A Dynamic Markov Biclustering Cache Replacement Policy for Mobile Environment

Hariram Chavan[1]
[1]Information Technology
Terna Engineering College
Mumbai University, India

Suneeta Sane[2]
[2]Computer Technology
V. J. T. I.
Mumbai, India

H. B. Kekre[3]
[3]Mukesh Patel School of Technology
Management & Engineering
NMiMS University, Mumbai, India

ABSTRACT
In mobile database systems caching proved itself as an important technique to optimize the way a mobile database is used. The desired caching can be achieved by convincingly accurate prediction of data items for the present and future query processing. Prefetching is a commonly used strategy to cut down network resources consumed as well as the access latencies observed by end users. In this paper, we propose a Dynamic Markov Biclustering Cache Replacement Policy (DMBCRP) which is a sophisticated combination of caching and prefetching for mobile database environment. We dynamically bicluster the data for location based services with second and/or first order Markov Model to predict the new data item(s) to be fetched based on user access patterns. The java implementation of DMBCRP, using trip data set and dynamic location specific resource biclustering results in different user access patterns and also user movement patterns.

Keywords
Markov model, Biclustering, caching, prefetching, users access patterns.

1. INTRODUCTION
The fast development of wireless communications systems and advancement in computer hardware technology has led to the seamlessly converged research area called mobile computing. The mobile computing research area includes the effect of mobility on system, users, data and computing. The seamless mobility has opened up new classes of applications and offers access to information anywhere and anytime.

The main reason for the growing popularity of the mobile computing is new kind of information services, called Location Dependent Information Services (LDIS). LDIS provides information to users depending upon his/her current location.

There are two common issues involved in client cache management: cache invalidation and cache replacement. A cache invalidation policy maintains data consistency between client and server. However, a cache replacement policy finds suitable subset of data item(s) for eviction from cache when it does not have enough free space to store a new data item. Due to the limitations of cache size on mobile devices, an efficient cache replacement policy is vital to ensure good cache performance. Thus a cache replacement policy becomes unavoidable in order to utilize the available cache effectively.

Considering several cache replacement policies have been proposed in the past, the real challenge is to extend them by using data mining techniques. Clustering is an unsupervised data mining technique used to place data elements into subsets (called *clusters*). The basis for clustering is often similarity between data items. In recent years, more and more attention has been paid to finding biclusters. The term biclustering has been used by Mirkin (1996) to describe "simultaneous clustering of both row and column sets in a data matrix". For our policy a bicluster is a subset of rows (locations) and a subset of columns (resources) indicates a sub-matrix which can be viewed as a local coherent access pattern for users.

The structure of the paper is as follows: section 2 describes Related Work, section 3 Motivation: Markov Biclustering, section 4 details proposed cache replacement policy. Section 5 details performance evaluation and comparisons. Section 6 concludes the paper.

2. RELATED WORK
Caching and prefetching is not at all a new concept. In the location-dependent system and mobile environments, where the client is near an infostation the information related to its area is hoarded and the table of resource data matrix (RDM) is maintained. The Prioritized Predicted Region based Cache Replacement Policy (PPRRP) [2] predict valid scope area for client's current position, assigns priority to the cached data items, calculates the cost on the basis of access probability, data size in cache and data distance for predicated region. The limitation of PPRRP policy is multiple replacements because of data size.

The Furthest Away Replacement (FAR) [5] depends on the current location and the direction of the client movement. The selection of victim for replacement is based on the current location of user. The limitation of FAR is it won't consider the access patterns of client and is not very useful for random movement of client. LRU-K [7] method is an approach to database disk buffering. The LRU-K takes into account only temporal characteristics of data access.

The dominating property for Location Dependent Data (LDD) [12] is the spatial nature of data. None of these existing cache replacement policies are very useful if client changes its direction of movement quite often. It is necessary to anticipate the future location of the client, based on current location, in the valid scope using mathematically proven hypothesis. So, in this paper, we propose DMBCRP based on Markov Model and biclustering. The key idea of DMBCRP is prediction of future user location and performance optimization by state pruning strategy of biclustering . The basic cellular mobile network for a wireless communication used is similar as discussed by Vijay Kumar et al [8].

An efficient node-deletion algorithm is introduced by Yizong Cheng and George M. Church [17] to find submatrices in expression data. Beyond the traditional clustering method, the term biclustering was first applied to expression data for simultaneous clustering of genes and conditions with high similarity score (mean squared residue). Discretization and missing data replacement by random number strategy may limit the ability of the algorithm to discover biologically relevant patterns.

A good idea to reduce the processing time, reduction in dimension and size related to data mining problem is feature ranking. A smaller group of representative features, retaining the most prominent characteristics of the data leads to more compact model and better generalization, which is achieved by Qinghua Huang, Lianwen Jin and Dacheng Tao [16]. In DMBCRP we have ranked features to find most relevant data for user query as valid scope, access probability, distance and data size.

3. MOTIVATION: MARKOV BICLUSTERING

The main aim of prefetching is to reduce latency, optimal use of cache space and best possible utilization of available bandwidth. But when we enhance prefetching policy it may prefetch data items which may not be eventually requested by the users and result into server load and network traffic. To overcome these limitations we can use high accuracy prediction model such as Markov Model. The biclustering approach overcomes some problems associated with traditional clustering methods, by allowing automatic discovery of similarity based on a subset of attributes.

Specifically, in mobile database caching and prefetching can complement each other. The caching utilizes the temporal locality which refers to repeated users accesses to the same object within short time periods. The prefetching utilizes the spatial locality of the data items. So in this paper we present an integrated approach of effective caching and prefetching by convincingly accurate prediction of client location by Markov model and reduction in data space by dynamic biclustering of resource data for the present and future query processing.

4. CACHE REPLACEMENT POLICY

LDIS is spatial in nature. A data item can show different values when it is queried by clients from different locations. For cache replacement we should account the distance of data from clients' current position as well as its valid scope area. Larger the distance of data item from the clients' current position the probability is low that client will enter into the valid scope area in near future. Thus it is better to replace the farthest data value when cache replacement takes place. Generally client movement is random so it is not always necessary that client will continue to move in the same direction. Therefore replacing data values which are in the opposite direction of client movement but close to clients' current position may degrade the overall performance. In DMBCRP we are considering previous few locations for the predication of next client location so there is very less probability that the data item with higher access probability related to previous location will get replaced by new data item.
The system set up for this paper deals with following assumptions: Let the space under consideration be divided in physical subspaces (locations) as

$S = \{L_1, L_2, \ldots, L_N\}$

There are data items with

$D = \{D_1, D_2, \ldots, D_M\}$

Such that each data item is associated with valid scope which is either a set of one or more subspaces from S. Formally shown as

$D_k = \{L_{k1}, L_{k2}\}$ where L_{k1}, L_{k2} belong to S.

Now consider the problem of predicting the next location of client visit. The trip data set is used for building Markov models. The actions for the Markov model correspond to the different locations visited by the client, and the states correspond to all consecutive trips. In case of first-order model, the states will correspond to single locations and that of second-order will correspond to all pairs of consecutive locations and so on. To illustrate, we are assuming the predicted region routes between the five different locations of geographical area with locations L_1, L_2, L_3, L_4 and L_5.

Connections on the move for a trip are very large. However if only the cell address where the calls were made limits to the chances of intermittent contact during the travel. To counter this it is proposed that cells that contribute to the deflection from the previous path be noted and continued. This will cause significant reduction in state space search, thus reducing the complexity of the algorithm used as well as keeping the needed details in the data. In this paper we consider such pruned data for further analysis.

Once the states of the Markov model have been identified, the transition probability matrix (TPM) can be computed. There are many ways in which the TPM can be built. The most commonly used approach is to use a *training* set of action-sequences and estimate each t_{ji} entry based on the frequency of the event that action a_i follows the state s_j. For example consider the second trip of customer $TR_2(\{L_3, L_5, L_2, L_1, L_4\})$ shown in Figure 1.a. If we are using *first-order Markov model* then each state is made up of a single location, so the first location L_3 corresponds to the

state s_3. Since state s_5 follows the state s_3 the entry t_{35} in the TPM will be updated (Figure 1. b). Similarly, the next state will be s_5 and the entry t_{52} will be updated in the TPM. In the case of higher-order model each state will be made up of more than one action.

Trips	
Trip No	Locations
TR_1	$\{L_3,L_2,L_1\}$
TR_2	$\{L_3,L_5,L_2,L_1,L_4\}$
TR_3	$\{L_4,L_5,L_2,L_1,L_5,L_4\}$
TR_4	$\{L_3,L_4,L_5,L_2,L_1\}$
TR_5	$\{L_1,L_4,L_2,L_5,L_4\}$

Fig 1: a) Trips

1^{st} Order	L_1	L_2	L_3	L_4	L_5
$s_1=\{L_1\}$	0	0	0	2	1
$s_2=\{L_2\}$	4	0	0	0	1
$s_3=\{L_3\}$	0	1	0	1	1
$s_4=\{L_4\}$	0	1	0	0	2
$s_5=\{L_5\}$	0	3	0	2	0

Fig 1: b) First Order TPM

Fig 1: Sample location Trips and 1^{st} order TPM.

Once the TPM is built, making prediction for different trips is straight forward. For example, consider a client that has visited locations L_1, L_5, L_4. If we want to predict the location that will be visited by the client next, using a first-order model, we will first identify the state s_4 that is associated with location L_4 and look up the TPM to find the location L_i that has the highest probability and use it. In the case of our example the prediction would be location L_5. The fundamental assumption for prediction based on Markov models is that the next state is dependent on the previous *k* states. The longer the *k* is, the more accurate the predictions are.

The MDS is developed with the following data attributes - valid scope, access probability, data size and distance related to respective location. This centralized database is percolated to MSS in order to make sure that the data relevant in the valid scope is available in the nearest MSS thus making sure that the concept of Prioritized Region is taken into account while considering the cache replacement policy. The feature ranking is in order of valid scope, access probability, distance and data size. The access probability is updated in database with each access to the cached data item.

When cache has no enough space to store queried data item then space is created by replacing existing data items from cache based on minimum access probability (Pa_i), maximum distance (ds_i) and scope invalidation (vs_i). If data items have same valid scope then replacement decision is based on minimum access probability. If valid scope and access probability is same then replacement decision is based on maximum data distance. In some cases, data size plays an important role in replacement. If fetched data item size is large enough and requires replacing more than three data items from cache then replacement is based on maximum equivalent size with minimum access probability, maximum distance and scope invalidation. Figure 2. shows the representation of valid scope and Table 1 gives data instance for the same. The advantage is the complete knowledge of the valid scopes.

Table 1. Valid Scope for Hospital Data Instance

Data Distance	Valid Scope
(nearby Hosp, {A,B})	{1,2}
(nearby Hosp, {C})	{3,4}

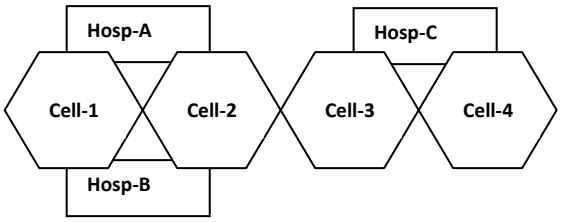

Fig 2: Valid scope representation for hospital.

The resources data accessed by client during different trips are represented in matrix. The matrix rows represent trips (different locations visited) and columns represent the different resources accessed by clients during trip. Each cell in the matrix represents the resources available and availed by client under a specific experimental condition. It is good, if we get a relevant group of resource for a subset of conditions. On the other hand, groups of conditions can be clustered by using different groups of clients (trips). In this case, it is important to do clustering on these two dimensions simultaneously. This led to the discovery of biclusters corresponding to a subset of trips and a subset of resources requested by a group of clients with a high similarity score by Cheng and Church [17]. The problem of biclustering consists of the simultaneous clustering of rows and columns of a matrix such that each of the submatrices induced by a pair of row and column clusters is as uniform as possible. Finding an optimal solution for the biclustering problem is NP-hard [19].

Example: consider two randomly selected trips and the resources accessed during trips within same set of location (i.e from location L_1 to L_5) as shown in figure 3: TR_1 = {4,7,3,1,6} and TR_2 = {4,6,3,7,1}. These sample trips and resources data

indicates that while visiting location from L_1 to L_5 or vice versa most of the users access resources {1,3,4,6,7} but in different order so we can make a bicluster of trips (actually locations) and resources accessed.

L_1	L_2	L_3	L_4	L_5
4	7	3	1	6
4	6	3	7	1
2	7	4	3	4
2	7	1	6	5
3	4	7	3	4
2	1	6	1	5
3	4	1	6	5
4	3	7	3	4
3	5	6	1	3
3	5	4	1	3

Fig 3: A snapshot of locations resources accessed by users during different trips.

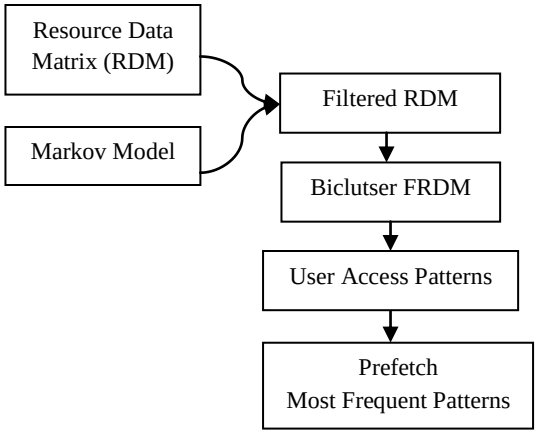

Fig 4: Proposed DMBCRP algorithm.

DMBCRP cache replacement policy works as follows:

- All customer data.
- Customer specific data.
- Pruned data for customer specific trips.
- Predict the next location of client by Markov Model.
- Dynamically bicluster resource matrix based on previous, current and next predicate location of client.
- Apply
    1. The valid scope of the data item.
    2. Access probability of the data item.
    3. Distance of the data item.
    4. Data size of data item.
- Find user access patterns
- Prefetch most frequent access patterns

Client-side cache management algorithm:

Clients current location is L_y and has traversed location L_x
Use Markov model and predict future location L_z
Filter resource data matrix for locations $\{L_x, L_y,\ldots,L_z\}$
Bicluster the resource data matrix for$\{L_x, L_y,\ldots,L_z\}$
Recursive call to findResource( ) for max score within locations.

```
if Di is valid and in cache then
    validate and return the data item Di
else if cache miss for data item Di then
        request for data item Di to DB server
        get data item Di from DB server
      if enough free space in cache then
        store data item Di in cache
        update Pai
      else if not enough free space in cache then
            while ( not enough space in cache)
                create enough space by replacing data
                item(s) from cache
                    ✓ Invalid scope (vsi)
                    ✓ Minimum Pai
                    ✓ Large distance (dsi)
              If (Multiple replacement )
                Large size with scope invalidation.
             end
               Insert data item Di
               update Pai
    end
end
```

The Markov model leads to set of locations where the client will be in near future. The findResource() based on biclustering and finds max count of resources accessed for set of locations recursively. The max count of first resource is used to reduce the number of rows to be scanned. The second iteration of findResource() will find the max count of resource within the rows of first resource and so forth. The terminating condition of recursive function is the number of locations within current and next predicated location.

5. PERFORMANCE EVALUATION AND COMPARISON

For implementation of DMBCRP the database is created with different regions with locations, location specific resources such as shown in Figure 5. Some resources has specialty such as Child and Maternity for hospital and as applicable for other resource.

Data for user movement and query firing was collected. A data set consists of ten thousand records. Data for specific customer is obtained which is further pruned to form the data sets for evaluation. As Markov Model is computational intensive for large values of k, we have used only first and second order TPM with server side processing to reduce load of client processing. Figure 6. shows a scenario of the request for different data resource. The client sends request for information. The server

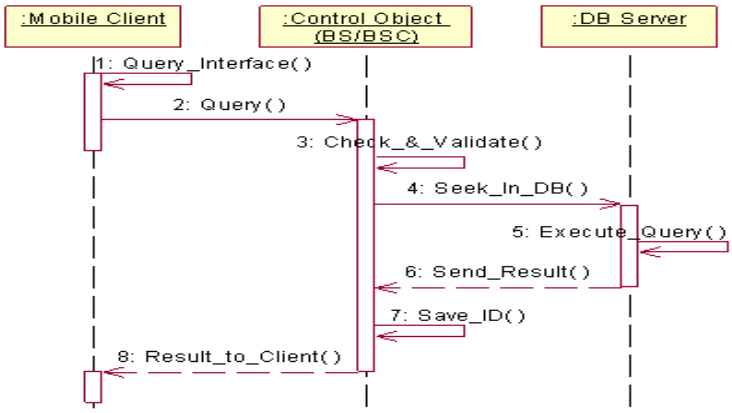

Fig 6: Sequence diagram for Information Request.

validates the client and for authenticated clients Database Server (DBS) responds with requested information.

Resource	ResourceID
Hospital	1
Police	2
Movies	3
ATM	4
Blood Bank	5
Medical Store	6
Restaurant	7

Fig 5: List of Location specific resource.

For our evaluation, the results are obtained when the system has reached the steady state, i.e., the client has issued at least 1000 queries, so that the warm-up effect of the client cache is eliminated. We have conducted experiments by varying the client speed, query interval and cache size. Query interval is the time interval between two consecutive client queries.

In order to predict and simulate the travel paths, close to realistic ones, some paths are chosen wherein no definite travel patterns are observed. One can say such mobile clients travel pattern show near random behavior. A typical output obtained from first and second order Markov model is shown in Figure 7. a) and Figure 7. b) with query interval from 10 to 200 seconds. Initially the result of PPRRP, MMCRP and DMBCRP is same since client will be in the same predicated region if query interval is small. DMBCRP outperforms when query interval is more because of accurate prediction of client movement and access patterns based on historical data better than MMCRP,PPRRP and far better than FAR and LRU.

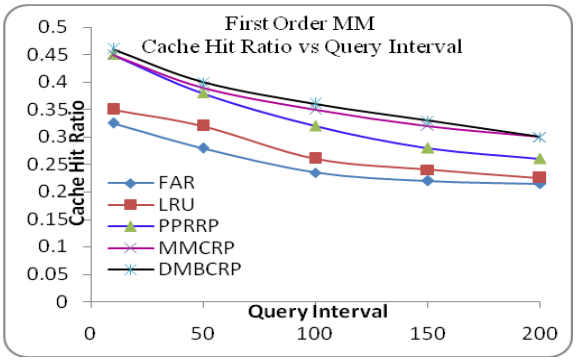

Fig 7: a) Cache hit Ratio vs Query Interval for first order.

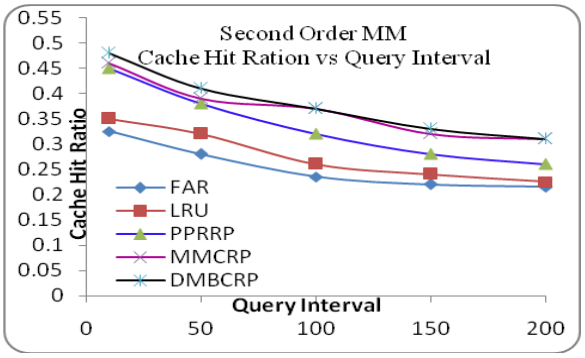

Fig 7:b) Cache hit Ratio vs Query Interval for Second order.

The Figure 8 shows the effect of cache size on performance of LRU, FAR, PPRRP, MMCRP and DMBCRP replacement policies. As shown and expected, the cache hit ratio of different policies increases with increase in cache since cache can hold more information which increases the probability of cache hit. The performance of DMBCRP will be increased substantially with increase in cache size so this result could be used to decide the optimal cache size in the mobile client.

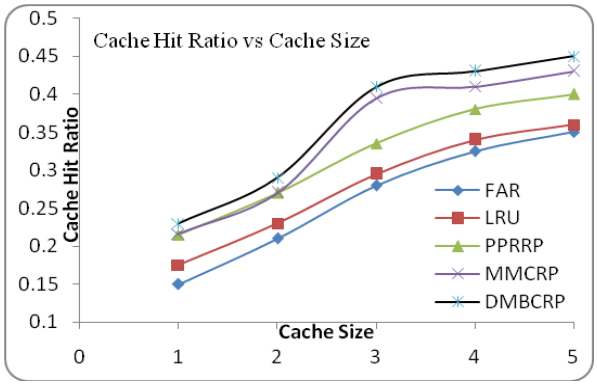

Fig 8: Cache hit Ratio vs Cache Size

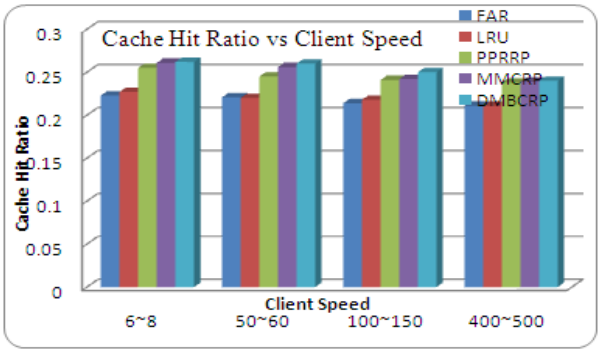

Fig 9: Cache hit Ratio vs Client Speed

The Figure 9 shows client speed vs cache hit ratio. Four clients' speed ranges have been considered: 6~8 km/hr, 50~60km/hr, 100~150 km/hr and 400~500 km/hr for walking human, car, train and plain respectively. For higher speed range, the cache hit ration decreases since client spend less time at each location and first/second order model, valid scope becomes less effective.

6. CONCLUSION

In this paper, we presented a cache replacement policy using first and/or second order Markov Model and biclustering. The presented DMBCRP takes into account both the spatial and temporal properties of client movement and access patterns to improve caching performance. DMBCRP takes into account valid scope, data size, data distance and access probability of data item for replacement. Whenever single (data item) storage results into multiple (three or more than three) replacements, we have considered this situation as critical and handled differently. Use of the time tested Markov model for prediction of client movement improves the performance. Simulation results for query interval, cache size and client speed show that the DMBCRP has significant improvement in performance than the LRU, FAR, PPRRP and MMCRP. In our future work we would like to incorporate a graph based Markov model for prediction and replacement.

7. REFERENCES

[1] D. Barbara 1999 Mobile Computing and Databases A Survey, In Proc. of IEEE Trans. on Knowledge and Data Engg.

[2] A. Kumar, M. Misra, A.K. Sarje 2006 A Predicated Region based Cache Replacement Policy for Location Dependent Data In Mobile Environment. IEEE-2006.

[3] D.L. Lee, Lee W. C, J. Xu, and B. Zheng 2002 Data Management in Location-Dependent Information Services, IEEE Pervasive Computing.

[4] B. Zheng, J. Xu, D. L. Lee 2002 Cache Invalidation and Replacement Strategies for Location-Dependent Data in Mobile Environments, In Proc. of IEEE Trans. on Comp.

[5] Q. Ren, M.H. Dhunham 2000 Using Semantic Caching to Manage Location Dependent Data in Mobile Computing, In Proc. of ACMIIEEE MobiCom.

[6] A. Balamash, M. Krunz 2004 An Overview of Web Caching Replacement Algorithms, In Proc. of IEEE Communications Surveys & Tutorials.

[7] E. O'Neil, P. O'Neil 1993 The LRU-k page replacement algorithm for database disk buffering, In Proc. of the ACM SIGMOD.

[8] Vijay Kumar, Nitin Prabhu, Panos K Chrysanthis 2005 HDC- Hot Data Caching in Mobile Database System, IEEE.

[9] I.A. Getting 1993 The Global Positioning System, In Proc. of IEEE Spectrum.

[10] A.Kumar, M. Misra, A.K. Sarje 2006 A New Cache Replacement Policy for Location Dependent Data in Mobile Environment, IEEE.

[11] Keqiu Li, Wenyu Qu, Hong Shen, Takashi Nanya 2005 Two Cache Replacement Algorithms Based on Association Rules and Markov Models, Proceedings of the First International Conference on Semantics, Knowledge, and Grid.

[12] M.H.Dunham, V. Kumar 1998 Location dependent data and its management in mobile databases, in Proceedings of the 9th International Workshop on Database and Expert Systems.

[13] Dimitrios Katsaros, Yannis Manolopoulos Prediction in Wireless Networks by Markov Chains.

[14] Hazem Hiary, Qadri Mishael, Saleh Al-Sharaeh 2009 Investigating Cache Technique for Location of Dependent Information Services in Mobile Environments, European Journal of Scientific Research.

[15] Heloise Mânica, Murilo Silva de Camargo 2004 Alternatives for Cache Management in Mobile Computing, IADIS International Conference Applied Computing.

[16] Qinghua Huang, Lianwen Jin, Dacheng Tao 2009 An unsupervised Feature Ranking Scheme by Discovering Biblusters, Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics. San Antonio, TX, USA.

[17] Y. Cheng, and G.M. Church 2000 Biclstering of Expression Data, in proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology (ISMB).

[18] Stefano Lonardi, Wojciech Szpankowski, Qiaofeng Yang 2006 Finding biclusters by random projections, Theoretical Computer Science, Volume 368, Issue 3, 10.

[19] Kai Puolamäki, Sami Hanhijärvi, Gemma C. Garriga, 2008 An approximation ratio for biclustering, Information Processing Letters.