International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:1, No:8, 2007

Cumulative Learning based on Dynamic Clustering of Hierarchical Production Rules (HPRs)

Kamal K.Bharadwaj, and Rekha Kandwal

***Abstract*—**An important structuring mechanism for knowledge bases is building clusters based on the content of their knowledge objects. The objects are clustered based on the principle of maximizing the intraclass similarity and minimizing the interclass similarity. Clustering can also facilitate taxonomy formation, that is, the organization of observations into a hierarchy of classes that group similar events together. Hierarchical representation allows us to easily manage the complexity of knowledge, to view the knowledge at different levels of details, and to focus our attention on the interesting aspects only. One of such efficient and easy to understand systems is Hierarchical Production rule (HPRs) system. A HPR, a standard production rule augmented with generality and specificity information, is of the following form

Decision **If** < condition>

Generality <general information>

Specificity <specific information>. HPRs systems are capable of handling taxonomical structures inherent in the knowledge about the real world. In this paper, a set of related HPRs is called a cluster and is represented by a HPR-tree. This paper discusses an algorithm based on cumulative learning scenario for dynamic structuring of clusters. The proposed scheme incrementally incorporates new knowledge into the set of clusters from the previous episodes and also maintains summary of clusters as Synopsis to be used in the future episodes. Examples are given to demonstrate the behaviour of the proposed scheme. The suggested incremental structuring of clusters would be useful in mining data streams.

***Keywords*—**Cumulative learning, clustering, data mining, hierarchical production rules.

I. Introduction

THE goal of the learner in conventional learning methods is to capture the inherent meaning of concepts meaning by observing concept examples, which can be given at once(batch learning) and incrementally. This paradigm works well for knowledge-based system applications which do not change in time. But many of the real life application are characterized by change of data. Even concepts are not static; they evolve over time. Applications such as dynamic knowledge-bases, intelligent agents and active vision systems violate many of the traditional assumptions of concept leaning. All training examples are not available at any given time; training examples are distributed over time. Consequently, the system must not only learn over time, but it may also learn a changing concept.[5]

The predominant representation of the discovered knowledge is the if-then rules because of its many advantages. However this representation often severely fragments the knowledge that exists in the data, thereby resulting in a large number of rules. The fragmentation also makes the discovered rules hard to understand and to use. Also the discovered knowledge is represented only at a single level of detail. This flat representation is not suitable for human consumption because we are more used to hierarchical representation of knowledge. Hierarchical representation allows us to easily manage the complexity of knowledge, to view the knowledge at different levels of details, and to focus our attention on the interesting aspects. A more efficient and easy-to-understand representation is in the form of Hierarchical Censored Production rules which has numerous applications in situations where decision must be taken in real time and with uncertain information. This representation is simple and intuitive, and also has a natural way of organizing the knowledge in a hierarchical fashion, which facilitates human analysis and understanding. Several extensions/generalizations of the system have been proposed (incorporating Fuzzy Logic [8],[13], DST [11], Genetic Algorithms [9] and Neural Networks [12]).

In this paper an attempt is made to exploit the inherent structural properties of HPRs, a form of HCPR where censors are completely neglected due to time constraint, to accommodate cumulative learning scenario. A dynamic system which comprehends the knowledge with each episode is developed. Results on the behaviour of the proposed scheme are also included.

II. Background

The concept of CPR as suggested by Michalski and Winston has the following form:

If P {premises/preconditions}
Then D {actions/decision}
Unless C {censor conditions}

A censor is a low likelihood condition when hold will block the rule. So when the system is having low resources, it can skip checking the censor conditions. If the resources are available, the censor conditions are examined, increasing the certainty factor of making a high speed decision or reversing the decision itself. The above concept of CPR has been extended to HCPR to incorporate both aspects of precision namely certainty and specificity. Two new

Kamal K.Bharadwaj is a Professor at School of Computer and System Sciences (SC&SS), Jawaharlal Nehru University (JNU), New Delhi-110067, India (phone:91-9810196636; e-mail: kbharadwaj@gmail.com).

Rekha Kandwal is a Ph.D scholar at SC&SS, JNU, New Delhi, India (phone: 91-9811529226; email: rekha.kandwal@gmail.com).

operator added to CPR and we have the concept of HCPR having the general form as follows:

D {decision/concept/action}
If P [p1,p2,p3,…,pn] {preconditions}
Unless C [c1,c2,…,cn] {censor conditions}
Generality [G%] {general information}
Specificity S [s1,s2,…,sk]
{mutually exclusive set of specific information}

As a special case, dropping the unless operator due to time constraint, HPR takes the form

D{decision/concept/action}
If P[p1,p2,…,pn] {preconditions}
Generality [G%] {general information}
Specificity S [s1,s2,…,sk]
{mutually exclusive set of specific information}

Here is an example set of related HPRs.

{level 0}
Is_in_city(X,Y):
If [Lives_in_city(X,Y)]
Generality []
Specificty[Is_at_home(X),Is_outside_home(X)]
{level 1}
Is_at_home(X):
If [Lives_in_city(X, Y), Time (night)]
Generality [Is_in_city(X, Y)]
Specificty[]
Is_outside_home(X):
If [Lives_in_city(X, Y), Time (day)]
Generality [Is_in_city(X, Y)]
Specificty[Is_working_outdoor(X)
,Is_entertaining_outdoor(X)]
{level 2}
Is_working_outdoor(X):
If [Lives_in_city(X, Y), Time (day), Day (working)]
Generality [Is_outside_home(X)]
Specificty[]
Is_entertaining_outdoor(X)]:
If [Lives_in_city(X, Y), Time (day), Day (Sunday)]
Generality [Is_outside_home(X)]
Specificty[]

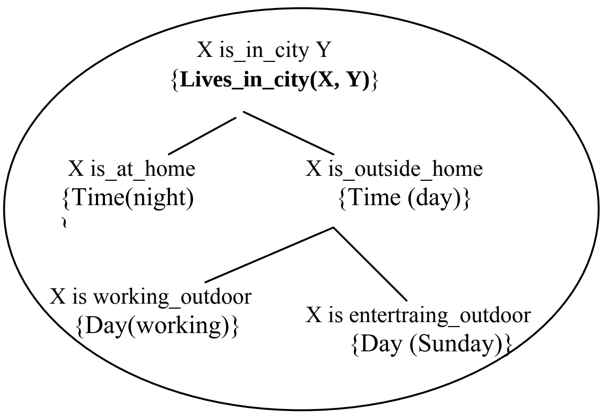


Fig. 1 HPR-tree- Cluster of related HPRs

In the following discussion a set of related HPRs is called a cluster and is represented as HPR tree. The cluster formed by the above HPRs is represented as HPR-tree in Fig.1. Now onwards instead of writing the set of HCPRs in a cluster i, only the HPR-tree$_i$ will be given.

The root represents the most general concept in a HPR tree and any child in tree is more specific case of its parent. As the concept becomes more specific, the number of elements in its precondition part will increase obviously. However it is not required to list all such elements because total inheritance is an inherent feature of the HPRs tree structure; each HPR inherits the entire preconditions set of its parent HPR, and thus of all of its ancestors. So the redundancy is minimized in the listing of preconditions in the child node. HPR system collect fragmented knowledge and represent these as collective one and hence significantly reducing the knowledge base. This representation scheme reduces the complexity of the discovered knowledge substantially, makes knowledge base easy to understand and efficient for future processing.

Jain and Bharadwaj [4] used the term "fusion" for merging two related HCPR trees. Two related HCPR trees can be merged into one if there are some common properties in the preconditions set of the roots of these two HCPR tree. The trees merged may not remain in their original form but the hierarchy of each tree is maintained .Fusion algorithm works as follows:

Fusion(X,Y):Merges two HCPR trees having roots X and Y
/* In the following discussion, IF(X) denotes the set of preconditions for the decision X*/

1. if (IF(X) ∩ IF(Y))= ϕ)
then printf (" No fusion possible")
2. if (IF(X) ⊂ IF(Y))
{ then { X will be the root of the new combined tree}
T1 ← X
T2 ← Y
}
else if (IF(Y) ⊂ IF(X))
{ then { Y will be the root of the new combined tree}
T1 ← Y
T2 ← X
}
else
{{A new root is created for new combined tree}
IF(new_root) ← {(IF(X) ∩ IF(Y)}
Specificity(new_root) ← {X,Y}
IF(X) ← IF(X)-IF(new_root)
IF(Y) ← IF(Y)-IF(new_root)
Generality(X) ← [new_root]
Generality(Y) ← [new_root]
}
3. Find where tree T2 would be attached in the tree with root T1 and attach it there.

Two related HCPR tree and their merging by Fusion algorithm is shown in Fig. 2.

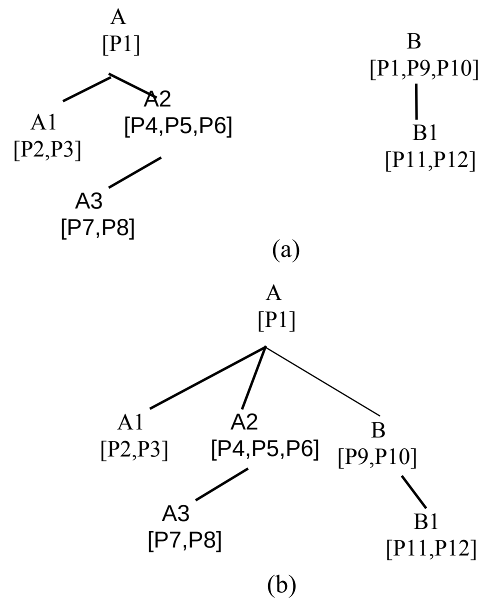


Fig. 2 (a) HCPR trees before fusion (b) Final tree after fusion

III. HPRs Clusters and Cumulative Learning

Clustering is the process of grouping the data into groups so that objects within a cluster have high similarity to each other and have dissimilarity to objects in other clusters. The objects here are grouped on the principle of maximizing the intraclass similarity and minimizing the interclass similarity. Representing data by a few clusters loses certain fine details but achieves simplification. Arranging voluminous data into few cluster is a challenging task as it is to be done using a limited memory. Clustering is a dynamic field of research in data mining. Many clustering algorithms have been developed. These can be categorized into partitioning methods, hierarchical methods, density-based methods, grid-based methods, and model-based methods.

The basic idea of Cumulative Learning in general is to have the agent solve a series of related tasks in some sequence, and then, while solving the tasks, speed up learning a particular task by using information or knowledge obtained solving from previous tasks. One way to look at Cumulative Learning is as a way to set bias for a new task using knowledge accumulated from solving previous tasks. Since the performance (in terms of no. of examples required to learn) of a learning agent depends to a large extent on the bias given to it in the beginning, Cumulative Learning helps speed up learning. Inherent properties of HPRs can be exploited to implement cumulative learning scenario in this system

Our focus is on the monitoring of cluster formation process so as to have deeper insight into the changing trend of data i.e. the comparison of clusters formed at different instances of time with the new piece of knowledge mined, and then adjusting cumulatively this new knowledge appropriately in one of the clusters or forming a brand new cluster of knowledge. The objective is online, dynamic detection and summarization on interesting changes, to know how well the model constructed from the previous data fits the new data or we can say that by how much the old model misrepresents the new data. An algorithm is proposed that accommodate the new piece of knowledge appropriately in one of the clusters of previous episode or forming an absolutely new cluster. All clusters of this new episode will act as the knowledge of previous episode.[10].

IV. Proposed Methodology

The new piece of knowledge obtained in each episode is compared with the previous clusters, a correspondence needs to be established between old and new clusters that is which new cluster is to be compared with which old cluster. After the clustering for an episode is done, a synopsis of the clustering is stored and is used for obtaining cumulative clustering as further stream arrives. The synopsis reflects the trends of the historical data. Once clusters are obtained and finalized in an episode, the comparison parameters are also calculated and stored in the synopsis. Here Synopsis is the set of clusters from previous episodes. The attribute of a particular cluster is its root_property, shown with bold face, is the information at the root node of the corresponding HPR tree. For any clusters C1and C2, we define a boolean function, compare_attribute (C1,C2) as:

$$1 \text{ if root\_property(C1)} \cap \text{root\_property(C2)} \neq \phi$$
and 0 otherwise.

The whole process is depicted in Fig. 3.

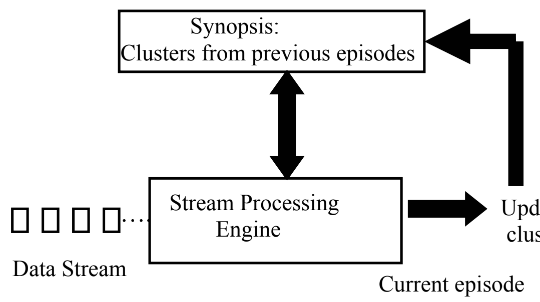


Fig. 3 Proposed Cumulative clustering approach

Algorithm: Cumulative Clustering Scheme
Input: A set of current clusters and Synopsis
Output: Updated Synopsis

1. **IF** compare_attribute (C_j, C_i)=1 for any cluster C_i, C_j, where C_j is cluster from current episode and C_i is the cluster from previous episode then
   a. Fusion(C_j, C_i) $\rightarrow$ C_i'
      The root_property of a cluster gets updated as per the following if-else statement:
      if (root_property (C_i) $\subset$ root_property(C_j))
        then root_property(C_i')= root_property(Ci)
      else if(root_property(C) $\subset$ root_property(C))
        then root_property(C_i')= root_property(C_j)
      else
        root_property(C_i')= root_property(C_j) $\cap$ root_property(C_i)
   b. Add cluster new C_j' to the synopsis and delete cluster C_i from the synopsis, delete C_j from current set of cluster .
2. **ELSE** add C_J to the synopsis, delete C_j from current set of cluster .

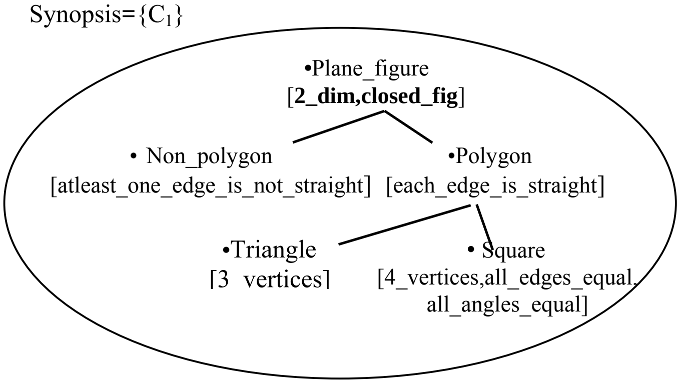
3. Repeat step 1-2 for remaining C_j, C_i where $C_i \in$ Synopsis and $C_j \in$ current set of clusters.

The Output, updated synopsis will be the Synopsis for the next episodes.

V. Experimental Results

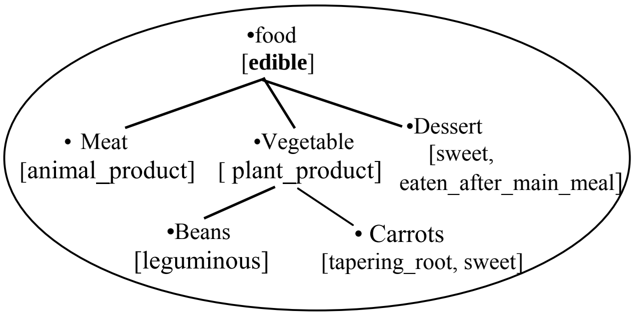
The algorithm is tested on a real life data and synthetic data.

Example 1: Suppose the following HPR tree is obtained in the current episode, depicting HPR cluster of plane figures. The synopsis is initialized with cluster C1, that is Synopsis={C_1}

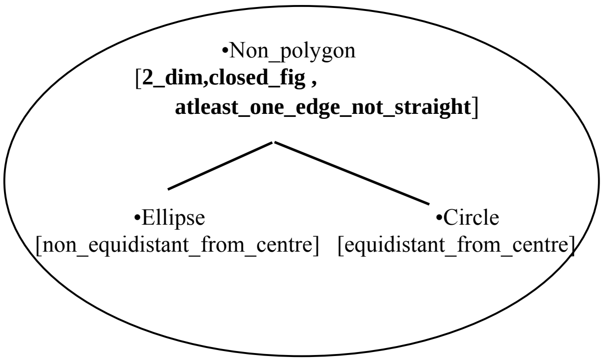


Suppose the cluster obtained in second episode, say C_2 is depicting the concept of Food items.

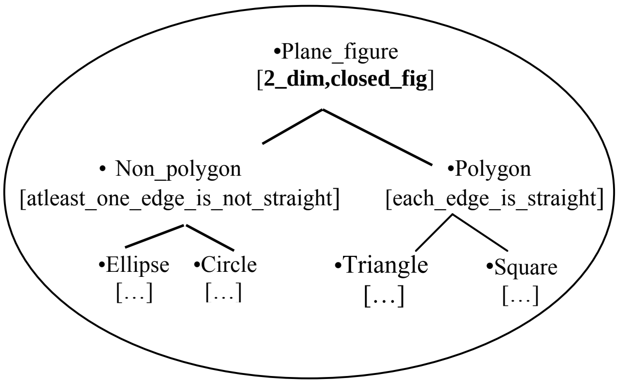
Now compare_attribute(C_2,C_1)=0, so a new cluster is added to synopsis. Synopsis={ C_2,C_1 }



Assuming that in the third episode,cluster C_3 is obtained depicting the concept of non polygon figures.



Now compare_attribute(C_3,C_2)=0 so Fusion(C_3,C_1) not possible and compare_attribute(C_3,C_1)=1 so C_3 gets merged in C_1 producing a more refined cluster C_1' for future episodes.



At this stage, Synopsis={ C_1',C_2}

Example 2: Consider Fig. 4 as the synopsis from the previous episode, contains two clusters C_1 & C_2. The synopsis obtained after two episode is shown in Fig. 5.

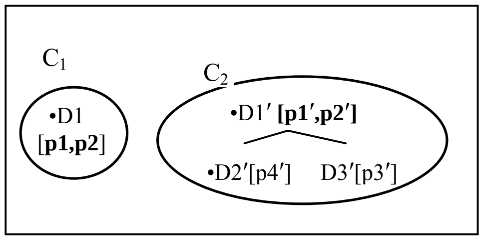
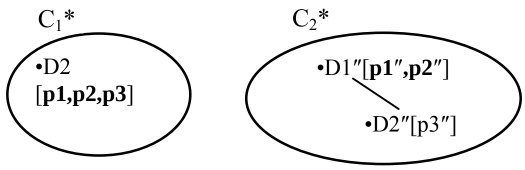


Fig. 4 Synopsis from previous episode

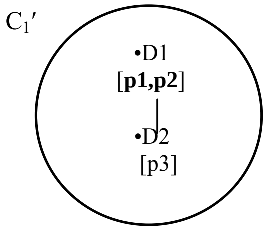
Episode 1:

Current set of clusters



Now compare_attribute(C_1*, C_1)=1 so

Fusion(C_1*, C_1) $\rightarrow$ C_1'



Add C_1' to the synopsis. delete C_1 from there. Delete C_1* from current set of clusters.

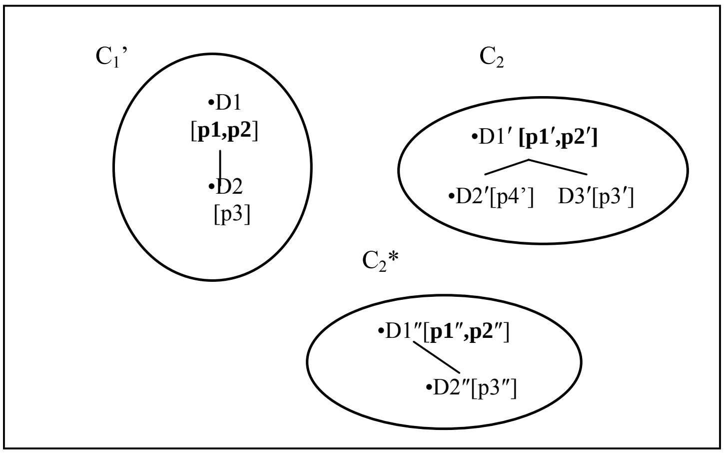
compare_attribute(C_2*, C_2)=0 so no Fusion.

compare_attribute(C_2*, C_1')=0 so no Fusion .

Add C_2* to the synopsis. Delete C_2* from current set of clusters.

ISSN: 2517-9942
Vol:1, No:8, 2007

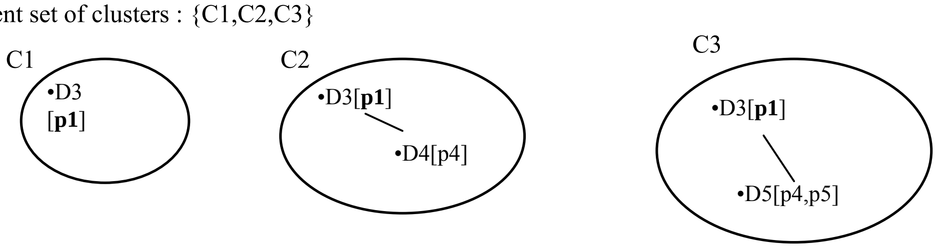
Synopsis after first episode:



This will act as a synopsis for next episode.

Episode 2:

Current set of clusters : {C1,C2,C3}



After second episode

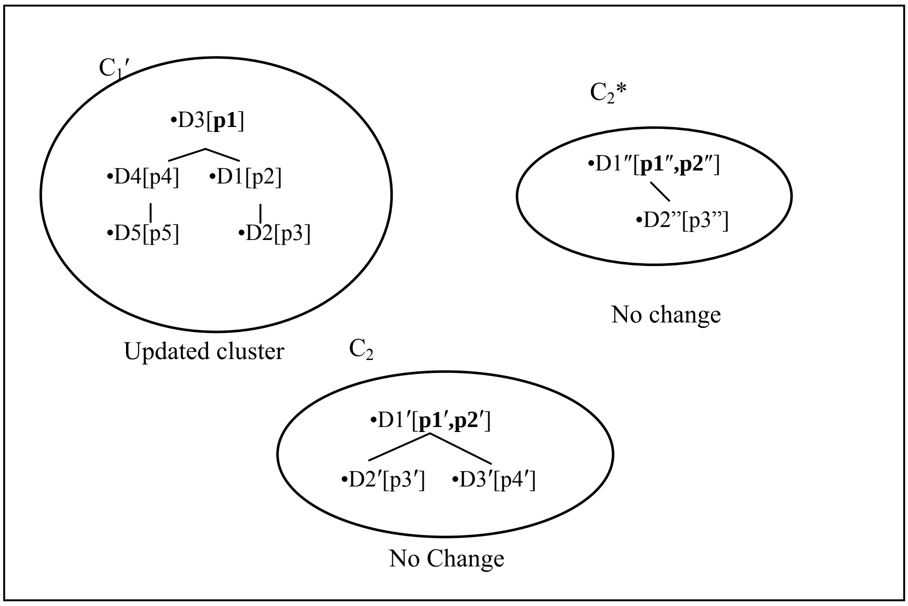


Fig. 5 Synopsis after two episode

VI. Conclusion

The paper has discussed a novel cumulative learning methodology based on dynamic structuring of Hierarchical Production Rules (HPRs) clusters. The main advantage of the method is the high comprehensibility of the knowledge representation used and the employment of a symbolic learning approach Fusion [4] that allows incorporation of new knowledge into the knowledge gained during previous episodes. The proposed system restructures clusters with each episode and maintains a summary of clusters with minimum redundancy for future episodes. The proposed methodology would be useful in mining data streams and in the development of dynamic knowledge based systems.

References

[1] Han, J., Kamber, M. "Data *mining: Concepts and Techniques*" Academic Press (2001).
[2] Adriaan, P., Zantingre, D. "Data *Mining*", Addison Wesley, 1999.
[3] Ryszard S. Michalski, Pavel Brazdil: Introduction, Special Issue on Multistrategy learning, Machine Learning, vol 50, pp 219-222, 2003.
[4] Jain, N.K. ,Bharadwaj, K.K.,: Some Learning Techniques in Hierarchical Censored Production Rules(HCPRs) System, International Journal of intelligent systems, John Wiley & sons, Inc.,vol. 13,pp 319-344,1997.
[5] Marcus A.Maloof and Ryszard S. Michalski : "Learning Evolving Concepts Using Partial-Memory Approach", *Working Notes of the 1995 AAAI Fall Symposium on Active Learning*, 1995.
[6] Bharadwaj, K.K., Jain, N.K.: Hierarchical Censored Production Rules (HCPRs) System, Data and Knowledge Engineering, vol.8 (North Holland), 1992.
[7] Fadl M.Ba-Alwi and K.K.Bharadwaj: "Automated discovery of hierarchical ripple-down rules(HRDRs)", *In the Proc of Twenty-thirdIASTED International Conference on Artificial Intelligence and Applications*(AIA 2005),Innsbruck,Austria,February 14-16,2005.
[8] Fadl M.Ba-Alwi and Bharadwaj, Kamal.K::"Discovery of Production Rules with Fuzzy Hierarchy",ENFORMATIKA ,vol 1 ,2005 ISBN 975-98458-3-0.
[9] Basheer M. Al-Maqaleh and Kamal.K.Bharadwaj::"Genetic Programming Approach to Hierarchical Production Rule Discovery",ENFORMATIKA ,vol 6,2005 ISBN 975-98458-5-7.
[10] Rekha Kandwal and Kamal.K.Bharadwaj: " A Cumulative Learning Approach to Data Mining Employing Censored Production Rules (CPRs)",to appear in the Proc of 5th International Enformatika Conference, Prague, Czech Republic, August 26-28, 2005.
[11] Bharadwaj, K.K., Neerja, Goel, G.C.: Hierarchical Censored Production Rules (HCPRs) Systems Employing the Dampster-Shafer Uncertainty Calculus, Information and Software technology, Butterworth-Heinemann Ltd. (U.K.) Vol. 36 No., 155-164, 1994.
[12] Jose Demisio Simoes da Silva, Bharadwaj K.K., "Integration of Hierarchical Censored Production Rules (HCRPs) System and Neural Networks" , *SBRN'98, Proceedings of IEEE Computer Society, Los Alamitos, California, USA, pp73-78, Dec 1998.*
[13] Neerja , Bharadwaj K.K., "Fuzzy Hierarchical Censored Production Rules System" Int. Journal of Intelligent Systems , John wiley & sons (New York), vol 11, No.1,pp 1-26 (1996).
[14] Brian Babcock, Shivnath Babu, Mayur data, Rajeev Motwani, and Jennifer Widom: *Models and Issues in data Stream Systems*, Proceeding of 21st ACM Symposium on Principles of Database Systems (PODS 2002).
[15] Guozhu Dong, Jiawei Han, laks V.S. Lakshmanan, Jian Pei, Haixun Wang, Philip S. Yu: *Online Mining of changes from data Streams: Research Problems and Preliminary Results*, In Proceedings of the 2003 ACM SIGMOID Workshop on Management and Processing of data Streams.