MODIFIED BIRCH CLUSTERING FOR REAL TIME CLUSTERING TECHNIQUE FOR RECOMMENDER SYSTEMS BASED ON COLLABORATIVE FILTERING

Ratchainant Thammasudjarit and Phayung Meesad***

*Department of Information Technology, Faculty of Information Technology
**Department of Teacher Training in Electrical Engineering, Faculty of Technical Education
King Mongkut's University of Technology North Bangkok
*rucci-rucci@hotmail.com, **pym@kmutnb.ac.th

ABSTRACT

Recommender Systems are one of the core components in e-commerce businesses to find personalized recommendations for products or services for interactive users. Collaborative Filtering is one of the successful techniques that apply to recommender systems for predicting lists of recommendations for interactive users by finding a neighborhood of users that has similar tastes as active users for recommendation generation. However, drawbacks of collaborative filtering are dealing with missing value and incremental data. Missing value not only effects to prediction accuracy, but also data clustering. Incremental data effects to response time of collaborative filtering. (e.g., incremental of number of user in e-commerce website) The system with slow response time will lost user's attention from e-commerce website and eventually lost business opportunities. Data clustering is applied to improve response time for incremental of user with acceptable prediction accuracy. However, incremental of user is not static while data clustering works with static data. Re-clustering can be applied, but it is related to cost of the system and limited of ability for real time clustering.

In this paper, we propose a new data clustering technique by modifying existed clustering algorithm, Balanced Iterative Reducing and Clustering Hierarchies (BIRCH), using a recursive indexing approach for handling incremental data and real time clustering then applying it to a movie recommender system based on collaborative filtering. The experiment have been done on the MovieLens dataset and represents comparison of prediction accuracy and response time between traditional collaborative filtering, collaborative filtering with modified BIRCH and collaborative filtering with modified BIRCH plus KNN-imputation missing value. Results of experiments show that the most effective algorithm is collaborative filtering with clustering in terms of optimizing both accuracy and response time.

Index Terms— Recommender Systems, e-Commence, Collaborative Filtering, BRICH Clustering, KNN Imputation

1. INTRODUCTION

The Rising of e-commerce has brought a new trading channel, which is independent in place and time for businesses. Each customer seeks preferred products and services from an enormous number of them. Recommender Systems aid users in dealing with information overload and provide personalized recommendations. Collaborative filtering is one of the successful techniques for finding users' preferred products, even if they have no experience with those products, by forming a users' neighborhood for predicting the desires of new customers. However, a drawback of neighborhood formation, that seeks other users who have similar tastes as active users and experience in those products from all users in the system, is that it is computationally costly. Moreover, the numbers of customers increasing in the system directly reflect the dimension of data that take longer response time. Today, recommender systems are not only requiring prediction accuracy but also acceptable response times for keeping users' attention on the e-commerce website.

In this paper, we propose a new data clustering technique by modifying existed clustering algorithm, Balanced Iterative Reducing and Clustering Hierarchies (BIRCH), using a recursive indexing approach for handling incremental data and real time clustering then applying it to a movie recommender system based on collaborative filtering. The experiment is based on the MovieLens dataset using a 10-fold cross validation to compare prediction accuracy and throughput of three algorithms:

1) *Traditional Collaborative Filtering (TCF)* referring to the approach of co-rated users for each predicted item to form a neighborhood

2) *Collaborative Filtering with Clustering (CFC)* adding data dimension reduction by using a clustering technique to collaborative filtering.

3) *Collaborative Filtering with KNN Imputation Missing Value and Clustering (CFIC)* increasing data density using KNN imputation missing value approach for clustering data and collaborative filtering.

2. RELATED WORK

In this section, we briefly discuss the existing state-of-the-art literature on recommender systems, collaborative filtering, and BIRCH clustering.

2.1 Recommender Systems

Recommender Systems can be classified into three groups. [6] (1) Collaborative filtering [2, 4, 5] is one of the most frequently used that works on k-nearest-neighbor method applied on users-items rating matrix. Collaborative filtering represents the idea of word-of-mouth promotion from users that have experiences with those products and have similar tastes to the purchaser. (2) Content Filtering has a different recommendation algorithm. Prediction of content filtering is based on explicit interests of users. It cannot have serendipitous recommendations like collaborative filtering. However, both content filtering and collaborative filtering don't provide deep knowledge of product domain. Both of them are excellent when applied to simple products such as books, movies, or music. In order to work with complex products such as computers, digital cameras, or financial services, customers need a more intelligent interaction mechanism and information from a recommender system for making decisions [1] such as (iii) knowledge-based filtering.

2.2 Collaborative Filtering

In the collaborative filtering technique, criteria of similarity measurement make different results in neighborhood formation and affect the result of prediction. The correlation-based technique is frequently applied in collaborative filtering. It uses similarity measurement such as Pearson's correlation and cosine similarity for finding neighborhoods of like-minded users for each interactive user then predicts a rating for un-rated products from the weight average of the neighborhood and finally recommends top-n products from top-n predicted ratings to users. However, it is computationally expensive to find similarity of pair users during the training phase but it can be reduced by using a clustering technique. [3, 7]

2.3 Incremental Data Clustering

Balanced Iterative Reducing and Clustering Hierarchies (BIRCH) are designed for very large databases by incrementally and dynamically clustering incoming multi-dimensional metric data points to produce clusters. [8] In this paper, the perception of BIRCH in scalability and threshold setting brought idea to create a new cluster algorithm using the recursive indexing approach. The cluster algorithm works on Euclidean's distance base and able to perform real time clustering for real time incremental data.

3. THE PROPOSED ALGORITHM

There are three phrase of collaborative filtering, in the 1st phrase – Data Transformation, data from e-commerce websites is usually represented as transactional data which consists of user ID, product ID, purchase quantity, etc. The Data transformation phase will transform transactional data to a users-items matrix and give a rating which represents the users' preferred inside the matrix. The rating might be calculated from the frequency of purchasing or the rating point that was rated by users as shown in figure 1.

Transaction Data

User ID	Product ID	Purchase Qty
001	01	3
001	02	2
001	03	5
002	02	3
002	06	1
002	03	3
003	02	4
003	04	2
003	05	3
004	05	5
004	02	2
004	01	4
004	03	3

Transform

User-Items Rating Matrix

		Product ID					
		01	02	03	04	05	06
User ID	001	3	2	5			
	002		3	3			1
	003		4		2	3	
	004		2	3	4		5

Note: □ represents non-rated data or missing value.

Figure 1 Data Transformation

In the 2nd phase, each active user needs to have their neighborhood for prediction by finding K-nearest neighbor based on similarity measurement. In this case, the similarity between two users, i and j, is measured by using the Pearson-r correlation from co-rated items of both user i and j. The prediction of item i for active user u_a from neighborhood (u) refer to the weight average prediction as in (1).

$$P_{u_a,i} = \bar{r}_{u_a} + \frac{\sum_{i=1}^{k} sim_{u_a,u_i}(r_{u_i,i} - \bar{r}_{u_i})}{\sum_{U=1}^{k} sim_{u_a,u_i}} \quad (1)$$

where $P_{u_a,i}$ is predict rating of active user (u_a) on item i, $\bar{r}_{u_a}$ is average rating given by active user (u_a), sim_{u_a,u_i} is similarity measurement between users, $r_{u_i,i}$ is rating given by i^th user (u_i) on item i, and $\bar{r}_{u_i}$ is average rating given by i^th user (u_i)

In the 3rd phase, all predicted ratings calculated from equation (1) of each active user on each item are sorted in top-n recommendations by listing the highest n ratings recommended for the active user, in this case is user ID 001, as shown in figure 3.

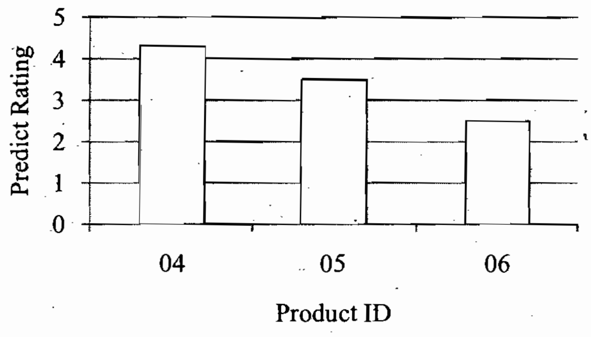

Figure 3 Top-N Recommendations

KNN Imputation Missing Value - missing values of the instance are imputed based on a given number of k-instances that are most similar to the instance of interest. Similarity measurement uses a distance function such as Euclidean. KNN imputation can predict in both qualitative and quantitative attributes. In the case of the qualitative, imputation is based on the most frequent value among k-nearest neighbors while quantitative imputation is based on mean among k-nearest neighbors. More data density is better for clustering. In this paper, we apply KNN imputation missing value with clustering to gain more data density. Imputed rating is calculated referring to weight average prediction as in (2):

$$\hat{R}_{(i,j)} = \frac{\sum_{i=1}^{k} w_{a,i} R_{i,j}}{\sum_{i=1}^{k} w_{a,i}} \quad (2)$$

where $\hat{R}_{(i,j)}$ is imputed rating for missing value of j^{th} item of i^{th} instance , $R_{i,j}$ is i^{th} instance's rating on j^{th} item, and $w_{a,i}$ is the inverse of distance between instance i and active instance (a).

Balanced Iterative Reducing and Clustering Hierarchies (BIRCH) work on the clustering feature (*CF*) and the *CF* tree. Clustering Features can be represented by *(N, LS, SS)*, where *N* is the number of data points in the cluster, *LS* is the linear sum of data points, and *SS* is the sum of squares of data points. Additive theorem of CF can be defined by assuming CF1 consists of *(N_1, LS_1, SS_1)* and *CF2* consists of *(N_2, LS_2, SS_2)* then CF_{total} can be determined as following equation (3):

$$CF_{total} = N_1 + N_2 + LS_1 + LS_2 + SS_1 + SS_2 \quad (3)$$

To make clustering decisions for each entry instance, we need to construct a *CF* tree. There are two parameters, threshold (*T*) and branch factor (*B*), to define the shape of the *CF* tree. Threshold (*T*) represents the maximum size of a cluster. Branch factor (*B*) represents the maximum number of branches for each node. Insertion of entry instance into the *CF* tree starts from the root node then traverses to the nearest leaf node. The nearest leaf node is the nearest cluster for entry instance. The insertion algorithm of entry instance considers threshold and branch factors. Entry instance will be inserted into the cluster if the cluster size is not exceeding the threshold. (Size ≤ *T*) In case the cluster size exceeds the threshold, the leaf node will be split and then entry instance can be inserted into the nearest node. Maximum splitting in each node will not exceed the branch factor, otherwise the leaf node will be split in the next level of the *CF* tree and it will become non-leaf node.

Figure 4 shows splitting of node in case of node size is greater than the threshold and the number of branches is less than the branch factor. The node will be split in the same layer.

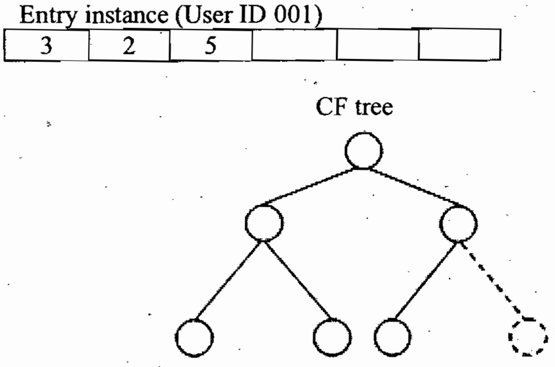

Figure 4 In layer splitting condition

Figure 5 shows splitting of node in case the node size is greater than the threshold and the number oT branches is equal to the branch factor, the node will be split to the next layer.

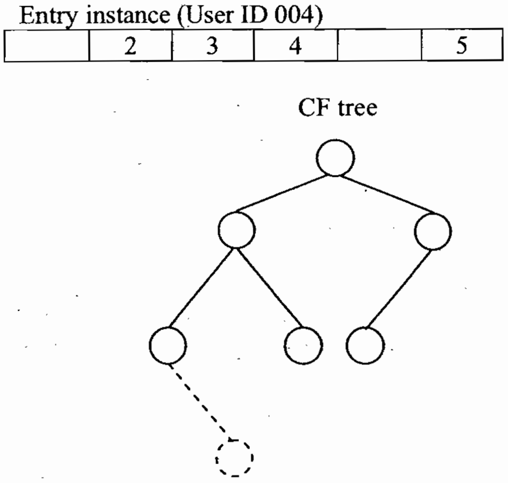

Figure 5 Cross layer splitting condition

The modified BIRCH with recursive indexing approach uses the idea of threshold (*T*) setting of cluster size from BIRCH then applies the bi-clustering to form clusters. The nth cluster is split into two new clusters where new clusters index are $2n$+1 and $2n$+2 respectively. This concept is similar to an amoeba's lifecycle when it grows up and divides itself. For cluster formation of training data, we consider all training data are member of one cluster as shown in figure 6.

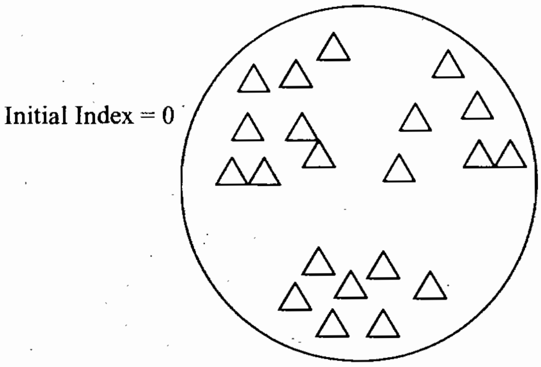

Figure 6 Initial of cluster at first iteration

Current size of all clusters is the key for activating recursive indexing algorithm. In case of any cluster size that greater than threshold, There is any cluster that its size exceed threshold, clustering formation keeps recursive splitting checking until the size of all clusters is less than the threshold. For example, size of cluster in figure 6, assume that index of cluster is 0 for initialize cluster, is greater than threshold. Recursive indexing will split the cluster into new two clusters and the new indexes are 1 and 2 respectively refer to $2n$+1 and $2n$+2 as shown in figure 7.

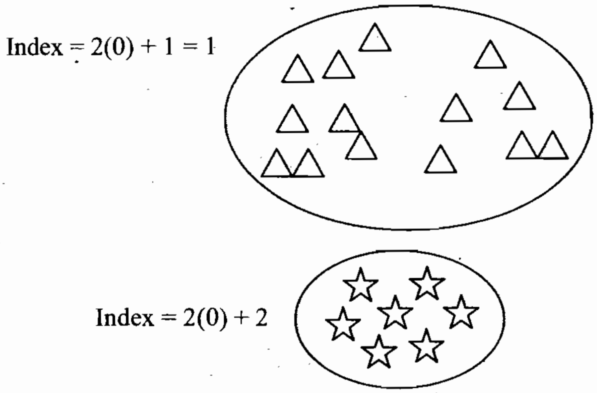

Figure 7 Cluster splitting in 2nd iteration

This algorithm recursive check cluster size and split exceed cluster size until the size of all clusters is less than the threshold. Recursive indexing approach works well with real time incremental data. Consider new entry data, its closest cluster in this case is cluster 3 as shown in figure 8.

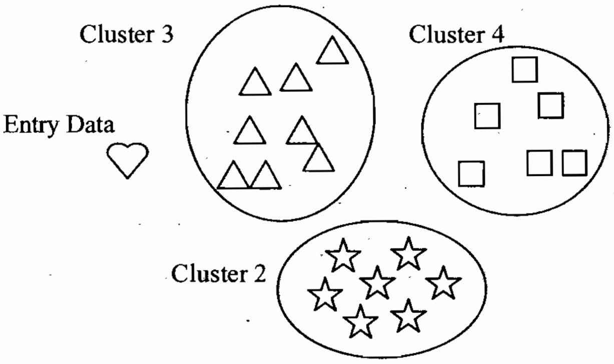

Figure 8 Entry data from incremental data

Insertion entry data into its closest cluster needs to check whether the new cluster size exceeds threshold. Recursive indexing algorithm will immediately split if the size of merging entry data in to its closest cluster exceed threshold. In this case, we merge entry data with cluster 3 and its size is greater than threshold. So, Cluster 3 will be split to be cluster 7 and cluster 8 as shown in figure 9.

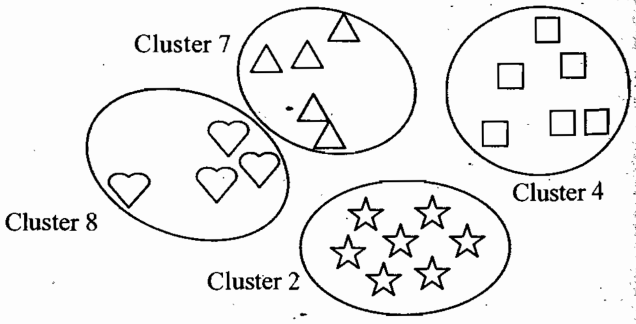

Figure 9 Real time update clustering for entry data

The threshold setting at larger values will reduce iteration of the recursive thereby getting faster cluster formation. In this paper, modified BIRCH with a recursive indexing approach is based on Euclidean distance vector. The advantage of this approach is that BIRCH is non-traversing to the nearest node and updating previous node is not required. The key feature of this clustering algorithm is real time update and it will be useful for future work of multi-layer of collaborative filtering for recommender system of complex products as shown in figure 10.

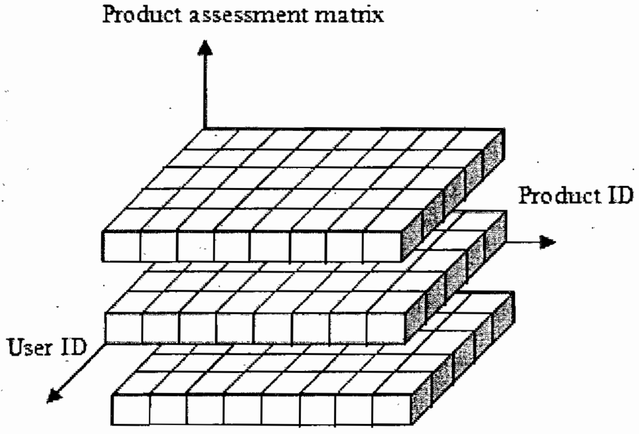

Figure 10 Multi-layer of collaborative filtering

The disadvantage of recursive indexing approach is the threshold setting. If the threshold is set too low, recursive iteration will be increased. However, the complexity of the next iteration will not be greater than the current iteration due to the reduction in splitting in the dimension of the data.

4. EXPERIMENTS AND DISCUSSION

To prove the concept of the proposed method, several experimental simulations have been performed. The data set used to perform these experiments was obtained from the MovieLens project website (http://movielens.umn.edu). The data set consists of 100,000 movie ratings with a one to five rating scale from 943 users on 1682 movies. The experiment in each algorithm has been validated with ten-fold cross validation. Final results are the average result of those ten sets.

Performance metrics for this experiment are prediction accuracy measured in Mean Absolute Error (MAE) and speed of recommendation measured in throughput of users per second. Each of these three approaches is in this experiment:

1) *Traditional Collaborative Filtering (TCF)* referring to the approach of co-rated users for each predicted item to form a neighborhood

2) *Collaborative Filtering with Clustering (CFC)* adding data dimension reduction by using a clustering technique to collaborative filtering.

3) *Collaborative Filtering with KNN Imputation Missing Value and Clustering (CFIC)* increasing data density using KNN imputation missing value approach for clustering data and collaborative filtering.

Performance summary of each algorithm based on 80% of data for training set and 20% of data for test set shows in table 1. CFC performs best in term of optimization between prediction accuracy and response time.

Algorithm	TCF	CFC	CFI
Average MAE	0.7631	0.8548	0.8338
Average response time (sec/instance)	1.3046	0.3026	0.4902
Average imputation lead time (sec)	NA	NA	311

Table 1 Performance summary

From table 1, Traditional Collaborative Filtering (*TCF*) performs the best *MAE* with co-rated users on predicting items due to it takes the less effect from noise of data in neighborhood formation. With clustering, *CFC* performs the worst *MAE*. The gap of difference is about 0.1 compared to the best performance from *TCF*. However, gaining more data density before clustering can improve prediction accuracy that represents the CFIC shows MAE is improved about 0.02 from CFC.

The average MAE performance of each algorithm in the neighborhood size 30 is shown in figure 11.

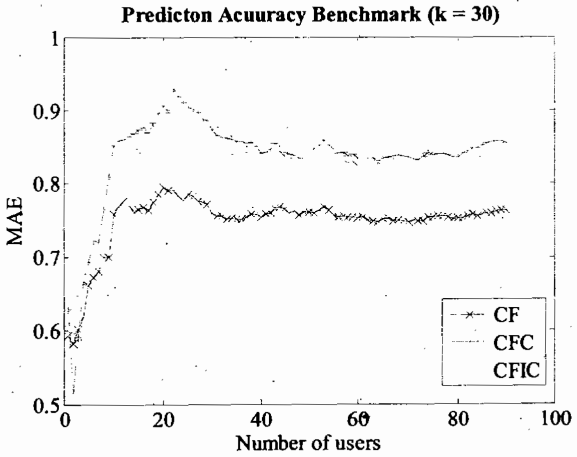

Figure 11 Prediction Accuracy Benchmark

Considering the throughput performance of all algorithms, clustering improves significantly the throughput performance. CFC performs best throughput performance and reduces the response time about 76% of CF. CFIC performs not much difference response time per users as CFC. However, it needs imputation lead time so that prediction can be started. For a long term recommendation, CFIC is better than CF in terms of throughput performance and give higher accuracy than CFC. An average throughput performance is shown in figure 12.

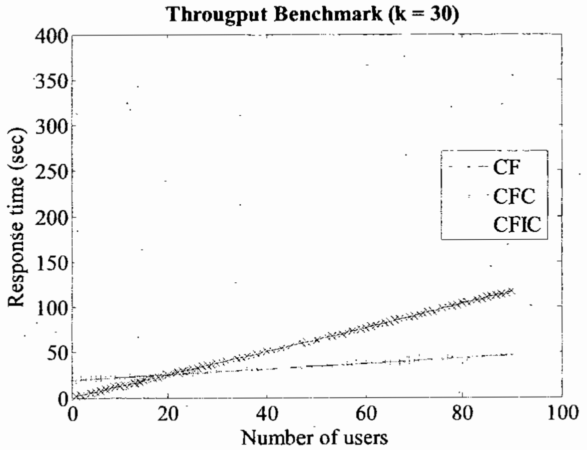

Figure 12 Throughput Benchmark

5. CONCLUSIONS

Recommender systems based on traditional collaborative filtering suffers from an incremental increase of users in e-commerce websites that directly affect the response time and scalability. Data clustering techniques are an effective approach for improving scalability of collaborative filtering. Whether the implementation of collaborative filtering with clustering technique for e-commerce websites can be done depends on the number of users and their transactions in the system. In the beginning of the implementation, the system is not overloaded from the number of users and their transactions. Prediction accuracy from collaborative filtering with clustering technique is acceptable. However, when the system is overload from incremental of number of users. Prediction accuracy from previous is getting drop. Re-clustering can help to get back prediction accuracy performance but it is the system cost. Real time clustering with recursive indexing approach can make thing different. Whether the system takes effect from incremental of user, modified BIRCH with recursive indexing approach can give real time update cluster information for collaborative filtering and will be useful for complex products recommender based on multi-layer collaborative filtering which offline clustering cannot work with.

6. REFERENCES

[1] Felfernig, A., *et al.* "Recommender Systems," *IEEE Intelligent Systems Special Issue on Recommender Systems May/June 2007*, IEEE Computer Society.

[2] Chen, A. Y., *et al.* "Collaborative Filtering for Information Recommendation Systems," *Encyclopedia of Data Warehousing and Mining*, Idea Group, 2005.

[3] Sarwar, B. M., *et al.* "Recommender Systems for Large-Scale E-Commerce: Scalable Neighborhood Formation Using Clustering," *Proceedings of the 5th International Conference on Computer and Information Technology (ICCIT).*

[4] Sarwar, B. *et al.* "Analysis of Recommendation Algorithms for E-commerce," *Proceedings of the 2nd ACM Conference on Electronic Commerce. ACM.* vii+271, 158-67.

[5] Sarwar, B. M., *et al.* "Item-Based Collaborative Filtering Recommendation Algorithms," *Proceedings of the 10th International World Wide Web Conference*, 285-295.

[6] Adomavicius, G., *et al.* "Recommendation Technologies: Survey of Current Methods and Possible Extensions. *Appears in Collections of Information Systems Working Papers. Stern School of Business, New York University. Report No. IS-03-06.*

[7] Haruechaiyasak, C. *et al.* "A Dynamic Framework for Maintaining Customer Profiles in E-Commerce Recommender Systems," *Proceedings of the 2005 IEEE International Conference*, pp. 768-771.

[8] Zhang, T., "BIRCH: An Efficient Data Clustering Method for Very Large Databases," *in SIGMOD'96: Proc. the 1996 ACM SIGMOD Int. Conf. Management of Data*, Montreal, Quebec, Canada, 1996, pp.103-114.