Diego Millan – NetID: diegom3
CS410: Text Information Systems - UIUC

Project Documentation

Reproducing a Paper: A Cross-Collection Mixture Model for Comparative Text Mining

Introduction

My project selection is the reproduction of the paper titled "A Cross-Collection Mixture Model for Comparative Text Mining" cited at the end of this document.

The idea is to propose an improvement over simple mixture model for comparative text mining by explicitly distinguish the collections being compared. This allows to generate topic language models that are specific to each collection along with a common topic language model that describes all collections for each cluster (See Figure 2).

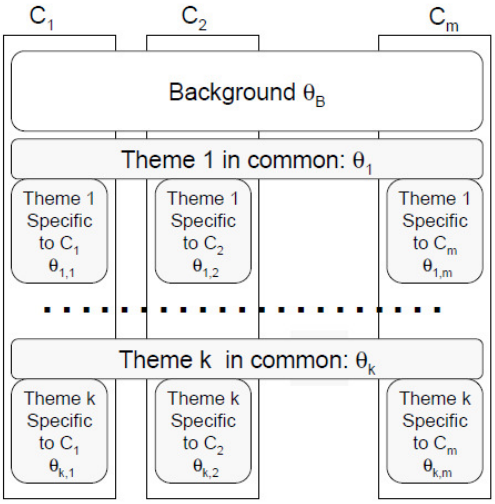

Figure 2: The Cross-Collection Mixture Model

The model

In a simple mixture model, the probability of a word is given by the sum of the portion of probability of being generated by a background model plus the probability of being generated by a topic model. Notice λ_B below is acting as a parameter that tunes the weight being applied to the background model. The parameter π is present on both models and simply describes the mix of topics of each document.

$$p_d(w) = \lambda_B p(w|\theta_B) + (1 - \lambda_B) \sum_{j=1}^{k} [\pi_{d,j} p(w|\theta_j)]$$

In the proposed cross-collection mixture model, the number of topic language models grows from k in the simple mixture mode to k + mk, being k the number of topics, V the vocabulary size and m the number of collections. We also add another tunable parameter λ_C that acts as the weight we give to the common topic models. So, in simple term, the probability of a word in a given collection is the

sum of it being generated by the background model plus the probability of being generated by either the common topic model or the collection specific topic model.

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

Implementation

For the implementation I used as base the PLSA programming assignment. In the assignment we did not have a background model so a function "build_background_model" was implemented using the whole corpus as follows:

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

The functions "build_corpus", "build_vocabulary", "build_term_doc_matrix" and "initialize_randomly" were mostly kept the same as they are also necessary for this model. They were only modified to explicitly separate the different m collections. For example, instead of relying in the "self.documents" matrix that contained all the documents, "self.documents_collections" was created which is a list of arrays that contain the documents of length m. A similar approach was used on the other mentioned functions.

The EM algorithm was heavily modified as the number of parameters grew considerably compared to the PLSA assignment. The following formulas were used to update the E and M step:

$$p(z_{d,C_i,w}=j) = \frac{\pi_{d,j}^{(n)}(\lambda_C p^{(n)}(w|\theta_j) + (1-\lambda_C)p^{(n)}(w|\theta_{j,i}))}{\sum_{j'=1}^{k}\pi_{d,j'}^{(n)}(\lambda_C p^{(n)}(w|\theta_{j'}) + (1-\lambda_C)p^{(n)}(w|\theta_{j',i}))}$$

$$p(z_{d,C_i,w}=B) = \frac{\lambda_B p(w|\theta_B)}{\lambda_B p(w|\theta_B) + (1-\lambda_B)\sum_{j=1}^{k}\pi_{d,j}^{(n)}(\lambda_C p^{(n)}(w|\theta_j) + (1-\lambda_C)p^{(n)}(w|\theta_{j,i}))}$$

$$p(z_{d,C_i,j,w}=C) = \frac{\lambda_C p^{(n)}(w|\theta_j)}{\lambda_C p^{(n)}(w|\theta_j) + (1-\lambda_C)p^{(n)}(w|\theta_{j,i})}$$

$$\pi_{d,j}^{(n+1)} = \frac{\sum_{w\in V} c(w,d)p(z_{d,C_i,w}=j)}{\sum_{j'}\sum_{w\in V} c(w,d)p(z_{d,C_i,w}=j')}$$

$$p^{(n+1)}(w|\theta_j) = \frac{\sum_{i=1}^{m}\sum_{d\in C_i} c(w,d)(1-p(z_{d,C_i,w}=B))p(z_{d,C_i,w}=j)p(z_{d,C_i,j,w}=C)}{\sum_{w'\in V}\sum_{i=1}^{m}\sum_{d\in C_i} c(w',d)(1-p(z_{d,C_i,w'}=B))p(z_{d,C_i,w'}=j)p(z_{d,C_i,j,w'}=C)}$$

$$p^{(n+1)}(w|\theta_{j,i}) = \frac{\sum_{i=1}^{m}\sum_{d\in C_i} c(w,d)(1-p(z_{d,C_i,w}=B))p(z_{d,C_i,w}=j)(1-p(z_{d,C_i,j,w}=C))}{\sum_{w'\in V}\sum_{i=1}^{m}\sum_{d\in C_i} c(w',d)(1-p(z_{d,C_i,w'}=B))p(z_{d,C_i,w'}=j)(1-p(z_{d,C_i,j,w'}=C))}$$

Figure 3: EM updating formulas for the cross-collection mixture model

However, I identified a mistake in the last formula that suggests a sum across collections to calculate the specific topic model probabilities which would end up removing the distinctions between collections and also removing one dimension of parameters. This was confirmed by the TAs as a mistake.

Finally, to monitor the performance of the algorithm, a log-likelihood function essentially the same as the one on PLSA assignment but with the additional parameters was implemented following the given formula on the paper:

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

I also made a simple mixture model implementation to compare the results, the EM updating formulas and log-likelihood estimation con be found on the referenced paper.

Experiments

The model was tested with 2 data sets, one with a collection of 59 news articles of the Iraq and Afghanistan war (26 from Afghanistan and 33 from Iraq war) from 2 different sources (BBC and CNN).

The second data set are laptop reviews of 3 different models (a MacBook, a Dell XPS and a Lenovo Yoga) with a total of 250 reviews obtained from Amazon.com.

λ_C and λ_B have to be tuned by the user. We can think of λ_B as how verbose the text is, the more it is, the more non-significant words we will find which in turn means we want to select a higher value. Generally, something higher than 0.9 is recommended.
For λ_C, as we go higher, we enrich the common topic models but hurt the specific, and vice versa, so it should be set very carefully based on the data and where we want the most emphasis.

The data sets used in the paper are not explicitly referenced so it was impossible to obtain the same exact data, which means I'm unable to reproduce the exact same response as in the paper and also have to tune my own λ parameters. Here are my results:

Theme 1								
Word	kit	wolfowitz	clothing	nbc	desert	buy	equipmen	logistics
Prob	0.03732396	0.03374832	0.02093988	0.01928082	0.01928082	0.01456099	0.0134511	0.0128544

Theme 2								
Word	ghraib	abu	symbol	prison	guards	photo	demolishe	usa
Prob	0.06427516	0.04786124	0.02022767	0.01926176	0.01506055	0.01155932	0.0115593	0.0092431

Theme 3								
Word	sarin	shell	nerve	neill	bowden	filled	agent	o
Prob	0.0378052	0.03047289	0.02514473	0.02077332	0.01689814	0.0160316	0.014525	0.013795

Theme 4								
Word	taliban	al	laden	bin	alliance	northern	he	hussein
Prob	0.01549612	0.00955844	0.00785728	0.00769362	0.00731475	0.00660803	0.0056029	0.0055208

Theme 5								
Word	eta	propaganda	black	trains	legality	undermine	planting	options
Prob	0.03331262	0.02464671	0.01716214	0.0166588	0.01129754	0.01129754	0.0112975	0.0112975

Table 1. Simple mixture model war news

While clustered words talk about certain topics common between both wars, it is not possible to make a distinction of which words belong to which collection.

	Common theme cluster 1									
Word	bin	laden	saudi	jihad	video	arabia	plot	family	father	authorities
Prob	0.18939329	0.0981552	0.03564389	0.03239276	0.03145353	0.02699685	0.0189802	0.0143503	0.01357955	0.01339957
	Cluster 1 - Collection Afghanistan war									
Word	laden	saudi	recorded	sudan	yemen					
Prob	0.1028003	0.02914018	0.02332654	0.02035473	0.02033913					
	Cluster 1 - Collection Iraq war									
Word	data	ritter	bowden	mi	dubious					
Prob	0.03806887	0.03382954	0.03000792	0.02535748	0.02112141					

	Common theme cluster 2									
Word	prisoners	prison	detainees	detention	ghraib	guards	executed	rights	amnesty	geneva
Prob	0.12874343	0.08034044	0.03878431	0.02395434	0.02115945	0.02067663	0.01651596	0.01642986	0.01595376	0.01563221
	Cluster 2 - Collection Afghanistan war									
Word	treated	guantanamo	cuba	base	bay					
Prob	0.03936906	0.03658937	0.03625978	0.02769193	0.02742886					
	Cluster 2 - Collection Iraq war									
Word	ghraib	abu	symbol	usa	photo					
Prob	0.07908194	0.07583343	0.02150438	0.01337695	0.01223926					

	Common theme cluster 3									
Word	blasts	spanish	aznar	seven	eta	king	policies	condolences	madrid	bombs
Prob	0.03022417	0.02932321	0.02662327	0.02131074	0.01481973	0.01430479	0.01400754	0.0121993	0.01211067	0.01150711
	Cluster 3 - Collection Afghanistan war									
Word	cache	bank	demonstrators	imf	southeastern					
Prob	0.03169261	0.02730445	0.02130979	0.01598093	0.01584349					
	Cluster 3 - Collection Iraq war									
Word	eta	trains	spain	express	karbala					
Prob	0.03305906	0.0198423	0.01936481	0.01058035	0.01055226					

	Common theme cluster 4									
Word	pentagon	cheney	wolfowitz	libya	europe	abroad	japan	controversial	vice	rumsfeld
Prob	0.04913761	0.03856055	0.02993884	0.02717369	0.02278614	0.02217882	0.01865351	0.01719155	0.01598577	0.01430421
	Cluster 4 - Collection Afghanistan war									
Word	french	karachi	propaganda	black	bus					
Prob	0.06793751	0.05502965	0.03675501	0.03230034	0.02751049					
	Cluster 4 - Collection Iraq war									
Word	wolfowitz	libya	japanese	gadhafi	dispatch					
Prob	0.06058456	0.05405023	0.02317438	0.02227723	0.01039494					

	Common theme cluster 5									
Word	the	of	to	that	and	in	is	was	a	we
Prob	0.07975547	0.03963266	0.03214286	0.02182307	0.0187433	0.01801543	0.01676557	0.01557104	0.01537607	0.01396474
	Cluster 5 - Collection Afghanistan war									
Word	taliban	afghanistan	alliance	northern	kandahar					
Prob	0.08405706	0.03487229	0.03465308	0.03116097	0.02699336					
	Cluster 5 - Collection Iraq war									
Word	iraq	saddam	iraqi	hussein	i					
Prob	0.04380953	0.0299254	0.02065532	0.01254649	0.00693652					

Table 2. Cross-collection mixture model war news

In the cross-collection model we can for example see cluster 2, where the common theme talks about prisoners, prison, detainees, detention, guards, while in the corresponding collection specific cluster we can see the location were these prisoners were being held. The Guantanamo bay detention camp is referenced in the Afghanistan cluster, which is relevant, while the Iraq war references Abu Ghraib, which is also a prison relevant to Iraq war.

The output of the model is a text file named either “SimpMix_output.txt” for the simple model and “CCMix_output.txt” for the cross-collection model. Also, due the nature of randomly initializing parameters, runs can vary so I included a few other runs in the “test” folder. In the same folder, output for the laptop reviews can be found that were omitted in this documentation .

Challenges and opportunities for improvement

During the implementation I encountered 2 issues that persisted and was not able to figure out.

The first one is related to my log-likelihood estimation which as stated in the lectures, the EM algorithm monotonically increases the likelihood. I ran into the problem that at some point relatively early in the

iterations, the likelihood decreases for a few iterations and then goes back to improve without interruption until convergence. I was unable to find any issues with my EM algorithm or the calculation of log-likelihood. I'm inclined to believe this could be an underflow issue during the E-step. I tried implementing the normalization to avoid underflow technique in lecture 10.4 but it did not make a difference, so the cause remains unknown.

The second problem was undefined divisions (division by 0) which caused the algorithm to crash. I was able to patch this issue by implementing pseudo counts in the M-step by adding a uniformly distributed prior with a parameter $\mu = 1$ as seen in lecture 9.9. This essentially guarantees each word "appears" at least once and avoids divisions by 0. I don't believe this diverges too much from reality so I thought it was a reasonable fix. Would be interested if the authors ran into a similar issue or if this is an implementation problem on my side.

Conclusion

The cross-collection mixture model can be a powerful tool when tuned optimally to create more meaningfully comparisons than the simple mixture model. In the experiments, for example the laptop reviews output, can be clearly used by manufactures to identify the weaknesses and strengths of their product while also learning valuable information of their competitors as analyzing reviews individually or pooled can easily misdirect us from what is really important.

But clearly we can extract insights from any comparable collections, maybe m different medical treatments response based on testimony from patients, political ideologies on social media and where they intersect or differ, etc.

Reference

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