International Journal of Applied Engineering Research
ISSN 0973-4562 Volume 10, Number 4 (2015) pp. 10507-10514

http://www.ripublication.com

Vector Space Document Representation of High Dimensional Data For Efficient Semi Supervised Clustering

R. Malathi Ravindran[1] , Dr.S.Antony Selvadoss Thanamani[2]
[1]Assistant Professor, PG Department of Computer Applications, NGM College, Pollachi-642002, TamilNadu, India, rajansunguru@gmail.com, 9486978648
[2]Associate Professor Department of Computer Science, NGM College, Pollachi-642002, TamilNadu, India, selvdoss@gmail.com, 9994433490

Abstract

Clustering the text documents is the process of unsupervised learning task where the similar documents are grouped into clusters. Traditional machine learning approaches to document clustering are fully automated and unsupervised where class labels are unknown a priori. Another important issue of traditional clustering approach is it's lose of algorithmic approach when handling high dimensional data. But in real world, due to social media, the size and the dimension of the data is much higher than imaginable and also the data is both labeled and unlabelled. In real application domains, however, side information about the domain or data sets can be often available. So, semi supervised clustering with vector space model is proposed in this paper to represent the high dimensional data as vectors and the ranking of the document can be used as the side information for the semi supervised clustering.

Keywords: Data Mining, Clustering, High dimensional data, Classification, Vector Space Model

Introduction

Clustering is a typical unsupervised machine learning task (Banerjee, A., Ghosh, (2006)). The field of machine learning has traditionally been divided into three sub fields: Supervised learning, Unsupervised learning and Reinforcement learning. Semi-supervised learning is a new machine learning technique where the learner observes the data items $\{x_i\}$ where i=1 to n and partial feedback. In Semi-supervised clustering, besides a set of unlabeled data items, the learning system also observes side information taking various forms. For example, the side information can say that items x_i and x_j are similar, items x_p and x_q are different, or a cluster can contain no more than m data items, etc. The side information serves as a weak supervision to the

learning system. So the learning task is different from unsupervised clustering, where the learning system cannot benefit from side information even if it is available. Clustering the high dimensional data is the challenging task today. Dimension Reduction is very important to efficiently cluster the high dimension data. In this paper, Vector Space Model is used to reduce the dimension of the text data and it gives the rank for each document. Side information in the form of pairwise constraints (Hakan Cevikalp et al.,2008) is general. In particular, there are two types of pairwise constraints

- **Must-links**: two data items x_i and x_j are similar and thus should be clustered together.
- **Cannot-links**: two data items x_i and x_j are different and thus should be placed into different clusters.

For example, labeled training data can be expressed by pairwise constraints but not for inverse. Moreover, pairwise constraints naturally originate from many real application domains. In this work apart from pairwise constraints, ranking from the vector space model is also used as the side information for effective semi supervised clustering of high dimensional data(Daoqiang Zhang et al.2007).

Related work

Katrin Erk and Sebastian Pad´o (2008) presented a novel structured vector space model that addresses the selection preferences for words argument positions. This makes it possible to integrate syntax into the computation of word meaning in context.

Mikhail Bilenko , Sugato Basu , Raymond J. Mooney(2004) presents a new semi supervised clustering algorithm that integrates the constraint based methods and distance function learning methods. Jitendra Nath Singh and Sanjay Kumar Dwivedi.(2012), proposed different approaches of vector space model to compute similarity score of hits from search engine and this will lead to a clear understanding of the issues and problems in using the vector space model in information retrieval.

Difficulties in handling High-Dimensional Data

High-dimensional data are prevalent in applications such as database, text mining, image processing, sensor data analysis, and bioinformatics. Learning from high-dimensional data involves high computation cost (Sunita Jahirabadkar and Parag Kulkarni. Article: (2013)). Besides, a learning system has the curse of dimensionality problem. In particular, as the number of features keep increasing, the learning performance can decrease after a certain point. This high-dimensionality difficulty has frequently been observed in practice (Michel Verleysen 2003). In supervised learning of high-dimensional data, the number of training data items is much less than the number of parameters to be learned. The learned model has high variance and does not generalize well.

For unsupervised learning, as the feature dimensionality increases, data points become increasingly sparse. Thus, data items in the high dimensional space are equally far apart from each other no matter whether they are from the same cluster or not. Since all the clustering approaches critically rely on pairwise distances between

data items, many clustering techniques lose their algorithm effectiveness when dealing with high-dimensional data.

So, it is important to reduce the dimensionality when dealing with high dimensional data. Feature selection and feature reduction are two ways to reduce dimensionality. Feature selection reduces dimensionality by selecting a subset of existing features. Thus, the physical interpretation of each feature is preserved in the reduced space. One may apply judicious feature selection to greatly reduce the number of features prior to learning from the data. The results demonstrated that in removing many features, information about the underlying data groups may be lost. Besides, a criterion function for feature selection is typically defined as a function of the classification error. Thus, feature selection is mainly used in supervised learning. For clustering tasks, since labels are not available, selecting an appropriate subset of features is difficult.

Feature reduction reduces dimensionality by combining features with linear or non-linear transformations. Feature reduction is applicable to both supervised learning and unsupervised learning, depending on the availability of training data. A feature reduction approach can greatly reduce the feature space dimensionality while still preserve discriminative information.

However, unlike in feature selection where the selected features retain their original physical interpretation, the new features generated by a feature reduction approach usually do not have a clear physical meaning. In general, the choice between feature reduction and feature selection depends on the application domain.

The feature reduction techniques can be linear or non-linear. Linear approaches are fast and suitable for practical application. However, when data lie in a complicated manifold, non-linear feature reduction algorithms are able to represent data better in the reduced space.

Proposed Work

This paper proposed the vector space representation of documents to reduce the dimension of a data and to rank the documents according to its relevance. Vector Space model is better than Boolean model where expressing complex user requests are difficult, handling high dimensional data is difficult and ranking the output is difficult (J. Mitchell, M. Lapata. 2008). Vector Space model is a statistical model of representation where a document is typically represented by a bag of words.

The procedure of a vector space model can be divided in to three stages. The first stage represents the document indexing where content bearing terms are extracted from the document text. The second stage is the weighting of the indexed terms to enhance retrieval of document relevant to the user. The third stage ranks the document with respect to the query according to a similarity measure.

The t distinct terms remain after preprocessing is called as index terms. So, the dimension of the data is represented as

Dimension = t = |vocabulary|

Each term i, in a document or query, j, is given a real-valued weight, wij. Both documents and queries are expressed as t-dimensional vectors:

$dj = (w1j, w2j, \ldots, wtj)$ ------- (1)

A term document matrix is used to represent the n documents in the vector space model.. Each entry in the matrix corresponds to the "weight" of a term in the document; zero represents the term has no significance in the document or it simply doesn't exist in the document.

$$\begin{array}{c|cccc} & T_1 & T_2 & \ldots. & T_t \\ D_1 & w_{11} & w_{21} & \ldots & w_{t1} \\ D_2 & w_{12} & w_{22} & \ldots & w_{t2} \\ : & : & : & & : \\ : & : & : & & : \\ D_n & w_{1n} & w_{2n} & \ldots & w_{tn} \end{array}$$

Terms are weighted locally or globally or both according to a given weighting model. If local weights are used, then term weights are normally expressed as term frequencies, *tf*. If global weights are used, the weight of a term is given by *IDF* values.

In this work Salton's Vector Space Model is used which incorporates both local and global weights. So it is called as *tf*IDF* weighting

weight of a term = *tf*IDF* ------- (2)

The important terms are the more frequent terms in a document.

fij = frequency of term *i* in document *j*

In order to normalize the *term frequency* (*tf*),

$tfij = fij / maxi \{fij\}$ ------- (3)

Some terms appear in two documents, some appear only in one document. Therefore, the *idf* values for the terms are:

log2 ($N/ df i$) ------ (4)

(N: total number of documents)

IDF provides high values for rare words and low values for common words. For example

log(10000/10000) = 0 ,

log(10000/5000) = .301,

log(10000/20) = 2.698 and

log(10000/1) = 4

The next stage is the weighting of the indexed terms to enhance retrieval of document relevant to the user .

$wij = tfij\ idfi = tfij$ log2 (N/ dfi) ------- (4)

Higher weight is given to frequently occurring term but rarely in the rest of the collection. Query vector is also treated as a document and it is also tf-idf weighted. A document should be normalized by document length because of two reasons: long documents have higher term frequencies that have the same term that appears more

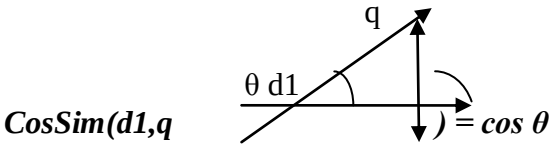
often and more terms increases the number of matches between a document and a query.

The third stage ranks the document with respect to the query according to a similarity measure. A similarity measure is a function that computes the *degree of similarity* between two vectors. A similarity measure is used between the query and each document to rank the retrieved documents in the order of presumed relevance and to enforce a certain threshold so that the size of the retrieved set can be controlled.

For binary vectors, the inner product is the number of matched query terms in the document (size of intersection). For weighted term vectors, it is the sum of the products of the weights of the matched terms. In order to find the vector space proximity, Euclidean distance is a choice but since it results in large value, we use cosine measure for ranking the documents.

Eventhough other coefficients like Jaccard and Dice are available, in this work the cosine coefficient, is used as the similarity which measures the angle between the a document vector and the query vector. Cosine is a monotonically decreasing function for the interval [0^o, 180^o]. It lessens the impact of long documents.

Similarity of a document vector to a query vector = cosine of the angle between them.



$$\text{CosSim}(d_j, q) = \frac{\vec{d_j} \cdot \vec{q}}{\left|\vec{d_j}\right| \cdot \left|\vec{q}\right|} = \frac{\sum_{i=1}^{t} (w_{ij} \cdot w_{iq})}{\sqrt{\sum_{i=1}^{t} w_{ij}^{\ 2} \cdot \sum_{i=1}^{t} w_{iq}^{\ 2}}}$$

Cosine is a normalized dot product. Documents ranked by a decreasing cosine value CosSim(d,q) = 1 when d=q, CosSim(d,q) = 0 when d and q share no terms.

Experimental Results

Consider the document collection with 5 documents

D_1 = The rate of gold is decreasing

D_2= Gold is imported from Dubai

D_3 = Silver is not most wanted

D_4 = Shipment of gold is through flight

D_5 = Gold shipment from Dubai

Query Q = Gold shipment from Dubai

The table shows the term frequency, document frequency, weightage of each term and the query. Each term in the document is sorted in the alphabetical order and the term frequency is tabulated.

The document frequency d_f is calculated based on the number of occurrences of the term in all documents. The inverse document frequency is calculated and it provides high values for rare words and low values for common words. In the same way weightage of the document and the query are also calculated as a vector.

Table 1: Term Vector Model based on W_i

		Count						Weights Wi = tf * idf						
Terms	Q	D_1	D_2	D_3	D_4	D_5	df	idf	Q	D_1	D_2	D_3	D_4	D_5
Decreasing	0	1	0	0	0	0	1	0.6989	0	0.6989	0	0	0	0
Dubai	1	0	1	0	0	0	1	0.6989	0.6989	0	0.6989	0	0	0
Flight	0	0	0	0	1	0	1	0.6989	0	0	0	0	0.6989	0
From	1	0	1	0	0	0	1	0.6989	0.6989	0	0.6989	0	0	0
Gold	1	1	1	0	1	1	4	0.0969	0.0969	0.0969	0.0969	0	0.0969	0.0969
Imported	0	0	1	0	0	0	1	0.6989	0	0	0.6989	0	0	0
Is	0	1	1	1	1	1	5	0	0	0	0	0	0	0
Metal	0	0	0	0	0	1	1	0.6989	0	0	0	0	0	0.6989
Most	0	0	0	0	0	1	1	0.6989	0	0	0	0	0	0.6989
Much	0	0	0	1	0	0	1	0.6989	0	0	0	0.6989	0	0
Not	0	0	0	1	0	0	1	0.6989	0	0	0	0.6989	0	0
Of	0	1	0	0	1	0	1	0.6989	0	0.6989	0	0	0.6989	0
Rate	0	1	0	0	0	0	1	0.6989	0	0.6989	0	0	0	0
Shipment	1	0	0	0	1	0	1	0.6989	0.6989	0	0	0	0.6989	0
Silver	0	0	0	1	0	0	1	0.6989	0	0	0	0.6989	0	0
The	0	1	0	0	0	1	2	0.3979	0	0.3979	0	0	0	0.3979
Through	0	0	0	0	1	0	1	0.6989	0	0	0	0	0.6989	0
Wanted	0	0	0	1	0	1	1	0.6989	0	0	0	0.6989	0	0.6989

Length Normalization

D_1 =Sqrt($(0.6989)^2$+ $(0.0969)^2$ + $(0.6989)^2$ + $(0.6989)^2$ + $(0.3979)^2$)= 1.8432

D_2 = Sqrt($(0.6989)^2$+ $(0.6989)^2$ + $(0.0969)^2$ + $(0.6989)^2$)= 1.6849

D_3= Sqrt($(0.6989)^2$+ $(0.6989)^2$ + $(0.6989)^2$ + $(0.6989)^2$) = 2.1640

D_4 = Sqrt($(0.6989)^2$+ $(0.0969)^2$ + $(0.6989)^2$ + $(0.6989)^2$ + $(0.6989)^2$) = 2.1733

D_5= Sqrt($(0.0969)^2$+ $(0.6989)^2$ + $(0.6989)^2$ + $(0.3979)^2$ + $(0.6989)^2$) = 1.7199

Length of the Query

Q = Sqrt($(0.6989)^2$+ $(0.6989)^2$ + $(0.0969)^2$ + $(0.6989)^2$) = 1.6849

Cosine Similarity Measures

Cosine similarity measures the cosine of the angle between two vectors.

Q * D_1= 0.0969 * 0.0969 = 0.0093

Q * D_2= (0.0969 * 0.0969) + (0.0969 * 0.0969) = 0.4977

Q * D_3 = 0

Q * D_4 = (0.0969 * 0.0969) + (0.6989 * 0.6989) = 0.4977

Q * D_5 = (0.0969 * 0.0969) = 0.0093

$$\text{CosineD1} = \frac{Q \cdot D1x}{|Q|\,|D1|} = \frac{0.0093}{1.6849 * 1.8432} = 0.0029$$

$$\text{CosineD2} = \frac{Q \cdot D2x}{|Q|\,|D2|} = \frac{0.4977}{1.6849 * 1.6949} = 0.1753$$

$$\text{CosineD3} = \frac{Q \cdot D3x}{|Q|\,|D3|} = 0$$

$$\text{CosineD4} = \frac{Q \cdot D4x}{|Q|\,|D4|} = \frac{0.4977}{1.6849 * 2.1799} = 0.1359$$

$$\text{CosineD5} = \frac{Q \cdot D5x}{|Q|\,|D5|} = \frac{0.0093}{1.6849 * 1.7199} = 0.0032$$

According to the similarity values, the final order in which the documents are presented as result to the query will be: d2, d4, d5,d1.

Obseravtions

This experiment illustrates frequent terms such as "a", "in", and "of" tend to receive a low weight -a value of zero in this case. Thus, the model correctly predicts that very common terms, occurring in many documents in a collection are not good discriminators of relevancy. This reasoning is based on global information; ie., the IDF term. Precisely, this is why this model is better than the term count model.

Conclusion

In this paper, a brief introduction to semi supervised clustering with an emphasis on the challenge of clustering high dimensional text data is given. The principal challenge in extending cluster analysis to high dimensional data is to overcome the "curse of dimensionality" . In this work vector space model is used as a feature reduction technique to reduce the feature of the text data by eliminating non content bearing words and it is used to rank the documents according to its relevance. This ranking can be used as the side information for semi supervised clustering. Vector Space Model is a simple, mathematically based approach. It considers both local (*tf*) and global (*idf*) word occurrence frequencies and also provides partial matching and ranked results.

References

[1] Barnes, M., 2001, "Stresses in Solenoids," J. Appl. Phys., 48(5), pp. 2000–2008.

[2] Banerjee, A., Ghosh, J., 2006, "Scalable clustering algorithms with balancing constraints. DataMining and Knowledge Discovery", 13(3).

[3] Daoqiang Zhang, Zhi-Hua Zhou, and Songcan Chen., 2007, "Semisupervised dimensionality reduction" in SIAM International Conference on Data Mining (SDM).

[4] Elsayed, T., Lin, J., & Oard, D., 2008, "Pairwise document similarity in large collections with map reduce". In Proceedings of Association for Computational Linguistics and Human Language Technology Conference 2008 (ACL-08: HLT), Short Papers pp.265–268, Columbus, Ohio. Association for Computational Linguistics.

[5] Erk, K., & Pad´o, S., 2008. "A structured vector space model for word meaning in context". In Proceedings of the 2008 Conference on Empirical

Methods in Natural Language Processing (EMNLP-08) , pp. 897–906, Honolulu, HI.

[6] Hakan Cevikalp, Jakob Verbeek, Fr_ed_eric Jurie, and Alexander Klaser", 2008, "Semi-supervised dimensionality reduction using pairwise equivalence constraints". In International Conference on Computer Vision Theory and Applications, pages 489 -496.

[7] H. P. Kriegel, P. Kroger and A. Zimek, 2009, "Clustering high-dimensional data : A survey on subspace clustering, Pattern-Based Clustering, and Correlation Clustering". ACM Transactions on Knowledge Discovery from Data (TKDD), Vol. 3, Issue 1, Article 1.

[8] J. Mitchell, M. Lapata, 2008, "Vector-based models of semantic composition". In Proceedings of ACL, 236– 244.

[9] Jitendra Nath Singh and Sanjay Kumar Dwivedi, 2012, "Analysis of Vector Space Model in Information Retrieval", *IJCA*, Proceedings on National Conference on Communication Technologies & its impact on Next Generation Computing *2012* CTNGC(2):14-18.

[10] K. Erk., 2007, "A simple, similarity-based model for selectional preferences" In Proceedings of ACL, 216–223.

[11] Michael Steinbach, Levent Ertz, and Vipin Kumar, 2003, "The challenges of clustering high-dimensional data in New Vistas in Statistical Physics: Applications in Econophysics, Bioinformatics, and Pattern Recognition".

[12] Mikhail Bilenko , Sugato Basu , Raymond J. Mooney, 2004, "Integrating Constraints and Metric Learning in Semi-Supervised Clustering", *Proceedings of the 21st International Conference on Machine Learning*, (ICML-2004), pp. 81-88.

[13] Muller, E. , Assesnt, I. , Gunnemann, S. and Seidl, T., 2011, "Scalable Density based Subspace Clustering". Proceedings of the 20th ACM Conference on Information and Knowledge Management (CIKM'11), pp: 1076-1086.

[14] M. R. Ilango and V. Mohan, 2010, "A survey of grid based clustering algorithms", International Journal of Engineering Science and Technology, Vol. 2(8), 3441-3446.

[15] Sunita Jahirabadkar and Parag Kulkarni. Article, 2013 "Clustering for High Dimensional Data: Density based Subspace Clustering Algorithms", International Journal of Computer Applications 63(20):29-35.

[16] Xiaojin Zhu and Andrew B. Goldberg, 2009, "Introduction to Semi-Supervised Learning", Morgan & Claypool Publishers.

[17] X. Zhu, J. Lafferty, and Z. Ghahramani. 2003, "Semi-supervised learning: from Gaussian field to Gaussian processes". Technical Report CMU-CS-03-175, CMU.

[18] Y. H. Chu, J. W. Huang, K. T. Chuang, D. N. Yang and M. S. Chen, 2010, "Density conscious subspace clustering for high dimensional data" IEEE Trans. Knowledge Data Eng. 22: 16-30.