Machine Learning and Computational Statistics: Project Report

Recently, neural network based models for computing continuous vector representations of words have gained popularity. In particular, Mikolov and colleagues [1, 2, 3] have proposed two new models for learning a continuous embedding space for words from raw text data. Their models are similar to the neural probabilistic language model [5] and recurrent neural networks models [6]. However, Mikolovs models are log linear, and thus have a much lower computational complexity. Furthermore, their models are trained on billions of words, which is several orders of magnitude larger than anything previously trained.

Mikolov et al. showed that the embedding space learned from these models has very interesting linear structure that can be exploited to solve a variety of language tasks. They showed that analogical questions of the form King is to man as Queen is to ____ can be solved by simple algebraic operations on the vector representations. This particular question would be solved by computing the vector King - man + Queen and then searching for the nearest word vector. The result should be woman. After training with the CBOW and Skip-gram models, these types of analogical questions (both semantic and syntactic) are able to be answered correctly using this vector offset method. They have also shown that embedding spaces that result for different languages have similar geometrical structure and thus words can be translated from language A to language B by learning a simple linear transformation from the embedding space for language A to the embedding space for language B.

In our project we will be exploring various aspects of the linear structure of these embedding spaces. Our hypothesis is that words belonging to a particular superclass, or said differently, words that are all related in a particular context, might have some low dimensional structure. For example, it might be the case that words denoting fruit such as apple,orange,pear and banana all lie close to a lower dimensional manifold. To be clear, we are suggesting something stronger than just the claim that these points should all cluster together. We already know that the latter should be the case. Given the linear structure that appears to be present in the embedding space, we further hypothesize that these manifolds will in fact be hyperplanes. As a result we will be approaching this problem as a subspace clustering problem, rather than a general manifold learning problem. We have several interrelated goals for this project. We divide the time from now until the presentation into three blocks of two weeks each.

The first step is to determine whether or not the linear subspace structure does in fact exist. We will test for this hypothesis as follows. First, we will run a subspace clustering algorithm to get an assignment of points to clusters. Then, for each cluster we will determine the rank K subspace that best fits the point assigned to the cluster and compute the reconstruction error given by the low rank approximation. We will do this for increasing K and plot the reconstruction error as K increases. If the points assigned to this cluster really do lie close to some low dimensional subspace, then we would expect the reconstruction error to drop off at some point for K ¡ the original dimension. We intend to conduct these experiments during the first two of weeks.

Given that the hypothesized structure exists, our next step will be to exploit the structure for some practical purposes. The first step here is to check if the subspaces (clusters) can be associated with semantically or syntactically meaningful concepts. We can use an existing word hierarchy (e.g .) to test for this.

Given a particular subspace, we can compute the percentage of words in the subspaces that fall within a superclass in the existing word hierarchy. Admittedly, we will have to consider a diverse set of categories as potential candidates to map to clusters. However, we think this can be accomplished by a simple a brute force search guided by heuristic. We allot the next two weeks for doing this. If we are in fact able to discover subspaces

that are syntactically or semantically meaningful, we can use this structure in several ways. We hypothesize that if we were to project the points down onto such spaces, it will be easier to make inferences about similar/dissimilar words that make sense within this context. For example, suppose we have several parameterized subspaces, each associated with a different superclass or context. Imagine one is related to fruits and another to electronic devices. If we take the word vector for apple we would expect that if it was projected onto the fruit subspace then its nearest neighbors would be pear, orange, etc and if we project it down onto the electronics subspace its nearest neighbors would be mac, iphone, android, etc. This means that, for words that have an ambiguous meaning, we could more easily find similar words if we condition on a given context. We can test this hypothesis by generating a set of ambiguous words and and computing nearest neighbors of the word vector after projecting it onto multiple relevant subspaces. This will be conducted in the last two weeks before the final presentation where we will also generate relevant plots and figures to showcase our experiments and analyses.

We also suspect that the low dimensional structure could be exploited to improve performance on the "analogical reasoning" test (described in section 4). As is described in section 1, we know that there exists a rich linear structure in the embedding space since analogical questions are solved by this simple vector offset method. However, the embedding space is likely quite noisy. Our hypothesis is that the performance on analogical questions would be improved if points that can be well approximated by a lower dimensional hyperplane are projected onto the plane before doing the vector algebra. We can evaluate this hypothesis by testing our performance on the analogical reasoning test set using the vector offset method before and after projection.

Open source code to train both the CBOW and the skip-gram architectures have been released by Google [7]. We also have access to 30 million vector representations of words trained on Googles original training data. In addition to the above, we intend to run the training algorithm on freely available text data such as the entire corpus of wikipedia in order to conduct our experiments. The algorithm trains on sentences preprocessed to have punctuations removed and phrases (such as ice-cream) appended together.

We propose two ways to evaluate the performance of our goals. In [3], the authors release a test set to measure relational similarity. Given a relationship of the form "A is to B as C is to D", the task involves finding D given A,B and C. Mikolov et al. solve this by doing simple vector algebra; they compute the vector representation of A-B+C and search for the nearest neighbor of the resulting vector. Their model outperforms baseline models at this task. We intend to test our performance on this analogical reasoning test. Our hypothesis is that the performance on analogical questions would be improved if points that can be well approximated by a lower dimensional hyperplane are projected onto the plane before doing the vector algebra. The second evaluation metric that considers how well the subspaces we find correspond to superclasses in an existing word hierarchy[8]. If the distribution of words in a subspace to superclasses in a hierarchy is largely unimodal, we can conclude that the subspaces do indeed correspond to loosely related concepts. Mikolov et al. proposed two distinct to compute word vectors: the Continuous Bag-of-Words (CBOW) model and the Skip-gram model. Both models consist of a projection layer followed by a hierarchical softmax output layer. The only nonlinearity is the exponentiation in the output layer. The two models have slightly different objective. Given a sequence of words, $w_0, \ldots, w_k, \ldots, w_{k+k}$, the CBOW model uses $w_0, \ldots, w_{k-1}, \ldots, w_{k+1}, \ldots, w_{k+1}$, to predict w_k. The input words are represented as 1-hot vectors and are all projected down into the same position by averaging the projected vectors. Note that this means all information about location of the words in the sentence is lost (hence the name bag of words). The projection

layer acts as input to a hierarchical softmax layer that aims to project the remaining word w_k. The Skip-gram model takes a single word as input and aims to predict the k previous and k future words. Performance of both models is comparable.

The thesis of subspace clustering algorithms is that points in a higher dimensional space are generated by projections from a lower dimensional space. The clustering algorithm aims to recover the number of subspaces and their dimensionalities along with the basis for the subspace that every point in the original space is deemed to lie in.