Feature Selection based Semi-Supervised Subspace Clustering

V. R. Saraswathy
Assistant Professor(SLG)
Kongu Engineering College
Erode, Tamilnadu, India

N. Kasthuri, Ph.D
Professor
Kongu Engineering College
Erode, Tamilnadu, India

M. Revathi
PG Student
Kongu Engineering College
Erode, Tamilnadu, India

ABSTRACT
Clustering is the process which is used to assign a set of n objects into clusters(groups). Dimensionality reduction techniques help in increasing the accuracy of clustering results by removing redundant and irrelevant dimensions. But, in most of the situations, objects can be related in different ways in different subsets of the dimensions. Dimensionality reduction tends to get rid of such relationship information and generate clusters which do not fully reflect the real cluster's properties. Subspace clustering preserves such relationships by detecting all clusters in all subspaces. The accuracy of the subspace clustering results can be improved by making use of semi-supervised learning method. But finding subspaces by considering all input dimensions may decrease the clustering accuracy. This paper proposes a feature selection based semi-supervised subspace clustering method which applies feature selection in the beginning to eliminate unnecessary dimensions. Later, subspace clustering can be performed on the resulting dataset. This approach tends to improve the accuracy of resulting clusters since subspace clustering is performed on a reduced dataset. Experimental results show that the proposed method produces high quality clusters than semi-supervised subspace clustering algorithm.

General Terms
Dimensionality Reduction, Clustering.

Keywords
Subspace, Feature Selection, Semi-supervised learning.

1. INTRODUCTION
Data mining is a process which extracts patterns that are hidden in large amounts of data. Clustering is an important task in the data mining process. Clustering is the process which is used to assign a set of n objects into clusters(groups) so that the objects lying in the same cluster tends to be closer to each other but the objects in different clusters are far away from each other. The objects are described by a set of attributes (dimensions). Clustering is especially useful in the situation where only a little knowledge about the given dataset is available. Many traditional clustering techniques are available [2].

Traditional clustering algorithms consider all of the dimensions (attribute) of an input dataset in an attempt to learn as much as possible about each object described. Many of the dimensions are often irrelevant in the high dimensional data. These irrelevant dimensions can hide clusters in the noisy data and can confuse the clustering algorithms. If a dataset has very high number of dimensions, all the objects will appear to be nearly equidistant from each other, and thereby will be completely masking the clusters.

One possible extension of conventional clustering algorithm is the application of dimensionality reduction techniques. These approaches lower the dimensionality first either by removing less important dimensions or by transforming the original space to a low dimensional space. Then conventional clustering techniques are applied to the dataset in reduced dimensions. However, because clusters can be formed in different subspaces, such kinds of dimension reduction can get rid of useful dimensional information for some clusters [3]. As a result of lost information, it can generate clusters that may not fully reflect the original cluster's properties.

Subspace clustering is the answer to this challenge. It achieves the clustering goal by allowing clusters to be formed with their own correlated dimensions i.e. an object can be a member of more than one cluster each of which exists in different subspaces (subspace- selected subsets of the dimensions). Subspace clustering is the task of detecting all clusters in all subspaces. Several different subspace clustering algorithms are available [4].

To improve the performance of subspace clustering algorithms, domain knowledge can be applied. Since it is expensive to acquire large amount of domain knowledge, semi-supervised approach (a little amount of domain knowledge) can be used. The domain knowledge is given in the form of either class labels or constraints. Several works have been done to use semi-supervised learning to the problem of dimensionality reduction [5]. But only a small amount of work is done to use semi-supervised learning techniques in the field of subspace clustering [6].

When subspaces are found in a high-dimensional data, the performance of clustering in subspaces may significantly be reduced. This paper proposes a method in which feature selection (relevant dimensions) is first done in high dimensional data. Then the subspaces can be formed in the selected relevant dimensions with a little amount of domain knowledge in the form of constraints. Clustering can be done in the resulting subspaces. This approach tries to improve the quality of the clusters produced.

The rest of the paper is organized as follows. In section 2, related works are reviewed. Section 3 describes the algorithm which is used to develop the model. Experimental results are discussed in section 4. Section 5 concludes the work.

2. LITERATURE REVIEW
2.1. Subspace clustering
There are two types of subspace clustering [4] methods: 1)Top-down search algorithms and 2)Bottom-up search algorithms. C.Aggarwal et al.,[7] proposed a method that samples the data, then selects a set of k medoids and iteratively improves the clustering. C.Aggarwal et al.,[8] proposed another method which is an extended version of

PROCLUS and it looks for non-axis parallel subspaces. K.Woo et al., [3] proposed a method which is similar in structure to PROCLUS and the other top-down methods, but uses a unique distance measure called the dimension oriented distance.

2.2. Semi-supervised clustering

Similarity-adapting based algorithms assume that the initial similarity measure cannot correctly reflect the target classification and has to be modified [9] with the supervision of domain knowledge. In a search-based algorithm, the clustering algorithm is modified so that constraints can be used to bias the search for an appropriate clustering. S.Basu et al.,[10], proposed a method in which a transitive closure of the constraints is performed and used to initialize clusters. K.Wagstaff et al.,[11], proposed a method in which the constraints are forced to be satisfied during the assignment procedure of the clustering process.

2.3. Semi-supervised subspace clustering

Y.Kevin et al., [12] proposed a partitional method similar to the k-medoids algorithm. M.Ahmed et al.,[13] proposed a method that finds clusters in the subspaces of the high dimensional text data where each text document has fuzzy cluster membership. E.Fromont et al., [6] developed an extended framework of bottom-up subspace clustering algorithms by integrating instance level constraints to speed up the enumeration of subspaces. X.Zhang et al.,[1] proposed a method that exploits constraint inconsistence for dimension selection in subspace clustering. This algorithm tries to remove inconsistent constraints from final subspaces.

3. THE PROPOSED METHOD

3.1. Algorithm

1) Find the relevance of each dimension by using the dispersion measure. For any dimension X_i, calculate the arithmetic mean (AM_i) and geometric mean(GM_i)[14]. The AM_i and GM_i are calculated by using the formulae, $AM_i = \frac{1}{n}\sum_{j=1}^{n} X_{ij}$, $GM_i = \left(\prod_{j=1}^{n} X_{ij}\right)^{\frac{1}{n}}$

where n is the number of objects in the dataset. The dispersion measure D_i [14] is given by the ratio, $D_i = \frac{AM_i}{GM_i} \in [1, +\infty)$.

2) Sort the dispersion measure in the descending order. The higher the dispersion measure, the higher will be the relevance of the dimension.

3) Perform cross-validation to decide on the number of dimensions to be selected from the sorted list.

4) Generate constraints. Take two points from same cluster (formed by traditional clustering algorithms) and form the must-link constraint. Take two points from different clusters and form the cannot-link constraint. This process is repeated until the desired number of constraints is generated.

5) Choose consistent dimension for each constraint. This is done by using the concept of K-Nearest neighbors. The consistent dimension for a must-link constraint will have large number of common nearest neighbors. The consistent dimension for a cannot-link constraint will have no common nearest neighbor.

6) Combine corresponding constraints. The correct dimensions will be selected based on the concept of correlation. For correct dimension, the correlation should be 1.

7) Assign data points to the clusters formed in each of the subspaces.

3.2. Basic framework

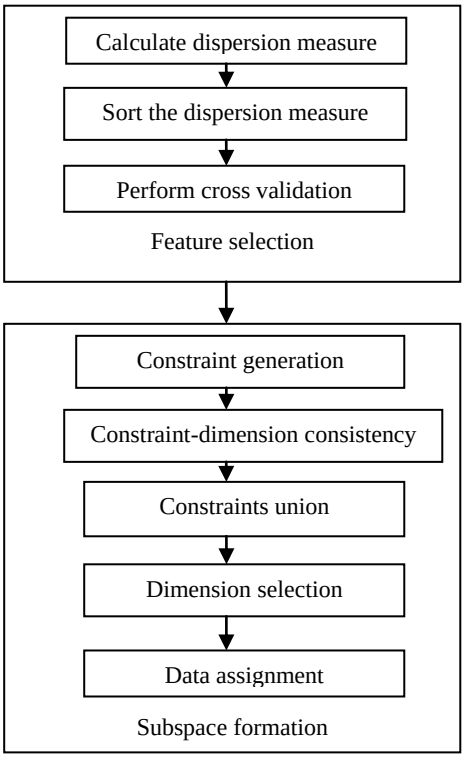

Fig 1: Feature selection based subspace formation

Figure 1 shows the basic framework of the proposed system. The dataset is given as the input to the feature selection module. The output of this module is the dataset with reduced dimensions. The resultant dataset and constraints are given as input to the constraint-dimension consistency module. The constraints along with their correlated dimensions are the output of the constraint-dimension consistency module.

The constraints along with their correlated dimensions are given as input to the constraints union module. The constraints with large number of common correlated dimensions are combined in the constraints union module. The output of this module will be the constraint union and it is given as input to the dimension selection module. The output of this module will be the constraint union along with their correlated dimension. The output of this module is given as input to the data assignment module. The final output will be the subspace clusters.

3.3. Feature selection

The original dataset with large number of dimensions is given as input to the feature selection module. Feature selection is the task of removing irrelevant and redundant dimensions. This method is based on the "principle of parsimony"[15]. This principle states that the system with the smallest number of dimensions can be preferred, which adequately represents the original data. The output of this module will be the dataset with relevant dimensions. The relevant dimensions are found by using the concept of dispersion measure [14]. There are several methods to calculate the dispersion measure. These

methods include mean absolute difference, ratio of arithmetic mean to geometric mean, mean-median. In this paper, the ratio of arithmetic mean to geometric mean is used to calculate the dispersion measure. The dimensions with higher value of dispersion measure will have higher relevance to the clustering process. Cross-validation [14] can be performed to decide on the number of dimensions to be selected so that it can be given as input to the subspace formation module. Cross-validation is the process of estimating how well the model developed from the training data is going to perform with the unseen test data in the future.

3.4. Subspace formation

3.4.1. Constraint generation

The resultant dataset is given as input to any one of the traditional clustering algorithms. K-means clustering algorithm can be used to form the initial clusters from the resultant dataset (relevant dataset). Two points [1] are taken from the same cluster and is used to form the must-link constraints. The must-link constraint says that the two data points must belong to the same cluster. Two points are taken from different cluster and is used to form the cannot-link constraints. The cannot-link constraint says that the two points must be placed in different clusters. This process is continued until the desired numbers of constraints are formed.

3.4.2. Constraint-dimension consistency

The dimensions in which the constraints are consistent [1] should be identified in this module. The consistency of constraints in the dimensions are found based on the methodology that in consistent dimensions, the two points lying on the must-link constraints must have large number of the nearest neighbors in common and also the two points lying on the cannot-link constraints should not have any neighbors in common. In general, the consistency between the constraint and the dimension can be identified by measuring the number of common nearest neighbors they share. The common neighbors between two points lying on a constraint can be found by using the formula [1], (N_K^D represents the number of k nearest neighbors of x_i in dimension D)

$$Sim_{i,j}^{D} = \frac{\left| N_k^D(x_i) \cap N_k^D(x_j) \right|}{k}$$

Using the above value, the correlation between each of the constraints with each of the dimensions can be found. The value of correlation matrix is 0 if the following condition holds [1],

If $\left(T_i \in \overline{M}\right)$ and $\left(Sim_{T_i}^{D_j} < \alpha\right)$ or if $\left(T_i \in \overline{C}\right)$ and $\left(Sim_{T_i}^{D_j} \geq \alpha\right)$

The value of correlation matrix is 1 if the following condition holds [1],

If $\left(T_i \in \overline{M}\right)$ and $\left(Sim_{T_i}^{D_j} \geq \alpha\right)$ or if $\left(T_i \in \overline{C}\right)$ and $\left(Sim_{T_i}^{D_j} < \alpha\right)$, where, T_i represents a constraint which can be either must-link or cannot-link, $\overline{M}$ represents the must-link constraints, $\overline{C}$ represents the cannot-link constraints, $0 \leq \alpha \leq 1$ is the threshold parameter.

3.4.3. Constraints union

Here, the constraints which have a large number of consistent dimensions in common are united. The consistent dimensions are the output of the above step. Each of the resulting constraint union corresponds to a subspace. For uniting the constraints, a method called support degree [1] is used. The support degree of constraint T_i by constraint T_j is given by the formula [1],

$$Sup_{i,j} = \frac{\left| D_{T_i} \cap D_{T_j} \right|}{\left| D_{T_i} \right|}$$

where D_T represents the set of consistent dimensions of a constraint T. Each of the constraints considered is first combined with the constraints that have the maximum support degrees to it. This process is done to produce the constraint unions.

34.4. Dimension selection

The dimensions corresponding to each constraint union [1] should be found and the subspaces are formed with these constraints and their corresponding dimensions. Here, three types of dimensions are identified. They are:

- Backbone dimension: The correlation that is calculated between the backbone dimension and the constraint union will be 1. This backbone dimension should be added to the subspace that corresponds to a constraint union.
- Unrelated dimension: The correlation that is calculated between the unrelated dimension and the constraint union is 0 and this type of dimension should be removed.
- Uncertain dimension: The correlation that is calculated between the uncertain dimension and the constraint union is either 0 or 1. We should decide upon adding this dimension to a constraint union. It is decided by calculating the difference between the mean distance of cannot-link constraints and the mean distance of must-link constraints.

3.4.5. Data assignment

For each of the resulting subspace, the points lying on the constraints [1] will be assigned to the corresponding clusters. The initial clusters are formed from these data points. The initial cluster centers are formed by using the initial cluster members. The unassigned points will be assigned to the cluster whose centroid is the closest to that point. The output of this module will be the subspace clusters.

4. EXPERIMENTAL RESULTS

4.1. Dataset

The datasets are taken from the UCI repository. The ionosphere dataset and spam dataset in the UCI repository are taken for analysis. The ionosphere dataset has 34 attributes and 351 instances. The instances in the dataset belong to two classes. The target here is the free electrons in the ionosphere. If the free electrons exhibit some structure, they are grouped under the name "good". If the free electrons do not exhibit any structure, they are grouped under the name "bad". The spam dataset consists of 57 attributes and 2 classes. Based upon the attribute values, the email should be clustered as "spam" or "not spam". The F1-value of both these datasets are calculated.

4.2. Parameters for evaluation

F1-value is used to access the quality of resulting subspace clusters. Precision and recall are used to calculate the F1-value as follows:

$$\text{F1-value}=\sum_{i=1}^{n_o}\frac{2\times\text{precision}_i\times\text{recall}_i}{\text{precision}_i+\text{recall}_i}$$

where n_0 denotes the number of subspace clusters. Precision is the ratio of the data points correctly predicted in subspace cluster_i to the total data points predicted in subspace cluster_i and recall is the ratio of the data points correctly predicted in subspace cluster_i to the total data points in original cluster_i. The increase in the value of precision and recall will lead to the increased value of F1-value and this results in the increased quality of clusters.

4.3. Experimental results

Table 1. Precision values

constraints	S3C (Ionosphre)	Feature selection based S3C (Ionosphere)	S3C (Spam)	Feature selection based S3C (Spam)
25	0.45	0.47	0.55	0.56
30	0.48	0.52	0.56	0.58
35	0.50	0.53	0.58	0.59
40	0.53	0.55	0.60	0.62
45	0.57	0.58	0.62	0.65

Table 2. Recall values

constraints	S3C (Ionosphre)	Feature selection based S3C (Ionosphere)	S3C (Spam)	Feature selection based S3C (Spam)
25	0.58	0.63	0.64	0.66
30	0.64	0.66	0.67	0.68
35	0.69	0.70	0.70	0.71
40	0.72	0.73	0.74	0.76
45	0.74	0.77	0.78	0.79

Table 3. F1-Values

constraints	S3C (Ionosphere)	Feature selection based S3C (Ionosphere)	S3C (Spam)	Feature selection based S3C (Spam)
25	0.51	0.53	0.63	0.64
30	0.55	0.58	0.68	0.69
35	0.58	0.60	0.70	0.72
40	0.62	0.63	0.73	0.74
45	0.64	0.67	0.76	0.79

Precision, recall and F1-value of Feature selection based Semi-supervised subspace clustering (S3C) is compared with S3C in Table 1, Table 2 and Table 3. Precision and recall is improved in Feature selection based S3C than S3C. This increased values of precision and recall leads to the increased F1-value. Also, it can be noted that the increase in number of constraints leads to the increase in the F1-value of Feature selection based semi-supervised subspace clustering. The increase in number of constraints will improve the quality of the resulting subspace clusters. This shows that the proposed method produces high quality clusters than S3C.

5. CONCLUSION

Feature selection based semi-supervised subspace clustering algorithm overcomes the difficulties in clustering with traditional clustering algorithms. The algorithm also overcomes the difficulties in clustering high dimensional data. This method tries to preserve the relationship among the data points. The resulting cluster will have a high accuracy. The UCI ionosphere dataset and spam dataset are given as input to the proposed algorithm. The output shows precision, recall and F1-value are increased by using Feature selection based S3C compared to S3C. The algorithm can be extended by using the concept of term weight. The algorithm can also be modified by using any methods other than dispersion measure for feature selection.

6. REFERENCES

[1] Zhang, X., Qiu, Y., & Wu, Y. 2011. Exploiting constraint inconsistence for dimension selection in subspace clustering: A semi-supervised approach.Neurocomputing, 74(17), 3598-3608.

[2] Berkhin, P. 2006. A survey of clustering data mining techniques. Grouping multidimensional data, 25-71.

[3] Woo, K. G., Lee, J. H., Kim, M. H., & Lee, Y. J. 2004. FINDIT: a fast and intelligent subspace clustering algorithm using dimension voting. Information and Software Technology, 46(4), 255-271.

[4] Parsons, L., Haque, E., & Liu, H. 2004. Subspace clustering for high dimensional data: a review. ACM SIGKDD Explorations Newsletter, 6(1), 90-105.

[5] Cevikalp, H., Verbeek, J., Jurie, F., & Klaser, A. 2008. Semi-supervised dimensionality reduction using pairwise equivalence constraints. In 3rd International Conference on Computer Vision Theory and Applications (VISAPP'08) (pp. 489-496).

[6] Fromont, E., Robardet, C., & Prado, A. 2009. Constraint-based subspace clustering.

[7] Aggarwal, C. C., Wolf, J. L., Yu, P. S., Procopiuc, C., & Park, J. S. 1999. Fast algorithms for projected clustering. ACM SIGMOD Record, 28(2), 61-72.

[8] Aggarwal, C. C., & Yu, P. S. 2000. Finding generalized projected clusters in high dimensional spaces (Vol. 29, No. 2, pp. 70-81). ACM.

[9] Basu, S., Bilenko, M., & Mooney, R. J. 2004, August. A probabilistic framework for semi-supervised clustering. In Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 59-68). ACM.

[10] Basu, S., Banerjee, A., & Mooney, R. 2002, July. Semi-supervised clustering by seeding. In Machine Learning International Workshop then Conference - (pp. 19-26).

[11] Wagstaff, K., & Cardie, C. 2000, June. Clustering with instance-level constraints. In Proceedings of the National Conference on Artificial Intelligence (pp. 1097-1097).

Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999.

[12] Yip, K. P., Cheung, D. W., & Ng, M. K. 2005, April. On discovery of extremely low-dimensional clusters using semi-supervised projected clustering. In Data Engineering, 2005. ICDE 2005. Proceedings. 21st International Conference on(pp. 329-340). IEEE.

[13] Ahmed, M. S., & Khan, L. 2009, December. Sisc: A text classification approach using semi supervised subspace clustering. In Data Mining Workshops, 2009. ICDMW'09. IEEE International Conference on (pp. 1-6). IEEE.

[14] Ferreira, A. J., & Figueiredo, M. A. 2012. Efficient feature selection filters for high-dimensional data. Pattern Recognition Letters.

[15] Hansen, M. H., & Yu, B. 2001. Model selection and the principle of minimum description length. Journal of the American Statistical Association, 96(454), 746-774.