Unsupervised Clustering Using Hyperclique Pattern Constraints

Conference Paper · January 2009

DOI: 10.1109/ICPR.2008.4761252 · Source: IEEE Xplore

4 authors, including:

Yuchou Chang
Barrow Neurological Institute
52 PUBLICATIONS **437** CITATIONS
SEE PROFILE

Lee Dah-Jye
Brigham Young University - Provo Main Campus
148 PUBLICATIONS **1,402** CITATIONS
SEE PROFILE

James K. Archibald
Brigham Young University - Provo Main Campus
91 PUBLICATIONS **1,786** CITATIONS
SEE PROFILE

Unsupervised Clustering Using Hyperclique Pattern Constraints

Yuchou Chang[a], Dah-Jye Lee[a], James Archibald[a], and Yi Hong[b]
[a]*Dept. of Electrical and Computer Engineering, Brigham Young University*
[b]*Dept. of Computer Science, City University of Hong Kong, Kowloon, Hong Kong*
ycchang@et.byu.edu djlee@ee.byu.edu jka@ee.byu.edu yihong@cityu.edu.hk

Abstract

A novel unsupervised clustering algorithm called Hyperclique Pattern-KMEANS (HP-KMEANS) is presented. Considering recent success in semi-supervised clustering using pair-wise constraints, an unsupervised clustering method that selects constraints automatically based on Hyperclique patterns is proposed. The COP-KMEANS framework is then adopted to cluster instances of data sets into corresponding groups. Experiments demonstrate promising results compared to classical unsupervised k-means clustering.

1. Introduction

Clustering is one of the most fundamental mental activities humans engage in. It is used to handle huge amounts of information received every day [1]. Although traditional clustering is normally used as an unsupervised method for data analysis, it has been modified by incorporating instance-level constraints to increase clustering accuracy in recent studies [2-4]. This constrained clustering can be considered to be semi-supervised clustering. The key idea behind it is that, for some domains, instances are constrained to reside together (or not to reside together) according to background knowledge. Because of its enhanced clustering quality, instance-level constraint-based semi-supervised clustering has been one of the most active research areas in pattern recognition, data mining, and machine learning [5, 6].

However, apart from unlabeled instances, semi-supervised clustering still needs supervision for some of the instances. Hyperclique patterns were first suggested as a means for identifying strong affinity patterns using an objective measure called *h*-confidence [7]. These transactions are from market information, which contains specific background knowledge [7]. An example application was its use as a data cleaning technique for noise removal [8]. We propose a solution based on k-nearest neighbor (KNN) to construct transactions of instances, so instance-level constraints can be automatically extracted for unsupervised clustering.

The rest of the paper is organized as follows. Section 2 briefly discusses relevant work and describes the proposed algorithm for solving the unsupervised clustering problem using Hyperclique pattern discovery. Section 3 gives experimental results and comparisons. Conclusions are drawn in Section 4.

2. Hyperclique pattern discovery for clustering with instance-level constraints

2.1. Relevant Work

Clustering has four major steps: data reduction, hypothesis generation, hypothesis testing, and prediction based on groups [1]. Based on unsupervised clustering algorithms, some semi-supervised clustering techniques have been proposed [2-4]. The main idea behind these semi-supervised clustering algorithms is to incorporate background knowledge in the form of instance-level constraints.

Two types of instance-level constraints—must-link and cannot-link [2]—were incorporated into the COBWEB [9] clustering algorithm to improve its performance. Wagstaff et al. [2] extended the k-means algorithm [10] with background knowledge resulting in an approach called COP-KMEANS [3]. COP-KMEANS was applied to the GPS lane-finding problem and achieved better experiment results than the unconstrained k-means algorithm. Furthermore, Basu et al. proposed a probabilistic model for semi-supervised clustering based on Hidden Markov Random Fields. Their analysis concluded that constrained clustering outperforms unsupervised clustering by partially labeling instance constraints before clustering procedures are executed.

978-1-4244-2175-6/08/$25.00 ©2008 IEEE

Constrained clustering requires human supervision to label instance-level constraints; it is therefore considered a semi-supervised method. The key problem of constrained clustering is how to choose instance-level constraints efficiently. In this paper, we propose a clustering algorithm called HP-KMEANS to produce instance-level constraints automatically to improve k-means clustering. It is an unsupervised clustering algorithm because it requires no human assistance to determine instance-level constraints. In the algorithm, Hyperclique patterns are generated by both support and h-confidence measures [7, 8]. H-confidence has three properties: anti-monotone property, cross-support property, and strong affinity property. As mentioned in Section 1, knowledge about transactions is usually not available before clustering. The modified Hyperclique pattern should be designed to improve unsupervised clustering.

2.2. Hyperclique Pattern K-MEANS Clustering Algorithm

The proposed HP-KMEANS algorithm can be divided into two parts. First, all potential instance-level constraints are automatically generated based on random sampled subsets of all instances and Hyperclique patterns are obtained using KNN. Then, essential constraints are extracted from potential constraints. COP-KMEANS is then applied to partition the complete data set using these essential constraints.

Random Sampling Consensus (RANSAC) [11] was firstly introduced for fitting a model to experimental data. Rather than using as much of the data as possible to obtain an initial solution and then attempting to eliminate invalid data points, RANSAC uses multiple data subsets to obtain a consensus set. Similarly, in this paper, subsets of all instances are randomly selected to extract potential constraints. In order to perform automatic generation of all potential instance-level constraints, an unsupervised KNN is used to automatically group the instances into different transactions, from which Hyperclique patterns are extracted as constraints for constrained k-means clustering. KNN is a pattern recognition method in which an object is classified by the majority votes of its neighbors [12]. It categorizes similar data into the same group to form a compact set. The underlying principle of Hyperclique patterns suggests that each item (instance) should belong to one or more transactions. For each current instance, KNN can sort all other instances based on similarity measures. It can be seen that KNN could be applied to automatically conglomerate similar instances into a compact set as one pattern.

The basic idea behind Hyperclique pattern is the h-confidence, which is an interestingness measure used for mining patterns containing certain items (instances) [7, 8]. These items are highly affiliated with each other, although some of them appear in the dataset less frequently (lower support measure). This observation was derived from analysis of retail data set. For example, one customer's purchase pattern may be more like <diaper, baby clothes> than <diaper, pesticide>. The pattern <diaper, baby clothes> may not appear as frequently as other patterns such as daily food combination (for example, <milk, bread>). Based one this observation, h-confidence was proposed for discovering hyperclique pattern with high interestingness measure (appear more frequently). Similar to retail transactions, we construct transactions by KNN to compute support measure and h-confidence for general data sets. Then, generated hyperclique patterns with positive h-confidence (appear more frequently) are chosen as potential constraints for later process. The algorithm for extracting potential constraints is summarized below.

1. Randomly choose the current subset M_c =*rand*(*1*) instances from the data set. *rand*(*1*)∈ [*1*, *M*] is a random integer number denoting the number of instances in current subset; *M* is the overall number of instances in the data set.
2. In the current subset, for each instance I_c, use KNN to find its *K* nearest neighbors (I_{c1}, I_{c2}, …, I_{ck}) from the current subset. I_c and I_{c1} (I_c's closest neighbor) are chosen to construct one pattern and I_c, I_{c1}, I_{c2}, …, I_{ck} to construct one transaction. (*K*=5 was used in this paper.)
3. Repeat step (2) until all M_c patterns and M_c transactions of M_c instances in the current subset are extracted.
4. For each instance's pattern, calculate its support measure according to the generated M_c transactions [7].
5. For each of the M_c unique patterns, calculate its h-confidence measure in M_c transactions [7].
6. For each pattern I_c and I_{c1} whose h-confidence measure value is greater than zero, construct potential constraint "must-link" (I_c, I_{c1}) [3], and store them as candidates (hyperclique pattern) for extracting essential constraints.
7. Repeat step (1) *IterNum* times. (*IterNum=100* was used in this study).

Assume that one instance is labeled as belonging to one category, and the other (M_c-1) instances are labeled as being in another category for one current subset. Unlabeled instances within different categories may be grouped into one pattern by KNN. Thus, after all potential constraints are obtained, they are refined to determine the essential constraints. In turn, these are

used to partition the data set using COP-KMEANS clustering. Suppose that there are N potential constraints generated by the above algorithm, each of which has one pair of instances with some particular distance between them. These N potential constraints are sorted according to the distances between their instances. If the distance between instances of a potential constraint is small, then these two instances are more likely to belong to the same clustering. On the other hand, if the distance is large, this constraint may be an invalid "must-link" constraint that is generated by KNN incorrectly. The essential constraints can be extracted automatically if a threshold can be found to split N potential constraints into two classes: essential constraints and invalid constraints.

For potential must-link constraints ((I_c_1, I_{c1}_1), (I_c_2, I_{c1}_2), …, (I_c_N, I_{c1}_N)), after sorting based on distance between the two instances, a sorted list of potential constraints (($I_c'\_1$, $I_{c1}'\_1$), ($I_c'\_2$, $I_{c1}'\_2$), …, ($I_c'\_N$, $I_{c1}'\_N$)) from small to large distances is obtained. Suppose *MinD* and *MaxD* are minimum and maximum distances of all pairs of instances in these N potential constraints. This distance range (between *MinD* and *MaxD*) is partitioned into L levels, and the distance of each pair of instances belongs to exactly one of these L levels. The goal of splitting the sorted potential constraints into essential and invalid constraints is transformed into finding one optimal threshold on L levels to extract instance pairs that have a distance smaller than the threshold of the essential constraints. Similar to Otsu's method [13], the optimal threshold is determined by the corresponding between-class variance over the total variance. Suppose that all potential constraints can be divided into two classes by a threshold k, $1 \le k \le L$. The number of constraints at level i is denoted as n_i, and N is the total number of constraints. Thus, the class means are given by the following

$$\mu_1(k) = \frac{\sum_{i=1}^{k} i n_i}{\omega_1} \quad , \quad \mu_2(k) = \frac{\sum_{i=k+1}^{L} i n_i}{\omega_2} \tag{1}$$

where

$$\omega_1(k) = \sum_{i=1}^{k} n_i \quad , \quad \omega_2(k) = \sum_{i=k+1}^{L} n_i \, . \tag{2}$$

The total mean and variance can be expressed as

$$\mu_T = \omega(L) = \frac{\sum_{i=1}^{L} i n_i}{N} = \frac{\omega_1(k)\mu_1 + \omega_2(k)\mu_2}{\omega_1(k) + \omega_2(k)} \tag{3}$$

and

$$\sigma_T^2 = \sum_{i=1}^{L} (i - \mu_T)^2 n_i \, . \tag{4}$$

The variance between two classes is calculated as

$$\begin{aligned} \sigma_B^2(k) &= \omega_1(k)(\mu_1(k) - \mu_T)^2 \\ &\quad + \omega_2(k)(\mu_2(k) - \mu_T)^2 \end{aligned} \tag{5}$$

The optimal threshold is obtained by maximizing η, given by

$$\eta = \sigma_B^2 / \sigma_T^2 \, . \tag{6}$$

The essential constraints are those instance pairs that have a distance smaller than the optimal threshold L_o. These automatically generated constraints are then input to the COP-KMEANS algorithm to obtain the final partition groups of the data set.

3. Experimental results

Six data sets from the **UCI** machine learning repository [14] are used to evaluate the performance of the proposed HP-KMEANS algorithm. Detailed data set information is shown in Table 1. In our experiments, we used Rand Index to measure the accuracy of the clustering solution [15].

Table 1. Six data sets for experiments

	Data Set	Instances	Classes
1	Soybean	47	4
2	Zoo	101	7
3	Wine	178	3
4	Statlog	94	4
5	Iris	150	3
6	Glass	214	6

Because the proposed algorithm is an unsupervised clustering method with automatically selected constraints, we compared it against the classical unsupervised k-means algorithm [10] with randomly chosen initial centers. For each data set in Table 1, we iteratively executed HP-KMEANS and classical k-means 10 times to test their average performance. It can be seen that HP-KMEANS has better performance than classical k-means as shown in Table 2. Results of these 10 executions on each data set are shown in Figure 1.

Table 2. Clustering Accuracy Comparison

	Data Set	HP-KMEANS	k-means
1	Soybean	83.32%	82.77%
2	Zoo	87.62%	87.65%
3	Wine	71.87%	70.21%
4	Statlog	62.04%	61.81%
5	Iris	85.83%	83.13%
6	Glass	68.87%	67.82%

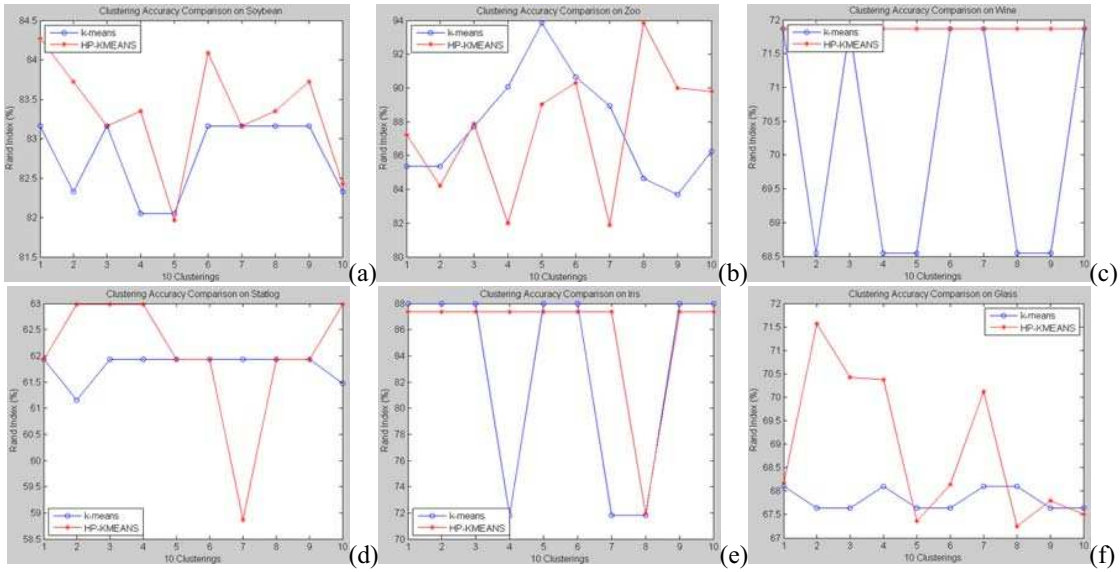

Figure 2. 10 Times Clustering Comparisons between HP-KMEANS and k-means (a) Soybean, (b) Zoo, (c) Wine, (d) Statlog, (e) Iris, and (e) Glass.

4. Conclusion

We have proposed a new unsupervised clustering algorithm based on Hyperclique pattern analysis and COP-KMEANS [3]. We use KNN to automatically discover potential constraints. Essential constraints are extracted for unsupervised clustering. The experimental results show that the proposed method outperforms the classical unsupervised k-means algorithm. Parameters such as K and IterNum are manually set in our research. Optimal parameter setting strategy such as clustering ensemble may be used to increase accuracy.

5. References

[1] S. Theodoridis, and K. Koutroumbas, Pattern Recognition, Second Edition, Academic Press, ISBN: 0126858756, February, 2003.

[2] K. Wagstaff, and C. Cardie, Clustering with Instance-level Constraints, Proceedings of the International Conference on Machine Learning, pp.1103-1110, 2000.

[3] K. Wagstaff, C. Cardie, S. Rogers, and S. Schroedl, Constrained K-means Clustering with Background Knowledge, Proceedings of International Conference on Machine Learning, pp. 577-584, 2001.

[4] S. Basu, M. Bilenko, R.J. Mooney, A Probabilistic Framework for Semi-Supervised Clustering, Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp.59-68, 2004.

[5] O. Chapelle, B. Scholkopf, and A. Zien, Semi-Supervised Learning, MIT Press, Cambridge, MA, London, England, ISBN: 978-0-262-03358-9, 2006.

[6] T. Hertz, N. Shental, A. bar-Hillel, and D. Weinshall, Enhancing Image and Video Retrieval: Learning via Equivalence Constraints, IEEE International Conference on Computer Vision and Pattern Recognition, vol.2, pp.668-674, 2003.

[7] H. Xiong, P.N. Tan, and V. Kumar, Mining Strong Affinity Association Patterns in Data Sets with Skewed Support Distribution, IEEE International Conference on Data Mining, pp.387-394, 2003.

[8] H. Xiong, G. Pandey, M. Steinbach, and V. Kumar, Enhancing Data Analysis with Noise Removal, IEEE Transactions on Knowledge and Data Engineering, vol.18, no.3, pp.304-319, 2006.

[9] D.H. Fisher, Knowledge Acquisition Via Incremental Conceptual Clustering, Machine Learning, vol.2, no.2, pp.139-172, 1987.

[10] J.B. MacQueen, Some Methods for Classification and Analysis of Multivariate Observations, Proceedings of the Fifth Symposium on Math, Statistics, and Probability, pp.281-297, University of California Press, Berkeley, CA, 1967.

[11] M.A. Fischler, and R.C. Bolles, Random Sample Consensus: A Paradigm for Model Fitting with Application to Image Analysis and Automated Cartography, Communications of ACM, vol.24, no.6, pp.381-395, 1981.

[12] B.V. Dasarathy, Nearest Neighbor (NN) Norms: NN Pattern Classification Techniques, IEEE Computer Society Press, ISBN: 0818689307, 1990.

[13] N. Otsu, A Threshold Selection Method From Grey-Level Histogram, IEEE Transactions on Systems, Man, and Cybernetics, vol.9, no.1, pp.62-66, 1979.

[14] http://archive.ics.uci.edu/ml/index.html

[15] W.M. Rand, Objective Criteria for the Evaluation of Clustering Methods, Journal of American Statistical Association, vol.66, no.336, pp.846-850, 1971.