Mining Changes of Classification by Correspondence Tracing *

Ke Wang† Senqiang Zhou‡ Chee Ada Fu§ Jeffrey Xu Yu¶

Abstract

We study the problem of mining changes of classification characteristics as the data changes. Available are an *old* classifier, representing previous knowledge about classification characteristics, and a *new* data. We want to find the changes of classification characteristics in the new data. An example of such changes is "members with a large family no longer shop frequently, but they used to". Finding this kind of changes holds the key for the organization to adopt to the changed environment and stay ahead of competitors. The challenge is that it is difficult to see what has really changed from comparing the old and new classifiers that could be very large and different. In this paper, we propose a technique to identify such changes. The idea is tracing the characteristics, in the old and new classifiers, that correspond to each other by classifying the same examples. We describe several ways to present changes so that the user can focus on a small number of important ones. We evaluate the proposed method on real life data sets.

1 Introduction

Changes can be opportunities to some people (organizations) and curses to others. A key to staying ahead in the changing world is knowing important changes and devising strategies for adopting to them. There are three steps in this process: detecting changes, identifying the causes of changes, and acting upon the causes to respond to the changes. Detecting changes in a form understandable to the user is the most important step because it alerts opportunities and challenges ahead and trigger the other steps. For example, by mining changes the user may find that many members with a large family no longer shop frequently. This information could alert the organization about a potential lose of customers and trigger actions to retain such customers.

In this paper, we study the *change mining* problem in the context of classification [15]. The classification refers to extracting characteristics called a classifier from a sample of pre-classified examples, and the goal is to assign classes, as accurately as possible, for other examples that follow the same class distribution as the sample examples. In the change mining problem, we have an old classifier, representing some previous knowledge about classification, and a new data set that has a changed class distribution. We want to find the changes of classification characteristics in the new data set.

For changes to be understandable to the user, two requirements are essential. First, changes must be described *explicitly*. Simply returning the pair of old and new classifiers does not work because it is not reasonable to expect the user to extract the changes from comparing two classifiers that are potentially large and dissimilar. For example, a decision tree classifier can easily have several dozens (if not hundreds) of rules, and a change at the top levels will make the classifier look very different. Second, the user should be told what changes are important because often more changes are found than what a human user can possibly handle.

Change mining is a difficult problem. First of all, it is not clear how the change of classification should be measured. Simply measuring the number of rules added and deleted does not work because a similar classification can be produced by dissimilar rules. Moreover, a small change in rules could account for most changes in classification accuracy. There are a few studies on this issue in the literature (see Section 2 for related work). In [11], to extract and understand changes, a new classifier is required to resemble the old classifier to some extent, i.e., follow a similar splitting in the decision tree construction. This restriction makes it less likely to find important changes. For example, important attributes often occur at top levels of the decision tree, and if such attributes change, the method in [11] cannot be used. In [9], the change between two classifiers is measured by the amount of work required to transform them into some common specialization. In the real life, the human user hardly thinks of changes in terms of such a common specialization.

We believe that a new classifier should best capture the characteristics in the new data set, even at the expense of losing similarity to the old classifier. It is the

*Supported in part by a research grant from the Natural Science and Engineering Research Council of Canada and by a research grant from Networks of Centres of Excellence/Institute for Robotics and Intelligent Systems

†Simon Fraser University, wangk@cs.sfu.ca

‡Simon Fraser University, szhoua@cs.sfu.ca

§The Chinese University of Hong Kong, adafu@cs.cuhk.edu.hk

¶The Chinese University of Hong Kong, yu@se.cuhk.edu.hk

task of change mining to find what characteristics have changed with respect to the old classifier. To perform this task, we propose a new change mining technique, called *correspondence tracing*, to trace the corresponding rules in the old and new classifiers through the examples that they both classify. This idea is analogous to identifying the difference between two catalogs I and II in the real life: for each section o (i.e., each rule in change mining) of I, we find how the products (i.e., examples in change mining) listed under o are listed in II by tracing the corresponding sections of II that list these products. In change mining, each rule is a characteristics of the examples classified, and we use the difference of corresponding rules to obtain a description of changes. The following example illustrates this technique.

EXAMPLE 1.1. *Consider training a classifier for admission decisions using a graduate applicant database. In the past, the TOEFL score was an important factor of admission. Recently, there is a policy change: reference letters and the standing of undergraduate schools are more important. Therefore, on the new data set processed under the new policy, the following old rule o based on the TOEFL score performs poorly: out of 50 examples classified, 20 are misclassified:*

o: $TOEFL = High \rightarrow Yes$ *(50, 20).*

Below, let X_o *denote the 50 examples classified by o. Suppose that we construct a new classifier from the new data set and that the examples in* X_o *are classified by the following new rules:*

n_1: $Letter = Not\_Strong \rightarrow No$ *(22/30, 3/5)*
n_2: $School = Good \rightarrow Yes$ *(28/40, 1/3).*

The notation $(22/30, 3/5)$ *for* n_1 *is read as follows:* n_1 *classifies 22 examples from* X_o*, of which 3 are misclassified, and classifies 30 examples not from* X_o*, of which 5 are misclassified. There is a similar reading for* n_2*. These information convey two aspects of changes.*

Characteristics change. *The new rules* n_1 *and* n_2 *"correspond" to the old rule o classifying the sub-population classified by o (in addition to other examples). We can use pairs* $< o, n_1 >$ *and* $< o, n_2 >$ *to describe the changes for this sub-population, read as: for the sub-population classified by o, the admission criterion has shifted from TOEFL score (i.e., o) to reference letters (i.e.,* n_1*) and the standing of schools (i.e.,* n_2*). Notice that understanding these changes does not require that the involved old rule and new rules be similar in syntax. This is an important difference between our approach and [11].*

Quantitative change. *The statistics given the bracket () can be used to quantify the significance of changes. Intuitively, new rules* n_1 *and* n_2 *are doing much better than the old rule o because they make only 3+1=4 misclassifications instead of originally 20. We can rank all characteristics changes by such an improvement to classification accuracy, the primary goal of classification. The user can then select important changes for action based on this informed ranking. Of course, to avoid overfitting, quantitative change should be estimated on the* whole *population, not on the given sample. (End of Example)*

The above approach can be summarized as follows. To find important changes, we abandon the restriction that the new classifier be similar to the old one, and we deal with extracting changes from potentially dissimilar classifiers (indeed, the old rule o and new rules n_i in the above example do not share syntax similarity). Our approach is to trace the corresponding new rules for each old rule through the examples classified and use them to describe the changes of the old rule. To present changes in an understandable manner, we rank all changes according to the improvement to classification accuracy. This ranking criterion makes sense because it addresses the primary goal of classification. With this ranking, the user typically only needs to examine the top few changes that account for most of the accuracy improvement. We will describe the details for finding characteristics changes, estimating quantitative changes, and presenting changes to the user.

In the rest of the paper, we review related works in Section 2, present our approach in Section 3, and report an experiment in Section 4. Finally, we conclude the paper.

2 Related Work

In the context of association rule mining [3], incremental mining [6] maintains the completeness of association rules in the presence of insertion/deletion of data, active mining [2] tracks the change of support and confidence over time, emerging pattern mining [8] and contrast-set mining [4] identify conditions whose support has changed substantially across two or more groups. In all these works, each rule or pattern is considered in isolation, consequently, changes are variations or consequences of one another. In [13], fundamental rule changes are considered in the context of pruning "redundant rules". A fundamental change of a rule (in support or confidence) is not a direct consequence of changes of some conditions in the rule. Such changes are restricted to rules of the generalization/specialization relationship. None of these works deals with the classification problem where changes should be extracted on the basis of improving the goal of classification, the classification accuracy.

The drifting environment [18, 10] concerns with producing a classifier by assigning more weight to recently arrived data. [5] exploits the user knowledge to construct an understandable classifier. None of these works addresses the change mining problem studied here. [9] presents a framework for measuring changes in two models such as two classifiers. A model is represented by a partition of the data space that summarizes the data. The change between two models is measured by the amount of work required to transform the two models into the common specialization obtained by overlaying the two models' partitionings. In practice, the human user hardly measures changes this way. Also, such an "editing distance" does not address the primary goal of classification, the classification accuracy. [11] extracts changes by requiring the new classifier to be similar to the old one, i.e., using either the same splitting attributes or the same splittings in the decision tree construction. This is a severe restriction because important changes may vanish from classifiers even though they exist in the data. The work on finding tree differences [16] is not applicable here because dissimilar decision trees could produce similar classification. Also, changes of classification depend not only on the structure of rules, but also on the statistical property of rules.

3 The Proposed Approach

We consider classifiers given by a set of rules. A rule has the form, $A_1\theta_1 a_1 \wedge \ldots \wedge A_k\theta_k a_k \rightarrow c$, where A_i is a predictor attribute, a_i is a value for A_i, θ_i is one of $=$, $\geq$, $\leq$. c is a class of the class attribute C. The only assumption we made about a classifier is that *exactly one* rule is used to classify a given example. This assumption is satisfied in most cases because each example is typically assigned to exactly one class, such as the decision tree classifier or the decision rule classifier [15], and association based classifiers [12, 17]. This includes the default rule that is used only if there is no matching rule for the given example.

In the *change mining* problem, we have an old classifier O and a new data set D. Alternatively, the old classifier can be replaced with an old data set from which the old classifier can be constructed. The task is to find how the classification characteristics has changed in the new data set relative to the old classifier. Notice that the terms "old" and "new" do not have to correspond to the time dimension. For example, we can apply the change mining to find the changes between a male population and an female population.

Before change mining, some methods can be applied to detect the *existence* of changes in the new data set. For example, we can construct a new classifier from the new data set and apply both the old classifier and the new classifier to the new data set. If the new classifier is significantly more accurate than the old classifier, the classification characteristics must have changed (assuming that both classifiers are constructed by the same algorithm). Even if the new classifier is not more accurate than the old classifier, it could still capture alternative characteristics as changes, and such changes may trigger alternative actions. Therefore, more precisely, the notion of changes here refers to the changes captured by the old and new classifiers, which do not necessarily imply a data distribution change. With this said, however, our primary interest in this paper is in those "real" changes that play an essential role in improving the classification accuracy.

3.1 The algorithm. We find changes in the new data set D in four steps. First, we construct a new classifier for D, by applying an existing algorithm. Second, for each example in D, we determine the classifying rules in both old and new classifiers. Third, for each old rule o, we identify the *corresponding* new rules, denoted $New(o)$, that classify the examples classified by o, and estimate the *quantitative change* (relative to o) for each new rule n_i in $New(o)$. Finally, we present *characteristics changes* of the form $< o, n_i >$ or $< o, New(o) >$ in a list ranked by quantitative change. This algorithm is described below.

- Step 1: Construct a new classifier from D, by adopting an existing algorithm. We use C4.5 for classifier construction in this paper.
- Step 2: For each example in D, identify the old and new classifying rules. This can be done by modifying a classifier to output the classifying rule for each example classified.
- Step 3: For each old rule o,
  - Step 3.1: identify the corresponding new rules, $New(o) = \{n_1, \ldots, n_k\}$, where each n_i classifies at least one example in D classified by o. This can be done in the same scan of examples as in Step 2: for each example in D, we draw an edge from the old classifying rule to the new classifying rule. $New(o)$ is the set of new rules to which o has an edge.
  - Step 3.2: estimate the quantitative change of $< o, n_i >$ for each rule n_i in $New(o)$. The detail is given in Section 3.2.
- Step 4: Present changes. This step presents the characteristics changes of the form $< o, n_i >$ or

$< o, New(o) >$, ranked by quantitative change, so that the user can focus on a small number of important changes. There are several ways to do this, depending on the level at which the user likes to know changes. The detail will be given in Section 3.3.

If a new rule classifies k examples in D, it corresponds to at most k old rules because no example is classified by more than one old rule. Therefore, we have the following observation.

OBSERVATION 3.1. *There are at most $|D|$ changes of the form $< o, n_i >$, where $|D|$ denotes the number of examples in D. (End of Observation)*

The complexity of the above algorithm is as follows. Step 1 is the standard C4.5 classifier construction, for which efficient algorithms exist. Step 2 takes two scans of the given data set because finding the classifying rule for a given example takes a constant time (for descending the decision tree). Step 3.1 can be done in the same data scan as in Step 2 and each example only adds one edge. Step 3.2 scans all changes $< o, n_i >$ once. From Observation 3.1, this work is bounded by the number of examples in the given data set D. Step 4 involves sorting all changes $< o, n_i >$. This work is bounded by $|D|log|D|$.

The above is *forward change mining* in that it starts with an old rule and identify the corresponding new rules. A forward change tells how each old characteristics evolves to new ones. In contrast, *backward change mining* starts with a new rule and identify the corresponding old rules that classify the examples classified by the new rule. A backward change tells how each new characteristic "originates" from old ones. Our discussion focuses on forward change mining, but it is equally applicable to backward change mining with the roles of old and new rules exchanged.

EXAMPLE 3.1. *We use the "Lenses" data set from the UCI repository [14] to illustrate our approach. There are four attributes, three classes, and 18 examples:*

Attributes:
- A_1*: Age: 1, 2, 3*
- A_2*: Spectacle Prescription: 1, 2*
- A_3*: Astigmatic: 1, 2*
- A_4*: Tear Production Rate: 1, 2*

Classes:
- C_1*: Hard Contact Lenses, 4 examples*
- C_2*: Soft Contact Lenses, 5 examples*
- C_3*: No Contact Lenses, 9 examples.*

TID	A1,A2,A3,A4	Class	Changed Class
0	1, 1, 1, 1	3	
1	1, 1, 1, 2	2	
2	1, 1, 2, 1	3	2
3	1, 1, 2, 2	1	
4	1, 2, 1, 2	2	
5	1, 2, 2, 2	1	
6	2, 1, 1, 2	2	
7	2, 1, 2, 1	3	2
8	2, 1, 2, 2	1	
9	2, 2, 1, 1	3	
10	2, 2, 1, 2	2	
11	2, 2, 2, 2	3	
12	3, 1, 1, 2	3	
13	3, 1, 2, 1	3	2
14	3, 1, 2, 2	1	
15	3, 2, 1, 1	3	
16	3, 2, 1, 2	2	
17	3, 2, 2, 1	3	2

On this data set, the C4.5 program [15] produces the 3 rules below (with the default class being C_3):

$o_1 : A_4 = 1 \rightarrow C_3$ *[0, 2, 7, 9, 13, 15, 17 (N=7,E=0)]*
$o_2 : A_3 = 1 \wedge A_4 = 2 \rightarrow C_2$ *[1, 4, 6, 10, 12, 16 (N=6, E=1)]*
$o_3 : A_3 = 2 \wedge A_4 = 2 \rightarrow C_1$ *[3, 5, 8, 11, 14 (N=5, E=1)]*

For each rule, the bracket [and] contains the ids of the examples classified, with N giving the number of such examples and E giving the number of misclassified.

Suppose now that examples 2, 7, 13 and 17 change their class from C_3 to C_2, as in the "Changed Class" column. These are the examples that are classified by o_1 and satisfy $A_3 = 2$. On the new data set, o_1 becomes less accurate with 4 misclassifications:

$o_1 : A_4 = 1 \rightarrow C_3$ *[0, 2, 7, 9, 13, 15, 17 (N=7,E=4)]*

We apply change mining to find the changes in the new data set. First, we construct the new C4.5 classifier from the new data set, obtaining 4 new rules (with the default class being C_2), where N and E refer to the new data set:

$n_1 : A_3 = 1 \wedge A_4 = 1 \rightarrow C_3$ *[0, 9, 15 (N=3,E=0)]*
$n_2 : A_3 = 1 \wedge A_4 = 2 \rightarrow C_2$ *[1, 4, 6, 10, 12, 16 (N=6, E=1)]*
$n_3 : A_3 = 2 \wedge A_4 = 1 \rightarrow C_2$ *[2, 7, 13, 17 (N=4, E=0)]*
$n_4 : A_3 = 2 \wedge A_4 = 2 \rightarrow C_1$ *[3, 5, 8, 11, 14 (N=5, E=1)].*

Comparing the classification of the two classifiers on the new data set, it is apparent that n_1 and n_3 classify the examples that were classified by o_1. Thus, $\{n_1, n_3\}$ are the corresponding new rules of o_1. We use $< o_1, n_1 >$ and $< o_1, n_3 >$ to describe the changes for the sub-population classified by o_1. These changes *are read as: the examples classified by o_1 that have $A_3 = 2$ have changed the class from C_3 to C_2, and that have $A_3 = 1$ remain unchanged. This is exactly the change that we made earlier. Changes $< o_2, n_2 >$ and $< o_3, n_4 >$ are trivial because old and new rules are identical. (End of Example)*

Often, there are more changes than what the human user can handle. It is very important that the user be informed of the importance of changes and that changes be presented so that it is easy to spot a small number of important changes. The rest of this section addresses these issues.

3.2 Estimating quantitative change. The importance of a change is measured by its relevance to the goal of classification. In particular, a change is important to the extent that recognizing it can improve the classification accuracy. This accuracy improvement is called *quantitative change*. The quantitative change should be measured (more precisely, estimated) on the whole population, not just on the given training sample D. Below, we present a method for estimating quantitative change.

Consider a characteristics change $< o, n_i >$, where n_i is a corresponding new rule of an old rule o. In the population of which D is a sample,

- let $Cover(o)$ denote the sub-population classified by o,
- let $Cover(n_i/o)$ denote the subset of $Cover(o)$ that is classified by n_i.

Notice that $Cover(o)$ and $Cover(n_i/o)$ refer to the underlying population, not the sample D. The quantitative change of $< o, n_i >$ is the change of the errors of the two rules on $Cover(n_i/o)$. We borrow the *pessimistic estimation* from [7, 15] to estimate these errors.

To explain the pessimistic estimation, we use the analogy of estimating the rate of left-handed people in a population of 1,000,000 people. Suppose that in a sample S of N people randomly selected from the population, we observed that E are left-handed. E/N is the left-handed rate for the sample S. The larger the sample size N is, the closer E/N is to the real left-handed rate in the population. For a given confidence level CF, we can determine an upper bound, denoted $U_{CF}(N, E)$, such that the chance that the left-handed rate in the population is more than $U_{CF}(N, E)$ is less than CF, or equivalently, the chance that the left-handed rate is no more than $U_{CF}(N, E)$ is at least $1-CF$. (The default value of CF used by C4.5 is 25%.) $U_{CF}(N, E)$ is a pessimistic estimation because it is an upper bound. The smaller the sample size N is, the less reliable the number E is, due to more randomness in a small sample, and a larger pessimistic estimation is necessary to satisfy a given confidence level. This property is used by C4.5 to prune overly specific rules that tend to have a large pessimistic estimation. We omit the exact computation of $U_{CF}(N, E)$, which can be found in the C4.5 code.

To estimate the error rate of an old rule o, we can map $Cover(o)$ to the population of people, map the examples in D classified by o to people in the sample S, and map the examples in D misclassified by o to left-handed people in the sample S.

- Let N be the number of examples in D classified by o, E of which are misclassified. In a C4.5 classifier, N and E are available for every rule.

The (upper bound of) error rate of o in $Cover(o)$ is estimated by $U_{CF}(N, E)$. If we select N examples randomly from $Cover(o)$, we have $1 - CF$ confidence that the number of errors is no more than $N \times U_{CF}(N, E)$.

The same estimation applies to the corresponding new rules $New(o) = \{n_1, \ldots, n_k\}$ of o.

- Let N_i be the number of examples in D classified by n_i, E_i of which are misclassified. N_i and E_i are available from a C4.5 classifier.
- Let d_i be the number of the above N_i examples that are also classified by o (in the old classifier). Notice that $1 \leq d_i \leq N_i$ and $N = d_1 + \ldots + d_k$. d_i can be computed in the scan of examples in Step 3.1.

$U_{CF}(N_i, E_i)$ is the pessimistic estimation of the error rate of new rule n_i for the sub-population $Cover(n_i/o)$. For any d_i examples randomly selected from $Cover(n_i/o)$, the number of misclassifications by n_i is estimated by $d_i \times U_{CF}(N_i, E_i)$, and the number of misclassifications by o is estimated by $d_i \times U_{CF}(N, E)$. Therefore, the change in the number of correct classifications, due to the change from o to n_i, is estimated by $d_i \times U_{CF}(N, E) - d_i \times U_{CF}(N_i, E_i)$.

Definition 3.1. (Quantitative change) The *quantitative change* of $< o, n_i >$ is

$$\Delta(o, n_i) = (d_i/|D|) \times (U_{CF}(N, E) - U_{CF}(N_i, E_i)),$$

where $|D|$ is the number of examples in the new data set D. (End of Definition)

Intuitively, $\Delta(o, n_i)$ measures the estimated accuracy increase (either positive or negative) due to the change from o to n_i. $\Delta(o, n_i)$ is large if n_i classifies many examples, in which case d_i is large, and is accurate, in which case $U_{CF}(N, E) - U_{CF}(N_i, E_i)$ is large. We can generalize this notion to more than one change in a natural way: the quantitative change of k changes $< o_1, n_1 >, \ldots, < o_k, n_k >$ is $\Sigma_{j=1}^{k}\Delta(o_j, n_j)$. That is, the accuracy improvement by several changes is the sum of the accuracy improvement by each change. The additivity follows from the disjointness of examples classified by different rules.

In the rest of the paper, $\Delta(o, New(o))$ denotes the quantitative change of all changes related to the old rule o, i.e., $\Sigma_{n_i \in New(o)}\Delta(o, n_i)$, and Δ denotes the quantitative change of all changes of classifier O, i.e., $\Sigma_{o \in O}\Delta(o, New(o))$.

DEFINITION 3.2. (CUTOFF COVERAGE) Consider a list $< o_1, n_1 >, \ldots, < o_k, n_k >$ and a prefix $< o_1, n_1 >, \ldots, < o_i, n_i >$, $i \leq k$.

- The *cutoff coverage* of the prefix with respect to the list is
$$\Sigma_{j=1}^{i}\Delta(o_j, n_j)/\Sigma_{j=1}^{k}\Delta(o_j, n_j).$$
- The *cutoff coverage* of the prefix with respect to the classifier is
$$\Sigma_{j=1}^{i}\Delta(o_j, n_j)/\Delta. \text{ (End of Definition)}$$

A similar notion of cutoff coverage can be defined for a list $< o_1, New(o_1) >, \ldots, < o_k, New(o_k) >$. The cutoff coverage measures the relative contribution of a prefix with respect to a longer list of changes or with respect to all the changes. Thus, the cutoff coverage of a prefix of changes tells how much change has been captured (in percentage) if the rest of the list is cut off. If we rank all changes by quantitative change, typically it suffices to examine a short prefix to have a large cutoff coverage because large quantitative changes concentrate near the top of the list.

EXAMPLE 3.2. *Continue with Example 3.1. Let us compute the quantitative change of $< o_1, \{n_1, n_3\} >$. Notice that n_1 and n_3 classify only the examples classified by o_1. Hence, $d_1 = N_1 = 3$ and $d_3 = N_3 = 4$. $|D| = 18$.*

$$\begin{aligned}\Delta(o_1, n_1) &= (3/18)(U_{CF}(7, 4) - U_{CF}(3, 0))\\ &= (3/18)(0.755 - 0.37) = 1.155/18 = 6.4\%,\\ \Delta(o_1, n_3) &= (4/18)(U_{CF}(7, 4) - U_{CF}(4, 0))\\ &= 4/18(0.755 - 0.293) = 1.848/18 = 10.3\%,\\ \Delta(o_1, \{n_1, n_3\}) &= 16.7\%.\end{aligned}$$

Thus, the change $< o_1, \{n_1, n_3\} >$ is important to the extent of increasing the estimated accuracy increases by 16.7%. In the ranked list, $< o_1, n_3 >, < o_1, n_1 >$, the cutoff coverage at $< o_1, n_3 >$ with respect to the list is $10.3/16.7 = 61\%$. *This is also the cutoff coverage with respect to the classifier because the other old rules, o_2 and o_3, do not change. (End of Example)*

3.3 Presenting changes.

3.3 Presenting changes. Usually, the user likes to see changes at certain levels or of certain types.

3.3.1 Changes at different levels. Changes can occur at different levels.

Example level changes. At the lowest level, the user finds changes by posing "what if" queries on selected examples. For example, an example level change in Example 1.1 can tell how a given applicant is admitted/rejected before and after the change. This change can be described by the old rule and new rule $< o, n_i >$ that classify the example, which contrasts the characteristics used and classes assigned in the old and new classifications. Moreover, $U_{CF}(N, E)$ and $U_{CF}(N_i, E_i)$ can be used to describe the certainty of classification, where N, E, N_i, E_i are as defined in Section 3.2.

Rule level changes. At the rule level, the user wants to know the changes for the sub-population classified by a given old rule o, i.e., $< o, New(o) >$. In Example 1.1, the rule level changes tell the policy change to n_1 and n_2 for the sub-population who used to be admitted based on a high TOEFL score. We can present such changes by a list $< o, n_1 >, \ldots, < o, n_k >$ ranked by $\Delta(o, n_i)$, where n_i are in $New(o)$. For each prefix $< o, n_1 >, \ldots, < o, n_i >$, the cutoff coverage tells the percentage of quantitative change, with respect to the list and with respect to the classifier, captured by the prefix. The user can read changes from left to right and cut off the list based on the cutoff coverage. Similarly, for changes below we assume that the cutoff coverage for every prefix is computed.

Class level changes. At the class level, the user wants to know the changes for a given class C. For example, the changes for class Yes in Example 1.1 tell how the successful applicants under the old policy are processed differently under the new policy. We can present class level changes by the list $< o_1, New(o_1) >, \ldots, < o_k, New(o_k) >$, ranked by $\Delta(o_i, New(o_i))$, where o_i is an old rule for class C. Alternatively, we can present the list $< o_1, n_1 >, \ldots, < o_k, n_k >$, ranked by $\Delta(o_i, n_i)$, where o_i is an old rule for class C and $n_i \in New(o_i)$. In the second presentation, the user does not have to see all the changes for one old rule before seeing some more important changes for another old rule

Classifier level changes. At this level, the user wants to know all the changes for the whole classifier. We can present such changes by a ranked list of $< o, n_i >$ changes or a ranked list of $< o, New(o) >$ changes.

3.3.2 Types of changes. We can also categorize changes into several types, not necessarily exclusive.

Global changes: A *global* change occurs if both characteristics change and quantitative change are large. Such changes are always interesting because new characteristics are significantly different and more accurate. For example, in the decision tree construction, if choosing a different splitting attribute at the top level results in a significant increase in accuracy, this is a global change because the most important attribute is changed. Example 1.1 also shows a global change where the new rules make use of different attributes than the old rule and result in a significant higher accuracy. Global changes cannot be found in [11] because new rules are highly dissimilar to old ones.

Alternatives changes. If a characteristics change is large, but its quantitative change is small, the new rules essentially represent *alternative* characteristics to the old rule, in that they are equally capable of the classification task. Alternatives changes occur for several reasons: the new classifier is constructed by a different algorithm that exploits a different search bias, or a small change in the data takes the search to follow a different path. Though alternatives changes do not improve accuracy, they provide alternative characteristics of the data, thus, new possibilities of actions. Finding alternatives changes for the purpose of such actions requires a different ranking criterion and more input from the user. In this paper, we are mainly interested in changes in terms of the action of improving the classification accuracy.

Target changes. In a *target* change, some sub-population classified by an old rule switches to a different class. In this case, the old rule will consistently misclassify the sub-population because it has not caught up the class change. Example 1.1 shows a target change where some applicants admitted previously are now rejected by the new admission criterion. The shopping example in Introduction is another target change where customers with a large family switch the status from frequent shopping to infrequent shopping. Target changes are always interesting because they alert changes of the target variable.

Specialization changes. A *specialization* change occurs when some sub-population classified by an old rule has changed so much, judged by an accuracy increase, that it is justified to have its own classification. This sub-population is captured by having additional conditions in a corresponding new rule. While a target change describes a sub-population that switches to a new class, a specialization change describes a sub-population that preserves the old class, but at a higher accuracy. Often, these two types of changes occur together. For example, an old rule

$$Member = Yes \rightarrow Frequent$$

may be involved in a specialization change:

$$Member = Yes \wedge Size = Small \rightarrow Frequent$$

and a target change:

$$Member = Yes \wedge Size = Large \rightarrow Infrequent.$$

Generalization changes. A *generalization* change occurs when some old characteristics become unimportant and several old rules containing them are combined after removing such characteristics. This change is useful to know because pruning unimportant characteristics not only increases the accuracy, but also focuses the user on real characteristics. In a generalization change, a new rule generalizes several old rules, so the backward change mining that starts with a new rule and finds corresponding old rules is more suitable.

Interval changes. An *interval* change occurs if there is a shift of boundary points, due to the emerging of new cutting points. In addition, an interval can be refined into several small intervals if it is justified for each small interval to have a separate classification characteristics, either having a different class or having higher accuracy.

4 Experiments

We evaluated the proposed method on two real-life data sets, German Credit Data from the UCI Repository of Machine Learning Databases [14], and IPUMS Census Data from [1]. These data sets were chosen because no special knowledge is required to understand the addressed applications. To verify if the proposed method finds the changes that are supposed to be found, we need to know such changes beforehand. For this reason, we "planted" several changes into the German Credit Data and verified if the proposed method finds them. For the IPUMS Census Data, we applied the proposed method to find the changes across different years or different sub-populations. For each change mining task, we have an "old" data set and a "new" data set. For concreteness, we consider tree classifiers built by the C4.5 program.

4.1 Experiments on German Credit Data. This data set has two classes, "good" and "bad" (credits), 7 numerical attributes and 13 categorical attributes. 700 examples belong to the "good" class and 300 examples belong to the "bad" class. We first extracted the C4.5 classifier from the given data set, which serves as our previous knowledge O. Each rule is named by an id output by C4.5. We then planted several changes in the data set, one at a time, and applied the proposed method to find them. Here, we report only changes at the rule level, which form the basis for mining changes at class and classifier levels.

4.1.1 Change 1: Target change.

The old rule o_{72} in O classifies 23 examples correctly in the *original* data set:

```
o72:
        Personal-status = single-male
        Foreign = no
        ->  good  [N=23,E=0]
```

12 of these examples have Liable-people=1, and the remaining 11 examples have Liable-people=2. We then changed the "good" class of the 12 examples with Liable-people=1 to the "bad" class, and keep the "good" class for the 11 examples with Liable-people=2. Let D denote the new data set. The new classifier built using the new data set shows the accuracy increase from 79.90% (of the old classifier on D) to 82.20%. We applied the change mining algorithm to O and D.

The following changes for old rule o_{72} are found: $< o_{72}, n_{201} >$, $< o_{72}, n_{199} >$, $< o_{72}, n_{66} >$, $< o_{72}, n_{115} >$, ranked by quantitative change, where n_i are new rules in the new classifier.

```
n201:
        Liable-people > 1
        Foreign = no
        ->  good  [N=11,E=0]

n199:
        Personal-status = single-male
        Liable-people <= 1
        Foreign = no
        ->  bad  [N=10,E=0]

n66:
        Savings-account = over1000DM
        Debtors = none
        Duration > 11
        ->  good  [N=24,E=4]

n115:
        Savings-account = less500DM
        Personal-status = single-male
        Job = skilled
        Credit <= 9857
        ->  good  [N=18,E=4]
```

Here are detailed statistics of the changes:

N, E, d_i, N_i, E_i are as defined in Section 3.2. For each prefix ending at rule n_i, the last three columns Δ_i, cc_i, cc'_i are quantitative change, cutoff coverage with respect to the list and with respect to the classifier. d_i

$Old(N,E)$	$New(d_i, N_i, E_i)$	Δ_i	cc_i	cc'_i
$o_{72}(23,12)$	$n_{201}(11,11,0)$	0.54%	49.54%	13.35%
	$n_{199}(10,10,0)$	1.02%	94.01%	25.25%
	$n_{66}(1,24,4)$	1.06%	97.24%	26.50%
	$n_{115}(1,18,4)$	1.09%	100%	27.25%

tells that the new rules from top to bottom classify 11, 10, 1 and 1 of the 23 examples classified by o_{72}. Consider the prefix $< o_{72}, n_{201} >, < o_{72}, n_{199} >$ of the list. The quantitative change of the prefix is 1.02%, the cutoff coverage is 94.01% with respect to the list and 25.25% with respect to the classifier. Here is the detailed computation:

$$\Delta(o_{72}, n_{201}) = (11/1000)(U_{CF}(23,12) - U_{CF}(11,0))$$
$$= (11/1000)(0.615 - 0.119) = 0.54\%.$$
$$\Delta(o_{72}, n_{199}) = (10/1000)(U_{CF}(23,12) - U_{CF}(10,0))$$
$$= (10/1000)(0.615 - 0.131) = 0.48\%.$$
$$\Delta(o_{72}, n_{66}) = (1/1000)(U_{CF}(23,12) - U_{CF}(24,4))$$
$$= (1/1000)(0.615 - 0.250) = 0.036\%.$$
$$\Delta(o_{72}, n_{115}) = (1/1000)(U_{CF}(23,12) - U_{CF}(18,4))$$
$$= (1/1000)(0.615 - 0.328) = 0.029\%.$$

The cutoff coverage up to n_{199} with respect to the list is

$$(0.54 + 0.48)/(0.54 + 0.48 + 0.036 + 0.029) = 94.01\%.$$

The cutoff coverage up to n_{199} with respect to the classifier is

$$(0.54\% + 0.48\%)/4.04\% = 25.25\%,$$

where 4.04% is the quantitative change of the classifier (computation not shown here).

The changes $< o_{72}, n_{201} >$ and $< o_{72}, n_{199} >$ can be read as: previously (Personal-status=single-male and Foreign=no) implies a "good" credit; now if Liable-people ≤ 1 also holds, the credit is "bad". We reproduced the classification by rule o_{72} on the *new* data set:

```
o72:
        Personal-status = single-male
        Foreign = no
        ->  good  [N=23,E=12]
```

Comparing this with the earlier classification by new rules n_{201} and n_{199}, the new rules are able to separate the class of most (new) examples classified by o_{72}, i.e., 11 with "good" class from 10 with "bad" class, by using different ranges of Liable-people. This is exactly the change we planted earlier in the data.

4.1.2 Change 2: Specialization change.

We planted a specialization change as follows. Consider the old rule o_{17} below, which classifies the largest number of examples in the original data, i.e., 164, with 47 misclassified. Notice that the 47 misclassified examples have "good" credit (because the class of the rule is "bad") and the remaining 117 correctly classified examples have "bad" credit. We changed the "Residence-time" value to 3 for the 47 misclassified examples and change the "Residence-time" value to 1 for the remaining 117 correctly classified examples. The intuition for this change is that a long "Residence-time" may be positively related to a "good" credit. This change is significant because building and not building a new classifier gives the accuracy of 88.30% and 81.10%, respectively.

On the changed data, our change mining algorithm finds the following characteristics changes $< o_{17}, \{n_8, n_{40}\} >$ at the top of the list:

$Old(N,E)$	$New(d_i,N_i,E_i)$	Δ_i	cc_i	cc'_i
$o_{17}(164,47)$	$n_8(119,138,12)$	2.48%	64.42%	32.38%
	$n_{40}(45,95,0)$	3.85%	100.00%	50.26%

```
o17:
        Status = 0DM
        Duration > 11
        Foreign = yes
        ->  bad  [N=164,E=47]

n8:
        Status = 0DM
        Residence-time <= 1
        ->  bad  [N=138,E=12]

n40:
        Status = 0DM
        Foreign = yes
        Duration > 11
        Residence-time > 1
        ->  good  [N=95,E=0]
```

The new rules n_8 and n_{40} classify 119 and 45 examples classified by o_{17} ($d_8 = 119$ and $d_{40} = 45$). These 45 examples were misclassified by o_{17} into the "bad" class, but now are correctly classified as the "good" class. We can read this change as: previously (Status=0DM and Duration > 11 and Foreign=yes) implies a "bad" credit; now those customers satisfying the extra condition Residence-time > 1 have the "good" credit. Failing to identify this change means losing good customers. The quantitative change of the classifier is 7.66%, so the cutoff coverage up to n_{40} with respect to the classifier is 3.85%/7.66% = 50.26%.

4.1.3 Change 3: Interval change.

Next, we planted an interval change. Examining the major rule o_{17}, we noticed that the boundary point Duration=11 plays an important role in deciding the credit of a customer. So we bring in a change by increasing the "Duration" value by 6 (months) for each example classified by o_{17}. Training a new classifier on the new data set improves the accuracy from 81.10% to 83.80%. Our algorithm finds $< o_{17}, \{n_{17}, n_{35}, n_{57}, n_2\} >$ as the top changes.

$Old(N,E)$	$New(d_i,N_i,E_i)$	Δ_i	cc_i	cc'_i
$o_{17}(164,47)$	$n_{17}(130,139,27)$	1.20%	89.78%	35.69%
	$n_{35}(27,59,14)$	1.27%	95.02%	37.77%
	$n_{57}(6,25,4)$	1.32%	98.76%	39.26%
	$n_2(1,26,2)$	1.34%	100.00%	39.75%

```
o17:
        Status = 0DM
        Duration > 11
        Foreign = yes
        ->  bad  [N=164,E=47]

n17:
        Status = 0DM
        Duration > 16
        Foreign = yes
        ->  bad  [N=139,E=27]

n35:
        Credit-history = duly-till-now
        Credit > 1386
        Existing-credits <= 1
        Telephone = yes
        Debtors = none
        Duration <= 30
        Liable-people <= 1
        ->  good  [N=59,E=14]

n57:
        Debtors = guarantor
        Housing = own
        ->  good  [N=25,E=4]

n2:
        Status = 0DM
        Duration <= 16
        Residence-time <= 3
        ->  good  [N=26,E=2]
```

Comparing o_{17} and n_{17}, the planted interval change was found for 130 ($d_{17} = 130$) out of the 164 examples changed: previously (Status = 0DM and Duration > 11 and Foreign = yes) implies a "bad" credit, now the "Duration" threshold is shifted to 16. The next new rule n_{35} classifies 27 examples classified by o_{17}, using different attributes, which allows to correctly classify the "good" examples that were previously misclassified by o_{17}. New rules n_{57} and n_2 classify only 6+1=7 examples classified by o_{17}, thus, are relatively minor in connection with o_{17}.

4.1.4 Change 4: Global change. We created a more drastic change that uses a different splitting attribute at the top level of the decision tree. The attribute "Status" is the first splitting attribute in the original decision tree. Suppose that we want to make "Debtor" the most important factor for a customer's credit. Below is the class distribution for the three values of "Debtor" (i.e., none, co-applicant, guarantor) before the change:

```
              none  co-applicant  guarantor
-----------------------------------------
"good" class  635   23            42

"bad" class   272   18            10
-----------------------------------------
```

We increased the information gain ratio [15] of "Debtor" (the attribute selection criterion used by the decision tree) by changing 150 examples with Debtor=none in the "bad" class to "good" class. Here is the new class distribution:

```
               none  co-applicant  guarantor
------------------------------------------
"good" class   785   23            42
"bad" class    122   18            10
------------------------------------------
```

After the change, "Debtor" has the highest information gain ratio, therefore, is selected at the first level of the decision tree. Building the new classifier on the new data set increases the accuracy from 79.10% to 88.40%. The following are a few most significant changes found by our algorithm:

$Old(N, E)$	$New(d_i, N_i, E_i)$	Δ_i	cc_i	cc'_i
$o_{17}(164, 104)$	$n_{80}(25, 106, 15)$	1.23%	22.70%	12.49%
	$n_{145}(17, 71, 7)$	2.13%	39.30%	21.63%
	$n_{39}(15, 52, 7)$	2.85%	52.60%	28.94%
	$n_{73}(12, 34, 3)$	3.47%	64.04%	35.24%
	$n_{82}(9, 29, 3)$	3.92%	72.34%	39.81%

```
o17:
        Status = ODM
        Duration > 11
        Foreign = yes
        ->  bad  [N=164,E=104]
n80:
        Debtors = none
        Existing-credits > 1
        ->  good  [N=106,E=15]
n145:
        Savings-account = unknown
        ->  good  [N=71,E=7]
n39:
        Debtors = none
        Property = real-estate
        ->  good  [N=52,E=7]
n73:
        Duration > 21
        Purpose = used-car
        ->  good  [N=34,E=3]
n82:
        Purpose = radio-tv
        Debtors = none
        ->  good  [N=29,E=3]
```

Comparing the above old rule and new rules, the main change is that the old rule contains attribute "Status", whereas none of the new rules does. Instead, three new rules, n_{80}, n_{39} and n_{82}, contain Debtor=none. This is consistent with the change planted in the data.

4.2 Experiments on IPUMS Census Data. The IPUMS database contains PUMS census data from the Los Angeles and Long Beach areas for the years 1970, 1980, and 1990. We chose the "1-in-100" sample in the source, and we chose "Vetstat" (the veteran status) as the class attribute. "Vetstat" has four values: "N/A", "no service", "yes" and "not ascertained". After removing all examples of the "N/A" or "not ascertained" values for "Vetstat", 24549, 56800 and 67236 examples remain for years 1970, 1980 and 1990, respectively. Table 1 depicts the race distribution of examples as given by the attribute "raceg" (race-general). We considered two ways to make up the data set for a change mining problem:

- compare the same race in two different years, and
- compare two different races in the same year.

Here we report the changes found for 1970 vs 1990 for "black", and the changes found for "black" vs "chinese" in 1990.

4.2.1 1970-black vs 1990-black. The sub-populations of "black" in 1970 and 1990 were used as the old and new data sets, respectively, denoted by 1970-black and 1990-black. Old and new classifiers were built from 1970-black and 1990-black. The accuracy of the old classifier (built from the 1970-black) is

Rule	N	E	d_i	Δ_i	cc_i	cc'_i
$o238 : 35 < age \leq 54 \rightarrow yes$	2049	1685	N/A	11.70%	100.00%	83.35%
$n274 : 40 < age \leq 72, sex = male \rightarrow yes$	932	366	501	3.05%	26.07%	21.73%
$n269 : age \leq 51, famunit = 1 \rightarrow no$	187	14	153	4.69%	40.09%	33.41%
$n141 : age \leq 40, nfams = 1, wkswork2 = [48, 49] \rightarrow no$	572	31	136	6.17%	52.74%	44.01%

Table 2: Top changes found from "1970-black" to "1990-black"

Rule	N	E	d_i	Δ_i	cc_i	cc'_i
$o274 : 40 < age \leq 72, sex = male \rightarrow yes$	224	195	N/A	8.13%	100%	86.59%
$n26 : bplg = china, incss \leq 5748 \rightarrow no$	167	6	104	4.56%	56.09%	48.57%
$n16 : age \leq 46, movedin \leq 5 \rightarrow no$	637	9	59	7.24%	89.05%	76.69%

Table 3: Top changes found from 1990-black to 1990-chinese

	1970	1980	1990
white	21,403	45,282	52,424
black	2,276	6,685	7,052
american indian	64	419	340
chinese	186	818	1,907
japanese	362	1,001	1,148
other asian	158	1,864	4,257
other race	100	731	108

Table 1: Race vs Year split

64.60%, compared to the accuracy of 89.70% of the new classifier (built from 1990-black). Table 2 shows several top changes. (The attribute $famunit$ refers to the family unit in the household the person belongs to, the attribute $nfams$ refers to the number of families in a household, and the attribute $wkswork2$ refers to the number of weeks worked last year.) In the first row, Δ_i, cc_i, cc'_i refer to those for all changes of the old rule. The quantitative change of the list is 6.17%. This accounts for 52.74% of the quantitative change of o_{238} and 44.01% of the quantitative change of the classifier. Here is the detailed computation:

$$\begin{aligned}
&\Delta(o238, n274)\\
&\quad = (d_i/|D|)(U_{CF}(2049, 1685) - U_{CF}(932, 366))\\
&\quad = (501/7052)(0.83 - 0.40) = 3.05\%\\
&\Delta(o238, n269)\\
&\quad = (d_i/|D|)(U_{CF}(2049, 1685) - U_{CF}(187, 14))\\
&\quad = (153/7052)(0.83 - 0.07) = 1.65\%\\
&\Delta(o238, n141)\\
&\quad = (d_i/|D|)(U_{CF}(2049, 1685) - U_{CF}(572, 31))\\
&\quad = (136/7052)(0.83 - 0.06) = 1.48\%
\end{aligned}$$

We can read the top change $< o238, n274 >$ as: in 1970, $35 < age <= 54$ was a characteristic of veterans, but in 1990, $40 < age \leq 72 \wedge sex = male$ was a new characteristic of veterans. The change of the upper limit of the age interval is due to the fact that a veteran of age 54 in 1970 will be captured as age 74 in 1990 if they participated in the 1990 data collection. The emerging of the new condition $sex = male$ is likely due to the fact that most new veterans in the 20 years between 1970 and 1990 were males. This experiment also showed that a small number of changes at the top of the list captured a significant portion, in this case 44.01%, of the overall change.

4.2.2 1990-black vs 1990-chinese.

In this experiment, we fixed the year at 1990 and compared the black sub-population (as the old data set) with the chinese sub-population (as the new data set). The accuracy of old and new classifiers (on the new data set) is 88.50% and 97.90%. Table 3 shows some significant changes found. (*bplg* refers to the birth place, *incss* refers to social security income, and *movedin* refers to the number of years prior to the census year that the individual moved into the present dwelling unit.) The cutoff coverage of the list is 7.24%/9.44% = 76.69%, where 9.44% is the accuracy change for the whole old classifier. We omitted the detailed computation. In this experiment, the top two changes captured 76.69% of overall change!

In summary, these experiments verified that the proposed method found the changes of classification characteristics as the data changes. Moreover, the experiments also verified that a small number of important changes are responsible for most of the accuracy change, and the proposed method found such changes at the top of the list.

5 Conclusion

In change mining, some previous knowledge about the data is known and the data has changed since then. The problem is to find what characteristics have changed relative to the previous knowledge. A solution to this problem holds the key for the organization to adopt to the changed environment. Despite its importance, there were very few studies on mining changes. In this paper, we studied the change mining problem in the context of classification. The study made two main contributions. First, we dealt with extracting changes from potentially very dissimilar old and new classifiers. For this purpose, we proposed a novel change mining technique called *correspondence tracing*. Second, we ranked the importance of changes with respect to the goal of classification and we proposed a method of presenting changes so that the user only needs to know a small number of changes. The experiments on real life data sets showed that our method is effective in finding important changes.

References

[1] In *http://kdd.ics.uci.edu/databases/ipums/ipums.html, http://www.ipums.umn.edu/*.

[2] R. Agrawal and G. Psaila. Active data mining. In *KDD*, 1995.

[3] R. Agrawal and R. Srikant. Fast algorithm for mining association rules. In *VLDB*, 1994.

[4] S. D. Bay and M. J. Pazzani. Detecting change in categorical data: mining contrast sets. In *SIGKDD*, 1999.

[5] S. Chen and B. Liu. Generating classification rules according to user's existing knowledge. In *SIAM Conference on Data Mining*, 2001.

[6] D. Cheung, J. Han, V. Ng, and C. Wong. Maintenance of discovered association rules in large databases: an incremental updating techniques. In *ICDE*, 1996.

[7] C. Clopper and E. Pearson. The use of confidence or fiducial limits illustrated in the case of the binomia. In *Biometrika, 26:4 (available from http://www.jstor.org/journals/bio.html)*, pages 404–413, 1934.

[8] G. Dong and J. Li. Efficient mining of emerging patterns: discovering trends and differences. In *KDD*, 1999.

[9] V. Ganti, J. Gehrke, and R. Ramakrishnan. A framework for measuring changes in data characteristics. In *PODS*, 1999.

[10] T. Lane and C. Brodley. Approaches to online learning and concept drift for user identification in computer security. In *KDD*, 1998.

[11] B. Liu, W. Hsu, H. S. Han, and Y. Xia. Mining changes for real-life applications. In *DaWak*, 2000.

[12] B. Liu, W. Hsu, and Y. Ma. Integrating classification and association rule mining. In *KDD*, pages 80–86, 1998.

[13] B. Liu, W. Hsu, and Y. Ma. Discovering the set of fundamental rule changes. In *SIGKDD*, 2001.

[14] C. J. Merz and P. Murphy. Uci repository of machine learning databases. In *http://www.cs.uci.edu/m̃learn/MLRepository.html*, 1996.

[15] J. R. Quinlan. C4.5: programs for machine learning. 1993.

[16] D. B. Skillicorn. A parallel tree difference algorithm. *Information Processing Letters*, 60(5):231–235, 1996.

[17] K. Wang, S. Zhou, and Y. He. Growing decision trees on association rules. In *SIGKDD*. SIGKDD, 2000.

[18] G. Widmer. Learning in the presence of concept drift and hidden contexts. In *Machine learning*, 1996.