Enhancing Semi Supervised Clustering Algorithm for High Dimensional Data

M. Pavithra and R.M.S. Parvathi

Assistant Professor, Department C.S.E, Jansons Institute of Technology, Coimbatore, India
Dean PG-Studies, Sri Ramakrishna Institute of Technology, Coimbatore, India

Abstract: Semi-supervised clustering employs limited supervision in the form of labeled instances or pairwise instance constraints to aid unsupervised clustering and often significantly improves the clustering performance. Despite the vast amount of expert knowledge spent on this problem, most existing work is not designed for handling high-dimensional sparse data. This paper thus fills this crucial void by developing a Semi-supervised Clustering method based on spherical KMeans via feature projection (SCREEN). Specifically, we formulate the problem of constraint-guided feature projection, which can be nicely integrated with semi-supervised clustering algorithms and has the ability to effectively reduce data dimension. Indeed, our experimental results on several real-world data sets show that the SCREEN methods can effectively deal with high-dimensional data and provides an appealing clustering performance. Clustering which tries to group a set of points into clusters such that points in the same cluster are more similar to each other than points in different clusters, under a particular similarity metric. In the generative clustering model, a parametric form of data generation is assumed and the goal in the maximum likelihood formulation is to find the parameters that maximize the probability (likelihood) of generation of the data. In the most general formulation, the number of clusters k is also considered to be an unknown parameter.

Key words: Data Mining • Knowledge Discovery in Databases • Clustering • Semi Supervised Clustering • High Dimensional Data

INTRODUCTION

A recent trend in machine learning research is to combine the techniques developed for unsupervised learning and supervised learning to handle datasets with partial external information. One of the foci is semi-supervised clustering, which actively uses the available domain knowledge in guiding the clustering process. These methods can be categorized according to the kinds of knowledge being input, the time that the knowledge is input and the way the knowledge is used to affect the clustering process. The simplest type of input is labeled objects [1]. In some cases, users do not know the exact class labels of objects, but they have some knowledge on which objects should be/should not be put into the same cluster, which can be specified by must-links and cannot-links [2].

Some other studies propose the input of classification rules [3], examples of similar objects [4], or even general comments like which cluster a particular object should not be put into [5]. The knowledge can be supplied at different time. It can be supplied before clustering to guide the clustering process [6], or after clustering to evaluate the clusters and guide the next round of clustering [7]. Some algorithms can also actively request users to supply some specific information at the most appropriate time [8]. There are various ways to use the input knowledge, such as guiding the formation of seed clusters [9], forcing or recommending some objects to be put in the same cluster or different clusters [10] and modifying the objective function, similarity function or distance matrix [11].

The algorithms perform well when each cluster is in the form of a hypercube and the parameter values are specified correctly, but in many cases these requirements cannot be met and the clustering results are quite unsatisfactory [12]. The number of seeds and neighboring objects required to try can also be so large that causes the algorithms to run for a long time.

The outputs of the algorithm are k clusters and their selected dimensions and a (possibly empty) set of outliers. The goal is to optimize an objective function
whose value (the objective score) reflects the quality of
the clusters. In the non-projected clustering algorithm
k-means [13], the objective function is defined as the total
within-cluster squared error. It can be shown that the
partition of objects that minimize the function
corresponds to the maximum likelihood hypothesis of
the above model when there are no irrelevant dimensions [14].
In [15], the objective function is modified for projected
clustering such that only relevant dimensions are
involved in the distance calculations and the part of
objective score from each cluster is normalized by the
number of selected dimensions. Due to the normalization,
the function tends to gives better (i.e., smaller) scores for
clusters with fewer selected dimensions [1], which forces
the algorithm to request users to supply the average
cluster dimensionality in order not to select only one
dimension per cluster. Also, as the function is based on
the summation of variances among different dimensions,
a worse dimension (one with larger variance) constitutes
more to the objective score.

**Related Work:** Semi-supervised Clustering: SSL aims at
enhancing the performance of classification systems by
exploiting an additional set of unlabeled data. Due to its
great practical value, SSL has a rich literature [2].
Amongst existing methods, the simplest methodology for
SSL is based on the self-training scheme [3] where the
system iterates between training classification models
with current ‘labeled’ training data and augmenting the
training set by adding its highly confident predictions in
the set of unlabeled data; the process starts from human
labeled data and stops until some termination condition is
reached, e.g. the maximum number of iterations. [4] and [5]
presented two methods in this stream for image
classification. While obtaining promising results, they
both require additional supervision: [6] need image tags
and [7] image attributes. The second group of SSL
methods is based on label propagation over a graph,
where nodes represent data examples and edges reflect
their similarities. The optimal labels are those that are
maximally consistent with the supervised class labels and
the graph structure. Well known examples include
Harmonic-Function [8], Local-Global Consistency [9] and
While having strong theoretical support, these methods
are unable to exploit the power of discriminative learning
for image classification. Another group of methods utilize
the unlabeled data to regularize the classifying functions
− enforcing the boundaries to pass through regions with
a low density of data samples. The most notable methods
are transductive SVMs [12], Semi-supervised SVMs and
semi-supervised random forests [13]. These methods have
difficulties to extend to large-scale applications and
developing an efficient optimization for the miss
still an open question. Readers are referred to [14] for a
thorough overview of SSL.

Existing semi-supervised clustering algorithms can be
classified into three categories: partitional, one cluster at
a time and hierarchical. The partitional approach
PROCLUS [1] is based on the traditional k-medoids
approach [2], with a goal of minimizing the average within-
class dispersion. The distance between different cluster
members is computed in the relevant subspace of the
cluster, which is determined by measuring the average
distance between the medoid and a set of “neighboring
objects” that are close to it when all dimensions are
considered.

To find a cluster, an object is randomly selected as
the seed and some other objects are randomly sampled to
determine the relevant subspace of the cluster. A
dimension is regarded as relevant to the cluster if all the
objects are within a distance $r_0$ from the seed along the
dimension. Each cluster is thus a hypercube of width $2r_0$.
The more objects and relevant dimensions a cluster has,
the less likely it is formed by chance and thus it receives
a better score. The relative importance between the
number of objects and relevant dimensions is controlled
by a user parameter $\beta$. The algorithm repeatedly tries
different seeds and neighboring objects and returns the
cluster with the highest score. Then the whole process
will be repeated for a new cluster. The algorithms perform
well when each cluster is in the form of a hypercube and
the parameter values are specified correctly, but in many
cases these requirements cannot be met and the
clustering results are quite unsatisfactory [3]. The number
of seeds and neighboring objects required to try can also
be so large that causes the algorithms to run for a long
time.

A recent trend in machine learning research is to
combine the techniques developed for unsupervised
learning and supervised learning to handle datasets with
partial external information. One of the foci is semi-
supervised clustering, which actively uses the available
domain knowledge in guiding the clustering process.
These methods can be categorized according to the kinds
of knowledge being input, the time that the knowledge is
input and the way the knowledge is used to affect the
clustering process. The simplest type of input is labeled
objects [4]. In some cases, users do not know the exact
class labels of objects, but they have some knowledge on
which objects should be/should not be put into the same cluster, which can be specified by must-links and cannot-links [5]. Some other studies propose the input of classification rules [6], examples of similar objects [7], or even general comments like which cluster a particular object should not be put into [8].

Proposed Work
Semi-Supervised Projected Clustering: The outline of the new algorithm SSPC (Semi-Supervised Projected Clustering) is shown in Listing 2. It is a partitional method similar to the k-medoids algorithms [9]. At the beginning it determines some seeds (potential medoids) and each cluster draws a medoid from them. Every object in the dataset is then assigned to the cluster that gives the greatest improvement to the objective score, where the value of $\tilde{\mu}_{ij}$ in Equation 4 is temporarily substituted by the projection of the medoid on $v_j$. If an object does not improve the $o_i$ score of any cluster, it will be put on the outlier list. After assigning all objects, the selected dimensions of each cluster are redetermined and the overall objective score is computed using the actual medians.

The Outline of Sspc Algorithm:
1. Initialization: determine the seeds and relevant dimensions of each cluster
2. For each cluster, draw a medoid from the seeds
3. Assign every object in the dataset to the cluster (or outlier list) that gives the greatest improvement to the objective score
4. Call Select Dim (Ci) for each cluster Ci and calculate the overall objective score
5. Record the clusters if they give the best objective score so far, restore the best clusters otherwise
6. Replace the cluster representative of each cluster and then remove its members
7. Repeat 3-6 until no score improvements are observed for a certain number of iterations.

Cluster Labeling by Majority (CLM): Cluster labeling by majority. The post-labeling method presented hereafter is called cluster labeling by majority (CLM) [10] and is described in Algorithm 3. It is composed of three steps.

- The first step consists in labeling all clusters containing at least one labeled sample. The label assigned to each of these clusters is the majority class of all labeled objects of the cluster.
- The second step labels the clusters containing no labeled sample with the label of the most similar already labeled cluster. The similarity measure $\Delta(K_i,K_k)$ depends on the clustering method and estimates the similarity between two clusters $K_i$ and $K_k$.
- Finally, in the third step, the final classifier is build according to all the new labeled objects.

The Outline Ofcluster Labeling by Majority (CLM) Algorithm:
1: build a clustering $K = \{K_l|l = 1,...,k\}$ on $X$
2: let $L_K = \emptyset$
3: for all $K_l, l = 1,...,k$ do
4: if $P_{q_j=1} c_{lj} \neq 0$ then
5: $y_{Kl} = \arg\max_{j \in \{1,...,q\}} \{c_{lj}\}$
6: $W_l = \{(x_i,y_{Kl}),x_i \in K_l\}$
7: $L_K := L_K \cup K_l$
8: end if
9: end for
10: for all $K_u: P_{q_j=1} c_{uj} = 0$ do
11: $K_m = \arg\max K_l \in L_K (\Delta(K_l,K_u))$
12: label all objects in cluster $K_u$ with label $y_{Km}$
13: $W_u = \{(x_i,y_{Km}),x_i \in K_u\}$
14: end for
15: build the final classifier $CW_1 \cup \ldots \cup W_k$

Semi-Supervised Clustering Enhanced by Multiple Clustering’s (SLEMC): Semi-supervised learning enhanced by multiple clustering’s the method that we propose, called Semi-supervised learning enhanced by multiple clustering’s (SLEMC), could be categorized as a post-labeling method. Indeed, it tries to improve the classification by first producing a clustering of the dataset. The clustering, computed on all the labeled and unlabeled objects, regroups the similar instances together, maximizing the intracluster similarity and the intercluster dissimilarity. If the classes are well separated in the feature space, we should be able to associate to each cluster one of the classes, using the class of the labeled samples belonging to the cluster. Unfortunately, in real world problems, classes are generally not well separated. It is then possible to have samples from different classes in one cluster, or no sample in others. To address this issue, the proposed method uses a combination of multiple clustering’s.

$v(x_i) = (d_1^i,...,d_p^i,K_1^i,...,K_p^i,y_1^i)$

The Outline of Semi-supervised Clustering Enhanced by Multiple Clustering’s (Slemc) Algorithm:
1: apply $b$ clustering algorithms $\{C_k\}_{1 \leq k \leq b}$ to the dataset $X$
2: each method $C_k$ produces a partition $K_k = \{K_1^k,...,K_n^k\}$ of the $n$ objects
3: for all (x_i, y_i) ∈ L do
4: v (x_i) = a_1 ... a_p, K_1 ... K_b, y_i
5: end for
6: apply a supervised learning method to produce a predictive model CV from V = {v (x_i)} m = 1

7: affect the features vector v_0 (x_j) = a_1 ... a_j, p, K_j ... K_j, b, 0 to each x_j ∈ U
8: use CV to label all objects of U.

The Framework of the Screen Algorithm: In the previous work [2], the pairwise constraints were used for learning an adaptive metric between the prototype of instances. However, learning a distance metric among high-dimensional instances is very time consuming. More importantly, recent research on high-dimensional space has shown that the concept of distance in high-dimensional space may not be meaningful [3]. Instead of using constraint-guided metric learning, in this paper we propose a constraint-guided feature projection approach (SCREENPROJ) to further improve the performance of semi-supervised clustering in the high-dimensional datasets. The objective is to learn the projection matrix F_d×k = {F_1,...,F_k} containing k orthogonal unit-length d-dimensional vectors, which can project the original datasets into a low-dimensional space such that the distance between any pair of instances involved in the cannot-link constraints are maximized while the distance between any pair of instances involved in the must-link constraints are minimized. The objective function we try to maximize.

\[ f = \sum_{(x_1, x_2) \in C_{GL}} || F^T (x_1 - x_2) ||^2 - \sum_{(x_1, x_2) \in C_{UL}} || F^T (x_1 - x_2) ||^2 \]

\[ f = \sum_{(x_1, x_2) \in G'_{GL}} || w_2 F^T (x_1 - x_2) ||^2 \]

HMRF Model for Semi-supervised Clustering: Partitional prototype-based clustering is the underlying unsupervised clustering setting under consideration. In such a setting, a set of data points is partitioned into a pre-specified number of clusters, where each cluster has a representative (or “prototype”), so that a well-defined cost function, involving a distortion measure between the points and the cluster representatives, is minimized.

The Hidden Markov Random Field (HMRF) probabilistic framework for semi-supervised constrained clustering consists of the following components:

- An observable set X = (x_1,...,x_n) corresponding to the given data points X. Note that we overload notation and use X to refer to both the given set of data points and their corresponding random variables.
- An unobservable (hidden) set Y = (y_1,...,y_n) corresponding to cluster assignments of points in X. Each hidden variable y_i encodes the cluster label of the point x_i and takes values from the set of cluster indices (1,...,K).
- An unobservable (hidden) set of generative model parameters \( \Theta \), which consists of distortion measure parameters \( A \) and cluster representatives \( M = (\mu_1,...,\mu_K) \): \( \hat{E} = \{A,M\} \).

Step1 Initialization
Step2 Constraint-guided feature projection
Step3 Constrained Spherical k-means on projected data

SCREENranksthirdduetheextra cost of feature projection.
SCREENismuchfasterthanthePCSKM+Malgorithmwhichemploysmetric learninginthehighdimensionaldata.
**Experiments:** In this section, we empirically demonstrate that our proposed semi-supervised clustering algorithm is both efficient and effective.

**Datasets:** Four real-world datasets with varied sizes are used in our experiments, which are:

- **Iris,** This is perhaps the best known database to be found in the pattern recognition literature. Fisher's paper is a classic in the field and is referenced frequently to this day. (See Duda & Hart, for example.) The data set contains 3 classes of 50 instances each, where each class refers to a type of iris plant. One class is linearly separable from the other 2; the latter are NOT linearly separable from each other.
- **Ionosphere,** This radar data was collected by a system in Goose Bay, Labrador. This system consists of a phased array of 16 high-frequency antennas with a total transmitted power on the order of 6.4 kilowatts. Received signals were processed using an autocorrelation function whose arguments are the time of a pulse and the pulse number. There were 17 pulse numbers for the Goose Bay system. Instances in this database are described by 2 attributes per pulse number, corresponding to the complex values returned by the function resulting from the complex electromagnetic signal.
- **Mushroom,** This data set includes descriptions of hypothetical samples corresponding to 23 species of gilled mushrooms in the Agaricus and Lepiota Family (pp. 500-525). Each species is identified as definitely edible, definitely poisonous, or of unknown edibility and not recommended. This latter class was combined with the poisonous one. The Guide clearly states that there is no simple rule for determining the edibility of a mushroom; no rule like "leaflets three, let it be" for Poisonous Oak and Ivy.
- **Primary Tumor,** All attribute values in the database have been entered as numeric values corresponding to their index in the list of attribute values for that attribute domain as given below.
  1. class: lung, head & neck, esophasus, thyroid, stomach, duoden & sm.int, colon, rectum, anus, salivary glands, pancreas, gallblader, liver, kidney, bladder, testis, prostate, ovary, corpus uteri, cervix uteri, vagina, breast.

**EXPERIMENTAL RESULTS**

**IRIS DATASET RESULTS**

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Accuracy</th>
<th>Precision</th>
<th>Recall</th>
<th>F-Measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSPC</td>
<td>90.67</td>
<td>84.98</td>
<td>80.91</td>
<td>81.34</td>
</tr>
<tr>
<td>CLM</td>
<td>89.11</td>
<td>78.08</td>
<td>79.08</td>
<td>85.56</td>
</tr>
<tr>
<td>SLEMC</td>
<td>87.67</td>
<td>73.67</td>
<td>73.67</td>
<td>84.23</td>
</tr>
<tr>
<td>SCREEN</td>
<td>92.11</td>
<td>82.67</td>
<td>89.67</td>
<td>88.67</td>
</tr>
<tr>
<td>HMRF</td>
<td>90.28</td>
<td>87.78</td>
<td>85.88</td>
<td>75.89</td>
</tr>
</tbody>
</table>

The above graph shows that performance of Iris dataset. The Accuracy of SCREEN algorithm is 92.11 which is higher when compare to other three (SSPC, CLM, SLEMC, HMRF) algorithms. The Precision of HMRF algorithm is 87.78 which is higher when compare to other three (SSPC, CLM, SLEMC, SCREEN) algorithms. The Recall of SCREEN algorithm is 89.67 which is higher when compare to other three (SSPC, CLM, SLEMC, HMRF) algorithms. The F-Measure of SCREEN algorithm is 88.67 which is higher when compare to other three (SSPC, CLM, SLEMC, HMRF) algorithms.
IONOSPHERE DATASET RESULTS

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Accuracy</th>
<th>Precision</th>
<th>Recall</th>
<th>F-Measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSPC</td>
<td>91.93</td>
<td>80.45</td>
<td>89.45</td>
<td>79.68</td>
</tr>
<tr>
<td>CLM</td>
<td>89.90</td>
<td>74.91</td>
<td>79.93</td>
<td>85.78</td>
</tr>
<tr>
<td>SLEMC</td>
<td>90.92</td>
<td>69.92</td>
<td>70.94</td>
<td>80.56</td>
</tr>
<tr>
<td>SCREEN</td>
<td>89.67</td>
<td>88.67</td>
<td>84.67</td>
<td>84.78</td>
</tr>
<tr>
<td>HMRF</td>
<td>88.13</td>
<td>72.78</td>
<td>86.78</td>
<td>83.92</td>
</tr>
</tbody>
</table>

The above graph shows that performance of Iris dataset. The Accuracy of SSPC algorithm is 91.93 which is higher when compare to other four (SCREEN, CLM, SLEMC, HMRF) algorithms. The Precision of SCREEN algorithm is 88.67 which is higher when compare to other four (SSPC, CLM, SLEMC, HMRF) algorithms. The Recall of SSPC algorithm is 89.45 which is higher when compare to other four (SCREEN, CLM, SLEMC, HMRF) algorithms. The F-Measure of CLM algorithm is 85.78 which is higher when compare to other four (SSPC, SCREEN, SLEMC, HMRF) algorithms.

MUSHROOM DATASET RESULTS

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Accuracy</th>
<th>Precision</th>
<th>Recall</th>
<th>F-Measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSPC</td>
<td>91.67</td>
<td>80.98</td>
<td>82.67</td>
<td>81.34</td>
</tr>
<tr>
<td>CLM</td>
<td>79.11</td>
<td>76.88</td>
<td>76.88</td>
<td>86.56</td>
</tr>
<tr>
<td>SLEMC</td>
<td>88.67</td>
<td>78.97</td>
<td>80.67</td>
<td>84.23</td>
</tr>
<tr>
<td>SCREEN</td>
<td>92.33</td>
<td>83.67</td>
<td>86.67</td>
<td>89.67</td>
</tr>
<tr>
<td>HMRF</td>
<td>90.98</td>
<td>87.78</td>
<td>89.48</td>
<td>80.89</td>
</tr>
</tbody>
</table>

The above graph shows that performance of Iris dataset. The Accuracy of SSPC algorithm is 91.67 which is higher when compare to other four (SCREEN, CLM, SLEMC, HMRF) algorithms. The Precision of HMRF algorithm is 87.78 which is higher when compare to other four (SSPC, CLM, SLEMC, HMRF) algorithms. The Recall of SSPC algorithm is 89.45 which is higher when compare to other four (SCREEN, CLM, SLEMC, HMRF) algorithms. The F-Measure of SCREEN algorithm is 89.67 which is higher when compare to other four (SSPC, SCREEN, SLEMC, HMRF) algorithms.
PRIMARY TUMOR DATASET RESULTS

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Accuracy</th>
<th>Precision</th>
<th>Recall</th>
<th>F-Measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSPC</td>
<td>90.67</td>
<td>84.98</td>
<td>80.91</td>
<td>81.34</td>
</tr>
<tr>
<td>CLM</td>
<td>71.11</td>
<td>78.08</td>
<td>79.08</td>
<td>85.56</td>
</tr>
<tr>
<td>SLEMC</td>
<td>87.67</td>
<td>73.67</td>
<td>73.67</td>
<td>84.23</td>
</tr>
<tr>
<td>SCREEN</td>
<td>92.11</td>
<td>82.67</td>
<td>89.67</td>
<td>88.67</td>
</tr>
<tr>
<td>HMRF</td>
<td>90.28</td>
<td>88.48</td>
<td>85.88</td>
<td>75.89</td>
</tr>
</tbody>
</table>

The above graph shows that performance of Iris dataset. The Accuracy of SCREEN algorithm is 92.11 which is higher when compare to other four (SSPC, CLM, SLEMC, HMRF) algorithms. The Precision of HMRF algorithm is 88.48 which is higher when compare to other four (SSPC, CLM, SLEMC, SCREEN) algorithms. The Recall of SCREEN algorithm is 89.67 which is higher when compare to other four (SSPC, CLM, SLEMC, HMRF) algorithms. The F-Measure of SCREEN algorithm is 88.67 which is higher when compare to other four (SSPC, SCREEN, SLEMC, HMRF) algorithms.

CONCLUSION

We have proposed a new projected clustering algorithm that is robust and is able to detect clusters of extremely low dimensionality as it uses a robust objective function and avoids distance calculations that involve all the dimensions. In addition, we have proposed ways to utilize any available domain knowledge in the form of labeled objects and labeled dimensions. Experimental results show that there is a clear accuracy improvement when some input knowledge is incorporated in the clustering process. The peak performance is readily reached when only a small amount of knowledge is supplied and when the knowledge covers only some of the classes. There are some obvious directions for further study. The most important one is to test the new algorithm on some real datasets that are expected to contain projected clusters, such as gene expression profiles. When applying to complex, noisy real data, the data model and objective function may have to be revised according to the observed data properties.

It relies on Hidden Random Markov Fields (HMRFs) to utilize both unlabeled data and supervision in the form of pairwise constraints during the clustering process. The framework can be used with a number of distortion (distance) measures, including Bregman divergences and directional measures and it facilitates training the distance parameters to adapt to specific datasets. An algorithm HMRF-KMeans for performing clustering in this framework has been presented that incorporates pairwise supervision in different stages of the clustering: initialization, cluster assignment and parameter estimation. Three particular instantiations of the algorithm, based on different distortion measures, have been discussed: squared Euclidean distance, which is common for clustering low-dimensional data and KL divergence and cosine distance, which are popular for clustering high-dimensional directional data. Finally, a new method has been presented for acquiring supervision from a user in the form of effective pairwise constraints for semi-supervised clustering – such an active learning algorithm would be useful in an interactive query-driven clustering framework. The HMRF model can be viewed as a unification of constrained-based and distance-based semi-supervised clustering approaches. It can be expanded to a more general setting where every cluster has a corresponding distinct distortion measure, leading to a clustering algorithm that can identify clusters of different shapes.
REFERENCES


