CMPSCI 689: Machine Learning Mini Project 1

Due: October 5 2015 on Moodle

Abstract

This is the first mini project and it is related to unsupervised learning. Read the entire document before starting the project. Your solutions must be computer formatted and submitted online on Moodle by **October 5 2015**. There will be no extensions of this deadline. Each student must turn in an individual solution based on their own work, with no joint work. **Do not wait until the last weekend to work on this project. You will not be able to complete it!**

Outline

The goal of this project is to explore a recently proposed solution to the problem of *learning word embeddings*, an unsupervised learning task that provides a foundation for subsequent natural language processing (e.g., document classification, ranking, spam filtering, summarization etc.). The project involves reading a paper describing the method, and running and creating some code to reason with learned word embeddings from `Wikipedia`. The paper, code, and data can be downloaded from the website `http://nlp.stanford.edu/projects/glove/`. The approach is called **GLOVE**, which stands for Global Vectors for Word Representation, proposed by Pennington, Socher, and Manning of Stanford University.

- First, download the code, the paper, and a few different word embeddings from Wikipedia on the web site (start with 50-dimensional embeddings, then try 100-dimensional embeddings etc.). Note that higher dimensional word embeddings create much larger files and will require more download time and storage space.
- Follow the instructions to unpack the code and run it on your favorite machine of choice (the code is in C and should run on any machine with a C compiler, such as gcc on Linux or Apple Macs, or any version of Windows Visual Studio etc.). The accompanying `demo.sh` script provides a small test program, but it requires MATLAB and will only run on a MATLAB equipped machine.
- From the Moodle web site, download the Google and MSR datasets for word analogies (these are both text files, and contain different types of analogies, such as X is to Y as A is to B).
- The goal of the mini project is to answer the questions below relating to the GLOVE technique, as well as to augment the results in Table 2 of the GLOVE paper by finding the performance of the GLOVE embeddings on the Google and MSR word analogy tasks for different sized embeddings (e.g., 50, 100 and 200).

Theory Question 1 (20 points)

To solve a word analogy task, such as `Man is to Woman as King is to X`, Mikolov et al. proposed using a simple cosine distance measure, called `COSADD`, whereby the missing word was filled in by solving the optimization problem

$$\text{argmax}_{y \in V} \delta(\omega_y, \omega_x - \omega_a + \omega_b) \tag{1}$$

for a generic word analogy problem of the form a is to b as x is to y, and where ω_i is the vector space D-dimensional embedding of word i and δ is the cosine distance given by

$$\delta(i,j) = \frac{\omega_i^T \omega_j}{\|\omega_i\|_2 \|\omega_j\|_2} \tag{2}$$

where $\|\omega_i\|_2 = \sqrt{\omega_i^T \omega_i}$.

In Lecture 2, we defined the abstract notion of an inner product between two vectors, denoted $\langle a, b \rangle$. Does the above cosine distance function satisfy the axioms of inner product given in the lecture notes. Show either that it does by proving that each condition holds (e.g., non-negativity, symmetry etc.), or give a counterexample to any of the conditions that do not hold.

Theory Question 2 (20 points)

In a subsequent paper, Goldberg and Levy proposed an alternative distance measure, called `COSMULT`, using the same cosine distance as Equation 2, but where the terms are used multiplicatively rather than additively as in Equation 1. Specifically, they proposed using the following multiplicative distance measure:

$$\text{argmax}_{y \in V} \frac{\delta(y,b)\delta(y,x)}{\delta(y,a) + \epsilon} \tag{3}$$

where ϵ is some small constant (such as $\epsilon = 0.001$). Consider solving the word analogy problem `London is to England as Baghdad is to X`. Using the pre-computed word embeddings, solve this analogy using COSADD and COSMULT for the specific case of setting X to `Mosul` (a large Iraqi city) vs. X to `Iraq`. Does either distance measure yield the right answer? Play with other examples and give a discussion of whether COSADD or COSMULT better represents the word analogy solution.

Programming Question 3 (30 points)

Implement the COSADD and COSMULT distance metrics and compare them on the Google word analogy task, and report on their relative performance. Also, compare your results to those shown in Table 2 of the GLOVE paper.

Programming Question 4 (30 points)

Implement the COSADD and COSMULT distance metrics and compare them on the MSR word analogy task, and report on their relative performance. Vary the size of the embeddings in this and the previous question to see how performance varies as dimension size is increased or reduced.

Optional Component: Bonus Project (30 points)

Can you suggest a new and improved distance metric that outperforms COSADD and COSMULT? .This is related to a project that I successfully carried out at IBM Watson Research during my sabbatical year.

You can find my solution to this problem in a recent paper that I wrote on my web page `https://people.cs.umass.edu/~mahadeva/papers/word-emb-grassmann.pdf` using a revised cosine distance measure based on modeling subspaces as points on a curved manifold called the *Grassmannian*. My method performs significantly better than COSADD and COSMULT, but requires a priori knowledge of the specific word analogy relation (that is, the distance metric is different for each relation). Can you think of a generic distance measure that works better, without needing a priori knowledge of the word relation?