Knowledge Discovery and Management

Problem Set 5

Name: Rakesh Reddy

Class ID: 20

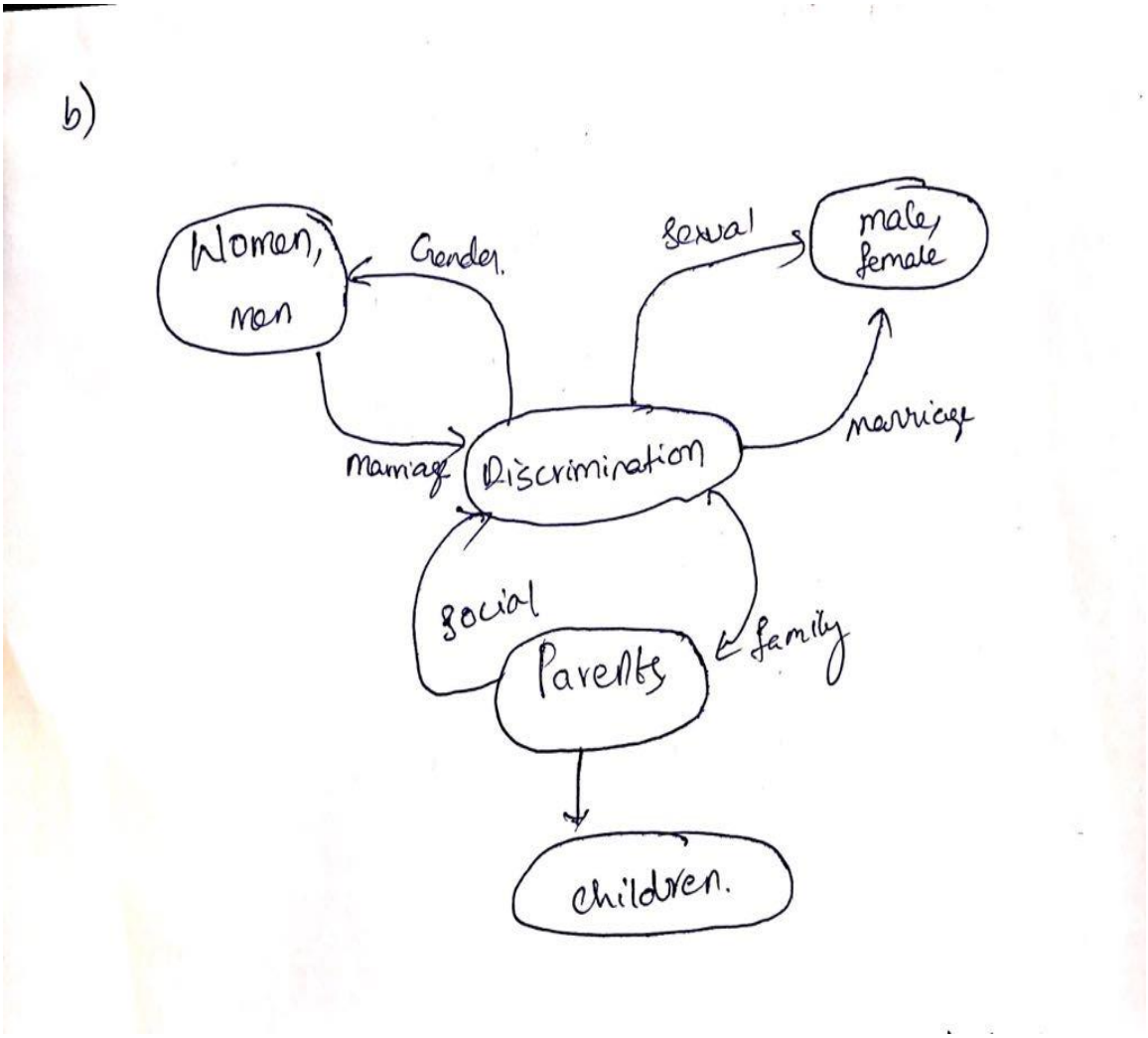
1.Ans : LDA

A. **The overall process to generate such topics from the corpus.**

LDA is the iterative model it need 3 params which means no.of topics and deep prior knowledge of the dataset

- Decide the number of topics informally by analyzing each word.
- Assign every word to a temporary topic.
- Check and update the assigned topic based on prevalence of that word across the topic and prevalence of that topic across the document. It means pick a random word and assign it to more likely topic. In many case the word, which remained after assigning words to topics, has the same change to be taken by other topics. So, that word may be assigned to some other topic. This iterative updating is the key feature of LDA that generates a final solution with coherent topics.
- LDA Algorithm:
- Input: Words W Edocuments d
- Output: topic assignments Z and counts $n_{d,k}$, n and n_k

B. A knowledge graph for Topic 3 in Yale Law Journal

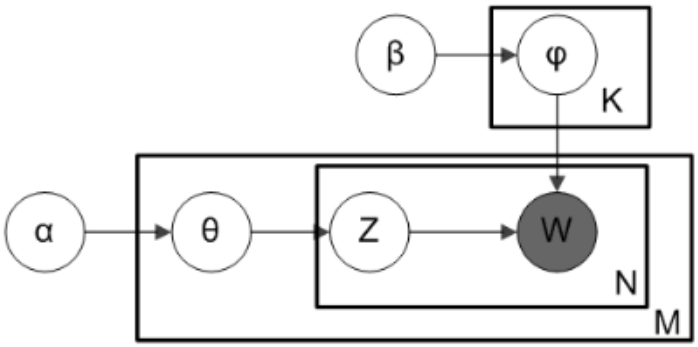


C.

To calculate the generality and specificity,

- First assign words to topics
- Get the most common words related to that topics by calculating the number of occurrence of that word.
- For any topic, you should have a set of library upon which you can take decision which is used for which word will specifically go to which topic.

D. The inference algorithm



According to figure, we have k number of topics. Parameter β defines the per-topic word distribution, which is given to ϕ(k) that is word distribution for topic k.

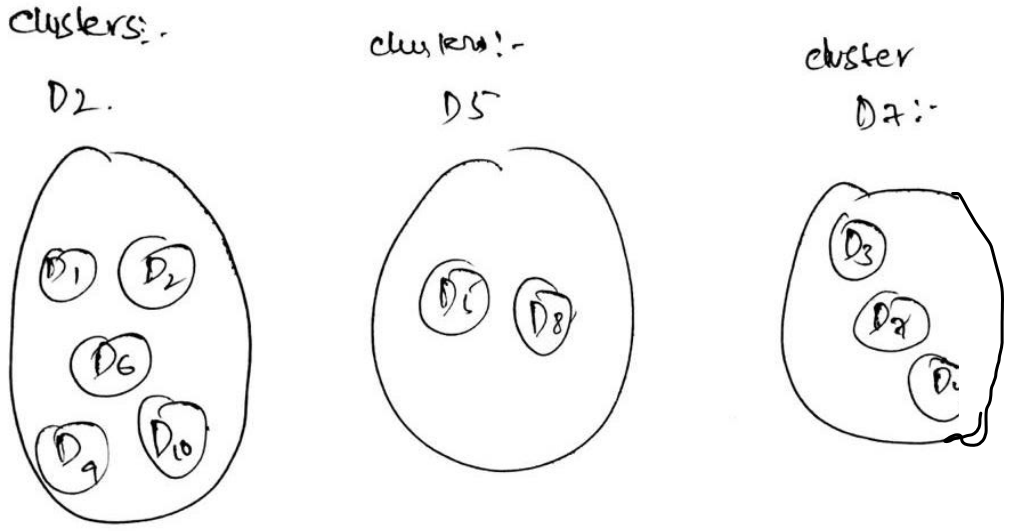
M indicates number of documents to be analyzed and N indicates number of words in the document. Using α, we can assign the Dirichlet-prior concentration parameter of the per-document distribution. At document level, after defining α, topic distribution is done using θ. Now, for taken word from the selected document is assigned to topic using ϕ(k) and z(i,j).

2.Ans

- There are 10 documents/mentions and 5 unique words. The document term matrix shows for how many times one word has appeared in the document. For example, in document 1 (D1), the words *online, book* and Delhi have each been mentioned once.
- We want to create K=3 clusters. First, three seeds should be chosen. Suppose, D2, D5 & D7 are chosen as initial three seeds.
- The next step is to calculate the Euclidean distance of other documents from D2, D5 & D7.
- Assuming: U=Online, V= Festival, X=Book, Y=Flight, Z=Delhi. Then the Euclidean distance between D1 & D2 would be:

  ((U1-U2)^2 + (V1-V2)^2+(X1-X2)^2+ (Y1-Y2)^2+(Z1-Z2)^2)^0.5

Clusters	# of Observations
D2	5
D5	2
D7	3



- Hence, 10 documents have moved into 3 different clusters. Instead of Centroids, Medoids are formed and again distances are re-calculated to ensure that the documents who are closer to a medoid is assigned to the same cluster.
- Medoids are used to build the story for each cluster.

B. Difference between K means and LDA

Both K-means and Latent Dirichlet Allocation (LDA) are unsupervised learning algorithms, where the user needs to decide a priori the parameter K, respectively the number of clusters and the number of topics.

If both are applied to assign K topics to a set of N documents, the most evident difference is that K-means is going to partition the N documents

in K disjoint clusters (i.e. topics in this case). On the other hand, LDA assigns a document to a mixture of topics. Therefore each document is characterized by one or more topics (e.g. Document D belongs for 60% to Topic A, 30% to topic B and 10% to topic E). Hence, LDA can give more realistic results than k-means for topic assignment.

Pros of K-Means:

- Practically work well even some assumptions are broken;
- Simple, easy to implement;
- Easy to interpret the clustering results;
- Fast and efficient in terms of computational cost, typically O(K*n*d);

Limitations of K-Means:

- Clustering non-clustered data: Run k-means on uniform data, and you will *still* get clusters, which leads the research to dead end.
- Sensitive to scale: Rescaling your datasets will completely change results.
- While you can run k-means on binary data (or one-hot encoded categorical data), the results will not be binary anymore. So you do get a result out, but you may be unable to interpret it in the end, because it has a different data type than your original data.

Limitations of LDA:

- One major limitation is perhaps given by its underlying unigram text model: LDA doesn't consider the mutual position of the words in the document. Documents like "Man, I love this can" and "I can love this man" are probably modelled the same way. It's also true that for longer documents, mismatching topics is harder. To overcome this limitation, at the cost of almost square the complexity, you can use 2-grams (or N-grams) along with 1-gram.
- Another weakness of LDA is in the topics composition: they're overlapping. In fact, you can find the same word in multiple topics (the example above, of the word "can", is obvious). The generated topics, therefore, are not independent and orthogonal like in a PCA-decomposed basis, for example.

This implies that you must pay lots of attention while dealing with them (e.g. don't use cosine similarity).

- For a more structured approach - especially if the topic composition is very misleading - you might consider the hierarchical variation of LDA, named H-LDA, (or simply Hierarchical LDA). In H-LDA, topics are joined together in a hierarchy by using a Nested Chinese Restaurant Process (NCRP). This model is more complex than LDA, and the description is beyond the goal of this blog entry, but if you like to have an idea of the possible output, here it is. Don't forget that we're still in the probabilistic world: each node of the H-DLA tree is a topic distribution.