Combining Multi-layer Perceptron and K-Means for Data Clustering with Background Knowledge

Donghai Guan, Weiwei Yuan, Young-Koo Lee*, Andrey Gavrilov,
and Sungyoung Lee

Department of Computer Engineering
Kyung Hee University, Korea
{donghai,weiwei,avg,sylee}@oslab.khu.ac.kr, yklee@khu.ac.kr

Abstract. Clustering is traditionally viewed as an unsupervised method for data analysis. However, in some cases information about the problem domain is available in addition to the data instances themselves. To make use of this information, in this paper, we develop a new clustering method "MLP-KMEANS" by combining Multi-Layer Perceptron and K-means. We test our method on several data sets with partial constrains available. Experimental results show that our method can effectively improve clustering accuracy by utilizing available information.

1 Introduction

Clustering plays an indispensable role in data analysis. Traditionally it is treated as part of unsupervised learning [1][2]. Usually in clustering, there is no available information concerning the membership of data items to predefined classes. Recently, a kind of new data analysis methods is proposed, called semi-supervised clustering. It is different with traditional clustering by utilizing small amount of available knowledge concerning either pair-wise (must-link or cannot-link) constrains between data items or class labels for some items [3][4][5].

In practical applications, semi-supervised clustering is urgently needed because in many cases user processes some background knowledge about the data set that could be useful in clustering. Traditional clustering algorithms are devised only for unsupervised learning and they have no way to take advantage of this information even when it does exist.

We are interested in developing semi-supervised clustering algorithms which can utilize background information. K-means is a popular clustering algorithm that has been used in a variety of application domains, such as image segmentation [6], and information retrieval [7]. Considering its widespread use, we develop a new clustering approach based on it. Before us, researchers have devised some k-means variants to make use of background information [8][9][10]. Compared with those algorithms, our algorithm does not adapt original k-means algorithm. Strictly speaking, our approach

[*] Corresponding author.

D.-S. Huang, L. Heutte, and M. Loog (Eds.): ICIC 2007, CCIS 2, pp. 1220–1226, 2007.

is not a k-means variant. It is a model which combines Multi-Layer Perceptron and k-means for data clustering.

In the next section, we will provide the form of background knowledge used in our method. In Section 3, we will present in detail about our clustering method. Then we describe our evaluation method in Section 4. Experimental results are shown in Section 5. Finally, Section 6 consists of conclusions and future works.

2 Background Knowledge for Clustering

In semi-unsupervised clustering, background knowledge refers to the available knowledge concerning either pair-wise (must-link or cannot-link) constrains between data items or class labels for some items. In current work, we will focus on using constrains between data items. Two types of pairwise constrains will be considered:

- *Must-link constrains* specify that two instances have to be in the same cluster.
- *Cannot-link constrains* specify that two instances must not be placed in the same cluster.

Must-link and Cannot-link are Boolean function. Assuming S is the given data set and P , Q are data instances, $P, Q \in S$. If P and Q belong to same class, $Must-link(P,Q)=True$. Otherwise, $Cannot-link(P,Q)=True$. Table 1 shows that pairwise constraints have two properties: symmetric and transitive.

Table 1. Properties of pairwise constraints

Symmetric: if $P, Q \in S$,
$Must-link(P,Q) \Leftrightarrow Must-link(Q,P)$
$Cannot-link(P,Q) \Leftrightarrow Cannot-link(Q,P)$
Transitive: if $P, Q, R \in S$,
$Must-link(P,Q) \,\&\,\& Must-link(Q,R) \Rightarrow Must-link(P,R)$
$Must-link(P,Q) \,\&\,\& Cannot-link(Q,R) \Rightarrow Cannot-link(P,R)$

3 MLP-KMEANS

3.1 K-Means Clustering

K-means clustering [11] is a method commonly used to automatically partition a data set into k groups. It proceeds by selecting k initial cluster centers and then iteratively refining them as follows:

1) Each instance d_i is assigned to its closest cluster center.

2) Each cluster center C_j is updated to be the mean of its constituent instances.

The algorithm converges when there is no further change in assignment of instances to clusters. In this work, we initialize the clusters using instances chosen at random from the data set. The data sets we used are composed solely of numeric features. Euclidean distance is used as measure of similarity between two data instances.

Table 2. MLP-KMEANS

```
Algorithm: MLP-KMEANS
  Input: data set $D$ must-link constrains $C_{m-link} \subseteq D \times D$
                      cannot-link constrains $C_{no-link} \subseteq D \times D$
                      Output: Partitions of instances in $D$
Stage 1: K-means clustering
    1. Let $C_1 ... C_k$ be the initial cluster centers.
    2. For each point $d_i$ in $D$, assign it to the closet cluster $C_j$.
    3. For each cluster $C_j$, update its center by averaging all of the
       points $d_j$ that have been assigned to it.
    4. Iterate between (2) and (3) until convergence.
    5. Return $\{C_1 ... C_k\}$.
Stage 2: Violate-Constraints Test
    6. $\{C_1 ... C_k\}$ makes new constrains $C_{k-m-link}$ and $C_{k-no-link}$
    7. For instances $d_i$ and $d_j$, if they have consistent constrains in
       original and new constraints, their labels generated by K-means
       are thought reliable. $D_r$ includes all the instances with reliable
       labels.
Stage 3: MLP Training
    8. MLP is trained by error back propagation (EBP) algorithm.
       Only $D_r$ and corresponding labels are used for training.
Stage 4: Clustering using MLP
    9. $D$ is inputted into MLP to cluster.
```

3.2 Combining MLP and K-Means for Clustering

Table 1 contains the algorithm MLP-KMEANS. The algorithm takes in a data set (D), a set of must-link constraints (C_{m-link}), and a set of cannot-link constrains $(C_{no-link})$. It returns a partition of the instances in D that satisfied all specified constrains.

In MLP-KMEANS, clustering consists of four stages. In the first stage, D is partitioned by K-means. K clusters $C_1...C_k$ are generated. The second step is Violate-Constraints test. The key idea of clustering in MLP-KMEANS is that MLP is trained using the output of K-means algorithm. So if the output of K-means clustering is not correct, MLP cannot be trained well. In turn, MLP cannot achieve high clustering accuracy. This step is used to filter out those samples whose labels generated by K-means might not be correct by violate-constraints test. Violate-constraints is Boolean function. For any two data instances P, Q, if $VC(P,Q) = True$, then P, Q are though mis-clustered by K-means. In detail, new constraints are generated based on the output of K-means. We call them k-must-link constraints $(C_{k-m-link})$ and k-cannot-link constraints $(C_{k-no-link})$. For P, Q, $VC(P,Q) = True$ in the following situations:

1) $Must - link(P,Q) \& \& K - Cannot - link(P,Q) = True$
2) $Cannot - link(P,Q) \& \& K - Must - link(P,Q) = True$

After Violate-Constrains test, the instances with $VC(P,Q) = False$ are gathered into D_r .Stage 3 is MLP training using D_r and corresponding labels. After training, in stage 4, MLP can be used for clustering instead of K-means.

4 Evaluation Method

The data sets used for the evaluation include a "correct answer" or label for each data instance. We use the labels in a post-processing step for evaluating performance.

To calculate agreement between our results and the correct labels, we make use of the Rand index [12]. This allows for a measure of agreement between two partitions. P_1 and P_2, of the same data set D. Each partition is viewed as a collection of $n*(n-1)/2$ pairwise decisions, where n is the size of D. For each pair of points d_i and d_j in D, P_i either assigns them to the same cluster or to different clusters. Let a be the number of decisions where d_i is in the same cluster as d_j in P_1 and in P_2. Let b be the number of decisions where the two instances are placed in different clusters in both partitions. Total agreement can then be calculated using the following equation.

$$Rand(P_1, P_2) = \frac{a+b}{n*(n-1)/2}. \tag{1}$$

We used this measure to calculate accuracy for all of our experiments.

5 Experimental Results Using Artificial Constrains

In this section, we report on experiments using three well-known data sets in conjunction with artificial-generated constrains. Each graph demonstrates the change in accuracy as more constrains are made available to the algorithm. The true value of k is known for these data sets, and we provide it as input to our algorithm.

The constraints were generated as follows: for each constraint, we randomly picked two instances from the data set and checked their labels, which are available for evaluation purpose but not visible to the clustering algorithm. If they had the same label, we generated a must-link constraint. Otherwise, we generated a cannot-link constraint.

The first data set is iris [13], which has 150 instances and 4 features. Three classes are represented in the data. Without any constrains, the k-means algorithm achieves an accuracy of 84%.

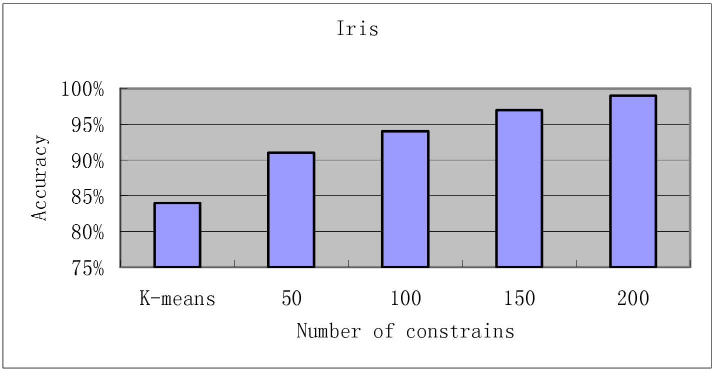

Fig. 1. MLP-KMEANS results on iris

Overall accuracy steadily increases with the incorporation of constrains, reach 99% after 200 random constrains.

We next turn to the Balance Scale data set [13], with 625 data instances and 4 attributes. It contains three classes. In this work, we randomly choose 20 instances for

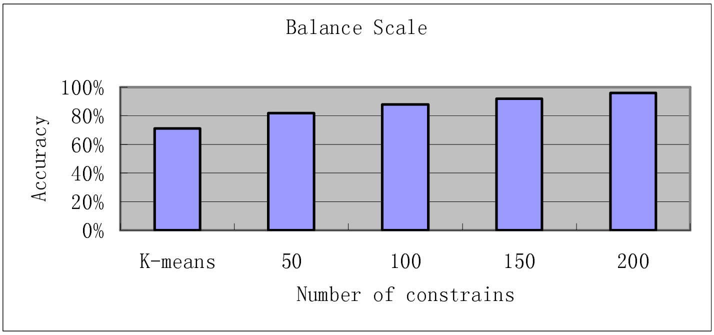

Fig. 2. MLP-KMEANS results on balance scale

each class. In the absence of constrains, the k-means algorithm achievers an accuracy of 71%. After incorporating 200 constrains, overall accuracy improves to 96%.

The third data set we used is soybean [13], which has 47 instances and 35 attributes. Four classes are represented in the data. Without any constrains, the k-means algorithm achieves an accuracy of 84%. After 100 random constrains, overall accuracy can reach 99%.

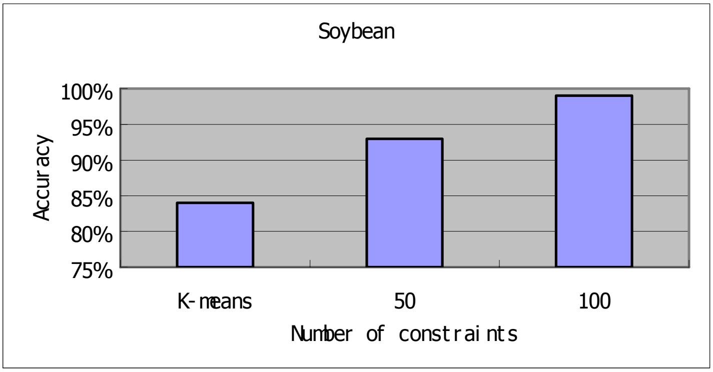

Fig. 3. MLP-KMEANS results on soybean

6 Conclusions and Future Work

In this paper, we propose a new data clustering method. It is a combination of Multi-Layer Perceptron and K-means. This method could make use of background information in the form of instance-level constrains. In experiments with random constrains on three data sets, we have shown significant improvements in accuracy.

In the future, we will explore how background information can be utilized in real applications. Then we will use our method in practical applications. Furthermore, background information also includes other form in addition to pairwise constrains, such as user feedback. We need to consider how to utilize those kinds of information in our method.

Acknowledgements. This research was supported by the MIC (Ministry of Information and Communication), Korea, Under the ITFSIP (IT Foreign Specialist Inviting Program) supervised by the IITA (Institute of Information Technology Advancement).

References

1. Xu, R., Wunsch, D.: Survey of Clustering Algorithms. In IEEE Transaction on Neural Networks, Vol. 16, (2005) 645-678
2. Jain, A.K., Murty, M.N., Flynn, P.J.: Data Clustering: A Review. In ACM Computing Surveys, Vol. 21 (1999) 264-323

3. Basu, S.: Semi-supervised Clustering with Limited Background Knowledge. In Proc. of the Ninth AAAI/SIGART Doctoral Consortium, (2004) 979-980
4. Basu, S., Arindam, B., Raymond J.M.: Semi-supervised Clustering by Seeding. In Proc. of the Nineteenth International Conference on Machine Learning (ICML), (2002) 19-26
5. Nizar, G., Michel C., Nozha B.: Unsupervised and Semi-supervised Clustering: a Brief Survey. A Review of Machine Learning Techniques for Processing Multimedia Content, 2004, http://www-rocq.inria.fr/~crucianu/src/BriefSurveyClustering.pdf
6. Luo, M., Ma, Y.F., Zhang, H.J.: A Spatial Constrained K-means Approach to Image Segmentation. In Proc. of the Fourth Pacific Rim Conference on Multimedia, (2003) 738-742
7. Bellot, P., Marc, E.B.: A clustering method for information retrieval. Technical Report IR-0199, Laboratoire d'Informatique d'Avignon, France, (1999)
8. Wagstaff, K.: Intelligent Clustering with Instane-level Constrains. Ph.D Thesis, Cornell University, (2002)
9. Basu, S.: Comparing and Unifying Search-based and Similarity-based Approaches to Semi-Supervised Clustering. In Proc. of the 20th International Conference on Machine Learning (ICML 2003), (2003) 42-49
10. Kiri, W., Claire, C., Seth, R., Stefan, S.: Constrained K-means Clustering with Background Knowledge. In Proc. of 18th International Conference on Machine Learning (ICML 2001), (2001) 577-584
11. MacQueen, J.B.: Some Methods for Classification and Analysis of Multivariate Observations. In Proc. of the Fifth Symposium on Math, Statistics, and Probability, Berkeley, CA: University of California Press, (1967) 281-297
12. Rand, W.M.: Objective Criteria for the Evaluation of Clustering Methods, In J. of the American Statistical Association, (1971) 846-850
13. Blake, C., Merz, J.: UCI Repository of Machine Learning Databases. http://www.ics.uci.edu/~mlearn/MLRepository.html