UNSUPERVISED LEARNING OF SEMANTIC CONCEPTS

VIKAS C. RAYKAR AND VINAY D. SHET

1. Goal the project

Given a large number of articles comprising of natural text we intend to train on the text and extract clusters of words which are semantically related. When I query the system with a word it will return all the words which are very related to the same theme. For example our training set could be all the articles appearing in a magazine. If I query the system with war it would return all the words like soldiers, aircrafts, Iraq etc. Note that all the words do not mean the same they are related by a common unifying theme of war. We want to apply the techniques of nonlinear manifold learning to this unsupervised learning task. We experimented with two techniques, a linear technique Principal Component Analysis and and non-linear manifold learning technique called Isomap [11]

2. Vector Representation of words

We represent each word as a vector where each element shows the number of times the word appears in each of the articles which we train on. Suppose we have say V words in our vocabulary and we are training on N articles. We represent all the words as V points in N dimensional space. We form a matrix X where the X_{ij} is the number of times the i^{th} word appears in the j^{th} document. As a gedanken experiment consider the following three different documents each consisting of 3 sentences each.

```
Document 1:

Panini was a Sanskrit grammarian who gave a comprehensive and
scientific theory of phonetics, phonology, and morphology.
Sanskrit was the classical literary language of the Indian Hindus
and Panini is considered the founder of the language and
literature. It is interesting to note that the word "Sanskrit"
means "complete" or "perfect" and it was thought of as the divine
language, or language of the gods.

Document 2:

Panini's grammar of Sanskrit is highly systematized and relies on
patterns found in the language. Features of language are
categorized according to their similarities, and then form the
subject matter of the set of ordered morphological rules which
constitute the bulk of the work. Inherent in the analytic approach
```

This report was written as a part of the course CMSC723: Natural Language Processing.

```
employed by Panini are the concepts of the phoneme and the
morpheme, only recognized by Western linguists millennia after he
used them.

Document 3:

Grammar is the study of the rules governing the use of a language.
That set of rules is also called the grammar of the language, and
each language has its own distinct grammar. Grammar is part of the
general study of language called linguistics.
```

The first two documents are about the Sanskrit grammarian Panini and the third one is about grammar in general. For example let us consider four words Panini, Sanskrit, Grammar and Language and count the number of times it appears in each of the documents [1]. In particular, we get the following count matrix X,

	Document 1	Document 2	Document 3
Panini	2	2	0
Sanskrit	3	1	0
Grammar	1	1	4
Language	3	2	4

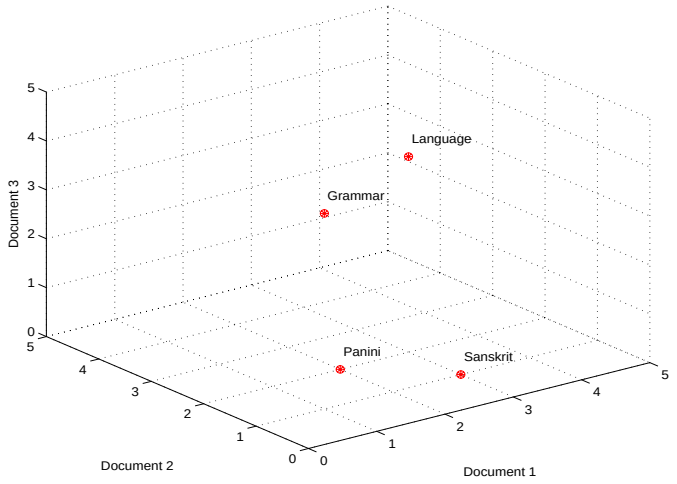

FIGURE 1. 4 words embedded in a 3 dimensional space.

[1]We use only the stem of each word. Stemming can be done using a simple algorithm like the Porter Stemmer. However for the current project we have not taken stemming into account.

Each of the words can be represented as a point in the 3 dimensional document space as shown in Figure 1. In general we typically have words embedded in higher dimensional spaces based on the number of documents we consider. The conjecture is that words similar in meaning and context often cluster together in this high dimensional space. For example the words Panini and Sanskrit are close together. So are grammar and language. The context in which the word appears in different documents provides sufficient information for them to cooccur in the high dimensional space. In our implementation we normalize the corresponding vector for each word so that each cell represents the frequency of occurence of a particular word in a particular document.

3. Principal Component Analysis

As of now we have a vector space representation of different words. We conjectured that words similar in meaning cluster together. So when I query the system and I want to find words similar in context, I find the neighbors of that particular point in the higher dimensional space. A smart thing to do first would be to find the principal components and project these points on a different low dimensional basis.

We can find a lower dimensional representation for the words using techniques like Principal Component Analysis (PCA) [5]. PCA is a statistical dimensionality reduction technique. Given N points in d dimensions PCA essentially projects the data points onto p, directions($p < d$)which capture the maximum variance of the data. These directions correspond to the eigen vectors of the covariance matrix of the training data points. Intuitively PCA fits an ellipsoid in d dimensions and uses the projections of the data points on the first p major axes of the ellipsoid. So essentially after doing PCA we have a lower dimensional representation of all the words.

4. Why Nonlinear Manifold Learning?

However dimensionality reduction techniques like PCA assume that the data essentially lies on a linear manifold. But it is very unlikely that they lie on a simple linear manifold. At best we can conjecture that these words lie on some interesting low dimensional non-linear manifold manifold. The shape of this manifold may depend on the semantic content of the words and documents. We call such a manifold a semantic manifold. As an example as shown in Figure 2 the words may lie in a one dimensional manifold embedded in a 2D space. If we naively use the euclidean distance then words paint and war should be close together. However if we unfold the 1D manifold, the two words are the farthest. inear dimensionality reduction techniques like Principal Component Analysis and Multi dimensional Scaling may not capture the perceptually relevant features if the data is embedded non linearly in the higher dimensional space.

Nonlinear manifold techniques essentially help to unfold the manifold [10] giving a low dimensional representation. Once we have unfolded the low dimensional semantic manifold words similar in context lie close to each other. Depending on the shape of this semantic manifold two words which are close together with respect to the euclidean distance metric may actually be far off on the semantic manifold. Ideally we need to use the geodesic distance on the manifold to get a true measure of how similar they are semantically.

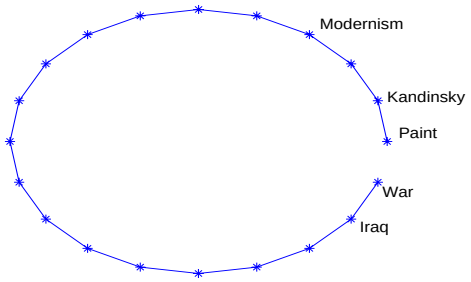

FIGURE 2. An interesting manifold in some high dimensional space

Manifold learning can be viewed as implicity inverting a generative model for a given set of observations. Let Y be a d dimensional domain contained in a Euclidean space R^d. Let $f : Y \rightarrow R^D$ be a smooth embedding for some $D > d$. The goal is to recover Y and f given N points in R^D. Isomap [11] and Locally Linear Embedding (LLE) [9] are two techniques which provide implicit description of the mapping f. Without imposing any restrictions of f the problem is ill-posed. The simplest case is a linear isometry i.e. f is a linear mapping from $R^d \rightarrow R^D$ where $D > d$. In this case Principal Component Analysis (PCA) recovers the d significant dimensions of the observed data. f can also be either a isometric embedding or a conformal embedding. The Isomap algorithm can recover an isometric embedding. The LLE can recover both isometric as well as conformal embeddings. In this project we experimented with both the LLE and the Isomap technique. However LLE did not give stable results and the program often crashed. So we used the Isomap algorithm for our experiments.

5. Isomap Algorithm [11]

The crux of the Isomap algorithm is finding an efficient way to compute the true geodesic distance between observations, given only their Euclidean distances in the higher dimensional observation space. The idea is that Euclidean distance is approximately equal to the geodesic distance for closeby points. For points which are faroff the geodesic distance has to be computed by a series of hops. The Isomap algorithm as proposed in [11] consists of three main steps.

(1) Construct the neighborhood graph G over all observation points. Connect points i ad j if they are closer than ϵ or if i is one of the K nearest neighbors of j. Set the edge lengths equal to distance between i and j. The distance could be either Euclidean or other domain specific distance metric.
(2) Compute shortest paths in the graph between every two points using either the Floyd's or the Djkstra's algorithm.
(3) Apply Multi Dimensional Scaling to the resulting geodesic distance matrix to find a d-dimensional embedding.

6. Details of the corpora used

The corpora that was used in this work was text from Computer Vision conference proceedings. We used papers from fours conferences (CVPR '03, CVPR '02, CVPR '99 and ICCV '01) which amounted to about 1012 articles. The papers

were all in PDF format. We used ghostscript's ps2ascii to extract the text from the pdf files. Perl scripts were then used to extract the word frequencies for each article. Using a combination of awk and bash scripts, data files for each word were generated which contained word frequency information. This was then imported into Matlab where the matrix of word against frequency in each article was generated. We were able to extract about 108,000 words, however, due to computation limitations we were able to use only about 3058 words only.

7. Implementation Details

We ran the PCA and the Isomap Algorithm on this 3058 x 1012 frequency matrix. We used the Isomap MATLAB code available on the authors website. PCA was implemented in MATLAB. For both we used the first 100 dimensions. For the Isomap algorithm we used 20 nearest neighbors. Figure 3 shows the first two dimensions as extracted for all the words for both PCA and Isomap. Note that the data is in 100 dimensional space and we are looking at the projection in two dimensions. A blow up of the first 50 words is also shown in the figure.

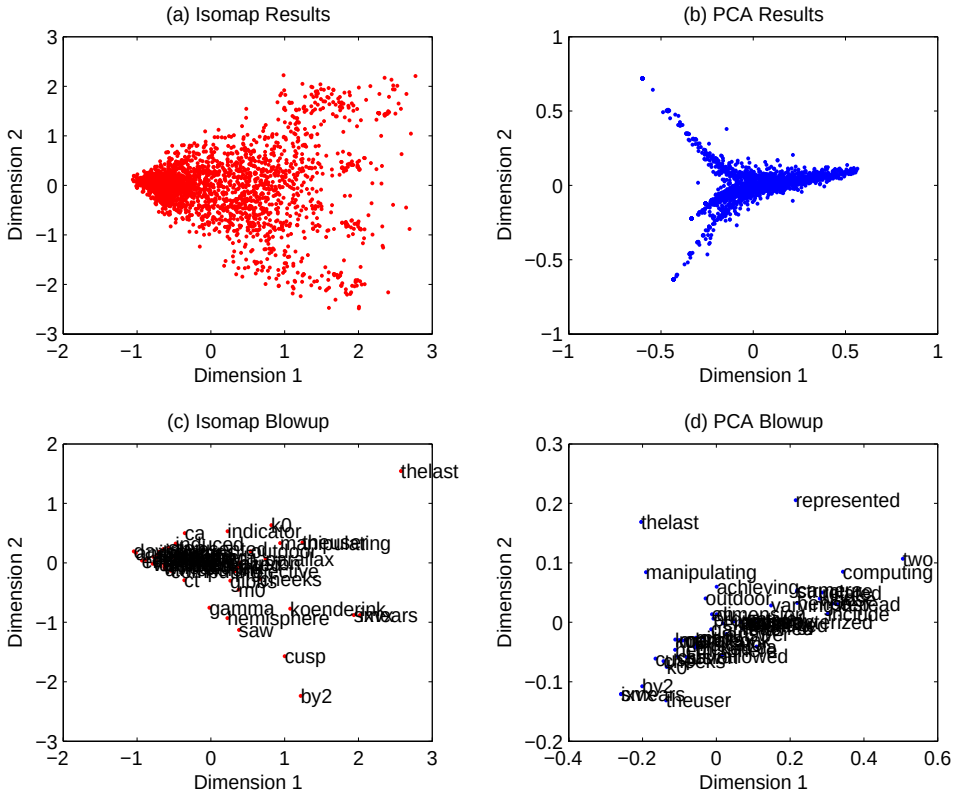

Figure 3. The first two dimensions extracted by the Isomap and the PCA. A blowup of the first 50 words.

Once we have the data we can query the system with a word in the vocabulary and ask it to return say K nearest neighbors. Following are some of the examples.

When I query the system with the word 'stereo' for 20 neighbors I get the following results. The first column is the result of Isomap and the second one is that of PCA. Note that these words do not mean stereo but these are the words that come to mind first for an expert in computer vision. Essentially by going through the training data we have learnt the context in which the words appear.

```
>> neig('stereo',Isomap_Index,Pca_Index,words_isomap,words_pca,20);
-----------------
Isomap        Pca
-----------------
ans =
    'stereo'             'stereo'
    'recovery'           'traditional'
    'left'               'zhang'
    'right'              'rig'
    'pair'               'scenes'
    'corresponding'      'recovery'
    'recovered'          'determining'
    'zhang'              'dense'
    'geometry'           'seitz'
    'correspondences'    'pair'
    'completely'         'thescene'
    'have'               'purely'
    'known'              'correspondences'
    'knowing'            'maps'
    'both'               'structured'
    'traditional'        'i3'
    'only'               'corporation'
    'two'                'visibility'
    'correspond'         'offset'
    'associated'         'pinhole'
    'remains'            'depth
```

When I query with the word RANSAC I get all words related to robust estimators.

```
>> neig('ransac',Isomap_Index,Pca_Index,words_isomap,words_pca,10);
-----------------
Isomap        Pca
-----------------
ans =
    'ransac'         'ransac'
    'inliers'        'inliers'
    'cartography'    'fischler'
    'outliers'       'cartography'
    'fischler'       'bolles'
    'outlier'        'torr'
    'lmeds'          'lmeds'
    'torr'           'outlier'
    'murray'         'outliers'
    'estimators'     'mestimators'
```

```
    'bolles'          'consensus'
```

When I queried with the name of two professors 'aloimonos' and 'jacobs' who work in computer vision here at Maryland, the neighbors corresponded to the areas that they work in and also the names of their research collaborators.

```
>> neig('aloimonos',Isomap_Index,Pca_Index,words_isomap,words_pca,10);
-----------------
Isomap        Pca
-----------------
ans =
    'aloimonos'         'aloimonos'
    'translational'     'spetsakis'
    'vision'            'translational'
    'translation'       'uy'
    'camera'            'translation'
    'rotation'          'egomotion'
    'computer'          'frommotion'
    'motion'            'eti'
    'flow'              'ofview'
    'constraint'        'qualitative'
    'robert'            'tx'

>> neig('jacobs',Isomap_Index,Pca_Index,words_isomap,words_pca,10);
-----------------
Isomap        Pca
-----------------
ans =
    'jacobs'         'jacobs'
    'yale'           'spanned'
    'lighting'       'ullman'
    'under'          'albedos'
    'lambertian'     'lighting'
    'kriegman'       'yale'
    'conditions'     'lambertian'
    'linear'         'under'
    'images'         'combination'
    'ullman'         'subspace'
    'spanned'        'insensitivity'
```

Some more results,

```
>> neig('result',Isomap_Index,Pca_Index,words_isomap,words_pca,10);
-----------------
Isomap        Pca
-----------------
ans =
    'result'           'result'
    'experimental'     'still'
    'main'             'results'
    'demonstrate'      'another'
    'provided'         'shows'
```

```
    'thus'            'also'
    'purpose'         'however'
    'consists'        'only'
    'called'          'without'
    'idea'            'into'
    'proven'          'has'

>> neig('kalman',Isomap_Index,Pca_Index,words_isomap,words_pca,10);
-----------------
Isomap        Pca
-----------------
ans =
    'kalman'       'kalman'
    'tracking'     'mk'
    'time'         'state'
    'filtering'    'recursive'
    'model'        'filtering'
    'tracked'      'online'
    'state'        'maintain'
    'filter'       'modelbased'
    'frame'        'demand'
    'blake'        'equipped'
    'current'      'incorporation'

>> neig('optical',Isomap_Index,Pca_Index,words_isomap,words_pca,10);
-----------------
Isomap        Pca
-----------------
ans =
    'optical'          'optical'
    'motion'           'flow'
    'translational'    'axis'
    'perpendicular'    'observer'
    'due'              'diverging'
    'been'             'schunck'
    'motions'          'motions'
    'rotation'         'focus'
    'domain'           'fields'
    'since'            'motion'
    'estimation'       'pinhole'
```

8. Evaluation

The result of both the Isomap and PCA algorithm is a set of m neighbors for any word w in the corpora. These m neighbors are words that co-occur with w (with high probability) in any given article that is drawn from a similar knowledge domain as the original corpora. The original corpora that we used was drawn from

four Computer Vision Conference Proceedings. From a different Computer Vision Conference Proceedings we extracted 50 articles for evaluation.

Given every word w_i^j in each test article j, we first check whether it exists in our original corpora (remember that due to computation constraints we were unable to use 'all' the words from our original corpora). If w_i^j exists, then we find its m_i^j nearest neighbors as reported by both Isomap and PCA approaches. Out of the m_i^j neighbors predicted, let p_i^j words be found in article j. The evaluation score for Isomap per article j then is going to be

$$score^j = \frac{\sum_{i=0}^{W^j} \frac{p_i^j}{m_i^j}}{W^j}$$

where W^j is the number of words in article j (which can be found in the original corpora). A similar evaluation score is also computed for PCA. Ideally $\frac{1}{50}\sum_{j=1}^{50} score^j$ should be 1.0 indicating that all words predicted have indeed co-occurred. In practise this is never the case, however, we can use these scores to get an estimate of how well each of these approaches have performed. Figure 4 shows this evaluation score for each of the 50 articles for different number of neighbors. Clearly, the Isomap approach show higher scores and therefore better models semantic co-occurrence between words.

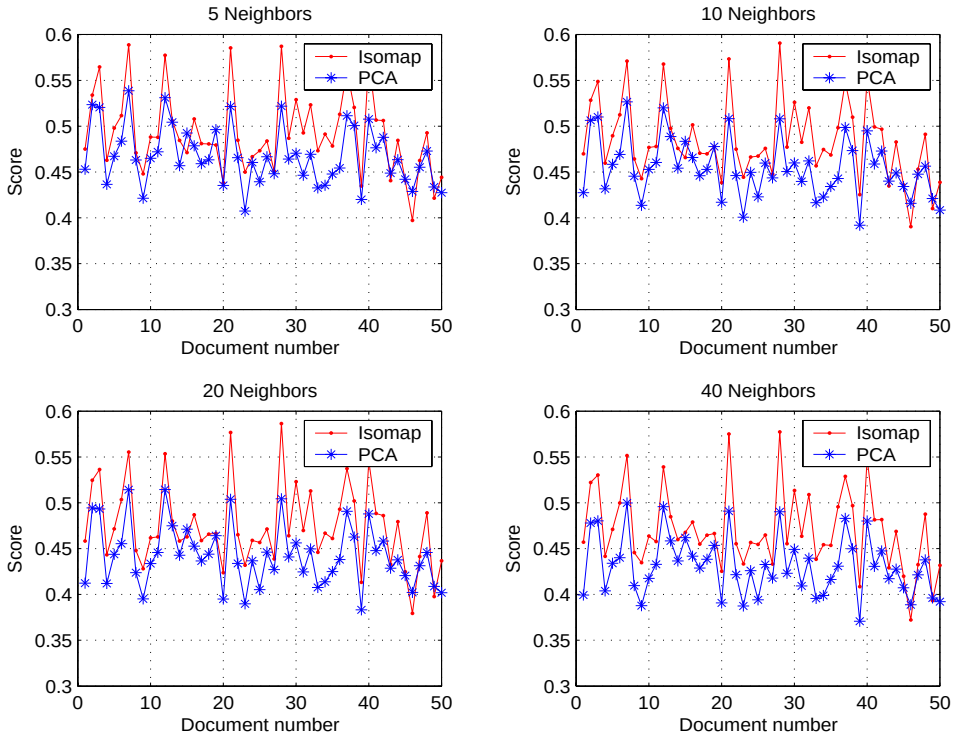

FIGURE 4. Comparison of scores for PCA and Isomap. The test set contained 50 documents. The plots show the scores for different number of neighbors queried.

9. Literature Survey

Much of the relevant work we found by searching the web on semantic analysis. [4] discusses using Latent Semantic Indexing for Information retrieval. The LSA group at Colorado Boulder has a lot of work on Latent semantic analysis [8][7]. They apply the technique of Singular value Decomposition (SVD) to learn a low dimensional manifold. They say that an optimal of 300 dimensions should be sufficient. They also demonstrate the performance of their system on some applications like domain knowledge tests, automatic essay grading etc. They also have a website where you can submit a summary and they say how well it matches with the original subject matter [1]. But essentially by using SVD they have assumed that the manifold is linear. We conjecture that by learning non-linear manifolds we should get a more perceptually relevant semantic space, probably of dimensions lower than 300. [12] gives a theoretical justification for the use of SVD. They use a sub-space model coupled with MDL for Latent Semantic Indexing. Telecordia technologies [3] says that they have a 'novel, patented information retrieval method developed using statistical algorithms ,LSI can retrieve relevant documents even when they do not share any words with a query'. There is a company called Knowledge Analysis Techniques [2] which uses these kinds of methods for automatically grading essays. There are also a lot of bit philosophical papers discussing on the implications of meaning and how humans acquire it and how the current approach can serve as a computational model for it [6]. There is a wealth of information related to Latent Semantic Indexing. I could not find methods which use non-linear techniques instead of linear ones.

10. Conclusion and future work

In this project, we implemented two different approaches for semantic analysis of words from large corpora. One was the traditional PCA based approach that used Euclidean distance measure in the high dimensional space and the other was a non-linear technique. We conjectured that if data lies in a non-linear manifold, the PCA based approach would not model the data appropriately. To this end we used the non-linear Isomap technique that uses the Geodesic distance along the manifold. The evaluation on independent test set has shown that Isomap performs marginally better than the traditional PCA based technique.

We believe that this approach has a lot of potential and the reason for only a marginal improvement is our computational limitation regarding the number of words we could handle. Had we used more words, the manifold would have been more densely sampled, leading to a better performance by the Isomap technique. When evaluating these approaches by hand, we observed other interesting phenomena like both PCA and Isomap approaches give similar results for words that are highly specialized to a given domain, while the Isomap technique performed better for slightly more generalized words.

Future work based on this research promises to be exciting. If well tuned, these approaches could be used for automatic classification of documents based on domain of knowledge (medical, engineering, literature, etc.). These approaches could also be potentially used to identify various styles adopted by different writers (by way of using certain words together).

References

1. *http://lsa.colorado.edu/summarystreet/.*
2. *http://www.knowledge-technologies.com/.*
3. C. Chen, N. Stoffel, N. Post, C. Basu, D. Bassu, and C. Behrens, *Telcordia lsi engine: Implementation and scalability issues.*, In Proceedings of the 11th Int. Workshop on Research Issues in Data Engineering (RIDE 2001): Document Management for Data Intensive Business and Scientific Applications, Heidelberg, Germany, 2001.
4. S. T. Dumais, G. W. Furnas, T. K. Landauer, and S. Deerwester, *Using latent semantic analysis to improve information retrieval.*, In Proceedings of CHI'88: Conference on Human Factors in Computing, New York: ACM, 281-285., 1988.
5. Hotelling H., *Analysis of a complex of statistical variables in principal compenents*, Journal of Educational Psychology **26** (1933).
6. T. K. Landauer and S. T. Dumais, *A solution to plato's problem: The latent semantic analysis theory of the acquisition, induction, and representation of knowledge*, Psychological Review **104** (1997), 211–240.
7. T. K. Landauer, P. W. Foltz, and D. Laham, *Introduction to latent semantic analysis.*, Discourse Processes **25** (1998), 259–284.
8. T. K. Landauer, D. Laham, and P. W. Foltz, *Learning human-like knowledge by singular value decomposition: A progress report.*, Advances in Neural Information Processing Systems **10** (1998), 45–51.
9. S. Roweis and L. Saul, *Nonlinear dimensionality reduction by locally linear embedding*, Science **290** (2000), 2323–2326.
10. H. S. Seung and D. D. Lee, *The manifold ways of perception*, Science **290** (2000), 2268.
11. J. B. Tenenbaum, V. de Silva, and J. C. Langford, *A global geometric framework for nonlinear dimensionality reduction*, Science **290** (2000), 2319–2323.
12. Hongyuan Zha, *A subspace-based model for information retrieval with applications in latent semantic indexing*, Technical Report No. CSE-98-002, Department of Computer Science and Engineering, Pennsylvania State University (1998).

Department of Computer Science, University of Maryland, College Park
E-mail address: vikas,vinay@cs.umd.edu