Industry and Applications Paper

Finding Meaningful Contrast Patterns for Quantitative Data

Rohan Khade
George Mason University
Fairfax, VA, USA
rkhade@gmu.edu

Jessica Lin
George Mason University
Fairfax, VA, USA
jessica@gmu.edu

Nital Patel
Intel Corporation
Chandler, AZ, USA
nital.s.patel@intel.com

ABSTRACT

Contrast set mining identifies patterns that can best distinguish between two groups of data. While many machine learning models share the same goal, contrast set mining focuses on data understanding and interpretability. Most existing work in contrast set mining focuses on categorical data. In this work, we propose an algorithm that discovers contrast patterns on *mixed* data (datasets that contain both categorical and continuous attributes). Our algorithm is able to discover multivariate interactions using a supervised adaptive binning strategy. The binning strategy identifies meaningful bin boundaries in continuous attributes based on their relationships with other attributes. This in turn allows us to form better and more meaningful contrast patterns than traditional techniques that use global, pre-binning approaches. We propose various pruning strategies to reduce the search space, and show the utility of our algorithm on simulated data, several datasets from the UCI repository, as well as real manufacturing data.

ACM Reference Format:
Rohan Khade, Jessica Lin, and Nital Patel. 2019. Finding Meaningful Contrast Patterns for Quantitative Data. In *Proceedings of 22nd International Conference on Extending Database Technology (EDBT), Lisbon, Portugal, March 26-29, 2019 (EDBT 2019),* 12 pages.
https://doi.org/

1 INTRODUCTION

The work presented in this paper was motivated by a desire to reliably detect factors resulting in failure at final test during the semiconductor packaging and test process and is the outcome of a multi-year research grant funded by Intel, Corporation to develop a solution that can be applied to their manufacturing facilities. As the industry moves to more complex packages that involve complex process flows, the amount of data collected during the processing increases, and the signals being detected (as related to cause of test failures) become more diluted. At the same time, the cost of missing these signals increases, and hence there is a growing need to develop machine learning algorithms that can quickly detect the potential cause of part failures and deliver timely feedback to the engineers so that adjustments in the manufacturing line can be made to avoid generating scrap. Note that packaging can contribute up to 50% of the cost to manufacture a CPU, hence any scrap avoidance is highly desirable. As an example, during the baking stage of the manufacturing line, if the ovens are run at a higher temperature than usual, resulting in low yield, a timely notice could minimize potential loss. The behavior of manufacturing data is often predictable; however, at times there exist anomalies such as low yield for a batch. To find potential causes of this low yield, one could create a classification model comparing good chips and bad chips. Apart from a few models such as decision trees, most models are not interpretable to the user and hence non-actionable. Even though models like decision trees can be used to find explainable patterns, usually there is a single global model built for the whole dataset using a greedy strategy. To find all patterns, the user would need to build all possible trees which can take exponential time. Therefore, while decision trees are good for classification, they are not suitable if the goal is to detect patterns in the dataset. The matter becomes more complicated when we want to capture multivariate relationships between attributes (e.g. XOR data), which require more computational time.

The analysis of semiconductor manufacturing data is non-trivial because the numbers of attributes and instances are large, and the engineer needs considerable amount of time for analysis. Our intent is to learn the patterns ("contrast sets") that distinguish two groups, e.g. a normal group and an anomalous group, automatically, without external knowledge. We note that the main goal here is data understanding and exploration, rather than prediction.

Pattern mining algorithms are often used during the initial stages of the data mining process to understand relationships among features, or in a decision making stage. A major concern is displaying results that misconstrue relationships between attributes or giving incorrect insights to make decisions. For example, due to the large number of relationships between attributes to be considered, there is a high probability of discovering uninteresting or potentially spurious patterns. A large amount of existing work in pattern mining has focused on reducing the number of such uninteresting patterns, which is also one of the goals in this paper. Another concern relating to pattern mining is the time and space complexity. This can usually be reduced by either building a more compact representation of the data or pruning the search space. In this paper, we try to reduce the search space by pruning uninteresting regions.

Contrast set mining is a set of algorithms under the pattern mining paradigm to find patterns for which the supports differ significantly among groups. It is closely related to, and can be directly compared [21] to subgroup discovery and emerging pattern mining. There has been a lot of work in the area of contrast set mining [4, 14, 15, 26, 29]; however, there remain some issues that need to be addressed. First, in the manufacturing of semiconductors, many attributes of interest are continuous. Most of the existing work in contrast set mining, emerging patterns and subgroup discovery focus on improving the efficiency of the algorithms to find categorical contrasts, i.e. by reducing the search space and the number of database scans. Continuous attributes are typically handled by either computing some statistics (such as mean) that meaningfully differ among groups, or by using a binning technique as a preprocessing step and then treating the attribute as a categorical one. The latter can potentially provide more information since it identifies local patterns as ranges of values in the continuous attributes that can be actionable. Therefore, in this work we focus on binning-based approaches. A software

suite Cortana has many state of the art subgroup discovery algorithms developed. It also has an implementation of an adaptive discretization method that we compare to in the experimental section. This approach, however, is a greedy approach and may miss (local) multivariate interactions between continuous features which we often find in semiconductor manufacturing data. As will be seen later, the patterns found using these algorithms seem to be redundant and cumbersome to interpret.

Binning or discretization is a fundamental and well-studied topic in data mining. Garcia et al. [10] published a detailed survey on discretization techniques, as well as a tool (KEEL) that contains implementations of 30 popular discretization algorithms. We applied these discretization algorithms extensively on various datasets, but we were not able to find an algorithm that satisfies all our requirements. Specifically, for our application (or applications containing continuous attributes in general), the discretizer has to be able to handle **multivariate data**, be **adaptive** (local bins with respect to a subset of attributes), and **dynamic** (tightly coupled algorithm with the end goal of finding the most meaningful contrast patterns). In addition, the algorithm needs to detect not only global correlations, but (local) multivariate interactions between features. Unfortunately, existing algorithms, including those implemented in Cortana, typically miss one or more of our requirements.

Another important aspect of the proposed algorithm is to show the user the most meaningful contrasts. The authors in [13, 27] have defined patterns that will most likely be interesting to a user. More specifically, a meaningful contrast is a pattern that is not redundant, is productive and independently productive. We define what each term means in the context of contrast pattern mining and extend it to cases where the patterns have only continuous, or have mixed features.

Our contributions are summarized as follows:

(1) We propose an algorithm, SDAD-CS (Supervised Dynamic and Adaptive Discretization for Contrast Sets), to find contrast patterns for datasets containing continuous (and categorical) attributes.
(2) Our binning technique is supervised, dynamic, and adaptive, and therefore finds better quality and meaningful bins as compared to the state of the art.
(3) The binning technique detects multivariate interactions and hence higher order contrasts can be detected.
(4) We introduce several pruning strategies to reduce the search space, which also results in finding more meaningful contrast patterns.
(5) We use statistical measures to find non-redundant, productive and independently productive contrast patterns.

2 RELATED WORK

A number of work on contrast set mining have been proposed [4, 14, 15, 26, 29]. In their pioneering work [4], the authors proposed an algorithm, STUCCO (Searching and Testing for Understandable Consistent COntrasts), that finds contrast sets in groups. STUCCO employs efficient search for contrast sets based on another rule mining algorithm, Max-Miner [5]. To assess the meaningfulness of the difference in support values across groups, the authors use a chi-square test on the null hypothesis that the support value is independent of group membership. In another work [26], the authors observe that existing commercial rule-finding system, Magnum Opus [25], can successfully perform the contrast-set mining task. The authors conclude that contrast-set mining is a special case of the more general rule discovery task. The techniques discussed above are only applicable to categorical data. A good survey on contrast sets, emerging patterns and subgroup discovery algorithms is provided in [21]. The authors also discuss how the interest measures (such as difference in support and WRACC) are compatible, i.e. the interest can be used interchangeably between communities.

Techniques derived from decision tree learning can be used; however, the authors in [16, 18, 23] explain some limitations of decision-tree-based method for our application. A number of work have been proposed to find subgroups in numerical domains. The authors in [20, 23] discretize numerical data into bins to find subgroups. The algorithm is implemented in an open source tool Cortana. Such techniques typically use an initial discretization method and then merge spaces based on an interest measure. We have compared against this approach in the experimental section. An interesting algorithm described in [11] also discretizes continuous attributes into bins for the problem of subgroup discovery. The algorithm uses optimistic estimates and horizontal pruning to prune the search space. This technique heavily relies on pruning based on the top-k subgroups, since the interest measure can be updated as soon as the algorithm reaches k subgroups. Finding all initial split points (exhaustive search in [11]) is expensive but if the initial partitions are not exhaustive, e.g. frequency or entropy based, the algorithm may miss interesting patterns that occur lower down the tree due to multivariate interactions. The current trend of recent algorithms tend to use sampling and user feedback to improve efficiency and quality of rules [6–8, 17]. This is an interesting direction; however, our goal is to develop an accurate and efficient discretizer in order to find contrast patterns. These algorithms can certainly be used in conjunction with our algorithm.

Quantitative association rule mining is well-studied and could potentially be used for contrast mining. Srikant proposed a discretization technique that partitions the range of the continuous attribute into n equal-frequency partitions, and assigns the partitions to consecutive integers [22]. If the supports for any consecutive partitions fall below the *minsup* threshold, they are merged. The problem, however, is setting the initial number of partitions n and handling multivariate interactions. If n is too small, it results in large partitions and potential information loss since elements in the same partition are indistinguishable. On the other hand, if n is too large, the algorithm becomes computationally expensive. In [2], the authors find extraordinary behavior by partitioning the antecedent of the rule into bins and finding the statistics of the consequent. The algorithm cannot handle multivariate interactions between continuous attributes. The algorithm described in [24] is a bottom-up merging algorithm. It merges contiguous parts of a feature based on the improvement of an interest measure. We also use a bottom-up approach to merge spaces; however, as will be shown later, our algorithm can handle multivariate interactions between continuous features and does not need initial small bins. MVD [3] proposed by Bay is able to detect multivariate interactions for continuous attributes. This algorithm is also a bottom-up approach which merges contiguous spaces if they are not statistically different.

3 PRELIMINARIES

Let DB be a dataset with m rows $R = \{r_1, r_2, \ldots, r_m\}$ and n attributes $A = \{a_1, a_2, \ldots, a_n\}$. An attribute can be either categorical or continuous. A categorical attribute can have multiple values, i.e.

for a categorical attribute a_i having l unique values, domain(a_i) = {v_{i1}, ...,v_{il}}. For continuous attributes, its values consist of real numbers i.e. domain(a_i) = $\mathbb{R}$. An item in DB is either a value in a categorical attribute, $a_i = v_{ix}$ where $v_{ix} \in$ domain(a_i), or range in a continuous attribute, $a_i \in [v_l, v_r]$ where $v_l \leq v_r$, $v_l \in \mathbb{R}$ and $v_l \in \mathbb{R}$. Using the above definitions, we note that items in a continuous attribute can have overlapping ranges. An *itemset c* is a combination of items in *DB*. Apart from having n attributes, *DB* has an extra attribute that contains the group information for each row (instance). Let $G = \{g_1, g_2, \ldots, g_k\}$ be a set of groups. Each row belongs to exactly one group and multiple rows can be a part of a single group. If $|g_k|$ is the number of instances in group k and $count_k(c)$ is the number of rows that contain itemset c in group k then **support** $supp_k(c)$ is:

$$supp_k(c) = \frac{count_k(c)}{|g_k|} \tag{1}$$

Bay [4] formally defines contrast set mining as follows. An itemset is a contrast between 2 groups i and j if the support difference between the 2 groups is **large** and **significant**. The support difference is large if

$$supp_i(c) - supp_j(c) > \delta \tag{2}$$

and significant if

$$\chi^2_{ij}(c) < \alpha \tag{3}$$

where α and δ are user-defined parameters.

Contrasts should be non-redundant. In the context of frequent itemset mining, an itemset c is redundant if it contains a proper subset d that has the same support as i where i is an itemset of d [27]. An example given in [27] explains that any superset of itemset female, pregnant is unlikely to be interesting since female subsumes pregnant, i.e. these itemsets are functionally dependent.

An itemset c is said to be productive if for every partition d, where $d \subset c$, $supp(c) > supp(d) * supp(c \backslash d)$. Since the dataset is usually a sample of the population, statistical test such as fisher's exact test and chi-squared tests are performed on each partition of an itemset to check for significance of the product. Although this makes the algorithms computationally more expensive, it is an important step to determine productivity and finding meaningful patterns.

Another requirement is that a pattern should be independently productive. To be independently productive, an itemset should not be explained by any of its supersets apart from being productive and not redundant. Statistical tests are also performed at this step, usually as a postprocessing step.

An optimistic estimate (oe) of an itemset is the maximum possible value of an interest measure in any of the itemsets specializations[12]. If X' is the specialization of X and Int(X') is the calculated interest measure of X' then

$$Int(X') \leq oe(X) \tag{4}$$

optimistic estimates are used to prune the search space by calculating the upper bound of the children nodes of each explored node in the search tree.

Top-k pattern mining algorithms display the best 'k' patterns to the user based on some user defined interest measure. The advantage is two-fold. First, it removes the need for the user to enter a minimum threshold for the interest measure. For example, support-based pruning is the first stage of the Apriori algorithm [1] for association rule mining. Determining the best minimum support ($minsup$) is non-trivial, if $minsup$ is too high or too low, it may find too few or too many patterns, respectively. The other advantage is that it helps prune the search space even if the interest measure is not monotonically decreasing based on optimistic estimates.

We use the following strategies to reduce the search space. An itemset is pruned if (1) it does not have support over δ in any group (*minimum deviation size* pruning); (2) its expected occurrence is less than 5, since statistical tests are not significant at that level; and (3) If the optimistic estimate of the χ squared value for the itemset's children is less than the current threshold. Also, to reduce the number of false positives, the value of α is adjusted according to Bonferroni's adjustment as explained in [4].

4 QUANTITATIVE CONTRAST SETS

4.1 Methodology

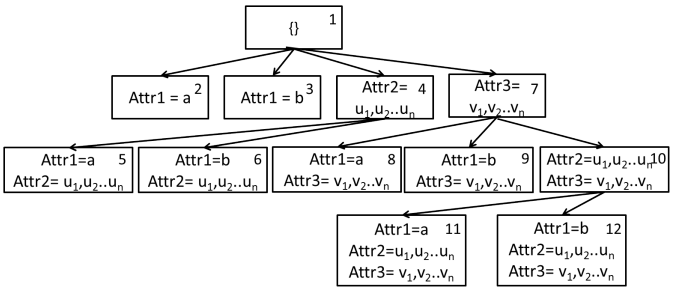

Figure 1: Search Tree for Mixed Data

To find combination of attributes (itemsets to be explored), any search algorithm such as breadth first search or depth first search can be used. Depth first search is not the preferred choice of search algorithm since it reduces the amount of pruning possible. More specifically, it may try to combine attributes which other algorithms may have found to be "non-combinable" early on. For example. if the support of a subset of an itemset is below the threshold, depth first search will not prune it. Breadth first search, on the other hand, can maximize pruning. However, the storage overhead at each level may be high. We use a search strategy [28] shown in Figure 1 since it can maximize pruning, and it requires less storage overhead than breadth first search. The figure shows how itemsets are combined, and the number on each node indicates the order in which it is explored. If an itemset contains only categorical attributes, calculating the support for each group is straightforward. When a combination containing at least one continuous attribute is encountered in the tree, our proposed algorithm, SDAD-CS, is called.

To explore contrast patterns for mixed (or exclusively continuous) itemsets, such as in nodes 4 to 12 in Figure 1, we propose SDAD-CS (Supervised Dynamic and Adaptive Discretization for Contrast Sets), a quantitative contrast-set mining algorithm. Given an itemset c containing 0 or more categorical items, and 1 or more continuous attributes $ca = \{a_1 \ldots a_n\}$ where n > 0, SDAD-CS finds itemsets that are contrasts between the groups. The contrasts found should also have the interest measure (such as difference in support) greater than the current minimum. Each contrast pattern returned should contain items from all the attributes (categorical and continuous) specified by the calling function.

The core idea behind SDAD-CS is to first divide the space (range) of a continuous attribute in a top-down fashion, calculate the interest measures and determine whether to stop searching or to explore further. After that, it merges similar contiguous spaces in a bottom-up fashion and refines the space.

Let input group DB be all the rows and columns of the dataset containing the groups of interest. Let c and ca be the categorical itemset and continuous attributes to be explored, respectively. The pseudocode for SDAD-CS is shown in Algorithm 1. The α and δ are user input parameters (α is adjusted during execution). Though not shown specifically in the algorithm, α and δ are used each time there is a check for significance and largeness, respectively. The parent measure, which is initially set to 0, stores the parent's interest measure (such as difference in support). Min support is set to the current minimum support in the current list of top-k contrasts. If the list does not have k contrasts, min support is set to δ. Starting with the top-down part, $partition(ca)$ (Line 4 of the Algorithm) divides each continuous attribute at the median or mean (we use median) into spaces. For example, a continuous attribute with a range 0 to 100 with median 35 will be divided into [0-35] and (35-100]. Next, $find\_combs(p)$ (Line 5) finds combinations of spaces between continuous attributes. For example, if there are two continuous attributes, dividing each by its median creates four rectangles (spaces) on a scatter plot. Each space together with the categorical itemset creates a candidate itemset. If *cont* is the number of continuous attributes, then the number of spaces is 2^{cont}. These spaces define our initial bin boundaries.

The algorithm iterates over each space created. It checks if the space can be pruned (Line 7). This is performed by either checking a lookup table or by performing some calculation and saving the information in a lookup table, as will be described later. We use a hash map with the itemset as the key. More space-efficient data structures such as a hierarchical hash map can be used if space is an issue. Our pruning strategies are explained in detail in a later section; however, at this point it suffices to note that a space is pruned if it is found in the lookup table.

The next step is to calculate the support of the itemset in each group in the current space r (Line 10). SDAD-CS then calculates the interest measure – in our case the difference in support (and Surprising Factor) (Line 11). The algorithm then needs to make a decision whether to explore the current space further. This is determined by calculating the optimistic estimates for the child space. If the current database contains n groups, the optimistic estimate is calculated as follows.

Let r be the current space being explored, cca_r be the itemset found at space r and $count_k(cca_r)$ be the number of instances of group k in space r. If $|g_1|$, $|g_2|$ ... $|g_n|$ are the number of instances in group *1*, *2* ... *n* respectively, then,

$$supp_1(cca_r) = \frac{count_1(cca_r)}{|g_1|} \tag{5}$$

is the support of itemset *cca* in space r in group 1. Similar definition follows for the support of the same itemset in group n.

Let *level* be the current level in the recursive tree of SDAD-CS, $|ca|$ be the number of continuous attributes, then

$$max\_instances\_child = \frac{|DB|}{2^{level+1} * |ca|} \tag{6}$$

indicates the maximum number of instances in the child spaces created by a recursive call of SDAD-CS. This comes from the fact that the continuous space is split at the median and hence distributes the data points among all child spaces equally. It should be noted that the assumption is that the data is real-valued, and each reading is unique. Some care has to be taken if the number of unique values is far less than the number of data points.

The maximum support for itemset cca_r in group 1 in any of the child spaces is

$$max\_supp\_g1 = min(\frac{max\_instances\_child}{|g_1|}, supp_1(cca_r)) \tag{7}$$

The first part of equation 7 calculates the maximum support possible in a child space for group 1. We note here that the median is calculated based on all the given instances and the groups can be imbalanced. We see that dividing the space may not reduce the supports proportionally in all groups. If the number of possible instances in the child space is greater than the number of instances in group 1, then the first value inside the 'max' function is greater than 1. This is not possible and is taken care of in the second part of the equation. Also, support is monotonically decreasing as the space reduces, and hence if the support of the current space is less than the maximum **possible** support of the child space, the maximum support of the child space is the current support. A similar argument can be made for the other groups.

We can calculate minimum support by following Eq. 6-8:

$$other\_instances\_g1 = |DB| - count_1(cca_r) \tag{8}$$

$other\_instances\_g1$ is the number of instances of the other groups apart from g1 in the current space r.

Let

$$min\_instances\_g1 = max\_instances\_child - other\_instances\_g1 \tag{9}$$

which will be negative if the majority of elements are not g1.

$$min\_supp\_g1 = max(0, \frac{min\_instances\_g1}{|g1|}) \tag{10}$$

Finally, the optimistic estimate for the child space is given by

$$oe(cca_r) = max(\forall i \forall j, i \neq j, max\_supp\_gi - min\_supp\_gj) \tag{11}$$

i,j = 1..n

If the optimistic estimate calculated is greater than the minimum support, the child spaces are recursively explored (Lines 12-13). If a better contrast pattern is found in the child space, it is added to the current list of contrast patterns (Lines 14-15), else if the current contrast pattern is large and significant, then it is either added to the current list of contrasts D or D_{temp} (Lines 16-21). The current itemset is added to D if the interest measure is greater than its parents. However, if it is not, the algorithm waits until all the spaces are explored and adds it if at least the interest measure in one space is greater than that of its parent (Lines 22-23).

After finding contrast spaces, the algorithm merges similar and contiguous spaces to get more general and comprehensible contrasts in a bottom up fashion (Lines 26-30). To merge partitions, the spaces are sorted in increasing order of size. We observe that, SDAD-CS finds fewer and more meaningful itemsets since there is more opportunity of merging smaller itemsets. If we plot the continuous attributes on a scatter plot, the spaces created by two continuous attributes is a rectangle and the size is the area of the rectangle; by plotting 3 continuous attributes the space

Algorithm 1: Algorithm SDAD-CS

Input: DB with group attribute, categorical items c in itemset, continuous attributes ca, δ, α, min support, parent measure pm
Output: Set of contrast patterns

1 **begin**
2 $D \leftarrow List\ of\ itemsets\ that\ are\ contrasts$ (Initially set to empty)
3 $D_{temp} \leftarrow List\ of\ itemsets\ that\ may\ be\ contrasts$ (Initially set to empty)
4 $p = partition(ca)$ %partition each continuous attribute at median
5 $r = find\_combs(p)$ % find all combinations of ranges found by p
6 **for** *each space in r* **do**
7 **if** $can\_prune(cca_r)$ **then**
8 Add itemset to pruned list
9 continue
10 Calculate s(cca_r) for each group
11 Calculate int(cca_r) user defined interest measure (such as difference in support) between each group
12 **if** $oe(cca_r) > min\ support$ **then**
13 D_{child} = SDAD_CS(DB_r, ca, δ, α, min support, int(cca_r))
14 **if** D_{child} *not empty and* **then**
15 Append(D,D_{child})
16 **else**
17 **if** cca_r *large and significant* **then**
18 **if** cca_r *greater than pm* **then**
19 Append (D,cca_r)
20 **else**
21 Append (D_{temp},cca_r)
22 **if** *len(D)>0* **then**
23 append(D,D_{temp})
24 **else**
25 return []
26 **if** *level==1* **then**
27 FIS = SORT(All spaces from smallest to largest)
while *No space left to combine in FIS* **do**
28 Check 2 contiguous spaces if combination is possible **if** *Comb possible* **then**
29 Combine itemsets; update contrast set
30 Return D

is a cuboid and the size is the volume of the cuboid. In general, hyper-planes create hyper-cubes and the size is *n*-volume.

Lines 28-29 loop through the spaces and try to merge contiguous and similar ones. Again, similarity is tested using a chi square test with α_r and the resulting contrast is still large and significant. If the itemsets are merged, the support, *PR* (to be defined later), hyper volume and bin boundaries are updated accordingly. More specialized itemsets are deleted and the new itemset is inserted in the appropriate sorted place.

4.2 Interest Measures

The default interest measure we use in the quantitative analysis is the difference in support; however, we find that looking at the homogeneity of a space while searching can help us find interesting patterns. We define an interest measure, *purity ratio (PR)*, which describes how homogeneous the current region is with respect to the group. In general, any interest measure, such as entropy, can also be used here depending on the problem definition. Our data is highly imbalanced and working with just supports of the group eradicates this issue. For *purity ratio (PR)*, a value closer to 1 indicates that the current space contains mostly data from the same group. Suppose i and j are the groups we are contrasting, c is the itemset with discretized quantitative attributes, s_{ic} is the support of group i in the space of itemset c, we define PR as:

$$PR(c) = 1 - \frac{min(s_{ic}, s_{jc})}{max(s_{ic}, s_{jc})} \tag{12}$$

One limitation with purity ratio is that it does not take the size of the itemset involved into consideration. For example, consider two itemsets: c_1 with supports of 0.02 and 0.04 in groups i and j respectively and c_2 with supports 0.30 and 0.60. Both have equal purity ratio. However, we notice that c_2 should be considered more interesting since it covers more instances. On the other hand, difference in support has another issue. Suppose we find two itemsets: c_1 with supports of 0.9 and 0.8 in groups i and j respectively and c_2 with supports 0.20 and 0.10, and we notice that both have similar support difference. However, c_2 (for our application) is more interesting, i.e. given the contrast c_2 the likelihood c_2 to be in group i is double that of j but there is almost a equal likelihood for c_1 to be from either of the groups. To overcome this, we define **SurPRising Measure**:

$$SurprisingMeasure(c) = PR(c) * Diff(c) \tag{13}$$

By multiplying difference in support (Diff) in each group to purity (PR) it takes the size of the contrast into consideration while giving equal weights to both groups.

The optimistic estimate for Surprising Measure is the same as Equation 11, since in the best case, PR will always be 1 in any partition (PR = 1 if there is only one instance in a partition).

4.3 Pruning

For itemsets containing only categorical attributes, we use the same pruning methods as in [4], i.e. minimum deviation size, expected value and chi-square bounds. This can be directly applied to itemsets containing both continuous and categorical or only continuous items once the bins are formed. Apart from the above technique, we try to prune redundant contrasts. An itemset is redundant if the support of the itemset is equal to the support of one of its subsets. The rationale behind this can be explained using an example. Consider an itemset $\{female\ \&\ pregnant\}$. Female subsumes pregnant i.e. the support of $\{female\ \&\ pregnant\}$ is equal to support of $\{pregnant\}$. Any contrast that is a superset of $\{female\ \&\ pregnant\}$ is likely redundant.

If an itemset is redundant, the support difference will be the same as its ancestors. Not expanding this itemset will reduce the number of redundant contrasts and search space. We note that the itemset should be redundant in all groups. In many real world datasets, there might be missing values, or incorrectly entered values. In addition, highly correlated features also tend to have many redundant contrasts. Hence, we loosen the requirement of

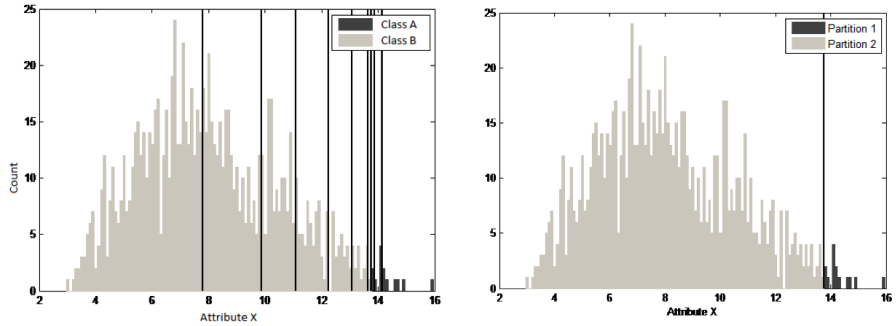

Figure 2: (Left) Vertical lines: all splits before merging. (Right) Final result after merging.

total subsumation and test whether the difference is statistically the same.

The datasets tested upon are samples of the population, and hence to make decisions for the population, statistical tests are needed. To check if two itemsets have statistically the same differences in support in the population, we use the central limit theorem. We choose this because we can assume that the difference in support for multiple samples of the population tend to follow a normal distribution. Extending the definition of central limit theorem to difference in support, it states that "Given no other samples, the best approximation of the mean of the difference in support for the population is the difference in support in the current sample."

Let α be the significance level, $|gx|$ and $|gy|$ be the sizes of groups x and y, respectively, $diff_{curr}$ be the current difference in the groups, $diff_{subset}$ be the difference of the subset, and $supp_x(c)$ and $supp_y(c)$ be the supports of itemset c in group x and y, respectively. For each subset, we calculate the bounds of the difference $diff_{bound}$. Let

$$a = \frac{supp_x(c) * (1 - supp_x(c))}{|gx|} \tag{14}$$

and,

$$b = \frac{supp_y(c) * (1 - supp_y(c))}{|gy|} \tag{15}$$

$$diff_{bound} = diff_{subset} \pm \alpha * \sqrt{a + b} \tag{16}$$

If $diff_{curr}$ is within the range of $diff_{bound}$, the difference support for the current itemset is statistically the same as its subset and hence may not be interesting and is pruned. Itemsets that are supersets of the current itemset will also not be meaningful.

Another case for redundancy for contrast patterns would be if there is a contrast found with support = 0 in a group but greater than δ in the other, then adding another item to the itemset may result in a redundant itemset. Extending this to itemsets containing continuous items, and looking at our definition of PR, we notice that when $PR = 1$ in a space, only one group is present in that space. Adding another item to the itemset would result in redundant contrasts. For example, consider a dataset containing attributes height and current country with groups toddler and adult. Consider we find a contrast height ∈]60,75] (inches) has support(adult)=0.8 and support(toddler)=0. Now adding current country to height may also result in a large and significant contrast, but it is clearly redundant between these groups.

A contrast pattern 'c' is productive if for every subset 'a' and 'c\a',

$$diff_c > supp_x(a) * supp_x(c \setminus a) - supp_y(a) * supp_y(c \setminus a) \tag{17}$$

if |g_x| > |g_y|.

If $diff_c$ is less than the product on the right-hand side of the equation for even one of the subsets, the contrast is clearly not productive. However, if it is greater, a statistical test is needed to confirm if it is indeed productive. We use chi-square test to check productivity. It should be noted that this formula is related to leverage in association rule mining which checks statistical dependence between variables.

At the end of the mining process, a check is performed to see if the contrasts are independently productive. Independently productive itemsets are meaningful, independent of their children or ancestor itemsets. For example, consider a dataset with two groups of days when a hurricane "develops" and "not develops," and a user wants to study the differences in the groups. There are a few necessary conditions for a hurricane to develop, e.g. temperature of water > 80 degrees Fahrenheit, depth of water > 200 feet and low wind shear. Considering these 3 features, the number of contrasts which will be found is 7. However, the only contrasts that the user might be interested in are the ones with all 3 conditions in it. Independently productive patterns provides the users with a compact set of patterns which are likely to be meaningful.

To check whether an itemset is independently productive, a check is performed on each superset of the itemset present in the final list. For example, consider itemset $\{A \& B\}$ and itemset $\{A\}$ in the list of contrasts found. Let r(A) be the indices of rows that item A is present, r(B) be the indices of rows that item B is present and r(A ∩ B) be the indices of rows that item A and item B are present. Now if itemset $\{A\}$ is independently productive then rows r(A) - r(A ∩ B) should also be a contrast, otherwise the contrast is found only because of itemset $\{B\}$. To check whether an itemset is a contrast, a chi-square test is performed to check significance difference in the groups. We note that an itemset may have multiple supersets and the check is performed only on supersets present in the final list. It is easy to see why this is the case by simple observation. If the superset is not a contrast, then it cannot be the case that the other features present in the superset caused the current itemset to be a contrast.

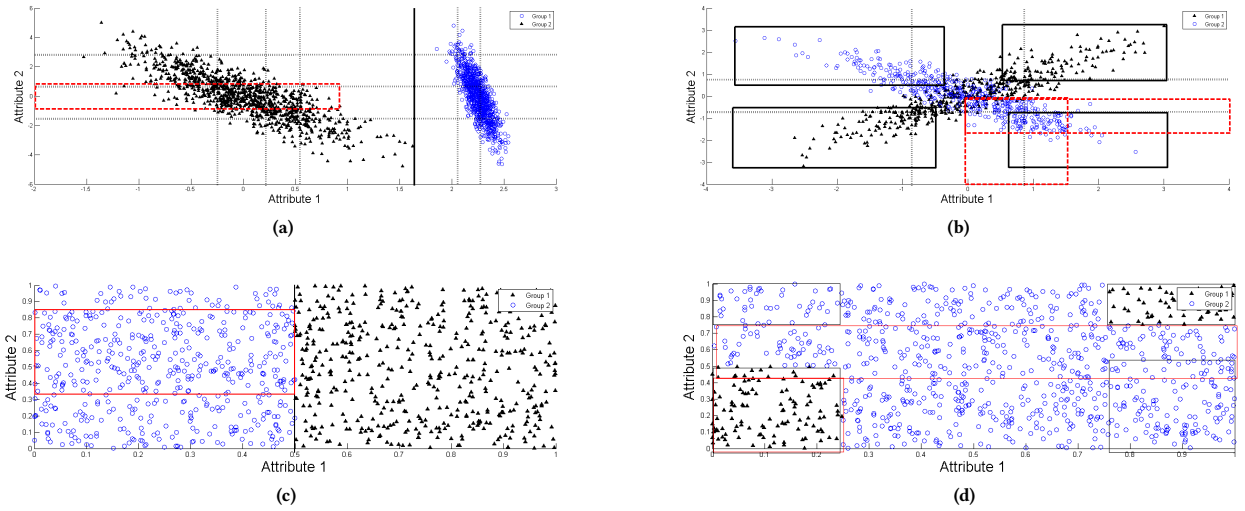

Figure 3: (a) Simulated Dataset 1 (b) Simulated Dataset 2 (c) Simulated Dataset 3 (d) Simulated Dataset 4

4.4 Example

In Figure 2 we show an example of discretizing an itemset $c = \{X\}$, where X is a continuous attribute. Let G be the group attribute with value "A" and "B". The figures shows the histograms of X, and the different shades of gray denote the two groups "A" and "B". The darker shade denotes $G =$ "A". Suppose 2% of all records belong in group "A", and the rest belong in group "B". X is first divided into two spaces at the median m, and SDAD-CS notices that PR in the left space is 1 since there are no instances of $\{X < m, Y =$ "A"$\}$. This is a pure space and does not need to be split further. In the right space, however, the PR is $1 - (48/98)/(2/2) = 0.51$ and the optimistic estimate is $1 - 23/98 = 0.76$. The algorithm continues dividing the space and the new PR becomes $1 - (23/98)/(2/2) = 0.76$. All the partitions are shown in Figure 2(Left). Spaces are then merged from smallest to largest. The final partition after merging contiguous regions is in Figure 2(Right).

5 EXPERIMENTAL EVALUATION

SDAD-CS is compared with 3 other popular algorithms, MVD, Fayyad's entropy based method [9], and Subgroup Discovery interval binning [20] implemented in the Cortana software suite.

Experimental Setup: For all experiments, initial $\alpha = 0.05$ and $\delta = 0.1$. The search tree was stunted to have a maximum of 5 levels. For the simulated dataset, we use Surprising Factor as our interest measure since it results in the best contrasts qualitatively. For the quantitative analysis, we compare all the algorithms with SDAD-CS NP (No Pruning) . This was to level the playing field since many redundant and non-productive contrasts have a high interest measure and are pruned out by our algorithm. We use mean difference in support as the interest measure since the other algorithms are not developed to optimize Surprising Factor or Purity and hence would not be a fair comparison. These experiments however show the utility of our algorithm compared to the state of the art. We also discuss the scale and effect of non meaningful patterns found by not using our pruning methods.

We compare our algorithm to MVD [3] with initial $\alpha = 0.05$ and $\delta = 0.01$ of the size of the dataset. For MVD, the datasets were initially discretized to have 100 instances per small bin as in [3]. For Fayyad's discretizer, the Group attribute is treated as the Class. For subgroup discovery using Cortana, we use the WRACC measure (equivalent to finding support difference in groups [21]) with a minimum value of 0.01 with beam search and use the 'intervals' option for continuous attributes. The other settings for Cortana include keeping the target as nominal, search width 100, maximum time to infinity, maximum subgroups to k (100 in experiments), minimum coverage to 2 and maximum coverage to the entire dataset. Although Cortana is suite of algorithms, these settings seem to be the fairest comparison to our algorithm, and from here on we will refer to these settings as 'Cortana'. For Cortana we ran the algorithms twice, once for each subgroup, and then used all the subgroups found as the contrast set. The first part of this section qualitatively analyzes some of the contrasts found, and later we quantitatively compare the algorithms. Please visit https://zenodo.org/badge/latestdoi/8891484 to access the application version.

5.1 Simulated Dataset 1

As a litmus test, we first conduct experiments on 4 simulated datasets, to check the validity of the bins found. The first simulated data consist of 2 attributes as shown in Figure 3a. The bold line indicate the bins found by SDAD-CS and the dotted lines are the bins found by MVD.

The only split point SDAD-CS finds is with Attribute 1. Since $PR = 1$ for both contrasts we cannot do "better" by adding another attribute hence we prune these spaces i.e SDAD-CS will not find a contrast between Attribute 1 and Attribute 2. Although we see that there is some interaction (correlation) between the 2 attributes, which is detected by MVD, the goal here is to find a boundary that separates the groups and there is no interest in the underlying relationships in this case. MVD misses this splitting point. The entropy based method and Cortana finds the same contrast as SDAD-CS, however Cortana also finds a bin outlined by the red box which seems meaningless.

5.2 Simulated Dataset 2

This experiment shows the algorithm's ability to find meaningful contrast patterns in multivariate data. This dataset consists of

two multivariate Gaussians in the shape of an "X" as shown in Figure 3b. Each Gaussian has a group attribute associated with it indicated by the markers.

The bin boundaries found by SDAD-CS are indicated by the rectangles in Figure 3b. We note that there is no rule found when we run SDAD-CS on each attribute individually. The contrasts found by MVD are similar to our algorithm. However, the entropy based method does not find any bins for this dataset. Cortana does not find the best bins which are shown by the red dotted boxes.

5.3 Simulated Dataset 3

In this experiment, we generated two variables uniformly distributed in the range 0 to 1. The only relationship in this dataset is that Attribute 1 in range 0 to 0.5 belongs to Group 2 and the rest Group 1. SDAD-CS finds contrasts at only level 1; however, Cortana finds meaningless contrasts at higher levels, shown in the red box. MVD finds similar contrasts as SDAD-CS.

5.4 Simulated Dataset 4

In Simulated dataset 4 we see interactions between attributes at level 2 of the search tree. We notice that during the search stage, there will be contrasts in the range 0 to 0.25 and 0.75 to 1 for Attribute 1 and 0 to 0.5 and 0.75 to 1 for Attribute 2. However, when the attributes are combined, the lower level contrasts are not independently productive and hence pruned by SDAD-CS. SDAD-CS finds a total of 6 contrasts. On the other hand, Cortana misses the top right contrasts and finds some meaningless regions shown in the middle of the figure.

5.5 Adult Dataset

5.5.1 Analyzing Bin Boundaries for Numeric Features. In this section, the differences in the contrasts found by the 5 algorithms on the Adult census dataset from the UCI Machine Learning repository [19] are shown. This experiment is a comparison between the 'Doctorate' and 'Bachelor' groups. We focus on Age and hours per week worked attributes since they highlight the differences between the algorithms.

Some of the quantitative contrasts found are shown in Table 1. Figure 4 shows the group support and the *PR* in each equi-frequency bin for Age and hours-per-week. The labels on the X-axis denote the bin boundary used and the Y-axis denotes the group support.

Looking at the contrasts found by SDAD-CS, we observe strong contrasts in the ranges 19–26 and 47–90 of the age attribute when we use PR as the interest measure to optimize. Looking at Figure 4a, we notice that, in the range 27–45, the supports for both groups are similar and hence it has a low *PR*. However, in the other ranges, there is clearly a dominant group. Similarly, in the range 50–100 for the hours-per-week attribute shown in Figure 4b, we see the majority belonging to the Doctorate group. The Bachelor group usually work less than 40 hours per week. The fifth contrast pattern discretizes {age, hours-per-week} which produces a better contrast (higher purity) than the contrasts found in the lower-order ones. This suggests that there is a multivariate relationship between these 2 attributes. We also notice the bin boundaries of {age, hours-per-week} change as compared to when they are discretized independently. This contrast shows that a global discretization may not work.

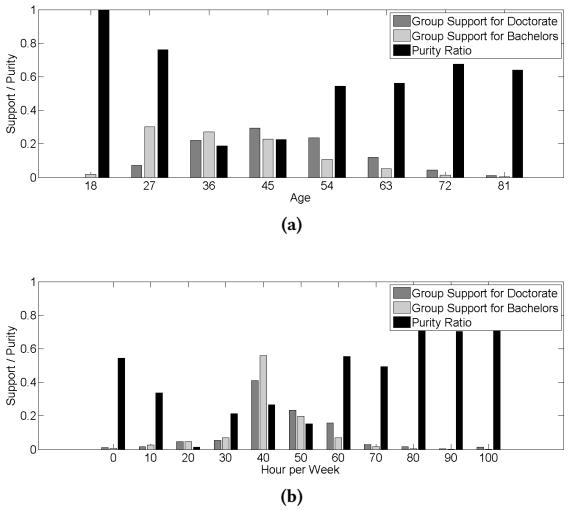

Figure 4: (a) Histogram comparing Age supports and purity ratio (b) Histogram comparing Hours per week supports and purity ratio

Cortana and SDAD-CS with support difference does not detect very good split points qualitatively. For example, in the range 19–26 in the age attribute we find only the group Bachelors. However, they find bigger bin boundaries. By looking at Figure 4a we notice the difference in supports and purity between ages 27 and 45 is small, however since the overall support in that space is much higher, the 2 algorithms find this as a large contrast. This is not surprising since the goal is to maximize the interest measure. A similar argument can be made for hours per week attribute. The contrasts found by Cortana when age and hours per week are combined are not *productive* according to the definition earlier. If we compare contrast 6 of Cortana and SDAD-CS with PR, Cortana finds a purer space. However, we also notice that in the range 19–26 of the age attribute, the support for the Doctorate group is close to 0, so *PR* is almost 1 for this contrast. Thus, we can prune this space for higher order combinations. An example of a contrast found without pruning this space is $19 \leq \text{Age} \leq 25$ and $1 \leq$ hours-per-week ≤ 40, which has support of 0 for the Doctorate group, and support of 0.10 for the Bachelors group. If this space is not pruned, SDAD would have found a purer contrast than Cortana, however, this clearly is a redundant space.

Fayyad's entropy discretizer and MVD detects level 1 interactions and finds strong contrasts, but fails to find any interaction between the attributes when combined. For the Doctorate group MVD forms a bin for Ages 48–59, which seems reasonable, however from 60–90 even though support difference is low, the purity (homogeneity) in favor of the Doctorate group is similar. Looking at the bar graph in Figure 4a, at around age 40, the support for both groups are similar, and as the age increases, we notice a higher support for the Doctorate group. MVD is not able to find the interaction between age and Hours worked

5.5.2 Analyzing top Patterns found. We now take a look at the top patterns found by Cortana (similar ones found by SDAD-CS without pruning). The setting for Cortana are as explained earlier; however, the depth of the tree was set as 2 for this discussion. The top 5 contrasts are shown in Table 3 (Cortana also displays contrasts such as $sex = Male$ *and occupation* $= Prof\ specialty$

Table 1: Contrast Sets for Adult Dataset

S.No	Contrast Set	Supp (Doc.)	Supp (Bach.)
	Contrasts Using SDAD-CS with PR		
1	$18 < Age <= 26$	0	0.16
2	$47 < Age <= 90$	0.48	0.22
3	$1 < hour\_per\_week <= 40$	0.45	0.60
4	$50 < hour\_per\_week <= 99$	0.28	0.14
5	$49 < Age <= 69$ *and* $50 < hour\_per\_week <= 99$	0.13	0.03
6	$25 < Age <= 39$ *and* $1 < hour\_per\_week <= 39$	0.11	0.26
	Contrasts Using SDAD-CS with Support Difference		
1	$18 < Age <= 39$	0.26	0.57
2	$38 < Age <= 90$	0.76	0.46
3	$1 < hour\_per\_week <= 48$	0.55	0.73
4	$40 < hour\_per\_week <= 99$	0.55	0.40
	Contrasts Using Subgroup Discovery with Cortana		
1	$39 < Age <= 80.0$	0.74	0.43
2	$-inf < Age <= 39$	0.26	0.57
3	$6 < hour\_per\_week <= 49$	0.53	0.72
4	$49 < hour\_per\_week < inf$	0.45	0.28
5	$32 < Age <= 69$ *and* $49 < hour\_per\_week < inf$	0.41	0.19
6	$-inf < Age <= 43$ *and* $6 < hour\_per\_week <= 49$	0.20	0.50
	Contrasts Using Fayyad Entropy Binning		
1	$18 < Age <= 26$	0	0.16
2	$26 < Age <= 32$	0.08	0.19
3	$46 < Age <= 90$	0.24	0.51
4	$0 < hour\_per\_week <= 55$	0.91	0.78
	Contrasts Using MVD		
1	$18 < Age <= 24$	0	0.13
2	$47 < Age <= 58$	0.32	0.15
3	$39 < hour\_per\_week <= 40$	0.30	0.43
4	$50 < hour\_per\_week <= 99$	0.28	0.14

Table 2: Datasets

Dataset	Groups	No. of instance per group	No. of Features/ Continuous Features
Adult	Bachelors/Doctorate	8025/594	13/5
Spambase	Spam/No Spam	1813/2788	57/57
Breast Cancer	Benign/Malignant	458/241	10/10
Mammography	Severe/Not Severe	445/516	5/5
Transfusion	Donated/Not Donated	570/178	4/4
Shuttle	Rad Flow/High	45586/8903	9/9
Credit Card	No/Yes	23363/6635	24/23
Census Income	Below 50K/Above 50K	187141/12382	39/11
Ionosphere	g/b	225/126	34/34
covtype	Spruce-Fir/Lodgepole Pine	211840/283301	54/10

which is clearly the same as contrast number 4 in the table and is not considered here). We notice that the top 5 contrasts has one item in common $occupation = Prof\ specialty$. The question arises if all the contrast are meaningful.

Itemsets i, ii and iii in Table 3 are singular itemsets required for calculation of the expected support for the top 5 itemsets shown as a, b and c in table. Looking at Table 3, we see itemsets 1, 4 and 5 are not meaningful since the difference in support is not statistically different from the expected difference in support. Itemset 2 is clearly redundant and functionally dependent to itemset 3. Hence, of the top 5 contrasts found by Cortana, only contrast 3 would be displayed by SDAD-CS. It should be noted that these itemsets are seeds to higher order itemsets (with 3 and more items) which further exacerbates the problem. Later on we discuss the pervasiveness of this in all the datasets we encounter.

5.6 Quantitative Analysis

In this section, we compare the mean difference in support. We compare the algorithms based on difference of support since it is shown to be compatible with WRACC [21] (they are directly proportional). It should be noted that SDAD-CS finds significantly better contrasts with respect to Surprising Factor, however, it would not be a fair comparison since Cortana is not optimized for this interest measure.

Table 3: Top Contrast Sets for Adult Dataset with Cortana

S.No	Contrast Set	Supp (Doc.)	Supp (Bach.)
	Top 5 Contrasts found by Cortana		
1	$occupation = Prof\ specialty\ and\ 28 <= Age < 80$	0.74	0.21
2	$occupation = Prof\ specialty\ and\ 19302 <= fnlwgt < 606111$	0.76	0.28
3	$occupation = Profspecialty$	0.76	0.28
4	$occupation = Prof\ specialty\ and\ sex = Male$	0.61	0.17
5	$occupation = Prof\ specialty\ and\ class = > 50K$	0.55	0.11
	Required Itemsets with 1 item		
i	$28 <= Age < 80$	0.98	0.8
ii	$sex = Male$	0.81	0.69
iii	$class = > 50K$	0.73	0.41
	Expected Supports for itemsets		
a	$occupation = Prof\ specialty\ and\ 28 <= Age < 80$	0.75	0.22
b	$occupation = Prof\ specialty\ and\ sex = Male$	0.61	0.19
c	$occupation = Prof\ specialty\ and\ class = > 50K$	0.55	0.11

Each algorithm finds different number of contrasts. To have a meaningful comparison, we only compare the top k contrasts where k is decided by the algorithm that finds the least number of contrasts or 100, whichever is smaller. The itemsets are sorted on the interest measure which is used to compare the algorithms in the experiment i.e. the itemsets are sorted in decreasing order on difference. The datasets are from the UCI repository and are shown in Table 2.

The * and - in Table 4 indicate that the distributions are not significantly different from SDAD-CS according to the Wilcoxon Mann Whitney test, and that the experiment was not able to be completed, respectively. The results indicate that on average, SDAD-CS NP finds the best results followed by Cortana. However, many of the contrast found are redundant when analyzed qualitatively. For example, in the Shuttle dataset, SDAD-CS seems to find very bad contrasts compared to Cortana. Further analysis of the patterns show that $Attr\_1$ in (-inf, 54.0] has probabilities 0.91 and 0.01 in the 2 groups respectively and $Attr\_9$ in (-inf, 2.0] has probabilities 0.77 and 0. Cortana then finds another pattern $Attr\_1$ in (-inf, 54.0] and $Attr\_9$ in (-inf, 62.0] with probabilities 0.91 and 0.01 which is clearly not improving the pattern found in the previous level. However, these strong patterns contribute towards increasing the average interest measure. Comparing the results manually indicate the SDAD-CS finds all the non-redundant contrasts. Moreover, if we restrict the algorithms to find only patterns at the first level, SDAD-CS finds stronger patterns. Additional experiments were conducted to validate our algorithm on semiconductor manufacturing data and initial results indicate SDAD-CS found the most interesting patterns qualitatively.

We compare the time cost for MVD, SDAD-CS and SDAD-CS NP in Table 5. It should be noted that the time observed is only representative and may not be an accurate comparison. The implementation standards were kept similar, however, it is possible that the algorithms could be made faster by some implementation optimizations. In general SDAD-CS explores more spaces but that may not correlate to time taken. This may be because at each space, MVD is more computationally expensive. SDAD-CS with pruning is the fastest in general.

For each dataset we show the number of Redundant, Unproductive and Independently Productive Contrasts in the top 100 patterns without applying the filter. The results are shown in Table 6. As shown in the table the majority of the contrasts may not be interesting to the user.

6 CASE STUDY: ANALYSIS OF MANUFACTURING DATA

The previous section showed the ability of our algorithm to find better contrast patterns than other state of the art algorithms, however, through this section, we show the utility of contrast pattern mining in a real world scenario. We demonstrate that contrast patterns have the capability to find insightful information in a dataset from a high-volume semiconductor packaging factory. Note that the data has been encoded and normalized for intellectual property reasons. The patterns shown here can be easily interpreted by engineers which may not be possible with other machine learning paradigms. There are many examples where we can apply our algorithm in the semiconductor manufacturing domain, such as, analyzing the difference between machines or finding contrasts between a high yield and a low yield batch.

A large amount of information is collected on a per package level as material moves through the packaging and test process. The segment of processing in the manufacture of CPUs which is of interest to us, lies between the wafer test and final test operations. Wafer test is the test performed on an entire wafer before it gets singulated and packaged. Final test occurs after the packaging process, and is used to ensure the product is going to perform as designed under specified operating conditions. The data collected are tied to the part identifier and can consist of variables that have continuous, as well as, discrete values. One has parameters that correspond to contextual information related to, for example, the sequence of equipment that processed the part, including relevant subentities (e.g. test heads, pick and place heads, oven lanes, bond heads etc.), material information, along with parametric measurement information from sensors on process tools (such as temperatures and pressures), along with parametric measurements from test, as well as, and categorical data related to device performance. The data volumes are quite substantial when one looks at the data collected across the entire manufacturing flow and to show viability of the methods presented in this paper, a limited data set was normalized and used for testing. The intent of the activity is to use the methods to quickly identify manufacturing conditions that are resulting

Table 4: Quantitative Analysis of Contrast Sets

Dataset	SDAD-CS NP	MVD	Entropy	Cortana-Interval
	Mean Support Difference			
Adult	0.26	0.16	0.18	**0.27***
Spambase	**0.60**	0.42	0.36	**0.60***
Breast	0.86	0.46	0.51	0.87*
Mammography	**0.54**	0.36	0.52*	0.43
Transfusion	0.34	0.12	0.29	**0.35***
Shuttle	0.87	0.24	0.45	**0.92***
Credit Card	**0.26**	0.17	-	0.19
Census Income	0.48	0.32	-	**0.49***
Ionosphere	**0.76**	0.43	0.35	0.75*
Covtype	**0.49**	0.41	-	0.45

Table 5: Time taken by SDAD-CS and MVD

Dataset	Time in Seconds			Number of Partitions Evaluated		
	SDAD-CS	MVD	SDAD-CS NP	SDAD-CS	MVD	SDAD-CS NP
Adult	11.11	22.92	13.28	742	171	1024
Spambase	899.97	1901.88	1909.02	121604	592	283714
Breast	0.59	3.38	1.98	72	30	376
Mammography	0.71	0.88	0.86	188	19	248
Transfusion	0.42	0.69	0.40	86	23	84
Shuttle	45.80	80.82	105.95	302	382	1018
Credit Card	441.82	873.88	639.22	12126	3260	17202
Census Income	1490.34	2256.63	4127.39	594	2566	19516
Ionosphere	960.54	983.21	1169.56	117199	122854	7371104

Table 6: Number of Meaningful Contrasts

Dataset	Count (Meaningful Contrasts)	Count (Meaningless Contrasts)
Adult	3	97
Spambase	12	88
Breast Cancer	5	95
Mammography	11	37
Transfusion	7	23
Shuttle	9	91
Credit Card	1	99
Census Income	8	92
Ionosphere	10	90
Covtype	3	97

in failures at final test to prevent generation of additional scrap material. Note that these failures are typically sporadic and the upstream signals often get diluted with increasing process complexity.

For this experiment we took a sample of the entire population and compared it with parts that failed a particular test. The data consists of 148 attributes including around 30 continuous attributes. A quick look at the contrasts indicate some information of the failing parts. These insights allow engineers to tweak or change things that are a probable cause of the failure for the test. In Table 7 we see categorical contrasts which suggest that most of the problems occur on a particular placement tool and pick head on a specific chip attach module (CAM) and most of the issues usually occur on the back row of the tray holding the parts. Both the location of the impacted parts in the trays and the specific placement tool point to a potential issue with the rear lane of the module. We also see in Table 7 that the time the impacted parts are spending above the solder liquidus temperature in the reflow oven is unusually higher. Another issue noted in 7 indicates that the average peak reflow temperature for the chips that failed the test seem to be higher as well. These results indicate an issue with the temperature control in the rear lane of the reflow oven on that specific module. With this information, feedback and changes along the manufacturing line can be made in a timely manner, including blocking any additional processing on that specific equipment/location until the issue has been addressed. Other algorithms however give a large number of contrasts and are sometimes hard to interpret and act upon.

In any practical scenario the scaling of the algorithm is very important. Apart from introducing some pruning mechanisms in the previous sections in a real world scenario the data usually does not fit in main memory. A usual way to handle this situation

Table 7: Contrast Sets for Manufacturing data

Contrast Set	Supp. Diff	Supp. (Population)	Supp. (Sample)
CAM entity-SCE	0.27	0.28	0.55
Placement tool-JVF	0.27	0.28	0.55
10.5106<=CAM peak temp std<=10.6534	0.18	0.45	0.62
67.1875<=Die temp above std<=67.2486	0.17	0.13	0.3
CAM row location -Rear	0.16	0.34	0.5
92.0373<=CAM time above liquidus<=92.8009	0.16	0.04	0.21
254.1609 <= CAM Peak temperature <= 256.8191	0.14	0.24	0.37

is by parallelizing the algorithm by using multiple machines (in a cluster). It should be noticed that SDAD-CS is run on combinations of features (itemsets) and can be run parallel of each other. Intermittent results can be used to prune the next stages. There are multiple strategies proposed in the literature of association rule mining (or search tree algorithms) to find candidate itemsets in parallel. A simple strategy is find contrast patterns at each level of the tree in parallel and then use those results to prune the next level of the tree. There is some loss of pruning of the search space across subtrees, but by using this strategy, we can treat each problem at the computing nodes as an independent problem, and use the pruning strategies discussed earlier within subtrees. The times taken to complete the experiments are 18, 106 and 225 minutes for samples containing 100000, 500000 and 1000000 instances respectively with 120 features.

7 CONCLUSION

In this paper we propose a method to find contrast sets in mixed data. Using a binning strategy that automatically determines the size and number of bins for the continuous attributes, we find meaningful contrasts even with the presence of multivariate interactions. Our algorithm is capable of finding meaningful contrasts which can potentially be more interesting to users by finding productive, independently productive and non-redundant patterns. We discuss strategies to reduce the search space. The experimental results show the utility of our algorithm in real datasets and how it finds better contrasts compared to existing techniques. The algorithm introduced in this work provides insights for analyzing data that fits in the main memory. Manufacturing data, as well as data in many application domains are very large. We discussed a way to scale up the algorithm in a parallel environment. This can be potentially used to provide more accurate and real time patterns to engineers.

REFERENCES

[1] Rakesh Agarwal, Ramakrishnan Srikant, and others. 1994. Fast algorithms for mining association rules. In *Proc. of the 20th VLDB Conference*. 487–499.

[2] Yonatan Aumann and Yehuda Lindell. 1999. A statistical theory for quantitative association rules. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 261–270.

[3] Stephen D Bay. 2001. Multivariate discretization for set mining. *Knowledge and Information Systems* 3, 4 (2001), 491–512.

[4] Stephen D Bay and Michael J Pazzani. 2001. Detecting group differences: Mining contrast sets. *Data mining and knowledge discovery* 5, 3 (2001), 213–246.

[5] Roberto J Bayardo Jr. 1998. Efficiently mining long patterns from databases. *ACM Sigmod Record* 27, 2 (1998), 85–93.

[6] Guillaume Bosc, Chedy Raïssy, Jean-François Boulicaut, and Mehdi Kaytoue. 2016. Any-time diverse subgroup discovery with monte carlo tree search. *arXiv preprint arXiv:1609.08827* (2016).

[7] Vladimir Dzyuba and Matthijs van Leeuwen. 2017. Learning what matters–Sampling interesting patterns. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 534–546.

[8] Vladimir Dzyuba, Matthijs van Leeuwen, and Luc De Raedt. 2017. Flexible constrained sampling with guarantees for pattern mining. *Data Mining and Knowledge Discovery* 31, 5 (2017), 1266–1293.

[9] Usama Fayyad and Keki Irani. 1993. Multi-interval discretization of continuous-valued attributes for classification learning. (1993).

[10] Salvador Garcia, Julian Luengo, José Antonio Sáez, Victoria Lopez, and Francisco Herrera. 2013. A survey of discretization techniques: Taxonomy and empirical analysis in supervised learning. *IEEE Transactions on Knowledge and Data Engineering* 25, 4 (2013), 734–750.

[11] Henrik Grosskreutz and Stefan Rüping. 2009. On subgroup discovery in numerical domains. *Data mining and knowledge discovery* 19, 2 (2009), 210–226.

[12] Henrik Grosskreutz, Stefan Rüping, and Stefan Wrobel. 2008. Tight optimistic estimates for fast subgroup discovery. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 440–456.

[13] Wilhelmiina Hämäläinen and Geoffrey I Webb. 2017. Specious rules: an efficient and effective unifying method for removing misleading and uninformative patterns in association rule mining. In *Proceedings of the 2017 SIAM International Conference on Data Mining*. SIAM, 309–317.

[14] Franciso Herrera, Cristóbal José Carmona, Pedro González, and María José Del Jesus. 2011. An overview on subgroup discovery: foundations and applications. *Knowledge and information systems* 29, 3 (2011), 495–525.

[15] Robert J Hilderman and Terry Peckham. 2005. A statistically sound alternative approach to mining contrast sets. In *Proceedings of the 4th Australia Data Mining Conference (AusDM-05)*. 157–172.

[16] Rohan Khade, Jessica Lin, and Nital Patel. 2015. Frequent Set Mining for Streaming Mixed and Large Data. In *Machine Learning and Applications (ICMLA), 2015 IEEE 14th International Conference on*. IEEE, 1130–1135.

[17] Rohan Khade, Jessica Lin, and Nital Patel. 2018. Finding Contrast Patterns for Mixed Streaming Data. In *Proceedings of the 21st International Conference on Extending Database Technology (EDBT),*. 632–641.

[18] Rohan Khade, Nital Patel, and Jessica Lin. 2015. Supervised Dynamic and Adaptive Discretization for Rule Mining. In *SDM Workshop on Big Data and Stream Analytics*.

[19] M. Lichman. 2013. UCI Machine Learning Repository. (2013). http://archive.ics.uci.edu/ml

[20] Michael Mampaey, Siegfried Nijssen, Ad Feelders, and Arno Knobbe. 2012. Efficient algorithms for finding richer subgroup descriptions in numeric and nominal data. In *Data Mining (ICDM), 2012 IEEE 12th International Conference on*. IEEE, 499–508.

[21] Petra Kralj Novak, Nada Lavrač, and Geoffrey I Webb. 2009. Supervised descriptive rule discovery: A unifying survey of contrast set, emerging pattern and subgroup mining. *Journal of Machine Learning Research* 10, Feb (2009), 377–403.

[22] Ramakrishnan Srikant and Rakesh Agrawal. 1996. Mining quantitative association rules in large relational tables. In *Acm Sigmod Record*, Vol. 25. ACM, 1–12.

[23] Matthijs van Leeuwen and Arno Knobbe. 2012. Diverse subgroup set discovery. *Data Mining and Knowledge Discovery* 25, 2 (2012), 208–242.

[24] Ke Wang, Soon Hock William Tay, and Bing Liu. 1998. Interestingness-Based Interval Merger for Numeric Association Rules.. In *KDD*, Vol. 98. 121–128.

[25] Geoffrey I Webb. 1995. OPUS: An efficient admissible algorithm for unordered search. *Journal of Artificial Intelligence Research* 3 (1995), 431–465.

[26] Geoffrey I Webb, Shane Butler, and Douglas Newlands. 2003. On detecting differences between groups. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 256–265.

[27] Geoffrey I Webb and Jilles Vreeken. 2014. Efficient discovery of the most interesting associations. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 8, 3 (2014), 15.

[28] Geoffrey I Webb and Songmao Zhang. 2005. K-optimal rule discovery. *Data Mining and Knowledge Discovery* 10, 1 (2005), 39–79.

[29] Gangyi Zhu, Yi Wang, and Gagan Agrawal. 2015. SciCSM: novel contrast set mining over scientific datasets using bitmap indices. In *Proceedings of the 27th International Conference on Scientific and Statistical Database Management*. ACM, 38.