Mining Influential Attributes That Capture Class and Group Contrast Behaviour

Elsa Loekito and James Bailey
NICTA Victoria Laboratory
Department of Computer Science and Software Engineering
University of Melbourne, Australia
{eloekito,jbailey}@csse.unimelb.edu.au

ABSTRACT

Contrast data mining is a key tool for finding differences between sets of objects, or classes, and contrast patterns are a popular method for discrimination between two classes. However, such patterns can be limited in two primary ways: i) They do not readily allow second order differentiation - i.e. discovering contrasts of contrasts, ii) Mining contrast patterns often results in an overwhelming volume of output for the user. To address these limitations, this paper proposes a method which can identify contrast behaviour across both classes and also groups of classes. Furthermore, to increase interpretability for the user, it presents a new technique for finding the attributes which represent the key underlying factors behind the contrast behaviour. The associated mining task is computationally challenging and we describe an efficient algorithm to handle it, based on binary decision diagrams. Experimental results demonstrate that our technique can efficiently identify and explain contrast behaviour which would be difficult or impossible to isolate using standard techniques.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications—*Data mining*; I.5.2 [**Pattern Recognition**]: Design Methodology—*Feature evaluation and selection*

General Terms

Algorithms, Design

Keywords

Contrast patterns, Second-order contrast patterns, Group contrast, Influential attributes, Emerging Patterns

1. INTRODUCTION

Contrast mining is a key tool for finding differences between sets of objects, or classes. For example, contrasting the class of people who are admitted to hospital versus the class of people who are not admitted to hospital, might reveal that young children with diabetes are overrepresented in the first class compared to the second. Mining of contrasts can be useful in many situations, such as when comparing sets of objects between different classes, comparing sets of objects from different time periods, comparing sets of objects from different spatial locations, or comparing sets of objects before and after some medical treatment. A well known type of contrast pattern (which we term *first-order contrast*) are the emerging patterns [2]. They represent combinations of attribute values that have a strong ability to discriminate between two classes. However, existing contrast patterns are limited in two significant ways. Firstly, they cannot identify second-order properties, that require the mining of 'contrasts of contrasts'. Such information is useful for discovering how differentiating factors can vary across groups. Secondly, an overwhelming number of patterns is often output from mining. E.g. in the census data set [6], millions of (first-order) contrast patterns that differentiate males from females can be discovered, based on only the first ten attributes in the data set. What is needed is the ability to summarise the meaning of a set of (first or second order) contrast patterns in a highly compact way.

Motivated by these two limitations, we address the following two challenges '*how do we discover and mine second-order differences?*' and '*how do we identify to the user those attributes which have the most impact with respect to a collection of second order-differences?*' We propose two solutions: i) A method that discovers the second-order differences between contrasts for one group of classes, compared to contrasts for some other group of classes. This problem differs from standard contrast mining scenarios, since one needs to be able to compare across groups of classes, as well as between classes. ii) A technique for ranking attributes, based on their degree of influence within a collection of second order differences. Such a ranking is far easier to interpret by a user, compared to returning millions of patterns. It aims to identify the key underlying factors responsible for change across groups.

Second-order differences are meaningful in a number of interesting situations. Two motivating examples are:

EXAMPLE 1. *In the census data set [6], one might wish to ask* 'what are the differences between males and females, which are characteristic for one race group, but less characteristic for another race group?' *Alternatively, consider the domain of plant physiology [12]. Plant biologists would*

CIKM'08, October 26–30, 2008, Napa Valley, California, USA.

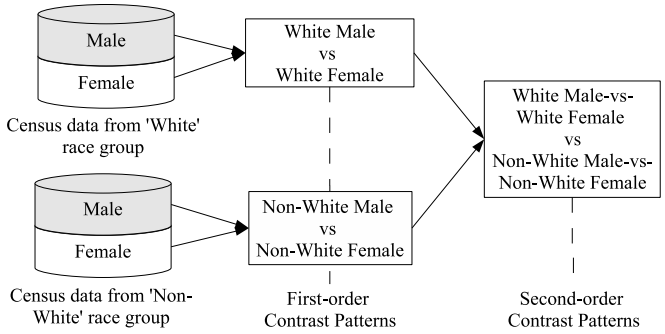


Figure 1: First order differences within groups and second order differences across groups

like to discover 'how does the response to a given treatment differ between the tip and base of a leaf?' *First-order contrast mining discovers treatment contrasts, comparing leaf samples which are given a treatment, against leaf samples which are not given the treatment. Second-order differentiation then compares treatment contrasts with respect to the tip of the leaf, against treatment contrasts with respect to the base of the leaf.*

In answering the first part of our research question, we introduce a class of second-order contrast patterns that we will call the **Group Discriminative Contrast (GDC) patterns**. They correspond to patterns of contrast that strongly differentiate the classes in one group, but whose discriminative power (i.e. ability to differentiate the classes) in the other group is weaker. To explain further, consider an example of a second-order contrast for the census data set [6]. When comparing the differences between male and female across two race groups, i.e. '*White*' and '*Non-White*', some patterns are able to strongly discriminate males and females if the individual belongs to the 'white' race group, but not if the individual belongs to the other race group. Figure 1 provides a conceptual diagram explaining the relationship between the first-order and second-order contrasts.

EXAMPLE 2. *In the* census *dataset, 1.5% males and 0.4% females in the 'White' population satisfy the pattern* 'older than 60 years and worked in a durable manufacturing industry'. *This shows that the rule consisting of age and the industry can significantly contrast males from females in the 'white' race group, since there are 4 times more males than females for which the rule is true. However, this pattern does not match any individual in the 'non-white' population and hence it is not a contrast for that group. We say this pattern is a Group Discriminative Contrast pattern, since it is a class contrast (between male and female) for one group ('White'), but it is not a class contrast for the other group ('Non-white').*

To answer the second part of our research question, we propose a technique for finding attributes which represent the underlying factors behind second-order contrast behaviour. In particular, we identify influential attributes, whose values can be used to find partitions of the original groups, such that these partitions show significant differences in contrast behaviour across the groups. Our work is motivated by the work in [8] which shows that variation in values for certain attributes may increase/decrease the discriminative ability of some contrast patterns. How to assess the degree to which an attribute is responsible in the discriminative ability of contrast patterns has so far been an open question. The number of contrast patterns is usually exponential in the number of attributes, whereas the number of influential attributes is smaller than the number of attributes.

EXAMPLE 3. *Recall the previous example. Suppose the 'working industry' of the individual is not included in the pattern. In the 'White' population group, 11.5% males and 15.2% females are 60 years old, or older, and in the 'Non-White' population group, 6% males and 10% females belong to that age group. This shows that considering the age by itself does not capture a strong contrast in either race group. Moreover, the industry specification attribute has some degree of group discriminative contrast influence, since when combined with age information, it helps the differentiation between males and females in the 'White' group, but does not help the differentiation in the 'Non-White' group. Furthermore, if the industry attribute had a similar effect when combined with many different patterns, we would rank it highly in terms of overall attribute influence.*

Challenges: A major challenge of our research is that it is not obvious how one can develop a concept of second order contrast that is simple, intuitive, and useful in practice. Addressing this question is a key aim of the paper. On the mining side, since we are discovering the patterns of second order contrast, as well as the influence of each attribute in those patterns, our mining task conceptually requires a repeated and potentially expensive exploration of the pattern space for each possible attribute. It is thus important to be able to push constraints deep into the mining process.

State of the art: Existing work in contrast mining [2, 15] has addressed the problem of finding differences between two classes, but it has not addressed the problem of finding differences between differences, i.e. the second-order contrasts. In regard to ranking how influential an attribute is, existing feature ranking techniques such as entropy or statistical measures, purely focus on the ability of a single attribute to determine a class label. They do not rank an attribute based on consideration of its participation in multi-variate behaviour, or on its ability to find subcategories that exhibit interesting contrast behaviour. This can be very limiting and may result in important attributes being overlooked. e.g. Work in [9] has shown that attributes which are ranked low according to entropy, may still be influential with respect to a set of contrast patterns. Our technique can uncover such attributes, since the influence of an attribute is measured with respect to its behaviour and participation within combinations of contrast patterns.

This paper makes the following important contributions:

- This paper addresses two levels of contrast: i) the contrast between classes within a group, ii) the contrast of those contrasts between groups. We introduce a formal definition for a novel type of contrast pattern, the *Group Discriminative Contrast* pattern, that differentiates the classes within a group, and at the same time, discriminates between the groups. Furthermore, we introduce a new attribute ranking method that measures the influence of an attribute with respect to its dis-

criminative power for second-order contrast patterns, termed the *Group Discriminative Contrast Influence*.

- We propose a mining technique which can efficiently explore the pattern space and mine the set of second order contrasts, as well as rank the degree of influence for each attribute within this set. Our algorithm is based on the use of Weighted Zero-suppressed Binary Decision Diagrams [11] and relies on a novel method for embedding group discriminative constraints within a prefix enumeration style framework.

- We experimentally evaluate our technique on real datasets, and compare our attribute influential scoring method against other classic feature ranking methods, such as entropy and correlational techniques. Our experiments demonstrate the efficiency of our mining technique and also show that our approach is able to discover some intuitively meaningful attributes, representing underlying influential factors that would be difficult or impossible to isolate using standard techniques.

2. PRELIMINARIES

Assume we have a data set D defined upon a set of k attributes. For every attribute A_i, $i \in \{1, 2..k\}$, the domain of its values (or items) is denoted by $dom(A_i)$. Let I be the aggregate of the domains items across all the attributes, i.e. $I = \bigcup_{i=1}^{k} dom(A_i)$. An *itemset* is a subset of I. Let p and q be two itemsets. We say p *contains* q if p is a superset of q, i.e. $p \supseteq q$. We require that an itemset can contain at most one item from the domain of any given attribute.

The data set D can be projected to a multi-dimensional space, where each attribute corresponds to a dimension in this space, and an itemset corresponds to a subspace. The *projection of p on dimension A*, denoted p_A, is the item in itemset p which belongs to the domain of attribute A, i.e. $p_A = p \cap dom(A)$. If $p_A \neq \{\}$, then p is called an *A-dependent itemset*, or p *depends* on the value of attribute A. Given an A-dependent itemset p, q is the *A-generalization* of p if q contains all items in p except the item which belongs to the domain of attribute A, i.e. $q = p \setminus p_A$.

EXAMPLE 4. *Let $p_1 = \{x_0, y_1, z_1\}$ be an itemset that depends on 3 attributes, where $\mathrm{dom}(A_1) = \{x_0, x_1\}$, $\mathrm{dom}(A_2) = \{y_0, y_1\}$, and $\mathrm{dom}(A_3) = \{z_0, z_1\}$. The projection of p_1 in dimension A_3 is $\{z_1\}$, and its A_3-generalization is $\{x_0, y_1\}$.*

A *dataset* is a collection of transactions, where each transaction is an itemset. The *support* of an itemset p in dataset D, i.e. $support(p, D)$, is the fraction of the transactions in D which contain p ($0 \leq support(p, D) \leq 1$). The support function is monotonic, that is, for all itemset q such that $p \supseteq q$, $support(p, D) \leq support(q, D)$.

In the context of first-order contrast mining, a data set contains a positive class, namely D_p and a negative class namely D_n. The *growth rate* of an itemset p, denoted $\mathrm{gr}(p)$, is the ratio between its support in D_p and its support in D_n, i.e. $\mathrm{gr}(p) = \frac{support(p,D_p)}{support(p,D_n)}$. For all itemsets q such that $p \supseteq q$, if $support(p, D_p) = support(q, D_p)$, then $gr(p) \geq gr(q)$, and if $support(p, D_n) = support(q, D_n)$, then $gr(p) \leq gr(q)$. Given α and β threshold values, where $0 \leq \alpha \leq 1$ and $0 \leq \beta \leq 1$, an itemset p is an **emerging pattern (EP) [2]** if $support(p, D_p) \geq \alpha$ and $support(p, D_n) \leq \beta$.

3. GROUP DISCRIMINATIVE CONTRAST

In this section, we define the second-order contrast characteristics between two groups of classes in terms of **group discriminative contrast patterns** and **group discriminative contrast influential attributes**, whose definitions generalise previous work on emerging patterns (EPs). Considering the data in a multi-dimensional space, an EP between the positive and the negative class in a particular group corresponds to a subspace that contains at least α positive instances and no more than β negative instances from that group. Such a subspace may have different contrasting ability between the classes in another group though. Hence, before introducing our second-order contrast definitions, we firstly introduce a formula for measuring the contrast strength of a pattern in a particular group, using a function similar to one in [3] as follows.

DEFINITION 1. *Let G_1 and G_2 be two groups of classes. Each group G, $G \in \{G_1, G_2\}$, contains a positive class and a negative class. Given an itemset p and a group G, we refer to the positive and the negative class in G, as D_p and D_n, respectively. Let $support_G(p, C)$ be the support of p in class C in group G, and $gr_G(p)$ be the growth rate of p in group G. The* **contrast intensity** *of p in group G, denoted $CI_G(p, D_p, D_n)$, is the discriminative power between the positive instances and the negative instances from group G which are contained in subspace p, and is defined as a function of the support and growth rate of p:*

$$\mathrm{CI}_G(p, D_p, D_n) = \mathrm{support}_G(p, D_p) * \frac{\mathrm{gr}_G(p)}{1 + \mathrm{gr}_G(p)}$$

Let p and q be two itemsets, such that $p \supseteq q$. In a given group G, the following monotonic properties hold between their contrast intensities:

- if $support_G(p, D_n) = support_G(q, D_n)$, then $\mathrm{CI}_G(p, D_p, D_n) \leq \mathrm{CI}_G(q, D_p, D_n)$
- if $support_G(p, D_p) = support_G(q, D_p)$, then $\mathrm{CI}_G(p, D_p, D_n) \geq \mathrm{CI}_G(q, D_p, D_n)$
- if $support_G(q, D_p) = 0$, then $\mathrm{CI}_G(q, D_p, D_n) = 0$ and $\mathrm{CI}_G(q, D_p, D_n) = \mathrm{CI}_G(p, D_p, D_n)$

3.1 Group Discriminative Contrast Patterns

In this sub-section, we will formally define **Group Discriminative Contrast (GDC) Patterns**, which correspond to subspaces that have strong contrast intensity between classes in one group, but they have relatively weaker contrast intensity in the other group. Firstly though, we define the following measurement for measuring how much stronger a subspace is for differentiating the positive and the negative class in one group than it is for differentiating the classes in the other group. We refer to the first group as the *primary group*, and the latter as the *secondary group*.

DEFINITION 2. *Let G_1 and G_2 be two groups of classes, where G_1 is the primary group and G_2 is the secondary group. Let D_{p_i} and D_{n_i} be the positive and the negative class in group G_i, respectively. The* **group-discriminating power** *of a pattern p, denoted* **gCIDiff**(p, G_1, G_2)*, is the difference between the contrast intensity of p in group G_1 and its contrast intensity in group G_2.*

$$\mathrm{gCIDiff}(p, G_1, G_2) = \mathrm{CI}_{G_1}(p, D_{p_i}, D_{n_i}) - \mathrm{CI}_{G_2}(p, D_{p_i}, D_{n_i})$$

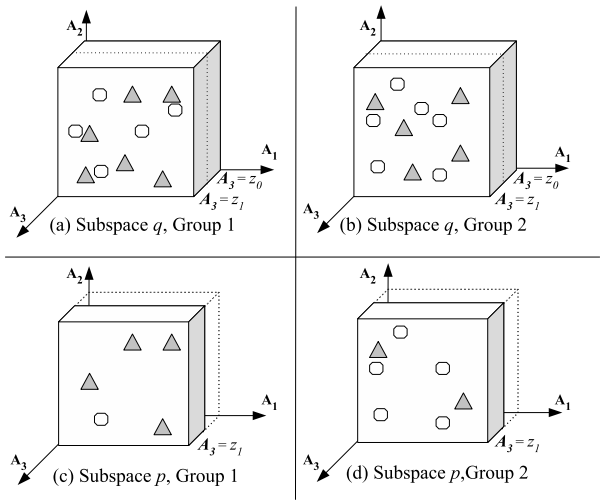


Figure 2: Subspace q is the generalization of subspace p in dimension A_3, where $p_{A_3} = z_1$. A triangle represents a positive instance, a circle represents a negative instance, in the specific group

EXAMPLE 5. *Figure 2 (a) shows a subspace q in the primary group G_1, which contains 6 positive instances and 5 negative instances. Suppose the total number of positive and negative instances in each group, respectively, is 10. Hence, we can calculate $CI_{G_1}(q) = 0.6 * \frac{0.6}{0.6+0.5} = 0.33$. Figure 2 (b) shows the same subspace in the secondary group, G_2, which contains 5 positive and 6 negative instances. $CI_{G_2}(q) =$ 0.23. The group discriminating power of q, i.e.* gCIDiff*, is 0.10, which shows that the contrast intensity of q between the positive and the negative class in group G_1 is larger by 0.10 from its contrast intensity in group G_2.*

An itemset with a positive group discriminating power corresponds to a subspace in which the contrast between the positive and the negative class in the primary group is stronger than the class-contrast in the secondary group. If the difference of its contrast strength exceeds a given threshold, then we call that itemset a Group Discriminative Contrast pattern, formally defined as follows.

DEFINITION 3. *Let p be a subspace that corresponds to an emerging pattern in group G_1. Given a positive minimum threshold, δ_{gdc}, p is a* **group-discriminative contrast (GDC) pattern** *with respect to the primary group G_1, if its group discriminating power is at least δ_{gdc}, i.e.* $\text{gCIDiff}(p, G_1, G_2) \geq \delta_{\text{gdc}}$.

3.2 Group Discriminative Contrast Influential Attributes

In this sub-section, we define a measurement, **group discriminative contrast influence**, for measuring the responsibility (or influence) of an attribute in the set of group discriminative contrast (GDC) patterns. To give an analogy, a subspace can be seen as a window which captures the contrast intensity between the classes in each group based on the instances which are contained in that subspace. The attributes whose values are specified in the pattern correspond to the dimensions of the frame of that window. As one dimension is removed from, or added to a window, i.e. the value of an attribute is generalized or specified, its contrast intensity may change due to the increase or decrease in the relative number of positive and negative instances in the new window. Such an increase of contrast intensity can thus be used for measuring the responsibility (influence) of an attribute in a particular pattern, which may vary between different patterns and different groups. Moreover, an attribute has some influence in a pattern only if one of its domain values is contained in the pattern, i.e. the pattern depends on that attribute.

Hence, we formulate the following requirements for defining the scoring function that measures the group discriminative contrast influence of an attribute: i) aggregates the attribute's influence across all GDC patterns. ii) for each pattern, measures the attribute's contrast influence in the primary group. iii) for each pattern, measures the attribute's difference of contrast influence between the groups. We refer to the influence of an attribute in a GDC pattern as its *local influence*, and the group-discriminative contrast influence of an attribute, or the influence of an attribute across all the GDC patterns, as its *global influence*. In the remainder of this section, we use the general term *pattern* for referring to a GDC pattern, unless stated otherwise.

DEFINITION 4. *Given an attribute A, and a pattern p such that p is A-dependent, the* **local influence** *of A in p, denoted localInfluence(p, A, G_1, G_2), measures the attribute's group discriminative contrast (GDC) influence locally in subspace p. Given the set of all A-dependent GDC patterns, S_A, the* **global influence** *of A, globalInfluence(S_A, A, G_1, G_2), aggregates the GDC influence of A across all patterns in S_A, i.e.* $\text{globalInfluence}(S_A, A, G_1, G_2) =$

$$\sum_{p \in S_A} \text{localInfluence}(p, A, G_1, G_2)$$

If *localInfluence*$(p, A, G_1, G_2) < 0$, we say that attribute A has a negative influence in p, as it shows that the inclusion of attribute A weakens the group discriminating power of p. If *globalInfluence*$(S_A, A, G_1, G_2) > 0$, then A is a **Group Discriminating Contrast Influential Attribute**, or GDC Influential Attribute for short, which means that A-dependent patterns exist, and the inclusion of dimension A strengthens the overall group discriminating power of those patterns.

An attribute's local contrast influence in a group: We now describe the measurement of the local influence of an attribute in a subspace, based on the following definition of *contrast influence*, that measures the gain in the contrast intensity of the subspace, as a result of including that attribute in its dimensions.

DEFINITION 5. *Given group G. Let D_p and D_n be the positive and the negative class in G. Let A be an attribute, p be an A-dependent pattern, and q be its A-generalization. The* **local contrast influence** *of A in p is the contrast intensity gained from its A-generalization.*

$$\text{CIGain}_G(p, A, D_p, D_n) = \text{CI}_G(p, D_p, D_n) - \text{CI}_G(q, D_p, D_n)$$

In the given group, a positive (resp. negative) CIGain of attribute A in subspace p shows that specifying the value of attribute A in p strengthens (resp. weakens) the class-discriminating ability of subspace p. This can be used for

Input parameter(s)	Appropriate group measurement	Appropriate between-groups measurement
A pattern	Contrast intensity (CI)	Group discriminating power (gCIDiff)
A pattern + an attribute	Contrast influence (CIGain)	Group discriminating influence (gCIDiffGain) Group discriminating influence ratio (gCIDiffGainR)
A set of patterns + an attribute		GDC influence (globalInfluence)

Table 1: Measurement categorisation according to input parameters and group applicability

measuring the contrast influence of an attribute in the primary group (satisfying requirement 2 of the scoring function). However, it is a group measurement, which does not tell us about the difference in influence of the attribute with respect to the secondary group (requirement 3 of the attribute's influence scoring function).

An attribute's local contrast influence difference between groups: The following formula measures the relative local influence of attribute A, in terms of how much group-discriminating power is gained as a result of specifying the value of attribute A in a pattern.

DEFINITION 6. *Let p be an A-dependent pattern, and q be its A-generalization. The* **group-discriminating influence** *of attribute A locally in p, denoted* gCIDiffGain*, is the gain in the group discriminating power of p with respect to its A-generalization.* $\text{gCIDiffGain}(p, A, G_1, G_2) =$

$$\text{gCIDiff}(p, G_1, G_2) - \text{gCIDiff}(q, G_1, G_2)$$

EXAMPLE 6. *Recall the subspace examples in Fig. 2. In group G_1, $CI_{G_1}(p, D_{p_1}, D_{n_1}) = 0.42$ and $CI_{G_1}(q, D_{p_1}, D_{n_1}) = 0.33$. Thus, attribute A_3 has a positive contrast influence of 0.09 in p. In group G_2, attribute A_3 has a negative contrast influence of -0.17. Thus, the group-discriminating influence of A_3 is 0.50, i.e. $gCIDiffGain(p, A_3, G_1, G_2) = 0.33 - (-0.17) = 0.50$, which shows that the inclusion of attribute A_3 in p increases the between-groups difference of its ability to capture contrast between the classes.*

Furthermore, $\text{gCIDiffGain}(p, A, G_1, G_2)$ also measures how much larger is the contrast influence of attribute A in the primary group than its contrast influence in the secondary group, locally in subspace p. Re-writing gCIDiffGain() in terms of the contrast intensities of the subspaces, we have $\text{gCIDiffGain}(p, A, G_1, G_2)$

$$\begin{aligned} &= \text{CI}_{G_1}(p) - \text{CI}_{G_2}(p) - \text{CI}_{G_1}(p \setminus A) + \text{CI}_{G_2}(p \setminus A) \\ &= \text{CIGain}_{G_1}(p, A) - \text{CIGain}_{G_2}(p, A) \end{aligned}$$

Note: $\text{CI}_{G_i}(p)$ refers to $\text{CI}_{G_i}(p, D_{p_i}, D_{n_i})$, $\text{CIGain}_{G_i}(p, A)$ refers to $\text{CIGain}_{G_i}(p, A, D_{p_i}, D_{n_i})$, where $i \in \{1, 2\}$.

Scoring function formulation: Let A_1 and A_2 be two attributes. If A_1 has a larger (resp. smaller) group discriminating influence than A_2, locally in a given pattern, the contrast influence of A_1 in the primary group is not necessarily larger (resp. smaller) than A_2. Thus, to satisfy both requirement 2 and requirement 3 of the scoring function, we further define the **group-discriminating influence ratio**, denoted gCIDiffGainR, that measures the relative between-groups difference of the influence of attribute A with respect to its influence in the primary group, locally in pattern p:

$$\text{gCIDiffGainR}(p, A, G_1, G_2) = \frac{|\text{gCIDiffGain}(p, A, G_1, G_2)|}{\text{CIGain}_{G_1}(p, A)}$$

Note that the absolute value of the group discriminating influence is used in gCIDiffGainR() to preserve the positive/negative sign of the attribute's influence in the primary group. Using this measurement, attribute A_1 is more influential than attribute A_2 if the between-groups difference of contrast influence of A_1 is larger than A_2, relative to their respective contrast influence in the primary group.

Finally, we can re-write the global group discriminative contrast (GDC) influence, given an attribute A, and a set S_A which contains A-dependent GDC patterns, as: $\text{globalInfluence}(S_A, A, G_1, G_2) =$

$$\sum_{p \in S_A} \text{gCIDiffGainR}(p, A, G_1, G_2)$$

A positive global influence indicates that an attribute has helped strengthening the overall group discriminating power of the GDC patterns. Hence, such an attribute is a key factor in the contrast behaviour of those patterns. To find a **ranking of GDC influential attributes** we sort the attributes so that the attribute with the largest score of global GDC influence is the most-influential attribute. Some of the GDC patterns used for measuring the global influence may correspond to overlapping subspaces. Overlaps cannot be straightforwardly eliminated, since all GDC patterns may potentially affect the contrast intensity of the subspace, as well as the influence of an attribute in that subspace. It is worth noting that overlaps are not necessarily problematic though, since classifiers based on emerging patterns allow overlaps, but have still proven extremely successful (e.g. [3]). More sophisticated techniques for handling overlaps are beyond the scope of this paper.

Table 1 shows the characteristics of each measurement defined in this section. The contrast intensity and group discriminating power depend on only a single pattern, the contrast influence and the group discriminating influence depend on a pattern and an attribute. In terms of the group-dependency, the contrast intensity and contrast influence are within-group measurements as they depend on a single group, whilst the group discriminating power and the group discriminating influence are between-groups measurements.

4. MINING ALGORITHM

This section introduces our algorithm, called *mineGDC*, which finds group-discriminative contrast (GDC) patterns and their influential attributes. Before we describe our algorithm in detail, let us consider a naive algorithm that consists of three steps: i) find all the emerging patterns (EPs) in the primary group; ii) apply the GDC constraint on those patterns, i.e. calculate their contrast intensities; iii) for each attribute, find the dependent patterns and calculate the attribute's influence in those patterns. This naive mining approach can suffer from significant redundancy, because not all of the EPs satisfy the GDC constraint, and many patterns depend on several attributes. Our technique integrates

those three steps and finds the GDC patterns while simultaneously calculating the influence of each attribute.

Mining challenges: Our mining task is challenging due to three reasons: i) It explores both the pattern space (for finding the GDC patterns) and the feature space (for finding the GDC influential attributes). Since the feature space has $O(n)$ search space, where n is the number of features, and the pattern space has $O(2^n)$ search space, performing the search in both spaces can be space and computationally expensive. None of the existing pattern mining or feature selection techniques deal with both search spaces simultaneously, which is a noteworthy feature of our algorithm. ii) Mining patterns with a GDC constraint is challenging because it depends on relative measurements of contrast intensity differences and ratio across groups. Such a constraint cannot be easily handled using the existing contrast pattern mining techniques, such as in [10], as they can only handle one positive and one negative class. iii) The influence scoring function for an attribute depends on the contrast intensity differences between each pattern and its generalization, and also the ratio of such differences across groups, which increases the computational complexity.

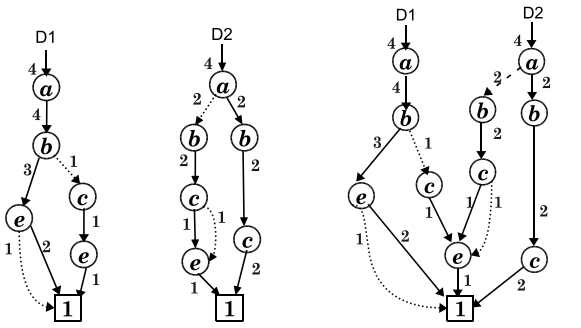
We address those challenges using a compact and efficient database representation, called a Weighted Zero-suppressed Binary Decision Diagram (WZDD) [11], which is a directed acyclic graph (DAG) data structure and has previously been used for efficient frequent pattern mining. WZDDs are useful for mining the second-order contrast, since they allow compact representation and efficient manipulation of the multiple classes and their intermediate projections.

Overview of *mineGDC*: To give a general overview, our mining framework follows a prefix growth mechanism which is typically used in the classical mining framework for finding frequent patterns [11], and EPs [10]. It recursively grows prefixes of the patterns and projects conditional databases which contain subsets of the database which are relevant to each prefix, allowing efficient support calculation. The classical *infrequent prefix* pruning strategy for finding EPs prunes a prefix (and its supersets) if its support in the positive class is less than the minimum threshold. The existing framework cannot solve our mining challenges, since the GDC constraint depends on 4 databases (i.e. two classes from both groups). Our technique projects secondary databases for all of those four classes for each prefix. To efficiently calculate the GDC influence calculation for all attributes, the global influence score is updated as soon as a GDC pattern is found, for each attribute whose values are contained.

Based on the monotonicity of contrast intensity, if the support of an itemset in the positive class is 0, then its contrast intensity and its supersets' are also 0. Therefore, when performing the conditional database projections, we order the classes so that the negative class (from each group) is projected only if the conditional positive class is not empty.

4.1 Weighted Zero-suppressed Binary Decision Diagrams (WZDDs)

A Weighted Zero-suppressed Binary Decision Diagram [11] (WZDD) is a DAG of labeled nodes and weighted edges. It consists of one source node, multiple internal nodes, and two sink nodes, sink-0 and sink-1, respectively. Each internal node has two child nodes, and it may have multiple parent nodes. The nodes are ordered so that the label of a node's children must be of higher index than the parent node. An internal node N with label x, denoted $N = node(x, N_x, N_{\overline{x}})$, encodes the set of itemsets S, s.t. $S = (x \times S_x) \bigcup S_{\overline{x}}$, where N_x encodes set S_x, and $N_{\overline{x}}$ encodes $S_{\overline{x}}$. The operation $(x \times S_x)$ denotes a set-multiplication between x and S_x. Sink-0 encodes the empty set ($\emptyset$); sink-1 encodes the set of empty itemsets ($\{\emptyset\}$). Each path to sink-1 represents an itemset in the database. The weight of N, denoted *weight*(N), refers to the weight of the edge incoming to N which represents the total support of the itemsets in S.



(a) D1 (b) D2 (c) Identical nodes are shared across D1 and D2

Figure 3: Examples of WZDD databases; D1 = {*abe*(2),*ace*(1),*ab*(1)}, D2 = {*bce*(1),*abc*(2),*be*(1)}; Var. ordering: $a < b < c < d < e$

WZDDs merge all identical nodes for ensuring that each node is unique (canonical), which is also enforced across multiple WZDDs. The efficiency of WZDD routines relies on its caching mechanism which stores the result its intermediate computations, so that the computed result can be re-used if the computation is re-visited. Fig. 3(a) and (b) show two WZDDs which represent two databases D1 and D2, that contain itemsets (with their support values): D1 = {*abe:2, ace:1, ab:1*}, D2 = {*abc:2, bce:1, be:1*}. The merged WZDDs (in Fig. 3(c)) share the bottom node e. Solid lines link each node N to N_x, and the dotted lines to $N_{\overline{x}}$. Sink-0 nodes are omitted from the illustrations in this paper.

4.2 Mining Second-Order Contrast With WZDDs

We will describe our mining algorithm as shown in Algorithm 1. The input databases correspond to the classes from both groups, represented as WZDDs. The positive class in the primary group serves as the *pattern generator*, since prefixes of the patterns are prefixes of the itemsets in this class. Prefixes are recursively grown using the item in the top node of the pattern generator. For each prefix, conditional database projections are performed for each input database, using the WZDD routines defined in [11] (line 3). Once the current prefix item and its supersets have been processed, the databases are reduced (line 4) to remove the current item. Let x be the top-item which belongs to the domain values of attribute A. Patterns which contain x are found from the conditional databases (line 5-6), all of which are A-dependent. Thus, the influence of attribute A can be incremented immediately after all patterns that contain x have been found (line 7-8), allowing the influence of all the attributes to be found as the algorithm returns. Detailed explanation about the influence calculation will be given later.

Once all patterns that contain the top-item have been found, other prefixes are grown from the reduced databases (line 9-10). The recursion terminates when the longest pre-

Algorithm 1 Mine GDC patterns and calculate the GDC influence of each attribute

mineGDC(D_{gen}**,** $[Dn_1, Dp_2, Dn_2]$ **)**
D_{gen} : the generator data set, corresponds to the positive class in group G_1
Dn_1: the negative class in group G_1
Dp_2, Dn_2: the positive and negative classes in group G_2.
All inputs are represented as WZDDs
1: x = the label of the top-node in D_{gen}.
2: D = a class in $[D_{gen}, Dn_1, Dp_2, Dn_2]$
3: $D_{(x)}$ = x-conditional of database D.
4: $D_{(\overline{x})}$ = reduce database D by item x.
5: Grow prefixes which contain item x:
6: res_x = **mineGDC**$(D_{gen(x)}, [Dn_{1(x)}, Dp_{2(x)}, Dn_{2(x)}])$
7: Update attribute influence from item x and the x-conditional
8: GDC patterns: **calcInfluence**$(x, res_x, G_1, G_2, G_{1(x)})$
9: Grow prefixes which do not contain item x:
10: $res_{\overline{x}}$ = **mineGDC**$(D_{gen(\overline{x})}, [Dn_{1(\overline{x})}, Dp_{2(\overline{x})}, Dn_{2(\overline{x})}])$
11: Build the output node : $result$ = node$(x, res_x, res_{\overline{x}})$
12: **return** $result$
13: **Terminal cases:**
14: Case 1: D_{gen} contains an empty itemset
15: $pref$ = the prefix which projects the current data sets.
16: Check GDC-constraint on $pref$, using its support*
17: in each class
18: **if** ($pref$ is a GDC pattern) **then return** sink-1
19: **else return** sink-0
20: **end if**
21: Case 2: D_{gen} is empty: **return** sink-0

(*): support($pref$,D) can be calculated using the WZDD routine: weight(D_{pref}), where $pref$ is an itemset and D_{pref} is its conditional database.

Algorithm 2 Calculate the GDC influence of item r in P

calcInfluence$(r,\ P, [G_{1(r)}, G_{2(r)}, G_1, G_2])$:-
P: GDC patterns and their $gCIDiff$ values
$G_{1(r)} = [Dp_{1(r)}, Dn_{1(r)}]$: r-conditional databases from group G_1
$G_{2(r)} = [Dp_{2(r)}, Dn_{2(r)}]$: r-conditional databases from group G_2
$G_1 = [Dp_1, Dn_1]$: r-reduced databases from group G_1
$G_2 = [Dp_2, Dn_2]$: r-reduced databases from group G_2
All inputs are represented as WZDDs
1: x = the label of the top-node in Dp_1.
2: G_i = a group in $[G_{1(r)}, G_{2(r)}, G_1, G_2]$
3: $G_{i(x)}$ = x-conditional of databases from Dp_{G_i} and Dn_{G_i}
4: $G_{i(\overline{x})}$ = reduce Dp_{G_i} and Dn_{G_i} in group G_i by x
5: Calculate influence of r in patterns which contain x:
6: **calcInfluence**$(r,\ P_x, [G_{1(r.x)}, G_{2(r.x)}, G_{1(x)}, G_{2(x)}])$
7: Calculate influence of r in patterns which do not contain x:
8: **calcInfluence**$(r,\ P_{\overline{x}}, [G_{1(r.\overline{x})}, G_{2(r.\overline{x})}, G_{1(\overline{x})}, G_{2(\overline{x})}])$
9: **return**
10: **Terminal cases:**
11: Case 1: P contains an empty itemset
12: $pref$ = the itemset that projects $G_{1(r)}$, $G_{2(r)}$
13: $pref_{gen}$ = the itemset that projects G_1, G_2, i.e. the
14: A-generalization of $pref$
15: Calc. gCIDiff for $pref$:
16: $gCIDiff_{spec}$ = CI($pref, G_{1(r)}$) - CI($pref, G_2(r)$)
17: Calc. gCIDiff for $pref_{gen}$:
18: $gCIDiff_{gen}$ = CI($pref_{gen}, G_1$) - CI($pref_{gen}, G_2$)
19: A = the attribute whose domain contains item r
20: Update the global GDC influence of attribute A:
21: influence[A] += $\frac{gCIDiff_{spec} - gCIDiff_{gen}}{gCIDiff_{spec}}$
22: **return**
23: Case 2: P is empty:**return**

Note: CI($pref$,G_i) is calculated using the weight of the WZDD nodes of the relevant classes,

fix for a particular candidate pattern has been found (line 14), or when the pattern generator is empty (line 21). For each fully grown candidate pattern, let it be $pref$, the GDC-constraint is checked (line 16-17) based on its support in each class which is represented as the weight of the WZDD node representation. To include $pref$ in the final output node, a sink-1 is returned (line 18), which incrementally builds up the final output WZDD. Finally, the output node contains the GDC patterns found from both the x-conditional databases and from the reduced databases (line 11).

Updating an attribute's influence: The procedure shown in Algorithm 2 calculates the influence of attribute A in a given set of A-dependent patterns, given all of those patterns contain the particular item r which is an item from the domain of A. The inputs are four databases, the first two correspond to the r-conditional databases for finding the contrast intensities of the patterns, the next two databases correspond to the reduced databases which exclude item r for finding the contrast intensities of the generalized patterns. The framework is similar to the pattern growth framework, which recursively projects conditional databases for each class, but the projections are guided by prefixes of the given patterns (line 1-8). The influence calculation is performed when it finds the database projections for the longest prefix of a particular pattern (line 11-21). Using the projected conditional databases represented as WZDDs, the contrast intensity and the contrast influence of the attribute can be easily computed using the pattern's support values which correspond to the weight of the relevant WZDDs.

The efficiency of this procedure relies on the use of WZDD's caching mechanism which allows intermediate computations, such as projecting secondary databases, to be shared across functions. So, the database projections performed in calcInfluence() may re-use the cached results from the database projections performed in mineGDC(), and vice versa, which avoids redundant computations.

5. EXPERIMENTAL RESULTS AND DISCUSSION

In this section we evaluate our method and the performance of our algorithm for mining second-order contrasts. Our algorithms were implemented in C++, using the WZDD library routines developed in [11]. All experiments were conducted on two Intel Xeon 3 GHz CPUs, 4 GB RAM, running Solaris. We carried out experiments on two real UCI data sets [6]: census and satimage. The objectives of our experiments include: i) to compare the volume of GDC patterns with emerging patterns (EPs), ii) to evaluate our proposed GDC based attribute ranking by comparing it against other methods; iii) to show that meaningful contrast influential attributes can be discovered by our method; iv) evaluate the runtime performance of our mining algorithm. In the census data set, we choose to find the first-order differences between male (as positive class) and female, and the second-order differences between two race groups: 'White', and the other races (combination of all other races in the data set) which we label as 'Non-White'.

Patterns volume comparison: Figure 4(a) shows the number of patterns in the census data set using the first 20 attributes, with GDC-constraint: $\alpha = 1\%$, $\beta = 0.5\%$,

and a varying δ_{gdc}, i.e. the minimum group discriminating power. In the 'white' race group, there are 5 million EPs. When δ_{gdc} is very small, almost every EP is a GDC pattern. As δ_{gdc} increases to 0.05, the number of GDC patterns drops by roughly 10% from the EPs. We identify 17 GDC influential attributes in such a scenario (shown in Table 3a). The satimage dataset has similar trends for the pattern volume comparison (shown in Figure 4b). The number of GDC patterns can still be overwhelming, but our technique can find the attributes which help explain the second-order contrast behaviour of those patterns, which we will discuss shortly.

Ranking comparison: The census data set contains several household attributes and income attributes describing census data from the year 1970. With threshold values $\alpha = 1\%$, $\beta = 0.5\%$, and $\delta_{gdc} = 0.05$, we found 17 influential attributes for capturing group discriminative contrasts with the 'white' race group as primary group (Table 3a), out of 20 attributes which are included in our experiment. To evaluate our attribute ranking, we compare it against other rankings which are based on entropy measure [5], and the statistical Pearson's correlation measure.

The **entropy-based ranking** is based on the information gain of an attribute, which measures its ability to improve class discrimination. The columns in Table 4 show the info gain of each attribute in each group, and the info-gain difference across the groups, labeled *IGDiff*. Attributes *group-quarter-type* (i.e. the type of housing) and *marital-status* appear in the top-5 attributes in our GDC based ranking as well as in this entropy-based ranking. It shows our technique is able to identify such attributes whose male-vs-female discrimination ability is stronger in the 'white' group than their discrimination ability in the other group. *Group quarter* and *farm* attributes, which are highly ranked by entropy, however, are not identified by our method. It suggests that patterns containing those attributes have weak group discriminating contrast influence. If we look closer at their info gain differences, the values are actually very small, which means that there is no significant difference of their class discrimination ability across the groups, explaining why they are not identified by our method. Our attribute ranking identifies other influential attributes which have low ranks in the entropy-based ranking, which shows the ability of the GDC based ranking to identify the interdependency between multiple attributes, whereas an entropy measure treats each attribute independently. Later in this section, we will show a more interesting result regarding those attributes whose entropy-based ranks are lower than their GDC-based ranks.

The top-10 attributes found using a **correlation** measure are listed in Table 5. For each attribute, $Corr(g)$ is the Pearson's correlation coefficient between the values of that attribute in the positive class and the values of that attribute in the negative class in group g. A large correlation value indicates that an attribute is a poor class-discriminator in g, because its values vary closely between the classes. The score of correlation difference between groups, denoted *CorrDiff* = Corr(G_2) - Corr(G_1), measures how much an attribute correlates with the classes in the secondary group, but does not correlate with the classes in the primary group. The most influential attribute in our ranking(see Table 3a), i.e. *group-quarter-type*, has a negative correlation difference score, meaning that it is a weaker class discriminator in the primary group compared to the secondary group. Like the entropy measure, this result shows that a correlation measure does not to identify the interdependency between multiple attributes, which can be identified by our method.

A meaningful discovery: The attributes which are influential for capturing GDC patterns when the 'white' race is chosen as primary group are shown in Table 3a. The GDC influential attributes when the 'non-white' race is chosen as primary group are shown in Table 3b. Attributes that have positive global GDC influence for one race group but have zero or negative influence in the other group are marked by asterisks (*). Interestingly, the first two attributes, *group-quarter-type*, (i.e. type of housing), and *num-of-families-in-household*, are the top-2 attributes in both groups, suggesting that they have an equally high importance for finding group-discriminative male-vs-female contrast in each group.

Based on the GDC based ranking for each race group, attribute *monthly-rent* appears to be influential only for the 'white' race and not for the other race group. Since it does not appear in the top-5 attributes in the other rankings (i.e. entropy-based and correlation-based), it shows that when considered individually, it does not differentiate the male and female differences across groups. However, our ranking suggests that *monthly-rent* can help capture the group differences when it is combined with some other attribute(s) within GDC patterns. To give a specific example, we found that specifying *monthly rent* in the following rule:

'do not live with a spouse, and monthly rent > $125,'

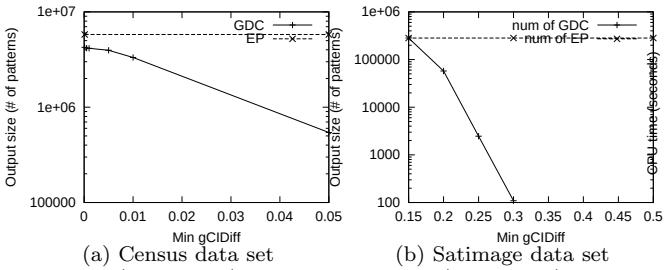
increases the between-groups difference of this rule's male-vs-female discriminating ability. Based on this rule, someone who does not live with a spouse and pays high monthly rent, is more likely to be a male in the 'white' race group, but without considering *monthly rent*, it is equally likely that the individual is a male or a female, in either race group. This example shows that a GDC influential attribute can help identify important sub-categories (corresponding to GDC patterns) in the population, such as a category of people who do not live with a spouse and pay high monthly rent, in which there is a strong differentiation between male and female in the 'white' race group but there is not a strong differentiation between male and female in the other group [1].

Time performance of mining algorithm: We measure the runtime performance of our algorithms for mining the GDC patterns and their influential attributes in the census and satimage data sets, with a varying minimum group discriminating power threshold, δ_{gdc}. Table 2 shows the class sizes in each data set. We implemented 2 algorithms: i) *naive*: a two-phase algorithm which finds the GDC patterns, then for each attribute, calculates its influence by projecting the relevant patterns. ii) *mineGDC*: the algorithm described in Sec. 4.3 which simultaneously finds the influential attributes in the pattern mining phase. Figures 5a and 5b show the runtime for each algorithm from the two datasets. When δ_{gdc} is high, there exist only a few patterns, for which both algorithms have similar runtime. As δ_{gdc} decreases, the *mineGDC* algorithm can achieve up to 4 times speed up than the *naive* algorithm, since *mineGDC* visits patterns which

[1]This rule corresponds to our intuition since it is likely that there are much more white people who pay high rent than non-white people, hence, the amount of rent is more useful for discriminating male and female in the white population than it is for the non-white population.

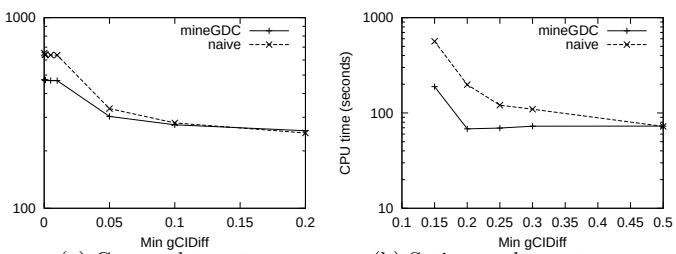
Data set	Group 1		Group 2	
	Positive class	Negative class	Positive class	Negative class
Census	White.Male (2957)	White.Female (3089)	Not-White.Male (448)	Not-White.Female (525)
Satimage	C1 (1072)	C7 (1038)	C2+C3 (1440)	C4+C5 (885)

Table 2: Class sizes in each data set



(a) Census data set (nattr = 20) (b) Satimage data set (nattr = 36)

Figure 4: Comparison between the number of GDC patterns and emerging patterns



(a) Census data set (nattr = 20) (b) Satimage data set (nattr = 36)

Figure 5: Runtime comparison between *mineGDC* algorithm and naive algorithm

Rank	Att.name	GDC influence
1	group-quarter-type	38408.1
2	num-of-families(in-household)	38408.1
3	monthly-rent*	36215.1
4	relationship-to-householder	35590.2
5	marital-status*	24987.3
6	house-ownership*	24971.6
7	mother-location(in-household)*	23022.3
8	father-location(in-household)*	15495.5
9	age-of-eldest-child(in-household)	11012.2
10	family-total-income*	7033.2
11	spouse-location(in-household) *	5756.4
12	age-of-youngest-child(in-household)	4424.0
13	num-of-fathers(in-household)	1171.1
14	family-size*	275.6
15	family-unit	119.8
16	age	119.2
17	house-value*	2.5
18	num-of-couples(in-household)	0.0
19	farm	0.0
20	group-quarter	0.0

(a) 'white' race as primary group

Rank	Att.name	GDC influence
1	num-of-families(in-household)	872.8
2	group-quarter-type	871.3
3	num-of-fathers(in-household)	315.1
4	father-location(in-household)	313.5
5	relationship-to-householder	300.0
6	num-of-couples(in-household)*	54.9
7	age-of-eldest-child(in-household)	11.6
8	age	4.3
9	family-unit	4.0
10	age-of-youngest-child(in-household)	2.3
11	marital-status	0.0
12	family-size	0.0
13	spouse-location(in-household)	0.0
14	mother-location(in-household)	0.0
15	family-total-income	0.0
16	monthly-rent	0.0
17	house-value	0.0
18	house-ownership	0.0
19	farm	0.0
20	group-quarter	0.0

(b) 'non-white' race as primary group

Table 3: Attribute ranking by GDC (global) influence (a) with 'white' race as the primary group, and (b) with 'non-white' race as the primary group; Attributes with 0 GDC influence scores do not appear in any GDC pattern in the given group; Attributes marked by (*) have positive GDC influence in at most one group

Rank	Att.name	IG(G1)	IG(G2)	IGDiff
1	marital-status	0.973	0.952	0.021
2	group-quarter-type	0.995	0.983	0.012
3	house-ownership	0.999	0.988	0.010
4	group-quarter	0.999	0.989	0.009
5	farm	0.999	0.995	0.005
6	num-of-fathers	0.999	0.995	0.004
7	num-of-families	0.999	0.995	0.004
8	monthly-rent	0.999	0.995	0.004
9	house-value	0.999	0.995	0.004
10	age-of-youngest-child	0.999	0.995	0.004

Table 4: Attribute ranking by group information gain (IG) difference; G1='white' race group, G2 = 'non-white' race group

Rank	Att. name	Corr(G_1)	Corr(G_2)	CorrDiff
1	num-of-families	-0.006	0.158	0.164
2	age	-0.005	0.099	0.104
3	mother-location	-0.007	0.086	0.094
4	family-size	-0.020	0.054	0.075
5	father-location	0.008	0.046	0.039
6	marital-status	-0.009	0.025	0.034
7	house-value	-0.018	-0.004	0.015
8	farm	-0.002	0	0.002
9	family-unit	-0.009	-0.015	-0.006
10	monthly-rent	-0.008	-0.015	-0.007

Table 5: Attribute ranking by group correlation difference; G1 = 'white' race group, G2 = 'non-white' race group

are shared by numerous attributes only once. This shows that the attribute's influence calculation can be performed efficiently using our technique, as WZDD allows multiple reuse of intermediate database projections when finding the patterns and calculating the influence of the attributes.

6. RELATED WORK

Our method for measuring an attribute's influence is pattern based, which has not been addressed in previous work. Existing feature selection techniques have a common objective of finding attributes which are most relevant to the data classification. Entropy [5] based techniques measure the class discriminating ability of an attribute independently of the other attributes. A recent work [7] proposes a correlation-based technique for finding a set of features which have high inter-correlation among themselves, and low correlation with the other features. Their method for measuring the significance of an attribute-set may be used for measuring the discriminating ability of an attribute in first-order contrast as well as second-order contrast. The difference, however, is that our model can identify subspaces, instead of the entire data space, in which second-order contrast occurs.

Group discriminative contrast patterns being identified in this paper correspond to subspaces of high class-contrast in the primary group and low class-contrast in the secondary group. Mining interesting subspaces has been previously studied for solving other data mining problems, such as in [1] for finding outliers in high-dimensional data sets. However, there has not been any work which addresses the problem of second-order differentiation. Work [13] addresses the problem of finding contrast sets, which are first-order contrasts between multiple groups (i.e. classes) of data instances.

Work in [2, 15], proposed techniques for finding (first-order) contrasts between classes. Moreover, work in [14] studies a technique for comparing frequent patterns between classes, which may be extended to comparing contrast patterns between groups. However, their method cannot identify the influence of each attribute that causes the differences, which is a novel aspect of our technique. Previous work in [4] uses χ^2-test to measure the significance of an item in the discriminating power of an emerging pattern. Such a measure may be used for measuring the attribute's local influence in a pattern in a given group of classes, but unlike our method, the χ^2 measure is independent to which class is chosen as the positive (or negative) class.

7. CONCLUSION AND FUTURE WORK

To conclude, we have introduced a method for finding attributes which are influential for capturing contrast between classes in a group, as well as the (second-order) contrast across groups, which we call group discriminative contrast (GDC) influential attributes. Our experiments showed that our method can overcome the limitation of classical attribute selection techniques, which do not take into account the inter-dependency between multiple attributes which may vary between patterns, and across groups. Using our method, moreover, an influential attribute can help explaining the key underlying factors of the contrast behaviour that is found in certain data sub-categories, instead of considering the entire data as in other classical feature selection techniques. We proposed a mining algorithm based on the use of a DAG data structure, namely the Weighted Zero-suppressed Binary Decision Diagram, which explores both the pattern space and the feature space simultaneously, and allows compact representation and efficient projections of multiple databases. For future work, variations for the attribute's influence measurements may be interesting to investigate,and exploring the usefulness of the influential attributes for solving multiple-class classification problems.

Acknowledgement: This work is partially supported by National ICT Australia. National ICT Australia is founded by the Australian Government's Backing Australia's Ability initiative, in part through the Australian Research Council.

8. REFERENCES

[1] C. C. Aggarwal and P. S. Yu. An effective and efficient algorithm for high-dimensional outlier detection. *VLDB*, 14(2):211–221, April 2005.

[2] G. Dong and J. Li. Efficient mining of emerging patterns: Discovering trends and differences. In *Proc. of KDD99*, pages 43–52, 1999.

[3] Guozhu Dong, Xiuzhen Zhang, Limsoon Wong, and Jinyan Li. CAEP: Classification by aggregating emerging patterns. *Discovery Science*, 1999.

[4] H. Fan and K. Ramamohanarao. Noise tolerant classification by chi emerging patterns. In *Proc. of PAKDD 2004*, pages 201–206, 2004.

[5] U. Fayyad and K. Irani. Multi-interval discretization of continuous-valued attributes for classification. In *Proc. of 13th International Joint Conference on Artificial Intelligence*, pages 1022–1029, 1993.

[6] S. Hettich and S. D. Bay. The UCI KDD archive [http://kdd.ics.uci.edu], 1999.

[7] M. E. Houle and N. Grira. A correlation-based model for unsupervised feature selection. In *Proc. of CIKM'07*, pages 897–900, Lisbon, Portugal, 2007.

[8] J. Li and L. Wong. Identifying good diagnostic gene groups from gene expression profiles using the concept of emerging patterns. *Bioinformatics*, 18(5):725–34, 2002.

[9] J. Li and Q. Yang. Strong compound-risk factors: Efficient discovery through emerging patterns and contrast sets. *IEEE Trans. on Information Technology in Biomedicine*, 11(5):544–552, 2007.

[10] E. Loekito and J. Bailey. Fast mining of high dimensional expressive contrast patterns using ZBDDs. In *Proc. of KDD'06*, 2006.

[11] E. Loekito and J. Bailey. Are Zero-suppressed Binary Decision Diagrams good for mining frequent patterns in high dimensional datasets? In *Proc. of AusDM'07*, pages 139–150, 2007.

[12] U. Roessner, J. H. Paterson, M. G. Forbes, G. B. Fincher, P. Langridge, and A. Bacic. An investigation of boron toxicity in barley using metabolomics. *Plant Physiology*, 142(3):1087–1101, November 2006.

[13] A. Satsangi and O. R. Zaiane. Contrasting the contrast sets: An alternative approach. In *The 11th Int'l Database Engineering and Applications Symposium (IDEAS)*, pages 114–119, 2007.

[14] J. Vreeken, M. van Leeuwen, and A. Siebes. Characterising the difference. In *Proc. of KDD'07*, San Jose, August 2007.

[15] G. I. Webb, S. Butler, and D. Newlands. On detecting differences between groups. In *Proc. of KDD'03*, pages 256–265, 2003.