K-maximin clustering: a maximin correlation approach to partition-based clustering

Taehoon Lee[1], **Seung Jean Kim**[2,3], **Eui-Young Chung**[4], **and Sungroh Yoon**[1,a)]

[1] *Electrical Eng., Korea University, Seoul 136–713, Korea*

[2] *Electrical Eng., Stanford University, Stanford, CA 94305, USA*

[3] *Citi Alternative Investments, New York, NY 10013, USA*

[4] *Electrical and Electronic Eng., Yonsei University, Seoul 129–749, Korea*

a) *sryoon@korea.ac.kr*

Abstract: We propose a new clustering algorithm based upon the *maximin correlation analysis* (MCA), a learning technique that can minimize the maximum misclassification risk. The proposed algorithm resembles conventional partition clustering algorithms such as k-means in that data objects are partitioned into k disjoint partitions. On the other hand, the proposed approach is unique in that an MCA-based approach is used to decide the location of the representative point for each partition. We test the proposed technique with typography data and show our approach outperforms the popular k-means and k-medoids clustering in terms of retrieving the inherent cluster membership.

Keywords: data mining, clustering, maximin correlation, k-means

Classification: Science and engineering for electronics

References

[1] J. Han and M. Kamber, *Data Mining*, 2nd ed., Morgan Kaufmann, San Francisco, CA, 2006.

[2] I. H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques*, 2nd ed., Morgan Kaufmann, CA, 2005.

[3] T. Hastie, R. Tibshirani, and J. H. Friedman, *The Elements of Statistical Learning, Springer*, Heidelberg, Germany, 2003.

[4] H. Avi-Itzhak, J. V. Mieghem, and L. Rub, "Subclass pattern recognition: A maximin correlation approach," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 17, no. 4, pp. 418–431, April 1995.

1 Introduction

Cluster analysis or *clustering* [1] is a popular technique that can group similar objects together while separating dissimilar groups. Depending on specific similarity measures and grouping algorithms used, various clustering algo-

© IEICE 2009
DOI: 10.1587/elex.6.1205
Received June 06, 2009
Accepted August 08, 2009
Published September 10, 2009

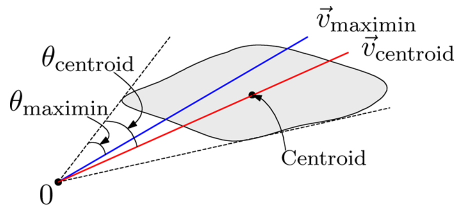

$\vec{v}_{\text{maximin}}$
$\vec{v}_{\text{centroid}}$
θ_{centroid}
θ_{maximin}
Centroid
0

Fig. 1. Maximin versus centroid

rithms can be defined. *Partitioning methods* perform partitioning the entire set of data, and resulting clusters do not overlap. *Hierarchical methods* aim at discovering hierarchical structures among objects, and the clusters found are usually allowed to overlap. *Density-based clustering* finds highly populated regions. *Model-based clustering* assumes a model for each cluster and finds the best fit of the model. *Grid-based methods* consider in the object space a number of cells that form a grid and perform clustering thereon.

The most popular partitioning methods may be k-means and k-medoids [1, 2, 3]. These techniques iteratively partition input data into k clusters, where each cluster center is the mean or median value of the objects in the cluster. In every iteration of these techniques, each object is assigned to its closest center. After one iteration, the center location of each cluster is re-calculated, which is repeated until convergence. These methods have been very popular in various disciplines, mainly due to their simplicity in implementation and reasonable performance in many applications. The worst-case complexity of k-means and k-medoids is $O(kNt)$, where k, N and t represent the numbers of clusters, objects and iterations, respectively [1].

We propose a new partitioning algorithm called *k-maximin*. The main difference between k-maximin and the conventional approaches lies in how the center of a cluster is determined. In k-maximin, each cluster center indicates the direction that maximizes the worst-case (*i.e.,* minimum) correlation with the group members (Fig. 1). When correlation is used as a distance measure, representing a group by this direction minimizes the maximum misclassification risk [4]. The problem of the *maximin correlation analysis* (MCA) [4] is to find such a direction. The difference between the work by [4] and the proposed method is that the former is for *supervised* classification, whereas the latter is for *unsupervised* clustering. To the best of the authors' knowledge, k-maximin is the first MCA-based unsupervised learning technique.

The center of a group computed by MCA serves as a representative value of the group, as the mean or median does. Thus, k-maximin clustering can be applied to various applications in which the notion of a group center makes sense. Furthermore, when a cluster can be regarded as a collection of subclusters, producing stratified layers of clusters, k-maximin should be very effective. This is because the MCA method, which k-maximin is based upon, can find the direction that can optimally represent a collection of subclusters.

© IEICE 2009
DOI: 10.1587/elex.6.1205
Received June 06, 2009
Accepted August 08, 2009
Published September 10, 2009

2 K-maximin clustering

2.1 Maximin Correlation Analysis (MCA)

The notion of MCA was originally proposed for multiple subclass pattern recognition, where each class is given as a collection of subclass templates [4]. For example, in multi-font optical character recognition, we are interested in recognizing not only each character but also its font. Each character is thus represented by a set of templates, each of which corresponds to a specific font representation of that character. For each character class, an aggregate template, which can represent multiple templates of the class, is sought. When correlation is used as a distance measure, the MCA-based template is optimal in that it maximizes the worst-case (*i.e.,* minimum) correlation with its subclass templates [4].

For example, consider Fig. 1 that compares the maximin-center and the centroid of a group. The two vectors, $\vec{v}_{\text{maximin}}$ and $\vec{v}_{\text{centroid}}$, represent the direction of the maximin-center and the centroid, respectively. The vector $\vec{v}_{\text{maximin}}$ is set in such a way that the worst-case (*i.e.,* minimum) correlation between $\vec{v}_{\text{maximin}}$ and group members is maximized. In contrast, the vector $\vec{v}_{\text{centroid}}$ simply represents the direction set by the centroid. Thus, $\theta_{\text{maximin}} < \theta_{\text{centroid}}$, which means that the worst-case correlation between $\vec{v}_{\text{maximin}}$ and group members is greater than the worst-case correlation between $\vec{v}_{\text{centroid}}$ and group members. This is because smaller θ between two vectors means greater correlation between them.

Consider two nonzero vectors x and y in $\mathbb{R}^n$. The *centered correlation* between x and y is then given by $\phi(x,y) = \frac{x^T y}{\|x\|\|y\|}$. For a nonzero vector x and a non-empty set $\mathcal{Y} \subseteq \mathbb{R}^n$, the *minimum correlation* between x and $\mathcal{Y} \subseteq \mathbb{R}^n \backslash \{0\}$ is defined as $\phi(x,\mathcal{Y}) = \inf_{y\in\mathcal{Y}} \phi(x,y)$, where inf denotes the infimum or the greatest lower bound. In MCA, we are interested in finding a nonzero vector $x \in \mathbb{R}^n$ that maximizes the minimum correlation with the set $\mathcal{Y}$:

$$\begin{array}{ll} \text{maximize} & \phi\,(x,\mathcal{Y}) \\ \text{subject to} & x \neq 0. \end{array} \tag{1}$$

An iterative solution to Eq. (1) was proposed in [4]. For fixed n, its worst-case complexity is $O(m^2)$, where m is the number of objects in $\mathcal{Y}$.

2.2 Proposed K-maximin algorithm

Algorithm 1 outlines the proposed k-maximin clustering algorithm. Given k, the number of clusters and D, a data set of N objects, the proposed k-maximin clustering returns a set of k clusters. In Line 2, the initial set of k centers are chosen arbitrarily out of D or by using some heuristics such as pre-clustering samples. The repeat block in Lines 4–6 constitutes an iteration. In each iteration, every object o in D is compared with the k maximin-centers and gets assigned to the cluster whose maximin-center has the largest correlation with o (*i.e.,* the closest maximin-center). Once the membership of every data object has been updated, the maximin-center of each cluster is re-calculated in Line 6 by MCA to find the new direction that can optimally

© IEICE 2009
DOI: 10.1587/elex.6.1205
Received June 06, 2009
Accepted August 08, 2009
Published September 10, 2009

Algorithm 1: K-Maximin Clustering

```
Input  : $k$, the number of clusters
Input  : $D$, a data set containing $N$ objects
Output: A set of $k$ clusters
1 begin
2     Choose $k$ initial centers;
3     repeat
4         foreach object $o$ in $D$ do
5             Assign $o$ to the cluster that has the maximin-center closest to $o$;
6         Update the maximin-center of each cluster by solving MCA formulated in Eq. (1);
7     until no change ;
8 end
```

represent the current cluster. Each cluster corresponds set $\mathcal{Y}$, and Eq. (1) is solved to find x, the new maximin-center of the current cluster represented by $\mathcal{Y}$. (The details of solving Eq. (1) is beyond the scope of this work; the interested reader is directed to [4].) The algorithm terminates if there is globally no change in cluster membership.

The worst-case complexity of Algorithm 1 remains linear in k and N, although the process to compute the maximin-center of a cluster can take at most $O(m^2)$ if the iterative method by [4] is used. Typically, m is small (*i.e.*, $m \ll N$), and calculating maximin-centers usually incurs negligible computational overhead.

3 Experimental results and discussion

We implemented the proposed k-maximin algorithm with MATLAB and tested it with typography data. For each of the 26 English capital letters, we created its images in 6 different fonts[1] and represented each image by an $f \times f$ binary vector, where $f = 10$, 20, 30 and 40. Fig. 2 (a) shows letter ‘A’ in the 6 different 40×40 fonts used. For each value of f, the images were collected into a single data set, which was then clustered by the k-maximin, k-means, and k-medoids algorithms in turn. The objective was to group all images of each letter into a cluster, producing one cluster per letter in the ideal case. Since the execution time of all algorithms was negligible (in the order of tens of seconds), the focus of the comparison below is on clustering accuracy (*i.e.*, how well each character set was rediscovered by clustering).

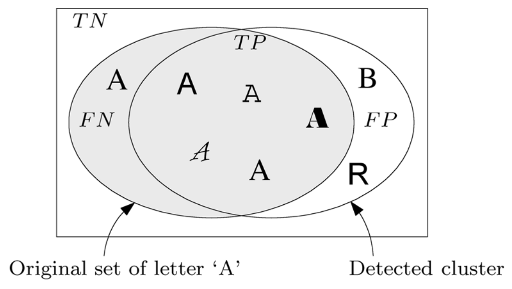
For quantitative comparison, we first define the concepts of *true positive* (TP), *false positive* (FP), *false negative* (FN) and *true negative* (TN), as informally shown in Fig. 2 (b). The shaded ellipse represents the original set of letter ‘A’ in different fonts. The other uncolored ellipse corresponds to the cluster discovered by a certain clustering algorithm (one of the three algorithms tested). We define true positives as those images of ‘A’ that have been successfully clustered as ‘A’. False negatives are the images of ‘A’ that fail to be clustered as ‘A’. False positives are the images of other letters that have been clustered as ‘A’. True negatives are the images of other letters that are not clustered as ‘A’. To decide which letter each discovered cluster

[1]The fonts used are as follows: Arabic Typesetting, Arial, Baskerville Old Face, Calibri, Eras Light ITC and Times New Roman.

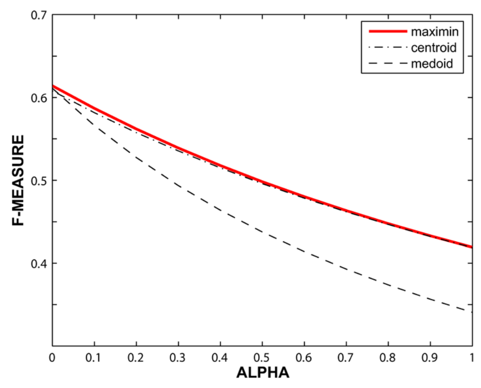
© IEICE 2009
DOI: 10.1587/elex.6.1205
Received June 06, 2009
Accepted August 08, 2009
Published September 10, 2009

© IEICE 2009
DOI: 10.1587/elex.6.1205
Received June 06, 2009
Accepted August 08, 2009
Published September 10, 2009

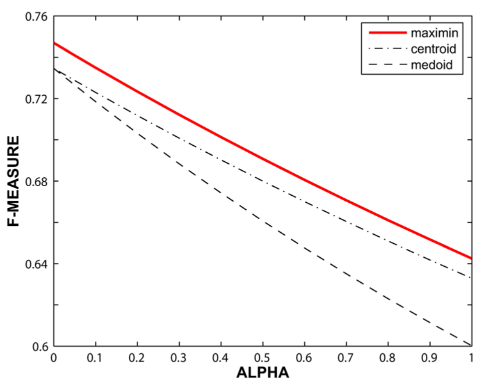
(a) Fonts



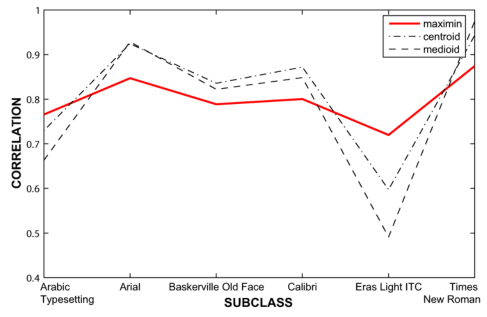
(b) Definitions of TP, FP, FN, TN



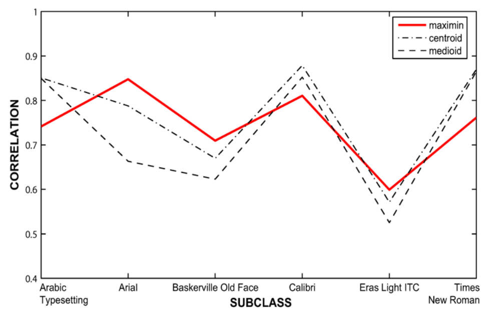
(c) 10×10 fonts (= 100 dimensions)



(d) 40×40 fonts (= 1600 dimensions)



(e) 10×10 fonts



(f) 40×40 fonts

Fig. 2. Experiments

is representing, the most popular letter in that cluster was used.

Based on the above definitions, we calculated the F-measure [2], a widely used performance metric given by $\frac{1}{\alpha/\text{precision}+(1-\alpha)/\text{recall}}$, where precision = $\frac{TP}{TP+FP}$, recall = $\frac{TP}{FN+TP}$ and α is a parameter determining the weighting of precision and recall. Precision represents the portion of true positives in a detected cluster, and recall indicates the fraction of detected true positives out of an original group. F-measure can therefore combine recall and precision into a single performance measure. The F-measure of a perfect clustering algorithm would be 1.

The average F-measure values obtained by each clustering algorithm on the two extreme fonts sizes (10×10 and 40×40) are plotted in Fig. 2 (c) and Fig. 2 (d), respectively. Each plot shows how F-measure changes as varying α from 0 to 1. F-measure is equal to recall when $\alpha = 0$ and gradually becomes precision as α increases to 1. Thus, this plot can reveal the relative performance of the compared techniques over all possible combinations of weights on precision and recall. For the 10×10 fonts, the performance of k-maximin was similar to that of k-means (labeled as *centroid* in the figure), and k-medoids showed the worst performance. In contrast, k-maximin evidently outperformed k-means and k-medoids for the 40×40 fonts and consistently produced the highest level of F-measure for the entire range of α values. For the fonts of intermediate sizes, the relative performance among the three techniques exhibited a similar pattern.

This result suggests that k-maximin outperforms the alternatives by a larger margin as the dimensionality of data increases. That is, the proposed k-maximin tends to cope with the curse of dimensionality (or lack of data separation in high-dimensional space) [3] better than k-means or k-medoids. Given that more and more high-dimensional datasets are being generated in various disciplines such as multimedia processing and bioinformatics, it would be highly beneficial to devise a clustering method that can effectively handle high-dimensional space.

Another reason for the performance difference between k-maximin and the others may be due to the characteristic of the data used, in which each cluster consists of multiple subclusters. This fact might have lowered the performance of the conventional partitioning clustering algorithms tested, which were originally designed without considering the multiple-subcluster scenario. In contrast, the k-maximin algorithm is based on the maximin correlation analysis, which was designed for multiple subclass classification, and handles multiple-subcluster data more effectively.

To determine the root cause of the performance difference, we further investigated how well each type of center (*i.e.,* maximin, centroid, and medoid) can represent a cluster. Each plot in Fig. 2 (e) and Fig. 2 (f) compares the correlation of the different centers of the cluster for letter 'A' with the centers of the subclusters. Evidently, the worst-case (*i.e.,* minimum) correlation of the maximin-center is the largest among the three. We observed the same phenomena for all the other letters tested. This demonstrates why k-maximin clustering can produce fewer errors (thus more accurate results) than the alternatives when a cluster consists of multiple subclusters.

4 Conclusion

We have proposed a new partitioning algorithm based upon the maximin correlation analysis. This algorithm calculates the center of a cluster as the direction that maximizes the minimum correlation between the center and cluster members. We tested the proposed algorithm with an image data set in which each English capital letter is represented by multiple fonts. We observed that the proposed approach outperforms conventional partitioning clustering algorithms such as k-means and k-medoids in terms of F-measure, a widely used performance metric. We thus believe that the proposed algorithm can be a very useful alternative to the k-means and k-medoids algorithms, especially when the number of data dimensions are high and multiple subclusters constitute a cluster.

© IEICE 2009
DOI: 10.1587/elex.6.1205
Received June 06, 2009
Accepted August 08, 2009
Published September 10, 2009

Acknowledgments

This work was supported in part by KOSEF grants funded by the Korean Government (MEST)(No. 2009-0079888, No. 2009-0060369 and No. 2009-0068969).

DOI: 10.1587/elex.6.1205
Received June 06, 2009
Accepted August 08, 2009
Published September 10, 2009