Social Topic Distributions

Hakan Akyürek

Mürüvvet Hasanbaşoğlu

June 2, 2020

Overview

1. **Finalize pre-processing**
2. **Finalize fine-tuning GloVe Embeddings**
3. **Train GMM**
4. **Calculating Probability Distributions**
5. **Next Tasks -** for the next 2 weeks

1. Finalize pre-processing

- Tokenization
  - `r"[a-zA-Z0-9]+|\.|\?|\!"`
- Lowercase
- Stop word removal
  - List taken from nltk.corpus
- Rare word removal (<=2)

- No lemmatization

	Tokens	Vocabulary
Processed	27M	220,518
Custom Processed	19M	88,119

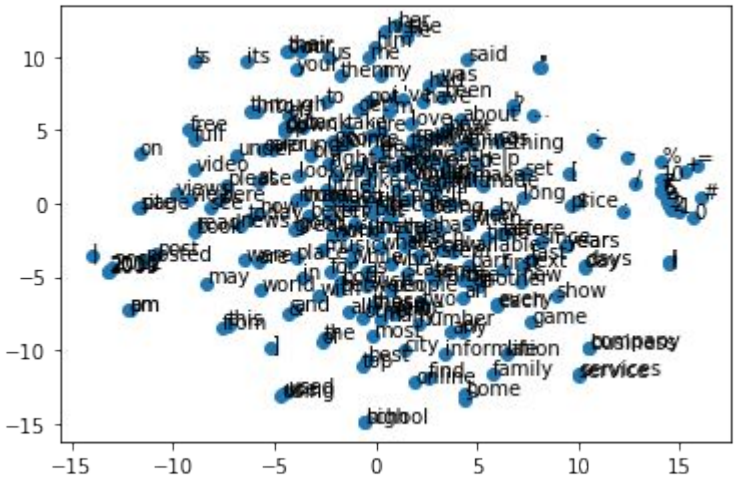
```
“Here's something quite similar from Berkley:.
http://newscenter.berkeley.edu/2014/12/09/organic-conv
entional-farming-yield-gap/”

“Here s something quite similar from Berkley . http
newscenter . berkeley . edu 2014 12 09 organic
conventional farming yield gap”

“here something quite similar berkley . http
newscenter . berkeley . edu 2014 12 09 organic
conventional farming yield gap”
```

2. Finalize fine-tuning GloVe Embeddings

- Common Crawl pre-trained word vectors
  - 42B tokens
  - 1.9M vocab
  - uncased
  - 300d vectors

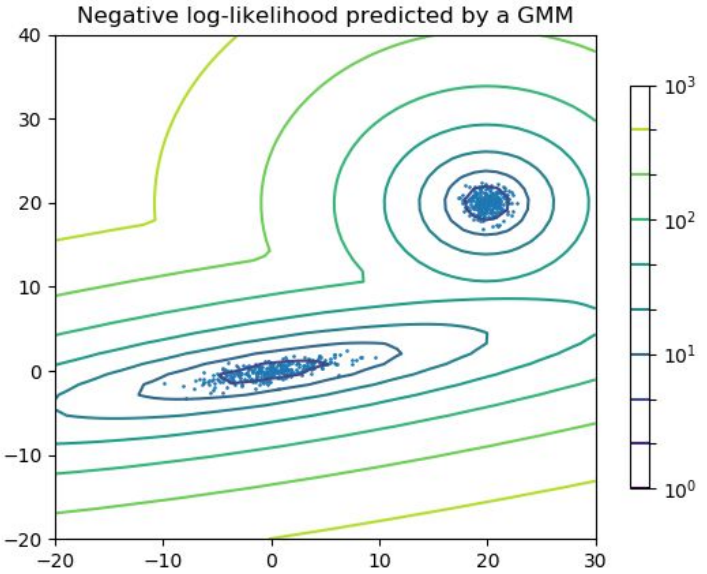
- Out-of-Vocabulary size: 7013 -> fine-tuned
- Size of word embeddings: 1,924,507

First 250 word embeddings into 2d



3. Train GMM

- Gaussian Mixture models from sklearn
  - ➜ implements the **expectation-maximization algorithm** for fitting mixture-of-Gaussian models
- Bayesian Information Criterion (BIC)
  - ➜ in order to select the number of clusters



3. Train GMM

Expectation Maximization:

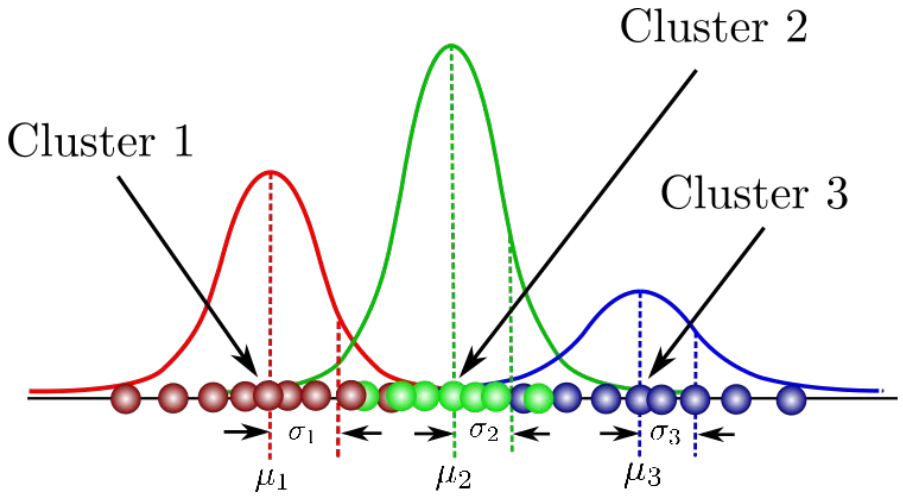
1. Initialize random clusters by mean (μ), covariance (Σ) and mixing probability (π)

$$\theta = \{\pi, \mu, \Sigma\}$$

2. For each data point, compute a probability of being generated by each cluster

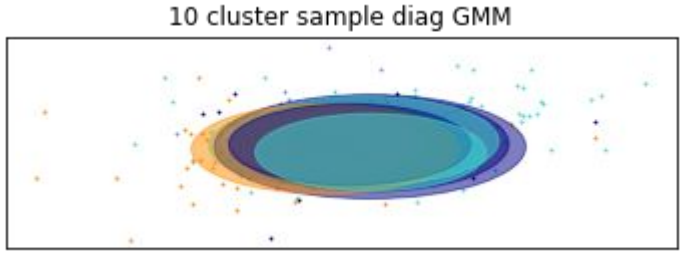
$$p(\mathbf{X}, \mathbf{Z}|\theta^*)$$

3. Adjust the parameters in order to maximize the likelihood of the data - iteratively



3. Train GMM

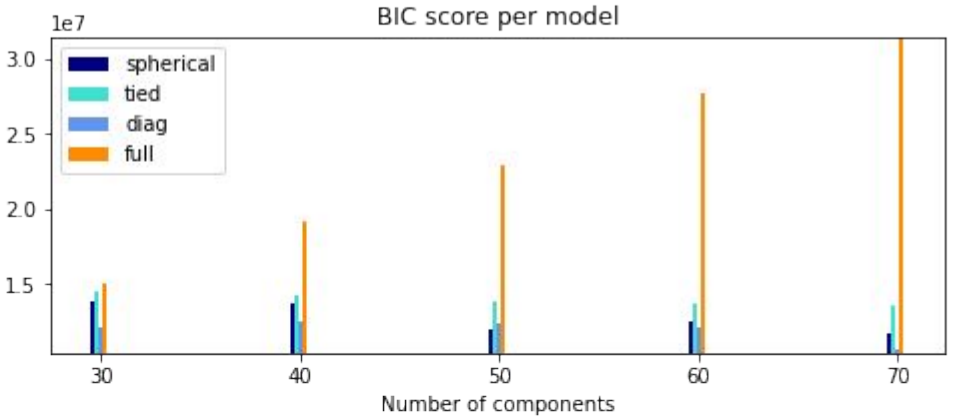
GaussianMixture(**covariance_type**='diag', **init_params**='kmeans', max_iter=100,
means_init=None, **n_components**=10, n_init=1,
precisions_init=None, random_state=None, reg_covar=1e-06,
tol=0.001, verbose=2, verbose_interval=1, warm_start=False,
weights_init=None)



- Covariance Types:
  - Full: Each cluster has it’s own covariance matrix
  - Tied: All clusters share the same covariance matrix
  - Diagonal: Each cluster has its own diagonal covariance matrix
  - Spherical: Each cluster has a single variance
- Initializing cluster means are done by k-means algorithm before the EM algorithm.

3. Train GMM

- While training we used the vocabulary instead of the whole words.
  - It doesn’t make sense to use duplicate data points while training a GMM
  - It makes easier for k-means to converge.
- We trained about 20 models in total and took the one with lowes Bayesian Information Criterion(BIC).
  - The best model we found has 70 clusters and the covariance type for that model is diagonal.



4. Calculating Probability Distributions

- We aren’t able to get the likelihoods we want: $p(x|k_n)$.
- Instead we can directly get posterior probabilities: $P(k_n|x)$.

$$k^* = \arg\max_{\theta_k} p(k|w'_1, \cdots, w'_N)$$

$$= \arg\max_{\theta_k} p(w'_1, \cdots, w'_N|k)p(k)$$

$$k^* = \arg\max_{\theta_k} p(k) \prod_{i=1}^{N} p(w'_i|k)$$

5. Next Tasks

- Define datasets to perform experiments: answering users, random users
- Implement Jensen-Shannon distance computation
- Implement cluster labeling

References

- Pennington, Jeffrey, Richard Socher, and Christopher D. Manning. "Glove: Global vectors for word representation." *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 2014.
- Fine tune GloVe embeddings using Mittens
- roamanalytics/mittens: A fast implementation of GloVe, with optional retrofitting
- https://medium.com/analytics-vidhya/basics-of-using-pre-trained-glove-vectors-in-python-d38905f356db
- https://scikit-learn.org/stable/modules/mixture.html

Questions