An Efficient Semi Supervised Clustering Techniques with Pairwise Constraints

M. Pavithra[1], Dr.R.M.S.Parvathi [2]

Assistant Professor, Department of C.S.E, Jansons Institute of Technology, Coimbatore, India[1].
Dean- PG Studies, *Sri Ramakrishna Institute of Technology, Coimbatore, India [2]*

Abstract
Semi-supervised clustering leverages side information such as pairwise constraints to guide clustering procedures. Despite promising progress, existing semi-supervised clustering approaches overlook the condition of side information being generated sequentially, which a natural setting is arising in numerous real-world applications such as social network and e-commerce system analysis. We consider the semi-supervised clustering problem where we know (with varying degree of certainty) that some sample pairs are (or are not) in the same class. Unlike previous efforts in adapting clustering algorithms to incorporate those pairwise relations, our work is based on a discriminative model. According to the principle of ensemble clustering, the optimal partition lies in the convex hull, and can thus be uniquely represented by an m-dimensional probability simplex vector. As such, the dynamic semi-supervised clustering problem is simplified to the problem of updating a probability simplex vector subject to the newly received pairwise constraints. We then develop a computationally efficient updating procedure to update the probability simplex vector in O (m2) time, irrespective of the data size n. Our empirical studies on several real-world benchmark datasets show that the proposed algorithm outperforms the state of-the-art semi-supervised clustering algorithms with visible performance gain and significantly reduced running time.

Keywords:
Data Mining, Knowledge Discovery in Databases, Clustering, Semi Supervised Clustering, Pairwise Constraints.

1. Introduction

In recent times, majority of the data available throughout the world are warehoused in databases. Data mining that has received immense attention from the research community because of its importance is the process of detecting patterns from extremely huge quantities of data collection [1]. Knowledge Discovery in Databases (KDD) is the other name for data mining which has been identified as a potential field for database research. Classification or bunching of these data into a set of categories or clusters is one of the essential methods in manipulating these data. Clustering is a delineative task that attempts to detect similar category of objects based on the implications of their features dimensions. One can detect the predominant distribution patterns and interesting correlations that exist among data attributes by clustering which can determine dense and sparse areas [4, 5].

Despite the promising progress, one issue often overlooked by existing semi-supervised clustering approaches is how to efficiently update the clustering results when the pairwise constraints are dynamic, i.e., the new pairwise constraints are generated sequentially. This condition stands natural and is closely related to many real-world applications. For example, one representative application in social network analysis is to identify user communities based on users' profiles as well as their social connections. If we respectively treat user profiles and connections as features and pairwise constraints, this application is essentially a semi-supervised clustering problem. Since new connections are being formed over time, user communities should also be frequently updated. Similar situations also occur in various real-world ecommerce platforms, which typically require to group items or customers based on their profiles (i.e., features) and dynamic co-purchasing histories (i.e., pairwise constraints).

There is an emerging interest in semi-supervised clustering algorithms in the machine learning and data mining communities. In addition to the data values, we assume there are a number of instance-level constraints on cluster assignment. More specially, we consider the following two types of pairwise relations:
• Must-link constraints specify that two samples should be assigned into one cluster. • Cannot-link constraints specify that two samples should be assigned into different clusters.

Pairwise relations naturally occur in various domains and applications. In gene classification, our knowledge that two proteins co-occurring in processes can be viewed as a must link [1]. In information retrieval, the expert critique is often in the form "these two documents shouldn't be in the same cluster", which can be viewed as a cannot-link [2]. Pairwise relations may arise from knowledge of domain experts [3], perceived similarity (or dissimilarity) [4], or even common sense [5]. Unfortunately, those pairwise relations are often determined in a subjective way [2] or with significant uncertainty [4].

Manuscript received February 5, 2025
Manuscript revised February 20, 2025
https://doi.org/10.22937/IJCSNS.2025.25.2.22

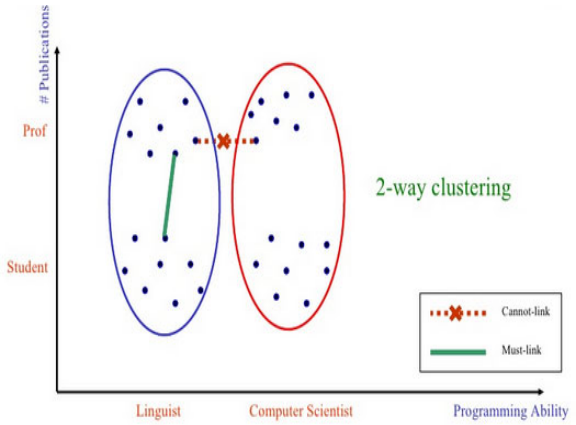
2. RELATED WORK

In this section, we divide the related work into three categories, namely semi-supervised clustering, clustering according to user's feedback, and dynamic network clustering. Most semi-supervised clustering methods can be categorized into two main groups [6]: constrained clustering methods and distance metric learning based methods. The constrained clustering methods employ side information to confine the solution space, and only seek feasible data partitions consistent with given constraints. Among them, hard constraints based methods only consider the cluster assignments such that all the constraints are strictly satisfied. For instance, Wagstaff et al. [3] modified the K-means clustering and self-organizing map algorithms to adjust the cluster memberships towards the given pairwise constraints. In, a generalized Expectation Maximization (EM) algorithm was applied to ensure that only the mixture models matching all the constraints are considered. The hard constraints based methods tend to be more sensitive to noise since some constraints may make the corresponding clustering problems infeasible [4].

To overcome this issue, a lot of studies have treated side information as soft constraints [5]. Instead of satisfying all the constraints, the soft constraints based methods aim to preserve those constraints as many as possible, while penalizing the number of violated constraints. In [7], probabilistic models were proposed to deal with semi-supervised clustering tasks, in which pairwise constraints were treated as Bayesian priors. In [7], pairwise constraints were formed as an additional penalty term in the objective of spectral learning. In [5], a parameter-free algorithm called Graph Scope was proposed to mine time-evolving graphs obeying the principle of Minimum Description Length (MDL). Facet Net [2] employed probabilistic community membership models to identify dynamic communities within a time-evolving graph. Kim and Han [8] further allowed a varying number of communities and presented a particle-and-density based algorithm to discover new communities or dissolve existing communities.

Albeit looking similar, dynamic network clustering is different from the focus of this paper due to the following reasons: (i) dynamic network clustering algorithms only use links to guide clustering but ignore the important feature information; (ii) they rely on a large amount of link information to conduct clustering, while our studied dynamic clustering only requires a small number of pairwise constraints. Due to the flexibility of handling both data features and dynamic relationships, our proposed semi-supervised clustering approach better fits conventional clustering applications.

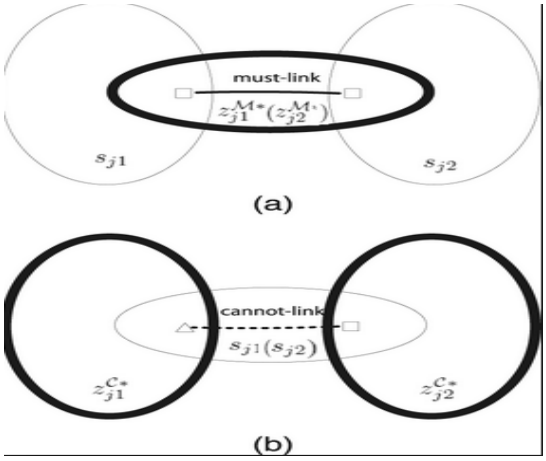
To apply the pairwise constraints to graph partition, existing methods either modify the affinity matrix directly [2], or constrain the underlying Eigen space [3]. In this work, we focus on the quadratic formulation for constrained spectral clustering proposed in [8] for two reasons: 1) the quadratic formulation matches nicely with the regularization framework for label propagation (referred to as the Generalized Label Propagation framework in [1]); 2) unlike other algorithms, the CSC algorithm in [8] can handle large amount of soft constraints, which is convenient for constraints generated from propagated labels. That being said, the equivalence we are to establish in Section IV is not limited to the formulation in [8], but also valid for other constrained spectral clustering formulation with a regularization framework



3. PAIRWISE CONSTRAINTS

We approach Problem 1 by considering pairs of correspondences. For each problem we will show how to detect geometric inconsistency of such a pair. Two geometrically inconsistent correspondences cannot both belong to the optimal solution and hence we look for large sets of pairwise consistent correspondences. In general pairwise consistency is not sufficient, but we will show how to get around this problem [7].

Assume we have established pairwise consistencies. We then build a graph with all hypothetical correspondences as vertices and edges connecting inconsistent ones [3]. Clearly a consistent subset according to Definition 1 cannot include any edges. Thus the maximal subset of pairwise consistent correspondences should be a good candidate for the optimal solution. Finding this set is equivalent to removing as few vertices as possible while covering all edges [6].



4. PROPOSED WORK

4.1 PAIRWISE CONSTRAINED COMPETITIVE AGGLOMERATION (PCCA)

The objective function to be minimized should combine the feature-based similarity between data points and the pair-wise constraints available. Let M be the set of must-link pairs such that (xi; xj) ϵ M implies xi and xj should be assigned to the same cluster, and C be the set of cannot-link pairs such that (xi; xj) ϵ C implies xi and xj should be assigned to different clusters [5]. Using the same notations as for CA, the objective function PCCA must minimize is:

$$\mathcal{J}(V,U) = \sum_{k=1}^{C}\sum_{i=1}^{N}(u_{ik})^2 d^2(\mathbf{x}_i,\mu_k)$$

The second term is composed of the cost of not respecting the pairwise must-link constraints and the cost of not respecting the pairwise cannot-link constraints. The penalty corresponding to the presence of two such points in different clusters (for must-link constraints) or in the same cluster (for cannot-link constraints) is weighted by their membership values [7]. This second term is weighted by α, which is a way to specify the relative importance of the supervision [11].

4.2 SEMI-SUPERVISED CLUSTERING WITH PAIRWISE CONSTRAINTS (SSPC)

In this section, we first present a general framework for semi-supervised clustering, followed by the proposed efficient algorithm for dynamic semi-supervised clustering [8]. Semi-Supervised Clustering Let X = (x1... xn) be a set of n data points to be clustered, where each data point xi ∈Rd, i ∈ [n] is a vector of d dimensions [3]. Let Mt be the set of must-link constraints generated until time t, where each must-link pair (xi,xj) ∈Mt implies that xi and xj should be in the same cluster. Similarly, let Ct be the set of cannot-link constraints generated until time t, where each cannot-link pair (xi,xj) ∈ Ct implies that xi and xj should belong to different clusters [5]. For ease of presentation, we also define Ωt = Mt∪Ct to include all pairwise constraints generated until time t. Similar to most studies on data clustering, we assume that the number of clusters r is given a priori[8]. Throughout this paper, we use a binary matrix F ∈ {0,1}n×r to represent the result of partitioning n data points into r clusters, where Fij = 1 indicates that xi is associated with the j-th cluster. We further denote F as the set of all possible clustering results.

$$\min_{F\in\mathcal{F}} \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{M}_t} \ell_+(F_{i,*},F_{j,*}) + \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{C}_t} \ell_-(F_{i,*},F_{j,*})$$

$$\text{s.t.} \quad d(F,K)\le\varepsilon, \qquad (4$$

$$\mathcal{F}=\{F\in\{0,1\}^{n\times r}: F_{*,i}^{\top}F_{*,j}=0 \ \ \forall i\neq j,\ \sum F_{k,*}=1\ \forall k\},$$

4.3 DYNAMIC SEMI-SUPERVISED CLUSTERING (DSSC)

The proposed algorithm is based on a key observation that the number of different clustering results F in the set Δ = {F ∈F: d (K, F) ≤ ε} is not very large when ε is relatively small and the eigenvalues of K follow a skewed distribution [1]. To see this, we denote by λ1... λn the eigenvalues of K ranked in descending orders, and v1... vn the corresponding eigenvectors. {λk} follows a q-power law if there exists a constant c such that λk ≤ ck−q, where q > 2. The following lemma summarizes an important property of K when its eigenvalues follow a q-power law [9]. Specifically, the proposed clustering process is composed of two steps: an offline and an online step [7]. In the offline step, we generate multiple partitions of the same dataset X and use such partitions to construct a convex hull Δ. In the online step, an efficient learning algorithm is developed to update the combination weights based on the newly received pairwise constraints.

$$\theta_n(\rho,r)\le\left[\frac{2n}{(r-1)s}\right]^{Cs(r-1)/(2\rho)}$$

$$s=\sqrt{\varepsilon n}\left(1+\frac{2}{q-2}\right)$$

IV d.

4.4 SEMI-SUPERVISED PAIRWISE GAUSSIAN PROCESS CLASSIFIER (SPGP)

We can now combine the likelihood (and its approximation) formulated in Equation (9) and (11), and a Gaussian prior based on the semi-supervised kernel. As mentioned in Section, the classification is given by the MAP solution off [10]. According to Proposition 2, the

optimization in equation (12) can be divided into the following two steps

$$\text{step 1: } \hat{\mathbf{f}}_c = \arg\min_{\mathbf{f}_c}\{\frac{1}{2}\mathbf{f}_c^T\mathbf{K}_c^{-1}\mathbf{f}_c - \sum_{w_{ij}\neq 0}\log\frac{e^{w_{ij}\{e^{f(x_i)+f(x_j)}+1\}+e^{f(x_i)}+e^{f(x_j)}}}{(e^{f(x_i)}+1)(e^{f(x_j)}+1)}\}$$

$$\text{step 2: } \hat{\mathbf{f}}_u = \mathbf{K}_{uc}\mathbf{K}_c^{-1}\hat{\mathbf{f}}_c.$$

Here K is one of the graph kernels, and both K and f are decomposed as in Section. The decomposition (step 1-step 2) effectively reduces the optimization over f to a subset fc, which is substantially cheaper when only a small portion of samples are constrained [10].

The objective function in step 1 consists of two terms: the empirical error

$$-\sum_{w_{ij}\neq 0}\log\frac{e^{w_{ij}\{e^{f(x_i)+f(x_j)}+1\}+e^{f(x_i)}+e^{f(x_j)}}}{(e^{f(x_i)}+1)(e^{f(x_j)}+1)},$$

V.

5. EXPERIMENTS

In this section, we empirically demonstrate that our proposed semi-supervised clustering algorithm is both efficient and effective.

5.1 DATASETS

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

• COIL20, a dataset containing 20 objects with 1,440 images in total. Each image is represented by a 1024-dimensional vector.

• USPS, a widely used handwritten digits dataset including 9,298 handwritten images. Each image is represented by a 256-dimensional vector that belongs to one of 10 classes.

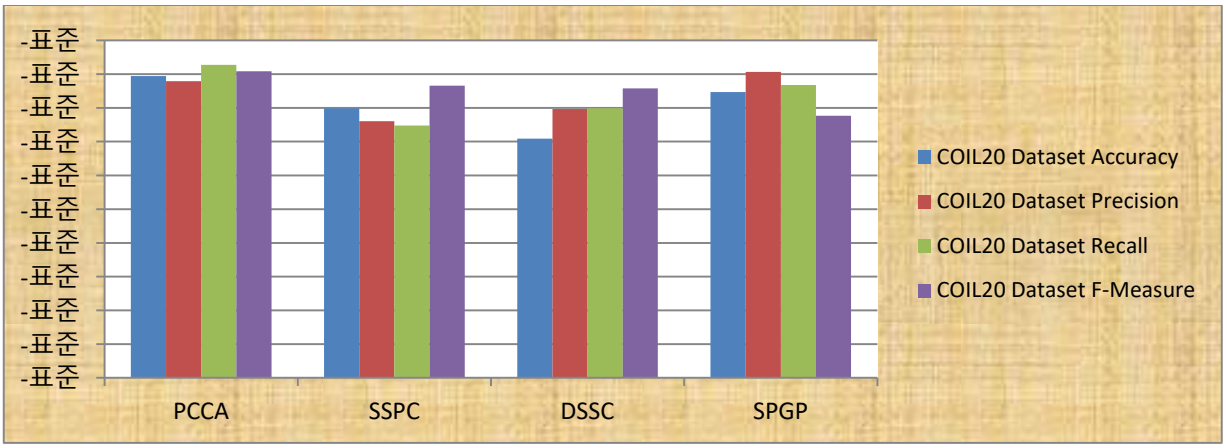
• Covtype5, a dataset used to predict forest covers types using cartographic variables. This dataset consists of 581,012 records belonging to seven cover type classes, i.e., spruce/fir, lodge pole pine, ponderosa pine, cottonwood/willow, aspen, Douglas-fir, and krummholz.

• MNIST8m, a dataset artificially enlarged from the MNIST handwritten digits dataset6. It contains a total of 8,100,000 samples that belong to 10 classes.

6. EXPERIMENTAL RESULTS

6.1 COIL20 DATASET RESULTS

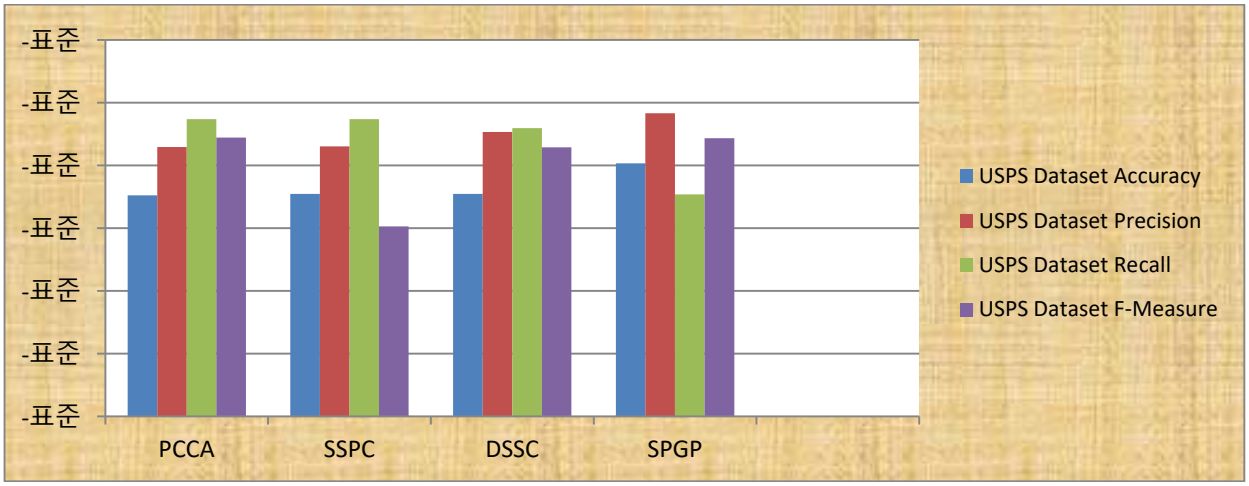
	COIL20 Dataset			
Algorithm	Accuracy	Precision	Recall	F-Measure
PCCA	89.45	87.91	92.77	90.89
SSPC	79.91	76.08	74.78	86.56
DSSC	70.92	79.67	79.89	85.78
SPGP	84.67	90.67	86.78	77.67



The above graph shows that performance of Coil20 dataset. The Accuracy of PPCA algorithm is 89.45 which is higher when compare to other three (SSPC, DSSC, SPGP) algorithms. The Precision of SPGP algorithm is 90.67 which is higher when compare to other three (SSPC, DSSC, PPCA) algorithms. The Recall of PPCA algorithm is 92.77 which is higher when compare to other three (SSPC, DSSC, SPGP) algorithms. The F-Measure of PPCA algorithm is 90.89 which is higher when compare to other three (SSPC, DSSC, SPGP) algorithms.

6.2 USPS DATASET RESULTS

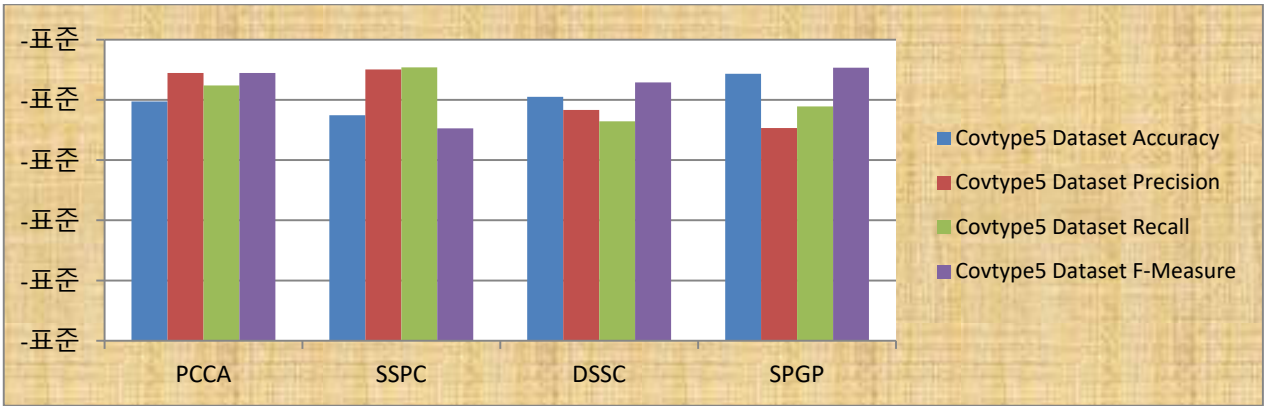
	USPS Dataset			
Algorithm	Accuracy	Precision	Recall	F-Measure
PCCA	70.45	85.91	94.77	88.89
SSPC	70.91	86.08	94.78	60.56
DSSC	70.92	90.67	91.89	85.78
SPGP	80.67	96.67	70.78	88.67



The above graph shows that performance of USPS dataset. The Accuracy of SPGP algorithm is 80.67 which is higher when compare to other three (SSPC, DSSC, PCCA) algorithms. The Precision of SPGP algorithm is 96.67 which is higher when compare to other three (SSPC, DSSC, PPCA) algorithms. The Recall of SSPC algorithm is 94.78 which is higher when compare to other three (PCCA, DSSC, SPGP) algorithms. The F-Measure of PPCA algorithm is 88.89 which is higher when compare to other three (SSPC, DSSC, SPGP) algorithms.

6.3 COVTYPE5 DATASET RESULTS

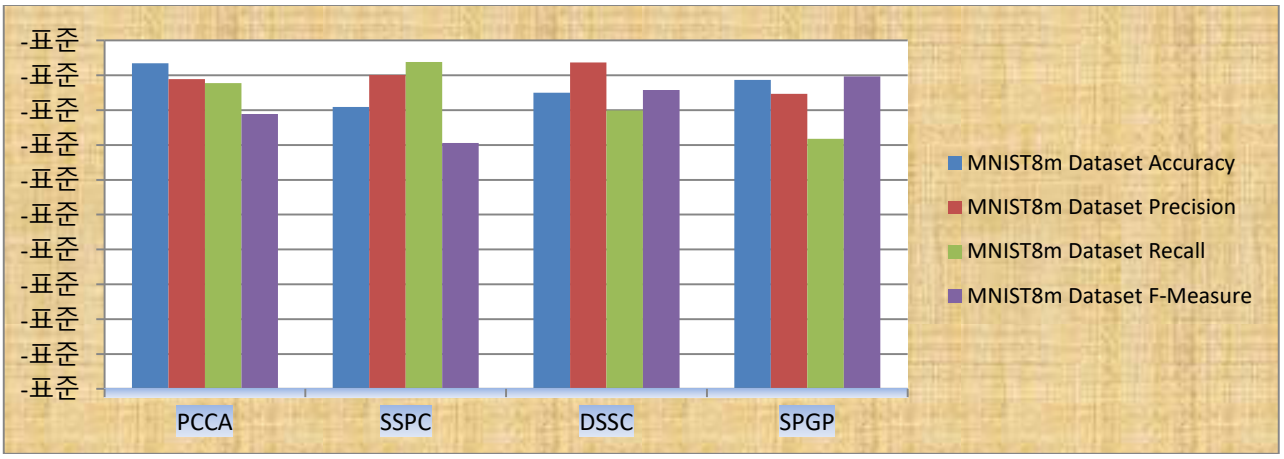
	Covtype5 Dataset			
Algorithm	Accuracy	Precision	Recall	F-Measure
PCCA	79.45	88.91	84.77	88.89
SSPC	74.91	90.08	90.78	70.56
DSSC	80.98	76.67	72.89	85.78
SPGP	88.67	70.67	77.78	90.67



The above graph shows that performance of Covtype5 dataset. The Accuracy of SPGP algorithm is 88.67 which is higher when compare to other three (SSPC, DSSC, PCCA) algorithms. The Precision of SSPC algorithm is 90.08 which is higher when compare to other three (SPGP, DSSC, PPCA) algorithms. The Recall of SSPC algorithm is 90.78 which is higher when compare to other three (PCCA, DSSC, SPGP) algorithms. The F-Measure of SPGP algorithm is 90.67 which is higher when compare to other three (SSPC, DSSC, PCCA) algorithms.

6.4 MNIST8M DATASET RESULTS

	MNIST8m Dataset			
Algorithm	Accuracy	Precision	Recall	F-Measure
PCCA	93.45	88.91	87.77	78.89
SSPC	80.91	90.08	93.78	70.56
DSSC	84.98	93.67	79.89	85.78
SPGP	88.67	84.67	71.78	89.67



The above graph shows that performance of MNIST8m dataset. The Accuracy of PCCA algorithm is 93.45 which is higher when compare to other three (SSPC, DSSC, SPGP) algorithms. The Precision of DSSC algorithm is 93.67 which is higher when compare to other three (SSPC, SPGP, PPCA) algorithms. The Recall of SSPC algorithm is 93.78 which is higher when compare to other three (PCCA, DSSC, SPGP) algorithms. The F-Measure of SPGP algorithm is 89.67 which is higher when compare to other three (SSPC, DSSC, PCCA) algorithms.

7. CONCLUSION

In this paper, we proposed a dynamic semi-supervised clustering algorithm which can efficiently update clustering results given newly received pairwise constraints. The key idea is to cast the dynamic clustering process into a search problem over a feasible clustering space that is defined as a convex hull generated by multiple ensemble partitions. Since any inner point of the convex hull can be uniquely represented by a probability simplex vector, the dynamic semi-supervised clustering problem can be reduced to the problem of learning a low-dimensional vector. Given a set of sequentially received pairwise constraints, we devised an updating scheme to update the data partition in an extremely efficient manner. Our empirical studies conducted on several real-world datasets confirmed both the effectiveness and efficiency of the proposed algorithm.

In recent work on semi-supervised clustering with pairwise constraints, [8] used gradient descent Pairwise Constrained Competitive Agglomeration (PCCA) combined with a in the context of SSPC clustering. [2] Proposed a Dynamic Semi-Supervised Clustering (DSSC) algorithm that uses must-link constraints to learn a Pairwise distance. [24] Utilized both must link and cannot link constraints to formulate a convex optimization problem which is local-minima-free. [5, 6] proposed a method based on Semi-supervised Pairwise Gaussian Process Classifier (SPGP) which learns a metric during clustering to minimize an objective function which incorporates the constraints. This is equivalent to the minimization of the posterior energy of the DSCC.

REFERENCES

[1] S. Basu, M. Bilenko, and R. J. Mooney, "A probabilistic framework for semi-supervised clustering", In KDD, pages 59–68, 2011.

[2] S Basu, A Banerjee, and R J. Mooney, "Semi-supervised clustering by seeding", In ICML, pages 27–34, 2010.

[3] H. Zeng and Y. Cheung, "Semi-supervised maximum margin clustering with pairwise constraints", IEEE Trans. Knowl. Data Eng., 24(5):926–939, 2012.

[4] C. Blake and C. Merz, "Uci repository of machine learning databases," 2011.

[5] Basu, A. Banerjee, and R. Mooney," Active semi-supervision for pairwise constrained clustering", In Proceedings of SIAM SDM, 2012.

[6] Dhillon, I. S., Fan, J., & Guan, Y," Efficient clustering of very large document collections", In Data Mining for Scientific and Engineering Applications, Kluwer Academic Publishers, 2011.

[7] Hinneburg, A., & Keim, D. A, "An efficient approach to clustering in large multimedia databases with noise", In Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining (KDD-98), pp. 58–65, 2013.

[8] Jain, A. K., & Dubes, R. C," Algorithms for Clustering Data", Prentice Hall, New Jersey, 2009.

[9] Jain, A. K., Myrthy, M. N., & Flynn, P. J, "Data clustering: A survey", ACM Computing Survey, 31(3), 264–323, 2009.

[10] Kaufman, L., & Rousseau, P, "Finding Groups in Data: An Introduction to Cluster Analysis", John Wiley and Sons, New York, 2010.

[11] Kleinberg, J., & Tardos, E, "Approximation algorithms for classification problems with pairwise relationships: Metric labeling and Markov random fields", In IEEE Symp. On Foundations of Comp. Sci, 2009.

[12] Zhang, T., Ramakrishna, R., & Livny, M, "Birch: An efficient data clustering method for very large databases", In Proceedings of the ACM SIGMOD International Conference on Management of Data, pp. 103–114, 2011.