Proc. of the Intl. Conf. on Future Trends in Computing and Communication -- FTCC 2013

ISBN: 978-981-07-7021-1

Generalizations of K-Means Algorithms for Constrained Clustering

Sadaaki Miyamoto, Haruka Kondoh

***Abstract*—A generalization of K-means clustering algorithms including cluster size variables and covariance variables are introduced. Moreover an on-line version of that is considered. A constrained clustering algorithm using these generalizations and the idea in the COP K-means is proposed. Performances of the proposed algorithms are compared using numerical examples.**

***Keywords*—Generalized K-means clustering, on-line algorithm, constrained clustering**

I. Introduction

With the increasing use of classification methods in a variety of applications, data clustering techniques [1,2,3] are also focused upon by many researchers. Special attention is paid to semi-supervised classification [4,5] that handles a small number of labeled samples and a larger number of unlabeled samples. The setting of semi-supervised classification has invoked a class of new methods in clustering called constrained clustering [6]. Two well-known techniques are the COP K-means [7] and constrained mixture of distributions [8,9,10].

We have studied a generalization of K-means that includes cluster size variables and covariance variables within clusters [11]. In this paper we proceed to propose algorithms of constrained clustering using the idea of COP K-means and the generalization of the basic K-means. An on-line version (cf. [2,3]) of the algorithms are also considered. The performances of these algorithms are shown by using an artificial data set that exhibits characteristics of the proposed algorithms and a real data set.

The rest of this paper is organized as follows. Section 2 provides preliminaries of the basic K-means and the COP K-means using pairwise constraints together with the generalizations with the additional variables. Section 3 then is devoted to the discussion of the proposed method of constrained clustering herein. Numerical examples are given in Section 4, where performances of different algorithms are compared. Finally, Section 5 concludes the paper.

Sadaaki Miyamoto, Haruka Kondoh
University of Tsukuba
Japan

This study has partly been supported by the Grant-in-Aid for Scientific Research, JSPS, Japan, No.23500269

II. Preliminary Consideration

Notations are given first and then the basic K-means algorithm and its generalization is discussed. The COP K-means algorithm is also introduced.

A. Notations

Let R^p be the *p*-dimensional Euclidean space and the set of objects for clustering is denoted by $X = \{x_1, \ldots, x_n\}$. Each object $x \in X$ is a point of the Euclidean space. The squared Euclidean distance is denoted by

$$D(x, y) = \| x - y \|^2 = \sum_j (x^j - y^j)^2.$$

The squared Mahalanobis distance is also used, which is denoted by

$$D(x, y; S) = (x - y)^T S^{-1} (x - y).$$

Clusters $G_1, \ldots, G_K$ are subsets of a partition of X:

$$\bigcup_{i=1}^{K} G_i = X, \qquad G_i \cap G_j = \emptyset \quad (i \neq j)$$

The cluster center denoted by $v(G_i)$ for G_i is given by the centroid (the center of gravity):

$$v(G_i) = \frac{1}{|G_i|} \sum_{x_k \in G_i} x_k$$

where $|G_i|$ is the number of elements in G_i. $v(G_i)$ is also written as v_i for simplicity.

B. K-means and a generalization

The K-means algorithm has different origins, but the name is after [12]. The basic and simple algorithm of K-means is as follows.

Step 1. Set initial clusters and calculate cluster centers as centroids of the clusters.

Step 2. Allocate each object to the cluster of the nearest cluster centers.

Proc. of the Intl. Conf. on Future Trends in Computing and Communication -- FTCC 2013

ISBN: 978-981-07-7021-1

Step 3. If the clusters are convergent, stop. Else calculate new cluster centers as the centroids of the new members of the clusters. Go to Step 2.

Here, the word 'nearest' means the shortest Euclidean distance.

There are many variations of this algorithm. We consider a generalization that includes cluster size variables

$$\alpha = (\alpha_1, \ldots, \alpha_K)$$

and covariance variables

$$S = (S_1, \ldots, S_K)$$

within clusters. Clusterwise Mahalanobis distance $D(x, y; S_i)$ is thus used. Moreover a positive parameter λ is introduced. The algorithm is as follows.

Step 1. Set initial clusters and calculate cluster centers as centroids (the center of gravity) of the clusters. Calculate the cluster size and cluster covariances:

$$\alpha_i = \frac{|G_i|}{n},$$

$$S_i = \frac{1}{|G_i|} \sum_{x_k \varepsilon G_i} (x_k - v_i)(x_k - v_i)^T.$$

Step 2. Allocate each object $x \in X$ using

$$x \in G_i \Leftrightarrow i = \arg\min_{1 \le j \le K} \{D(x, v_j; S_j) - \lambda \alpha_j\}$$

Step 3. If the clusters are convergent, stop. Else calculate new cluster centers as the centroids of the new members of the clusters. Calculate also the new cluster size and the covariance using the same formulas. Go to Step 2.

C. *K-means with pairwise constraints*

Constrained clustering uses two sets $ML = \{(x, x')\}$ and $CL = \{(y, y')\}$ of pairs of objects. They are is called the set of must-links and that of cannot-links, respectively. $(x, x') \in ML$ means that (x, x') has to be in the same cluster, while $(y, y') \in CL$ means that (y, y') should not be in the same cluster.

A simple algorithm related to K-means called the COP K-means has been developed that takes pairwise constraints into account.

Step 1. Perform K-means with an initial value.

Step 2. If a certain constraint is broken, return FAILURE, else return SUCCESS.

In this COP K-means algorithm, each object should be allocated to the cluster of the nearest center that does not break the constraints.

III. Generalized On-Line K-Means and Constrained Clustering

We propose an on-line version [2] of the generalized K-means algorithm and constrained clustering using the idea of the COP K-means. The derivation of the generalized on-line algorithm is somewhat complicated and the details are omitted here.

A. *Generalized On-line K-means*

In an ordinary K-means algorithm, centroids are updated after all objects are reallocated. In contrast, they are updated after an object is reallocated in an on-line version of the K-means algorithm [2]. More precisely, assume that an object x_k is moved from cluster G_i to G_j using the same equation:

$$x_k \in G_j \Leftrightarrow j = \arg\min_{1 \le l \le K} \{D(x_k, v_l; S_l) - \lambda \alpha_l\}.$$

Then the two centroids $v(G_i)$ and $v(G_j)$ should be updated:

$$v(G_i) = v(G_i) - \frac{x_k - v(G_i)}{|G_i| - 1},$$

$$v(G_j) = v(G_j) - \frac{x_k - v(G_j)}{|G_j| + 1}.$$

In the generalized K-means algorithm, the cluster size α_i and α_j, as well as the covariance S_i and S_j should also be updated when x_k is moved from G_i to G_j:

$$\alpha_i = \alpha_i - \frac{1}{n}, \qquad \alpha_j = \alpha_j + \frac{1}{n}$$

$$S_i = \frac{B_i'}{|G_i'|} - v_i v_i^T$$

$$S_j = \frac{B_j'}{|G_j'|} - v_j v_j^T$$

$$S_i^{-1} = |G_i'| B_i'^{-1} + \frac{|G_i'|^2}{1 - |G_j'| v_i^T B_i'^{-1} v_i} B_i'^{-1} v_i v_i^T B_i'^{-1}$$

$$S_j^{-1} = |G_j'| B_j'^{-1} + \frac{|G_j'|^2}{1 - |G_j'| v_j^T B_j'^{-1} v_j} B_j'^{-1} v_j v_j^T B_j'^{-1}$$

where

$$B_i = \sum_{x \in G_i} x x^T, \quad B_j = \sum_{x \in G_j} x x^T$$

Proc. of the Intl. Conf. on Future Trends in Computing and Communication -- FTCC 2013

ISBN: 978-981-07-7021-1

and

$$B_i' = B_i + x_k x_k^T \quad B_j' = B_j - x_k x_k^T$$

Note that the Sherman-Morrison formula [13] for calculating the inverses of S_i and S_j are used.

Hence the above calculations are repeated until convergence. The generalized algorithm in the former section (Section II.B) is called a batch algorithm in contrast to the on-line algorithm in this section.

B. *Constrained clustering*

Constrained clustering using the generalized K-means with the pairwise constraints is straightforward: we can just use the algorithm in Section II.C with the generalized K-means formulas. On-line COP K-means with the generalized algorithm can also be developed without difficulty, as seen in the description of the COP K-means in the same section.

IV. Numerical Examples

We show two numerical examples. First example uses an artificial data set of points on the plane. Second example handles a real data set in the UCI repository [14].

A. *An artificial data*

Figure 1 shows an artificial data set on the plane. Suppose that we wish to divide them into two clusters (*K=2*) and the true clusters are a small group in the upper side and the larger one in the lower side. The point is that the configuration of points looks like the two 'wedges' and at the same time two linear groups. The results of the constrained clustering are shown in Fig. 2 and Fig. 3, where the vertical axis shows the value of the Rand index to measure the accuracy of the results. The former figure uses the must-links alone without any cannot-link, while the latter figure uses the cannot-links alone without any must-link. The procedure of the experiment was as follows.

The four methods of

A. the COP K-means (the red curve),

B. an EM algorithm using Shental's method [10] (the green curve),

C. the generalized COP K-means of the batch version (the blue curve), and

D. the generalized COP K-means of the on-line version (the pink curve)

were used. The parameter value was $\lambda = 10$. The horizontal lines of Figs. 2 and 3 imply the randomly generated number of must-links and cannot-links, respectively. To generate the results, we used 100 trials with randomly generated initial values to avoid dependency to the initial values.

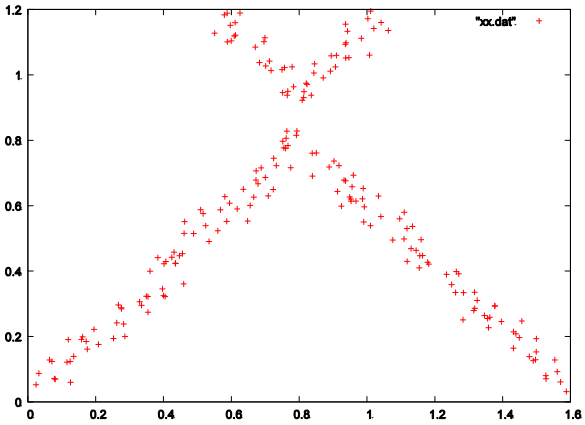

Figure 1. An artificial data set

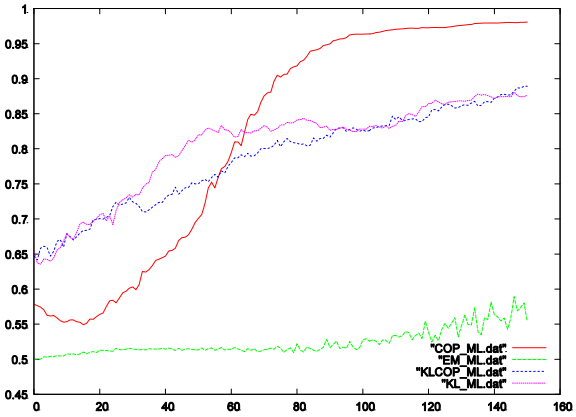

Figure 2. Comparison of different methods using must-links.

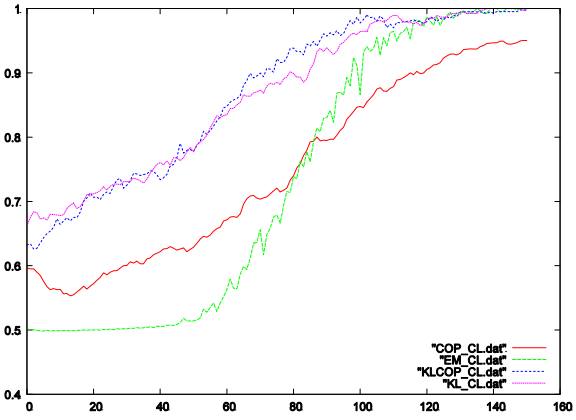

Figure 3. Comparison of different methods using cannot-links.

Proc. of the Intl. Conf. on Future Trends in Computing and Communication -- FTCC 2013

ISBN: 978-981-07-7021-1

B. *Comparison on a set of real data*

MONK data set in [14] was selected for comparing performances of different algorithms. The results are shown in Fig. 4 and Fig. 5, where the former result is with must-links alone and the latter with cannot-links alone. The four methods applied to this data were the same as those for the artificial data set, except that the results from COP K-means is shown by blue curves, those from the EM algorithm by pink curves, those from the batch version of the generalized K-means by red, and those from the on-line version of the generalized algorithm is by green.

C. *Discussion on two results*

An overall tendency observed from these figures shows that cannot-links work better than the must-links. Thus to mix must-links and cannot-links is not useful than to use cannot-links alone. When we compare the four methods in these examples, it seems that the both methods, i.e., the on-line and the batch versions, of the generalized K-means work better than the other two methods with the exception of the artificial data set with must-links, where the simple COP K-means seems the best of the four.

Note that to mix must-links and cannot-links is not useful in general, judging from our experiences in many experiments.

Although we omit the details, we cannot easily tell which is the best algorithm in general, as the performance seems to be dependent on the data. However, we can at least say that the generalized K-means algorithms are useful, judging from the experiments.

V. Conclusion

We have developed two generalized K-means algorithms for constrained clustering. Both batch version and on-line versions were included. Two examples were tested and the developed methods worked well on these data. As a future study, we should continue numerical experiments on many data sets. Moreover other variations such as kernel-based K-means with constraints should be developed and tested using real examples.

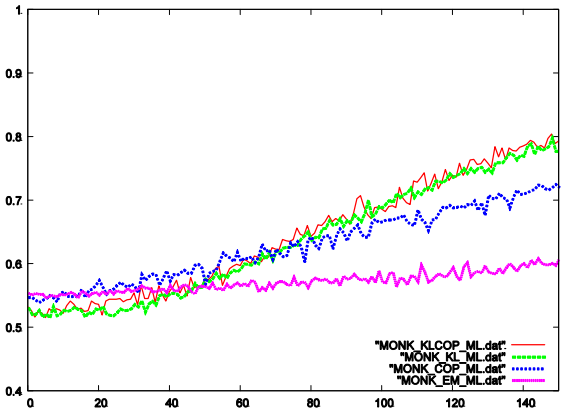

Figure 4. Comparison of different methods using must-links on MONK data set.

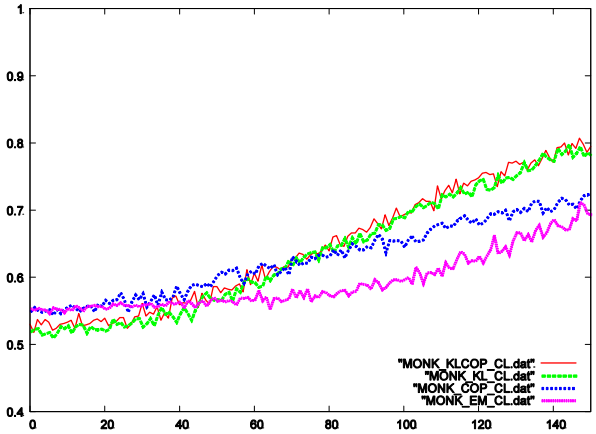

Figure 5. Comparison of different methods using cannot-links on MONK data set.

References

[1] B. S. Everitt, Cluster Analysis,3rd ed., Arnold, 1993.

[2] R. O. Duda, P. E. Hart, Pattern Classification and Scene Analysis, Wiley, 1973.

[3] R. O. Duda, P. E. Hart, D. G. Stork, Pattern Classifiation, 2nd ed., Wiley, 2001.

[4] O. Chapelle, B. Schoelkopf, A. Zien, eds., Semi-Supervised Learning, MIT press, 2006.

[5] X. Zhu, A. B. Goldberg, Introduction to Semi-Supervised Learning, Morgan and Claypool Publishers, 2009.

[6] S. Basu, I. Davidson, K. Wagstaff, Constrained Clustering: Advances in Algorithms, Theory, and Applications, CRC Press, 2009.

[7] K. Wagstaff, C. Cardie, S. Rogers, S. Schroedl, Constrained K-means Clustering with Background Knowledge, Proc. of the 9th ICML, pp.577--584, 2001.

[8] S. Basu, A. Banerjee, R. Mooney, Semi-supervised Clustering by Seeding, Proc. of the 19th International Conference on Machine Learning, pp.19--26, 2002.

[9] S. Basu, A. Banerjee, R. Mooney, Active Semi-Supervision for Pairwise Constrained Clustering, Proc. of the 2004 SIAM International Conference on Data Mining, (SDM-04) , pp.333--344, 2004.

[10] N. Shental, A. Bar-Hillel, T. Hertz, D. Weinshall, Computing Gaussian Mixture Models with EM Using Equivalence Constraints, In: Advances in Neural Information Precessing Systems, Vol.16, pp.465--472, 2004.

[11] S. Miyamoto, H. Ichihashi, K. Hoonda, Algorithms for Fuzzy Clustering, Springer, 2008.

[12] J. B. MacQueen, Some Methods of Classification and Analysis of Multivariate Observation, Proc. of 5th Berkeley Symposium on Math. Stat. and Prob., pp.281--297, 1967.

[13] J. Sherman, W. J. Morrison, Adjustment of an Inverse Matrix Corresponding to Changes in the Elements of a Given Column or a Given Row of the Original Matrix, Ann. Math. Stat., Vol.20, No.4, pp.620-624, 1949.

[14] UCI Machine Learning Repository, \url{http://archive.ics.uci.edu/ml/} (accessed January 6, 2013).