5-2005

K-means Clustering with Multiresolution Peak Detection

Guanshan Yu
University of Nebraska-Lincoln, gyu@cse.unl.edu

Leen-Kiat Soh
University of Nebraska, lsoh2@unl.edu

Alan B. Bond
University of Nebraska-Lincoln, abond1@unl.edu

K-means Clustering with Multiresolution Peak Detection

Guanshan Yu
University of Nebraska-Lincoln
Department of Computer Science and Engineering
gyu@cse.unl.edu

Leen-Kiat Soh
University of Nebraska-Lincoln
Department of Computer Science and Engineering
lksoh@cse.unl.edu

Alan Bond
University of Nebraska-Lincoln
School of Biological Sciences
abond1@unl.edu

Abstract

Clustering is a practical data mining approach of pattern detection. Because of the sensitivity of initial conditions, k-means clustering often suffers from low clustering performance. We present a procedure to refine initial conditions of k-means clustering by analyzing density distributions of a data set before estimating the number of clusters k necessary for the data set, as well as the positions of the initial centroids of the clusters. We demonstrate that this approach indeed improves the accuracy and performance of k-means clustering measured by average intra to inter-clustering error ratio. This method is applied to the virtual ecology project to design a virtual blue jay system.

1. Introduction

Data mining uses a combination of machine learning, statistical analysis, modeling techniques and database technology to find patterns and subtle relationships in data and infers rules that allow the prediction of future results. K-means clustering is one of the most widely used approaches in data mining. It is used as the primary pattern recognition approach in the virtual blue jay project, in which a computer system is designed to simulate preying behaviors of the blue jays.

It has been realized that k-means clustering suffers from its sensitivity of initial conditions. Difficulties were encountered during the implementation of k-means clustering initialization to achieve high performance. In this paper, we address the issue of k-means clustering initialization and introduce an image quantization methodology, which effectively improves the intelligence of clustering initialization in k-means clustering. This strategy is tested to be efficient and can be used in a wide range of applications.

We will first introduce some background researches of the virtual ecology project, and related work of k-means clustering and initial condition refinement. Then we will introduce the multiresolution peak detection technique and density detection approach. At last, we will demonstrate some experimental results generated by applying proposed strategy to our data.

2. Background

2.1. Frequency-dependent Selection and Virtual Ecology

It has been noticed that the polymorphism among many prey types is caused by a phenomenon called *frequency-dependent selection*, in which prey types with similar appearances are attacked disproportionately more often by the predators [3]. It is not trivial to evoke and verify this phenomenon under controlled settings simply because evolution takes time. Bond and Kamil have successfully created a virtual ecology model [3], which made study of visual selection under artificial environment possible.

The system is driven by five captive blue jays that hunt for artificial, digital moths on computer displays. The digital moths (shown in Figure 1) are 64-level grayscale triangles with a height of 6mm, overlaid on different complex granular backgrounds. Each moth is constructed from 75 distinctive pixels, which we call phenotypes. These phenotypes are generated by a virtual chromosome of 160 val-

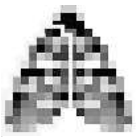

Figure 1. A digital moth constructed by 75 distinctive pixels

2005 IEEE International Conference on Electro Information Technology, Lincoln, NE, USA, 22-25 May 2005.
DOI: 10.1109/EIT.2005.1626978

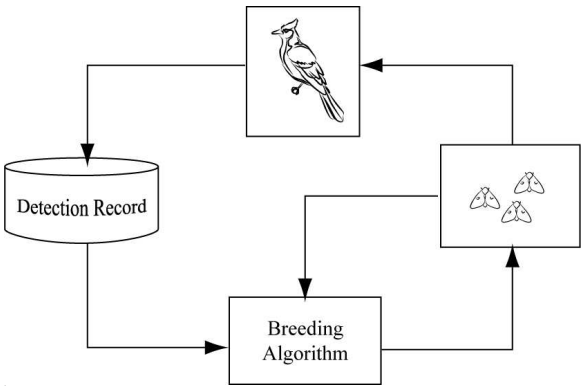

Figure 2. Virtual ecology

ues using genetic algorithms. To run the test, a generation of moths, which contains 200 individuals, is exposed to five specially trained blue jays, one at a time on different positions of the background. The accuracy and latency of the birds' detection of these moths are recorded. More specifically, the number of successful detections and the time spent on these detections in milliseconds are monitored and later used as a reference for the breeding algorithm. The offspring are generated by selecting moths that have higher surviving capability. A test run contains 100 successive, non-over lapping generations. The project diagram is shown in Figure 2.

The current virtual ecology has several drawbacks, one being the amount of time it required to collect data from the blue jays' response. Between each pair of successive tests, these blue jays need 30 days of exposure to the parental population under stationary, non-evolving conditions to return them to a consistent baseline. As a result, the average period of one test run can take up to 100 to 120 days. In addition, the dedication and motivation of the blue jays vary from time to time and cannot be controlled by the experiment. Because of this, a virtual blue jay system that functions similar to a real blue jay on preying the digital moths is desired in order to improve the efficiency of the experiment.

A fundamental problem of designing a virtual blue jay is the ability of pattern recognition among a moth population. This is an essential step in the study of the underline relationship between a moth group with a certain appearance and their capability of surviving predation. The virtual blue jay is expected to function in such a way that it identifies the group a moth belongs to and generates realistic preying statistics (accuracy and latency information), which resembles that of a real bird. This study can be done using clustering approaches.

2.2. Clustering

In its basic form, a clustering problem is defined as the problem of finding homogeneous groups of data points in a given data set [2]. Many studies on data mining and clustering have been conducted, such as the ones report in [9][10][1][7][4][13][6]. Disjoint clustering, which aims at partitioning a giving data set into disjoint subsets so that specific clustering criteria are optimized, is the simplest form of clustering. One of the most widely used disjoint clustering algorithms is the k-means clustering. It partitions the data set in the following steps:

1. Place k points into the space represented by the objects that are being clustered. These points represent initial group centroids.
2. Assign each object to the group that has the closest centroid.
3. When all objects have been assigned, recalculate the positions of the k centroids.
4. Repeat Steps 2 and 3 until the centroids no longer move. This produces a separation of the objects into groups from which the metric to be minimized can be calculated.

It is known that k-means clustering is especially sensitive to initial starting conditions. The performance of the clustering can be greatly affected by both the number of clusters k necessary and where the initial centroids of the k clusters are placed in the data space. Several studies have been done on this issue, including [8], which suggested a refinement algorithm that that defines a refined initial starting condition and helps to improve the solutions; [2], which introduces a recursive global k-means algorithm that setting initial centroids one at a time; [11], which proposes a method named the *gap statistics* for estimating the number of clusters in a set of data [5], which introduces a prediction-based resampling method, *Clest*, for estimating the number of clusters.

3. Methodology

We propose a new method that is based on an intuitive observation that the number of clusters and their centroids are largely determined by areas of different densities in the data space. Clusters with centroids located at the center of dense areas will result in low clustering error and thus better clustering performance.

The data space density analysis is based on a technique named multiresolution peak detection. It is proposed to estimate the number of clusters k and the locations of these clusters by detecting centroids of the potential dense areas. These centroids are chosen from the data points by scoring

each one of them based on a cumulative weight. This weight is obtained by summing up sub-weights on each coordinate based on its dominance over its neighborhood.

We will first introduce multiresolution peak detection. Then we will demonstrate how it is implemented to analyze the density of the data space.

3.1. Multiresolution Peak Detection

According to Soh [12], image quantization aims to encode data from a source into as few bits as possible in a way that reproduction may be recovered from the bits with as high quality as possible. One important step in digital image quantization is detecting peak values in a histogram. Multiresolution peak detection (MPD), first introduced by Soh in [12], is a dynamic and adaptive image quantization methodology. It refines peak values from the histogram of a digital image by analyzing the signals at different resolutions.

The essence of MPD is to use a dynamic peak-detection parameter O along the spectrum of the cumulative distribution function (CDF) to identify a range of peaks by their significance. O can be seen as a sliding window, in which each value in this window is compared to the local mean within the window. Those that greatly differ the local mean are candidates of peaks. With different O values, peaks of different scales can be recognized. Note that the bigger O is, the lower the resolution, and vice versa. MPD takes following steps:

1. Create a multiresolution map of the histogram peaks, which is an "integrated" histogram generated by setting different O values.
2. Track each peak through the scale space to score each peak's significance.
3. Filter out spurious peaks and localize significant peaks.

In particular, the scoring algorithm follows the principles below:

1. Peaks found at a larger O are more significant.
2. Peaks found at a smaller O are more accurate in terms of localization.
3. Neighboring peaks suggest a significant peak. A peak is more significant if it is surrounded by other peaks.
4. The significance of a peak is proportional to its height.

Figure 3 and 4 show an example of how MPD works. Notice how the peaks are weighted based on the 4 principles. For example, gray scale level 31 is weighted higher than 63 because 31 is surrounded by several other peaks, which all contribute to the weight of 31. This reflects principle 3.

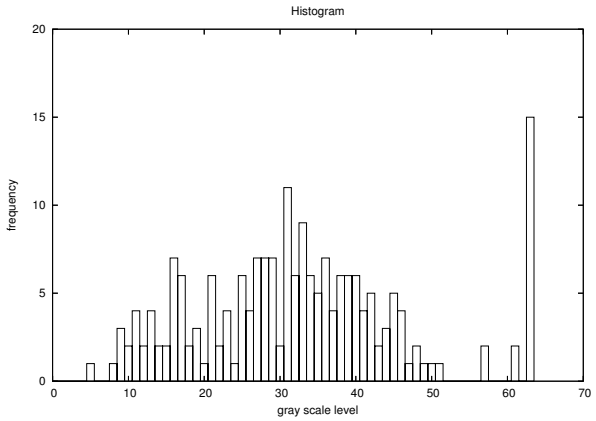

Figure 3. Histogram

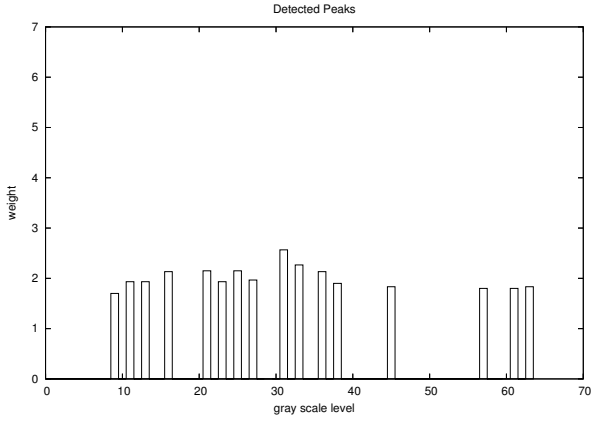

Figure 4. Detected peaks

3.2. Density Analysis

MPD has the ability to detect and weight peak values from a histogram, which essentially detects dense areas of a one-dimensional data set. Therefore, to calculate the density level of a multi-dimensional data space, we run MPD on each dimension of the data set and sum up the results on all dimensions (in our project, 75 dimensions). By doing so, each data point is assigned with an aggregate weight that indicates the degree of the point being in a dense region. An example is illustrated in Figures 5, 6 and 7. The two centroids (shown in circles in Figure 5) are determined by:

1. Detect peaks on x axis (Figure 6). This resulted in two peaks on $x = 1$ and $x = 2$ as they both have positive weights.
2. Detect peaks on y axis (Figure 5). This resulted in one peak on $y = 1$.

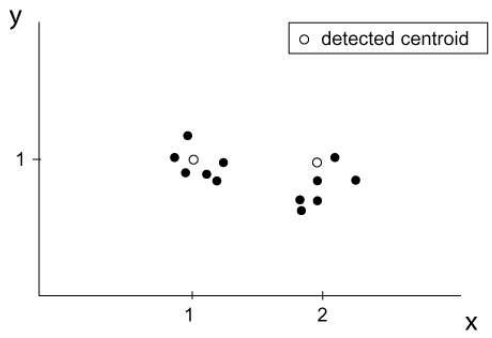

Figure 5. A 2D data set

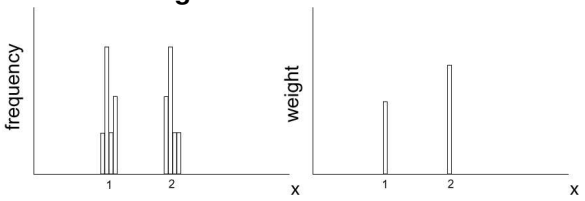

Figure 6. Histogram and detected peaks on x axis

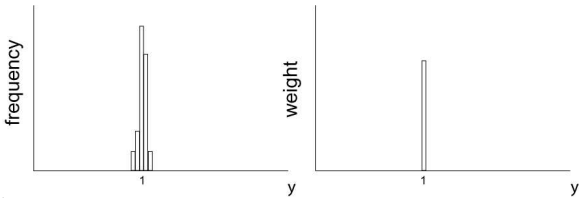

Figure 7. Histogram and detected peaks on y axis

3. For each point in the data set, calculate the overall weight by summing weights determined by steps 1 and 2.

4. With a weight assigned to each point, the number of clusters and their centroids are further determined by filtering using a threshold. Points with a weight value greater than the threshold are chosen to be initial centroids. Here we eventually select point $(1, 1)$ and $(2, 1)$ as centroids because their weights are larger than the chosen threshold.

When designing the virtual blue jay system, the threshold is obtained by running multiple trials on 3 sets of data from the virtual ecology project and choosing the one with the best performance.

4. Results

In order to test the performance of k-means clustering with MPD, let us first define average intra to inter-clustering error ratio, R. Average intra-cluster error, or clustering error, D_{intra} is defined as the average squared Euclidean distances between each data point x_i and the centroid m_j of the cluster x_i is in:

$$D_{intra} = \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{k} I(x_i \in C_j) \|x_i - m_j\|^2 \qquad (1)$$

where N is the size of the data set, k is the number of clusters, and $I(X) = 1$ if X is true and 0 otherwise. Average inter-clustering error D_{inter} is defined as the average squared Euclidean distance between every two cluster centroids m_i and m_j:

$$D_{inter} = \frac{1}{k^2} \sum_{i=1}^{k} \sum_{j=1}^{k} \|m_i - m_j\|^2. \qquad (2)$$

So R is:

$$R = \frac{D_{intra}}{D_{inter}} \qquad (3)$$

A small R indicates a good clustering performance and vice versa.

Figure 8 shows the results of k-means clustering using the MPD approach compared with the defacto standard of initialization, which randomly chooses initial cluster centroids. We choose $k = 7$ for random initialization, which provides the best performance after testing k ranged from 2 to 20 on our data. One may argue that the tests may favor the MPD approach by dynamically choosing a large k and thus reducing D_{intra} (clustering error). However, because equation (3) is used as the performance criterion instead of the conventional D_{intra} only, increasing k will result in a small D_{inter} as well. Therefore a large k will give very slight or even no effect on R.

The graphs indicate that the MPD approach results in a much lower R in general compared to the defacto standard of initialization. However, the performance of MPD declines near the end of each of the three data sets. This is due to the polymorphism phenomenon introduced earlier, in which moth population evolve into very diverse groups in terms of wing patterns. In other words, it shows that MPD does not perform well when the data points are sparse or less clustered. After all, the experiments still prove MPD's ability of choosing the right number of clusters k and estimating their centroids and thus is shown to give better clustering performance.

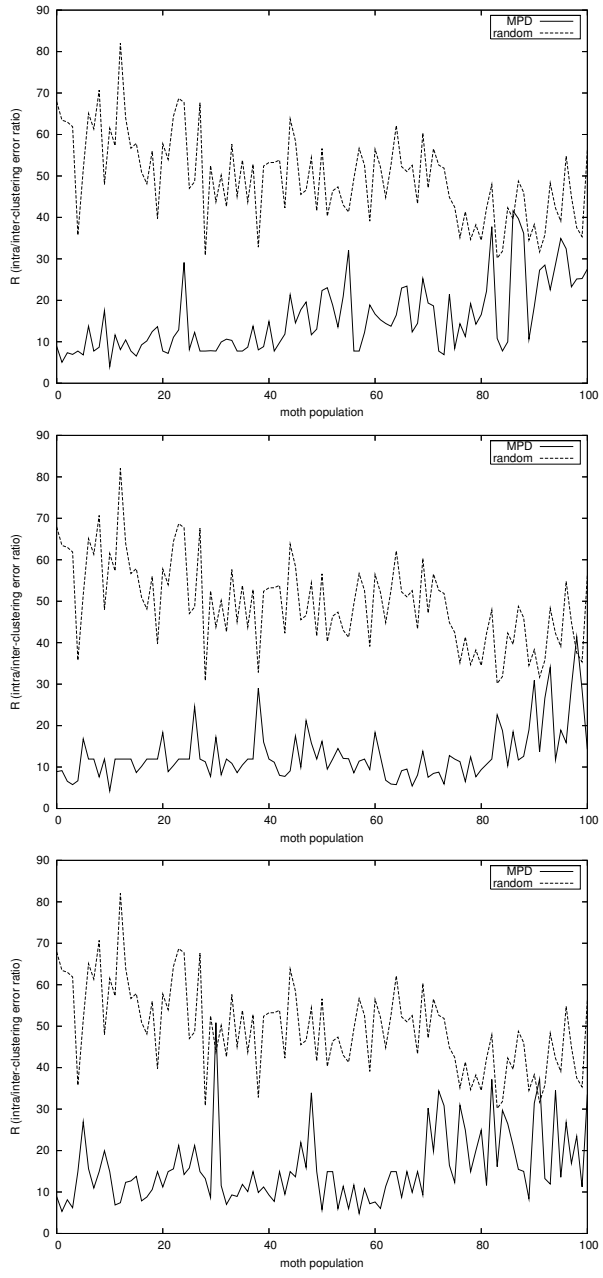

Figure 8. Test diagram on three sets of data

5. Conclusion

The multiresolution peak detection methodology has proved to be an effective way of implementing k-means clustering with high performance. It improves the accuracy of the clustering algorithm by refining the clustering initialization. It first evaluates the density distribution of the data space and then adaptively places centroids among areas with higher data density. However, the performance of this method suffers when the data space is very sparse.

Data analysis using clustering has established the framework of the pattern recognition functionality of the virtual blue jay. From here, a virtual blue jay prototype can be designed. Because k-means clustering using MPD is very efficient compared to many other iterative clustering approaches, the virtual blue jay prototype is able to read in a moth population dynamically, cluster it into groups, and use the associated detection records together with the clustering results as references to generate the detection accuracy and latency of preying an individual moth based on the group it belongs to.

Our future studies include refining the MPD algorithm to improve its performance on very sparse data sets; implementing k-means clustering in the frequency domain (it is currently implemented in spatial domain); and testing MPD on other clustering algorithms, such as overlapping clustering, hierarchical clustering and probabilistic clustering.

References

[1] A. Borgida, and R. J. Brachman. “Loading Data into Description Reasoners.” *SIGMOD* (1993): 217-226.

[2] A. Vlassis, and J. Verbeek. “The global k-means clustering algorithm.” *Pattern Recognition* 36, 2 (2003): 451-461.

[3] A. Bond, and A. Kamil. “Visual predators select for crypticity and polymorphism in virtual prey.” *Nature* 415 (2002): 609-613.

[4] D. Keim and H. Kriegel and T. Seidl. “Supporting Data Mining of Large Databases by Visual Feedback Queries.” *Data Engineering* 10(1994): 302-313

[5] G. Piatetsky-Shapiro, and W. J. Frawley. *Knowledge Discovery in Databases*. AAAI/MIT Press, 1991

[6] J. Han, Y. Cai and N. Cercone. “Knowledge Discovery in Databases: an Attribute-Oriented Approach.” *VLDB* 18(1992): 574-559.

[7] L.K. Soh, “Multiresolution, dynamic and adaptive image quantization methodology: automation and analysis.” *Journal of Electronic Imaging* 12(2) (2003): 229-243.

[8] P. Bradley, and U. Fayyad. “Refining Initial Points for K-Means Clustering.” *International Conf on Machine Learning, Morgan kaufmann* 15(1998).

[9] R. Agrawal, S. Ghosh, T. Imielinski, B. Iyer, and A. Swami. "An Interval Classifier for Database Mining Applications." *VLDB* 18 (1992): 560-573.

[10] R. Agrawal, S. Ghosh, T. Imielinski, B. Iyer, and A. Swami. "Mining Association Rules between Sets of Items in Large Databases." *SIGMOD* (1993): 207-216.

[11] R. Tibshirani, G. Walther, and T. Hastie. "Estimating the number of clusters in a dataset via the gap statistics. Technical report." Stanford University, September 2001.

[12] S. Dudoit, and J. Fridlyand. "A prediction-based resampling method for estimating the number of clusters in a dataset." *Genome Biol* 3 (2002): 1-21.

[13] W. Lu, J. Han and B.C. Ooi. "Discovery of General Knowledge in Large Spatial Databases." *Far East Workshop on Geographic Information Systems, Singapore* (1993): 275-289