such as an email). Let the document be classified as confidential ($Y = 1$) or not-confidential ($Y = 0$). Let the documents be drawn according to the distribution $P(Y, X_1, ..., X_D)$.

1. Assuming words are conditionally independent of each other given Y, write the form of the joint distribution.
2. Express $P(Y = 1|x_1, ..., x_D)$ in terms of $P(x_i|Y = 1)$ using Bayes rule. Can you relate this to logistic regression?
3. Describe how one can estimate the parameters of the distribution.
4. Describe if the independence assumption made is reasonable or not.

6 D-separation (8pt)

Let $A = \{X_2\}, B = \{X_3, X_5\}$ and $C = \{X_1, X_6\}$. Let the DPGM be as shown in Figure 1. Is $A \perp B \mid C$? Justify your answer.

7 Inference on DPGM (10pt)

Consider the DPGM in Figure 2 that represents a maintenance sensor network for a machine that manufactures two goods. Each G_i represents the health of a component in a machine. $G_i = 1$ if the component is running and $G_i = 2$ if the component failed. G_1 is the common component needed for both goods whereas G_2 and G_3 are specific the goods. Also, G_2 and G_3 can be influenced by the failure of G_1. X_i is a continuous random variable that measures the quantity of each good type produced by the machine, which is high if the component is running and low if it is not. The conditional probability

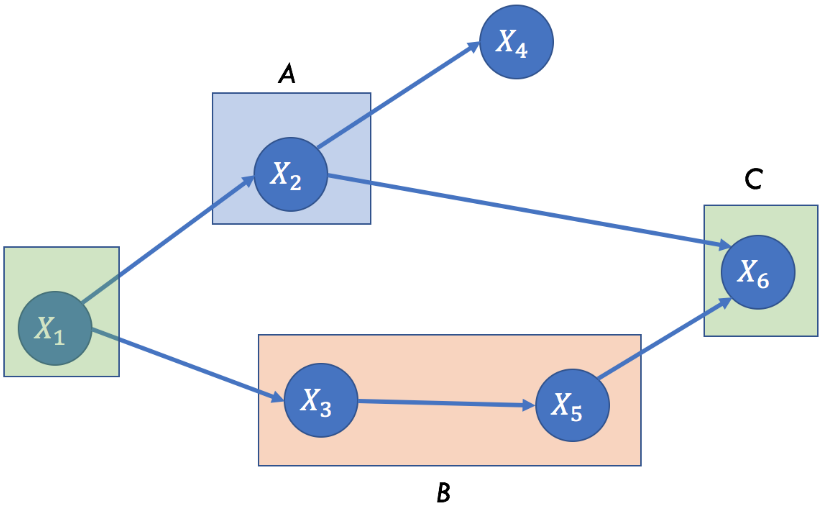

Figure 1: DPGM for Question 6.

distributions are:

$$\begin{aligned} P(G_1) &= [1/2, 1/2] \\ P(G_i = G_1|G_1) &= 0.8,\ i \in \{2, 3\} \\ P(X_i|G_i = 1) &= \mathcal{N}(X_i|\mu = 100, \sigma^2 = 10) \\ P(X_i|G_i = 2) &= \mathcal{N}(X_i|\mu = 10, \sigma^2 = 20) \end{aligned}$$

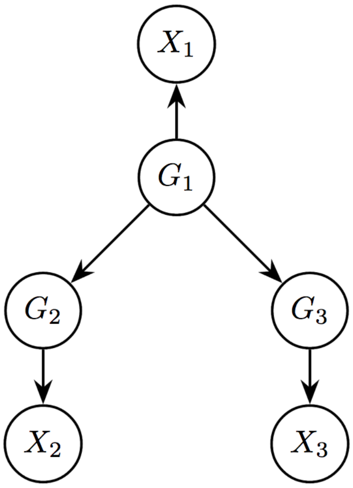

Figure 2: DPGM for Question 7.

1. If we observe $X_2 = 100$, what is the posterior belief on G_1. That is, compute $P(G_1|X_2 = 100)$, and show your work.

2. If both X_2 and X_3 are observed, then what is $P(G_1|X_2, X_3)$? In particular, what are the values when the observations are: (a) $X_2 = 100$ and $X_3 = 100$, (b) $X_2 = 10$ and $X_3 = 100$, and (c) $X_2 = 10$ and $X_3 = 10$. Explain your answers.

8 Belief Propagation Implementation (25pt)

Implement the Sum-Product version of Belief Propagation in Python (using the networkx package to represent the factor graph) to compute the marginal distribution $P(A = a, B = b)$ $\forall a, b$ for the factor graph shown in Figure 3. That is, use the graph object from networkx to define the factor graph and implement the Sum-Product algorithm to pass messages. The support of the corresponding random variables A, B, C, D and E are $\{1, 2\}$. The factors are as follows:

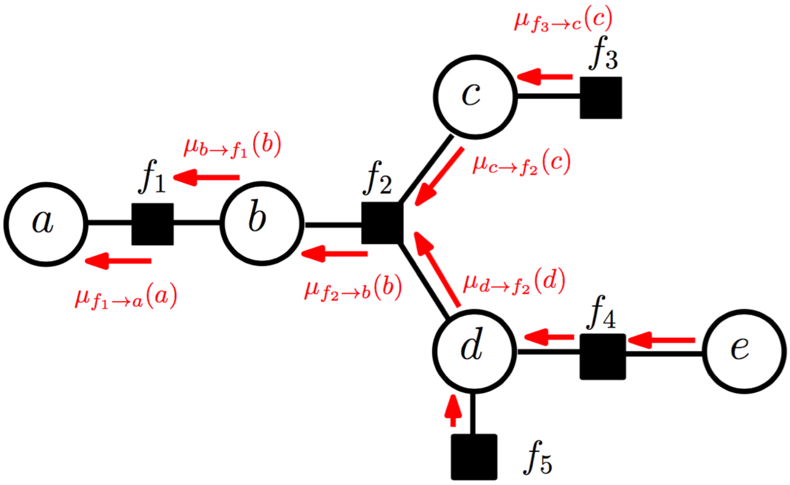

Figure 3: Factor Graph for Question 8.

1. $f_1(a, b) = a * b$ (for example, $f_1(a = 1, b = 2) = 1 * 2 = 2$).
2. $f_2(b, c, d) = 2 * (5 - b * c) - d + 1$.
3. $f_4(d, e) = d * e$.
4. $f_3(c) = 3 - c$.
5. $f_5(d) = 3 - d$.

Report the marginal distribution (plot/table) as well as your implementation (py/ipynb). Additionally,

1. Briefly describe how you implemented the algorithm (data structures and code organization).
2. What is the complexity (number of additions and multiplications if any) of computing an outgoing message from a variable node given that it is connected to F factors and has a support of k values?

3. Similarly, what is the complexity of computing an outgoing message from a factor node given it is connected to V variables each of which have a support of k values.?

9 Sampling (20pt)

Let $X \sim \pi$ be a distribution. To create a Monte Carlo estimate of the expectation of some function of X, i.e., $E_\pi[f(X)]$, we will do Metropolis-Hastings (MH) MCMC sampling. For this, we start with an initial $x = x_0$ and do the following:

- $x' \sim g(\cdot|x)$.
- $\alpha = \min(1, \frac{\pi(x)g(x'|x)}{pi(x')g(x|x')})$.
- With probability α accept x' as the next sample x_t.
- Set $x = x'$ and repeat.

An example of the proposal distribution is $g(x'|x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x'-x)^2}{\sigma^2}}$.

1. Implement the MH sampler for estimating $E[(X^2 + 10)x]$ when $\pi(x) = \frac{1}{\sqrt{2\pi}} e^{-(x-3)^2}$. Choose a suitable proposal distribution.
2. Plot the estimate as a function of the number of samples used in the estimation.

10 Estimation with Partially Observed Data (15pt)

Consider learning the DPGM $A \to B \to C$, with the following data table where entries z_1 and z_2 are missing at random.

A	B	C
T	T	F
F	F	F
F	T	F
z_1	F	T
F	T	T
T	F	F
T	z_2	F

1. Estimate the initial parameters of the network using maximum likelihood estimation.
2. Show the calculations by hand for two iterations of the Expectation Maximization algorithm for this problem. That is, estimate the values of the missing data , use these to reestimate the parameters of the DPGM. Again, estimate the values of the missing data using this updated model, and finally update the parameters of the model with these updated estimates of missing data.