Redescription Mining With Three Primary Data Mining Functionalities

M. Kamala Kumari[#] and Prof. Suresh Varma[*]

[#]Department of Computer Science , Adikavi Nannaya University ,Rajahmundry , Andhra Pradesh, India
kmarepalli@yahoo.com

[*]Department of Computer Science , Adikavi Nannaya University, Rajahmundry, Andhra Pradesh, India
psverma@gmail.com

Abstract

Describing an object in two ways or shifting the vocabulary of the same concept is Redescription. Not a new problem, Redescription Mining premise had resulted the subsets of objects that afford multiple definitions, in a given Universal set of the same, and a collection of features to describe them. Now-a-days, huge amounts of data available either to classify or to categorize leads us to ambiguous state as it is accomplished with complementary and contradictory ways. Hence data has to be reduced. This involves cataloging, classification, identifying rules among the data, segmentation or partitioning of the data. The Learning algorithms of data mining techniques on this data can often be viewed as a further form of data reduction. This Sine-qua-non data has been characterized by the multitude of descriptors. In a way, these descriptors are also made equivalent and hence reduced. The methodology of redescriptions can be obtained in scores of data mining techniques. In this paper we overview how data mining functionalities like classification, clustering and Association rule mining achieve the goal of redecsriptions.

Keywords

Data mining, Redescription mining, algorithms, Association rules, Classification, Clustering.

1. Introduction

Redescription mining comes in to the category of data mining problem which is not a new one [3] . This aims at finding the subsets of data that afford multiple definitions. The input to this problem will generally have universal set of objects of some domain and some features or vocabulary to describe them. We try to give the subsets of objects that afford at least two definitions as output. For example, given the Natural Hazards as descriptors and the affected cities as objects, we can find the subsets of cities described in another way by including, set theoretic expressions between hazards. This can be accomplished by finding the association rules and closed itemsets between them, by classification rules as expressions by constructing decision trees the classes obtained as objects from them, by clustering where the objects defined by similar descriptors are grouped in to a cluster. The table below shows the records according to United Nations Environment Programme the hazards as descriptors and the affected cities as objects in Table 1. Redescriptions can also be considered as a useful way to reason about overlaps, similarities and differences in the given vocabulary [4]. The example below shows the Natural Hazards as descriptors and the most affected countries of Asia-Pacific regional countries as objects. We list the most affected countries of each natural hazard considered. This data if we

DOI : 10.5121/ijcsit.2012.4406

take as an input to Redescriptions we generate an output with a changed vocabulary for the subset of countries.

Table 1. An example input to Redescription mining.

Cyclones	Philippines,(Ph) Bangladesh,(B) Vietnam,(V) Pacific Islands(PI), Solomon islands(SI)
Earthquakes	Japan(J), Philippines(Ph), India(I)
Volcanoes,	Philippines,(Ph), Indonesia (I), Japan (J), New Zealand(NZ), Solomon Islands(SI)
Tsunamis	Japan (J), Philippines(Ph), India(I)
Droughts	Philippines(Ph), Thailand (T), Australia (A), Pacific Islands (PI).
Floods	India (I), Bangladesh(B)

In a simple way of set related expressions, we can generate the same result with two equivalent statements. To see how, consider the hazards and the countries affected shown in Table 1. ‘Countries with Earthquakes history’ $\Leftrightarrow$ ‘Countries with Tsunamis’. Here the redescription involves a subset definable in two ways. The result of this is 3 countries, Japan, Philippines, India. So these countries satisfy both the descriptors. And hence they can be said to have redescribed. Similarly, we also have, $E \cap V \Leftrightarrow T - F$. The strength of this redescription is measured by symmetric Jaccard’s Coefficient, JC, θ. The goal of redescription mining is to find equivalence relationships of the form $A \Leftrightarrow B$ that hold at or above a given Jaccard’s Coefficient, θ i.e., $| A \cap B| / | A \cup B |$. Where A and B are set theoretic expressions involving given disjoint descriptors. Here in our example the strength of the redescription is 1, as it is 3/3 =1. Descriptors on either side can involve more than one descriptor as well as more than one entity too. In that case, there can be the values of Jaccard’s coefficient as 0.6 or 0.56 or 0.8 and so on. The consideration of JC depends on what threshold we give to it. This redescriptions can be obtained by three primary Data mining functionalities viz., Association Rules, Classification, and clustering concepts.

2. Redescriptions with Classification

With Classification, according to [3] a typical approach has been introduced to mine patterns like set theoretic expressions. Initially, a tree will be constructed based on one set of descriptors as features and the other set as classes. Here the descriptors should be unique. Next, the tree is fixed and the second iteration takes by considering the classes as features and vice-versa. The property of redescription space exploits them in pairs. For example, if $A \Leftrightarrow B$ then $B \Leftrightarrow A$ co-exists. The properties of the trees are when exploiting are specified as, if the nodes in the tree correspond to Boolean membership variables of the given descriptors, then we can interpret paths to represent set intersections, differences, or compliments; Unions of paths would correspond to disjunction, a partition of the path in the tree corresponds to a partition of objects. These properties are employed in CARTWheels which grows trees in two different directions so that they are joined at leaves as shown in Fig 1.

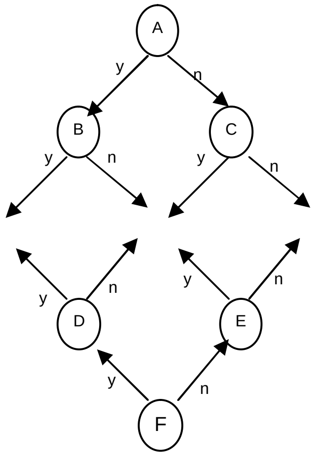

Fig: 1. CARTWheels tree merging at leaves. The coincided leave paths can be considered as redescriptions.

One tree grows with the partition of one subsets of objects and the other tree is grown in opposite direction to match this partition using other subset of objects. If partition correspondence is established, then the paths that join can be read off as redescriptions. These trees are regrown alternatively by changing features to classes and classes to features. By suitably configuring this alternations and matchings, we aim to get the redescriptions of the dataset if they exist with non zero probability. From the above example, the two alternating trees merge at leaves. Redescriptions can be constructed from the paths where they meet. If we consider the Universal set of Objects as O, and the subsets as A,B,C,D,E and F, then the redescriptions from Fig 2 are, O-A-C ⇔ O-F-E. Similarly, A-B ⇔ F-D. The sequence of steps involved in CARTWheels are given as below:

The search for redescriptions in CARTWheels can be viewed as a problem of identifying correlated random variables. The inputs to the algorithm are the Universal set of objects, O and descriptor sets {A} and {B}.Output will be Redescriptions, R which are nothing but the set expressions. The parameters considered are θ for Jaccard's coefficient, ρ for number of class participations allowed and η for maximum number of consecutive unsuccessful alternations.

The initializations are as follows:
Set answer set Redescriptions, R = { }
Set class participation counts for all {A} , {B} = 0
Set feature set F = {B}; set classes C = {A} Set dataset D= construct_dataset(O,F,C)
Set tree t = construct_tree(D,d) C = paths_to_classes(t) Flag = false, count = 0

The methods that are used are:

Impurity(t, l) : To assign the second best class label to the chosen leaf, l This is to maintain impurity.
Eval(t, θ) : This returns redescriptions satisfying the Jaccard's threshold, θ.
Construct_dataset(D, d) : Constructs dataset from objects, features and classes.
Construct_tree(D, d) : Constructs tree from the above dataset, D and depth of the tree given by d.
Paths_to_classes(t) : Identifying paths from the tree constructed.

For tree alternation:

Initially set Universal set G to classes, X

Repeat Until the limit for alternations is permitted

- Change features to classes and vice- versa based on flag setting
- Construct dataset, D
- Construct tree, t
- Impurify tree if all leaves have the same class Else
- return redescriptions if θ is satisfied
- If there are no Redescriptions yet, Increment counter as it is unsuccessful, Else
- Let the counter be zero
- Check for all, whether a class(path from tree) is in some redescription
- Collect those descriptors containing that classes,
- For each descriptor, D increase its participation count
- If it's count is greater than ρ , remove it from Universal G
- Get R by adding with the old ones
- Change flag value, not(flag)
- Get Classes to paths, C(t)

The important tunable parameter is ρ, controlling tradeoff between redundancy and exploration. A participation count is incremented each time a given descriptor appears in a redescription in its role as part of class, and when this reaches ρ, the descriptor is removed from consideration. The parameter specifies the maximum of number of alternations that CARTWheels can go through without mining any redescriptions.

3. Redescriptions with Association rules

Mining frequent patterns or closed item sets or maximal item sets is a fundamental and essential problem in many data mining applications. The result of this can be the discovery of association rules, strong rules, correlations, sequential rules, episodes, multidimensional patterns, and redescriptions.

Take an example with items as Natural hazards and their occurrences as hazardset for every year where we consider this as transaction period. CHARM is an efficient algorithm, for enumerating the sets of all frequent closed hazardsets, and CHARM-L is efficient algorithm to generate frequent closed item set lattice [6]. CHARM performs a search for closed frequent sets which are item set, HT over a novel HT-tree Fig 2, search space. Each node in the HT-tree, represented by an hazardset-tidset pair, is an prefix based class. All the children of a given node belong to its equivalence class since they all share the same prefix. Frequent pattern enumeration is straight forward in the HT-Tree framework. For a given node or prefix class, we can perform the intersections of the tidsets of all pairs of elements in a class and check if minimum support is met. This support counting is parallel with generation of new nodes. Each resulting frequent hazardset is a class unto itself, with its own elements, that will be recursively expanded. CHARM performs on this basic enumeration scheme, using the conceptual framework provided by the HT-tree. Moreover, there are four basic properties of HT-pairs that CHARM leverages for fast exploration of closed sets. CHARM algorithm by [8], starts by initializing the prefix class of nodes to be examined to the frequent single items and their tidsets. Next CHARM checks the four properties of each combination of HT- pairs appearing in the prefix class. For each HT-pair, it combines with other HT-pairs, that come after it according to a specific order given. This routine will modify the current class by deleting HT-pairs that are already subsumed by other pairs. It also inserts the new HT pairs in the new class. It has flexibility to modify the prefix class based on the properties of hazardset pairs. We then insert the hazardset in the closed set, provided that the hazardset is not subsumed by a previously found closed set. This process repeats recursively and in a depth-first manner. CHARM finally outputs the closed hazardsets. To generate minimal redescriptions we need to construct the lattice of descriptor sets. Lattice allows us to efficient mining of association rules based on closed hazardsets. This results in efficient and effective method for mining closed hazardsets.. As CHARM does not output the lattice explicitly, this can be achieved by an efficient algorithm CHARM-L. This begins as the same as CHARM by initializing the parent class with the frequent items and find closed hazardsets through satisfying hazardsets and tidsets/dsets properties. The basic idea is that whenever a new closed set X, is found, we efficiently determine all its possible closed supersets, $S = \{Y \mid Y \in C \cap X$ is a subset of $Y\}$, where C is the closed sets. The minimal elements of S form the immediate supersets or parents of X in the closed dset/hazardset lattice. It then makes a call to the extension subroutine, passing it the parent equivalence class and the lattice root as the current lattice node. Though it is possible to construct the lattice from the set of closed sets, since its time complexity is $O(|C|^2)$ [10] which is too slow for large number of closed sets, CHARM itself is extended to compute lattice while it generates closed hazardsets in CHARM-L.

3.1 Example

Here we show an example database of Natural Hazards in Fig 1 containing Notation for Natural hazards, their frequency in any transaction or we can consider those transactions as years and the HAzardsets with minimum support as 50%. We also show a Hazardset and transaction set tree as HT-tree.

DISTINCT DATBASE HAZARDS

Earth Quakes	Volcanoes	Tsunamis	Floods	Cyclones
E	V	T	F	C

DATABASE

Transaction	Hazards
1	EVFC
2	VTC
3	EVFC
4	EVTC
5	EVTFC
6	VTF

FREQUENT HAZARDSET WITH MINIMUM SUPPORT= 50%

Support	Hazardsets
100%(6)	V
83%(5)	C, VC
87%(4)	E,TF,EV,EC,VT,VF,EVC
50%(3)	EF,TC,FC,EVF,EFC,VTC,VFC,EVFC

Fig 2. Example Database showing notation for Hazards, Hazard database and Frequent Hazards.

CHARM performs a search for closed frequent sets over a novel HT-tree search space, shown in Fig 3. Each node in this tree represents an hazardset-tidset pair, $X \times t(X)$, is in fact a prefix based class. All the children of a given node X belongs to its equivalence class since they all share the same prefix X. We denote an equivalence class as [P] = {l1, l2….ln}, where P is the parent node and each li is a single item, representing the node, Pli × t(Pli). A class member represents one child of the parent node. A class represents items that the prefix can be extended with to obtain a new frequent node. Obviously no subtree of an infrequent node has to be extended. The power of the equivalence class approach is that it breaks the original search space in to independent subproblems.

3.2 Findidng out Frequent Patterns

To find out frequent pattern enumeration for a given node or prefix class, one can perform intersections of the tidsets of all pairs of elements in a class and check if minimum support is met, where support counting is simultaneous with generation. Each resulting frequent itemset in a class unto itself, with its own elements, that will be recursively expanded. This approach goes in a depth first exploration of the HT-tree for all frequent patterns. CHARM improves upon this basic enumeration scheme, using the conceptual framework provided by the HT-tree.

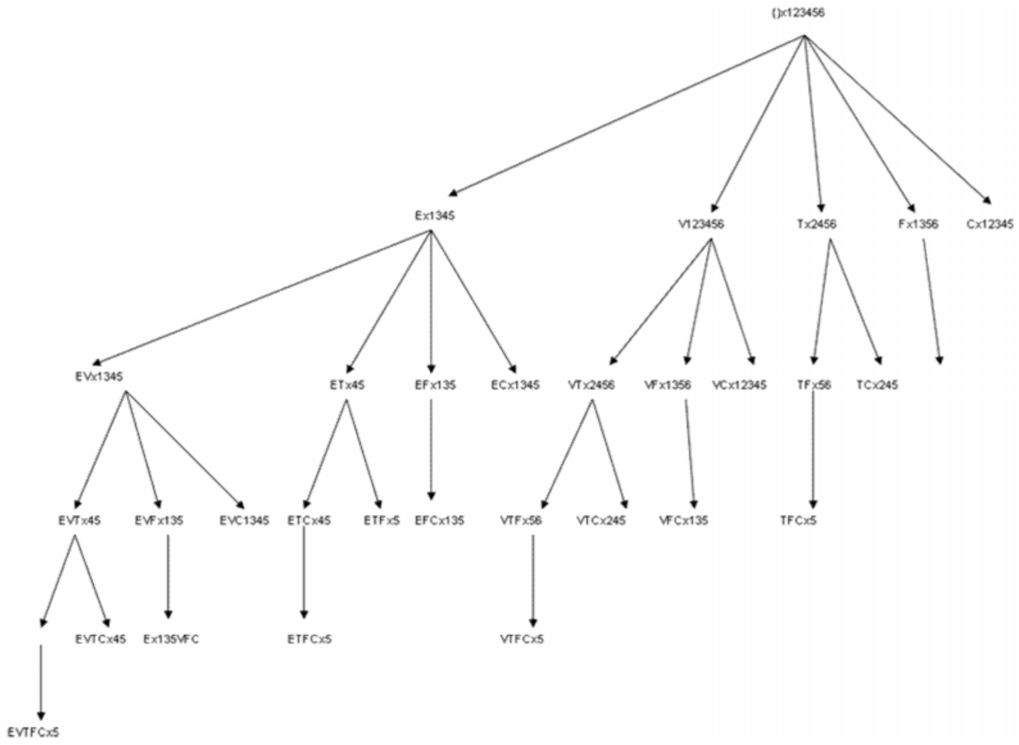

Fig 3. HT-tree: Hazardset- transaction set search tree.

3.3 Lattice generation by CHARM-L

CHARM-L starts in the same way as CHARM by initializing the parent class with the frequent items. It then makes a call to the extension procedure, CHARM-L-EXTEND passing it the parent equivalence class and the lattice root as the current lattice node. This extension procedure takes as input the current lattice node, and an equivalence class of IT-pairs. Whenever CHARM-L generates new closed itemset, it assigns it a unique closed itemset identifier, called *cid*. In CHARM-L, each element has associated with it a *cidset*, $\mathbb{C}$ which is the set of all *cid*s of closed itemsets that are supersets of that particular element. CHARM-L enumerates all closed sets which are not subsumed, but in addition it also generates a new Lattice L_n, for the new closed set and inserts it in the appropriate place in the closed itemset Lattice, L. This new Lattice becomes the current node in the next recursive call of the extension subroutine. Since the list of closed supersets of an element, may change whenever a new closed itemset is added to the lattice a check is made to update *cidset*, $\mathbb{C}$ of that element for each remaining element in the class. CHARM-L shares the optimizations for computing length 2 itemsets. To check if a new itemset, P is closed or not, a new method subsumption-check-lattice-gen is called. This routine takes current lattice node, the new itemset X, and the cidset $\mathbb{C}(X)$ as input. It then checks if X is subsumed. That is consider all itemsets, S which are supersets of X. If X has the same support as S then X is subsumed and we return. Else, the new lattice node is initialized as $L_n = X$. Each node maintains a list of parents which are immediate subsets and children which are list of immediate supersets. We now add L_n as the child of current node, L_c and L_c as the parent of L_n. Out of all the closed supersets of L_n, minimal supersets, S^{min} are found. Each minimal superset becomes a parent of Ln. For every child of the superset,*CS* if the child is a subset of the new lattice node L_n generated, then its parents pointers have to be adjusted; remove the identified superset from the *CS*'s parent link. Finally return new lattice node, $L_{n.}$ Once the set of all closed dsets, $\mathbb{C}$ for a given dataset has been found using CHARM-L, to get redescriptions we next need to generate minimal generators, *MG*(*X*) for each dset *X* belongs to $\mathbb{C}$. A minimal generator Z of a closed set X is a minimal dset that is a subset of X, but not a subset of any other X's immediate closed subsets in the closed dset lattice, L.

4. Redescription Mining with Conceptual clustering

One of the datamining tasks, clustering comes in to unsupervised classification. According to [5] ,the approach to clustering which clusters objects in to groups representing a priori defined conceptual entities is called conceptual clustering. Conceptual clustering generates an unique concept description for the obtained classes. Most of the conceptual clustering methods are capable of generating hierarchical category structures. The conceptual clustering differs from data clustering by not only forming cluster structures of the data but also, the data description language which is available to the learner. Thus, a statistically strong grouping in the data may fail to be extracted by the learner if the prevailing concept description language is incapable of describing that particular regularity. Now to justify the redescriptions with conceptual clustering we need to derive at least two concepts for a cluster. This gives two descriptions for a cluster. The obtained descriptions for a single cluster can be called as redescriptions. The primary goal of conceptual clustering is not a simple generation of mathematical function of cluster but a more abstract notion of cluster describability [6]. Clusters are chosen with a more favor of ease with given vocabulary.

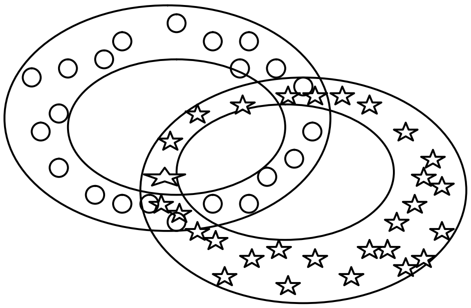

Fig 4: A conceptual cluster : Some intersected points in both the clusters share some property from the entire set of given data set. These points belong to both the clusters.

The input to conceptual clustering is typically a dataset, along with a set of features for labeling clusters. Concepts that characterize the output clusters from conceptual clustering are usually conjunctions between relations for variables that describe the entities. In the fig 4, there are two intervened circles and the areas where they have intervened can be described by the concepts of both the circles. Conceptual clustering is motivated towards making the results of clustering techniques describable and hence produces results that are similar to redescriptions. However, redescription mining imposes a stricter requirement that few objects in a cluster be describable in two ways i.e., by another cluster concept.

5. Related work

Redescription Mining has been proposed and implemented in different ways by many authors. The concept has utilized the important data mining tasks. Initially this was developed by N. Ramakrishnan et al. [3]. They have presented CARTWheels algorithm based on learning decission trees. CARTWheels had a limitation to the depth of the tree and the variables in the formulae. Zaki and Ramakrishnan[7] gave an algorithm to find exact minimal conjunctive descriptions. While Parida and Ramakrishnan [3] gave algorithms for finding exact and approximate monotone redescriptions in CNF and DNF. Parida and Ramakrishnan studied the space of redescriptions, identified impossibility as well as strong possibility of when redescriptions are feasible and presented several scenarios to custom build redescription mining solutions for various scenarios. Gallo and Pauli [1] considered 0-1 dataset for redescriptions. They searched for pairs (α, β) of Boolean formulae such that $\alpha \rightarrow \beta$ and $\beta \rightarrow \alpha$ both hold with high accuracy, provided α and β are different. They have introduced Greedy algorithm and MID, Mining Interesting Descriptors algorithms. These algorithms are heuristic search methods that try to form useful pairs or tuples of queries. Greedy algorithm prunes its search space can lead to suboptimal results because the algorithm can prune initial pairs that could be used to create good descriptions. Also the algorithm prunes the level-wise search tree in a way that can yield to pruning of items that could be used later.

Deept Kumar extended [2] redescription mining framework to another interesting task as *storytelling algorithm*, where the goal is to find the consecutive redescriptions that relate completely disjoint elements. In Gallo, the goal is to find the *m*-tuples of formulae, while in storytelling, the similarities between descriptions next to each other is required to be high, but the similarity between first and last formulae must be zero.

M J. Zaki and N. Ramakrishnan [7] obtained redescriptions by reasoning about sets. They viewed redescriptions as generalizations of association rule mining from finding implications to equivalences; Also, redescriptions as a form of conceptual clustering, where the goal is to identify clusters that afford dual characterizations. And also as a form of constructive induction to build features based on given descriptors that mutually reinforce each other. Zaki and Ramakrishnan presented an algorithm CHARM-L , to mine all minimal redescriptions underlying a dataset using notions of minimal generators of closed itemsets. They also show the use of the algorithms in an interactive context, supporting constraint-based exploration and querying.

6. Conclusions

We have gathered how redescription mining can be developed with three different data mining functionalities. We have considered the merits and demerits of the algorithms mentioned. The enhancements to this work can be the development of a model where we can embed CHARM and Storytelling to get more efficient results.

References

[1] Gallo.A, Meittinen.P and Mannila.H; Finding subgroups having Several Descriptions: Algorithms for Redescription Mining. In Proc SIAM International Conference on Data Mining (SDM), 2008, 334–345.

[2] Kumar. D et al; Algorithms for Storytelling. In KDD, pages 604-610, 2006.

[3] Parida.L and Ramakrishnan.N; Redescription Mining: Structure theory and Algorithms. In AAAI, pages pp. 837-844, 2005.

[4] Ramakrishnan.L et al. Turning CARTWheels: An alternating algorithm for mining rede scriptions. In KDD, pages 266-275, 2004.

[5] Ramakrishnan.N; Zaki.M (2009):Redescription Mining and Applications in Bioinformatics. In Biological Data Mining. CRC Press 2009.

[6] Ryszard.S.Michalski; Knowledge acquisition through conceptual clustering: A theoretical Framework and an algorithm for partitioning data in to conjunctive concepts. pages 218-244
Journal of Policy Analysis and Information Systems, Vol 4 No 3, September 1980.

[7] Zaki.J.M; Ramakrishnan.M (2005):Reasoning about sets using Redescription Mining. In Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining(KDD ’05), pages 364-373, Aug 2005.

[8] Zaki.J.M; Generating non-redundant association rules. In Proceedings of 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining(KDD ’00), pages pp. 34-43, August 2000

[9] Zaki. J. M and Hiao .C.J; CHARM: An efficient algorithm for closed itemset mining. In Proc of 2nd SIAM International Conference on Data Mining, pages pp. 457-473.April 02.

[10] Zaki. J. M and Hsiao .C.J; Efficient algorithms for mining closed itemsets and their lattice structure. IEEE Transactions on Knowledge and Data Engineering, Vol 17, pages pp.462-478, April 2005.

[11] Zaki. J. M and Hsiao .C.J; CHARM: An efficient algorithm for closed association rule mining. 1999 cs-99-10 Rensslaer Library.

[12] Zhao.L; Zaki.J.M and Ramakrishnan.N; BLOSOM: A framework for mining arbitrary Boolean Expressions. In KDD, pages 827-832, 2006.