PROJECT DOCUMENTATION

Reproduce A Paper: A cross-collection mixture model for comparative text mining

Overview

This code is used to discover latent themes across collections. We provide two models, the simple mixture model (simplemix.py) and the cross-collection mixture model (ccmix.py). The simple mixture model treats multiple collections as one single collection and discovers k latent common themes in it. The Cross-collection mixture model explicitly distinguishes themes from different collections. It not only captures k common themes that characterize common information across all collections but also k collection-specific themes for each collection. The value of k is set by users and each theme is characterized by a multinomial word distribution.

Implementation

Dataset

The paper uses 2 datasets: war news, and laptop reviews. We are not able to obtain exactly the same datasets as those are used in the paper.

For the war news, we manually searched and downloaded 30 news articles from BBC or CNN for each of the two wars, published in one-year span (May 2003 - April 2004 for Iraq war, Nov 2001 to Oct 2002 for Afghanistan war) to approximate the war news dataset that is used in the paper. (code/data/war_dataset.txt)

For the laptop reviews, we chose 3 laptops from the Amazon.com best sellers in laptop computers: Acer Aspire 5 Slim Laptop, Apple MacBook Air, HP Chromebook 14-inch HD Laptop, and manually downloaded the top 40 reviews from Amazon.com. (code/data/laptop_reviews.txt)

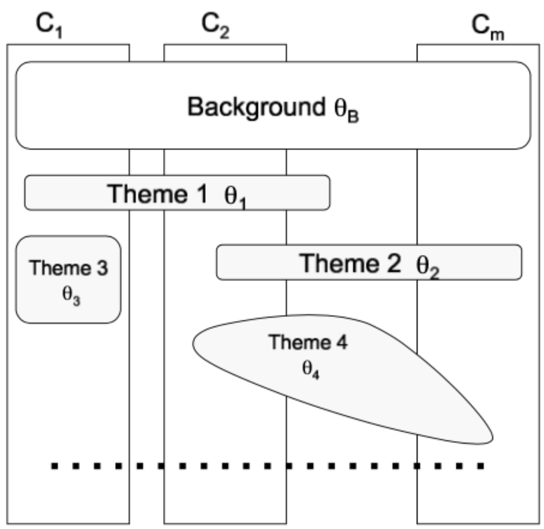
Each document occupies one line in the .txt file, prepended by an integer that indicates which collection the document is from.

Preprocessing

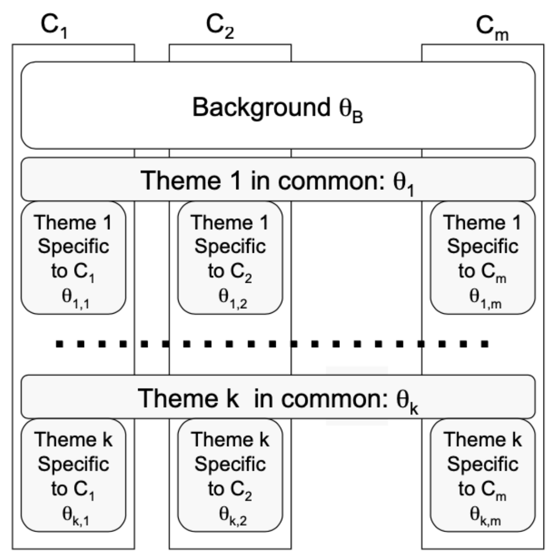
We preprocessed the data by removing stop words (including punctuation marks), words that contain less than 3 characters, and stemming the documents to map inflected words to their stems. Preprocessing is important to ensure that the resulting clusters after applying the mixture model would contain more meaningful words for evaluation.

Initialization

SimpMix Model (baseline)



CCMix Model



In SimpMix model, there are two types of themes: the background theme, and the shared themes (between collections)

In the CCMix model, there are three types of themes: the background theme, the common themes throughout all collections, and themes specific to the collections.

Both SimpMix and CCMix models use the EM algorithm to find the clusters iteratively. The two models also share the same background theme (*topic_word_prob_background*, dimension: 1 * *vocabulary_size*) which is calculated by taking the maximal likelihood of each word in the whole corpus (includes all the collections) before the iterative EM steps.

$$\hat{p}(w|\theta_B) = \frac{\sum_{i=1}^{m} \sum_{d \in C_i} c(w,d)}{\sum_{i=1}^{m} \sum_{d \in C_i} \sum_{w' \in V} c(w',d)}$$

The other parameters λ_B(*lambda_B*) and λ_C(*lambda_C*) which are the probability of selecting the background theme and the common theme respectively. We used 0.95 for $\lambda_B$and 0.25 for $\lambda_C$as the paper suggested.

Expression	Name in code	Initialization function	Dimension
λ_B	lambda_B		Scalar, 0.95
λ_C	lambda_C		Scalar, 0.25
$c(w, d)$	term_doc_matrix	build_term_doc_matrix (self) This function counts the term frequency in each document, and computes the background distribution.	number_of_collections * number_of_documents * vocabulary_size
$p(w\|\theta_B)$	topic_word_prob_background		1 * vocabulary_size
$\pi_{d,j}$	document_topic_prob	initialize_randomly (self, number_of_topics) This function randomly initialize	number_of_collections * number_of_documents * number_of_topics
$p(w\|\theta_j)$	topic_word_prob		number_of_topics * vocabulary_size
$p(w\|\theta_{j,i})$	topic_word_prob_collection_specific		number_of_collections * number_of_topics * vocabulary_size
ε	epsilon		Scalar, 0.00001

The EM algorithm

The implementation of the baseline SimpMix model is very similar to MP3 except for introducing a background model. Therefore, in this section, we will focus on how to implement the EM algorithm for the CCMix model.

<u>E step</u>

E step is to estimate the hidden variables: probability of selecting a theme in a mixture model.

Expression	Name in code	Calculation function	Dimension
$p(z_{d,C_i,w} = j)$ Collection specific theme	topic_prob_j	expectation_step (self, number_of_topics, verbose) This function performs E step	number_of_collections * number_of_documents * vocabulary_size * number_of_topics

$p(z_{d,C_i,w} = B)$ Background theme	topic_prob_B		number_of_collections * number_of_documents * vocabulary_size * 1
$p(z_{d,C_i,w} = C)$ Common theme	topic_prob_C		number_of_collections * number_of_documents * vocabulary_size * number_of_topics

<u>M step</u>

M step is to use the hidden variables to re-estimate the distributions in each theme, maximizing the likelihood.

Expression	Name in code	Calculation function	Dimension
$\pi_{d,j}^{(n+1)}$	document_topic_prob	maximization_step(self, number_of_topics, verbose) This function performs M step	number_of_collections * number_of_documents * number_of_topics
$p^{(n+1)}(w\|\theta_j)$	topic_word_prob		number_of_topics * vocabulary_size
$p^{(n+1)}(w\|\theta_{j,i})$	topic_word_prob_collection_specific		number_of_collections * number_of_topics * vocabulary_size
$log\ p(C)$	likelihoods	calculate_likelihood(self) This function calculates likelihood	List of scalars

The iteration of E-M steps continues until the difference between likelihood in adjacent iterations is less than εor the maximum iteration number is reached.

Usage

Python version

Python 3.6

Download Repo

git clone https://github.com/luoxix/CourseProject.git

Install Dependencies

pip install metapy
pip install numpy

Run code

To run the simple mixture model:
python simplemix.py --input_path ./data/laptop_reviews.txt --output_path ./result/result_simple_laptop.txt --lambda_b 0.95 --max_iterations 500 --number_topics 5 --number_top_words 8 --verbose True

To run the cross-collection mixture model:
python ccmix.py --input_path ./data/laptop_reviews.txt --output_common_path ./result/common_laptop.txt --output_specific_path ./result/specific_laptop.txt --lambda_b 0.95 --lambda_c 0.25 --max_iterations 1000 --number_topics 5 --number_top_words 8 --verbose True

python ccmix.py --input_path ./data/war_dataset.txt --output_common_path ./result/common_war.txt --output_specific_path ./result/specific_war.txt --lambda_b 0.95 --lambda_c 0.25 --max_iterations 1000 --number_topics 5 --number_top_words 8 --verbose True

Meaning of each argument

input_path: the path of the input file which contains all collections. Each line contains a document and the first number denotes which kind of collection it is from.
output_path: the path of the output file which contains k themes, for each theme, several top words with highest probability are shown.
output_common_path: the path of the output file which contains common themes

output_specific_path: the path of the output file which contains specific themes
lambda_b: the weight of the background model
lambda_c: the weight of common theme
max_iterations: the number of iterations for EM algorithm
number_topics: the number of latent themes
number_top_wods: the number of top words which are shown in the output file
verbose: whether to output the immediate information

Results and Evaluations

Run the code with instructions described above until the likelihood value converges.

Laptop Reviews

	Cluster 1	Cluster 2	Cluster 3
Common theme words	catalina0.0198 remov 0.0148 harddriv 0.0148 ad 0.0148 new 0.0102 second 0.0102 internet 0.0099 found 0.0099	samsung 0.0169 2400 0.0169 numer 0.0169 tech 0.0169 top 0.0169 edg 0.0169 left 0.0169 hour 0.0148	fan 0.0311 2020 0.0311 temperatur 0.0271 thermal 0.0271 cpu 0.0216 zoom 0.0203 extern 0.0203 bar 0.0203

	Cluster 1	Cluster 2	Cluster 3
Common	air 0.0252 connect 0.024 2020 0.022 thermal 0.0165 pictur 0.0137 poor 0.0125 bar 0.0124 new 0.0124	mode 0.0247 app 0.0179 differ 0.0172 side 0.0169 2400 0.0162 samsung 0.0162 support 0.014 download 0.013	call 0.0304 cpu 0.0162 amazon 0.0156 charg 0.0155 thermal 0.0153 brand 0.015 noth 0.012 bar 0.0115
Acer	drive 0.0274 ad 0.0201 harddriv 0.0197 remov 0.0194 click 0.0178 pictur 0.0175 new 0.0165 second 0.0142	remov 0.0204 mode 0.0163 harddriv 0.0146 ad 0.0146 wouldn't 0.0129 veri 0.0121 second 0.0121 case 0.0114	call 0.0354 amazon 0.0191 did 0.0166 tech 0.0157 bla 0.0155 brand 0.015 wait 0.0139 minut 0.0135

HP	connect 0.0347 amaz 0.0221 pictur 0.0206 love 0.017 unit 0.016 chrome 0.0158 drive 0.0151 nice 0.0151	cuz 0.0413 love 0.028 i'm 0.0207 didn't 0.0207 daughter 0.0207 polici 0.0207 glad 0.0207 aren't 0.0207	call 0.0306 charg 0.0204 brand 0.0197 amazon 0.0163 noth 0.0157 minut 0.0145 did 0.0139 differ 0.0136
Macbook Air	upgrad 0.1062 hard 0.0704 16gb 0.0551 drive 0.0522 probabl 0.0475 beauti 0.0475 instal 0.0452 suppos 0.0448	receiv 0.0258 lock 0.0234 dissapoint 0.0162 possibl 0.0162 owner 0.0162 anazon 0.0162 recoveri 0.0162 immedi 0.0162	call 0.0304 cpu 0.0162 amazon 0.0156 charg 0.0155 thermal 0.0153 brand 0.015 noth 0.012 poor 0.0115

From the result, we can see that SimpMix model is only able to find the themes in the whole corpus. The themes are shared among collections. It tells about what topics the overall corpus covers, but it is not able to identify topic differences between different collection. On the other hand, the CCMix model is able to find the common themes throughout the collections, and is also able to identify the different specific themes in each collection. For example, in Cluster 2, all three collections share the same common theme. However, there is a high frequency of "love" in HP collection, whereas there is a high frequency of "disappoint" in the Macbook Air collection. We may infer these two opposite attitudes maybe towards the same topic in the common theme. Probably this HP laptop provides app or support that buyers love, whereas Macbook Air disappointed buyers in these two aspects. Therefore, we can conclude that, in reviews evaluation, the CCMix model is able to identify different performance of similar products on the same aspects.

Another observation is that not all specific themes are well distinguished from the common theme / other specific themes within the same cluster (e.g., Cluster 3). This is probably because we use a uniform λ_C for all clusters. However, for some clusters, there are more overlaps in topics among collections and less differences, and λ_C should be larger to account for the common topics.

The data we use are from Amazon, and many of them are expression of feelings, and purchase experience with Amazon, instead of technical reviews. Therefore, the results are more on customer satisfaction. On the other hand, performing CCMix model on technical reviews will find more about the performance of each laptop.

War Dataset

	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5
Common theme words	khalifa 0.0307 o'neil 0.0199 newsweek0.0174 quran 0.0154 hous 0.0141 dyke 0.0113 cbs 0.0102 kennedi 0.0092	mirror 0.0161 threat 0.0139 palac 0.0134 wmd 0.0122 religi 0.0113 morgan 0.0092 nuclear 0.0087 pictur 0.0076	flag 0.0475 gun 0.0158 design 0.0091 nasratullah0.0088 kit 0.0088 equip 0.0087 zardad 0.0083 leak 0.0072	woodward0.0247 powel 0.0220 kerri 0.0210 marin 0.0201 matti 0.0174 gen 0.0125 clinton 0.0114 bandar 0.0110	draft 0.0237 opium 0.0211 hamdi 0.0202 rape 0.0169 wolfowitz0.0149 farmer 0.0149 poppi 0.0132 erad 0.0114

	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5
Common theme words	Flag 0.0450 chang 0.0158 design 0.0131 women0.0096 repres 0.0095 new 0.0086 threat 0.0085 equip 0.0084	gun 0.0332 draft 0.0296 leak 0.0173 kennedi 0.0158 katharin 0.0154 o'neil 0.0143 prosecut 0.0130 secret 0.0128	Wolfowitz 0.0403 soro 0.0370 group 0.0304 rumsfeld 0.0277 independ 0.0230 money 0.0220 rais 0.0185 moveon.org 0.0168	mirror 0.0277 religion 0.0218 god 0.0207 zardad 0.0195 koran 0.0164 morgan 0.0164 dearing 0.0158 cramer 0.0135	marin 0.0178 woodward 0.0165 powel 0.0147 coalit 0.0145 matti 0.0123 gen 0.0122 opium 0.0101 sunday 0.0095
Iraq theme words	flag 0.0450 chang 0.0158 design 0.0133 women 0.0096 repres 0.0095 new 0.0088 threat 0.0085 equip 0.0084	gun 0.0338 draft 0.0295 leak 0.0172 kennedi 0.0158 katharin 0.0153 o'neil 0.0142 prosecut 0.0129 secret 0.0128	wolfowitz 0.0403 soro 0.0370 group 0.0304 rumsfeld 0.0277 independ 0.0230 money 0.0220 rais 0.0185 moveon.org 0.0168	mirror 0.0277 religion 0.0218 god 0.0207 zardad 0.0195 koran 0.0164 morgan 0.0164 dearing 0.0158 cramer 0.0135	zapatero 0.0295 spain 0.0269 spanish 0.0167 coalit 0.0134 sunday 0.0132 marin 0.0124 woodward 0.0115 powel 0.0102

Afghan theme words	women 0.0326 rape 0.0119 elect 0.0118 soviet 0.0117 khalifa 0.0112 live 0.0099 nasratullah 0.0092 villag 0.0091	kerri 0.0389 hamdi 0.0382 hous 0.0315 clinton 0.0217 cohen 0.0186 wednesday 0.0154 foam 0.0152 polystyren 0.0133	money 0.1082 group 0.1033 rumsfeld 0.0746 million 0.0695 rais 0.0515 candid 0.0456 link 0.0384 campaign 0.0375	mirror 0.0654 zardad 0.0472 morgan 0.0396 daili 0.0285 tortur 0.0244 qlr 0.0216 pictur 0.0212 goldsmith 0.0202	newsweek 0.0554 quran 0.0489 magazin 0.0274 toilet 0.0228 desecr 0.0228 isikoff 0.0196 dirita 0.0196 investig 0.0151

Similarly, the result for the war dataset also demonstrates the ability of CCMix in differentiating the specific themes between collections. For example, in Cluster 1, we can see that in Iraq war news, people are more interested in reporting flag and mental changes, whereas in Afghanistan war news, women raped were reported in the highest frequency. In Cluster 2, Iraq war news reported more on gun and draft, while Afghanistan war news reported more on the two persons: Kerry and Hamdi.

Another observation with the war dataset is that in Cluster 1 to 4, the common theme has high similarity with the Iraq theme, and has much smaller overlap with the Afghan theme. This is probably because the Iraq specific theme is a very tight cluster where the top words have very high frequencies, such that the common theme is only able to account for partial frequencies of the top words. Only in Cluster 5, both themes are very different from the common theme.

Conclusion

In conclusion, we can see that CCMix model is able to address the task of comparative text mining by its capability to discover the latent common themes across all collections, and to summarize the similarity and differences of the collections along each common theme. However, the performance of CCMix varies on the choice of λ_C and the characteristics of the dataset that it is applied on.

Contribution

Xi Luo (xiluo4)	Yuheng Zhang (yuhengz2)
Algorithm implementation: CCMix Documentation: - Implementation - Results and Evaluations - Conclusion	Algorithm implementation: SimpMix Documentation: - Overview - Usage Tutorial presentation