Homework 4

DSE 220: Machine Learning

Due Date: 26 May 2017

1 Instructions

A report for this Homework should be submitted on Gradescope and the code should be submitted on github by 26 May 2017. The code will only be evaluated from github so make sure you have correct code on github. To secure full marks both the report and the code should be in sync and logically correct. Please only submit relevant and legible code. Even if your method does not run well, you will gain marks for comprehensiveness of your analysis as and where applicable. **Please complete this homework individually.**

2 Overview

The large number of English words can make language-based applications daunting. To cope with this, it is helpful to have a *clustering* or *embedding* of these words, so that words with similar meanings are clustered together, or have embedding that are close to one another.

But how can we get at the meanings of words? John Firth (1957) put it thus: *You shall know a word by the company it keeps.*

That is, words that tend to appear in similar contexts are likely to be related. In this assignment, you will investigate this idea by coming up with an embedding of words that is based on co-occurrence statistics.
The description here assumes you are using Python with NLTK.

1. First, download the Brown corpus (using nltk.corpus). This is a collection of text samples from a wide range of sources, with a total of over a million words. Calling brown.words() returns this text in one long list, which is useful.

2. Remove stopwords and punctuation, make everything lowercase, and count how often each word occurs. Use this to come up with two lists:

    - A vocabulary V, consisting of a few thousand (e.g., 5000) of the most commonly-occurring words.

- A shorter list C of at most 1000 of the most commonly-occurring words, which we shall call context words.

3. For each word $w \in V$, and each occurrence of it in the text stream, look at the surrounding window of four words (two before, two after):

$$w_1 \quad w_2 \quad w \quad w_3 \quad w_4.$$

Keep count of how often context words from C appear in these positions around word w. That is, for $w \in V$, $c \in C$, define

$$n(w, c) = \#\text{ of times c occurs in a window around } \quad w.$$

Using these counts, construct the probability distribution $\Pr(c|w)$ of context words around w (for each $w \in V$), as well as the overall distribution $\Pr(c)$ of context words. These are distributions over C.

4. Represent each vocabulary item w by a $|C|$-dimensional vector $\phi(w)$, whose c'th coordinate is:

$$\phi(w) = max(0, log\frac{\Pr(c|w)}{\Pr(c)})$$

This is known as the (positive) pointwise mutual information, and has been quite successful in work on word embedding. (20 marks)

5. Suppose we want a 100-dimensional representation. How would you achieve this?

6. Investigate the resulting embedding in two ways:

   - Cluster the vocabulary into 100 clusters. Look them over; do they seem completely random, or is there some sense to them?
   - Try finding the nearest neighbor of selected words. Do the answers make sense?

7. The Brown corpus is very small. Current work on word embedding uses data sets that are several orders of magnitude larger, but the methodology is along the same lines.

3 What to turn in

On the due date, turn in a **typewritten** report containing the following elements (each labeled clearly).

1. *A description of your 100-dimensional embedding.*

   The description should be concise and clear, and should make it obvious exactly what steps you took to obtain your word embeddings. Below, we will denote these as $\Psi(w) \in R^{100}$, for $w \in V$. Also clarify exactly how you selected the vocabulary V and the context words C. (30 marks)

2. *Nearest neighbor results.*

   Pick a collection of 25 words $w \in V$. For each w, return its nearest neighbor $w' \neq w$ in V. A popular distance measure to use for this is *cosine distance*:

$$1 - \frac{\Psi(w).\Psi(w')}{||\Psi(w)|| \quad ||\Psi(w')||}$$

   Here are some suggestions for words you might choose:

   communism, autumn, cigarette, pulmonary, mankind, africa, chicago, revolution, september, chemical, detergent, dictionary, storm, worship

   Do the results make any sense? You can use other distance measures apart from cosine distance to improve the results. (20 marks)

3. *Clustering.*

   Using the vectorial representation $\Psi(.)$, cluster the words in V into 100 groups. Clearly specify what algorithm and distance function you using for this, and the reasons for your choices.

   Look over the resulting 100 clusters. Do any of them seem even moderately coherent? Pick out a few of the best clusters and list the words in them. (30 marks)

In this homework, you are expected to come up with three strategies (one for each task). You should to turn-in a concise report (2-3 pages) with the results of the above tasks along with a description of your final approach. You should clearly describe your approach along with all the parameters you use. The description should be enough for someone else to recreate your model/results. Along with the description, you are expected to provide the performance analysis of your model including the shortcomings (if any) and any ideas to further improve it. You should also include a brief description of all the major approaches you tried (with code on github), and the reason why you selected your current model over them.

You won't be graded on the performance of your model but you'll be graded (30%) on the comprehensiveness of your analysis. You are not expected to come up with a new algorithm (though that would be great!), but your report should justify that the approach you suggest is equivalent/better than some of the standard models discussed in the class. In essence, to secure full marks in this section, your analysis of this problem should be in-depth and thorough.