CS410 Project Report

Cross-collection Mixture Model for Comparative Text Mining

Function Overview:

This software implements the method described in the paper A Cross-Collection Mixture Model for Comparative Text Mining**.** The paper proposes a method for discovering common themes across all the collections and for each theme, discover what is unique to a particular theme for each collection. In other words, for k topics among c collections, k common themes are discovered along with kc special themes for each collection-theme pair. A theme is modeled as probability distribution over words.

Implementation:

The skeleton of the code is similar to that of MP3. Models to be learned are randomly initialized, hidden latent variables are declared and the EM algorithm is implemented according to the formula listed below (copied from paper):

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

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

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

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

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

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

One of the issues we ran into when implementing were underflow errors due to very miniscule probabilities. Our workaround was to pad these very small probabilities to a predefined number (we used 1e-600).

Usage:

To run the code:

python3 cross.py {collectionName}

where collectionName is the name of the folder under data/collections.

Inside each folder under data/collections, there should be N files that make up the N collections, with each line in each of the N files representing a document. Our provided data includes one for wine (pinot noir and chardonnay) and covid-related articles by region (usa, asia, europe). Words are tokenized using the nltk tokenizer and any word < 3 characters are discarded to remove too much background noise in the theme models.

Example output:

	Cluster 1	Cluster 2	Cluster 3	Cluster 4
	december	vaccine	school	equipment
	government	efficacy	military	protective
	religious	capacity	event	hospital
	together	really	august	necessary
Common	found	early	surging	leave
	easter	clinical	environment	career
	message	inside	march	protect
	right	situation	viral	times
--	--	--	--	--
	nepal	chinese	olympic	medicine
	flight	technology	personnel	bitter
Asia	korean	storage	stadium	grief
	board	prompt	bases	volume
	enforceable	giant	sports	woman
--	---	---	---	---
	christmas	intensive	plasma	russia
	celebrate	german	blood	russian
Europe	christian	rising	denmark	vaccination
	vatican	force	danish	vladimir
	moscow	french	northern	circulation
--	---	---	---	---
	model	hurricane	curfew	lawsuit
	semester	shift	miami	union
US	count	recommend	pharmacy	court
	saved	agreement	mutation	letter
	relief	shutdown	enrolled	inadequate
--	---	---	---	---

Contribution:

Mackt2:
Implementation of cross-collection mixture model - (EM algorithm)
Documentation (Progress Report)

Hhc3:
Debugging of code - research using log-space and padding to avoid underflow errors
Data scraping - Sanitization and manipulation of various datasets found on kaggle (ad-hoc scripting for various formats, so not included in code)
Presentation (Video)