Project Documentation

Lin Yutong
Yutong21@illinois.edu

1. Overview

This software is designed to complete the text mining task by a contextual generative model, which is an extension of the PLSA generative model. Several texts from different collections, several parameters (lambda c, lambda b, number of clusters etc.) will be passed in, and the software will produce the matrix of clusters and top k words in each language model (common model, collection-specific models) in the form similar to the original paper shown in Figure 1. The idea is that a word could be generated from a common theme model with lambda c probability, but also has (1 – lambda c) probability to be generated from a collection specific theme model. The probability of a word given the collection is shown in Figure 2, and the mixture model is illustrated in Figure 3.

Table 2: cross-collection mixture model results on War news data

	Cluster1	Cluster2	Cluster3	Cluster4	Cluster5
Common theme words	us 0.042	mr 0.029	killed 0.0361	monday 0.0362	united 0.042
	nation 0.0299	marines 0.0252	month 0.0316	official 0.032	nations 0.04
	will 0.0238	dead 0.023	deaths 0.0231	i 0.029	with 0.03
	action 0.022	general 0.022	one 0.0226	would 0.0279	is 0.025
	re 0.0216	defense 0.019	died 0.0222	where 0.0253	it 0.024
	border 0.0194	key 0.0179	been 0.0218	do 0.0253	they 0.023
	its 0.0171	since 0.0179	drive 0.0178	spokesman 0.022	diplomatic 0.0229
	ve 0.0161	first 0.0158	according 0.0149	political 0.021	blair 0.022
Iraq theme words	god 0.022	iraq 0.022	troops 0.0164	intelligence 0.049	n 0.03
	saddam 0.0157	us 0.021	hoon 0.015	weapons 0.034	weapons 0.0237
	baghdad 0.0129	baghdad 0.0167	sanchez 0.0116	inquiry 0.0278	inspectors 0.0227
	your 0.0124	nato 0.0147	billion 0.01	commission 0.0168	council 0.016
	live 0.01	iraqi 0.0129	spokeswoman 0.008	independent 0.0164	declaration 0.0152
Afghan theme words	paper 0.0205	story 0.028	taleban 0.0259	bin 0.031	northern 0.0404
	afghan 0.019	full 0.026	rumsfeld 0.020	laden 0.031	alliance 0.0398
	meeting 0.0139	saturday 0.016	hotel 0.012	steinberg 0.0268	kabul 0.0297
	euro 0.0121	e 0.015	front 0.0113	taliban 0.0229	taleban 0.0248
	highway 0.0118	rabbani 0.0116	dropped 0.0099	chat 0.0186	aid 0.0197

Figure 1: example output

$$p_d(w|C_i) = (1-\lambda_B)\sum_{j=1}^{k}[\pi_{d,j}(\underbrace{\lambda_C p(w|\theta_j)}_{\text{common theme}} + (1-\lambda_C)p(w|\theta_{j,i}))] + \lambda_B p(w|\theta_B)$$

Figure 2: Pd(w|Ci)

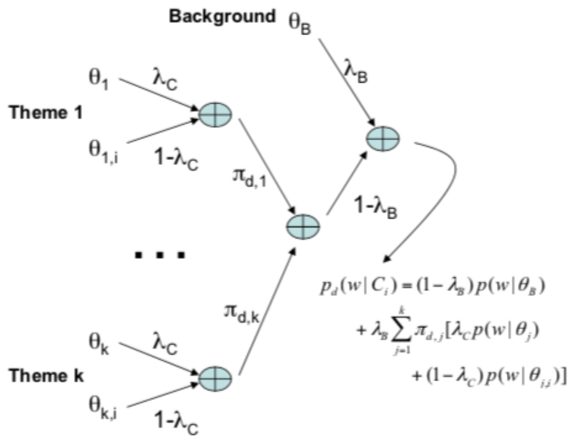

Figure 3: the cross-collection mixture model

2. Implementation Details

The class Corpus consists of the following functions:
__init__(document_path): initialize a Corpus object.

build_corpus(): read the document in document path, store the collection number and the document in self.collection and self.documents.

build_vocabulary(): read the documents and build the vocabulary for the whole dataset.

build_background_model(): build the background model from the whole dataset.

build_term_doc_matrix(): Construct the term-document matrix where each row represents a document, each column represents a vocabulary term.self.term_doc_matrix[i][j] is the count of term j in document i.

initialize(self, number_of_collections, number_of_clusters, random=True): initialize the matrices document_topic_prob , topic_word_prob and collection_topic_word_prob.

expectation_step(number_of_collections,number_of_clusters,lambda_b, lambda_c): the E-step updates the P(zd ci w = j), i.e. the topic_prob matrix, p(zd,Ci,w = B) i.e. the bg_prob matrix, and p(zd,Ci ,j,w = C), i.e. the common_topic_prob matrix.

maximization_step(number_of_collections, number_of_clusters): the M-step updates the $\pi_{d,j}^{(m+1)}$, $p^{(m+1)}(w|\theta_j)$ and $p^{(m+1)}(w|\theta_{j,i})$.

calculate_likelihood(number_of_collections,lambda_b, lambda_c): Calculate the current log-likelihood of the model using the model's updated probability matrices.

ccmm(number_of_collections, number_of_clusters, max_iter, lambda_b, lambda_c, epsilon): execute the text mining on the document passed in in max_iter times of iteration.

In each iteration, execute the E-step and the M-step, calculate the likelihood. Stop when the likelihood converges and print the topic models.

The function **ccmm(*)** is the core function of the project. The update of each matrix and the calculation of likelihood are based on the following equations.

$$\log p(\mathcal{C}) = \sum_{i=1}^{m} \sum_{d \in C_i} \sum_{w \in V} c(w,d) \log[\lambda_B p(w|\theta_B)$$
$$+(1-\lambda_B) \sum_{j=1}^{k} \pi_{d,j} (\lambda_C p(w|\theta_j) + (1-\lambda_C) p(w|\theta_{j,i}))]$$

Figure 4: log-likelihood calculation

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

Figure 5: E-step and M-step updates

3. Usage Documentation

The project uses two examples to test the text mining performance, both are similar from the example in the original paper. The data in the first example was scraped from BBC and CNN websites. The news URLs were selected by the author, so the contents are different from the news used in the original paper. The data in the second example was scraped from BestBuy.com, which are customer reviews on three kinds of laptop (Macbook-air-13-3-laptop, Dell-g5-15-6-fhd-gaming-laptop, Lenovo-yoga-c940-2-in-1-14-touch-screen-laptop). The scraper code and the scraped texts could be found in the project folder.

To run the scraper code, run “**jupyter notebook**”.

The code could be run by command “**python model.py -h**”.

To run the first example, “**python model.py --document wars_news.txt --clusterNumber 5 --collectionNumber 2 --c 0.25 --b 0.91**”

To run the second example, “**python model.py --document laptop_reviews.txt --clusterNumber 4 --collectionNumber 3 --c 0.7 --b 0.96**”. Notice here we set a smaller cluster number than the original paper, due to the content difference and the worse performance with 8 clusters in experiment.

The result will be saved in **results.txt.**

An example output of the first example text mining. The collection 0 is the Iraq-theme model, and the collection 1 is the Afghanistan-theme model.

```
Below is the common theme model
[[(0.02809961129191377, 'north'), (0.02530668267028961, 'security'), (0.02782356996193242, 'you'), (0.029513225804787997, 'used'), (0.02103130613936139, 'hit')],
[(0.024648154984289877, 'base'), (0.01818921231956598, 'become'), (0.022507060567200154, 'she'), (0.02168242276137832, 'information'), (0.018219938211881812, 'missile')],
[(0.024150552030896784, 'convoy'), (0.017561086504341093, 'support'), (0.018645797739428035, 'stories'), (0.017546011319900013, 'offered'), (0.016784916198551957, 'japan')],
[(0.021094752178102104, 'men'), (0.01662671152878418, 'christmas'), (0.015903369717084583, 'her'), (0.01723880200833875, 'destruction'), (0.015631970835733203, 'plans')],
[(0.019705150628160795, 'black'), (0.014058625564708233, 'man'), (0.015181210303341899, 'able'), (0.01481769568599101, 'kill'), (0.015454429278526157, 'governing')],
[(0.01613487830833186, 'tanks'), (0.011970518259997976, 'serious'), (0.013997192210679885, 'rather'), (0.014567209780303176, 'weapons'), (0.014335095350826537, 'higher')],
[(0.014851865310511784, 'attacks'), (0.011820997481331003, 'hundred'), (0.013467738632259446, 'wrong'), (0.013505444104360026, 'led'), (0.01423723296428102, 'met')],
[(0.01392837918728288, '33'), (0.011682834456867968, 'could'), (0.012504353842487624, 'hospital'), (0.01328844406519791, 'possible'), (0.013765170802453411, 'helicopter')]]

Below is the collection-specific theme model of collection 0
[[(0.02222287024012091, 'squadron'), (0.01790906672440042, 'cardinal'), (0.02136413350807891, 'public'), (0.012191454282515972, 'sarin'), (0.01986291209136344, 'mr')],
[(0.016721226254247778, 'bushs'), (0.01515942125789841, 'chipman'), (0.01416612853594781, 'cordesman'), (0.01106356910430808, 'hussein'), (0.017684210280416404, 'iran')],
[(0.01671086257989579, 'flight'), (0.01239922209652567, 'father'), (0.013649506006004626, 'newspapers'), (0.009867396189119294, 'gas'), (0.016861860775413542, 'council')],
[(0.015907503767411485, 'raf'), (0.011790252321389905, 'dr'), (0.013318440766525209, 'italian'), (0.00954428023177811, 'nerve'), (0.012792853547409276, 'hakim')],
[(0.015565805295329126, 'lieutenant'), (0.010829083980931936, 'zarqawi'), (0.01193354510464417, 'situation'), (0.00946193474039273, 'arrests'), (0.011186138268368326, 'bremer')]]

Below is the collection-specific theme model of collection 1
[[(0.016875426139356876, 'killed'), (0.01733303632598229, 'intelligence'), (0.048888342251760364, 'her'), (0.0317351619959092, 'west'), (0.018668070856679007, 'japanese')],
[(0.012637718509803235, 'deaths'), (0.01388406669946451, 'saddams'), (0.03969241044212267, 'lynch'), (0.020386653656079914, 'navy'), (0.014159376390513161, 'hickey')],
[(0.010771234409575498, 'fallujah'), (0.013057227749244907, 'gun'), (0.03920320898841882, 'she'), (0.016905013977020614, 'banner'), (0.01076721819980867, 'tuesday')],
[(0.010202654664547605, 'division'), (0.010704595725759073, 'odierno'), (0.023809631425137503, 'my'), (0.01687306121295398, 'wests'), (0.010240776940068318, 'plan')],
[(0.010066477883138187, 'cargie'), (0.009985743045275329, 'brigade'), (0.01397307765590269, 'rescue'), (0.013475968986894267, 'speech'), (0.009543519305460792, 'send')]]
```

4. Work Distribution

This is an individual project. All work was done by the author.