Discovering Local Subgroups, with an Application to Fraud Detection

Abstract. In Subgroup Discovery, one is interested in finding subgroups that behave differently from the 'average' behavior of the entire population. In many cases, such an approach works well because the general population is rather homogeneous, and the subgroup encompasses clear outliers. In more complex situations however, the investigated population is a mixture of various subpopulations, and reporting all of these as interesting subgroups is undesirable, as the variation in behavior is explainable. In these situations, one would be interested in finding subgroups that are unusual with respect to their neighborhood. In this paper, we present a novel method for discovering such *local subgroups*. Our work is motivated by an application in health care fraud detection. In this domain, one is dealing with various types of caregivers, who sometimes specialize in specific patient groups (elderly, disabled, etc.), such that unusual claiming behavior in itself is not cause for suspicion. However, unusual claims with respect to a reference group of similar patients do warrant further investigation into the suspect associated caregiver. We demonstrate experimentally how local subgroups can be used to capture interesting fraud patterns.

1 Introduction

In this paper, we present a method for discovering local patterns in data. Our method, called *Local Subgroup Discovery (LSD)*, is inspired by Subgroup Discovery (SD) techniques [4, 8], which to some degree have a local focus, but the notion of locality plays a more important role here than in standard SD. When a dataset contains natural variations that are *explainable*, traditional SD methods will focus on these. By contrast, when relatively rare events such as fraud or terrorism are concerned, we aim to find *local deviations* within these natural fluctuations. Hence, the LSD method intends to provide interesting and actionable phenomena in the data, *within* the natural variations in the data that are not interesting to report. As we are interested in local deviations, we will use a neighborhood concept, in terms of a basic distance measure over the data.

In order to define the notion of locality, we work with a *reference group* that represents a subpopulation or neighborhood, in the context of which we want to evaluate subgroups. A *local subgroup* is a subgroup within this reference group. Such local subgroups will be judged in terms of their unusual distribution of the target attribute, compared with that of the reference group (rather than comparing with the entire population). Both the reference group and the subgroup will be defined in terms of distances from a common subgroup center. Standard quality measures from the SD literature are used to quantify how interesting a subgroup is with respect to its reference group. Figure 1 shows an example of

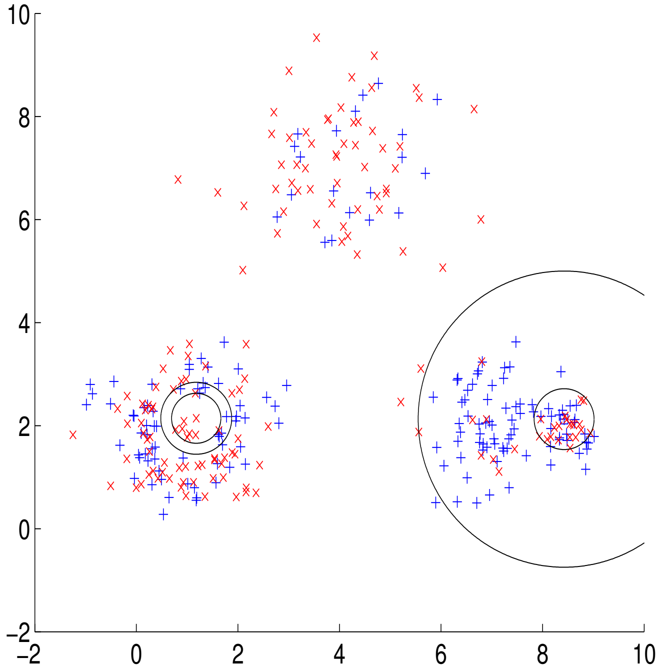

Fig. 1. Two local subgroups that are hard to find with traditional subgroup techniques. The smaller of the two concentric circles indicates the subgroup, the bigger of the two circles indicates the reference group.

two local subgroups within an artificial dataset. In SD, we are interested in finding groups that contain relatively many positive examples. The local subgroup consists of points within the smallest circle, the corresponding reference group are points within the bigger circle. Both the reference group and the subgroup are defined in terms of distances from their subgroup center. Both subgroups contain relatively many red crosses, compared to their local neighborhood.

In this paper we are interested in finding the most interesting subgroups within reference groups in a dataset. The main contributions of this paper are: defining the Local Subgroup Discovery task, presenting a new algorithm that searches for local subgroups efficiently, and showing the usefulness of the local subgroup idea in a real-life fraud detection application.

1.1 Motivation: Health Insurance Fraud Detection

The motivation for this problem comes from the field of fraud detection in health insurance data, where we are interested in identifying suspicious claim behavior of caregivers. The problem is essentially an unsupervised one, since we are not presented with any examples of known fraud cases. The goal is to identify groups of claims that are interesting to investigate by fraud investigators. We do this by comparing the claim distribution of one caregiver (typically, a high-cost claimer) with that of the remaining caregivers. The health insurance data we consider in our experiments contains information about patients and caregivers. A single record in this dataset summarizes the care (treatment, medication, costs, etc.) an individual has received over a selected period. Although the data is described in terms of patients, we are actually more concerned with an analysis of this data on the level of caregivers. After all, consistent 'inefficient' claim behavior (in the extreme case: fraud) committed by specific caregivers has a far more substantial commercial impact than such behavior committed by specific patients.

We will explain the setting with the help of Figure 1. Suppose that each point represents a patient. The dataset clearly consists of three clusters, which we can interpret as groups of patients with the same type of disease. Within these clusters the differences between patients are caused by treatment costs, variations in medication types, and so on. The approach we take in this paper is to single out an individual caregiver, and define a temporary binary column that identifies the patients of this caregiver. This column will then serve as the target in a Subgroup Discovery run, in order to find patterns in the claim behavior that distinguishes this caregiver from the others. In Figure 1, patients of the marked caregiver are represented by red crosses. Patients of other caregivers appear as blue plusses. The two subgroups (smaller circles) indicate a difference in claim behavior within a reference group of patients having the same disease type (larger circles). This is because in the subgroup, the proportion of red plusses is much higher than the proportion of plusses in the reference group. If substantial local subgroups can be found for an individual caregiver, this is a strong indication of inefficient or fraudulent practice. By repeating this process, each time focusing on a different caregiver, we can produce a list of all suspicious caregivers, along with the necessary details about the unusual behavior.

2 Related Work

Describing distributional differences between the target and non-target set is usually referred to as Subgroup Discovery [8], but also as Contrast Set Mining [1], and Emerging Pattern Mining [2]. These methods find subgroups that are interesting compared to the whole dataset, whereas the method we propose finds locally interesting subgroups. Also, in Subgroup Discovery the subgroups are usually described by conditions on the non-target attributes whereas in our application we use a distance measure to define subgroups. This is similar to epidemiology where a spatial scan statistic [6] is often used to identify regions where a certain type of disease (the target attribute) is more frequently present than in the rest of the country. Here a likelihood ratio statistic is often used to identify interesting regions. In our approach we also look for interesting regions in our data. Unlike with the spatial scan, our approach allows for any quality measure (instead of the likelihood ratio statistic) and finds reference groups together with subgroups. Calculating the quality measure on a subset of the data has been done before [7]. The subset of the data is obtained by using a distance measure. Luong et al. fix a nearest neighbor parameter k, calculate a quality measure on a part of the data based on this k, and then describe interesting regions in the dataset for which this quality measure is high. The difference with our approach is that we are interested in searching for interesting subgroups, automatically finding relevant values of the nearest neighbor parameter k.

To the best of our knowledge, this is the first approach to unsupervised fraud detection by using Subgroup Discovery to compare between entities (here these entities are caregivers, but they could be shops, cashiers, etc.). Other approaches to fraud detection are supervised methods (where fraud is labeled beforehand), and outlier detection methods. Since we do not have a labeled fraud set, and we are interested in differences on the aggregated level of claims of caregivers rather than finding single outliers, these methods are beyond the scope of the paper.

3 Preliminaries

Throughout this paper we assume a dataset D with N elements (*examples*) that are $(h+1)$-dimensional vectors of the form $x = \{a_1, .., a_h, t\}$. Hence, we can view our dataset as an $N \times (h+1)$ matrix, where each data point is stored as a row $x^i \in D$. We call $a^i = \{a^i_1, .., a^i_h\}$ the *attributes* of x^i, and t^i its *target*. We assume that each target is binary, and each vector of attributes comes from an undefined space $\mathcal{A}$ on which we have a distance measure $\delta : \mathcal{A} \times \mathcal{A} \rightarrow \mathbb{R}$.

A *subgroup* can be any subset of the dataset $S \subseteq D$. A *quality measure* $q : 2^D \rightarrow \mathbb{R}$ is a function assigning a numeric value to any subgroup. Quality measures describe how interesting subgroups are, and usually take into account the size of a subgroup (larger is better) as well as its difference in target distribution (higher proportion of the target is better).

To deal with locality we propose a distance-based approach to find subgroups and reference groups, based on prototypes. A *prototype* can be any point in attribute space $x \in \mathcal{A}$. The *distance-based subgroup* S_σ based on x for parameter $\sigma \in \mathbb{N}$, consists of the σ nearest (according to δ) neighbors of x in D. The *reference group* R_ρ based on the same x for parameter $\rho \in \mathbb{N}$ s.t. $\rho \geq \sigma$, consists of the ρ nearest neighbors of x in D. The idea is that R_ρ forms a region in input space where the target distribution is different from that distribution over the whole dataset, and we strive to find subgroups $S_\sigma \subseteq R_\rho$ in which the target distribution is different from the distribution in the reference group.

We write $S(x, \sigma, \rho)$ for the subgroup S_σ in a reference group R_ρ, which we call a *reference-based subgroup*. The prototype can be seen as the center of this subgroup, and as the center of the reference group encompassing the subgroup. A quality measure calculated for a reference-based subgroup considers only examples inside the reference group: the quality of the subgroup is calculated on a contingency table of the data, as if the reference group were the entire dataset.

$$\mathit{ranking}(x) = \{\ +, +, -, +, -, +, +, \underset{\overset{\uparrow}{\sigma}}{+}, -, -, -, +, -, \underset{\overset{\uparrow}{\rho}}{-}, \ldots\ \}$$

Fig. 2. Ranking of the target vector for a prototype x. The target vector is sorted according to the distances to x. All examples from the leftmost observation (the closest point to x) to σ are in the subgroup. All examples from the leftmost observation to ρ are in the reference group.

Given a prototype x, a distance measure δ, and the target vector t, we can obtain a ranking of the target variable (see Figure 2). This ranking is a sorted list of targets, where the leftmost point is closest to x and the rightmost point is the point furthest from x. To find the optimal reference and subgroup for a given x, we have to set σ and ρ in such a way that the quality measure is maximized. For example, let us calculate the Weighted Relative Accuracy (WRAcc) quality measure for the subgroup and reference group in Figure 2, for the parameters $\sigma = 8$ and $\rho = 14$. The WRAcc of a subgroup S with respect to target t is defined as $P(St) - P(S)P(t)$, where $P(St)$ is the probability of the target and subgroup occurring together, $P(S)$ is the probability of a record being in the subgroup, and $P(t)$ is the probability of the target being true. These probabilities are all calculated given that a point belongs to the reference group. In this example the WRAcc is thus given by: ${}^{6}/_{14} - {}^{8}/_{14} \cdot {}^{7}/_{14} = {}^{1}/_{7} \approx 0.143$. If we would set ρ to 11 instead of 14 this would lead to a somewhat higher quality: ${}^{6}/_{11} - {}^{8}/_{11} \cdot {}^{6}/_{11} = {}^{18}/_{121} \approx 0.148$.

4 Finding Local Subgroups

In this section we explain how the optimal subgroups and their reference groups are found. First we explain how we search for the most interesting subgroups with the highest quality. We also explore our approach to two problems encountered when searching for local subgroups. The first problem is how to compare qualities found on different reference groups, with different reference group sizes and different numbers of positives. The second problem is concerned with the potential redundancy in the collection of reported subgroups.

4.1 Searching for the Optimal Values

In LSD, subgroups are described by optimal combinations of the prototype x and the parameters σ and ρ. Assume we are considering a candidate prototype x. We then loop over possible values of ρ, and for each value, try all possible values for σ and calculate the quality. Per value of ρ, the highest quality obtained in this way is called the *optimal quality*, and the value $\sigma(x, \rho)$ at which this maximum is obtained is called the *optimal value* for σ, given x and ρ.

If we were to search for optimal values of the quality measures in this way, we would find that for the ranking in Figure 2, the optimal value for WRAcc would be obtained at $\rho = 3$, and $\sigma = 2$, with a WRAcc value of $^2/_9 \approx 0.222$. Unfortunately, this subgroup is not that interesting because it is quite small. Nor is it very significant; in any dataset and for any prototype, we can typically construct such a tiny subgroup and a reference group that perfectly separates positive from negative examples. This behavior of favoring very small reference groups in which we can perfectly separate positive from negative cases does not only occur for the WRAcc measure; any quality measure will suffer from this problem. Hence, guiding the search solely by high quality will not lead to interesting results. The size of the reference group should also be big enough to ensure that a subgroup with a high quality is really interesting. Hence, we need to deal with the significance of the found quality.

4.2 Significance and Interestingness

To compute the significance of a candidate x, ρ, and associated $\sigma(x, \rho)$, we use a swap randomization technique, creating a baseline *distribution of false discoveries* (DFD). This method [3] was originally designed for SD where subgroups are defined by attribute-value descriptions. We modify the method for use with distance-based subgroups, as follows. First, the target variable in the reference group is permuted, while keeping the rest of the dataset intact. Within Figure 2 this corresponds to permuting all plusses and minusses up to ρ. Since we leave the attribute space intact, all distances between examples remain the same. Next we search for the optimal quality in this permuted reference group. The optimal quality found can be considered a false discovery, because it is a discovery made on data in which the attribute space is left intact, but its connections with the targets are randomized. We can repeat this procedure to obtain more false discoveries. Together, these qualities constitute a sample of the DFD. The DFD for a quality measure thus differs for each combination of reference group size, and the number of times the target is positive in the reference group. Using this DFD we can assign p-values to subgroups having a certain quality. As [3] describes, a normal distribution can be used to estimate p-values, corresponding to the null hypothesis that a subgroup with the given quality is a false discovery.

The p-value obtained gives us a fair measure to compare qualities found for different reference groups. A low p-value indicates a low probability of finding the quality by chance. Within our approach, we compare subgroups found on different reference groups by comparing their p-value. In Section 5 we show how to search for subgroups with the lowest p-values efficiently.

4.3 Choosing Prototypes and Removing Redundancy

In the previous section, we explained how to find optimal values for σ and ρ, for a given prototype x. Since we will find optimal values for each prototype, that is each point in the dataset, this will lead to discovering many (redundant) subgroups. In this section, we describe how to find optimal (non-redundant) values for x, with the goal of presenting a concise list of subgroups to the user. To achieve this, we use a top-k approach: only the k most interesting subgroups are mined. Additionally we will use two techniques to remove redundancy from this top-k list. The first technique, based on the *quality neighborhood* of examples, will prevent redundant subgroups to enter the top-k. The second technique postprocesses the top-k to select a small group (generally 3 to 6 subgroups) as the least redundant subgroups from the top-k list.

Consider only Local Maxima Points that are close to each other in the dataset will generally have the same neighbors. When these examples are considered as prototypes, they will have similar optimal qualities, since their optimal subgroups and reference groups will strongly overlap. Given a subgroup size σ and a reference size ρ, we can compute the quality of the subgroup $S(x, \sigma, \rho)$ for each prototype x. In this way we can determine a *quality landscape* for our data. Within this quality landscape, we then look for local optima. To this end, we define the quality neighborhood $q_{\text{neighborhood}}(x, \sigma, \rho)$, of a point x. We do this by considering the set $X_x \subseteq D$ of the σ nearest neighbors of x. For each neighbor we determine the quality of its reference-based subgroup. These values form the quality neighborhood: $q_{\text{neighborhood}}(x, \sigma, \rho) = \{q\left(S(x', \sigma, \rho)\right) | x' \in X_x\}$, where $q()$ is the quality measure on the subgroup. A prototype x is a local maximum if its quality is maximal among its quality neighborhood: $q\left(S(x, \sigma, \rho\right) \geq \max q_{\text{neighborhood}}(x, \sigma, \rho)$. Prototypes that are no local maximum are considered not interesting, and will therefore not be reported.

Post-processing the Top-k There still may be some largely overlapping subgroups and reference groups of prototypes in each others neighborhood that have only a slightly different value for σ and ρ. Hence we employ redundancy removal techniques such as joint entropy and the exclusive coverage heuristic [5]. We select the combination of subgroups with the highest value for the heuristic.

5 Reduction of DFD Estimations

Distance-based methods can be computationally intensive. We use different pruning strategies to reduce the number of times the DFD has to be computed.

From all possible reference group sizes ρ for a prototype, we would like to report the optimal quality with the lowest p-value. To obtain p-values, we have to estimate the DFD. The estimation of all DFDs is computationally intensive (it requires $O(s\rho)$ calculations, where s is the sample size). In total there are n possible values of ρ so (if the DFD is not stored) it has to be recomputed n times for each prototype, where n is the number of examples. Computing the DFD is unnecessary for a subgroup that will not enter the top-k anyway. We present a user-set parameter and two pruning techniques to reduce the number of DFD calculations, and show the pseudocode.

Algorithm 1: Lowest p-Value for Prototype(db, x, $q_{optimal}$, $\sigma_{optimal}$)

```
input   : database $db$, prototype $x$, $q_{optimal}$, $\sigma_{optimal}$
output  : $pval_{best}, q_{best}, \sigma_{best}, \rho_{best}$
$q_{threshold} = -\infty$;
$pval_{best} = \infty$;
foreach $\rho$ in decreasing order do
    if isLocalMaximum$(x, \sigma_{optimal}, q_{optimal}(\rho)) \wedge (q_{optimal}(\rho) \geq q_{threshold})$ then
        compute $DFD_{x,\rho}$;
        p-value $\leftarrow DFD_{x,\rho}(q_{optimal}(\rho))$;
        if p-value $\leq pval_{best}$ then
            $pval_{best} \leftarrow$p-value ;
            $\rho_{best} \leftarrow \rho$;
            $q_{best} \leftarrow q_{optimal}(\rho)$;
            $\sigma_{best} \leftarrow \sigma_{optimal}(\rho)$;
            $q_{threshold} \leftarrow q_{optimal}(\rho)$;
        else
            $q_{threshold} \leftarrow \Phi^{-1}_{DFD}(1 - pval_{best})$;
```

Maximum value for ρ To decrease computation time, and ensure locality of the patterns at the same time, the user can set a maximum value for the reference group size.

Consider only local maxima. We can check whether a point is a local maximum before the DFD is estimated. If the point is not a local maximum, the DFD does not have to be estimated since there is a largely similar neighboring subgroup with a better quality.

Pruning the ρ-search space based on sample size. To search efficiently we prune away parts of the search space where we know that the p-value can not be lower than the one already found. If for two different reference group sizes the same quality is obtained, the quality calculated on the largest reference group will be the most significant finding, and thus have the lowest p-value.

We explain how this pruning strategy works step by step, by using the pseudocode in algorithm 1. For each prototype we keep in memory a threshold on the quality, denoted by $q_{threshold}$. We also keep in memory the optimal p-value that is found so far for this prototype, $pval_{best}$. We are interested in finding the lowest p-value for this prototype. We start by calculating the p-value for a prototype for the maximal value for ρ in the first iteration.

Next, we observe the qualities found for the same prototype, for smaller values of ρ, in decreasing order. If the optimal quality found for such a smaller reference group, $q_{optimal}(\rho)$, is lower than the threshold, we skip the estimation of the DFD and the calculation of the p-value, because we know the p-value will be lower. For smaller values of ρ that have a higher optimal quality, we compute the DFD. We also obtain the cumulative distribution function of the DFD, Φ_{DFD}, and the inverse cumulative distribution function, Φ^{-1}_{DFD}. From the

DFD we obtain the p-value of the quality found. If the newly obtained p-value is lower than $pval_{best}$, this subgroup is the best subgroup found so far. For this prototype x, the corresponding values for $\sigma_{optimal}$ and $\rho_{optimal}$ as well as its quality $q_{optimal}$, and the p-value are stored. The quality threshold is updated and is set to the quality corresponding to the new optimum. If the newly obtained p-value is higher than the one previously found on a bigger sample, these variables are not updated. The quality threshold is updated by inserting $1 - pval_{best}$ into the inverse cumulative distribution function of the DFD.

Pruning the ρ search space using a top-k approach. A subgroup only enters the top-k if its p-value is lower than the current maximum p-value in the top-k. The key idea (again) is to update the threshold each time the DFD is estimated. The only difference with pruning-per-prototype is that we can update the quality threshold by inserting the current maximum p-value of the top-k list into Φ^{-1}_{DFD} instead of inserting the minimum p-value found so far for this prototype, to obtain the new threshold.

6 Experiments and Results

Artificial Data We start by testing our method on the artificially generated data that already featured in the introduction (Figure 1). This two-dimensional data consists of 252 examples, with a roughly equal spread between positive and negative examples. 20 subgroups were obtained by considering each individual example in the data as a prototype, using Euclidean distance and WRAcc as quality measure. From these 20 subgroups, we select 3 non-redundant subgroups by using the exclusive coverage measure [5]. We compare the results with those of a 'traditional' SD algorithm which features in the generic SD tool Cortana[1]. The traditional SD algorithm describes subgroups in terms of attribute-value descriptions.

The first subgroup discovered by LSD is also found by Cortana. In Figure 1 this corresponds to the big cluster at the top. The second subgroup, found in the lower left cluster in Figure 1, is not found using traditional SD methods, because there are many subgroups from the big cluster with the same size that also have a high proportion of positive examples. The third subgroup is situated in the lower right corner. This subgroup is not detected with Cortana. This is because the density of the target variable in the subgroup is the same as the density of the target in the entire dataset, so the density of the target differs only locally.

In order to gauge the efficiency of our method and the different pruning options, we generated datasets of various sizes, while keeping the original distribution intact (all artificial datasets will be made available online). Figure 3 shows the influence of the different pruning strategies on the computation time. In general, a combination of pruning and local maxima offers the best performance, over an order of magnitude faster than either method alone. For comparison, the brute force approach at $n = 200$ takes 4,548 seconds, over two orders of magnitude slower (157.9 times) than the fastest result.

[1] datamining.liacs.nl/cortana.html

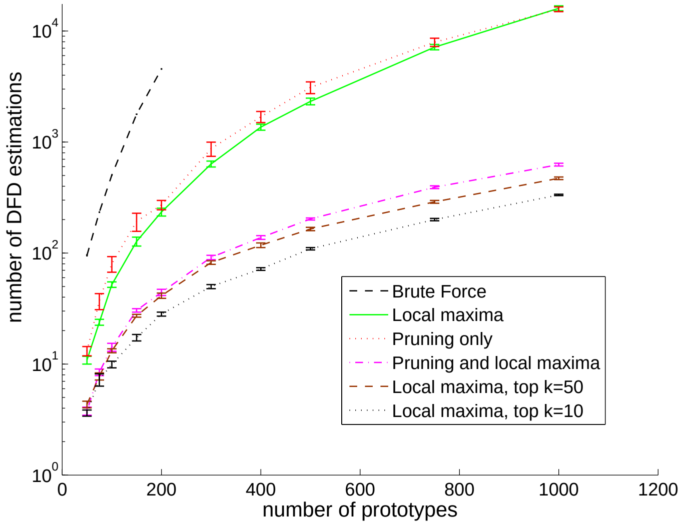

Fig. 3. The increase of computation time for different numbers of prototypes for the artificial dataset, using different pruning strategies. Because of the very long run time, the brute force approach is only performed up to 200 prototypes. The results are averaged over 10 runs.

We also observed the effect the local maxima pruning strategy has on the Subgroup Discovery results. If the local maxima strategy is not used, this will lead to the presentation of many redundant subgroups from the big cluster in the top. Each point in this cluster is then presented as an interesting prototype.

Fraud Detection Our health care application concerns fraud amongst dentists. Each patient is represented by a binary vector of treatments received during a year. The dataset contains 980,350 patients and 542 treatment codes. We use Hamming distance, and quality measure WRAcc. Note that because of the discrete nature of the data, there are many duplicate examples (many patients with identical treatments). Additionally, the distance of a point to different neighbors may be identical, which limits the number of values of σ and ρ that need to be tested. We restrict ρ to a maximum of 10% of the data.

We select a dentist with a markedly high claiming profile, and define the target accordingly. For 5,567 patients (0.57% of the population) this target is true.

Table 1 shows the local subgroups found by our proposed method, LSD. These results were obtained after mining the top 50 subgroups first, and then selecting for diversity using the exclusive coverage heuristic. The interpretation for subgroup S1 is that for patients receiving a *regular consult* (C11, C12) and *dental cleaning* (M50), the dentist often charges extra costs for a *plaque score* (M31) and an *orthopantomogram x-ray picture* (X21). Also an *anamnese* (in-

Table 1. Prototypes and their support in the subgroup, and their support in the reference group excluding the subgroup. The codes indicate treatments that were charged for a patient, the supports indicate the fraction of patients receiving those treatments respectively. The columns # t and # $\neg t$ are the counts within those groups of positive and negative examples.

Subgroup	**Prototype and Supports**	# t	# $\neg t$	**WRAcc**	**p-value**
$S1$ prototype	{C11,C12,C22,M31,M50,X21}				
$S1$	{1.00,0.97,0.17,0.49,0.93,0.60}	54	78		
$R1 \setminus S1$	{1.00,0.94,0.03,0.12,0.95,0.13}	667	10,734	0.0042	< 0.0001
$S2$ prototype	{C11,M31}				
$S2$	{1.00,1.00}	30	2,189		
$R2 \setminus S2$	{1.00,0.11}	94	35,566	0.0006	< 0.0001
$S3$ prototype	{C13,X10,X21}				
$S3$	{0.38,0.71,0.18}	85	12,177		
$R3 \setminus S3$	{0.03,0.11,0.01}	55	30,417	0.0010	< 0.0001

vestigating the patients history, C22) is charged much more often for this group of patients. In subgroup S2, patients receiving a regular consult and a plaque score are occurring relatively more frequently than patients having only a regular consult without the plaque score (which are in the reference group outside the subgroup). In subgroup S3, codes X10 and X21 are *x-ray pictures*, and C13 means an *incidental consult.* Note that treatment C11 is not in the prototype, but still has a support of 77% in the subgroup and a support of 98% in the reference group outside the subgroup. We can conclude that this dentist charges relatively many x-ray pictures of type X10 with a regular or incidental consult. The qualities for the subgroups S1, S2 and S3 are 32, 15 and 14 standard deviations from the mean of the DFD, respectively, which results in p-values near zero.

As a baseline, we compare the results using traditional SD in Cortana, using WRAcc, a search depth of 3 and beam search with beam width 100. We obtain the top 50 subgroups first, and then select for diversity using the exclusive coverage heuristic [5]. There are two subgroups that cover the other subgroups in the top 50: X21 = 1, and C11 =1 $\wedge$ V21 = 1. Code X21 represents an *orthopantomogram* (x-ray photo), code C11 represents a *consult*, and code V21 is used for *polishing a sealing.* The subgroup sizes are 100,643 and 369,748, with 2,438 and 3,228 positive examples, respectively, which leads to a WRAcc of 0.0020, and 0.0014 respectively. The main difference between LSD and traditional SD is that the local approach presents locally deviating patient groups and provides information about the patient group's neighborhood. The resulting subgroups are easier to evaluate by domain experts, and detailed enough to be investigated further by fraud investigators. The traditional SD approach returns global patterns that are not interesting or specific enough to trigger an action.

With our LSD algorithm, we were able to mine interesting subgroups in 5.5 hours in this dataset containing almost a million examples and over 500 attributes. The Cortana traditional SD algorithm took 24 minutes. Both were run on the same machine with 32 GB of main memory, and 8 cores. Although the runtime of the LSD approach depends on the dataset, and the parameter for the maximum value of ρ, this shows that the LSD approach is scalable to fairly big datasets.

We applied our method to compare pharmacies as well as different type of dentists, also using other distance measures like standardized Euclidean distance (in this case, the emphasis is on costs per treatment rather than combinations of treatments only). The results were presented to the fraud investigation department of an insurance company, and were considered very interesting for further investigation. The absence of 'cheap' patients in the reference group as well as the presence of relatively many similar, but more expensive, patients in the subgroup is very useful for indicating inefficient claim behavior.

7 Conclusion and Further Research

In this paper, we present a new approach to find local subgroups in a database. These local subgroups are very relevant within a fraud detection application because systematically committed fraud leads to local distribution changes. Inspired by the fraud detection application, there are numerous directions to investigate further. One promising direction will be cost-based Subgroup Discovery to find even more interesting subgroups. Instead of the distance-based approach, we can also investigate a more traditional, descriptive approach to find local subgroups.

References

1. S. Bay and M. Pazzani. Detecting group differences: Mining contrast sets. *Data Mining and Knowledge Discovery*, 5(3):213–246, 2001.
2. G. Dong and J. Li. Efficient mining of emerging patterns: discovering trends and differences. In *Proceedings of KDD '99*, pages 43–52, New York, NY, USA, 1999.
3. W. Duivesteijn and A. Knobbe. Exploiting false discoveries - statistical validation of patterns and quality measures in subgroup discovery. In *proceedings ICDM*, 2011.
4. W. Klösgen. *Handbook of Data Mining and Knowledge Discovery, chapter 16.3: Subgroup Discovery.* Oxford University Press, New York, 2002.
5. A. Knobbe and E. Ho. Pattern teams. In *Proceedings ECML PKDD 2006*, volume LNCS 4213, pages 577–584. Springer, 2006.
6. M. Kulldorff. A spatial scan statistic. *Communications in Statistics - Theory and Methods*, 26(6):1481–1496, 1997.
7. B. T. Luong, S. Ruggieri, and F. Turini. k-nn as an implementation of situation testing for discrimination discovery and prevention. In *Proceedings of KDD '11*, pages 502–510, New York, NY, USA, 2011.
8. S. Wrobel. An algorithm for multi-relational discovery of subgroups. *Proceedings of PKDD*, pages 78–87, 1997.