DSE 220: Machine Learning

Homework 4: Embedding of Words

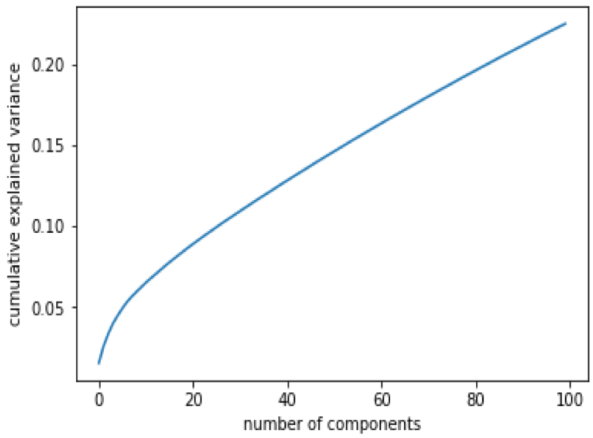
Kevin Kannappan
May 28, 2019

1 100-Dimensional Embedding Description

The Brown Corpus was developed in the 1960s and it contains 500 samples of English language text, totaling roughly 1 million words (Wikipedia). Leveraging the documentation found in the assignment, we found that the initial document found on NLTK contains $\approx$ 1.16M separate elements which were filtered down to remove both punctuation and "stop" words (or words considered to be useless based on high frequency and no informative value). Hence, we were left with $\approx$ 54K unique and valuable words to be leveraged for our embedding.

Utilizing a counter function, we were able to aggregate the counts of the occurrence of each word and sort them in order from most frequent to least. From that sorted list, we generated a *vocabulary* **V** and *context words* **C** based on the first 5000 and 1000 words respectively. Using those two lists, we were able define a probability distribution based on the number of context words, $c \in C$, surrounding the words in our vocabulary, $w \in V$, as $Pr(c|w)$. We also defined an overall distribution of the context words $Pr(C)$, both of which are distributions over **C**.

The last prerequisite step prior to building our 100-dimensional embedding was to generate the positive point-wise mutual information matrix, where effectively each word in the vocabulary has a C-dimensional vector indicating the likelihood of context words appearing with that vocabulary. This is defined as $\phi(w) = max(0, \log \frac{Pr(c|w}{Pr(C)})$ where the log scale is leveraged to reduce skew. At this point, our data is 1000-dimensional with relevant contextual information on relevant words around the target vocabulary words. To create our 100-dimensional embedding, $\psi(w) \in R^{100}$, we apply **Principal Component Analysis (PCA)** to reduce the dimensionality of our context words to our top 100 eigenvectors representative of the variance of those words. PCA was chosen since it has been proven robust for linear dimensional embedding, an assumption that was validated by subsequent results (outlined in Sections 2 and 3). The result of the PCA transformation removed a substantial amount of the variance, leaving $\approx$ 23% explained in the resulting embedding (depicted below):



2 Nearest Neighbor Results

2.1 Method and Description

A subset collection of words was selected from the embedding and **K Nearest Neighbors (with K=1)** was applied to return its nearest neighbor. Using cosine distance as given was proven to be quite effective, as the following results were intuitive. The most naive neighbor search implementation involves the "brute-force" computation of distances between all pairs of points in the dataset, which was implemented since the data no longer had large dimension constraints.

2.2 Results and Analysis

Nearest Neighbor results of the selected words are depicted in the figure below:

```
For communism , the nearest neighbor is =  utopian
For autumn , the nearest neighbor is =  storm
For cigarette , the nearest neighbor is =  bullet
For pulmonary , the nearest neighbor is =  artery
For mankind , the nearest neighbor is =  world
For africa , the nearest neighbor is =  asia
For chicago , the nearest neighbor is =  portland
For revolution , the nearest neighbor is =  modern
For september , the nearest neighbor is =  july
For chemical , the nearest neighbor is =  drugs
For detergent , the nearest neighbor is =  fabrics
For dictionary , the nearest neighbor is =  text
For storm , the nearest neighbor is =  saturday
For worship , the nearest neighbor is =  christian
For employees , the nearest neighbor is =  devoted
For million , the nearest neighbor is =  billion
For wife , the nearest neighbor is =  mother
For husband , the nearest neighbor is =  wife
For education , the nearest neighbor is =  public
For world , the nearest neighbor is =  war
For christ , the nearest neighbor is =  god
For would , the nearest neighbor is =  could
For cattle , the nearest neighbor is =  beef
For thousand , the nearest neighbor is =  hundred
For new , the nearest neighbor is =  york
```

In the sample of 25 words, only 2 appear to be counterintuitive: cigarrette matched with bullet and storm matched with saturday. Otherwise, the resulting values appear to be doing very well with both outlier words (communism to utopian) and common words (would to could). In conclusion, the nearest neighbor algorithm was successful in finding relevant related words. Future research could experiment with different distance metrics or higher values of K to see the collections of nearby words. There would be little value to tuning the search implementation as the data-size is still small.

3 Clustering

3.1 Method and Description

Using the vectorial representation of the embedding, we clustered the data into 100 separate clusters using the **K-Means++** algorithm. The advantage of K-Means++ is that its centroid selection is not random and it is better able to approximate the true center than regular K-Means as there are likely outlier values present. The algorithm used was the default method since our data is dense and did not need to be tuned. Lastly, the value of the number of times that the algorithm would be run with different centroid seeds (n init) was tuned so as to minimalize redudancy with higher n init values.

3.2 Results and Analysis

Sample (best) K-Means clusters are depicted in the figure below:

```
Cluster for economics:
 ['tax', 'pay', 'paid', 'sales', 'income', 'rates', 'share', 'annual', 'workers', 'capital', 'gain', 'increases', 'd
u', 'estimated', 'employees', 'gross', 'sets', 'rising', 'wage', 'vehicles', 'bills', 'raise', 'expense', 'extra', 'b
onds', 'insurance', 'dollar', 'shares', 'percentage', 'taxes', 'load', 'excess', 'wages', 'spending', 'estimate', 'co
nsumer', 'license', 'retired', 'dealers', 'adjustment', 'producing', 'net', 'adjusted', 'household', 'reducing', 'bui
lders', 'decline', 'buying', 'utility', 'proportion', 'customer', 'revenues', 'marginal', 'allowances', 'dealer', 'pr
ospects', 'monthly', 'saving', 'retail', 'stocks', 'earnings']

Cluster for state affairs:
 ['program', 'national', 'education', 'defense', 'medical', 'aid', 'planning', 'activities', 'assistance', 'education
al', 'policies', 'longterm']

Cluster for state affairs (2):
 ['social', 'power', 'law', 'political', 'economic', 'individual', 'society', 'community', 'personal', 'religious',
'christian', 'moral', 'influence', 'science', 'organization', 'understanding', 'religion', 'institutions', 'cultura
l']

Cluster for quantities:
 ['million', '10', 'cent', '15', '12', '30', 'daily', '20', '25', '14', '11', 'approximately', '18', 'billion', '13']

Cluster for names or pronouns:
 ['man', 'old', 'young', 'wife', 'mother', 'father', 'son', 'friend', 'met', 'husband', 'lived', 'poor', 'hospital',
'married', 'jack', 'spoke', 'died', 'captain', 'named', 'remembered', 'lady', 'murder', 'brother', 'daughter', 'merce
r', 'smiled', 'sweet', 'fellow', 'baby', 'wilson', 'talked', 'lewis', 'wondered', 'fathers', 'uncle', 'alive', 'love
d', 'joe', 'wished', 'dear', 'alfred', 'warren', 'cousin', 'sick', 'lucy', 'younger', 'adam', 'lawyer', 'anne', 'kat
e', 'papa', 'handed', 'thompson', 'sister', 'harry', 'bride', 'johnnie', 'blanche', 'aunt']
```

The clusters definitely appear to be moderately coherent. Values (numbers), subjects (names and pronouns), and even more granular clusters like economics or political areas are clustered together as well. Future research would benefit with less clusters, as some of the words appear to be clustered in smaller subsets than intuitively would make sense. Hence,

larger groups of words would be able to provide more insight - potentially leaving space to investigate **Hierarchical Clustering**. In terms of the tuned algorithm implemented, I am not sure changing the other hyperparameters would provide a performance (more coherence) increase.

4 Conclusion

Based on analyses of performance in clustering and nearest neighbor approaches, we can safely conclude that the 100-dimensional embedding chosen is an effective one. Improvements in both areas were highlighted in the corresponding sections. With regards to improving the utility of the embedding data, there are possibilities to use non-linear dimensionality reductions (e.g. Isomap), yet considering the effectiveness of PCA, they were not pursued. There are shortcomings with producing the embedding, as the loop written was very inefficient - potentially parallelism would help to reduce the computational requirements. Above all, the code and decisions made were to optimize over the Brown Corpus.