Turning CARTwheels: An Alternating Algorithm for Mining Redescriptions

Naren Ramakrishnan*, Deept Kumar*, Bud Mishra†, Malcolm Potts#, and Richard F. Helm#
*Department of Computer Science, Virginia Tech, VA 24061
†Courant Institute of Mathematical Sciences, New York University, NY 10003
#Department of Biochemistry, Virginia Tech, VA 24061
Contact: naren@cs.vt.edu

ABSTRACT

We present an unusual algorithm involving classification trees—CARTwheels—where two trees are grown in opposite directions so that they are joined at their leaves. This approach finds application in a new data mining task we formulate, called *redescription mining.* A redescription is a shift-of-vocabulary, or a different way of communicating information about a given subset of data; the goal of redescription mining is to find subsets of data that afford multiple descriptions. We highlight the importance of this problem in domains such as bioinformatics, which exhibit an underlying richness and diversity of data descriptors (e.g., genes can be studied in a variety of ways). CARTwheels exploits the duality between class partitions and path partitions in an induced classification tree to model and mine redescriptions. It helps integrate multiple forms of characterizing datasets, situates the knowledge gained from one dataset in the context of others, and harnesses high-level abstractions for uncovering cryptic and subtle features of data. Algorithm design decisions, implementation details, and experimental results are presented.

Categories and Subject Descriptors: H.2.8 [Database Management]: Database Applications - Data Mining; I.2.6 [Artificial Intelligence]: Learning

General Terms: Algorithms.

Keywords: Classification trees, redescriptions, data mining in biological domains.

1. INTRODUCTION

Classification and regression trees (CART) were among the earliest proposed approaches for pattern classification and data mining [3]. While being powerful in terms of accuracy and efficiency of induction, their results are also simple to understand as they mimic the decision-making logic of human experts. The renewed emphasis on data mining propagated by the KDD community in the 1990s has fueled a resurgence of interest in tree-based methods [7, 9].

In this paper we introduce a new data mining task—*redescription mining*—and also propose a novel tree-based algorithm (CARTwheels) for mining redescriptions. A redescription is a shift-of-vocabulary, or a different way of communicating information about a given subset of data; the goal of redescription mining is to find subsets of data that afford multiple descriptions.

Consider the set of all countries in the world. The elements of this set can be described in various ways, e.g., geographical location, political status, scientific capabilities, and economic prosperity. Such features allow us to define various subsets of the given (universal) set, called *descriptors.* Examples of these are shown in Fig. 1.

Redescriptions are equivalences describing a subset in two ways, for instance:

'Countries with > 200 Nobel prize winners' $\Leftrightarrow$
'Countries with > 150 billionaires'

Both sides of this redescription refer to the singleton set {US}. Such relationships can be mined using techniques from the association rules literature [1], but our view of redescriptions is broader in scope and also includes set-theoretic expressions involving descriptors:

'Countries with defense budget > \$30 billion' $\cap$
'Countries with declared nuclear arsenals' $\Leftrightarrow$
'Permanent members of U.N. Security Council' $-$
'Countries with history of communism'

Here, we have constructed a set intersection on the left and a set difference on the right, from the given descriptors, and obtained a redescription for the 3-element set: {US, UK, France}. A typical approach to mining such patterns would be to first fix the form of the set-theoretic expressions and then search within the space of possible instantiations. The goal of this paper is to present an algorithmic framework that *simultaneously* constructs set-theoretic expressions and searches in the space of possible redescriptions.

Formally, the inputs to redescription mining are the universal set of objects O (e.g., countries) and two sets (X and Y) of subsets of O. The elements of X are the descriptors X_i, and are assumed to form a covering of O ($\bigcup_i X_i = O$). Similarly $\bigcup_i Y_i = O$. The only requirement of a descriptor is that it be a proper subset of O and denote some logical grouping of the underlying objects (for ease of interpretation). The goal of redescription mining is to find equivalence relationships of the form $E \Leftrightarrow F$ that hold at or above a given Jaccard's coefficient θ (i.e., $\frac{|E \cap F|}{|E \cup F|} \geq \theta$), where E and

KDD'04, August 22–25, 2004, Seattle, Washington, USA.

Countries with > 200 Nobel prize winners	=	{										US}
Countries with > 150 billionaires	=	{										US}
Countries with history of communism	=	{	China							Russia		}
Countries with defense budget > $30 billion	=	{		France	Germany			Japan			UK	US}
Permanent members of the U.N. Security Council	=	{	China	France						Russia	UK	US}
Countries with declared nuclear arsenals	=	{	China	France		India	Israel		Pakistan	Russia	UK	US}

Figure 1: Six descriptors defined over a universal set of countries.

F are set-theoretic expressions involving X_i's and Y_i's, respectively. For tractability purposes, some restrictions on the length of the allowable set-theoretic expressions (not their form) is assumed to be provided. Redescription mining hence involves constructive induction (the task of inventing new features) and exhibits traits of both unsupervised and supervised learning. It is unsupervised because it finds conceptual clusters underlying data, and it can be viewed as supervised because clusters defined using descriptors are given meaningful characterizations (in terms of other descriptors).

Why is this problem relevant? We posit that today's high-throughput data-driven sciences are drowning in not just the dimensionality of data, but also in the multitude of descriptors available for characterizing data. Consider gene expression studies using bioinformatics approaches. The universal set of genes in a given organism (O) can be studied in many ways, such as functional categorizations, expression level quantification using microarrays, protein interactions, and biological pathway involvement. Each of these methodologies provides a different vocabulary to define subsets of O (e.g., 'genes localized in cellular compartment nucleus,' 'genes up-expressed two-fold or more in heat stress,' 'genes encoding for proteins that form the Immunoglobin complex,' and 'genes involved in glucose biosynthesis'). While traditionally we would custom-build data mining algorithms to work with each of these vocabularies, redescription mining provides a uniform way to characterize and analyze the results from any of them. In addition, it helps bridge diverse experimental methodologies by uniformly relating subsets across the corresponding vocabularies.

We further argue that redescription mining serves as a fundamental building block of many important steps in the iterative, often unarticulated, knowledge discovery process. A shift of vocabulary allows a given subset of data to be interpretable in a different context, and allows us to harness existing knowledge from this other context. For instance, if we are able to redescribe results from a new stress experiment onto, say, a heat shock experiment studied earlier, we will be able to study the new results in terms of known biological knowledge about heat shock. Chains of redescriptions allow us to relate diverse vocabularies, through important intermediaries.

Even redescriptions that hold with Jaccard's coefficient < 1 find application in many domains. An approximate redescription implies a common meeting ground for two concerted communities of objects. A chain of such approximate redescriptions can effectively relate two subsets that have nothing in common! This is especially useful in *story telling* and *link analysis* applications. A query such as 'what is the relationship between people traveling on Flight 847 and the top 10 wanted list by the FBI?' can be posed in terms of redescription finding.

While related problems have been studied in the data mining community (most notably, conceptual clustering [5, 12], niche finding, and profiling classes [19]), we believe that the above formulation of redescription mining has not been attempted before. Our contributions here are both the introduction of this new data mining problem, as well as a novel tree-based algorithm for mining redescriptions.

2. REDESCRIPTION MINING AS ALTERNATING TREE INDUCTION

We now introduce an approach (CARTwheels) to mining redescriptions that involves growing two trees in opposite directions, so that they are matched at their leaves. The decision conditions in the first tree (say, top) are based on set membership checks in entries from X and the bottom tree is based on membership checks in entries from Y; thus matching of leaves corresponds to a potential redescription. This idea hence uses paths in the classification trees as representations of boolean expressions involving the descriptors.

The CARTwheels algorithm is an alternating algorithm, in that the top tree is initially fixed and the bottom tree is grown to match it. Next, the bottom tree is fixed, and the top tree is re-grown. This process continues, spouting redescriptions along the way, until designated stopping criteria are met.

2.1 Working Example

For ease of illustration, consider the artificial example in Fig. 2 that shows two sets of descriptors for the universal set $O = \{o_1, o_2, o_3, o_4, o_5\}$. Here, the set X corresponds to the set of descriptors $\{X_1, X_2, X_3, X_4\}$ and Y corresponds to $\{Y_1, Y_2, Y_3, Y_4\}$. The cardinalities of X and Y may not be the same in the general case. Further, in a realistic application, the number of descriptors would far exceed the number of objects.

To initialize the CARTwheels alternation, we prepare a traditional dataset for classification tree induction, where the entries correspond to the objects, the boolean features are derived from one of X or Y, and the classes are derived from the other. In the dataset shown in Fig. 3 (left), the features correspond to set membership in entries of Y and each object is assigned a *unique* class, chosen from the X_i's it participates in. We employed a greedy set covering of the objects using the entries of X in order to establish the class labels in Fig. 3 (left). For instance, o_2 belongs to both X_1 and X_3, but the tie is broken in favor of X_1. Notice that in this process, X_3 does not receive any representation in the prepared dataset.

A classification tree can now be grown using any of the impurity measures studied in the literature (e.g., entropy, Gini index, misclassification rate). Fig. 3 (right) depicts a possible tree. The leaves of the tree deterministically predict a class label from X, typically the majority class. At this point, the specific details of how the tree was induced are not important, only that any such tree will induce a partition of the underlying objects. In this case, the tree induces a 3-partition which mirrors the 3-class partition present in the original dataset, but is not exactly the same. The left most path corresponds to the region $Y_3 \cap Y_2$, the right most path

$$\begin{array}{lclllllll} X_1 & = & \{ & & o_2, & o_3 & & & \} \\ X_2 & = & \{ & & & o_3, & o_4 & & \} \\ X_3 & = & \{ & & o_2, & & o_4 & & \} \\ X_4 & = & \{ & o_1, & & & & o_5 & \} \end{array} \qquad \begin{array}{lclllllll} Y_1 & = & \{ & o_1, & o_2, & & & & \} \\ Y_2 & = & \{ & & o_2, & o_3, & o_4 & & \} \\ Y_3 & = & \{ & & & o_3, & & o_5 & \} \\ Y_4 & = & \{ & o_1, & o_2, & & & o_5 & \} \end{array}$$

Figure 2: Example data for illustrating operation of CARTwheels algorithm.

object	Y_1	Y_2	Y_3	Y_4	class
o_1	$\surd$	$\times$	$\times$	$\surd$	X_4
o_2	$\surd$	$\surd$	$\times$	$\surd$	X_1
o_3	$\times$	$\surd$	$\surd$	$\times$	X_1
o_4	$\times$	$\surd$	$\times$	$\times$	X_2
o_5	$\times$	$\times$	$\surd$	$\surd$	X_4

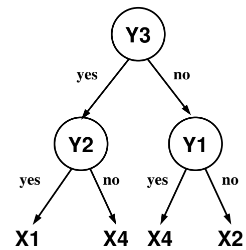

Figure 3: (left) Dataset to initialize CARTwheels algorithm. (right) induced classification tree.

obj.	X_1	X_2	X_3	X_4	class
o_1	$\times$	$\times$	$\times$	$\surd$	$(Y_3 - Y_2) \cup (Y_1 - Y_3)$
o_2	$\surd$	$\times$	$\surd$	$\times$	$(Y_3 - Y_2) \cup (Y_1 - Y_3)$
o_3	$\surd$	$\surd$	$\times$	$\times$	$Y_3 \cap Y_2$
o_4	$\times$	$\surd$	$\surd$	$\times$	$O - Y_3 - Y_1$
o_5	$\times$	$\times$	$\times$	$\surd$	$(Y_3 - Y_2) \cup (Y_1 - Y_3)$

obj.	Y_1	Y_2	Y_3	Y_4	class
o_1	$\surd$	$\times$	$\times$	$\surd$	$(X_3 \cap X_1) \cup (X_4 - X_3)$
o_2	$\surd$	$\surd$	$\times$	$\surd$	$(X_3 \cap X_1) \cup (X_4 - X_3)$
o_3	$\times$	$\surd$	$\surd$	$\times$	$(O - X_3 - X_4)$
o_4	$\times$	$\surd$	$\times$	$\times$	$(X_3 - X_1)$
o_5	$\times$	$\times$	$\surd$	$\surd$	$(X_3 \cap X_1) \cup (X_4 - X_3)$

Figure 4: (left) Dataset for second iteration of CARTwheels algorithm. Notice that class labels are now set-theoretic expressions involving Y_i's. (right) Dataset for third iteration of CARTwheels algorithm.

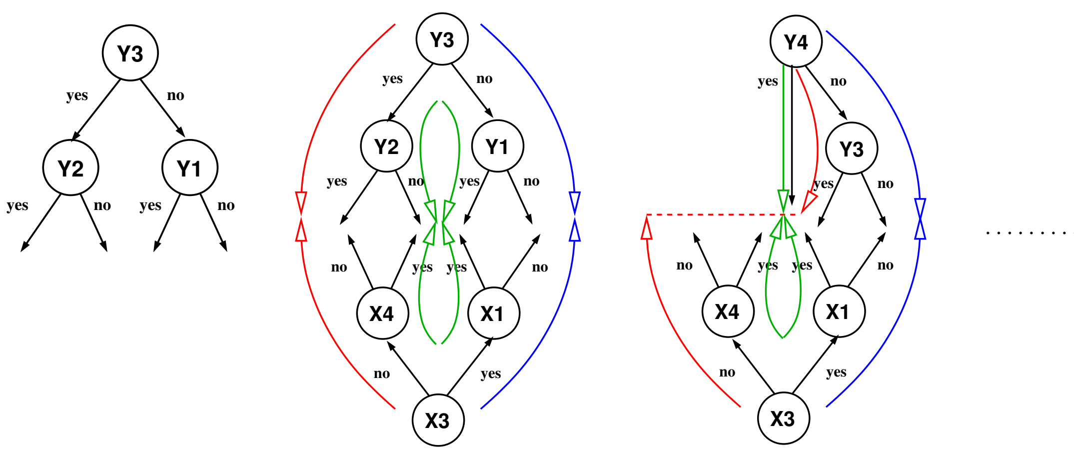

Figure 5: Alternating tree growing in the CARTwheels algorithm. The alternation begins with a tree (first frame) defining set-theoretic expressions to be matched. The bottom tree is then grown to match the top tree (second frame), which is then fixed, and the top tree is re-grown (third frame). Colored arrows indicate the matching paths. Redescriptions corresponding to matching paths at every stage are read off and subjected to evaluation by Jaccard's coefficient.

corresponds to $O - Y_3 - Y_1$, and the union of the two middle paths gives $(Y_3 - Y_2) \cup (Y_1 - Y_3)$. The reader can verify that these regions do not have a one-to-one correspondence with the regions X_1, X_2, and X_4 in the original partition. For instance, only X_2 enjoys such a correspondence, with $O - Y_3 - Y_1$. In 'reading off' a partition from a tree in this manner, a conjunction thus results from a path of length > 1, a disjunction results from multiple paths predicting the same class, with negations corresponding to following the 'no' branch from a given node. This partition is used as the starting point for the alternation (Fig. 5, first frame).

We now prepare a dataset with entries from X as the features and the regions thus formed (involving Y_i's) as the classes, as shown in Fig. 4 (left). Inducing a classification tree from this dataset really corresponds to growing a second tree to match the first tree at the leaves, as depicted in Fig. 5 (second frame). In this case, the second tree also learns a 3-partition and we can evaluate each of these matchings using the Jaccard's measure. This produces three redescriptions:

$$\begin{aligned} (X_3 \cap X_1) \cup (X_4 - X_3) &\Leftrightarrow (Y_3 - Y_2) \cup (Y_1 - Y_3) \\ (X_3 - X_1) &\Leftrightarrow (O - Y_3 - Y_1) \\ (O - X_3 - X_4) &\Leftrightarrow (Y_3 \cap Y_2) \end{aligned}$$

all of which hold at Jaccard's coefficient 1. This need not be the case in general. The bottom tree might be able to match only some paths in the top tree, or the matches might not pass our Jaccard's cutoff. This process is then continued, now with Y_i's as features and the partitions derived from the bottom tree as classes (see right of Fig. 4). The new matchings yield the redescriptions:

$$\begin{aligned} (X_3 \cap X_1) \cup (X_4 - X_3) &\Leftrightarrow Y_4 \\ (O - X_3 - X_4) &\Leftrightarrow (Y_3 - Y_4) \\ (X_3 - X_1) &\Leftrightarrow (O - Y_3 - Y_4) \end{aligned}$$

which, fortuitously, also have a Jaccard's coefficient of 1. Notice that, this time, the root decision node that has been picked is Y_4 (see third frame of Fig. 5) and the tree actually resembles a *decision list* (a tree where every internal node has a leaf on its 'yes' branch). The alternation can be continued (see Sec. 2.4 for ways to configure the search).

If we limit the size of the trees at every iteration, it is easy to see that the set-expressions constructed cannot get arbitrarily long. In our running example, we use a depth limit of 2 so that all expressions on either side of a mined redescription can involve at most three descriptors. The longest expressions result from unions of two paths involving different subtrees.

2.2 Why does CARTwheels work?

The use of trees to mine one-directional implications (rules) is well understood and is the idea behind algorithms such as C4.5 [15]. In CARTwheels, we exploit the duality between *class partitions* and *path partitions* to posit the stronger notion of equivalence. In fact, if a tree reduces the entropy to zero, it is clear that there must be a one-to-one correspondence between its path partitions and class partitions, which are really path partitions from the other tree. Keep in mind that different paths are union-ed when they predict the same class, and this property is crucial to establishing the duality.

The search for redescriptions in CARTwheels can be viewed as a problem of identifying (and creating) correlated random variables. We present a simple analysis in the case of one-level tree (the extension to more levels is beyond the scope of this paper). A descriptor, e.g., D, can be considered to be a discrete random variable that takes on values from O. Every object in D occurs with probability $\frac{1}{|D|}$ and other objects occur with probability zero, to yield total probability mass of 1. Notice that this makes the self entropy of such a random variable to be the logarithm of the size of the descriptor. Now consider running a CARTwheels alternation with a depth limit of 1 for the classification trees. Mining a redescription with Jaccard's coefficient of 1 is equivalent to identifying a random variable D' whose *entropy distance* from D is zero. The entropy distance is given by:

$$H(D, D') - I(D; D')$$

where $H(D, D')$ is the joint entropy function of $\{D, D'\}$ and I qualifies the mutual information, in turn given by:

$$I(D; D') = H(D) - H(D|D')$$

where $H(D)$ is the self-entropy of D and $H(D|D')$ is the conditional entropy of D given D'. In other words, the average reduction in uncertainty about D due to knowing D' is exactly the self entropy of D, causing an entropy distance of 0. Entropy distance is a true distance measure, unlike measures such as the Kullback-Leibler (KL) divergence. Smaller values of entropy distance hence imply higher values of Jaccard's coefficient.

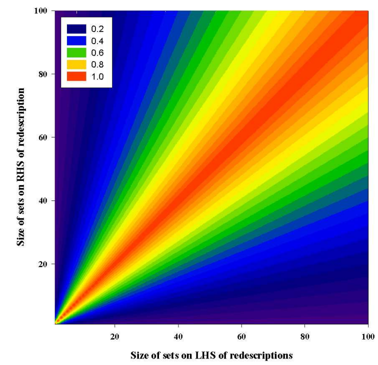

Figure 6: Contour plot depicting best attainable Jaccard's coefficient, for different set sizes.

2.3 Configuring Alternations in CARTwheels

CARTwheels provides a general framework to explore a space of redescriptions; to configure its alternation, there are several issues to be considered.

We will begin by observing that the continuation of CARTwheels alternation, after mining a redescription, is really an attempt to explore and stay within a relatively small region of high Jaccard's coefficient. Fig. 6 shows an idealized scenario where descriptors (or expressions derived from them) occur in all possible sizes, with the best possible overlaps. In a realistic dataset, the regions of high Jaccard's coefficient might be disjoint, and a good exploration policy must try to visit all potential regions.

In contrast to traditional classification tree induction which is motivated at *reducing* entropy, CARTwheels must actually *maintain* entropy in some form, since impurity drives exploration. However, if the impurity in the underlying datasets remains constant, some redescriptions are bound to be found over and over again. The tradeoff here is clearly between ex-

Input: objects O, descriptor sets $\{X_i\}$, $\{Y_i\}$
Output: redescriptions $\mathcal{R}$
Parameters:
    θ (Jaccard's coefficient),
    d (depth of trees),
    ρ (# of class participations allowed/descriptor), and
    η (max. # of consecutive unsuccessful alternations).
Initialization:
    set answer set $\mathcal{R} = \{\}$
    set class participation counts for all $\{X_i\}$, $\{Y_i\} = 0$
    set feature set $\mathcal{F} = \{Y_i\}$
    set classes $\mathcal{C} = \{X_i\}$
    set dataset $\mathcal{D}$ = construct_dataset(O, $\mathcal{F}$, $\mathcal{C}$)
    set tree t = construct_tree($\mathcal{D}$, d)
    if (all leaves in t have same class $c \in \mathcal{C}$)
        set l = random leaf in t having non-zero entropy
        impurify(t,l)
    $\mathcal{C}$ = paths_to_classes(t)
    flag = false; count = 0
Alternation:
    $\mathcal{G} = \{X_i\}$
    while (count $< \eta$)
        $\mathcal{F} = \mathcal{G}$
        if (flag = false)
            $\{X_i\} = \mathcal{G}$; $\mathcal{G} = \{Y_i\}$
        else
            $\{Y_i\} = \mathcal{G}$; $\mathcal{G} = \{X_i\}$
        endif
        $\mathcal{D}$ = construct_dataset(O,$\mathcal{F}$,$\mathcal{C}$)
        t = construct_tree($\mathcal{D}$,d)
        if (all leaves in t have same class $c \in \mathcal{C}$)
            set l = random leaf in t having non-zero entropy
            impurify(t,l)
        endif
        $\mathcal{R}_{new}$ = eval(t,θ)
        if ($\mathcal{R}_{new} = \{\}$)
            count = count + 1
        else
            count = 0
            foreach $c \in \mathcal{C}$
                if c is involved in some $r \in R_{new}$
                  $\mathcal{H}$ = descriptors(c)
                  foreach descriptor $g \in \mathcal{G} \cap \mathcal{H}$
                    increase g's class participation count
                    if g's class participation count $> \rho$
                    remove g from $\mathcal{G}$
                  endfor
                endif
            endfor
        end if
        $\mathcal{R} = \mathcal{R} \cup \mathcal{R}_{new}$; flag = not(flag)
        $\mathcal{C}$ = paths_to_classes(t)
    end while

Table 1: CARTwheels algorithmic framework.

ploration and redundancy: to support sufficient exploration, we must accept redundancy, and conversely if we desire to reduce redundancy, we must settle for insufficient coverage of the redescription space. This tradeoff suggests that a tunable parameter for CARTwheels alternation is the number of times that a descriptor is allowed to participate in redescriptions.

2.4 The CARTwheels Algorithmic Framework

Table 1 describes the CARTwheels algorithmic framework in detail. The outline follows the example shown previously: *construct_dataset* prepares a dataset suitable for CART induction as in Fig. 3; *construct_tree* creates the decision tree of depth d; and *paths_to_classes* reads expressions off an induced tree, to be used as classes in the next step for each object in O. Notice the use of an *impurify* function in both the initialization and the alternation steps, which typically assigns the second-best class label to the chosen leaf l. Additional impurification steps, to aid exploration, are included in our implementations of *construct_tree* (e.g., we do not always branch on the attribute with the best entropy gain and sometimes perform randomized moves at the root level).

The *eval* function returns redescriptions satisfying the Jaccard's threshold θ. Our implementation of *eval* requires redescriptions to hold in both the mined and complementary forms, e.g., for the equivalence $E_1 \cup E_2 \Leftrightarrow F$ to be considered as a redescription, it must hold with Jaccard's coefficient at least θ, as must its complement: $\neg E_1 \cap \neg E_2 \Leftrightarrow \neg F$. This ensures that every redescription truly induces a partition of $O \times O$ space. *descriptors* is a function that analyzes a set-theoretic expression and returns the set of descriptors participating in it.

The important tunable parameter in Table 1 is ρ, controlling the 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 a class, and when this reaches ρ, the descriptor is removed from consideration. The parameter η specifies the maximum number of alternations that CARTwheels can go through without mining any redescriptions.

2.5 Assessing Significance of Mined Redescriptions

There are many ways to assess significance of redescriptions mined by CARTwheels. They vary in their formulation of the null hypothesis. For instance, given a redescription $X \Leftrightarrow Y$ with Jaccard's coefficient θ we can ask 'how likely is it that two descriptor expressions of size $|X|$ and $|Y|$ have θ as their Jaccard's coefficient?' or 'how significant is it that expressions having the same syntactic bias as X and Y have θ as their Jaccard's coefficient?' The first approach focuses on the sizes of the descriptor expressions whereas the second is concerned with the way expressions are constructed, and must inherently utilize the distribution of descriptor sizes (and maybe second order information, such as commonality or differences). We adopt the first approach in this paper.

Specifically, we assess if the Jaccard's coefficient (θ) can happen by chance if we had chosen sets X and Y randomly from the available universal set O, keeping $|X|$ and $|Y|$ fixed. This yields a simple statistical test giving a p-value based on the distribution of set overlaps for the given set sizes (details omitted for space considerations). Keep in mind that one way to get a strong p-value would be to have very small sizes for X and Y (which in turn, make the achievement of a respectable θ difficult). On the other hand, if X and Y are large, the ease with which they could overlap increases, and hence even high Jaccard coefficients might not correspond to a strong p-value. Therefore, for interpretation purposes, it is important to not think of intersection size as a surrogate for significance of redescriptions. In the experiments reported here, we have found statistically significant redescriptions involving as few as 1 object to as large as 80 objects.

2.6 Implementation Details

CARTwheels is implemented in C++ atop a Postgres database providing access to the descriptors. We use an AD-tree data structure [13] for fast counting purposes and estimation of entropy (this is distinct from the classification tree

Table 2: Summary of universal sets and descriptors.

	G_1	G_2	G_3
# stresses	5	7	7
# expts	7	9	9
# ORFs	74	332	171
GO (biological process) descriptors	210	479	382
GO (cellular component) descriptors	42	112	97
GO (molecular function) descriptors	126	298	204
Expression level range descriptors	224	373	344
k-means clusters	70	270	0
Histone expression range descriptors	152	168	162
# descriptors	824	1700	1189

that combines the descriptors). The AD-tree provides access to the distributions of 'class labels' for every combination of 'features' and, since the definition of features and class labels change at every iteration, is rebuilt continually. Notice that the data structure is expected to provide both the sizes of descriptors as well as their negations (when we follow the 'no' branch) and hence, the depth of the AD-tree is set to just greater than the allowable depth of the classification trees. The CARTwheels algorithm consults the AD-tree whenever it must make a choice of a decision node (except when its move is exploratory). After evaluating matchings, set-expressions read off the trees are subjected to tabular minimization, in order to arrive at a canonical form.

The implementation allows for configuring the space of redescriptions that are explored. The depth limit for the top and bottom trees can be individually specified, and we can also preferentially include or exclude certain types of expressions in mined redescriptions. For instance, syntactic constraints on redescriptions (e.g., only conjunctions are allowed) can be incorporated as biases in the tree construction phase of CARTwheels.

3. APPLICATIONS IN BIOINFORMATICS

We now present an application of CARTwheels to studying gene expression datasets from microarray experiments conducted on the budding yeast *Saccharomyces cerevisiae*. Bioinformatics is fertile ground for application of CARTwheels and *S. cerevisiae* is arguably the most well studied (and documented) model organism through bioinformatics techniques. Practically every experimental methodology applied toward yeast can be viewed as a way to define descriptors. Even the results of other data analysis/mining algorithms can be used as a source of descriptors! The underlying universal set of objects could be initialized to the set of genes, proteins, or processes, in *S. cerevisiae*. CARTwheels hence brings many computational and experimental technologies to bear upon redescription mining. It supports the capture of both similarities and distinctions among descriptors derived from these diverse sources.

3.1 Datasets

The redescription process begins by defining the universal set of genes (or open reading frames, ORFs) G, which is dependent on our biological goals. Here, we are interested in characterizing similarities and differences in yeast gene expression behavior across related families of stresses. Gasch et al. ([8]) is an important source for such a study since it provides results from more than 170 comparisons, across a variety of environmental stresses. We use three different universal sets, to illustrate diverse ways of using the CARTwheels framework:

G_1: the set of ORFs that show significant change in expression (more than 1-fold up- or down-regulation) in some time point in each of the five stresses from (heat shock from 25°C to 37°C, hyper-osmotic shock, hypo-osmotic shock, H_2O_2 exposure, and mild heat shock at variable osmolarity).

G_2: the set of ORFs that show more than 4-fold up- or down-regulation change in expression in some time point in each of the seven stresses from (heat shock from 25°C to 37°C, hyper-osmotic shock, hypo-osmotic shock, H_2O_2 exposure, mild heat shock at variable osmolarity, heat shock from 37°C to 25°C, and heat shock from 29°C to 33°C). Notice that two additional stresses are included, from how G_1 was constructed.

G_3: the set of ORFs more than 4-fold up- or down-regulation change in expression in some time point in each of the seven stresses in G_2 and that do not belong to the set of ESR (Environmental Sress Response) genes as characterized by Gasch et al. ([8]). The ESR dataset (comprising 868 ORFs) constitute a characterization of yeast ORFs that show a marked uniformity of expression across diverse stresses, and hence have been excluded by many researchers in their analyses – see for instance, ([17]).

The choice of the universal set can be viewed as a conditioning context and must be kept in mind when interpreting any mined redescriptions. It can be viewed as an implicit descriptor occurring on both sides of every mined redescription, e.g., $E \Leftrightarrow F$ in G_1 can be viewed as $E \cap G_1 \Leftrightarrow F \cap G_1$.

3.2 Descriptor Definition

We defined descriptors for the genes in the chosen universal sets in a variety of ways. One class of descriptors was derived from categories in the GO (Gene Ontology) biological process, GO cellular component, and GO molecular function taxonomies, that have representation among the chosen genes. The microarray results from the stresses of Gasch et al. (relevant to each universal set) were bucketed to yield range descriptors of the form 'expression level ∈ [%x, 0] in time point %y of stress experiment %z' (for negative %x) and 'expression level ∈ [0, %x] in time point %y of stress experiment %z' (for positive %x). Notice that we are not constrained to pick descriptors from only the stresses used to define the universal set, although we have made that choice here. Further, k-means clustering was performed using the Genesis software suite ([18]) on each of the stresses individually, with a setting of 10 clusters for G_1 and 10 and 20 clusters for G_2. No descriptors based on k-means clustering were defined for G_3. Since heat shock and mild heat shock at variable osmolarity are actually pairs of experiments, this step yields $(5+2) \times 10 = 70$ (for G_1) and $(7+2) \times (10 + 20) = 270$ (for G_2) descriptors, depicting clusters of genes with similar temporal profiles. It must be kept in mind that each of these experiments in turn comprise of multiple time points, different for each stress. Finally, we included microarray results from a histone depletion experiment conducted by Wyrick et al. ([20]) and created range descriptors similar to the Gasch stresses; this is to allow us to relate the effect of histone depletion to that of environmental stresses.

Table 2 summarizes the number of descriptors of each type defined for each of the universal sets, and provides count statistics. Fig. 8 presents frequency plots for the sizes of the descriptors in each of the universal sets. As expected, a majority of descriptors in each case have very few number of ORFs.

3.3 CARTwheels Configuration

To invoke CARTwheels for a particular universal set, we initialized X to be all descriptors derived from the Gasch et al. dataset (which includes the range descriptors as well as the k-means clusters). This ensures that all redescriptions will involve some aspect of the Gasch et al. experiment and prevents the possibility of, say, mining a redescription between two GO taxonomies. Y was initialized to the set of all descriptors; thus, there is some overlap between X and Y. In order to prevent obvious redescriptions arising from this overlap, the algorithm was precluded from utilizing descriptors in one tree if they are already present in the other tree.

We employed a Jaccard's threshold θ of 0.5 and a depth-limit $d \leq 2$ in both the top and bottom tree induction alternations. The limit on the number of allowable alternations η is set to 10, and ρ was varied from 1 to 6. Redescriptions mined by CARTwheels are subjected to a 'tightening' step, akin to rule pruning in packages like C4.5 ([15]). This might involve attempting to drop terms from both sides of the redescription, or restricting range descriptors (if they occur in the redescription), and determining whether this causes significant degradation of Jaccard's coefficient. If no degradation is observed, then the redescription can be tightened. A p-value cutoff of 0.001 for significance of redescriptions was utilized in this paper. We first describe the qualitative nature of biological results obtained through redescription and then assess the exploratory behavior of CARTwheels.

3.4 Example Redescriptions

Seven key mined redescriptions (R1–R7) are depicted in Fig. 7. R1-R3 are defined over universal set G_1, R4-R6 over G_2, and R7 over G_3. These redescriptions were selected for both their biological interest as well as for their feature construction novelties. The proteins encoded by genes in a redescription may interact with one another or, with other proteins not included in the redescription. Such analyses make it possible to uncover cryptic and subtle features of gene expression and regulation.

R1 is a redescription where both sides involve descriptors from gene expression bucketing. It relates negatively expressed ORFs in the histone depletion experiment with similarly expressed ORFs in a Gasch comparison (heat shock). This redescription can be read as 'of the 74 ORFs in the first universal set, the ORFs negatively expressed in the histone depletion experiment (6 hours) are also those that are negatively expressed two-fold or more in the heat shock (10 minutes) experiment.' This redescription holds with a Jaccard's coefficient of 0.78. Since each side contains a single descriptor, this redescription does not present any set construction. R1 involves 7 ORFs, three of which are reported to be regulated by similar mechanisms, according to the work of Segal et al. ([17]). These ORFs comprise functions related to metabolism, catalytic activity, and are located in the cytoplasm. The Pearson coefficients for these ORFs in the histone depletion experiments match very strongly, showcasing the use of redescription in identifying a concerted set of ORFs.

R2 relates a k-means cluster to a set difference of two *related* GO cellular component categories. While the 8 ORFs in R2 appear to be part of different response pathways, 5 of these 8 ORFs are similarly regulated according to the work of Segal et al.; these genes relate to the cellular hyperorganization and membrane dynamics in the regulation network.

R3 is actually a triangle of redescription relationships that illustrates the power of CARTwheels. Three different experimental comparisons are involved in this circular chain of redescriptions, with 10 ORFs being implicated in all three descriptors. From a biological standpoint, this is a very interesting result – the common genes indicate concerted participation across stress conditions; whereas the genes participating in, say, two of the descriptors, but not the third, suggest a careful diversification of functionality. 6 of the 10 ORFs are related to cell growth and maintenance. 5 of the 10 ORFs have binding motifs related to the DNA binding protein REB1. The importance of phosphate and ribosomes appears to be salient in this redescription. It is important to note that the circularity of R3 is not directly mined by CARTwheels, but inferred post-hoc from a linear chain.

The theme in R4 is ribosome assembly/biogenesis and RNA processing. R4 is a linear chain comprising two redescriptions, and uses a GO descriptor as an intermediary between two expression-based descriptors. It is also interesting that this redescription involves a set of 45 ORFs!

R5 is an even longer chain involving 41 ORFs that are common to all descriptors. Notice the rather complicated set construct involving a disjunction of a conjunction and a difference, involving three different GO biological categories. Incidentally, this is the most complicated set expression representable in a 2-level tree.

R6 is a relationship between two k-means clusters, between heat shock stresses. The ORFs participating in R6 demonstrate a clear focus on sugar or sugar phosphate metabolism.

R7 is a redescription relating a disjunction of descriptors to a GO cellular component category. It is also our first example of a redescription where a rectangular region is mined in a 2D space involving two different experimental comparisons. Usually such a region would require a 4-level tree, but since it is bounded by the extremal values specific to each experiment, it can be captured by a conjunction of merely two descriptors.

3.5 Effect of ρ and η

If we view the alternation process as one of information retrieval, we can adapt traditional precision and recall metrics for algorithm assessment. Precision here refers to the number of unique redescriptions as a fraction of the total number of redescriptions mined. Recall refers to the number of unique redescriptions as a fraction of the total number of redescriptions possible. Unfortunately, the latter metric is nearly impossible to attain, even for our depth limit of 2. For even the smallest universal set considered here, the size of the space of possible redescriptions is $O(10^{14})$! Our approach hence is to track precision and the total number of redescriptions, across various values of ρ.

Fig. 9 shows the monotonic decrease of precision as ρ is increased, and Fig. 10 depicts the steady increase in the total number of redescriptions mined. These graphs indicate that the tradeoff between redundancy and exploration holds

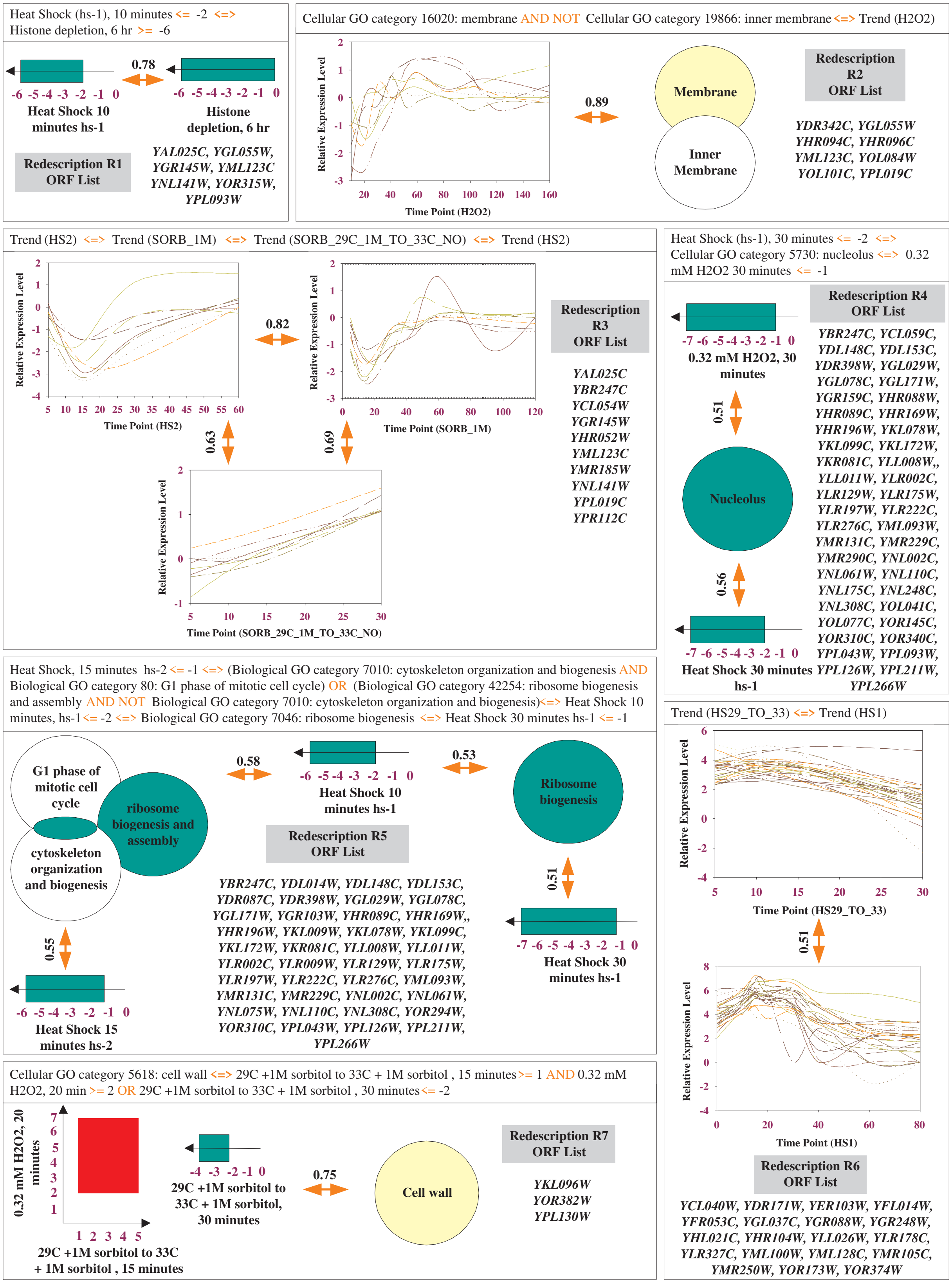

Figure 7: Seven redescriptions mined from gene expression studies on *Saccharomyces cerevisiae*. Each box gives a readable statement of the redescription, presents it in graphical form, and identifies the ORFs conforming to the redescription. R1-R3 are defined over universal set G_1, R4-R6 over G_2 and R7 over G_3. The Jaccard's coefficient is displayed over the redescription arrow. Notice that some redescriptions (e.g., R7) involve few ORFs, whereas others such as R5 involve larger numbers.

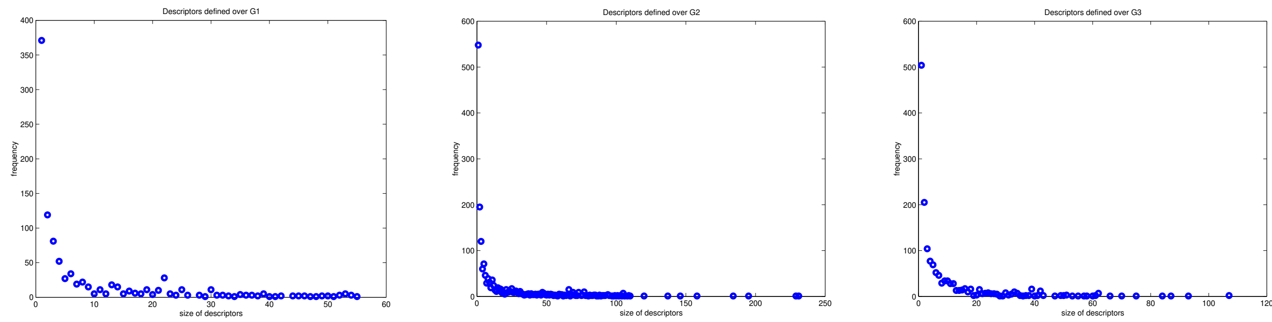

Figure 8: Frequency plot of descriptor sizes for universal set G_1, G_2, and G_3, respectively.

across all the datasets considered here. A formal characterization is underway.

The effect of η parameter, that controls the number of unsuccessful consecutive alternations, is as expected: increasing η results in a greater number of (total and unique) redescriptions mined (not shwon for space considerations).

4. DISCUSSION

This paper is a first exploration into the formulation of the redescription mining problem and has presented an approach for mining redescriptions automatically. Redescriptions can be thought of as generalizations of one-directional implications (e.g., association rules [1], rules in ILP [14]), where one descriptor is required to be a proper subset of the other. This generalization coupled with the automatic identification of set-theoretic constructions makes CARTwheels a very powerful approach to mining (approximate) equivalence relations. We have demonstrated the effectiveness of CARTwheels in a domain that exhibits a richness of descriptors, and shown how it captures patterns involving small as well as large sets of objects.

The work presented here has close connections with ideas pursued in the schema matching [16], clustering categorical data [6], and model management [2] literature. The relationships considered in schema matching research are primarily of the foreign key nature or otherwise operate at the instance level, whereas we consider more complex set-theoretic relationships. Clustering categorical data focuses on defining similarity measures in non-metric spaces and this research can be fruitfully integrated with our work. However, notice that we are not merely clustering data but also imposing describability constraints. Model management is a framework that recognizes the complex inter-relationships that would exist in multi-database enterprises and provides union, intersection, and difference operators for reconciliation, integration, and migration purposes. The relationships here are assumed to be user provided, and the emphasis is on actually 'executing a redescription.' CARTwheels can thus be usefully employed here as a driver for determining what these relationships should be.

We now outline some directions for future research. The connection between Jaccard's coefficient and algorithmic driver parameters (such as entropy) deserves further study. Other ways of evaluating redescriptions [10, 11] are also pertinent here (e.g., Dice coefficient) and some of these could support more efficient tree-based algorithms than the Jaccard's coefficient. Ideally, an evaluation metric would obey some closure properties in the space of redescriptions, which can be used to configure an exploration strategy. In addition, it is preferable that an evaluation metric lend itself to the design of a statistical test of significance for redescriptions.

Thus far, we have assumed a 'flat' organization of the given descriptors and do not recognize any structural relationships between them. However, some descriptor vocabularies (e.g., derived from GO) enjoy a hierarchical structure, which can be exploited by the mining algorithm. Specialized redescription algorithms can thus be designed for targeted descriptor families.

There are various other formulations of the redescription mining problem, in particular the question of identifying a *generating set* of redescriptions is open. This will avoid having to find all redescriptions and instead allow us to exploit the algebraic structure of descriptors, akin to the strategy pursued by Zaki for mining a non-redundant set of association rules [21].

There is an intrinsic limit to a dataset's potential to reveal redescriptions, which can be studied through statistical analysis of set size distributions and estimates of overlap potential. Of particular interest here is qualifying the 'expected' results from a CARTwheels alternation before actually performing the alternation, using notions such as the *entropy rate* of the stochastic process underlying the alternation [4].

Our current focus is on using redescriptions to automatically span multiple levels of abstraction (e.g., gene subsets $\rightarrow$ pathways $\rightarrow$ biological processes). This would firmly establish the importance of redescription in bridging the diverse levels at which information is created and characterized.

Acknowledgements

NR and DK are supported by NSF grants EIA-9984317 and EIA-0103660. The work of MP and RH is supported through the Multidisciplinary University Research Initiative (MURI) program of the Department of Defense (Biomimetic Cell and Tissue Stasis; N00014-01-1-0852) and the Metabolic Engineering for Cellular Stasis program of DARPA (N00173-98-1-G005-P00004 and N00173-02-1-G016). BM is supported by grants from NSF's ITR, QuBIC, and SGER programs, DARPA, the US Air Force (AFRL), National Institutes of Health (NIH), and New York State Office of Science, Technology & Academic Research (NYSTAR). We thank Chris Bailey-Kellogg and T.M. Murali for useful comments.

5. REFERENCES

[1] R. Agrawal and R. Srikant. Fast Algorithms for Mining Association Rules in Large Databases. In *Proceedings of VLDB'94*, pages 487–499, Sep 1994.

[2] P.A. Bernstein, R. Pottinger, and A.Y. Halevy. A Vision for Management of Complex Models. *SIGMOD*

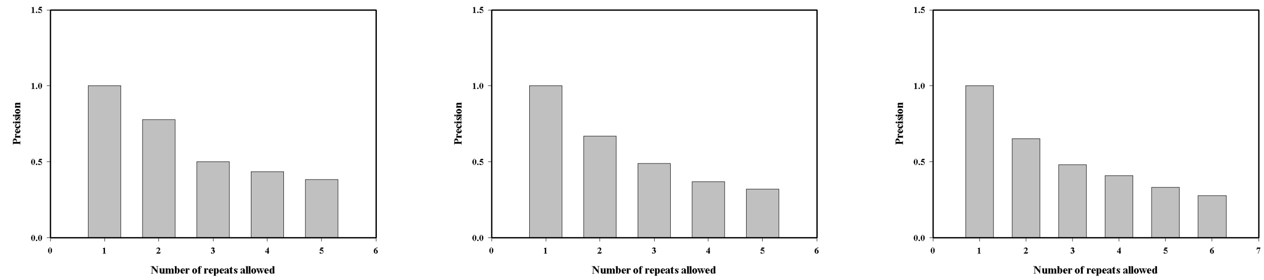

Figure 9: Precision for redescriptions mined vs. ρ for universal set G_1, G_2, and G_3, respectively.

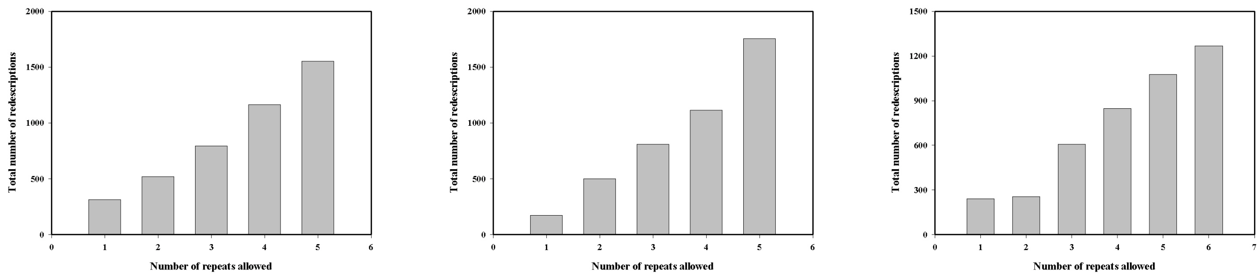

Figure 10: Total number of redescriptions mined vs. ρ for universal set G_1, G_2, and G_3, respectively.

Record, Vol. 29(4):pages 55–63, Dec 2000.

[3] L. Breiman, J.H. Friedman, R.A. Olshen, and C.J. Stone. *Classification and Regression Trees*. Chapman and Hall/CRC, 1984.

[4] T.M. Cover and J.A. Thomas. *Elements of Information Theory*. John Wiley and Sons, 1991.

[5] D.H. Fisher. Knowledge Acquisition via Incremental Conceptual Clustering. *Machine Learning*, Vol. 2(2):pages 139–172, 1987.

[6] V. Ganti, J. Gehrke, and R. Ramakrishnan. CACTUS: Clustering Categorical Data using Summaries. In *Proceedings of KDD'99*, pages 73–83, Aug 1999.

[7] V. Ganti, J. Gehrke, and R. Ramakrishnan. Mining Very Large Databases. *IEEE Computer*, Vol. 32(8):pages 38–45, Aug 1999.

[8] A.P. Gasch, P.T. Spellman, C.M. Kao, O. Carmel-Harel, M.B. Eisen, G. Storz, D. Botstein, and P.O. Brown. Genomic Expression Programs in the Response of Yeast Cells to Environmental Changes. *Molecular Biology of the Cell*, Vol. 11:pages 4241–4257, 2000.

[9] J. Gehrke, R. Ramakrishnan, and V. Ganti. RainForest: A Framework for Fast Decision Tree Construction of Large Datasets. *Data Mining and Knowledge Discovery*, Vol. 4(2/3):pages 127–162, July 2000.

[10] J.C. Gower and P. Legendre. Metric and Euclidean Properties of Dissimilarity Coefficients. *Journal of Classification*, Vol. 3:pages 5–48, 1986.

[11] W.P. Jones and G.W. Furnas. Pictures of Relevance: A Geometric Analysis of Similarity Measures. *Journal of the American Society for Information Science*, Vol. 38(6):pages 420–442, 1987.

[12] R.S. Michalski. Knowledge Acquisition through Conceptual Clustering: A Theoretical Framework and Algorithm for Partitioning Data into Conjunctive Concepts. *International Journal of Policy Analysis and Information Systems*, Vol. 4:pages 219–243, 1980.

[13] A.W. Moore and M.S. Lee. Cached Sufficient Statistics for Efficient Machine Learning with Large Datasets. *JAIR*, Vol. 8:pages 67–91, 1998.

[14] S. Muggleton. Scientific Knowledge Discovery using Inductive Logic Programming. *CACM*, Vol. 42(11):pages 42–46, Nov 1999.

[15] J.R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993.

[16] E. Rahm and P.A. Bernstein. A Survey of Approaches to Automatic Schema Matching. *VLDB Journal*, Vol. 10(4):pages 334–350, 2001.

[17] E. Segal, M. Shapira, A. Regev, D. Pe'er, D. Botstein, D. Koller, and N. Friedman. Module Networks: Identifying Regulatory Modules and their Condition-Specific Regulators from Gene Expression Data. *Nature Genetics*, Vol. 34(2):pages 166–176, 2003.

[18] A. Sturn, J. Quackenbush, and Z. Trajanoski. Genesis: Cluster Analysis of Microarray Data. *Bioinformatics*, Vol. 18(1):pages 207–208, 2002.

[19] R.E. Valdes-Perez, V. Pericliev, and F. Pereira. Concise, Intelligible, and Approximate Profiling of Multiple Classes. *International Journal of Human Computer Studies*, Vol. 53(3):pages 411–436, 2000.

[20] J.J. Wyrick, F.C. Holstege, E.G. Jennings, H.C. Causton, D. Shore, M. Grunstein, E.S. Lander, and R.A. Young. Chromosomal Landscape of Nucleosome-Dependent Gene Expression and Silencing in Yeast. *Nature*, Vol. 402:pages 418–421, 1999.

[21] M. Zaki. Generating Non-Redundant Association Rules. In *Proceedings of KDD'00*, pages 34–43, 2000.