ISSN:2229-6093



# A Two layer semi-supervised Clustering method for text retrieval

Mohammad Darvishi Padook, Eghbal Mansoori, Reza Boostani

*Department of Computer Science, Islamic Azad University, Branch of Gachsaran, Gachsaran, Iran*

## Abstract

*Accurate clustering of text is a challenging problem among the information retrieval society. In some cases experts possesses prior knowledge about the data that can enhance the clustering performance. In this paper a two layer semi-supervised clustering method is proposed to improve the text clustering accuracy. The novel approach uses Space Level Constraints Clustering (SLCC) method as a first layer to categorize the data which novel the prior knowledge for the second layer. K-means clustering is an efficient method but the bottleneck of this algorithm is its sensitivity to the number of clusters and initial centers. K-means is employed as the second layer in the proposed structure and its drawbacks is solved by incorporating prior knowledge found by SLCC (in the first layer) such as number of partitions and their centers. Here Reuters-21578 dataset along with some standard sets from UCI repository are selected as a rich benchmark to evaluate our method. Therefore, accuracy of the clustering methods can be precisely determined. The combinatorial scheme is applied on a high dimensional reuters-21578 data and the clustering results lead to a higher accuracy compare to utilize just SLCC or K-means on the data set and also got high improvement on the other datasets.*

**Keywords:** space level constraints clustering, K-means, text retrieval, semi-supervised clustering.

## Introduction

There has been a growing interest on clustering of text in order to classify the information categories. In this way, many attempts have been performed to grouping text data via clustering methods. Automatic knowledge extraction from text data is possible; hence, semi-supervised methods are suitable for clustering of these data. Existing methods for semi-supervised clustering can be generally divided into three categories [9]; constraint based methods, distance (metric) based methods and hybrid methods. In the first approach, the constraint-based methods which are aimed at guiding the clustering process with pairwise instance constraints [5] or initialize cluster centers by labeled instances [6]. Segal *et al.* [17] describe a model with constraints that combines a binary Markov network derived from pairwise protein interaction data and a Naive Bayes Markov network modeling gene expression data. In the second approach, the distance-based methods employ metric learning techniques to get an adaptive distance measure used in the clustering process based on the given pairwise instance constraints [27]. In distance-based approaches, each clustering method that uses a particular distance function between data points can be employed; however, the distance function is parameterized which should be learned to bring must-linked points together and take cannot-linked points further apart [13,14,15,16]. Finally, the hybrid method proposed by *Basu et al.* [8] and *H. C. Law* [2] unifies these two methods to increase the accuracy of the semi-supervised clustering. Several clustering methods have been introduced and applied to many applications. The most well-known methods are including K-means [3, 5, 25 and 28], Min-Max [29], Single Linkage and Complete Linkage [30], Rival Penalized Competitive Learning (RPCL) [7,31,32], Divisive [33] and Fuzzy-*cmeans* [34]. Among these methods, K-means is a popular clustering algorithm [1] that has been used in a variety of applications such as image segmentation [3], information retrieval [4] and data mining [21]. Due to low complexity and fast convergence of K-means and also its performance in different applications, in this study, a new structure based on K-means is developed with a higher performance. K-means performance suffers from the lack of prior knowledge such as exact number of clusters and their center locations [18]. Iso-data clustering method is an advanced version of k-means which number of clusters can be found adaptively and k-means process is executed several times. Moreover in this scheme clusters can be merged and split but it has several free parameters that finding their optimum values is complicated. To overcome this problem, another algorithm as a pre-clustering in the first layer can be considered and K-means can benefit from the extracted knowledge by the first layer algorithm. To provide this information for K-means, the algorithm in the first layer should be a clustering algorithm; therefore, other state-of-art clustering algorithms are investigated. Ma *et al.* [35, 36] utilized distance-sensitive rival penalized competitive learning (DSRPCL) method to cluster the wine data. They showed DSRPCL can improve the results when the cluster centers are not located in the middle of clusters (local minimum). Klein *et al.* [15] applied Space Level Constraints Clustering (SLCC) method on the SOYBEAN and IRIS data sets and the results provided high clustering accuracy. There have been some researches on the improvement of SLCC [8, 20, 21] which are led to develop different versions of SLCC algorithm. Wagstaff *et al.* [5] presented an instance based SLCC, which was an improved version of K-means and they illustrated that instance based SLCC has a better performance than K-means. Another version of SLCC was developed by Klein *et al.* [15] that used the shortest path algorithm to modify the distance between instances, and then find the initial cluster centers. The combination of prior knowledge and shortest path algorithm is led to the better performance compare to instance based SLCC method [15]. Strong point of SLCC method is the ability of propagating the constraints to perform better clustering (by integrating the prior knowledge of the data with a capable grouping method based on the shortest path method). Therefore in datasets that a class of data clustered in two or more clusters, SLCC can increases the accuracy, but we are still faced with

some drawbacks. First, we do not have the prior knowledge for all data sets, second, if our constraints contain the outlier instances, accuracy is decreased because SLCC propagates incorrect constraint to neighbors of outlier patterns, therefore in this situation its grouping ability is not strong as K-means method. In order to solve this problem we can select the constraints from a subset of the data. In order to improve the SLCC method, a two layer structure has been presented in this paper which improve the clustering results on different data sets compare to the standard version of SLCC method. The combinatorial structure consists of SLCC in the first layer and after clustering the data, K-means in the second layer is applied to the clustered data and the experiments show an improvement achieved by the proposed method compare to utilize just SLCC or K-means on the genes datasets. The rest of this paper is organized as follows. In section 2, Reuters-21578 and UCI repository datasets are described. In section 3, the SLCC and K-means are introduced, in section 4, the modified approach is presented. In section 5, results of applying the mentioned clustering algorithms on the datasets are shown. Finally, the paper will be followed by a discussion and conclusion part.

## 2. Datasets

The Reuters-21578 test collection, together with its earlier variants, has been such a standard benchmark for the text categorization (TC) task[37] throughout the last ten years. Reuters-21578 is a set of 21578 news stories appeared in the Reuters newswire in 1987, which are classified according to 135 thematic categories, mostly concerning business and economy. This collection has several characteristics that make it interesting for TC experimentation:

• Similarly to many other applicative contexts, it is multi-label, i.e. each document $d_i$ may belong to more than one category;

• The set of categories is not exhaustive, i.e. some documents belong to no category at all;

• The distribution of the documents across the categories is highly skewed, in the sense that some categories have very few documents classified under them ("positive examples") while others have thousands;

• There are several semantic relations among the categories (e.g. there is a category WHEAT and a category GRAIN, which are obviously related), but these relations are "hidden" (i.e. there is no explicit hierarchy defined on the categories).

## 3. Clustering Methods

Clustering is a grouping or unsupervised classification method which tries to grouping(cluster) those samples which are similar or their distance is less than a threshold. In this part, SLCC and k-means methods are explained. First, SLCC and its other versions are introduced and then, k-means is briefly discussed and the combinatorial structure is introduced.

### 3.1. SLCC

Space Level Constraints Clustering (SLCC) is a semi-supervised clustering method which uses a hybrid method that benefit from both constrained information and intelligent distance measure. There are two types of constraints; must and cannot link. In the must link constraints each two marked samples should be grouped in one cluster and this grouping scheme is based on the prior knowledge of the problem. In contrast for cannot link constraint, prior knowledge is employed to avoid grouping of two samples in one cluster. Incidentally SLCC uses the shortest path as its metric which can be considered as an adaptive for intelligent metric instance. Regarding to the mentioned properties SLCC is an efficient and flexible scheme to form the primary clusters. The distinct point of SLCC in compare with other clustering methods is utilizing the prior knowledge to form the significant clusters [5, 10].

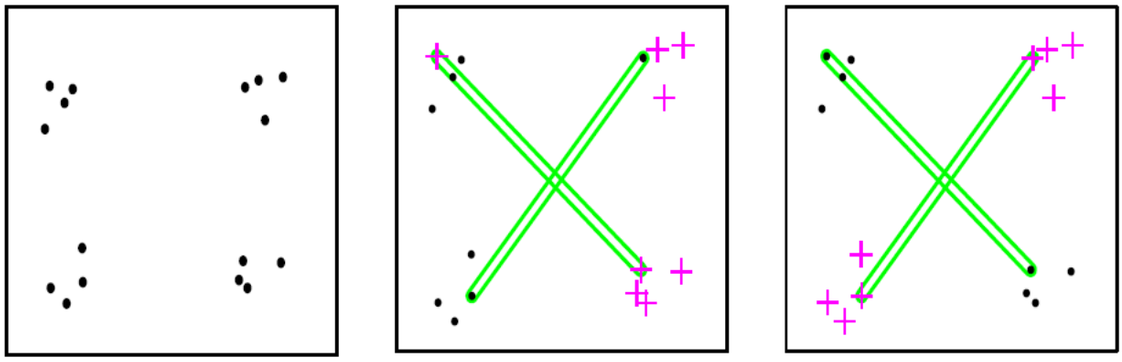
SLCC implementation can be described as follows: **first**; the distance of those sentences which should be grouped in one cluster, is set to zero. **Next**, according to cannot link constraints the distance of those samples which should not be in one cluster, should be infinite. In a **third** stage; propagation method is employed. Which acts according to **shortest path** metric [15] that uses the user defined constraints. As a brief description of SLCC its pseudo code is shown in figure 3.

### 3.2. Instance vs. Space Level Constraints

When we used SLCC, it means that while it is important for a clustering algorithm to satisfy mentioned constraints, also it is important for the algorithm to satisfy the *implications* of those constraints. For example, in figure 1, must link constraints in both sets of clusters (1b and 1c) want to show XOR clusters, but as we see the results in 1b is not logical. This is because the constraints suggest space-level generalizations, therefore not only points that are must-linked, should be in the same cluster, but the points that are near these points should probably also be in the same cluster [15]. Cannot-links have similar spatial generalizations.

### 3-2. Constraint Applicability

It is helpful for clustering algorithm to discuss about the situations that using constraints. As we know if the data are form clusters that are well separated, there is no need for prior knowledge at all. Likewise, if no distinction can be made between classes in feature space, then by using prior knowledge, we can't separate data correctly, and using constraints isn't so helpful. Prior knowledge will therefore be most useful when patterns are at least partially separable, but a clustering algorithm will not detect them correctly without using background knowledge. This situation can arise in many ways.



(a) Data (b) Instance Bias (c) Spatial Bias

Figure 1. The effects of adding two diagonal must-link constraints to th data in (a): an instance-level inductive bias results in single outliers (b) while a stronger space-level bias results in qualitative changes to the clusters (c).

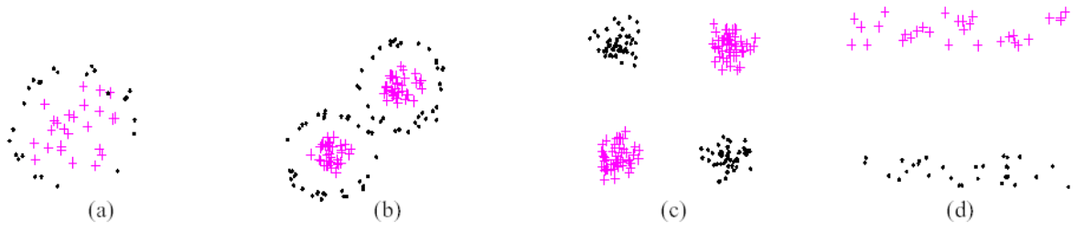



Figure2. Synthetic data sets: target clustering. (a) CIRCLES, (b) TWOCIRCLES, (c) XOR, and (d) STORMCLOUDS

### 3-3. Imposing and Propagating Constraints

The general algorithm is as follows. We have some datasets as input. When we have the instances, we can create the proximity matrix, and with any constraints we give from user, we should propagate the constraints on the matrix and update the proximity matrix. Then we supply this new matrix to a proximity-based clustering algorithm.

In SLCC algorithm it would like specific instances that wants to be in the same class to be very close together, and two instances that wants to put in different classes should be very far apart, and for using this change in proximity matrix, it should increase the distance between two cannot-linked points and decrease the distance between two must-linked points, and propagate the distances between other points with shortest-path algorithm, we know this phase as *imposing constraints.* After this phase the rest of the algorithm is lookalike the instance-level algorithm. After applying the algorithm, *if points $X_i$ and $X_j$ are very close together, then points that are very close to $X_i$ are close to $X_j$ and if points $X_i$ and $X_j$ are very far apart, then points that are very close to $X_i$ are far from $X_j$ .*

To apply the constraints on the proximity matrix, we interpret the proximity matrix as weights for a complete graph over the data points, and we impose must-link constraints by decreasing the distance between the must-linked points to zero and impose all cannot-linked entries to $\max_{i,j}(D_{ij}+1)$ or $\infty$, and allow all other entries to vary.

### 3-4. Clustering

Our new method (SLCC) used the complete-link hierarchical agglomerative clustering [23] (Jain, A. K., & Dubes, R. C. (1988)) as clustering algorithm. Complete-Link (CL) merges clusters in order of proximity; the closest clusters will be merged first, and the furthest clusters will be merged last. By setting the must-link entries in the proximity matrix to 0, and the cannot-link entries to to $\max_{i,j}(D_{ij}+1)$, we can achieve a direct operational (instance-level) interpretation of the constraints .The propagation of the cannot-link constraints occurs through the merges. At each merge, CL creates a *reduced proximity matrix*, with one less row and column.

Because CL defines the distance between clusters as the maximum distance between points in each cluster, if A is cannot-linked to B, merging A and C will cause C to also be cannot-linked to B. In this way, CL achieves implicit propagation of cannot-link constraints.

```
Function two_layer_Clustering (Matrix sp, MustLink mlink, CannotLink notlink, int Min_num_of_classes)
        CompleteLink (Matrix sp, int Min_num_of_classes)
        Centers = Propagate_constraints (sp, mlink, notlink)
        K-means (Centers, sp);
end
Function CompleteLink (Matrix sp, int Min_num_of_classes)
        Clusters = {C_i,  for each point i}
        Linkage starts empty
        Distance (i, j) = sp(i, j);
        While (size(Clusters ) > Min_num_of_classes)
                [ min_row, min_col] = min (sp)
                Add (min_row, min_col) to linkage
               merge (min_row, min_col) to C_new  in Clusters
                for C_i  ∈ Clusters
                        Distance (C_i, , C_new,) = max{ Distance (C_i, , C_1,), Distance (C_i, C_2, )}

Function Propagate_constraints (Matrix sp, MustLink mlink, CannotLink notlink)
        Update_Distance_For_MustLinks (sp, mlink);
        Propagate_MustLinks (sp, mlink);
        Update_Distance_For_CannotLinks (sp, mlink, notlink);
        Propagate_CannotLinks (sp, mlink, notlink);%this function done implicitly by Function CompleteLink.

Function Update_Distance_For_MustLinks (Matrix sp, MustLink mlink)
        for (i, j) ∈ mlink
                sp (i, j ) =0;
                sp (j , i)  = 0;
Function Propagate_MustLinks (Matrix sp,  MustLink mlink)
                        sp = shortest_path(sp, mlink);
                        foreach (i, j) ε  sp
                                if sp(i,j) = 0
                                        mlink = mlink ∪{(i, j)}
Function Update_Distance_For_CannotLinks (sp, mlink, notlink)
        for (i, j) ∈ notlink and (j , k) ∈ mlink
                        sp(i, k) = ∞;
                                sp(k, i) = ∞;
Function sp = Shortest_path(sp, mlink);
    for k = 1: size(mlink)
        for i = 1: size(sp)
            for j = 1 : size(sp)
                sp (i, j)= min{sp(i, j), sp(i, k) + sp(k , j)};
```

figure 3.pseduo code for SLCC-Kmeans method

**3-4 K-means Clustering:**

K-means is a well-known clustering method [1]which have been applied in many clustering and mining applications. The structure of k-means is very simple but its performance is brilliant. The only parameter that k-means takes from the user, is the number of clusters. It starts from k random centers and assign each sample to the nearest centers. After forming the primal clusters, the new centers are determined according to mean of each cluster samples. Again the samples are assigned to the nearest centers and this process continues till cluster centers do not change into successive iteration. The pseudo code of k-means algorithm is shown in fig4.

Nevertheless k-means algorithm suffers from high sensitivity to the initial centers, And also correct number of clusters. In order to overcome this two drawback several variants of the *k*-means algorithm have been reported in different researches [3, 5, 25 and 28]. Some of them attempt to select a good initial partition such that the algorithm is more likely to find the global minimum value. Another version of k-means which is called Iso-data is capable to split and merge the formed clusters in each iteration [24]. Typically, a cluster is split when its variance is above a pre-specified threshold, and two clusters are merged when the distance between their centers is below than a pre-defined threshold. Nevertheless Iso-data performance is seriously affected by the predefined parameters and finding the suitable values for these parameters makes this algorithm too complicated.

**4. The Proposed Clustering Method**

The proposed scheme is designed based on two successive layers. In the first layer, SLCC algorithm applied to input data, regarding to user defined prior knowledge. SLCC categorize the data according to the must-link and cannot- link constraints. After applying the constraints on the data, shortest path tries to form the initial clusters. This distance metric is somehow intelligent, because it considers the constraints when it determines the sample distances. CL acts according agglomerative approach in which each sample considered as a cluster, and iteration by iteration, this cluster are merged to form bigger clusters based on constraints and shortest path. After forming the initial clusters, clustering knowledge is extracted from the input data and applied to the second layer. Know k-means in the second layer is initialized according to the initial clusters which are formed by the first layer. Initial centers for k-means

are found by averaging of each formed clusters by SLCC. The pseudo code of this algorithm is structured in fig3.

**5. Results**

In this section the experimental results on the reuters-21578 dataset along with Iris, Bupa and Glass belong to UCI datasets are presented.

First the combinatorial scheme is applied to the text and UCI datasets. As far as all used datasets are supervised, clustering results can be exactly calculated. To have a fair comparison, k-means and SLCC are separately applied to the described datasets. Experimental results which shown in table 1 show that our method provide higher clustering accuracy in comparison with the two other methods.

`

1) Create an arbitrary initial clustering with centroids $\{\mu_1, \ldots, \mu_k\}$

2) for each object $x_i \in X$ :

(1) Compute $||x_i - \mu_c||$ for $1 \leq C \leq K$

(2) Reassign $x_i$ to the nearest cluster centroid.

3) Update cluster centroids.

4) Repeat from step 2 until a termination criterion is satisfied

*Figure 4. standard k-means algorithm*

| | K-means | SLCC | Combined SLCC-K-means |
|---|---|---|---|
| **Reuters-21578** | **%55(7)** | **%62(3)** | **%63(2)** |
| Iris | **%92.66 (4.7164)** | **%89.33(1)** | **%96.00(1)** |
| Bupa | **%52.43(2)** | **%60.28(2.3)** | **%52.46(0)** |
| Glass | **%45.79(0.3)** | **%50.93(3.7)** | **%51.87(0)** |

Figure 5. Results on different datasets

| *Datasets* | K-means | Spherical-KM | EM | SLCC-KM |
|---|---|---|---|---|
| Reuters-21578 | %55(7) | %55(5) | %50(7) | **%63(2)** |
| EXC | %45(7) | %43(6) | %40(6) | **%47(2)** |
| PEO | %58(4) | %59(5) | %55(7) | **%63(1)** |
| TOP | %62(5) | %62(5) | %56(8) | **%64(2)** |

Figure 6. Results of 4 methods on reuters-21578 and 3 subset of it.

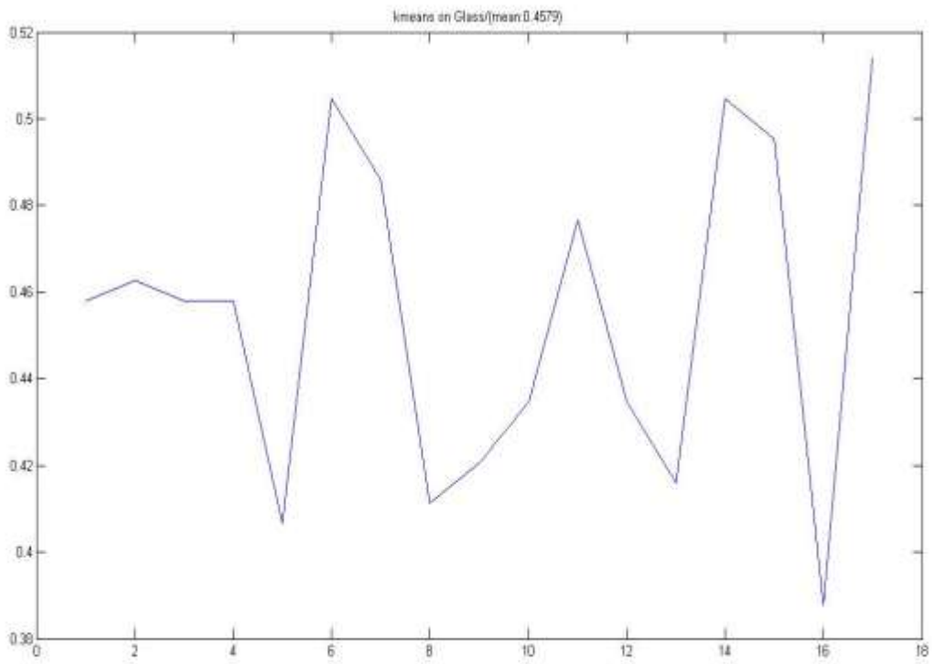



Figure 7: Results of k-means on Glass dataset

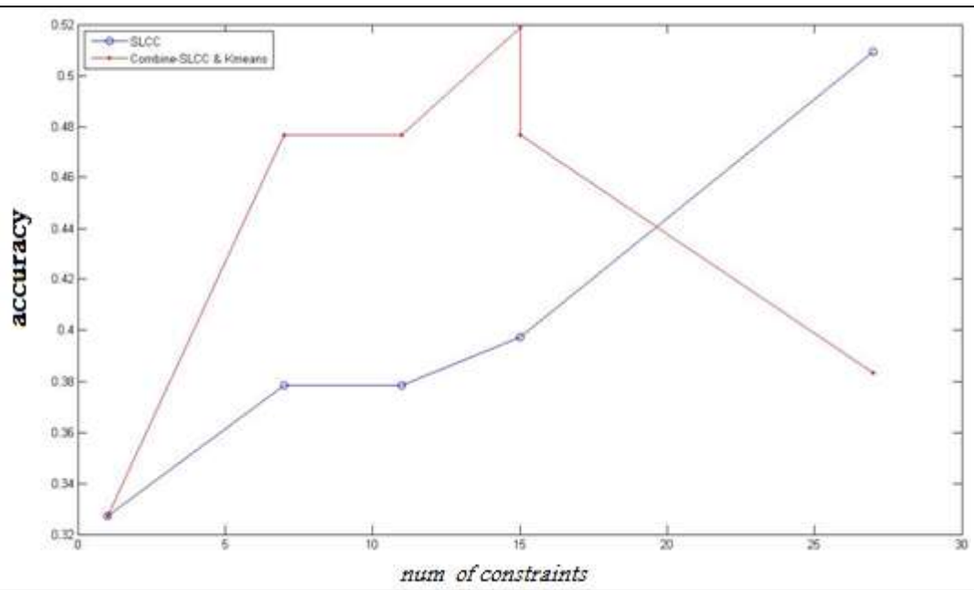



Figure 8: SLCC and SLCC-K-means on Glass dataset

## 6. Discussion and Conclusion

This study is aimed to presenting the new combinatorial clustering method to improve the clustering accuracy for text datasets. As an applicable case Reuters-21578 and 3 of their subsets is considered and the proposed scheme along with the traditional methods such as k-means, spherical k-means, EM and SLCC are applied to this datasets and the results are showed on this datasets. In order to have a better evaluation of our algorithm, some standard datasets from UCI repository set are employed which simulate the variety of real applications. This datasets are selected to cover all complicated situations such as high-dimensional inputs (Bupa dataset), large number of clusters (Glass dataset), noisy data (Glass dataset) and low-dimensional input data (Iris dataset). Experimental results show that in most datasets, our method was superior to the k-means and SLCC, except in the case of Bupa dataset. Although even in this dataset the novel scheme let to higher accuracy compare to k-means method but SLCC behaves better than our algorithm.

The reason can be justified in multimodal datasets prior knowledge plays the very important roles to form the right clusters. But if we remove the prior knowledge each part(modal) of a multimodal class will be considered as a separate cluster. K-means mostly remove this information in multimodal cases, because k-means decide based on the simple distance, not on the intelligent distance, therefore those clusters which are far from each others, are

considers as different classes, while in multimodal case, too far classes can belong to same class.

As it shown, our method performs better than k-means on Bupa dataset, because our method, still carry a part of prior knowledge with itself which provided by SLCC, while standard k-means which does not benefit from any prior knowledge has the lowest performance compare to the two other methods. In conclusion the proposed scheme benefit from both approaches of clustering with and without prior knowledge. In those cases that the constraints are applied to marginal samples (noisy samples) SLCC performance is dramatically dropped, because noisy data absorb the neighbor cluster samples. Therefore clustering methods based prior knowledge and simple clustering methods cannot perform well in different situations, nevertheless our method which is combination of both approaches, can show an acceptable behavior in most cases.

As a future work outlier, this semi-supervised method should be designed such that it automatically discover the noisy patterns on the datasets and do not apply any constraint to them.

This scheme will reduce affect of noisy patterns to absorb the neighbor clustering samples.

Moreover in the cases of multimodal classes, clustering decision should be taken by the prior knowledge Clustering based methods. In other word the decision weight of SLCC in the whole process should be adaptively changed in different situation. This decision weight can also be defined by the user based on the prior knowledge.

Note that SLCC method is a very good method for clustering CIRCLES, TWOCIRCLES, XOR and STORMCLOUDS datasets (figure 8).

## References:

[1] J.B. MacQueen. “Some methods for classification and analysis of multivariate observations”. In Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability, pages 281–297, 1967.

[2] H. C. Law, “Clustering, Dimensionality Reduction, and Side Information,” Submitted to Michigan State University in partial fulfillment of the requirements for the degree of DOCTOR OF PHILOSOPHY, Department of Computer Science & Engineering, 2006.

[3] J. Marroquin, & F. Girosi, (1993). Some extensions of the k-means algorithm for image segmentation and pattern recognition AI Memo 1390). Massachusetts Institute of Technology, Cambridge, MA.

[4] Bellot, P., & El-Beze, M. (1999). “A clustering method for information retrieval (Technical Report IR-0199)”. Laboratoire d'Informatique d'Avignon, France.

[22] M.H. Law, A. Topchy, and A.K. Jain. Model-based clustering with probabilistic constraints. In Proc. SIAM International Conference on Data Mining, pages 641{645, 2005.

[5] K. Wagstaff, C. Cardie, S. Rogers, and S. Schroedl. “Constrained k-means clustering with background knowledge.” In Proc. of the 18th International Conference on Machine Learning (ICML-01), pages 577–584, 2001.

[6] S. Basu, A. Banerjee, and R. J. Mooney. “Semi-supervised clustering by seeding.” In Proc. of the 19th International Conference on Machine Learning (ICML-02), pages 19–25, 2002.

[7] J. Ma and T. Wang, “A Cost-Function Approach to Rival Penalized Competitive Learning (RPCL).”, IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART B: CYBERNETICS, VOL. 36, NO. 4, AUGUST 2006

[8] S. Basu, M. Bilenko, and R. J. Mooney. A probabilistic framework for semi-supervised clustering. In Proc. of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-04), pages 59–68, 2004.

[9] W. Tang, H. Xiong, S. Zhong, and J. Wu, Enhancing semi-Supervised clustering: A Feature Projection Perspective

[10] S. Basu, “Semi-Supervised Clustering with Limited Background Knowledge”, In Proc. *of the Ninth AAAI/SIGART Doctoral Consortium*, pp. 979-980, San Jose, CA, July 2004.

[11] S. Basu,M. Bilenko,A. Banerjee, and R. Mooney, Probabilistic Semi-Supervised Clustering with Constraints, In Semi-Supervised Learning, O. Chapelle, B. Schoelkopf, and A. Zien (eds.), MIT Press, 2006, to appear.

[12] A. Demiriz, K. P. Bennett, and M. J. Embrechts. Semi-supervised clustering using genetic algorithms. In Proceedings of ANNIE, pages 809–814, 1999.

[13] M. Bilenko and R. J. Mooney. Adaptive duplicate detection using learnable string similarity measures. In Proceedings of ACM SIGKDD, pages 39–48, Washington, DC, 2003.

[14] D. Cohn, R. Caruana, and A. McCallum. Semi-supervised clustering with user feedback. Technical Report TR2003-1892, Cornell University, 2003.

[15] D. Klein, S. D. Kamvar, and C. Manning. From instance-level constraints to space-level constraints: Making the most of prior knowledge in data clustering. In Proceedings of ICML, pages 307–314, Sydney, Australia, 2002.

[16] E. P. Xing, A. Y. Ng, M. I. Jordan, and S. Russell. Distance metric learning, with application to clustering with side-information. In NIPS 15, pages 505–512, 2003.

[17] E. Segal, H. Wang, and D. Koller. Discovering molecular pathways from protein interaction and gene expression data. Bioinformatics, 19:i254–i272, July 2003.

[18] G. Fung, a Comprehensive Overview of Basic Clustering Algorithm, June 2001

[19] C. Reina U.M. Fayyad and P.S. Bradley. Initialization of iterative refinement clustering algorithms.

[20] R. J. Kate and R. J. Mooney, Semi-Supervised Learning for Semantic Parsing using Support Vector Machines, In *Proc. of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics, Short Papers (NAACL/HLT-2007)*, pp. 81-84, Rochester, NY, April 2007.

[21] S. Basu, A. Banerjee, and R. J. Mooney. Active Semi-Supervision for Pairwise Constrained Clustering. In Proc. of the *SIAM International Conference on Data Mining*,(SDM-2004), pp. 333-344, Lake Buena Vista, FL, April, 2004.

[23] A. K. Jain, & R. C. Dubes (1988). *Algorithms for clustering data*. Englewood Cliffs, NJ: Prentice Hall.

[24] A.K. JAIN, M.N. MURTY, AND P.J. FLYNN, Data Clustering: A Review
[25] ANDERBERG, M. R. *Cluster Analysis for Applications*. Academic Press, Inc., New York, NY. 1973
[26] Richard O. Duda, David G. Stork, and Peter E. Hart. *Pattern Classification, 2nd ed.* Wiley, New York, NY, 1999.
[27] E. P. Xing, A. Y. Ng, M. I. Jordan, and S. Russell. Distance metric learning with application to clustering with side-information. In Advances in Neural Information Processing Systems 15 (NIPS-02), pages 505–512, 2003.
[28] K. Shin and A. Abraham, Two phase semi-supervised clustering using background knowledge, IDEAL 2006, LNCS 4224, pp. 707 – 712, 2006. Springer-Verlag Berlin Heidelberg 2006
[29] C. H. Q Ding, H. Xiaofeng, Z. Hongyuan, G. Ming and H.D. Simon, A min-max cut algorithm for graph partitioning and data clustering.proc. on IEEE conference on Data mining,2001.
[30] C. A. Glasbey, Complete linkage as a multiple stopping rule for single linkage clustering,journal of Classification, Springer New York ,1987.
[31] L. Xu, A. Krzyzak and E. Oja, "Rival Penalized Competitive Learning for Clustering Analysis, RBF Net,
and Curve Detection," *IEEE Transactions on Neural Networks*, vol. 4, pp. 636-649, 1993.

[32] Y.M. Cheung, Z.H. Lai and L. Xu, "Adaptive Rival Penalized Competitive Learning and Combined Linear Predictor with Application to Financial Investment," *to appear on Proc. of Conference on Computational Intelligence for Financial Engineering (CIFEr)*, 1996.
[33] S.K. Tasoulis and D.K. Tasoulis, " Improving Principal Direction Devisive Clustering," In *The 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2008)*, August 24-27, 2008, Las Vegas, USA.
[34] S. Nascimento , B. Mirkin and F. Moura-Pires, "A Fuzzy Clustering Model of Data and Fuzzy c-Means," The Ninth IEEE international Conference on Fuzzy Systems, may 7-10,2000.Gene Analysis
[35] W. Xiong, Z. Cai and J. Ma, "ADSRPCL-SVM Approach to Informative Gene Analysis", 2008 Beijing Genomics Institute Published by Elsevier Ltd.
[36] J. Ma, T. Wang and L. Xu, "Convergence Analysis of Rival Penalized Competitve Learning(RPCL) Algorithm", Neural Networks,2002,Proceeding of the 2002 International Joint IEEE.
[37] Fabrizio Sebastiani. 2002. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47