REMIX: Automated Exploration for Interactive Outlier Detection

Yanjie Fu
Missouri U. of Science & Technology
Missouri, USA
fuyan@mst.edu

Charu Aggarwal
IBM T. J. Watson Research Center
NY, USA
charu@us.ibm.com

Srinivasan Parthasarathy
IBM T. J. Watson Research Center
NY, USA
spartha@us.ibm.com

Deepak S. Turaga
IBM T. J. Watson Research Center
NY, USA
turaga@us.ibm.com

Hui Xiong
Rutgers University
NJ, USA
hxiong@rutgers.edu

ABSTRACT

Outlier detection is the identification of points in a dataset that do not conform to the norm. Outlier detection is highly sensitive to the choice of the detection algorithm and the feature subspace used by the algorithm. Extracting domain-relevant insights from outliers needs systematic exploration of these choices since diverse outlier sets could lead to complementary insights. This challenge is especially acute in an *interactive setting*, where the choices must be explored in a time-constrained manner.

In this work, we present REMIX, *the first system to address the problem of outlier detection in an interactive setting*. REMIX uses a novel mixed integer programming (MIP) formulation for *automatically selecting and executing a diverse set of outlier detectors within a time limit*. This formulation incorporates multiple aspects such as (i) an *upper limit on the total execution time of detectors* (ii) *diversity in the space of algorithms and features*, and (iii) *meta-learning for evaluating the cost and utility of detectors*. REMIX provides two distinct ways for the analyst to consume its results: (i) a partitioning of the detectors explored by REMIX into *perspectives* through low-rank non-negative matrix factorization; each perspective can be *easily visualized as an intuitive heatmap of experiments versus outliers*, and (ii) an *ensembled set of outliers* which combines outlier scores from all detectors. We demonstrate the benefits of REMIX through extensive empirical validation on real-world data.

1 INTRODUCTION

Outlier detection is the identication of points in a dataset that do not conform to the norm. This is a critical task in data analysis and

is widely used in many applications such as financial fraud detection, Internet traffic monitoring, and cyber security [4]. Outlier detection is highly sensitive to the choice of the detection algorithm and the feature subspace used by the algorithm [4, 5, 35]. Further, outlier detection is often performed on high dimensional data in an unsupervised manner without data labels; distinct sets of outliers discovered through different algorithmic choices could reveal complementary insights about the application domain. Thus, unsupervised outlier detection is a data analysis task which inherently requires a principled exploration of the diverse algorithmic choices that are available.

Recent advances in interactive data exploration [10, 17, 33, 34] show much promise towards automated discovery of advanced statistical insights from complex datasets while minimizing the burden of exploration for the analyst. We study unsupervised outlier detection in such an *interactive setting* and consider three practical design requirements: **(1) Automated exploration:** the system should automatically enumerate, assess, select and execute a diverse set of outlier detectors; the exploration strategy should guarantee coverage in the space of features and algorithm parameters. **(2) Predictable response time:** the system should conduct its exploration within a specified time limit. This implies an exploration strategy that is sensitive to the execution time (cost) of the detectors. **(3) Visual interpretability:** the system should enable the user to easily navigate the results of automated exploration by grouping together detectors that are similar in terms of the data points they identify as outliers.

1.1 Key Contributions

We present REMIX, a *modular* framework for automated outlier exploration and the first to address the outlier detection problem in an interactive setting. To the best of our knowledge, none of the existing outlier detection systems support interactivity in the manner outlined above – in particular, we are not aware of any system which automatically explores the diverse algorithmic choices in outlier detection within a given time limit. The following are the key contributions of our work.
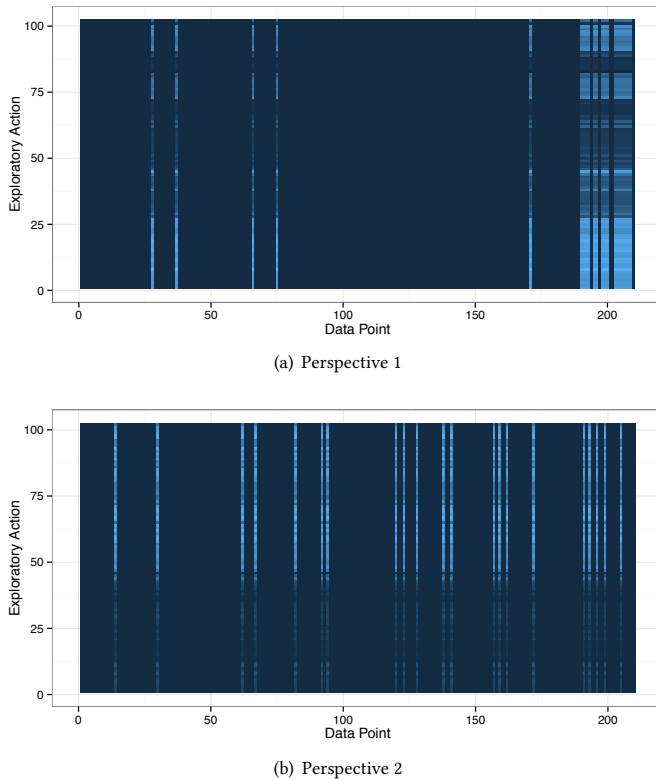
1.1.1 MIP based Automated Exploration. REMIX systematically enumerates candidate outlier detectors and formulates the exploration problem as a mixed integer program (MIP). The solution to this MIP yields a subset of candidates which are executed by REMIX. The MIP maximizes a novel *aggregate utility measure* which trades off between the total utility of the top-k vs all selected candidates;

(a) Perspective 1

(b) Perspective 2

Figure 1: Factorization of an outlier matrix into two perspectives. Each perspective is a heatmapped matrix whose columns are data points and rows are detectors. The intensity of a cell (s, p) in a perspective corresponds to the extent to which detector s identifies point p as an outlier. Each perspective clearly identifies a distinct set of outliers.

the MIP also enforces (i) an upper limit on the total cost (budget) of the selected candidates, and (ii) a set of *diversity constraints* which ensure that each detection algorithm and certain prioritized feature subspaces get at least a certain minimum share of the exploration budget.

1.1.2 Meta-Learning for Cost and Utility. The REMIX MIP requires an estimate of cost and utility for each candidate detector. In order to estimate cost, REMIX trains a meta-learning model for each algorithm which uses the number of data points, size of the feature subspace, and various product terms derived from them as meta-features. It is significantly harder to estimate or even define utility. REMIX handles this by defining the utility of a detector as a proxy for its accuracy on the given data. REMIX estimates this by training a meta-learning model for each algorithm that uses various statistical descriptors of a detector's feature subspace as meta-features; this model is trained on a corpus of *outlier detection datasets that are labeled* by domain experts [2].

1.1.3 Perspective Factorization. The diversity of feature subspaces and algorithms explored by REMIX will result in different

detectors marking a distinct set of data points as outliers. REMIX provides a succinct way for the data analyst to visualize these results as heatmaps. Consider the outlier matrix Δ where $\Delta_{s,p}$ is the normalized ($[0, 1]$-ranging) outlier score assigned by detector s to data point p. REMIX uses a low rank non-negative matrix factorization scheme to bi-cluster Δ into a small user-specified number of *perspectives* such that there is consensus among the detectors within a perspective. The idea of outlier perspectives is *a generalization of the idea of outlier ensembles*. A notable special case occurs when the number of perspectives equals one: here, the results from all the detectors are *ensembled* into a single set of outliers.

REMIX can be used in two modes. **(i) Simple:** An analyst can gain rapid visual understanding of the outlier space by providing two simple inputs: an *exploration budget* and the *number of perspectives*. This yields bi-clustered heatmaps of outliers that are easily interpretable as in Figure 1. **(ii) Advanced:** As discussed in Section 4.3.1, an advanced user can also *re-configure* specific modules within REMIX such as utility estimation or the enumeration of prioritized feature subspaces. Such re-configuration would steer the REMIX MIP towards alternate optimization goals while still guaranteeing exploration within a given budget and providing factorized heatmap visualizations of the diverse outlier sets.

The rest of the paper is organized as follows. We survey related work in Section 2, and provide background definitions and an overview in Section 3. In Sections 4.1 – 4.5, we present the details of feature subspace and candidate detector enumeration, cost and utility estimation, MIP for automated exploration, and perspective factorization. We present an evaluation of REMIX on real-world datasets in Section 5 and conclude in Section 6.

2 RELATED WORK

Outlier ensembles [4, 5, 23, 31] combine multiple outlier results to obtain a more robust set of outliers. Model centered ensembles combine results from different base detectors while data-centered ensembles explore different horizontal or vertical samples of the dataset and combine their results [4]. Research in the area of subspace mining [8, 18, 19, 23, 30] focuses on exploring a *diverse family of feature subspaces* which are interesting in terms of their ability to reveal outliers. The work in [28] introduces a new ensemble model that uses detector explanations to choose base detectors selectively and remain robust to errors in those detectors.

REMIX is related to ensemble outlier detection since setting the number of perspectives to one in REMIX leads to ensembling of results from the base detectors. However, REMIX has some notable distinctions: *the idea of perspectives in REMIX generalizes the notion of ensembles*; setting the number of perspectives to a number greater than one is possible in REMIX and results in complementary views of the outlier space which can be visualized as heatmaps; further, REMIX seeks to guarantee coverage not just in the space of features or feature subspaces, but also available algorithms – *subject to a budget constraint on the total time available for exploration*. None of the existing approaches in literature provide this guarantee on the exploration time.

Multiview outlier detection [9, 13, 27, 29] deals with a setting where the input consists of multiple datasets (views), and each view provides a distinct set of features for characterizing the

objects in the domain. Algorithms in this setting aim to detect objects that are outliers in each view or objects that are normal but show inconsistent class or clustering characteristics across different views. REMIX is related to multiview outlier detection through non-negative matrix factorization (NMF) which is often used here for creating a combined outlier score of objects. In REMIX, NMF is used not just for ensembling detector scores but also for factorizing detector results into multiple heatmap visualizations.

Automated exploration is a growing trend in the world of commercial data science systems [1, 3] as well as machine learning and statistical research [6, 12, 14, 20, 25, 32]. While [14] focuses on model selection, [12, 25] focus on exploration for non-parametric regression models, [1, 3, 6, 32] deal with algorithm exploration for classification models, while [20] deals with automated feature generation for classification and regression. **Budgeted allocations and recommendations** have also been studied in the context of problems other than outlier analysis [24, 26].

3 PRELIMINARIES

3.1 Baseline Algorithms

Our implementation of REMIX uses the following set $\mathcal{A}$ of five baseline outlier detection algorithms that are well-known. **1) Local outlier factor:** LOF [7] finds outliers by measuring the local deviation of a point from its neighbors. **2) Mahalanobis distance:** MD [16] detects outliers by computing the Mahalanobis distance from a point and the center of the entire dataset as outlier score. **3) Angle-based outlier detection:** ABOD [22] identifies outliers by considering the variances of the angles between the difference vectors of data points, which is more robust that distance in high-dimensional space. ABOD is robust yet time-consuming. **4) Feature-bagging outlier detection:** FBOD [23] is an ensemble method, which is based on the results of local outlier factor (LOF). During each iteration, a random feature subspace is selected. LOF then is applied to calculate the LOF scores based on the selected data subset. The final score of FBOD is the cumulative sum of each iteration. **5) Subspace outlier detection:** SOD [21]: SOD aims to detect outliers in varying subspaces of a high dimensional feature space. Specifically, for each point in the dataset, SOD explores the axis-parallel subspace spanned by its neighbors and determines how much the point deviates from the neighbors in this subspace.

3.2 Interactive Outlier Exploration

A dataset in REMIX is a real-valued matrix A with m columns and n rows. A feature subspace is a subset of columns in A. An outlier detector $D_{a,f}$ is simply a combination of an algorithm $a \in \mathcal{A}$ and a feature subspace f of A. The cost and utility values $c_{a,f}$ and $u_{a,f}$ are positive values associated with $D_{a,f}$ which are intended to be estimates of the execution cost and accuracy of $D_{a,f}$ respectively. REMIX enumerates candidate outlier detectors based on a family of prioritized feature subspaces $\mathcal{F}_p$ and a family of randomly constructed feature subspaces $\mathcal{F}_r$.

The *interactive outlier exploration problem* is a budgeted optimization problem which selects a subset of detectors with a **maximization objective** that is a linear combination of two quantities: (i) the total utility of all the selected detectors, and (ii) the total utility of the top-k selected detectors, **subject to** the following budget and
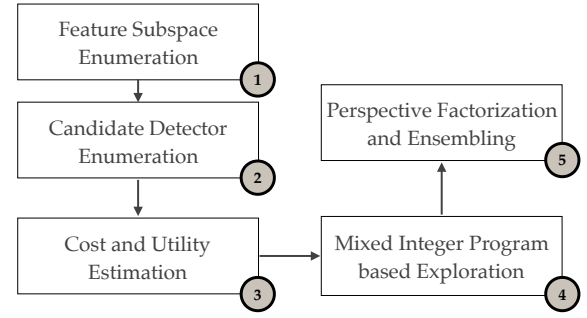
Figure 2: REMIX Components: REMIX starts by enumerating multiple feature subspaces from the given dataset, which in turn is used to enumerate candidate detectors. Next, REMIX evaluates the cost and utility of all the enumerated detectors (the meta-learning components for training the cost and utility models is not shown in this figure). The cost and utility values are used as part of a mixed integer program (MIP) which selects a subset of candidates for execution based on a utility maximization objective and budget and diversity constraints. The outlier results from the detectors executed by REMIX are factorized into perspectives, and also ensembled into a single set of results.

diversity constraints: (i) the total cost the selected detectors does not exceed the budget T_{total}, (ii) each algorithm gets a guaranteed share of the exploration budget, and (iii) each prioritized feature gets a guaranteed share of the exploration budget.

4 THE REMIX FRAMEWORK

We now describe the five components of REMIX shown in Figure 2 starting with feature subspace enumeration.

4.1 Feature Subspace and Candidate Detector Enumeration

Algorithm 1 describes feature subspace enumeration and has three parts: (i) creating a non-redundant feature bag F_{nr} (lines 1 – 13), (ii) creating a prioritized family of subspaces $\mathcal{F}_p$ (lines 14-18) using a feature ranking approach and (iii) creating a randomized family of subspaces $\mathcal{F}_r$ (lines 19 – 23) used for maximizing coverage and diversity during exploration. Redundant features $\notin F_{nr}$ are not part of subspaces in $\mathcal{F}_p$ or $\mathcal{F}_r$.

Algorithm 1 begins by initializing F_{nr} to all features in A (line 1). It then iteratively looks for a member in F_{nr} which can be considered redundant and hence dropped from F_{nr}. In order for a feature $A_{*,p} \in F_{nr}$ to be considered redundant, it needs to (i) be in a maximally correlated feature pair $\{A_{*,p}, A_{*,q}\} \subseteq F_{nr}$ (line 4), (ii) the correlation coefficient $\sigma_{p,q}$ must be above the REMIX's redundancy threshold α (line 5), and (iii) of the two features in the pair, $A_{*,p}$ must have a mean correlation with other features in F_{nr} that $\geq$ the mean correlation of $A_{*,q}$ (line 7). We set the default value of α in REMIX to 0.9 based on our experimental evaluation. It is easy to see that at the end of line 13, Algorithm 1 yields a non-redundant feature bag F_{nr} with the following property.

Algorithm 1 Feature Subspace Enumeration

Input: Data Matrix A
Output: Feature subspace families $\mathcal{F}_p$ and $\mathcal{F}_r$

1: $F_{nr} = \bigcup_{j=1}^{m} A_{*,j}$ ▷ Initialize non-redundant feature bag
2: **while** $|F_{nr}| \geq 2$ **do** ▷ Bag has at least 2 features
3:     $\forall A_{*,j} \in F_{nr}, \hat{\sigma}_j \leftarrow \frac{\sum_{(i \neq j) \wedge (A_{*,i} \in F_{nr})} \sigma_{i,j}}{|F_{nr}|-1}$ ▷ Mean correlation
4:     $(p,q) \leftarrow \arg\max_{(i \neq j) \wedge (A_{*,i} \in F_{nr}) \wedge (A_{*,j} \in F_{nr})} \sigma_{i,j}$ ▷ Max
5:     **if** $\sigma_{p,q} \geq \alpha$ **then** ▷ High max correlation?
6:        **if** $\hat{\sigma}_p \geq \hat{\sigma}_q$ **then** ▷ Greater average correlation?
7:           $F_{nr} \leftarrow F_{nr} \setminus \{A_{*,p}\}$ ▷ Drop $A_{*,p}$
8:        **else**
9:           $F_{nr} \leftarrow F_{nr} \setminus \{A_{*,q}\}$ ▷ Drop $A_{*,q}$
10:        **end if**
11:     **else break** ▷ Break out of while loop
12:     **end if**
13: **end while**
14: $\mathcal{F}_p = \Phi$ ▷ Initialize to null set
15: **for** $\ell \leftarrow 1, |F_{nr}|$ **do**
16:     $S_\ell \leftarrow$ Top-ℓ features in F_{nr} ranked by their Laplacian scores
17:     $\mathcal{F}_p \leftarrow \mathcal{F}_p \cup \{S_\ell\}$ ▷ Add prioritized subspace
18: **end for**
19: $\mathcal{F}_r = \Phi$ ▷ Initialize to null set
20: **for** $i \leftarrow 1, \Gamma$ **do**
21:     $T_i \leftarrow$ Random subspace with each feature sampled independently at random without replacement from F_{nr} with probability $\frac{1}{2}$
22:     $\mathcal{F}_r \leftarrow \mathcal{F}_r \cup \{T_i\}$ ▷ Add random subspace
23: **end for**

OBSERVATION 4.1. *For any pair of features* $\{A_{*,p}, A_{*,q}\} \subseteq F_{nr}$, $\sigma_{p,q} < \alpha$. *For any feature* $A_{*,r} \notin F_{nr}, \exists A_{*,s} \in F_{nr}, \sigma_{r,s} \geq \alpha$.

The family of randomized feature subspaces $\mathcal{F}_r$ is created for the purpose of guaranteeing feature coverage and diversity during exploration. In particular, we select Γ subspaces, where each subspace consists of features selected independently at random without replacement with probability $\frac{1}{2}$ from F_{nr}. We set the default value of Γ in REMIX to $|\mathcal{F}_p|/2$ to balance the size of the two families.

The family of prioritized feature subspaces $\mathcal{F}_p$ is created as follows. The features in F_{nr} are first sorted according to their Laplacian scores [15]. We add $|F_{nr}|$ subspaces to $\mathcal{F}_p$, where the ℓth subspace is the set of top-ℓ features in F_{nr} ranked by their Laplacian scores. We now provide a brief justification for our use of the Laplacian score. Due to lack of space, refer the reader to [15] for the exact details of the Laplacian computation. The Laplacian score is computed as a way to reflect a feature's ability to preserve locality. In particular, consider the projection A' of all points in A onto the subspace F_{nr}; now, consider the r-nearest neighborhood of points in A'. Features that respect this r-nearest neighborhood provide a better separation of the inlier class from outlier class within the data. We note that our experimental evaluation in Section 5 consistently demonstrates improved outlier detection accuracy with $\mathcal{F}_p$ and $\mathcal{F}_r$ as opposed to purely $\mathcal{F}_r$. We also discuss potential alternatives for prioritized feature subspace construction in Section 4.3.1.

We enumerate candidate detectors by a cartesian product of $\mathcal{F}_p \cup \mathcal{F}_r$ and the set of baseline detection algorithms $\mathcal{A}$.

4.2 Cost Estimation

The cost of a candidate detector can be modeled as a function of its algorithm as well as the size of its feature subspace f (n and $|f|$). The runtime complexity of algorithms in the 'big-O' notation provides an asymptotic relationship between cost and input size; however, in REMIX, we seek a more refined model which accounts for lower order terms and the constants hidden by 'big-O'. We do this by training a multivariate linear regression model for *each algorithm*. Specifically, consider the following polynomial: $(1 + |f| + n + \log|f| + \log n)^3 - 1$. There are $\binom{7}{3} - 1$ distinct terms in the expansion of this polynomial which can be derived exactly given a feature subspace. The terms[1] form the explanatory variables while cost is the dependent variable in the linear regression model.

4.3 Utility Estimation

Given a detector $D_{a,f}$, the utility estimation algorithm (Algorithm 2) first normalizes the features in f, and computes a variety of feature-level statistics for each feature $\psi \in f$. These statistics include the Laplacian score which is a useful measure for unsupervised feature selection (Section 4.1), the standard deviation, skewness which measures the assymetry of the feature distribution, kurtosis which measures the extent to which the feature distribution is heavy-tailed, and entropy which is a measure of the information content in the feature. We note that these computations are done once for each feature in F_{nr} and is reused for any given any detector. Next, the algorithm computes the meta-feature vector $MFV(f)$ of *feature-subspace-level* statistics by combining feature-level statistics. For instance, consider the Laplacian scores of all the features in f; the mean, median, median absolute deviation, min, max, and standard deviation of all the Laplacian scores provide 6 of the 30 distinct components of $MFV(f)$ in this step. Finally, it uses an algorithm specific utility model $U_a(MFV(f))$ to estimate the utility $u_{a,f}$ of $D_{a,f}$. REMIX trains five distinct linear regression models $U_{LOF}, U_{MD}, U_{ABOD}, U_{LOF}$, and U_{SOD} corresponding to each algorithm. These models are trained based on distinct *expert labeled* datasets from the outlier dataset repository [2]. The explanatory variables for this linear regression are the feature-subspace-level statistics described in Algorithm 2. The dependent variable is the detection accuracy, measured as the fraction of the data points on which both the detector and the expert labeled ground truth agree on the outlier characterization.

4.3.1 Discussion. Estimating or even defining the utility of a detector is *significantly harder* than estimating its cost. The goal of utility estimation in REMIX is *not* to learn a perfect model; rather, the goal is merely to learn a utility model which can effectively steer the solution of the mixed integer program (MIP) used by REMIX for exploration (Section 4.4). Our experiments in Section 5 demonstrate that this is indeed the case with REMIX. Further, REMIX is intended to be a flexible framework where alternative mechanisms for utility estimation can be plugged in. For instance, consider Cumulative Mutual Information (CMI) metric and the Apriori-style

---

[1]The exponent 3 suffices to model the cost of most known outlier detection algorithms

---

Algorithm 2 Utility Estimation

Input: A candidate detector $D_{a,f}$
Output: The utility $u_{a,f}$ of the detector

1: $\forall \psi \in f$, normalize ψ ▷ One time procedure applied to F_{nr}
2: $\forall \psi \in f$, extract the following 5 feature-level statistics: Laplacian score, standard deviation, skewness, kurtosis, and entropy
  ▷ One time procedure applied to F_{nr}
3: Extract the following 30 feature-subspace-level statistics: mean, median, median absolute deviation, min, max, and standard deviation for each of the 5 feature-level statistics extracted in Step 2. Let $MFV(f)$ contain these feature subspace-level statistics.
4: Return $u_{a,f} = U_a(MFV(f))$, where U_a is the utility estimation model learnt for algorithm a

---

feature subspace search algorithm presented in [8] for subspace outlier detection. REMIX can use CMI as the detector utility, and this Apriori-style algorithm as an alternative for enumerating prioritized feature subspaces. We chose the meta-learning approach in our implementation since this approach is algorithm agnostic and hence can be generalized easily.

4.4 MIP based Exploration

REMIX executes only a subset of detectors enumerated by its detector enumeration component (Section 4.1). This subset is determined by solving the following mixed integer program (MIP).

$$\max \underbrace{\sum_{a \in \mathcal{A}} \sum_{f \in \mathcal{F}_p \cup \mathcal{F}_r} z_{a,f} u_{a,f}}_{\text{utility of top-k detectors}} + \lambda \underbrace{\sum_{a \in \mathcal{A}} \sum_{f \in \mathcal{F}_p \cup \mathcal{F}_r} y_{a,f} u_{a,f}}_{\text{utility of all detectors}} \quad (1)$$

$$\sum_{a \in \mathcal{A}} \sum_{f \in \mathcal{F}_p \cup \mathcal{F}_r} y_{a,f} c_{a,f} \leq T_{total} \quad (2)$$

$$\forall a \in \mathcal{A} : \sum_{f \in \mathcal{F}_p \cup \mathcal{F}_r} y_{a,f} c_{a,f} \geq \frac{T_{total}}{2 \cdot |\mathcal{A}|} \quad (3)$$

$$\forall f \in \mathcal{F}_p : \sum_{a \in \mathcal{A}} y_{a,f} c_{a,f} \geq \frac{T_{total}}{2 \cdot |\mathcal{F}_p|} \quad (4)$$

$$\forall a \in \mathcal{A}, \forall f \in \mathcal{F}_p \cup \mathcal{F}_r : z_{a,f} \leq y_{a,f} \quad (5)$$

$$\sum_{a \in \mathcal{A}} \sum_{f \in \mathcal{F}_p \cup \mathcal{F}_r} z_{a,f} \leq k \quad (6)$$

$$\forall a \in \mathcal{A}, \forall f \in \mathcal{F}_p \cup \mathcal{F}_r : y_{i,j} \in \{0, 1\} \quad (7)$$

$$\forall a \in \mathcal{A}, \forall f \in \mathcal{F}_p \cup \mathcal{F}_r : z_{i,j} \in \{0, 1\} \quad (8)$$

Given an algorithm a and a feature subspace f, $y_{a,f}$ is the binary indicator variable in the MIP which determines if the detector $D_{a,f}$ is chosen for execution in the MIP solution. Recall that $c_{a,f}$ denotes the estimated cost of $D_{a,f}$. We observe the following.

OBSERVATION 4.2. *Constraint 2 guarantees that in any feasible solution to the MIP, the total cost of the selected detectors is $\leq T_{total}$.*

The value of the total exploration budget T_{total} is provided by the analyst as part of her interaction with REMIX.

OBSERVATION 4.3. *Constraint 3 guarantees that in any feasible solution to the MIP, each algorithm is explored for an estimated duration of time which is $\geq \frac{T_{total}}{2 \cdot |\mathcal{A}|}$.*

OBSERVATION 4.4. *Constraint 4 guarantees that in any feasible solution to the MIP, each prioritized feature is explored for an estimated duration of time which is $\geq \frac{T_{total}}{2 \cdot |\mathcal{F}_p|}$. This also implies that the exploration focuses at least half its total time on prioritized features.*

Consider the binary indicator variable $z_{a,f}$ corresponding to the detector $D_{a,f}$. Constraint 5 ensures that in any feasible solution to the MIP, $z_{a,f}$ can be 1 only if it is chosen in the solution (i.e., $y_{a,f} = 1$). Constraint 6 ensures at most k of the detectors chosen by the solution have their z-values set to 1. Now consider an *optimal* solution to the MIP. Since utility values are non-negative for all detectors, the first part of the objective function $\sum_{a \in \mathcal{A}} \sum_{f \in \mathcal{F}_p \cup \mathcal{F}_r} z_{a,f} u_{a,f}$ is maximized when exactly k detectors have their z-values set to 1 (and not fewer) **and** when the detectors whose z-values are set to 1 are the ones with the highest utilities amongst the selected detectors. This leads us to the following guarantee.

THEOREM 4.5. *The optimal solution to the MIP maximizes the sum of utilities of the top-k detectors with highest utilities and the total utility of all the selected detectors scaled by a factor λ.*

We set $k = 10$ and $\lambda = 1$ in our implementation of REMIX which balances the utility of the top-10 detectors vs the total utility of all the selected detectors.

4.5 Perspective Factorization

Each detector executed by REMIX provides an outlier score for each data point and the results from different detectors could be potentially divergent. We now present a factorization technique called **NMFE** (non-negative matrix factorization and ensembling) for bi-clustering the detection results into a few succinct *perspectives*. All detectors within a perspective agree on how they characterize outliers although there could be disagreement across perspectives.

Let $\Delta_{s,p}$ be the outlier score assigned by detector s for data point p normalized across data points to be in the range $[0, 1]$. Consider the matrix of outlier scores $\Delta \in [0, 1]^{t \times n}$, where t is the number of detectors executed by REMIX and n is the number of data points. We perform a rank-g non-negative matrix factorization of $\Delta \approx \Lambda \Omega^\top$, where $\Lambda \in \mathbb{R}^{t \times g}$ and $\Omega \in \mathbb{R}^{n \times g}$, by minimizing the Kullback-Leibler divergence between Δ and $\Lambda \Omega^\top$ [11]:

$$\min_{\Lambda, \Omega \geq 0} \sum_{s,p} \Delta_{s,p} log \frac{\Delta_{s,p}}{\Lambda_{s,*} \Omega_{p,*}} - \Delta_{s,p} + \Lambda_{s,*} \Omega_{p,*} \quad (9)$$

The matrix $\Lambda \Omega^T$ by definition can be expressed as a sum of g rank-1 matrices whose rows and columns correspond to detectors and data points respectively. In any of these rank-1 matrices, every row (column) is a scaled multiple of any other non-zero row (column), and every entry is between 0 and 1. These properties make it possible for the rank-1 matrices to be visualized as *perspectives*, or heatmaps where the intensity of a heatmap cell is the value of the corresponding entry in the perspective, as shown in Figure 1. The number of perspectives g is specified by the user as an

input to REMIX. Setting $g = 1$ simply results in a direct averaging (ensembling) of all the detector results into a single perspective.

5 EXPERIMENTS

5.1 Data Collection

We chose 98 datasets for our study from the outlier dataset repository [2] of the Delft University of Technology. This corpus contains outlier detection datasets labeled by domain experts. These datasets span varied domains such as flowers, breast cancers, heart diseases, sonar signals, genetic diseases, arrhythmia abnormals, hepatitis, diabetes, leukemia, vehicles, housing, and satellite images. Each dataset has benchmark outlier labels, with an entry 1 representing outliers and 0 representing normal points.

Figure 3 illustrate some statistics of the 98 data sets. Specifically, Figure 3(a) shows the numbers of features for each dataset sorted in a descending order. In this figure, we can observe that most of the datasets contains less than 100 features while only a small portion of these datasets have more than 1000 features. Figure 3(b) shows the outlier ratio to the total number of data points for each dataset sorted in a descending order. In this figure, we can find that the outlier ratios of more than 50% of the datasets are less than 23.2%.
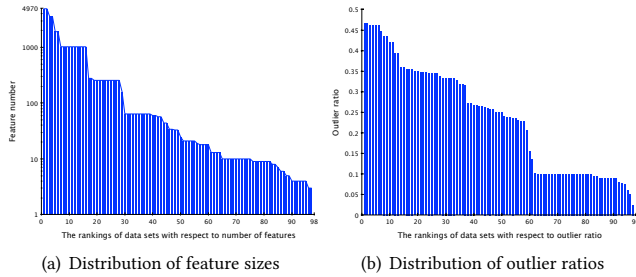

(a) Distribution of feature sizes   (b) Distribution of outlier ratios

Figure 3: Statistics of the experimental datasets

5.2 Cost and Utility Estimation

Among the 98 datasets, we used 96 datasets to train and test the cost and utility estimation models for each of the 5 baseline algorithms in our REMIX implementation (with a 70%-30% split for train vs test). Figures 4(a) and 4(b) present the performance of the cost and utility models for the LOF algorithm from a specific run.

Recall that the utility $u_{a,f}$ estimates the fraction of the data points on which the detector $d_{a,f}$ and the expert labels agree on the outlier characterization. Consider the Hamming distance between the outlier bit vector (1 if outlier, 0 otherwise) created by the detector and the outlier bit vector created by the expert labels. Clearly, this Hamming distance equals $n \cdot (1 - u_{a,f})$ where n is the number of data points. Figure 4(b) plots the estimated Hamming distance on the y-axis and the actual distance as observed by running the detector on the x-axis. The red lines in Figures 4(a) and 4(b) represent ideal predictors. As expected, the cost estimator clearly performs better than the utility estimator. However, as mentioned in Section 4.3.1 our goal in utility estimation is not a perfect predictive model but merely to steer the MIP towards better solutions. We demonstrate this to be the case in our next set of experiments.
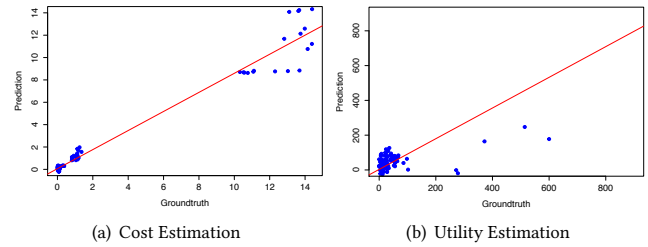

(a) Cost Estimation   (b) Utility Estimation

Figure 4: Cost and Utility Estimation

5.3 Detection Accuracy and Cost

We now present a detailed study of the detection accuracy vs cost of REMIX on three datasets from the corpus. The first dataset is cardiac arrhythmia (CA). Irregularity in heart beat may be harmless or life threatening. The dataset contains medical records like age, weight and patient's electrocardiograph related data of arrhythmia patients and outlier healthy people. The task is to spot outlier healthy people from arrhythmia patients. The second dataset is a skewed subsample of CA (SkewCA) which we created for the sake of diversity by subsampling from CA: 90% of the points in SkewCA were points that were considered normal and the remaining 10% were considered outliers by expert labels. The third dataset is about sonar signals (SONAR) bounced off of a metal cylinder. The dataset contains outlier sonar signals bounced off a roughly cylindrical rock. The task is to separate outlier rock related sonar signals from cylinder-related sonar signals.

5.3.1 Evaluation Metrics. Recall that the REMIX perspective factorization scheme can be used as an outlier ensembling technique simply by setting the number of perspectives to 1. We use REMIX in this ensembling mode for the rest of our experiments. We now define Precision@N, Recall@N, and F-measure@N which we use to compare the detection accuracy of REMIX with various other approaches. Let 1 denote outlier label and 0 denote the label for normal points in the expert labeled data.

Precision@N Given the top-N list of data points $\mathcal{L}_N$ sorted in a descending order of the predicted outlier scores, the precision is defined as: Precision@$N = \frac{|\mathcal{L}_N \bigcap \mathcal{L}_{=1}|}{N}$ where $\mathcal{L}_{=1}$ are the data points with expert outlier label = 1.

Recall@N Given the top-N list of data points $\mathcal{L}_N$ sorted in a descending order of the predicted outlier scores, the recall is defined as: Recall@$N = \frac{|\mathcal{L}_N \bigcap \mathcal{L}_{=1}|}{|\mathcal{L}_{=1}|}$ where $\mathcal{L}_{=1}$ are the outlier data points with label = 1.

F-measure@N F-measure@N incorporates both precision and recall in a single metric by taking their harmonic mean: F@$N = \frac{2 \times \text{Precision@}N \times \text{Recall@}N}{\text{Precision@}N + \text{Recall@}N}$

5.3.2 Baseline Algorithms. We report the performance comparison of REMIX vs baseline algorithms in terms of Precision, Recall, and F-measure. In this experiment, we provide sufficient time for the baseline algorithms in $\{LOF, MD, ABOD, FBOD, SOD\}$ to complete their executions. Meanwhile, we set a limited exploration time budget of 0.5 second for REMIX.
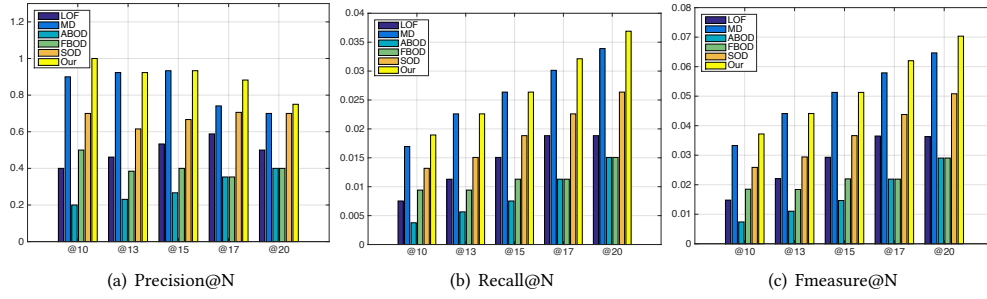
(a) Precision@N

(b) Recall@N

(c) Fmeasure@N

Figure 5: Comparison of REMIX with baseline algorithms on the CA dataset

(a) Precision@N

(b) Recall@N

(c) Fmeasure@N
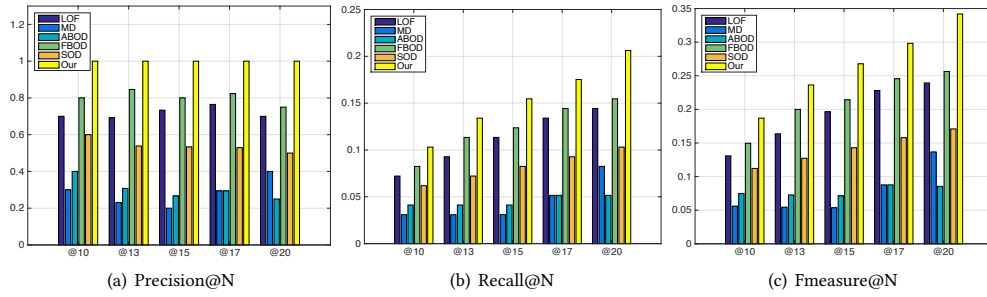
Figure 6: Comparison of REMIX with baseline algorithms on SONAR dataset

(a) Precision@N
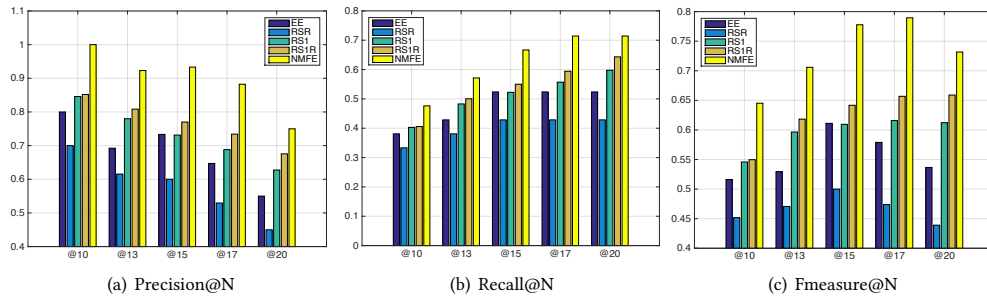
(b) Recall@N

(c) Fmeasure@N

Figure 7: Comparison of REMIX with other exploration and ensemble strategies on the SkewCA dataset

(a) Precision@N

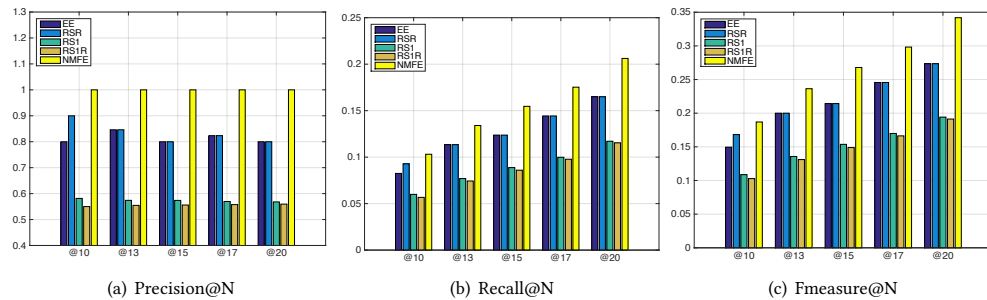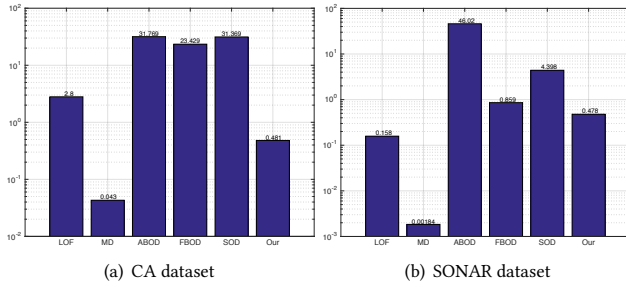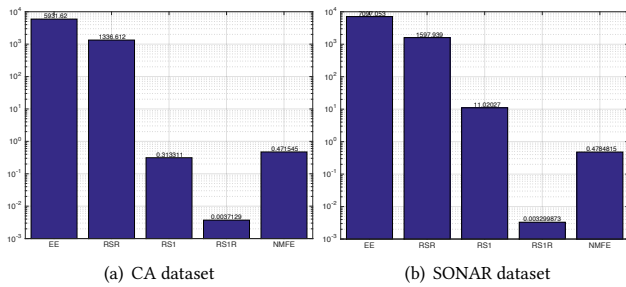(b) Recall@N

(c) Fmeasure@N

Figure 8: Comparison of REMIX with other exploration and ensemble strategies on the SONAR dataset

Figure 9: Comparison of execution costs of REMIX and baseline algorithms

Figure 10: Comparison of execution costs of REMIX and other exploration strategies

Results on Effectiveness Comparison. Figure 5 shows this on the CA dataset. REMIX outperforms the five baseline algorithms in terms Precision@N, Recall@N, and Fmeasure@N (N=10, 13, 15, 17, 20). Figure 6 shows this on the SONAR dataset. REMIX is consistently better than the baseline algorithms in terms Precision@N, Recall@N, and Fmeasure@N (N=10, 13, 15, 17, 20).

Results on Efficiency Comparison. Figure 9 jointly shows this on the CA and SONAR datasets. Mahalanobis distance (MD) takes the least time; Angle-based outlier detection (ABOD) takes the most time as angle is expensive to compute; REMIX falls in the middle of this cost spectrum.

5.3.3 The Effectiveness and Efficiency of REMIX NMFE. We study the effectiveness of the REMIX NMFE ensembling strategy, in which we used the rank-1 NMF to factorize the outlier score matrix $\Delta \approx \Lambda \Omega^{\top}$ and treat Ω as the predicted ensemble outlier scores. Let t denote the number of detectors selected in the REMIX MIP solution. Let C be the set of all candidate detectors enumerated by REMIX. We compared REMIX with the following exploration and ensembling strategies: **(1) Exhaustive Ensemble (EE):** we execute all the detectors in C and averaged all the outlier scores; **(2) Randomly select t detectors (RSR):** we randomly select and execute t detectors from C, and then average the outlier scores of these detectors; **(3) Randomly select 1 detector (RS1):** we randomly select one detector and use its results; **(4) Randomly select 1 detector in the MIP solution (RS1R):** we randomly select one detector in the MIP solution and use its results; In this experiment,

we provide sufficient time for all strategies complete their execution. We set a limited time budget of 0.5 seconds for REMIX.

Results on Effectiveness Comparison. Figure 7 shows that on the SkewCA dataset our strategy outperforms the other exploration and ensembling strategies in terms of Precision@N, Recall@N, and Fmeasure@N (N=10, 13, 15, 17, 20). Figure 8 shows that on the sonar signal dataset, REMIX is consistently better than the other strategies in terms of Precision@N, Recall@N, and Fmeasure@N (N=10, 13, 15, 17, 20).

Results on Efficiency Comparison. Figure 10(a) shows that on the CA dataset REMIX takes 0.48 second, and EE, RSR, RS1 require much more time. While RS1R takes only 0.0037 second, its detection accuracy is lower than ours. Figure 10(b) shows on the SONAR dataset, our strategy takes only 0.49 second, which is much less than the time costs of EE, RSR, RS1. While RS1R takes only 0.0033 second, our method outperforms RS1R with respect to detection accuracy.

6 CONCLUSIONS AND FUTURE DIRECTIONS

In this paper, we presented REMIX, a modular framework for outlier exploration which is the first to study the problem of outlier detection in an interactive setting. At the heart of REMIX is an optimization approach which systematically steers the selection of base outlier detectors in a manner that is sensitive to their execution costs, while maximizing an aggregate utility function of the solution and also ensuring diversity across algorithmic choice points present in exploration. Data analysts are naturally interested in extracting and interpreting outliers through multiple detection mechanisms since distinct outliers could lead to distinct actionable insights within the application domain. REMIX facilitates this understanding in a practical manner by shifting the burden of exploration away from the analyst through automation, and by summarizing the results of automated exploration into a few coherent heatmap visualizations called perspectives.

We believe many of the techniques presented in this paper could be of independent interest to other machine learning problems. We are interested in extending the REMIX exploratory approach beyond outlier detection to clustering of high-dimensional data. Another interesting direction of research is sequential recommendations for visual outlier exploration: in particular, we are interested in approaches for presenting outlier insights approximately through a sequence of visualizations (like heatmaps) such that both the length of this sequence as well the perceptual error involved across the visualizations is minimized while coverage across the various exploratory choice points is maximized. Also the study of outlier aspect mining in an interactive setting – which deals with the inverse problem of finding explanatory features which characterize a given set of points as outliers in a cost sensitive manner – presents a new and interesting direction of research.

REFERENCES

[1] 2017. Data Robot. http://www.datarobot.com. (2017).
[2] 2017. Outlier Detection Datasets. http://homepage.tudelft.nl/n9d04/occ/index.html. (2017).
[3] 2017. SYTREE. http://www.skytree.net. (2017).
[4] Charu C Aggarwal. 2013. *Outlier analysis.* Springer Science & Business Media.
[5] Charu C Aggarwal and Saket Sathe. 2015. Theoretical Foundations and Algorithms for Outlier Ensembles. *ACM SIGKDD Explorations Newsletter* 17, 1 (2015), 24–47.
[6] Alain Biem, Maria Butrico, Mark Feblowitz, Tim Klinger, Yuri Malitsky, Kenney Ng, Adam Perer, Chandra Reddy, Anton Riabov, Horst Samulowitz, Daby M. Sow, Gerald Tesauro, and Deepak S. Turaga. 2015. Towards Cognitive Automation of Data Science. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA.* 4268–4269. http://www.aaai.org/ocs/index.php/AAAI/AAAI15/paper/view/9721
[7] Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander. 2000. LOF: identifying density-based local outliers. In *ACM sigmod record*, Vol. 29. ACM, 93–104.
[8] Klemens Bhm, Fabian Keller, Emmanuel Mller, Hoang Vu Nguyen, and Jilles Vreeken. 2013. CMI: An Information-Theoretic Contrast Measure for Enhancing Subspace Cluster and Outlier Detection.. In *SDM.* SIAM, 198–206. http://dblp.uni-trier.de/db/conf/sdm/sdm2013.html#BohmKMNV13
[9] Santanu Das, Bryan L Matthews, Ashok N Srivastava, and Nikunj C Oza. 2010. Multiple kernel learning for heterogeneous anomaly detection: algorithm and aviation safety case study. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining.* ACM, 47–56.
[10] Kyriaki Dimitriadou, Olga Papaemmanouil, and Yanlei Diao. 2014. Explore-by-example: An Automatic Query Steering Framework for Interactive Data Exploration. In *Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data (SIGMOD '14).* ACM, New York, NY, USA, 517–528. DOI:http://dx.doi.org/10.1145/2588555.2610523
[11] Chris H. Q. Ding, Tao Li, and Michael I. Jordan. 2010. Convex and Semi-Nonnegative Matrix Factorizations. *IEEE Trans. Pattern Anal. Mach. Intell.* 32, 1 (Jan. 2010), 45–55. DOI:http://dx.doi.org/10.1109/TPAMI.2008.277
[12] David Duvenaud, James Robert Lloyd, Roger Grosse, Joshua B. Tenenbaum, and Zoubin Ghahramani. 2013. Structure Discovery in Nonparametric Regression through Compositional Kernel Search. In *Proceedings of the 30th International Conference on Machine Learning.*
[13] Jing Gao, Wei Fan, Deepak Turaga, Srinivasan Parthasarathy, and Jiawei Han. 2011. A spectral framework for detecting inconsistency across multi-source object relationships. In *Data Mining (ICDM), 2011 IEEE 11th International Conference on.* IEEE, 1050–1055.
[14] R.B. Grosse, R. Salakhutdinov, W.T. Freeman, and J.B. Tenenbaum. 2012. Exploiting compositionality to explore a large space of model structures. In *Uncertainty in Artificial Intelligence.*
[15] Xiaofei He, Deng Cai, and Partha Niyogi. 2005. Laplacian score for feature selection. In *Advances in neural information processing systems.* 507–514.
[16] Victoria J Hodge and Jim Austin. 2004. A survey of outlier detection methodologies. *Artificial intelligence review* 22, 2 (2004), 85–126.
[17] Alexander Kalinin, Ugur Cetintemel, and Stan Zdonik. 2014. Interactive Data Exploration Using Semantic Windows. In *SIGMOD Conference.* http://database.cs.brown.edu/papers/sw-paper.pdf
[18] Fabian Keller, Emmanuel Müller, and Klemens Böhm. 2012. HiCS: high contrast subspaces for density-based outlier ranking. In *Data Engineering (ICDE), 2012 IEEE 28th International Conference on.* IEEE, 1037–1048.

[19] Fabian Keller, Emmanuel Mller, Andreas Wixler, and Klemens Bhm. 2013. Flexible and adaptive subspace search for outlier analysis.. In *CIKM*, Qi He, Arun Iyengar, Wolfgang Nejdl, Jian Pei, and Rajeev Rastogi (Eds.). ACM, 1381–1390. http://dblp.uni-trier.de/db/conf/cikm/cikm2013.html#KellerMWB13
[20] Udayan Khurana, Deepak Turaga, Horst Samulowitz, and Srinivasan Parthasarathy. 2016. Cognito: Automated Feature Engineering for Supervised Learning. *Proceedings of the International Conference on Data Mining* (2016).
[21] Hans-Peter Kriegel, Peer Kröger, Erich Schubert, and Arthur Zimek. 2009. Outlier detection in axis-parallel subspaces of high dimensional data. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining.* Springer, 831–838.
[22] Hans-Peter Kriegel, Arthur Zimek, and others. 2008. Angle-based outlier detection in high-dimensional data. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining.* ACM, 444–452.
[23] Aleksandar Lazarevic and Vipin Kumar. 2005. Feature bagging for outlier detection. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining.* ACM, 157–166.
[24] Lei Li, Ding-Ding Wang, Shun-Zhi Zhu, and Tao Li. 2011. Personalized news recommendation: a review and an experimental investigation. *Journal of computer science and technology* 26, 5 (2011), 754–766.
[25] James Robert Lloyd, David Duvenaud, Roger Grosse, Joshua B. Tenenbaum, and Zoubin Ghahramani. 2014. Automatic Construction and Natural-Language Description of Nonparametric Regression Models. In *Association for the Advancement of Artificial Intelligence (AAAI).*
[26] Tyler Lu and Craig Boutilier. 2011. Budgeted social choice: From consensus to personalized decision making. In *IJCAI*, Vol. 11. 280–286.
[27] Alejandro Marcos Alvarez, Makoto Yamada, Akisato Kimura, and Tomoharu Iwata. 2013. Clustering-based anomaly detection in multi-view data. In *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management.* ACM, 1545–1548.
[28] Alex Memory and Ted Senator. 2015. Towards Robust Anomaly Detection Ensembles using Explanations. In *In the ODDx3 workshop of KDD2015.*
[29] Emmanuel Muller, Ira Assent, Patricia Iglesias, Yvonne Mulle, and Klemens Bohm. 2012. Outlier ranking via subspace analysis in multiple views of the data. In *Data Mining (ICDM), 2012 IEEE 12th International Conference on.* IEEE, 529–538.
[30] Emmanuel Müller, Matthias Schiffer, and Thomas Seidl. 2011. Statistical selection of relevant subspace projections for outlier ranking. In *Data Engineering (ICDE), 2011 IEEE 27th International Conference on.* IEEE, 434–445.
[31] Shebuti Rayana and Leman Akoglu. 2015. Less is More: Building Selective Anomaly Ensembles. *arXiv preprint arXiv:1501.01924* (2015).
[32] Ashish Sabharwal, Horst Samulowitz, and Gerald Tesauro. 2016. Selecting Near-Optimal Learners via Incremental Data Allocation. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA.* 2007–2015. http://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/view/12524
[33] Tarique Siddiqui, Albert Kim, John Lee, Karrie Karahalios, and Aditya Parameswaran. 2016. Effortless Data Exploration with Zenvisage: An Expressive and Interactive Visual Analytics System. *Proc. VLDB Endow.* 10, 4 (Nov. 2016), 457–468. DOI:http://dx.doi.org/10.14778/3025111.3025126
[34] Abdul Wasay, Manos Athanassoulis, and Stratos Idreos. 2015. Queriosity: Automated Data Exploration.. In *BigData Congress*, Barbara Carminati and Latifur Khan (Eds.). IEEE, 716–719. http://dblp.uni-trier.de/db/conf/bigdata/bigdata2015.html#WasayAI15
[35] Arthur Zimek, Ricardo J.G.B. Campello, and Jörg Sander. 2014. Ensembles for Unsupervised Outlier Detection: Challenges and Research Questions a Position Paper. *SIGKDD Explor. Newsl.* 15, 1 (March 2014), 11–22. DOI:http://dx.doi.org/10.1145/2594473.2594476