Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16)

A Framework for Outlier Description Using Constraint Programming

Chia-Tung Kuo
Department of Computer Science
University of California, Davis
tomkuo@ucdavis.edu

Ian Davidson
Department of Computer Science
University of California, Davis
davidson@cs.ucdavis.edu

Abstract

Outlier *detection* has been studied extensively and employed in diverse applications in the past decades. In this paper we formulate a related yet understudied problem which we call *outlier description*. This problem often arises in practice when we have a small number of data instances that had been identified to be outliers and we wish to explain why they are outliers. We propose a framework based on constraint programming to find an optimal subset of features that most differentiates the outliers and normal instances. We further demonstrate the framework offers great flexibility in incorporating diverse scenarios arising in practice such as multiple explanations and human in the loop extensions. We empirically evaluate our proposed framework on real datasets, including medical imaging and text corpus, and demonstrate how the results are useful and interpretable in these domains.

Introduction

Outlier *detection* is a core technique used extensively in AI along with clustering and classification. The core outlier detection problem aims to identify which subset of data instances are most dissimilar to the vast majority. For example, in a dataset of individuals containing their physical measurements an outlier may be an extremely tall or short person. The recent survey on the topic of anomaly detection (Chandola, Banerjee, and Kumar 2009) outlines variations of the core problem such as contextual outlier detection which identifies a context where the point is an outlier. In our earlier example, a 6-foot tall person is not an outlier in general but a 6-foot tall eight-year-old certainly is. Outlier detection has numerous practical applications, such as fraud detection, network intrusion detection and climate anomaly detection (Chandola, Banerjee, and Kumar 2009).

However, in many real world problems a more typical setting is that the outliers are *already known* and discovering what makes them outliers will have an immense impact. Consider the example of automobile recalls. An automobile recall occurs when a small, yet sufficient, number of cars are found to have defective steering. Here we are given examples of the outliers (recalled cars) and many examples of the non-outliers (non-recalled cars), with the aim to find a description of what makes the identified outliers anomalous. The recall can then be limited to those automobiles which match the description instead of all automobiles.

We call such setting the outlier description problem and address it with the definition below.

Problem 1 ***The General Outlier Description Problem.*** *Given a collection of instances that are deemed normal, N, and another separate collection deemed outliers, O, where instances in both N and O are in feature space S, find a feature mapping $t : S \rightarrow F$ that maximizes the difference between O and N according to some outlier property measure $m : F \rightarrow \mathbb{R}$ and an aggregate objective $\texttt{obj} : (\mathbb{R}^{|N|}, \mathbb{R}^{|O|}) \rightarrow \mathbb{R}$. That is,* $\max_{t,m} \texttt{obj}(m(t(N)), m(t(O)))$.

The functions t and m can be naturally viewed as the description/explanation of the outlying behavior where t describes the space where the behavior is exhibited and m defines what we deem to be an outlier in this space. In this paper we focus on a particular form of t which projects the data to a subspace and use an m that measures the local neighborhood density around each point. The objective $\texttt{obj}$ here is the difference between the densities surrounding the outliers and normal instances. Notice that this problem is distinguished from metric learning in that t takes single instances as inputs, instead of pairs, and we do not enforce all normal instances to be close together.

A key requirement of outlier description is flexibility since each domain may have its domain knowledge with which the description should be consistent. To address this we propose using constraint programming (CP) which is a declarative paradigm of solving optimization problems, not unlike mathematical programming, but offers a more expressive language. A strength of the CP framework is that it allows the functions t, m and $\texttt{obj}$ to take a wide variety of forms unincumbered by the limitations associated with mathematical programming. CP has only recently been used in data mining, in particular association rule mining (De Raedt, Guns, and Nijssen 2008; 2010) but has not yet been fully exploited in the community. Our current work is another example of using this powerful mature paradigm which interfaces with highly parallelized state-of-the-art solvers.

Our contributions are as follows:

- We introduce and define the outlier description problem.
- We propose a framework for outlier description using CP languages and explore diverse variations of our framework, each of which addresses practical scenarios.
- We provide a complexity analysis of our CP framework in terms of the number of variables and constraints required.
- We experimentally demonstrate the utility of our work on real datasets of medical images and text documents.

Related Work

Supervised Outlier Detection. Outlier detection has been extensively studied for decades and applied to diverse applications. Recent survey articles (Chandola, Banerjee, and Kumar 2009; Hodge and Austin 2004) and book chapters (Han, Kamber, and Pei 2011; Aggarwal 2015) have covered and categorized various methods based on their assumptions, principal techniques and application domains. Supervised outlier detection is similar to the classic task of making binary prediction with the additional issues of highly imbalanced class distribution and its success largely depends on the assumptions and limitations of the chosen predictive models. In general models whose target function is complicated (e.g. neural networks) or that learn from more complex projected feature space (e.g. kernel SVM) lack natural interpretation for our purpose. Our current work, on the other hand is not focused on **prediction** of future outliers but rather attempts to address the problem of **describing** how the given outliers are different than normal instances.

Outlying Properties/Explanations/Subspaces Mining. We summarize this line of work in Table 1 and discuss how our work differs. First, most of these methods focus on a single query point and it is not straightforward to extend to a set of outliers at once. Second, some work is only applicable to either finding contextual outlying property as a single feature (Angiulli, Fassetti, and Palopoli 2009) or only applicable when features are numerical/categorical (Duan et al. 2015; Angiulli, Fassetti, and Palopoli 2009). In addition all the work, to our knowledge, still requires the users to input hyperparameters (or implicitly set some "rules of thumb") such as the threshold of outlying measure, size of the subspace, or the bandwidth of a density kernel, etc. Our work aims to propose a constraint programming framework that is flexible in handling diverse situations (e.g. numerical/categorical features, contextual outliers, human in the loop, etc) and allows the above-mentioned hyperparameters to be learnt in the model.

Constraint Programming in Data Mining. Constraint programming (CP) is a declarative paradigm to solve combinatorial satisfaction/optimization problems. These combinatorial problems are in general NP-Hard (Papadimitriou and Steiglitz 1998) but efficient solvers had been developed to tackle a wide range of problems arising in applications such as scheduling (Bartak 1999). Recently CP had been studied in the data mining community for frequent itemset mining (De Raedt, Guns, and Nijssen 2008; 2010). Some other work (Angiulli, Greco, and Palopoli 2007; Angiulli, Ben-Eliyahu-Zohary, and Palopoli 2008) has tried to formally define outlier detection in knowledge-based systems using the logic programming paradigm but this line of work addresses a different context and objective than our current work.

A Framework using CP Formulation

Here we define one outlier description problem more precisely, introduce our CP framework for the problem and its variants. Following those, we describe how to encode our framework in a modern CP software platform. All our experiments are conducted in the CP language `Numberjack` (Hebrard, O'Mahony, and O'Sullivan 2010) but this is a personal preference and other popular languages such as `MiniZinc` (Nethercote et al. 2007) could have been used.

As mentioned in Problem 1 the essence of outlier description is to search for functions, t and m, that describe what makes the outliers different compared to the inliers. In this paper we restrict the feature mapping to selecting a subspace of the feature set both for its simplicity and natural interpretation. Further we use the local density criterion for outliers based upon the assumption that a normal instance should have many other instances in proximity whereas an outlier has much fewer neighbors. This criterion is a common characterization of outliers and has been utilized by many existing outlier detection methods (Chandola, Banerjee, and Kumar 2009; Keller, Muller, and Bohm 2012; Knorr, Ng, and Tucakov 2000). Selecting feature subspace where the outlying behavior is most exhibited is also well argued. Not only is a lower dimensional subspace easier to understand by humans, but the curse of dimensionality renders any distance measures meaningless in very high dimensions (Parsons, Haque, and Liu 2004). A natural objective in this context is to maximize the difference of numbers of neighbors between normal points and outliers. A large gap would truly substantiate the assumption of local density between outliers and normal points. With these choices made we can formally state the definition of outlier description problem studied in this paper.

Problem 2 ***The Subspace Outlier Description Problem.***
Given a set of normal instances N and a set of outliers O in a feature space S, find the tuple (F, k_N, k_O, r) where $k_N - k_O$ is maximized, $F \subset S$ and $\forall x \in N$, $|\mathcal{N}_F(x, r)| \geq k_N$, and $\forall y \in O$, $|\mathcal{N}_F(y, r)| < k_O$. $\mathcal{N}_F(x, r)$ is the set of instances within radius r of x in subspace F.

This definition embodies the belief that the normal points are locally denser than the outliers and the core of the problem is to find the feature subspace where this occurs. In terms of the languages used in Problem 1, t is characterized by F and simply zeros out some components in the original feature space S; $m(x) = |\mathcal{N}(x, r)|$; and $\texttt{obj}(A, B) = \min A - \max B$. Our CP formulation aims to encode this precise problem (and some of its variants) using properly defined variables and constraints.

Optimization Models

In this section we propose three CP optimization models that address our outlier description problem. The first formulation is a direct translation of Problem 2 where the vari-

(Duan et al. 2015)	Given a query outlier, look for the minimal subspace where the point ranks highest in outlyingness as measured by estimated density using Guassian kernel (and bandwith set according to (Härdle 1991)).
(Angiulli, Fassetti, and Palopoli 2009)	Given a query outlier q, find disjoint subspaces S and E such that q is outlying in S with respect to all points sharing the same values with q on E (i.e. **contextual** explanation). Outlyinyness is measured by a notion relative frequency. Need user-specified thresholds θ (for outlyingness) and σ (for size of E). Only for categorical variables.
(Zhang and Wang 2006)	Define outlying degree (OD) of a point as sum of distances to its k nearest neighbors; look for all subspaces in which (given) outliers have OD $>$ some threshold T; employ genetic algorithms to search for subspaces.
(Knorr and Ng 1999)	Find outliers first based on neighborhood coverage (similar to ours, but with user-specified radius d and coverage proportion p); define notions of strong and weak outliers and search for smallest subspaces in which the detected outliers are outlying.

Table 1: Some existing related work that looks for explanations (e.g. properties, subspaces) where a query point exhibits the most outlying characteristics.

ables (F, k_N, k_O, r) are defined and properly constrained. The other two formulations show the flexibility offered by a CP framework. The second formulation allows an outlier to be in either one of multiple subspaces, whilst the third formulations allows placing a human in the loop to answer queries. We motivate each of these formulations with simple but practical examples and thus demonstrate the flexibility of the CP framework.

Formulation #1: Learning A Single Outlier Description. The CP model is formulated as in equation (1). The subspace F in which the outlying behavior is exhibited is defined as a binary vector variable; the bits that are set in the solution correspond to the subspace. One major strength of CP formulation is that the users only need to supply the bounds of hyperparameters $(k_{max}, k_{min}, r_{max})$ to be searched over. In other work of outlier detection/description these hyperparameters/criteria are commonly chosen beforehand according to domain knowledge in applications (Knorr, Ng, and Tucakov 2000) or certain rule of thumb (Duan et al. 2015).

An alternative choice of objective can be minimizing $\sum_i f_i$ since a small subspace is often more interpretable in applications (Knorr and Ng 1999; Duan et al. 2015). It is worth noting that CP models can also be solved without an objective, but instead requiring the density gap to be at least some constant; in such cases, the solver searches for all feasible solutions and it is up to the user to decide which is the most suited in his/her context.

$$\begin{aligned}
&\text{Objective} && \text{Maximize } k_N - k_O \\
&\text{Variables} && F = [f_1, f_2, \ldots, f_{|S|}] \in \{0,1\}^{|S|} \\
& && k_{min} \le k_O \le k_N \le k_{max} \\
& && 0 \le r \le r_{max} \\
&\text{Constraints} && (C1)\ \forall x \in N,\ |\mathcal{N}_F(x, r)| \ge k_N \\
& && (C2)\ \forall y \in O,\ |\mathcal{N}_F(y, r)| < k_O
\end{aligned} \tag{1}$$

Formulation #2: Outliers in Multiple Subspaces Here we consider the case where outliers can reside in different (outlying) subspaces. This fits the setting where there could be multiple reasons/explanations why a point is an outlier. Recall the automobile manufacturing example from the introduction section. This formulation allows describing outliers due to multiple parts or their combinations. For clarity we explain the formulation with two subspaces. In this case we have two sets of feature subspace selectors $F = \{f_1, \ldots, f_{|S|}\}$ and $G = \{g_1, \ldots, g_{|S|}\}$ as variables. Normal instances must satisfy the dense neighborhood condition (C1) in **both** subspaces, whereas an outlier is outlying in **either** subspace defined by F or G. Note this is **not** the same as using a single feature selector being the disjunction of F and G; a single feature selector $H = F \vee G$ would require the outliers to be outlying in a higher dimensional subspace that includes dimensions from **both** F and G.

The optimization model is formulated similarly to (1) except we add subspace selector $G = [g_1, g_2, \ldots, g_{|S|}] \in \{0,1\}^{|S|}$ and include two neighborhood radii r_F, r_G to the variable set and substitute constraints $(C1)$ and $(C2)$ with

$$\begin{aligned}
&(C3)\ \forall x \in N,\ |\mathcal{N}_F(x, r_F)| \ge k_N \ \texttt{AND}\ |\mathcal{N}_G(x, r_G)| \ge k_N \\
&(C4)\ \forall y \in O,\ |\mathcal{N}_F(y, r_F)| < k_O \ \texttt{OR}\ |\mathcal{N}_G(y, r_G)| < k_O
\end{aligned} \tag{2}$$

Formulation #3: Human in the Loop In practice, often the known outliers are hand labeled (e.g. defects in car recall, frauds in transactions, etc) and thus the labels are considered more accurate. The set of normal points, on the other hand, can also potentially contain outliers not yet reported/found. Therefore we formulate the constraints that allow some points in the normal set to violate the normal density conditions. These contentious points can then be referred to a human expert for further examination. This can be achieved by defining an additional set of binary variables $W = \{w_1, \ldots, w_n\}$ each being an indicator for whether a normal point x_i obeys the neighborhood constraint. An additional pre-specified upper bound, w_{max}, is necessary to rule out trivial solutions.

The model can be obtained from equation (1) by adding violation points indicator $W = [w_1, w_2, \ldots, w_n], w_i \in \{0,1\}$ to the variable set and replacing constraint $(C1)$ with

$$\begin{aligned}
&(C5)\ \forall x_i \in N,\ |\mathcal{N}_F(x_i, r)| \ge (1 - w_i) k_N \\
&(C5.5)\ \sum_{i=1}^{n} w_i \le w_{max}
\end{aligned} \tag{3}$$

Encoding Constraints in CP

Modern CP software platforms typically offers a range of common but simple constructs (constraints, functions/operations on variables, etc). More complicated constraints are typically achieved by defining additional *auxiliary variables*. Auxiliary variables are often artifacts of

the problem and allow for an easy to understand and implement formulation. Auxiliary variables can also be constrained by basic constructs such as summation, subtraction and $<$ and $>$, etc. Here we describe how to encode the neighborhood density constraints $(C1)$ and $(C2)$ with auxiliary variables; other variations of constraints can be encoded similarly (Note that logical `AND` and `OR` are simple constructs in most CP software). We first define binary auxiliary variables $\{z_{ij}\}$ which is set to 1 if instance j is within the r-neighborhood of instance i, or 0 otherwise. This is achieved by requiring $z_{ij} = (d_F(x_i, x_j) \leq r)$ where d_F is the distance function in the subspace F and the right hand side evaluates to either 1 (`true`) or 0 (`false`). The constraints $(C1)$ and $(C2)$ can then be written as

`/* Define the auxiliary variable` z_{ij}

$$\forall x_i, x_j \in N \cup O, \quad z_{ij} = (d_F(x_i, x_j) \leq r)$$

`/* Constrain` z_{ij}

$$\forall x_i \in N, \sum_j z_{ij} \geq k_n \qquad \forall x_i \in O, \sum_j z_{ij} < k_O$$

Other constraints involving the numbers of neighbors can be encoded similarly. It is however worth noting that in the definitions above, we also make the assumption that the distance d_F can be written as a simple function of F (over individual features separately). This is often no trouble for most commonly used distances such as ℓ^p norm, Hamming distance, etc. For example, if the Euclidean distance (ℓ^2 norm) is used, then $d_F(x_i, x_j) = \sum_{k=1}^{d} f_k(x_i^{(k)} - x_j^{(k)})^2$; in this case we can pre-compute a table of entry-wise squared differences $||x_i - x_j||_2^2$ between each pair of instances beforehand, and during the optimization d_F amounts to simply entry-wise multiplying F against pre-computed constants.

Insights into Formulations

Here we discuss some insights into our work in particular in relation to existing work. In this work we study a mapping that projects the instances to a subspace where the set O are considered outliers according to a neighborhood density criterion. Similar criteria in outlier detection (Knorr, Ng, and Tucakov 2000) would identify exactly the same outliers in the projected (reduced) subspace.

We can also view our subspace description in terms of the connectivity among normal instances. One can construct a nearest neighbor graph from the output from m, where nodes are data instances and edges are present between neighbors within radius r in the projected (reduced) feature subspace F. By definition, the normal nodes will each have at least k_N links whereas outlier nodes at most $k_O - 1$ (since it's $< k_O$). If the gap $k_N - k_O$ is large enough, a "random surfer" (Page et al. 1998) on this graph would end up in a normal node with much higher probability than an outlier node. This could potentially offer more insights when coupled with the Human-in-the-loop formulation (#3). In practice even the normal instances could have a range of the numbers of neighbors (or connectivity, or ranks). Examining the ranks of nodes and the changes of the gap $k_N - k_O$ in formulation #3 could lead to more accurate identification of fuzzier outliers not yet identified.

Complexity of Models

Here we provide a discussion of the complexity of our proposed framework in terms of the number of variables and basic constraints as functions of the numbers of data instances and their dimension. We analyze these numbers using formulation #1 though similar analysis is applicable to all other variant models. First the variables explicitly defined are F, k_N, k_O and r. To allow the complexity to be quantified we require that r take on a discrete set. One straightforward approach is to simply specify a step size, s, such that $r \in \{0, s, 2s, \ldots, r_{max}\}$. This discretization does not apply to k_N and k_O as the numbers of neighbors are by nature integers. As explained above, the actual encoding of the neighborhood constraints requires additional auxiliary variables and constraints: one z_{ij} for each pair of data instances and one constraint to set its value. Once we have these in place, enforcing the number of instances within a neighborhood is a single constraint for each instance. Overall the complexity for formulation #1 is summarized in Figure 1. Though the combinatorial nature of the problem makes it seem complex, it is important to note that our formulations not only allow great flexibility but also perform an exhaustive optimization and return a global optimum.

Variables	Size	Domain
F	p	$\{0, 1\}$
k_N, k_O	2	$\{k_{min}, \ldots, k_{max}\}$
r	1	$\{0, s, 2s, \ldots, r_{max}\}$
z_{ij}	$\binom{n}{2}$	$\{0, 1\}$

(a) Variables

Constraints	Size
$k_O \leq k_N$	1
$z_{ij} = (d_F(x_i, x_j) \leq r)$	$\binom{n}{2}$
$\sum_i z_{ij} \geq k_N$ (or $< k_O$)	n

(b) Constraints

Figure 1: Complexity summary in terms of the numbers of variables (and their domains) and basic constraints. $n = |N \cup O|$ is the total number of data instances and $p = |S|$ is the dimension of the feature space.

Empirical Evaluation

In this section we aim to empirically evaluate our proposed framework and demonstrate its applicability and usefulness with experiments on real datasets. We describe tasks and report results on two datasets, medical imaging data and text corpus. In each experiment we designate a small number of instances from a known group as "outliers" and aim to find the most suitable descriptions in terms of the neighborhood density in some feature subspace. We start each experiment with a description on the setup and preprocessing (if any).

Experiment 1

Dataset In this experiment we study a functional magnetic resonance imaging (fMRI) dataset. This dataset consists of resting state scans from 61 elderly subjects, of which 21 subjects were diagnosed as demented, another 21 were mildly cognitively impaired (MCI) and the remaining 19 subjects were healthy. Each scan consists of a sequence of 3D snapshots of the voxels' blood oxygenation levels over 100+ time stamps. For the ease of experiment and presentation, we only pick a middle slice of the brain images to work on such that our scans are 2D images over time.

For each scan we construct the feature set as follows. We use a mask of the know anatomical regions (AR) in the brain, provided by Neurology professionals (see Figure 2(a)), to divide the brain into 27 disjoint parts. A feature is constructed from each pair of distinct ARs: we compute the Pearson's correlation coefficients between the time series of all pairs of voxels across these two ARs and record the average of these correlations as the value for this feature. Such pairwise correlation is a natural measure for measuring degree of co-activation between time sequences and is common in brain connectivity studies in neuroscience commmunity (Friston 2011). Eventually each scan is characterized by $\binom{27}{2} = 351$ features where each takes values in the interval $[-1, 1]$. In the experiments we use the ℓ_1 norm as the distance, i.e. the sum of entry-wise absolute differences.

Tasks and Results **Demented Subjects as Outliers.** One question of interest to the neuroscience community is the identification of parts of the brain whose deformation is responsible for people to develop dementia. In this experiment we use the 19 scans from the healthy subjects as normal points and randomly choose 3 scans from the demented subjects as outliers and search for an outlier explanation using our formulation #1 with $k_{min} = 1, k_{max} = 10$ and discretized domain for $r \in \{0, 0.5, 1, \ldots, 10\}$. Further we add a bound constraint on the size of the learnt subspace with $\sum_i f_i \leq 30$ both to speed up the optimization and also because lower dimensional subspaces are often easier to interpret. The results for the neighborhood density criteria are $k_N = 4, k_O = 2$ and $r = 2.5$ and the subspace F has 17 entries set to 1. Our results suggest that each healthy scan has at least 3 other scans sharing similar (within 2.5) correlations on these 17 pairings of ARs whereas each demented scan has no other scans sharing. We examine the learnt feature subspace (i.e. entries set to 1 in F) and show the top 3 regions involved in most pairs being selected in F and also the pairings with the top 1 region in Figure 2. These results offer insightful and interpretable descriptions that allow neurology professionals to further inspect the data in this regard.

Experiment 2

Dataset In this experiment we apply our methods to the text corpus, 20 Newsgroups Dataset.[1] This popular text dataset contains $\sim$ 20000 documents collected from 20 newsgroups. Each document was recorded as bag-of-word features from a dictionary of 61188 words.

Task and Results The task at hand is to see if we can successfully describe outliers where we can select all documents from one major newsgroup as the normal instances and a smaller number of documents from another newsgroup as outliers. The goal here would be to identify the subset of words that can separate those outliers from the majority documents in terms of their numbers of neighbors.

Since the dictionary is large in size and most documents only have a handful of words present at all, for the ease of experiments we preprocess our data by using only the top 50 non-stop-words from each of the selected newsgroups as features. We sample 50 documents from newsgroup *comp.sys.ibm.pc.hardware* (denoted by **Hardware**) as normal instances and we sample 3 documents from each of the two newsgroups *rec.sport.baseball* (**Baseball**) and *talk.religion.misc* (**Religion**) as outliers. Table 2 lists some most frequently occurring words from these 3 newsgroups, respectively. It is worth noting that in this experiment we use ℓ_1 norm as our distance as opposed to the more standard cosine distance in text mining. Since we have a very small (but meaningful) vocabulary set as features, many documents will not share any non-zero features at all. This renders the inner product (i.e. cosine distance) a poor choice since most pairs of instances (both outliers and normal) would have a distance of 0 or a extremely small measure, making the results difficult for any insight.

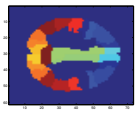
Hardware	drive, scsi, can, edu, com, one, will, card, mb, idle, system, ...
Baseball	edu, write, year, can, article, game, don, one, will, team, ...
Religion	god, one, will, edu, can, writes, people, jesus, com, article, ...

Table 2: Most frequently occurring words from each of the 3 newsgroups used in our experiments. We can see words that are characteristic of the particular topics in the newsgroups, such as *drive, scsi, team, baseball, god, jesus*, etc. But another set of words common in all 3 newsgroups are ones like *edu, com, will, can*, etc.

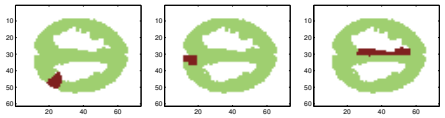
Outliers Described by a Single Subspace. In this first experiment we apply formulation #1 to maximize the difference between the numbers of neighbors between normal instances and outliers. The parameter bounds are set as follows. $k_{max} = 20$ and $r \in \{0.01, 0.02, \ldots, 2.0\}$ and we enforce a constraint on the size of the subspace, $\sum_i f_i \leq 10$. The optimal solution is shown in Figure 3(a). This suggests that for this small dataset in the 10-dimensional feature subspace corresponding to the 10 words, all 6 outliers (**Baseball** and **Religion**)) can be isolated without any neighbor (i.e. $< k_O = 2$) with radius $r = 0.15$ whereas all normal instances (**Hardware**) have at least 20 instances in the same neighborhood. We can also see that the words identified by F are very characteristic of the differences between newsgroups and provide an interpretable description of the outlying behavior.

Outliers Described by Multiple Subspaces. Here we want to utilize our multiple subspace formulation to see if we can identify 2 distinct feature subspaces each character-

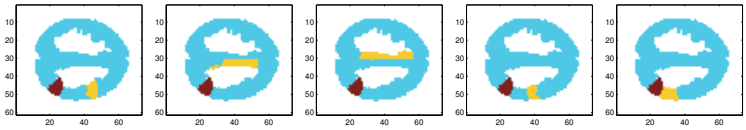
[1] The version we used is from http://qwone.com/~jason/20Newsgroups/ where cross-posts and some headers were removed.



(a) Color-coded known anatomical regions in the brain.



(b) The top 3 anatomical regions appearing the most times in the pairs selected in F; Number of occurrences are 5 (left), 3 (middle) and 3 (right).



(c) The pairs of anatomical regions selected in F that involve the most frequent occurring region (left in 2(b)).

Figure 2: A closer look of the feature subspace F in the experiments of describing the demented outliers. Each feature in the original feature space corresponds to a pair of known anatomical regions in Figure 2(a).

izing one group of outliers. Note that formulation #2 is a strict generalization of formulation #1. This implies if all parameters are set identically in both models, then formulation #2 would simply output the same one subspace as found by formulation #1 and the other subspace would be trivial (say, $r_G = 0$ hence every pairs are neighbors). To avoid this, we explicitly set the objective to minimize $\sum_i f_i + \sum_i g_i$ instead of maximizing $k_N - k_O$, effectively looking for the most compact subspaces in which the outlying behavior is exhibited. To ensure the local density criteria still in tact, we add an additional bound on $k_N - k_O \geq 5$. The choices of k_{max} and r_F, r_G are the same as above. The results are shown in Figure 3(b). Again the two subspaces F and G conform to our "truth" of selecting outliers from two sources, **Baseball** and **Religion**, respectively. The minimality of subspace dimensions make them practical to be further examined by human experts.

$\sum_i f_i$	Words identified by F	k_N	k_O	r
10	last, season, like, bible, team, baseball, god, play, say, true	20	2	0.15

(a) Optimal solution to the single subspace model.

$\sum_i f_i$	Words identified by F	$\sum_i g_i$	Words identified by G
1	season	2	bible, man

k_N	k_O	r_F	r_G
20	15	0.03	0.15

(b) Optimal solution to the multiple subspaces model.

Figure 3: Results for the outlier descriptions on 20 Newsgroups datasets using both a single subspace model and a two subspaces model.

Each of our experiments (fMRI and text documents) took about 5 minutes to run on a 12-core workstation. Note that the additional bounds such as $\sum_i f_i \leq 10$ play an important role in speeding up the computation as they could greatly reduce the domains of the variables and thus the search space. Solving these formulations on large data sets could potentially be time-consuming; we offer some directions to this issue in the conclusion.

Conclusion

In this paper we formally define the outlier description problem, which aims to find explanations about how a given set of outliers deviate from the normal instances. We focus on one particular definition based on neighborhood density criterion and subspace selection. We propose a constraint programming based framework to encode the problem. Our framework offers great flexibility: the user can solve for an optimal objective or only look for a feasible solution; learn all parameters at once or only a subset of them while supplying others; and encode additional constraints and other problem variations arising in practical applications. We demonstrate the usefulness of our proposed framework by experiments on real datasets of medical imaging and text corpus.

One major limitation of the framework is the scalability issue stemming from combinatorial optimization in general and the numbers of auxiliary variables needed in encoding the problem. This is partially alleviated since many CP languages interface with state-of-the-art integer program solvers such as Gurobi (Gurobi Optimization 2015) and CPLEX[2] which efficiently utilize multi-core architectures. Machines with more cores such as Amazon's AWS could further exploit the parallelization. Another way to scale up is to implement more efficient propagators in a more specialized CP environment such as Gecode (Gecode Team 2006).

Acknowledgment

The authors gratefully acknowledge support of this research via ONR grant N00014-11-1-0108 and NSF Grant NSF IIS-1422218.

[2]www.ibm.com/software/commerce/optimization/cplex-optimizer/

References

Aggarwal, C. C. 2015. *Data Mining: The Textbook*. Springer.

Angiulli, F.; Ben-Eliyahu-Zohary, R.; and Palopoli, L. 2008. Outlier detection using default reasoning. *Artificial Intelligence* 172(16):1837–1872.

Angiulli, F.; Fassetti, F.; and Palopoli, L. 2009. Detecting outlying properties of exceptional objects. *ACM Trans. Database Syst.* 34(1):7:1–7:62.

Angiulli, F.; Greco, G.; and Palopoli, L. 2007. Outlier detection by logic programming. *ACM Trans. Comput. Logic* 9(1).

Bartak, R. 1999. Constraint programming: In pursuit of the holy grail. In *in Proceedings of WDS99 (invited lecture)*, 555–564.

Chandola, V.; Banerjee, A.; and Kumar, V. 2009. Anomaly detection: A survey. *ACM Computing Surveys (CSUR)* 41(3):15.

De Raedt, L.; Guns, T.; and Nijssen, S. 2008. Constraint programming for itemset mining. In *ACM SIGKDD*, 204–212. ACM.

De Raedt, L.; Guns, T.; and Nijssen, S. 2010. Constraint programming for data mining and machine learning. In *AAAI 10*, 1671–1675.

Duan, L.; Tang, G.; Pei, J.; Bailey, J.; Campbell, A.; and Tang, C. 2015. Mining outlying aspects on numeric data. *Data Mining and Knowledge Discovery* 1–36.

Friston, K. J. 2011. Functional and effective connectivity: a review. *Brain connectivity* 1(1):13–36.

Gecode Team. 2006. Gecode: Generic constraint development environment. Available from `http://www.gecode.org`.

Gurobi Optimization, I. 2015. Gurobi optimizer reference manual.

Han, J.; Kamber, M.; and Pei, J. 2011. *Data Mining: Concepts and Techniques*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 3rd edition.

Härdle, W. 1991. *Smoothing techniques: with implementation in S*. Springer Science & Business Media.

Hebrard, E.; O'Mahony, E.; and O'Sullivan, B. 2010. Constraint Programming and Combinatorial Optimisation in Numberjack. In *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems, 7th International Conference, CPAIOR 2010*, 181–185.

Hodge, V. J., and Austin, J. 2004. A survey of outlier detection methodologies. *Artificial Intelligence Review* 22(2):85–126.

Keller, F.; Muller, E.; and Bohm, K. 2012. Hics: high contrast subspaces for density-based outlier ranking. In *ICDE*, 1037–1048. IEEE.

Knorr, E. M., and Ng, R. T. 1999. Finding intensional knowledge of distance-based outliers. In *VLDB*, volume 99, 211–222.

Knorr, E. M.; Ng, R. T.; and Tucakov, V. 2000. Distance-based outliers: algorithms and applications. *VLDB* 8(3-4):237–253.

Nethercote, N.; Stuckey, P. J.; Becket, R.; Brand, S.; Duck, G. J.; and Tack, G. 2007. Minizinc: Towards a standard cp modelling language. In *Principles and Practice of Constraint Programming–CP 2007*. Springer. 529–543.

Page, L.; Brin, S.; Rajeev, M.; and Terry, W. 1998. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford University.

Papadimitriou, C. H., and Steiglitz, K. 1998. *Combinatorial optimization: algorithms and complexity*. Courier Corporation.

Parsons, L.; Haque, E.; and Liu, H. 2004. Subspace clustering for high dimensional data: a review. *ACM SIGKDD Explorations Newsletter* 6(1):90–105.

Zhang, J., and Wang, H. 2006. Detecting outlying subspaces for high-dimensional data: the new task, algorithms, and performance. *Knowledge and information systems* 10(3):333–355.