Closed Set Mining of Biological Data

John L. Pfaltz *
Dept. of Computer Science
Univ. of Virginia
Charlottesville, VA 22904-4740
jlp@virginia.edu

Christopher M. Taylor
Dept. of Computer Science
Univ. of Virginia
Charlottesville, VA 22904-4740
cmt5n@virginia.edu

ABSTRACT

We present a closed set data mining paradigm which is particularly effective for uncovering the kind of deterministic, causal dependencies that characterize much of basic science. While closed sets have been used before in frequent set data mining, we believe this is the first algorithm to incrementally combine closed sets one at a time to actually mine associations.

Keywords

knowledge discovery, concept analysis, closure, incremental, logical implications

1. INTRODUCTION

By data mining we mean the discovery of associations $A \Rightarrow B$ in large data sets.[1] The association may be deterministic, that is the occurrence of A *always* implies B; or it may be probabilistic, that is A *often* implies B. The latter describes market basket analysis which extracts common item associations from point-of-sale data streams. As is well known, the first step is to find sets of items that frequently occur together so as to ensure statistical significance. This kind of frequent set data mining became practical with the *apriori* algorithm [2]. Its success has provided impetus to the entire field.

But, frequent set data mining has two shortcomings. Its principal foundation is the observation that if a set X is frequent, then every subset $Y \subseteq X$ must be frequent as well. Consequently, if X is a frequent set of n items, or attributes, or behaviors, then it has 2^n frequent subsets.

Thus, the first shortcoming of frequent set mining arises when large frequent sets of size n are possible. This is often the case when interpreting biological or medical data sets. For example, [3] cites an instance of an important association of 23 attributes. The requirement of exponential storage, in their case $2^{23} \approx 8.3M$ frequent subsets, made frequent set mining impossible.

The second weakness of frequent set mining is evident when it generates too many associations. Suppose the set $abcde$, a compact notation for $\{a, b, c, d, e\}$, is frequent; and using the rule [1]

$$support(A \cup B)/support(A) \geq \gamma$$

we determine that $A \Rightarrow B$ when $A = ab$ and $B = cde$. Then for many values of γ we would have $ab \Rightarrow c$, $ab \Rightarrow d$, $abc \Rightarrow de$, $abd \Rightarrow ce$, $abe \Rightarrow cd$, $abcd \Rightarrow e$, and so forth, for upto 15 redundant associations that are subsumed by $ab \Rightarrow cde$.[2]

When confronted with these shortcomings of frequent set mining, Godin and Missaoui [11], Pasquier *et al.* [15], Zaki [20; 19] and Brossette and Sprague [3] have all turned to the use of closed sets to resolve their problems. So shall we. But, we will use closed sets, not frequent sets, as the basic mining tool. This will lead to a practical approach for discovering the kind of deterministic associations, $A \Rightarrow B$, that are often found in science where A always implies B, or alternatively A causes B. As a bonus, we obtain a process that encourages the incremental addition of data, as is the case with on-going biological experiments. We believe that the algorithm we present in Section 2.2 will be as basic to closed set mining as *apriori* has been to frequent set mining. In Section 2 we lay some background about closure systems, and in Section 3 we illustrate results obtained from this form of data mining. Since our closed set method is a form of discrete, deterministic data mining (DDDM) which presumes perfect causal dependence, we must consider the possibility of error. This we do in Section 4.

Before continuing, let us point out that closed set mining also has its shortcomings. Frequent set and closed set mining are complementary, not competitive, systems.

When the relation is sparse, as in transactional analysis, neither of the shortcomings of frequent set mining that we cited above are of concern. It is clearly the method of choice. Closed set mining should be considered when the relation is dense. And, it is best when the associations are deterministic, or nearly so, as is the case with certain kinds of

*Research supported in part by DOE grant DE-FG05-95ER25254.

[1] The term "data mining" has come to mean the discovery of almost any significant pattern in a data set, such as finding clusters of similar items [5]. All these have validity and many have their own extensive literature.
We should also note here that in the abstract, data mining is conducted over a binary relation R. In practice, it may be a mapping $R : T \rightarrow I$ of a set T of "transactions" into a set I of "items" transacted. Or we could regard $R : O \rightarrow A$ as a collection of observations of a set O of objects, each exhibiting some of the attributes, or properties, of A. In this paper we prefer the latter interpretation, and so will speak of attributes, objects, and observations.

[2] Setting the confidence γ sufficiently high to avoid this multiplicity of associations in one case can cause other important associations to be missed altogether.

biological and scientific data. As the associations become more and more probabilistic, the efficacy of closed set mining decreases.

In those applications when neither method by itself is optimal there can be hybrid systems. The authors cited above used closed sets to improve a frequent set approach. In Section 4 we indicate how frequency of support can be used to improve closed set mining.

2. CLOSURE SYSTEMS

We use the concept of closure to extract the desired logical implications from a relation, R. A closure operator φ is one that satisfies the three basic closure axioms: $X \subseteq X.\varphi$; $X \subseteq Y$ implies $X.\varphi \subseteq Y.\varphi$; and $X.\varphi.\varphi = X.\varphi$, for all X, Y.[3] There are many different closure operators. The geometrical convex hull operator is perhaps the most familiar [6], whereas monophonic closure on chordal graphs [7] is a bit obscure.

The intersection of any two closed sets (that is, those Z for which $Z.\varphi = Z$) of a closure space must also be a closed set of the space. The collection of all closed sets, partially ordered by inclusion, forms a lattice, $\mathcal{L}_\varphi$ [13; 16]. Of central importance to our development is the concept of "generators". A set X is a generator of a closed set Z if $X.\varphi = Z$. It is a minimal generator if $Y \subset X$ implies $Y.\varphi \subset X.\varphi = Z$. A closed set can have several minimal generators. By $Z.\gamma_i$ we mean the i^{th} minimal generator of Z, and by $Z.\Gamma$ the collection $\{Z.\gamma_i\}$ of all minimal generators. (From now on, "generator" will mean minimal generator.)

Figure 1 illustrates a closure lattice $\mathcal{L}_\varphi$ denoting a closure

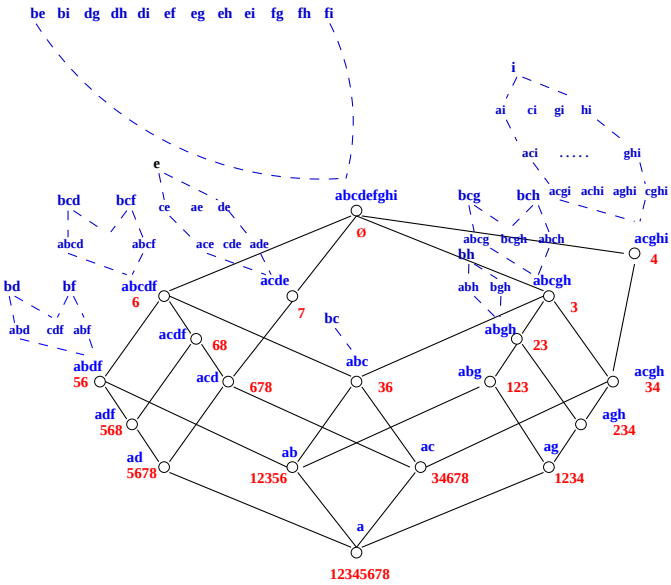

Figure 1: A lattice $\mathcal{L}_\varphi$ of closed sets with some generators shown.

operator φ over a set, or universe $\mathbf{U} = \{a, b, c, d, e, f, g, h, i\}$ of elements. Solid lines connect the closed sets. By dashed lines we have tried to indicate a few of the generating sets. All sets enclosed by them have the same closure. The singleton set $\{e\}$ is a minimal generator of $\{acde\}$. (From now on we elide the curly braces $\{\dots\}$ around sets of elements of $\mathbf{U}$ whenever possible, and retain them to denote collections of sets.) As we will show below, the sets bd and bf are each minimal generators of $abdf$. Thus $abdf.\Gamma = \{abdf.\gamma_1, abdf.\gamma_2\} = \{bd, bf\}$. For the time being we ignore the numbers associated with each node.

If all generators are unique the space is said to be antimatroid. Antimatroid closure spaces are particularly interesting [4; 10; 16]. But, readily, the closure space of Figure 1 is not antimatroid.

The whole space $abcdefghi$ is closed, as required with any closure operator. It has 12 minimal generators, ranging from be through fi. The closure of any subset containing be, or any other generator, must be the whole set $\mathbf{U}$. We observe that, unlike closed sets which are closed under intersection, these generating sets are closed under union.

Let $\mathcal{F}$ be any family of sets. A set B is said to be a blocker of $\mathcal{F}$ if $\forall X \in \mathcal{F}, B \cap X \neq \emptyset$. The difference between a closed set Z and the closed sets Y_i that it covers in $\mathcal{L}_\varphi$ we call the faces F_i of Z.[4] Thus, the faces of $abcdf$ are b, c and df. The faces of the whole space $abcdefghi$ are $eghi$, $bfghi$, $defi$, and $bdef$. The faces of any closed set Z, its generators and blockers are closely related by

THEOREM 2.1. *If Z is closed and $Z.\Gamma = \{Z.\gamma_i\}$ is its family of minimal generators then Z covers X in $\mathcal{L}$ iff $Z - X$ is a minimal blocker of $Z.\Gamma$.*

A proof of this theorem can be found in [17; 18]. The closed set $abdf$ and its generators bd, bf provide a good illustration of this theorem. Since the two faces of $abdf$ are $b = abdf - adf$ and $df = abdf - ab$ and since bd and bf are minimal blockers of these faces, they must be the generators of $abdf$, as asserted by the theorem. The reader should verify that each of the doubleton generators $be \dots fi$ of $abcdefghi$ is a blocker of all of its 4 faces.

2.1 Closed Set Data Mining

Theorem 2.1 applies to any closure system. And any closure system can be represented as a latice, $\mathcal{L}_\varphi$, of closed sets with generators. But, our concern here is to use a specific closure operator to mine relational data. We define our closure operator φ so that the attributes Y of each observable object, o_i, are closed. Since the intersection of closed sets must be closed, these determine all the remaining closures. The lattice $\mathcal{L}_\varphi$ of Figure 1 was created from a small 8×9 relation R, shown in Figure 2 by our procedure. Observe that each

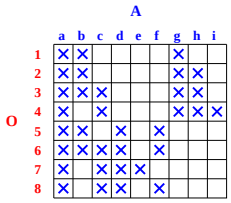

Figure 2: A relation R giving rise to the closed set lattice of Figure 1.

row of R, for instance $< a, b, g >$ and $< a, c, g, h, i >$ is a

[3]We use suffix notation to denote set valued operators. So read $X.\varphi$ as "X closure".

[4]Recall that Z covers Y_i if $Y_i \subset Z$ and there exists no subset Y' such that $Y_i \subset Y' \subset Z$. The term "face" is derived from an application of closure in discrete geometry.

closed set in the lattice $\mathcal{L}_\varphi$ of Figure 1. We interpret these as attributes of the rows (or objects) of O. The numbers under each node of Figure 1 denote which rows contain (or exhibit) that attribute set. They are its support. This example is absurdly small; but it illustrates the relationships between the given relation R, its closed sets in $\mathcal{L}_\varphi$ and their generators. A clear visualization of these relationships is fundamental to understanding the mining technique.

Formal concept theory has been developed by Ganter and Wille [8], and the relation of Figure 2 and the lattice of Figure 1 constitute their first example.[5] It captures the biological description of pond life presented on a children's educational TV show. The attributes A are:

a *needs water to live*,
b *lives in water*,
c *lives on land*,
d *needs chlorophyl to prepare food*,
e *two little leaves grow on germinating*,
f *one little leaf grows on germinating*,
g *can move about*,
h *has limbs*, and
i *suckles its offspring*.

The observations O of objects that exhibited these attributes were: 1 *leech*, 2 *bream*, 3 *frog*, 4 *dog*, 5 *spike-weed*, 6 *reed*, 7 *bean*, and 8 *maize*. While one expects real biological data sets to be a bit more sophisticated and much larger, this provides an excellent introductory example.

In formal concept analysis, Figure 1 is called a "concept lattice"; each closed set is called a "concept". In their excellent book, they demonstrate that the closure, φ, we use is a Galois connection between O and A with many elegant mathematical properties. Most important, they show that if $R : O \rightarrow A$ is a binary relation, or data set, then if Z is a closed set of attributes and X is its generator, then we can translate this as

$$(\forall o_i \in O)[X(o_i) \Rightarrow Z(o_i)].$$

Consider once again bd and bf which generate $abdf$ in Figure 1. In Figure 2, there are only rows with attributes b and d; they are rows 5 and 6. These same two rows are the only ones with attributes b and f. So "if an object (in O) has attributes bd or bf, it must have attributes $abdf$ (the intersection of these two rows)". They are minimal generators, or antecedents of the implication; 5, 6 constitute the support.

One can make logically valid, deterministic assertions about the characteristics of observed objects in R. For example, if each letter in the lattice of Figure 1 denotes a biological property, as it does in [8], then one can assert that: $bcd \vee bcf \Rightarrow abcdf$, and $e \Rightarrow abcde$, and $bcg \vee bch \Rightarrow abcgh$, and so forth. The lattice completely captures the logical structure of R, as the reader can verify.

2.2 Incremental Algorithm

Bernard Ganter [9; 8] has presented algorithms which, given an entire relation R, will determine all the closures (without generators) of $\mathcal{L}_\varphi$. In this section we describe a process which incrementally creates the closure lattice (with generators), one row at a time. It is the major contribution of this paper.

[5]However, they employ a somewhat different closure algorithm which makes repeated sweeps over the binary relation R.

Assume we have made a number of observations of biological objects and have created $\mathcal{L}_\varphi$ based on this data. We now make a new observation o_i of some phenomena exhibiting the set Y of attributes. After reading o_i, Y must be inserted into $\mathcal{L}_\varphi$. If Y already exists in $\mathcal{L}_\varphi$, we only update the support of Y. By the support of a closed attribute set we mean all objects, o_k, that exhibit all of those attributes. The numbers of Figure 1 denote those rows of R that support the closed set.

If $Y \notin \mathcal{L}_\varphi$, it must be added. It is inserted below the smallest closed set Z such that $Y \subset Z$. Z is said to cover Y in $\mathcal{L}_\varphi$. Next the intersection $Y \cap X_k$ must be computed for all X_k covered by Z. If $Y \cap X_k \in \mathcal{L}_\varphi$, nothing more need be done.[6] If $Y \cap X_k \notin \mathcal{L}_\varphi$, it must be recursively entered as a new closed set. Pseudocode for this recursive insertion procedure is shown below:

```
insert_closed_set (SET Z, SET Y, LATTICE L)
      // Insert the closed set Y into L so that
      // it is covered by Z
      {
      SET Y[];

      update_gen (Z, Y, L);
      for_each Y[i] covered by Z do
           {       // Y[i] will be a sibling of Y
           if (not empty(Y meet Y[i]) )
                {
                update_gen (Y, Y meet Y[i], L);
                if (Y meet Y[i] in L)
                     continue;
                X = new SET (Y meet Y[i]);
                insert_closed_set (Y[i], X, L);
                }
           }
      }
```

Godin and Missaoui first proposed this kind of construction in [11]. Our contribution is the UPDATE_GEN process which appears twice in the preceding pseudocode. This process, based on Theorem 2.1, identifies the generators of each closed set, either on-the-fly during incremental construction or in batch mode when displaying or searching the lattice. We believe this is completely new. The following pseudocode describes in detail the procedure for updating the generators incrementally. In this code, the covering set Z is called cov_c and the new closed set Y being inserted is new_c. The COL data type denotes a collection of SETs.

```
void update_generators( concept cov_c, concept new_c )
      // "cov_c" will cover the new concept "new_c"
      // update cov_c.gens to reflect the face
      // cov_c.atts DIFF new_c.atts
      {
      ELEMENT elem;
      SET     face, g_sup;
      COL     new_GEN, keep_GEN, diff_GEN;
      int     keep;
                          // determine the new face
      face = cov_c.atts DIFF new_c.atts;
      new_GEN = EMPTY_COL;
```

[6]Properly, the support of all $X < Y$ in $\mathcal{L}_\varphi$ should be incremented by o_i. But, for efficiency we usually defer this until we display the lattice.

```
for each gen_set in cov_c.gens
     if( (gen_set INTERSECT face) != EMPTY_SET )
          {  // keep any generators that intersect
             // the new face
          add_to_col(gen_set, new_GEN);
          }

keep_GEN = new_GEN;
diff_GEN = cov_c.gens DIFF keep_GEN;
               // for each of the other generators
for each gen_set in diff_GEN
     {
     for each elem in face
          {  // adding this 'elem' of the face
             // will create a blocker
          keep = 1;
          g_sup = gen_set UNION {elem};
          for each keep_set in keep_GEN
               if( keep_set IS_SUBSET_OF g_sup )
                    {  // 'g_sup' not minimal blocker
                    keep = 0;
                    break;
                    }
          if( keep )
               add_to_col(g_sup, new_GEN);
          }
     }
cov_c.gens = new_GEN;
}
```

Obtaining the generators of these closed sets is important for two reasons. First, as noted in Section 1, the authors of [3; 11; 15; 19; 20] all observed that requiring $A \cup B$ to be closed significantly reduces the number of rules $A \Rightarrow B$ returned by frequent set mining. Retaining only those where A is minimal (*i.e.* a minimal generator) reduces the set even further.

To illustrate INSERT_CLOSED_SET and UPDATE_GENERATORS, suppose we return to the pond and observe a plant growing in the water that germinates with two small leaves. It is characterized by the attributes *abe* and the data set of Figure 2 becomes that of Figure 3.

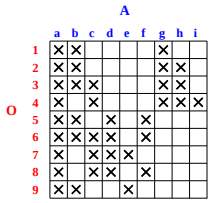

Figure 3: The relation R after adding a new observation/row 9.

Figure 4 illustrates the changed closure system $\mathcal{L}_\varphi$. The new closed set $Y = abe$ has been inserted under $abcdefghi$, the smallest closed set containing it. The siblings of abe are $abcdf, acde, abcgh$ and $acghi$; pairwise intersection yields only ab (already closed) and ae (which is new). It is easy to see that be is the generator of abe since its two faces are $b = abe{-}ae$ and $e = abe{-}ab$. The single generator of ae (which is not shown in the figure to avoid clutter) is $e = ae{-}a$. The generators of $abcdefghi$ are more interesting. The former generator be can no longer be one because it does not block the new face $cdfghi = abcdefghi{-}abe$. So, UPDATE_GENERATORS tries to augment it with the elements of c, d, f, g, h, i. Each combination, such as bce, bde or $bef\ldots$ must block all faces and must be generators. But, bef is not minimal (since $ef \subset bef$) as are all other combinations but bce and bde.

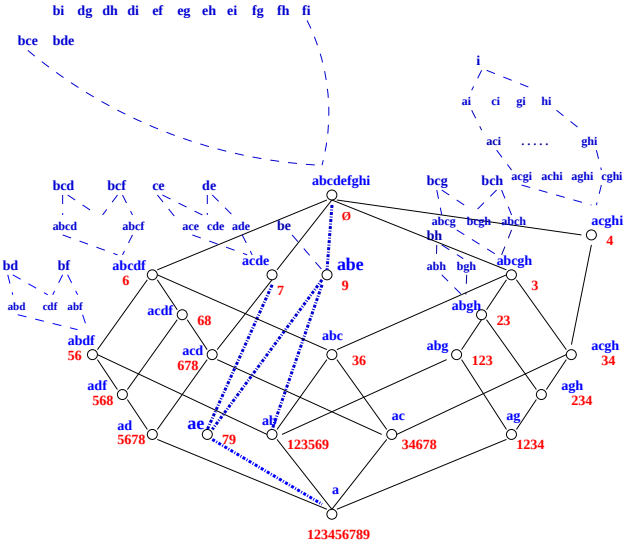

Figure 4: A lattice $\mathcal{L}_\varphi$ of closed sets with a few generators shown.

The generators of the lattice *supremum* $= A = abcdefghi$ have an additional significance. This closed set has $\emptyset$ for support. No object has been observed with all 9 attributes. Consequently, any generator of $abcdefghi$, whether minimal or not, corresponds to logical contradiction over the finite world encompassed by O. A rule based system that not only enumerates all implications that *must* be true, but also enumerates all combinations that *cannot* be true (given the observations to date) can be quite valuable.

Once a system has thousands of observations, any new observation which changes the generators of A, such as row 9 in Figure 3, would be examined carefully. Changing what has heretofore been regarded as an empirical contradiction should not be taken lightly.

3. BIOLOGICAL IMPLICATIONS

We will use the MUSHROOM data set obtained from the UCI Data Repository to illustrate some of the characteristics of closed set data mining. This data set was derived from "The Audubon Society Field Guide to North American Mushrooms" [12] and, since these characteristics of mushrooms are deterministic, it is representative of descriptive biological data. It has also been used in many other data mining studies and thus provides a good point of comparison.

Each record consists of 22 attributes, all of which have nominal values. Because each of these attributes can have multiple values, there are effectively 85 binary attributes. Consequently, for the purposes of this paper we consider only the first 9 attributes which are enumerated in Figure 5. These were converted to binary attributes by appending to letter "values" on the left, the attribute number on the right.

```
Attr-0  edibility:
  e=edible, p=poisonous
attr-1  cap shape:
  b=bell, c=conical, f=flat, k=knobed, s=sunken, x=convex
attr-2  cap surface:
  f=fibrous, g=grooved, s=smooth, y=scaly
attr-3  cap color:
  b=buff, c=cinnamon, e=red, g=gray, n=brown, p=pink,
  r=green, u=purple, w=white, y=yellow
attr-4  bruises?:
  t=bruises, f=doesn't bruise
attr-5  odor:
  a=almond, c=creosote, f=foul, l=anise, m=musty,
  n=none, p=pungent, s=spicy, y=fishy
attr-6  gill attachment:
  a=attached, d=descending, f=free, n=notched
attr-7  gill spacing:
  c=close, d=distant, w=crowded
attr-8  gill size:
  b=broad, n=narrow
```

Figure 5: The first 9 attributes of the MUSHROOM data set, with nominal values.

Thus, for example, a mushroom with attributes `g2` and `p5` has a "grooved cap surface" and a "pungent odor". A single mushroom will exhibit precisely 9 of these 42 possible binary attributes. The first two rows of this reduced data set are

```
p0 x1 s2 n3 t4 p5 f6 c7 n8
e0 x1 s2 y3 t4 a5 f6 c7 b8
```

Readily, the attributes `e0` (edible) and `p0` (poisonous) are of considerable interest. Many of the data mining experiments of the literature have treated this data set as categorical data mining, that is discovering which of the other attributes can be used to assign a specific mushroom to one of these two edibility categories.

Frequent set data mining using *apriori* yields 25,210 rules when we set $min\_sup = 1\%$ and $min\_conf = 90\%$. The discrete data mining technique described in Section 2.2 yields 2,641 closed concepts.[7] There is no nice visualization such as Figure 1. Since some concepts have several generators (*c.f.* concept 667 in Figure 7), this translates into 3,773 distinct rules. We observe the nearly 10-fold reduction described by Zaki [19].

To provide some sense of this data set we list in Figure 6 all those rules $A \Rightarrow B$ which have a singleton attribute for A. These are often regarded as conveying the most information because they are easiest to comprehend and facilitate transitive reasoning. We have added the concept number to the left to indicate where this rule was uncovered in the discrete data mining process and its support to indicate the min_sup necessary to consider it frequent. If $\sigma = 1\%$, $min\_sup = 81$; so all but 6 of these rules would have been uncovered by frequent set mining.

Medical data mining is commonly concerned with a small set of distinct "outcomes", such as survival rates or diagnoses.

[7]If all attributes are encoded we generate a concept lattice with 104,104 closed concepts! It was not pretty. Our algorithm took over a day and a half. On the other hand, an open source implementation of *apriori* available in the public domain, executing on a server with 2 GB of main memory, required nearly 3 days to return 15,552,210 rules with $\sigma = 1\%$, $\gamma = 90\%$. There are frequent sets of size 19 in the MUSHROOM data set. (The same implementation required less than 6 hours when run on a server with 4 GB of memory.)

The performance and growth patterns of closed set data mining are quite interesting and will be the topic of a different paper.

CONCEPT	IMPLICATION			SUPPORT
60	w3	->	f6	1040
105	t4	->	f6	3376
109	n8	->	f6	2512
117	a5	->	e0, t4, f6	400
144	l5	->	e0, t4, f6	400
313	s1	->	e0, f2, f4, n5, f6, c7, n8	32
556	f2	->	f6	2320
604	w7	->	f6	1312
668	c5	->	p0, x1, f4, f6, n8	192
720	y3	->	f6	1072
898	b3	->	t4, f6, c7, b8	168
924	f5	->	p0, f6, c7	2160
1007	p3	->	f6	144
1081	u3	->	e0, y2, f4, n5, f6, c7, n8	16
1401	g2	->	p0, w3, t4, n5, f6, w7, n8	4
1553	r3	->	e0, y2, f4, n5, f6, c7, n8	16
1597	s5	->	p0, f4, f6, c7, n8	576
1687	y5	->	p0, f4, f6, c7, n8	576
2019	a6	->	f4, c7, b8	210
2022	m5	->	p0, y2, f4, c7, b8	36
2162	e3	->	c7	1500
2562	c1	->	p0, n5, f6, w7, n8	4

Figure 6: All implications in MUSHROOM with a single antecedent.

We then seek attributes that will effectively categorize these outcomes, as in [14]. The "edible" and "poisonous" categories of the MUSHROOM data set are also examples. The implications of Figure 6 show that "smell" is an important criterion of edibility, with `c5, f5, s5, y5` and `m5` all indicating poisonous and `a5` and `l5` implying edibility. These 7 implications are invariably found by frequent set mining. But, we see that `c1` and `g2` also indicate poisonous; and with a support of only 4 instances, they are unlikely to be found. But, if you eat any of the 4 kinds of mushroom with a conical cap or grooved cap cover it might be serious.

Are there simple combinations of attributes that also denote poisonous? Figure 7 illustrates those non-trivial implications $A \Rightarrow B$ for which $|A| = 2$ and `p0` $\in B$. Again recall

CONCEPT	IMPLICATION			SUPPORT
666	p3, w7	->	p0, x1, f4, c5, f6	32
667	p3, f4	->	p0, x1, c5, f6, n8	64
667	p3, n8	->	p0, x1, f4, c5, f6	64
696	f2, p3	->	p0, x1, f4, c5, f6, n8	32
1184	b1, n8	->	p0, n5, f6	12
1495	b1, b3	->	p0, t4, n5, f6, c7, b8	12
1567	b1, p3	->	p0, t4, n5, f6, c7, b8	12
2081	y3, n5	->	p0, f4, f6, n8	24
2177	e3, f4	->	p0, c7	876
2181	y2, a6	->	p0, f4, m5, c7, b8	18
2372	c3, a6	->	p0, y2, f4, m5, c7, b8	6
2470	e3, a6	->	p0, y2, f4, m5, c7, b8	6
2561	c1, y3	->	p0, y2, f4, n5, f6, w7, n8	2
2561	c1, f4	->	p0, y2, y3, n5, f6, w7, n8	2
2563	c1, y2	->	p0, n5, f6, w7, n8	3

Figure 7: Rules with two attribute antecedents that denote poisonous mushrooms.

that if $\sigma = 1\%$, $min\_sup = 81$, so only one of these rules would have been discovered by frequent set analysis.

4. IMPRECISE DATA

Closed set mining is discrete and deterministic; it presumes perfect data. But unfortunately, sometimes scientific observation is imprecise. We can use frequency, as indicated by the support of each closed set, to help us in two different

ways.

First, suppose in a study of animal characteristics we have the implication

$$\text{“}nurse\ young\text{”} \Rightarrow \text{“}give\ live\ birth\text{”} \quad .$$

This association is supported by thousands of observations. Then we observe a duck-billed platypus, which nurses its young but lays eggs. Scientific enquiry often includes surprises! Our procedure can be programmed so that when an association with high support would be changed by a single observation, o_k, it is flagged for latter scrutiny and not processed. The observation may be in error; or like the platypus, it may be the exception that tests the rule. In either case, it deserves special attention.

Second, suppose that property a implies properties b, c and d. That is, logically $a \Rightarrow abcd$. But, suppose the test for d is suspect; it is only 95% accurate. Because a only implies bc in a deterministic fashion, the resulting closed set lattice will have the structure shown in Figure 8. Attribute d is

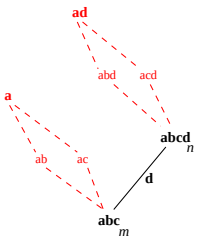

Figure 8: A concept lattice fragment showing only two closed sets.

sometimes associated with a. Here we have labeled the face d and indicated the support by m and n. By Theorem 2.1, the face d gives rise (correctly) to its place in the generator ad of $abcd$. By the strength of a face we mean the difference $n - m$. When this strength is relatively small it makes sense to combine the two closed sets, as in this case

If one expects substantial experimental error or probabilistic associations, closed set mining is not appropriate. But, by using the support of each concept, which one would want in any case, it can be made resistent to minor imprecisions.

5. SUMMARY

Experimental studies of deterministic scientific phenomena can take hours, days, or years. It is not like collecting point-of-sale data where response time is essential. Data that is painstakingly collected deserve a careful, thorough analysis that can reveal unusual, even rare, associations. Closed set mining provides a tool to do this.

Closed set data mining accommodates closed sets of great length; it finds all associations, but even then far fewer than frequent set mining; it supports the incremental observation of additional objects as well as additional attributes; and it creates a permanent lattice structure that can be easily researched for specific associations. For the knowledge discovery niche that it fills, we believe closed set mining is the most effective tool to date.

6. REFERENCES

[1] Jean-Mark Adamo. *Data Mining for Association Rules and Sequential Patterns*. Springer Verlag, New York, 2000.

[2] Rakesh Agrawal, Tomasz Imielinski, and Arun Swami. Mining Association Rules between Sets of Items in Large Databases . In *Proc. 1993 ACM SIGMOD Conf.*, pages 207–216, Washington, DC, May 1993.

[3] Stephen E. Brossette and Alan P. Sprague. Medical surveillance, frequent sets and closure operations. *J. Combinatorial Optimization*, 5:81–94, 2001.

[4] Brenda L. Dietrich. Matroids and Antimatroids — A Survey. *Discrete Mathematics*, 78:223–237, 1989.

[5] Saso Dzeroski. Data Mining in a Nutshell. In Saso Dzeroski and Nada Lavrac, editors, *Relational Data Mining*. Springer Verlag, 2001.

[6] Paul H. Edelman and Robert E. Jamison. The Theory of Convex Geometries. *Geometriae Dedicata*, 19(3):247–270, Dec. 1985.

[7] Martin Farber and Robert E. Jamison. Convexity in Graphs and Hypergraphs. *SIAM J. Algebra and Discrete Methods*, 7(3):433–444, July 1986.

[8] Bernard Ganter and Rudolf Wille. *Formal Concept Analysis - Mathematical Foundations*. Springer Verlag, Heidelberg, 1999.

[9] Bernhard Ganter and Klaus Reuter. Finding All Closed Sets: A General Approach. *Order*, 8(3):283–290, 1991.

[10] Paul Glasserman and David D. Yao. Generalized semi-Markov Processes: Antimatroid Structure and Second-order Properties. *Math. Oper. Res.*, 17(2):444–469, 1992.

[11] Robert Godin and Rokia Missaoui. An incremental concept formation approach for learning from databases. In *Theoretical Comp. Sci.*, volume 133, pages 387–419, 1994.

[12] G. H. Lincoff. *The Audubon Society Field Guide to North American Mushrooms*. Alfred A. Knopf, New York, 1981.

[13] Bernard Monjardet. A Use for Frequently Rediscovering a Concept. *Order*, 1:415–416, 1985.

[14] Carlos Ordonez, Edward Omiecinski, Levien de Braal, Cesar A. Santana, Norberto Ezquerra, Jose A. Taboada, David Cooke, Elizabeth Krawczynska, and Ernst V. Garcia. Mining Constrained Association Rules to Predict Heart Disease. In *IEEE International Conf. on Data Mining, ICDM*, pages 433–440, 2001.

[15] Nicolas Pasquier, Yves Bastide, Rafik Taouil, and Lofti Lakhal. Discovering Frequent Closed Itemsets for Association Rules. In *Proc. 7th International Conf. on Database Theory (ICDT)*, pages 398–416, Jan. 1999.

[16] John L. Pfaltz. Closure Lattices. *Discrete Mathematics*, 154:217–236, 1996.

[17] John L. Pfaltz and Robert E. Jamison. Closure Systems and their Structure. *Information Sciences*, 139:275–286, 2001.

[18] John L. Pfaltz and Christopher M. Taylor. Concept Lattices as a Scientific Knowledge Discovery Technique. In *2nd SIAM International Conference on Data Mining*, pages 65–74, Arlington, VA, Apr. 2002.

[19] Mohammed J. Zaki. Generating Non-Redundant Association Rules. In *6th ACM SIGKDD Intern'l Conf. on Knowledge Discovery and Data Mining*, pages 34–43, Boston, MA, Aug. 2000.

[20] Mohammed J. Zaki and Ching-Jui Hsiao. CHARM: An Efficient Algorithm for Closed Association Rule Mining. Technical Report TR 99-10, RPI, 1999.