A k-means++ Algorithm for Internet Shopping Search Engine

Jian-Ji Ren*, Jae-kee Lee*
*Dept. of Computer Engineering, Dong-A University
jimey@donga.ac.kr

Abstract

Nowadays, as the indices of the major search engines grow to a tremendous proportion, vertical search services can help customers to find what they need. Search Engine is one of the reasons for Internet shopping success in today's world. The import one part of search engine is clustering data. The objective of this paper is to explore a k-means++ algorithm to calculate the clustering data which in the Internet shopping environment. The experiment results shows that the k-means++ algorithm is a faster algorithm to achieved a good clustering.

1. Introduction

Since the number of Internet shopping sites has also been increasing rapidly, it has become almost impossible to obtain information from the Internet shopping sites without using search engines. Users desire to define their preferences and customize the purchase information within the Internet shopping environment according to their individual needs. The result from general search engines is flat, and users have to go through all of results to find what they want. When users submit a query, the Internet shopping search will returns a list of results, ranked by its relevance. It is time-consuming to locate their interesting products with the low relevance. In most situations, they are not able to evaluate all available alternatives and typically follow a two-step model to fulfill their purchasing processes. In the first step, they identify a subset of the available alternatives by choosing from a vast range of products, and, in a second step, they perform relative comparisons among these to arrive at their final decisions [9]. Therefore, it is imperative to make users browse their interesting products easier; sorting the products over clustering is a good approach.

The performance of intelligent search engine is mainly evaluated by its accuracy and speed. Users are usually anxious to receive the accurate information, so the accuracy of system should be high. At the same time, data clustering algorithms promise to deliver efficient solutions to many of the problems arising from the interactions of consumers with the increasing volume of information in Internet shopping environments. On the other hand, the tolerable time of users waiting in front of the browser is generally limited, so the speed of the system responding to the users should be fast.

The remainder of this paper is organized as follows. In Section 2, we introduce related works. In Section 3, we present the k-means++ algorithms. Experimental studies and conclusions are given in Sections 4 and 5, respectively.

2. Related Work

The clustering algorithms have been proposed many years. First of all, the ISODATA algorithm [1] used the technique of merging and splitting clusters in order to obtain the optimal partition starting from any arbitrary initial partition, utilizing appropriate threshold values for performing this process. The dynamic clustering algorithm permitted other representations than the center of a cluster utilizing maximum-likelihood estimation, selecting a different criterion function [11]. Other research efforts improved computational complexity by reducing the number of (dis)similarity calculations [2]. Two very important steps in the evolution of the k-means algorithm family involve its extension to categorical and mixed numeric and categorical values [5]. The k-modes algorithm [2,3] extends the k-means paradigm to categorical domain by using a simple matching dissimilarity measure for categorical objects, modes in-stead of means for clusters, and a frequency-based method to update modes while the k-prototypes defines a combined dissimilarity measure, integrating the k-modes and k-means algorithms to allow for clustering of mixed numeric and categorical attributes. Based on k-modes algorithm, Ref. [4] proposes an adapted mixture model for data clustering, which gives a probabilistic interpretation of the criterion optimized by the k-modes algorithm. A fuzzy k-modes algorithm is presented in [5] and the tabu search technique is applied in [6] to improve fuzzy k-modes algorithm. However, most of the techniques used in the literature in data clustering are based on the hierarchical methodology, which are not efficient in the Internet shop-ping environment which include clustering large data sets.

3. Research Approach

In this section, Firstly we formally introduce the k-means++ algorithm [1]. It offers an efficient solution for users to model their preferences along multiple dimensions, search for product information, and then produce the data clusters of the products retrieved to enhance their purchase decisions.

3.1. k-means++ Algorithms

The k-means++ algorithm is inspired by k-means algorithms. The k-means algorithm is a non-hierarchical data clustering algorithm suitable for classifying large amounts of data into corresponding patterns. It is a simple and fast algorithm that attempts to locally improve an arbitrary k-means clustering. It works as follows.

1. Arbitrarily choose k initial centers$C = \{c_1, \dots, c_k\}$.
2. For each $i \in \{1, \dots, k\}$, set the cluster C_i to be the set of points in χ that are closer to c_i than they are to c_j for all $j \neq i$.
3. For each $i \in \{1, \dots, k\}$, set c_i to be the center of mass of all points in C_i: $c_i = \frac{1}{|C_i|} \sum x \in {C_i}^x$.
4. Repeat Steps 2 and 3 until C no longer changes.

It is standard practice to choose the initial centers uniformly at random from χ. For Step 2, ties may be broken arbitrarily, as long as the method is consistent.

Steps 2 and 3 are both guaranteed to decrease ϕ, so the algorithm makes local improvements to an arbitrary clustering until it is no longer possible to do so.

The k-means algorithm begins with an arbitrary set of cluster centers. So k-means++ propose a specific way of choosing these centers any given time, let D(x) denote the shortest distance from a data point x to the closest center we have already chosen. The k-means++ works as follows:

1a. Choose an initial center c_1 uniformly at random from χ.
1b. Choose the next center c_i, selecting $c_i \neq x' \in \chi$ with probability $\frac{D(x')^2}{\sum_{x \in \chi} D(x')^2}$.
1c. Repeat Step 1b until we have chosen a total of k centers.
2-4. Proceed as with the standard k-means algorithm.

In the Internet shopping environment users state their preferences by defining value inter-vals for each one of them. For example, price, service, delivery, product quality etc. For the sake of simplicity and visualization purposes we assume users state only two. In the first one their preference lies between the values x1 and x2 and in the second between the values y1 and y2. In this way, an iso-oriented rectangle is formed, named R, a rectangle with sides parallel to the axis. The products offered by Internet shopping environment are depicted as two-dimensional points with values p_i and q_i, and

$$p_i, q_i \in A = \{(p_1, q_1)(p_2, q_2), \dots, (p_n, q_n) | \text{ where } (p_i, q_i) \in \Re^2, i, n \in I\}$$

The k-means++ algorithm is suitable for large sets of numeric objects despite the fact that it is computationally expensive. It is sensitive to the selection of the initial partition and may converge to a local minimum of the criterion function value if the initial partition is not properly chosen. On the other hand, most variants of the k-means++ algorithms have been proven convergent while some variants like the ISODATA algorithm and k-means algorithm include a procedure that searches for the best k cluster means at the cost of some performance. More formally, let c(A) denote the center of mass of the data points in A, the k-means++ algorithm tries to minimize the squared error as it is described in function(1):

$$e^2 = \sum_{j=1}^{k} \frac{1}{|A|} \sum_{i=1}^{n_j} \left\| x_i^{(j)} - y_j \right\|^2 \qquad (1)$$

where $x_i^{(j)}$ is the *i*th pattern belonging to the *j*th cluster and y_j is the center of the *j* th cluster.

A number of variations to the k-means algorithm and k-means++ algorithm have been developed in an effort to improve its computational efficiency or extend its expressiveness in categorical or mixed data collaborative filtering. The k-means++ algorithm allow for clustering of mixed numeric and categorical attributes. This combined dissimilarity measure is described in function (2):

for each cluster C_j, where $1 \leq j \leq k$

$$e^2 = \sum_{j=1}^{k} \frac{1}{|C_j|} \sum_{i=1}^{n_j} \sum_{p \in V_1} (x_{i,p}^{(j)} - y_{j,p})^2 + \gamma \sum_{j=1}^{k} \sum_{i=1}^{n_j} \sum_{p \in V_2} \delta(x_{i,p}^{(j)}, \ y_{j,p}) \qquad (2)$$

where V_1 is the set of the numeric attributes and V_2 is the set of the categorical attributes. The weight γ is used to avoid favoring either type of attribute [7]. More specifically the dissimilarity measure is depicted in function (3) and is referred to as simple matching [11]:

$$\delta(x_i^{(j)}, \ y_j) = \begin{cases} 0 & (x_i^{(j)} = y_j) \\ 1 & (x_i^{(j)} \neq y_j) \end{cases} \qquad (3)$$

The computational cost of the k-means++ algorithm is $O(T \log kn)$, where T is the number of iterations, k is the number of clusters and n is the number of data items in the input data set. Although the run-time of both the k-means and the k-means++ algorithms appears to increase as the number of clusters and the number of data items increase, the k-means++ algorithm is much faster than k-means [11].

3.2. Multi-dimensional Range Search

A formal description of the multi-dimensional range search is:

Input: A set S of n data points in the *d*-dimensional space, so that

$$S = \{(s_1, s_2, \dots, s_n) \mid \text{where } s_i \in \Re^d, \text{and } i, d \in I\}$$

A *d*-dimensional rectangle R, defined by a set of two-dimensional points, each one representing a rectangle dimension,

$$R = \{(x_1, y_1)(x_2, y_2), \dots, (x_d, y_d) | \text{ where } (x_i, y_i) \in \Re \ \ \text{, and } i, d \in I\}$$

Output: All data points m lying inside the rectangle R.

3.3. k-means++ Range Algorithm

The proposed k-means++ range algorithm is a two-step process involving a multi-dimensional range search followed by a k-means++ clustering step in the case of numeric data points and categorical values.

Therefore a description of the proposed k-means++ range algorithm is as follows:

```
input n data points in the d-dimensional space, a d-dimensional rectangle R.
calculate all data points, be it $m$, lying inside the rectangle using a d-dimensional range tree search.
 input $k$ (number of the cluster means)
initialize center $c_1, c_2, \ldots c_k$
repeat
   do
     choose an initial center $c_1$ uniformly at random from $\chi$.
     choose the next center $c_i$, selecting $c_i \neq x' \in \chi$ with probability $\frac{D(x')^d}{\sum x \in \chi D(x')^d}$.
     until have a total of k center $y_1, y_2, \ldots y_k$
repeat
   for each input data point $x_{i,}$ $1 \leq i \leq m$
      do
         choose $x_i$ to the jth cluster with nearest mean $y_j$, such that the quantity $(x_{i,p}^{(j)} - y_{j,p})^2$ or $\gamma\delta(x_{i,p}^{(j)},\ y_{j,p})$ , depending on the nature of the attribute, is minimum for all j, where $1 \leq j \leq k$
   for each cluster $C_j$, where $1 \leq j \leq k$
         do recalculate the clustering accuracy $r = \frac{1}{|C_j|} \sum_{y_j \in C_j} y_j$
   calculate the function
$$\sum_{j=1}^{k} \frac{1}{|C_j|} \sum_{i=1}^{n_j} \sum_{p \in V_1} (x_{i,p}^{(j)} - y_{j,p})^2 + \gamma \sum_{j=1}^{k} \sum_{i=1}^{n_j} \sum_{p \in V_2} \delta(x_{i,p}^{(j)},\ y_{j,p})$$
   until no data point has changed clusters.
```

4. Experiment Result

Until now, there is no well-recognized standard methodology for data clustering experiments. In order to evaluate the k-means++ algorithm in practice, we tested them in C++ to apply both k-means++ and k-means in six datasets. We run the experiments on a Celeron 433MHz machine running Linux and memory size is 64 Mbytes. The gcc version is gcc-4.2.3.

The datasets are retrieved from the Internet shopping site taobao.com where each dataset is recorded with six attributes (userId, price, sales, rank of shopping environment, rank of shopping service, product quality). We use the taobao's [12] traditional search engine to get 6 products' retrieved information which consist 500 products, 1000 products, 1500 products, 2000 products, 2500 products, 3000 products. For each dataset we compute the average execution time of the k-means algorithm and k-means++ algorithm.

The performance of the experiment is presented in Fig. 1. The results show that our algorithm improves the performance of the k-means algorithm in the Internet shopping environment.

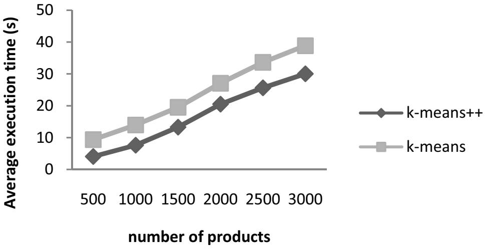

Fig. 1 Scalability comparison of different clustering algorithms

5. Conclusion and Future Work

The main goal of this paper was the development of a new way for personalized clustering in Internet shopping environment. It allows users to assign their preferences by defining value intervals for each one of products and sort the value over clustering. Furthermore, the experiment results demonstrated that our algorithm can improve improves the performance of the k-means algorithm in the Internet shopping environment.

This algorithm is not achieved in Cloud Computing environment. Future work directions involve the development of the application k-means++ algorithm over MapReduce[10] operators.

References

[1] G.H. Ball, D.J. Hall, A clustering technique for summarizing multivariate data, Behavioral Science 12 (1967) 153–155.

[2] Z. Huang, A fast clustering algorithm to cluster very large categorical data sets in data mining, in: Proc. of 1997 SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery, 1997, pp. 1–8.

[3] Z. Huang, Extensions to the k-means algorithm for clustering large data sets with categor-ical values, Data Mining and Knowledge Discovery 2 (3) (1998) 283–304.

[4] F. Jollois, M. Nadif, Clustering large categorical data, in: Proc. Of PAKDD'02, 2002, pp. 257–263.

[5] Z. Huang, M.K. Ng, A fuzzy k-modes algorithm for clustering categorical data, IEEE Transactions on Fuzzy Systems 7 (4) (1999) 446–452.

[6] M.K. Ng, J.C. Wong, Clustering categorical data sets using tabu search techniques, Pattern Recognition 35 (12) (2002) 2783–2790.

[7] L. Kaufman, P.J. Rousseeuw, Finding Groups in Data—An Introduction to Cluster Anal-ysis, Wiley, 1990.

[8] David Arthur, Sergei Vassilvitskii K-means^{++}: The Advantages of Careful Seeding, In Proceedings of SODA'2007

[9] D. Willard, New data structures for orthogonal range queries, SIAM Journal on Compu-ting 14 (1985) 232–253

[10] J. Dean and S. Ghemawat. Mapreduce: Simplified data processing on large clusters. In OSDI'04: Sixth Symposium on Operating System Design and Implementation, December 2004.

[11] D. Patterson and J. Henessy. Computer Architecture. A Quantitative Approach. Morgan Kaufmann Publishers, fourth edition, 2006.

[12] www.taobao.com