CS 440 Project: Cross-Collection Mixture Model

Student: Chirag C. Shetty (cshetty2)

Paper: ChengXiang Zhai, Atulya Velivelli, and Bei Yu. 2004. A cross-collection mixture model for comparative text mining. In Proceedings of the 10th ACM SIGKDD international conference on knowledge discovery and data mining (KDD 2004). ACM, New York, NY, USA, 743-748. DOI=10.1145/1014052.1014150 [link]

Github link: https://github.com/chiragcshetty/CourseProject

Introduction

The paper explores a further improvement like PLSA in mining topics. In PLSA, k topics are mined from the entire collection. However, collection may have subset and we may be interested in knowing the topics within a collection while also comparing across different collections. The paper adds one more level of generative variable (lambda_c) and tries to achieve this.

Data

The original paper from 2004 had used a set on news articles and reviews fro epionions.com. The site is no longer active and the dataset wasn't archived anywhere. So I decided to write a scraper, starting with the codes used in the MP's. I chose CNN, which has a search feature on its webpage. So I scrap the webpage resulting from searching a topic of interest and extract the news articles. This mostly involved handcrafting the extraction process.

Procedure for scraping

The main python file is called scrap.py

1. Edit the 'name' variable to indicate the topic. Files extracted will be stored with this name
2. no_pages: Number of pages to search. Each page has 10 articles
3. Run scrap.py (tested for python3.5), by setting dir_url to a topic search page on cnn webpage
   Example: For example this webpage shows for the search 'election':
   https://www.cnn.com/search?q=election

4. run python (3.5 used) scrap.py. The extracted docs will be stored in the folder 'cnn'
5. You can run it for as many topics as you wish

Baseline model

For baseline, the paper uses the standard PLSA model. Starting with PLSA code from MP3, background model was added. Thus complete PLSA was implemented at plsa_proj.py.

Cross-Collection Mixture Model

The model is implemented at ccmix.py. Following at the EM update equations from the paper

$$
\begin{aligned}
p(z_{d,C_i,w} = j) &= \frac{\pi_{d,j}^{(n)}(\lambda_C p^{(n)}(w|\theta_j) + (1-\lambda_C)p^{(n)}(w|\theta_{j,i}))}{\sum_{j'=1}^{k} \pi_{d,j'}^{(n)}(\lambda_C p^{(n)}(w|\theta_{j'}) + (1-\lambda_C)p^{(n)}(w|\theta_{j',i}))} \\
p(z_{d,C_i,w} = B) &= \frac{\lambda_B p(w|\theta_B)}{\lambda_B p(w|\theta_B) + (1-\lambda_B)\sum_{j=1}^{k} \pi_{d,j}^{(n)}(\lambda_C p^{(n)}(w|\theta_j) + (1-\lambda_C)p^{(n)}(w|\theta_{j,i}))} \\
p(z_{d,C_i,j,w} = C) &= \frac{\lambda_C p^{(n)}(w|\theta_j)}{\lambda_C p^{(n)}(w|\theta_j) + (1-\lambda_C)p^{(n)}(w|\theta_{j,i})} \\
\pi_{d,j}^{(n+1)} &= \frac{\sum_{w \in V} c(w,d)p(z_{d,C_i,w} = j)}{\sum_{j'}\sum_{w \in V} c(w,d)p(z_{d,C_i,w} = j')} \\
p^{(n+1)}(w|\theta_j) &= \frac{\sum_{i=1}^{m}\sum_{d \in C_i} c(w,d)(1 - p(z_{d,C_i,w} = B))p(z_{d,C_i,w} = j)p(z_{d,C_i,j,w} = C)}{\sum_{w' \in V}\sum_{i=1}^{m}\sum_{d \in C_i} c(w',d)(1 - p(z_{d,C_i,w'} = B))p(z_{d,C_i,w'} = j)p(z_{d,C_i,j,w'} = C)} \\
p^{(n+1)}(w|\theta_{j,i}) &= \frac{\sum_{i=1}^{m}\sum_{d \in C_i} c(w,d)(1 - p(z_{d,C_i,w} = B))p(z_{d,C_i,w} = j)(1 - p(z_{d,C_i,j,w} = C))}{\sum_{w' \in V}\sum_{i=1}^{m}\sum_{d \in C_i} c(w',d)(1 - p(z_{d,C_i,w'} = B))p(z_{d,C_i,w'} = j)(1 - p(z_{d,C_i,j,w'} = C))}
\end{aligned}
$$

Procedure:

1. Run scrap.py, by setting dir_url to a topic search page on cnn webpage. Set appropritae variables as described in scrap.py
2. Set N - number of docs of each kind in the collection
3. name_set=list of names of each collection eg: ['elon','bezos']
4. Set number_of_topics
5. Run the code
6. The output displays top_n words in each distribution

Important notes

1) In calculating c(w,d) that count of word w in doc d across all words and docs, smoothing must be applied. No c(w,d) should be exactly 0. Esle it'll cause divison by zero problem. In the code, term_doc_matrix stores c(w,d)

2) In the EM update steps given in the paper, observe the update for P(w/theta j,i) i.e the collection specific word distributions. Since both numerator and denominator are summed over the entire collection, P(w/theta j,i) will not capture features specific to the sub-collections. They will all behave similarly. Hence in implementation, the summations are only taken over the docs in collection concerned

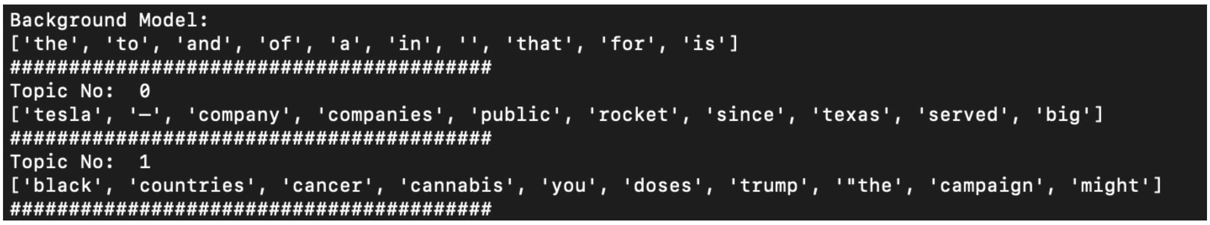
Experiments and Results

To experiment we need related document sub-collections, each of which have a common theme. One good example of such a collection is about famous people in related fields. I chose 'Elon Musk' and 'Bill Gates'. Both are billionaire businessmen, hence there will be similarities in news articles about them. However of-late they are in news for very different reasons. So each sub-collection has its own features. Articles are scrapped from cnn, in order of recency and stored in folder cnn. There are 29 files for each category.

PLSA Baseline

lamba_b is kept at 0.9. Increasing it too much seemed to include informative but words into background model. The word 'tesla' for instance for the chosen dataset. Decreasing lamba_b too much lets stopwords leak into topic distributions. Around 0.9 seemed the right compromise. Number of topics is taken to be 2

PLSA gives the following result:

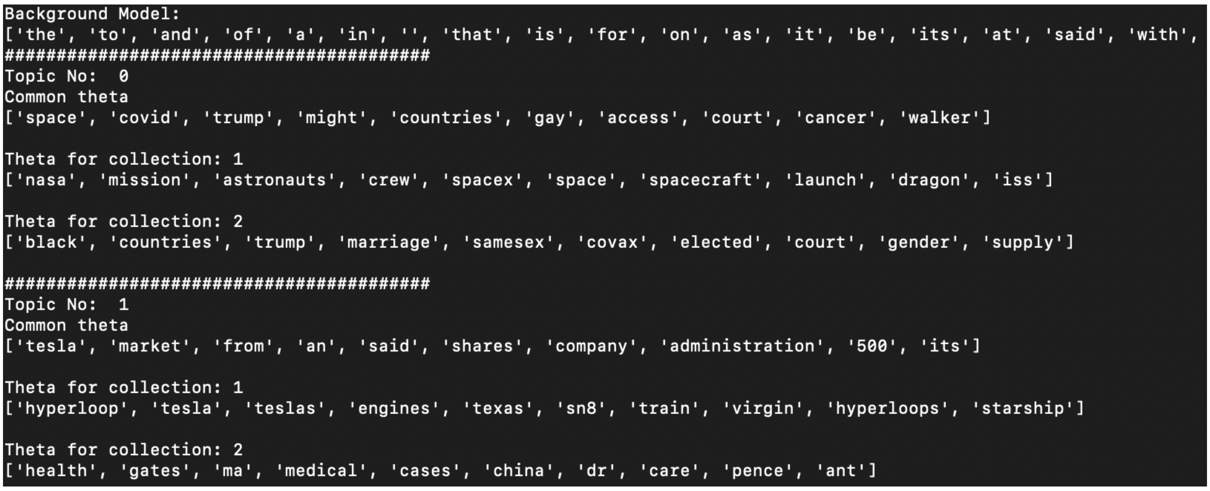


```
Background Model:
['the', 'to', 'and', 'of', 'a', 'in', '', 'that', 'for', 'is']
#########################################
Topic No:  0
['tesla', '—', 'company', 'companies', 'public', 'rocket', 'since', 'texas', 'served', 'big']
#########################################
Topic No:  1
['black', 'countries', 'cancer', 'cannabis', 'you', 'doses', 'trump', '"the', 'campaign', 'might']
#########################################
```

Clearly the topic 0 refers to 'Elon Musk' with words like 'tesla', 'rocket', 'texas' (Musk moving to Texas has been in news a lot recently). Topic 1 is not as clearly associated with Gates. However given Gates' charity work, especially in healthcare, the topic 1 makes sense

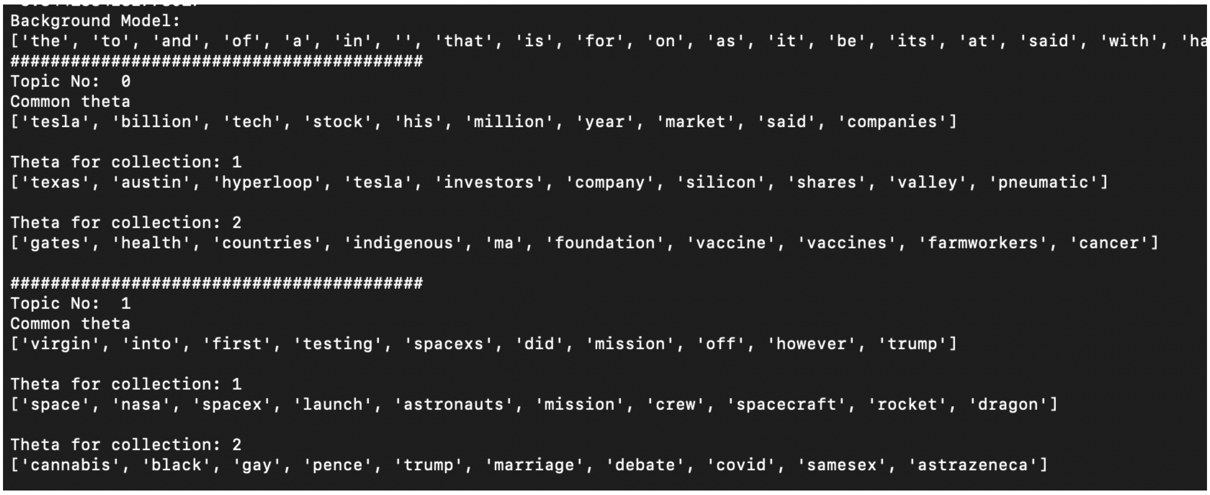
CCM Model

lamba_b was retained at 0.9 and lambda_c was taken as 0.7. With same dataset as above this is the result of CCM:



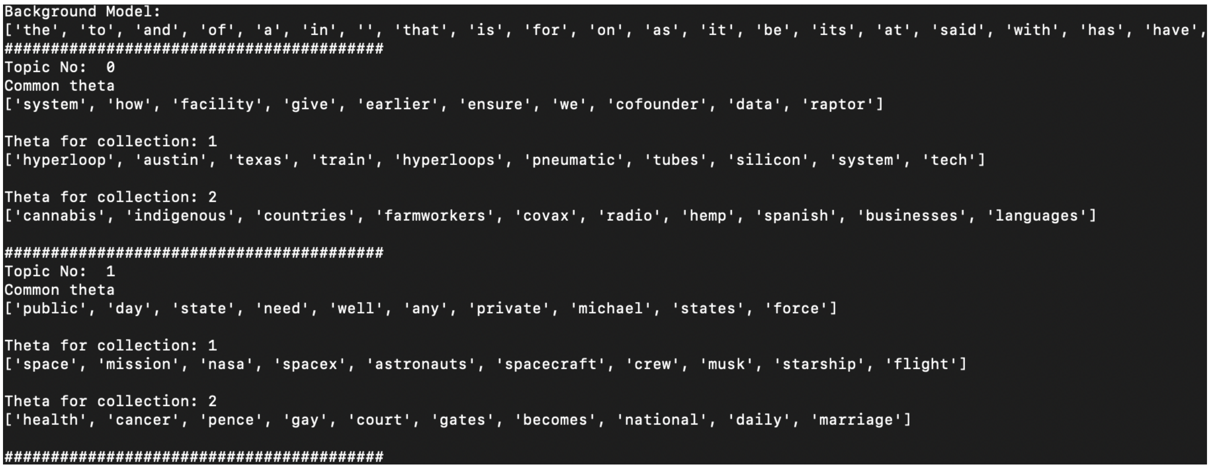
```
Background Model:
['the', 'to', 'and', 'of', 'a', 'in', '', 'that', 'is', 'for', 'on', 'as', 'it', 'be', 'its', 'at', 'said', 'with',
#########################################
Topic No:  0
Common theta
['space', 'covid', 'trump', 'might', 'countries', 'gay', 'access', 'court', 'cancer', 'walker']

Theta for collection: 1
['nasa', 'mission', 'astronauts', 'crew', 'spacex', 'space', 'spacecraft', 'launch', 'dragon', 'iss']

Theta for collection: 2
['black', 'countries', 'trump', 'marriage', 'samesex', 'covax', 'elected', 'court', 'gender', 'supply']

#########################################
Topic No:  1
Common theta
['tesla', 'market', 'from', 'an', 'said', 'shares', 'company', 'administration', '500', 'its']

Theta for collection: 1
['hyperloop', 'tesla', 'teslas', 'engines', 'texas', 'sn8', 'train', 'virgin', 'hyperloops', 'starship']

Theta for collection: 2
['health', 'gates', 'ma', 'medical', 'cases', 'china', 'dr', 'care', 'pence', 'ant']
```

With lamba_c = 0.6



```
Background Model:
['the', 'to', 'and', 'of', 'a', 'in', '', 'that', 'is', 'for', 'on', 'as', 'it', 'be', 'its', 'at', 'said', 'with', 'ha
#########################################
Topic No:  0
Common theta
['tesla', 'billion', 'tech', 'stock', 'his', 'million', 'year', 'market', 'said', 'companies']

Theta for collection: 1
['texas', 'austin', 'hyperloop', 'tesla', 'investors', 'company', 'silicon', 'shares', 'valley', 'pneumatic']

Theta for collection: 2
['gates', 'health', 'countries', 'indigenous', 'ma', 'foundation', 'vaccine', 'vaccines', 'farmworkers', 'cancer']

#########################################
Topic No:  1
Common theta
['virgin', 'into', 'first', 'testing', 'spacexs', 'did', 'mission', 'off', 'however', 'trump']

Theta for collection: 1
['space', 'nasa', 'spacex', 'launch', 'astronauts', 'mission', 'crew', 'spacecraft', 'rocket', 'dragon']

Theta for collection: 2
['cannabis', 'black', 'gay', 'pence', 'trump', 'marriage', 'debate', 'covid', 'samesex', 'astrazeneca']
```

lamba_c= 0.4



```
Background Model:
['the', 'to', 'and', 'of', 'a', 'in', '', 'that', 'is', 'for', 'on', 'as', 'it', 'be', 'its', 'at', 'said', 'with', 'has', 'have',
########################################
Topic No:  0
Common theta
['system', 'how', 'facility', 'give', 'earlier', 'ensure', 'we', 'cofounder', 'data', 'raptor']

Theta for collection: 1
['hyperloop', 'austin', 'texas', 'train', 'hyperloops', 'pneumatic', 'tubes', 'silicon', 'system', 'tech']

Theta for collection: 2
['cannabis', 'indigenous', 'countries', 'farmworkers', 'covax', 'radio', 'hemp', 'spanish', 'businesses', 'languages']

########################################
Topic No:  1
Common theta
['public', 'day', 'state', 'need', 'well', 'any', 'private', 'michael', 'states', 'force']

Theta for collection: 1
['space', 'mission', 'nasa', 'spacex', 'astronauts', 'spacecraft', 'crew', 'musk', 'starship', 'flight']

Theta for collection: 2
['health', 'cancer', 'pence', 'gay', 'court', 'gates', 'becomes', 'national', 'daily', 'marriage']

########################################
```

The difference between Collection 1 and collection 2 specific distribution is stark and informative. They clearly are clustering around Musk's articles and Gates' article. However, the topics themselves do not show much distinction. There is some flavour of topic 0 being about technology while topic 1 is about society and governance. But one can not decisively say so.

It might also be possible that the articles donot have enough variety to force the EM to cluster well. Better data with more latent topics may reveal further benefits of the CCM model. Also, knobs of lambda_c and lambda_b can be optimized further.