Consiglio Nazionale delle Ricerche
Istituto di Calcolo e Reti ad Alte Prestazioni

Outlying Property Detection with Numerical Attributes

Fabrizio Angiulli[1], Fabio Fassetti [1],
Giuseppe Manco[2], Luigi Palopoli[1]

RT-ICAR-CS-11-05 **Ottobre 2011**

1 Università Della Calabria – Dipartimento di Elettronica Informatica e Sistemistica (DEIS) – Via P. Bucci 41C, 87036 Rende, Italy, URL: *www.unical.it*

2 Consiglio Nazionale delle Ricerche, Istituto di Calcolo e Reti ad Alte Prestazioni (ICAR) – Sede di Cosenza, Via P. Bucci 41C, 87036 Rende, Italy, URL: *www.icar.cnr.it*

Outlying Property Detection with Numerical Attributes

Fabrizio Angiulli* Fabio Fassetti† Giuseppe Manco‡ Luigi Palopoli§

Abstract

The *outlying property detection problem* (OPDP) is the problem of discovering the properties distinguishing a given object, known in advance to be an outlier in a database, from the other database objects. This problem has been recently analyzed focusing on categorical attributes only. However, numerical attributed very relevant and widely used in databases. Therefore, in this paper, we analyze the OPDP within a context where also numerical attributes are taken into account, which represents a relevant case left open in the literature. As major contributions, we present an efficient parameter-free algorithm to compute the measure of object exceptionality we introduce, and propose a unified framework for mining exceptional properties in the presence of both categorical and numerical attributes.

1 Introduction

While anomaly detection in datasets has been one of the most widely investigated problems in data mining, the related problem of anomaly justification received less attention in the literature. In a recent paper [3], the OPDP was studied, that is, given a dataset characterized by certain attributes and a single input object known in advance to be anomalous in that dataset, find a set of attributes explaining why this object is actually anomalous or, in other terms, detect the unexpected properties (if any) this anomalous object possesses. The cited paper considers a case where attributes whose values justify the given object anomaly are categorical. In several relevant application cases, though, the input dataset has numerical attributes which may well account for the anomaly of a given input anomalous object. The appropiate treatment of such non-categorical attributes was a problem left open in [3] and precisely the problem we face in this paper.

As an example, assume you are analyzing health parameters of a sick patient, which include several numerical features such as body temperature, blood pressure measurements and others. If an history of healthy patients is available, then it is relevant to single out that subset of those parameters that mostly differentiate the sick patient from the healthy population. It is important to highlight here that the abnormal individual, whose peculiar characteristics we want to detect, is provided as an input to the problem, that is, this individual has been recognized as anomalous in advance by the virtue of some external information, mean or procedure.

The main contribution of this work amounts to provide the *outlierness* measure, representing a refined generalization of that proposed in [3], which is able to quantify the exceptionality of a given *property* featured by the given input anomalous object with respect to a reference data population. In particular, this measure is able to quantify the degree of "unbalanceness" between the frequency of the value under consideration and the frequencies of the rest of the database values. This is done by taking into account the curve of the cumulative distribution function (*cdf*) associated with the occurrence probability of the domain values. It is worth noting that our measure is able to correctly recognize exceptional properties independently of the form of the underlying probability density function (*pdf*), since it compares the occurrence probabilities of the domain values rather than directly comparing the domain values themselves. As a further contribution, we present an efficient parameter-free algorithm that computes outlierness in time $O(n \log n)$. Thus, we propose an approach able to uniformly mining exceptional properties in the presence of both categorical and numerical attributes, so that a fully automated support is provided to decode those properties determining the abnormality of the given object within the reference data context.

To illustrate, given a dataset DB (stored in the form of a relational table) and an object o deemed to be abnormal (on the basis of available external knowledge), we adopt shall the the point that a property, or set of attributes, witnesses the abnormality of the object o if the combination of values o assumes on these attributes is very infrequent with respect to the overall distribution of the attribute values in the data set: to this end, in the following, we introduce a measure by which it is possible to faithfully capture how much a set of attributes should be considered relevant in explaining the abnormality of the given object o. In the following, we shall also carry

*DEIS, University of Calabria, f.angiulli@deis.unical.it
†DEIS, University of Calabria, f.fassetti@deis.unical.it
‡ICAR-CNR, manco@icar.cnr.it
§DEIS, University of Calabria, palopoli@deis.unical.it

out a discussion on the characteristics of the measure and its relationship with related measures (Section 2). ¿From this discussion, and from some of the results of the experiments, it shall clearly turn out that our measure is a sensible and significant one in the context of the analyzed abnormality explanation problems.

The rest of the paper is organized as follows. Section 2 introduces the mining task and discusses relationship and differences with outlier detection. Section 3.1 introduces the outlierness measure and the concept of explanation. Section 4 describes the method for computing outlierness and determining associated explanations. Section 5 discusses experimental results, including a real-life case study. Finally, Section 6 presents conclusions and discusses future work.

2 Background and Related Work

To begin with, we next introduce some preliminary definitions and fix the notation. An *attribute* a is an identifier with an associated domain, also denoted $\mathbb{D}(a)$. Let $\mathbf{A} = a_1, \ldots, a_m$ be a set of m attributes[1]. Then, an *object* o on $\mathbf{A}$ is a tuple $o = \langle v_1, \ldots, v_m \rangle$ of m values, such that each v_i is a value in the domain of a_i. The *value* v_i associated with the attribute a_i in o will be denoted by $o[a_i]$. A *database* DB on a set of attributes $\mathbf{A}$ is a multi-set (that is, duplicate elements are allowed) of objects on $\mathbf{A}$.

In the following, first the outlier detection approach is recalled and, then, the task here pursued is introduced, and differences with outlier detection and related works are pointed out.

Outlier detection. Give a database DB over an attribute schema $\mathbf{A}$, we aim at studying the notion of *outlier*, i.e., an object o *in*DB that is "exceptional", as it significantly differs from the rest of the data in DB. The notion of outlierness has been extensively studied in the current literature.

Outlier detection is a knowledge discovery task which has its roots in statistics, machine learning, and data mining [14, 23, 28]. A classical definition of outlier is provided in [14]: "An *outlier* is an observation that differs so much from other observations as to arouse suspicion that it was generated by a different mechanism". Thus, *outlier detection* in data mining considers the following task: "Given a set of data points or objects, find the objects that are considerably dissimilar, exceptional or inconsistent with respect to the remaining data". These exceptional objects as also referred to as outliers. Outlier detection represents an active research field that has many applications in all those domains that can lead to illegal or abnormal behavior, such as fraud detection, network intrusion detection, medical diagnosis, and many others [15, 7].

Approaches to outlier detection can be classified in supervised, semi-supervised, and unsupervised. Supervised methods exploit the availability of a labeled data set, containing observations already labeled as normal and abnormal, in order to build a model of the normal class [8]. Since usually normal observations are the great majority, these data sets are unbalanced and specific classification techniques must be designed to deal with the presence of rare classes. Semi-supervised methods assume that only normal examples are given. The goal is to find a description of the data, that is a rule partitioning the object space into an accepting region, containing the normal objects, and a rejecting region, containing all the other objects [26]. These methods are also called one-class classifiers or domain description techniques, and they are related to novelty detection since the domain description is used to identify objects significantly deviating form the training examples. Unsupervised methods search for outliers in an unlabelled data set by assigning to each object a score which reflects its degree of abnormality. Scores are usually computed by comparing each object with objects belonging to its neighborhood. Among the unsupervised approaches to detect outliers there are statistical-based [5], deviation-based [4], distance-based [18], density-based [6, 16], projection-based [1], MDEF-based [24], angle-based [20], isolation forest-based [22], local outlier probability-based [19], and others [7].

Outlying property detection. It must be noticed that the problem addressed here is completely different from supervised and semi-supervised outlier detection, and, moreover, is to be considered orthogonal to the unsupervised outlier detection task. Indeed, in outlier detection, a set of observations is given in input and we are interested in discovering those observations (i.e., the outliers) that are mostly dissimilar from the remaining ones, while here the outliers (anomalous subpopulations) are given in input and we are interested in discovering the motivations underlying their abnormality.

As a matter of fact, the focus of this paper is the discovery of outlying properties: In practice, we are interested in unveiling the hidden structures that make an object $o \in DB$ special w.r.t. a population in DB. To this purpose, we assume that the set $o_1, \ldots, o_k$ of outliers are already given, and we are instead interested in characterizing each o_i. This can be accomplished by:

1. Detecting the subsets of $S \subseteq DB$ that represent

[1] For the sake of simplicity and without loss of generality, we are assuming that an arbitrary ordering of the attributes in $\mathbf{A}$ has been fixed.

a population, and such that $o_i \in S$. Intuitively, S represent a set of objects that share similar features.

2. Identifying a set $\{a_{i_1}, \ldots, a_{i_m}\} \in \mathbf{A}$ where $o_i[a_1, \ldots, a_m]$ substantially differentiates from the other objects in S.

In [3] a data population is assumed to be given, characterized by a certain number of attributes, and the information is provided that one of the individuals in the data population is abnormal. In this context, it is considered the problem of discovering sets of attributes that account for the (a-priori stated) abnormality of such an individual.

Each subset of attributes is intended to represent a *property* of individuals. A property witnesses the abnormality of an object if the combination of values the object assumes on these attributes is very infrequent with respect to the overall distribution of the attribute values in the dataset, and this is measured my means of the so called *outlierness* function. Global and local properties are introduced. Global properties are subsets of attributes explaining the given abnormality with respect to the entire data population. With local ones, instead, two subsets of attributes are singled out, where the first one justifies the abnormality within the data sub-population selected by using the values assumed by the exceptional individual on those attributes included in the second one.

The outlierness score introduced in [3] is based on measuring how much the frequency of the combination of values assumed by that object on those attributes is rare as opposite to the frequencies associated with the other combinations of values assumed on the same attributes by the other objects in the population (and, in fact, in [3] the outlierness was shown to to have some connections with the *Gini index* employed to measure the *heterogeneity* of a statistical distribution). The outlierness score presented in [3] has been specifically designed for categorical attributes and has been shown to be effective on this kind of data.

One may suggest to use that score also on numerical data by first discretizing numerical attributes. However, it must be pointed out that this kind of strategy will be unsatisfactory. First of all, the result of the analysis will strongly depend on the kind of discretization. Second, this drawback is aggravated by the peculiarities of the outlierness measure, which assigns a score close to 1 to very unbalanced distributions (as in the case of frequencies $\frac{1}{n}$ versus $\frac{n-1}{n}$), while its value rapidly decreases when frequencies spread, even in presence of rare frequencies (e.g., the score associated with the distribution of frequencies $\frac{1}{n}$, $\frac{n-1}{2n}$, $\frac{n-1}{2n}$ is about 0.5). It can be concluded that the discretization should be, in some sense, guided by the outlierness score, in order to detect in the first place the bins that would magnify the score itself.

The notion of outlierness introduced here shares a common rationale with that already proposed in [3], but aims at overcoming the aforementioned drawbacks in presence of numerical data, as accounted for in the following section.

3 Outliers and Explanations

In the following, we shall characterize populations in a "rule-based" fashion, by denoting the subset of DB that embodies them.

Formally, a *condition* on $\mathbf{A}$ is an expression of the form $a \in [l, u]$, where (i) $a \in \mathbf{A}$, (ii) $l, u \in \mathbb{D}(a)$, and (iii) $l \leq u$, if a is numeric, and $l = u$, if a is categorical. If $l = u$, the interval $I = [l, u]$ is sometimes abbreviated as u and the condition as $a \in I$ or $a = I$.

Let c be a condition $a \in [l, u]$ on $\mathbf{A}$. An object o of DB satisfies the condition c, if and only if $o[a]$ equals l, if a is categorical, or $l \leq o[a] \leq u$, if a is numerical. Moreover, o satisfies a set of conditions C if and only if o satisfies each condition $c \in C$. Given a set C of conditions on $\mathbf{A}$. The *selection* DB_C of the database DB w.r.t. C is the database consisting of the objects $o \in DB$ satisfying C.

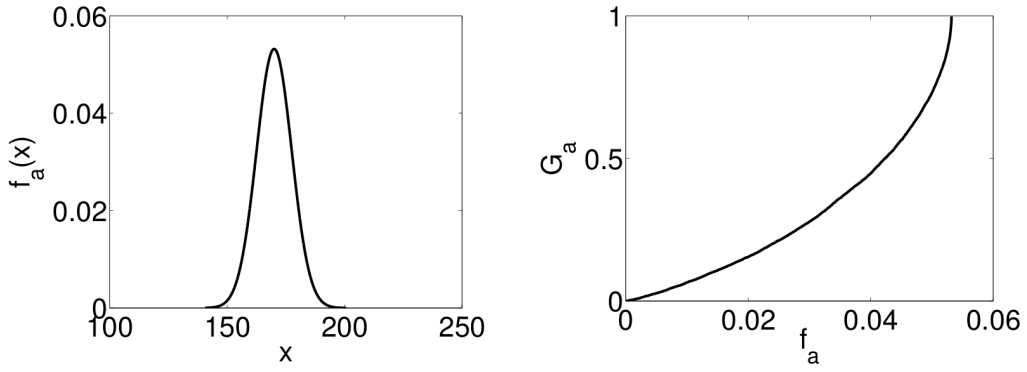
Next, the definition of outlierness (Section 3.1) and of explanation (Section 3.2) are introduced.

3.1 Outlierness. We introduce now the notion of *outlierness*, a measure used to quantify the exceptionality of a property. The intuition underlying this measure is that an attribute makes an object exceptional if the relative likelihood of the value assumed by that object on the attribute is rare if compared to the relative likelihood associated with the other values assumed on the same attribute by the other objects of the database.

Let a be an attribute of A. We assume that a random variable X_a is associated with the attribute a, which models the domain of a. Then, with $f_a(x)$ we denote the pdf associated with X_a. Let X_a^f denote the random variable whose pdf represents the relative likelihood for the pdf f_a to assume a certain value. The cdf G_a of X_a^f is:

$$G_a(f) = \int_0^f Pr(X_a^f \leq f)\ \mathrm{d}f. \tag{3.1}$$

Example 1. Assume that the height of the individuals of a population is normally distributed with mean $\mu = 170cm$ and standard deviation $\sigma = 7.5cm$. Then, let a be the attribute representing the height, X_a is a



(a) (b)

Figure 1: Example of function $G_a(\cdot)$.

random variable following the same distribution of the domain and $f_a(x)$ is the associated pdf, reported in Figure 1(a). The pdf $f_a(x)$ assumes value in the domain $[0, f_a(\mu) = 0.0532] \subset \mathbb{R}$. Consider, now, the random variable X_a^f. The cdf $G_a(v)$ associated with X_a^f denotes the probability for f_a to assume value less than or equal to v. Then, $G_a(v) = 0$ for each $v \leq 0$ and $G_a(v) = 1$ for each $v \geq 0.0532$. To compute the value of $G_a(v)$ for a generic v, the integral reported in Equation (3.1) has to be evaluated. The resulting function is reported in the Figure 1(b).

□

The *outlierness* $\text{out}_a(o, DB)$ (or, simply, $\text{out}_a(o)$) of the attribute a in o w.r.t. DB is defined as follows:

$$\text{out}_a(o) = \Omega\left(\int_{f_a(o[a])}^{+\infty} (1 - G_a(f))\ \mathrm{d}f + \right. \\ \left. - \int_0^{f_a(o[a])} G_a(f)\ \mathrm{d}f\right). \tag{3.2}$$

where Ω denotes a suitable function mapping from $\mathbb{R}$ to $[0, 1]$ such that (i) $\Omega(x) = 0$ for $x < 0$, and (ii) Ω is monotone increasing for $x \geq 0$. In the following we employ the mapping

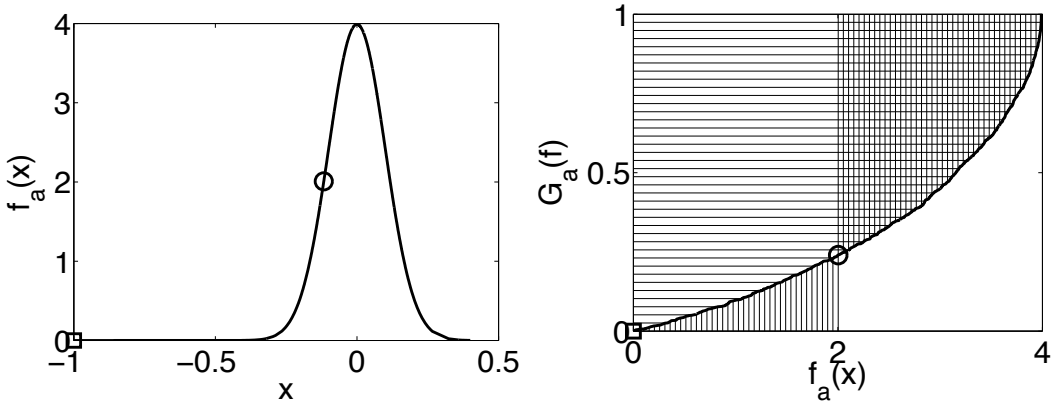
$$\Omega(x) = \frac{1 - \exp(-x)}{1 + \exp(-x)}.$$

The first integral measures the *area above* the cdf $G_a(f)$ for $f > f_a(o[a])$, while the second integral measures the *area below* the cdf G_a for $f \leq f_a(o[a])$. Intuitively, the larger the first term, the larger the degree of unbalanceness between the occurrence probability of $o[a]$ and that of the values that are more probable than $o[a]$. As for the second term, the smaller it is, the more likely the value $o[a]$ to be rare. Thus, the outlierness value ranges within $[0, 1]$ and in particular it is close to zero for usual properties; By contrast, values closer to one denote exceptional properties.

Example 2. Consider fig. 2, reporting on the left a Gaussian distribution $f_a(x)$ (with mean $\mu = 0$ and standard deviation $\sigma = 0.1$). Consider the values $v_1 = -1$ and $v_2 = -0.12$, for which $f_a(v_1) \approx 0$ and $f_a(v_2) \approx 2$ hold. Assume that an outlier object o exhibits value v_1 on a. The associated outlierness $\text{out}_a(o)$ corresponds to the whole area (filled with horizontal lines) above the cdf curve, that is $\Omega(3.06) = 0.91$. For an object o' exhibiting value v_2 on a, instead, the associated outlierness corresponds to the difference between two areas (filled with vertical lines) detected at frequency 2, that is $\Omega(1.17 - 0.10) = 0.49$. □

For the sake of clarity, in the above example we considered a pdf having a simple form. However, we wish to point out that our measure is able to correctly recognize exceptional properties irrespectively of the form of the underlying pdf, since it compares the occurrence probabilities of the domain values rather than directly comparing the original domain values.

3.2 Explanations. *Explanations* are useful in our framework to provide a justification of the anomalous value characterizing an outlier. Intuitively, a attribute $a \in \mathbf{A}$ of o that behaves normally with respect to the database as a whole, may be unexpected when the attention is restricted to a portion of the database. We shall call this anomalous attribute a *property* of o. Relevant subsets of the database upon which to investigate outliernes can be hence obtained by selecting



(a) (b)

Figure 2: Example of outlierness measure.

the database objects satisfying a condition, and such that a property is exceptional for o.

A condition c (set of conditions C, resp.) is, intuitively, an *explanation* of the property a. if $o \in DB_c$ ($o \in DB_C$, resp.) and a is exceptional for o w.r.t. DB_c (DB_C, resp.) (i.e., the value $\text{out}_a(o, DB_C)$ is close to 1). Finally, the *outlierness* of the set property a in o w.r.t. DB with *explanation* C is defined as $\text{out}_a^C(o, DB) = \text{out}_a(o, DB_C)$.

It is worth noticing that, according to the relative size of DB_C, not all the explanations should be considered equally relevant. In the following, we concentrate on σ-explanations, i.e., conditions C such that $\frac{|DB_C|}{DB} \geq \sigma$, where $\sigma \in [0, 1]$ is a user-defined parameter.

Thus, given an object o of a database DB on a set of attributes $\mathbf{A}$, and parameters $\sigma_\theta \in [0, 1]$ and $\Omega_\theta \in [0, 1]$, the problem of interest here is: *Find the pairs* (E, p), *with* $E \subseteq \mathbf{A}$ *and* $p \in \mathbf{A} \setminus E$, *such that* E *is a* σ_θ*-explanation and* $\text{out}_p^E(o, DB) \geq \Omega_\theta$. Such an attribute p is also called an *outlying property*.

4 Detecting Outlying Properties

In order to detect outlying properties and their explanations, we need to solve two basic problems: (1) computing the outlierness of a certain multiset of values and (2) determining the conditions to be employed to form explanations. The strategies we have designed to solve these two problems exploit a common framework, which is based on Kernel Density Estimation (KDE). Specifically, given a numerical attribute a, in order to estimate the pdf f_a we exploit *generalized kernel density estimation* [17], according to which the estimated density at point $x \in \mathbb{D}(a)$ is

$$\hat{f}_{\mathbf{m},\mathbf{w},\mathbf{b}}(x) = \left(\sum_{i=1}^{k} w_i\right)^{-1} \sum_{i=1}^{k} \frac{w_i}{b_i} K\left(\frac{x - m_i}{b_i}\right), \tag{4.3}$$

Here, K is a kernel function, and $\mathbf{m} = (m_1, \ldots, m_k)$, $\mathbf{w} = (w_1, \ldots, w_k)$ and $\mathbf{b} = (b_1, \ldots, b_k)$ are k-dimensional vectors denoting the *kernel location*, *weight*, and *bandwidth*, respectively. The above mentioned strategies are detailed next, together with the method for mining outlying properties.

Function *EstimatePDF*($\mathbf{x}$)

Input: $\mathbf{x} = x_1, \ldots, x_n$
Output: $\hat{\mathbf{f}} = \hat{f}_1, \ldots, \hat{f}_n$

Set h to $1.06 \cdot \text{std}(\mathbf{x}) \cdot n^{-1/5}$ `// Rule of thumb`
Set $\boldsymbol{\beta}$ to $(1, \ldots, 1)$;
for $t = 1$ **to** 5 **do**
  $\hat{\mathbf{f}} = \mathit{ComputePDF}(\mathbf{x}, h, \mathbf{w})$;
  $f_m = (\prod_{i=1}^{n} \hat{f}_i)^{1/n}$;
  for $i = 1$ **to** n **do**
    Set β_i to $(f_m / \hat{f}_i)^{1/2}$;
return $(\hat{f}_1, \ldots, \hat{f}_n)$;

4.1 Outlierness computation. In order to compute the outlierness, we specialize formula in Equation (4.3) by setting $\mathbf{m} = (x_1, \ldots, x_n)$ and $\mathbf{w} = \mathbf{1}$, thus obtaining

$$\hat{f}_a(x) = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{b_i} K\left(\frac{x - m_i}{b_i}\right), \tag{4.4}$$

Function *ComputePDF*$(\mathbf{x}, h, \mathbf{w})$

Input: $\mathbf{x} = x_1, \ldots, x_n$: a set of values
h : a bandwidth
$\mathbf{w} = w_1, \ldots, w_n$: a set of weights
Output: $\hat{\mathbf{f}} = \hat{f}_1, \ldots, \hat{f}_n$: the density estimate at points $\mathbf{x}$

1 Sort the sequence $L = x_1^l, \ldots, x_n^l$, according to the values $\{x_i - \frac{w_i h}{2} : 1 \leq i \leq n\}$, and record the associated indexes $l_1, \ldots, l_n$;
2 Sort the sequence $U = x_1^u, \ldots, x_n^u$, according to the values $\{x_i + \frac{w_i h}{2} : 1 \leq i \leq n\}$, and record the associated indexes $u_1, \ldots, u_n$;
3 **for** $i = 1$ **to** n **do**
4 Find the last element $x_{l^*}^l$ of L not greater than x_i;
5 Find the first element $x_{u^*}^u$ of U not smaller than x_i;
6 Set J to $\{l_1, l_2, \ldots, l^*\} \cap \{u^*, \ldots, u_{n-1}, u_n\}$;
7 Set $\hat{f}_i$ to $\frac{1}{nh} \sum_{j \in J} \frac{1}{w_j}$;
8 **return** $(\hat{f}_1, \ldots, \hat{f}_n)$;

where $x_1, \ldots, x_n$ are the values in $\{y[a] : y \in DB\}$, each term b_i is equal to $h\beta_i$, with h a global bandwidth and $\prod_{i=1}^n \beta_i = 1$. The rationale underlying this choice is that we want that each value at hand ($\mathbf{m} = \mathbf{x}$) contributes in equal manner ($\mathbf{w} = \mathbf{1}$) to the estimation of the underlying pdf. Moreover, we employ the *Parzen window* kernel function, that is $K(x) = 1$, for $|x| \leq 1/2$, and $K(x) = 0$ otherwise, since this kernel represents a good trade off between simplicity of computation and accuracy. Indeed, we are able to provide a parameter-free function that computes an accurate estimate $\hat{f}_a$ of the pdf f_a in time $O(n \log n)$. We also notice that, since the outlierness depends on the cdf of the pdf values, this greatly mitigates the impact of the non-smoothness of the estimate of the pdf through Parzen windows, other than making the measure robust w.r.t. deviations of the estimate from the real distribution.

Let $\mathbf{x}$ denote the vector $(x_1, \ldots, x_n)$, and $\boldsymbol{\beta}$ denote the vector $(\beta_1, \ldots, \beta_n)$. The function *ComputePDF* computes the vector $\hat{\mathbf{f}}$, whose generic element $\hat{f}_i$ represents the value of density $\hat{f}_a(x_i)$ at point x_i, as computed by exploiting Equation (4.4). In particular, when the Parzen window is employed, the computation of $\hat{f}_a(x)$ reduces to determine the value $\frac{1}{nh} \sum_{j \in J} \frac{1}{\beta_j}$, where J is the set containing the indexes j of the elements x_j of $\mathbf{x}$ such that $\left|\frac{x - x_j}{\beta_j h}\right| \leq \frac{1}{2}$ or, in other words, such that $x_j - \frac{\beta_j h}{2} \leq x$ and $x \leq x_j + \frac{\beta_j h}{2}$. The set J associated with a specific value x, can be determined by performing two binary searches and one intersection, as shown in the pseudo-code. Since this computation is executed n times, this leads to an overall cost $O(n \log n)$.

The function *EstimatePDF* is in charge of computing the right values for the parameters h and $\boldsymbol{\beta}$. It exploits the algorithm for calculating a variable bandwidth KDE [25]. The method starts with a density estimate by using a fixed-bandwidth kernel, with h determined by means of a rule of thumb [27] (see Function *EstimatePDF*, line 1) and $\boldsymbol{\beta} = \mathbf{1}$. Then, the bandwidths β_i are updated to a value which is inversely related to the density estimate. It was observed [12] that iterations produce little changes: hence, we execute it a fixed number of times in order to. keep the computational cost to $O(n \log n)$.

The function *ComputeOutlierness* exploits *EstimatePDF* to compute the numerical estimate $\hat{\mathbf{f}}$ of the pdf f_a. Then, it computes the distribution function G_a (see Equation (3.1)) by setting G_i to $|\{f_j \leq \tilde{f}_i : 1 \leq j \leq n\}|/n$, that is i/n, and, finally, the outlierness value *out* (see Equation (3.2)), by performing a numerical integration, which costs $O(n)$. Thus, the dominating operations of *ComputeOutlierness* are the call to the function *EstimatePDF* and the sort of the elements of $\hat{\mathbf{f}}$, with a resulting overall cost $O(n \log n)$.

4.2 Condition building. Proper conditions are the basic building blocks for the explanations. To single them out, our strategy consists in finding, for each attribute a, the "natural" interval I_a including $o[a]$, namely, an interval of homogeneous values on a. The rationale underlying this choice is to avoid the risk of overfitting: a guided search for a proper condition can easily yield an ad-hoc fragment of the data where the outlierness measure is "artificially" maximized. On the other side, proper conditions which encode the genuine intervals for each attribute domain can relevantly impact on the detection of significant outlier explanations.

The search for feasible intervals still relies on adopt-

Function *ComputeOutlierness*(o, a, DB)

Input: o : an outlier object
a : a dataset attribute
DB : a dataset
Output: *out* : the outlierness of the attribute a in o w.r.t. DB

Set $\mathbf{x}$ to $DB[a]$;
Set $\hat{\mathbf{f}}$ to *EstimatePDF*$(\mathbf{x})$;
Determine the sequence $\tilde{f}_1, \ldots, \tilde{f}_n$, by sorting the elements of the set $\{\hat{f}_i : 1 \leq i \leq n\}$;
for $i = 1$ **to** n **do**
  Set G_i to $|\{f_j \leq \tilde{f}_i : 1 \leq j \leq n\}|/n = i/n$;
Let i^* be such that $\tilde{f}_{i^*}$ is the value in $\hat{\mathbf{f}}$ associated with $o[a]$;
Set *out* to 0;
for $i = i^* + 1$ **to** n **do**
  Set $out = out + (\tilde{f}_i - \tilde{f}_{i-1})(2 - G_i - G_{i-1})/2$;
for $i = 2$ **to** i^* **do**
  Set $out = out - (\tilde{f}_i - \tilde{f}_{i-1})(G_i + G_{i-1})/2$;
return $\Omega(out)$;

ing the kernel density family introduced so far. In practice, for each attribute a, we estimate f_a by means of $\hat{f}_{\mathbf{m},\mathbf{w},\mathbf{b}}$. This latter function can be interpreted as a mixture density over the parameter sets $\mathbf{m}, \mathbf{w}, \mathbf{b}$. Also, the adoption of a Gaussian kernel

$$K(x) = \phi(x) = (2\pi)^{-1/2} \exp(x^2/2)$$

allows the estimation of the parameter set via a standard EM-based maximum likelihood approach. In particular, the approach will partition the values $\{y[a] : y \in DB\}$ into a set of j^* disjoint intervals $I_a^1, \ldots, I_a^{j^*}$ and, then, the interval $I_a^{j_o}$ which $o[a]$ belongs to will be selected as the proper condition I_a for the attribute a.

The resulting iterative scheme draws from [17], and updates locations and bandwidths according to the following equations:

$$m_j = \frac{1}{\sum_i \gamma_{ij}} \sum_{i=1}^{n} x_i \gamma_{ij}, \tag{4.5}$$

$$b_j^2 = \frac{1}{\sum_i \gamma_{ij}} \sum_{i=1}^{n} \gamma_{ij} (x_i - m_j)^2. \tag{4.6}$$

Here, γ_{ij} represents the mixing probability that value i is associated with the j-th interval and, in its turn, is computed at each iteration as:

$$\gamma_{ij} = \frac{w_j \phi_{b_j}(x_i - m_j)}{\hat{f}_{\mathbf{m},\mathbf{w},\mathbf{b}}(x_i)} \tag{4.7}$$

We also adapt the annihilation procedure proposed in [11], which allows for an automatic estimation of the optimal number j^* of intervals, as well as to ignore the initialization issues. The estimation of the parameters is accomplished iteratively for each interval I_a^j, where each weight is computed as

$$w_j = \frac{\max\{0, \sum_{i=1}^{n} \gamma_{ij} - \frac{n}{2}\}}{\sum_{j=1}^{j^*} \max\{0, \sum_{i=1}^{n} \gamma_{ij} - \frac{n}{2}\}} \tag{4.8}$$

Whenever a weight equals to 0, the contribution of its component annihilates in the density estimation. As a consequence, the iterative procedure can start with a high initial value j^*, and the initialization of each mixing probability can be done randomly without compromising the final result. Function *ComputeInterval* reports the overall scheme. We also call the interval reported by this function, the *natural interval* of a in o w.r.t. DB.

4.3 The mining method. Putting things together, in order to search for outlying properties, we employ the following strategy. Given a dataset DB on the set of attributes $\mathbf{A} = \{a_1, \ldots, a_m\}$, an outlier object o, parameters $\sigma_\theta \in [0, 1]$, $\Omega_\theta \in [0, 1]$, and positive integer $k_\theta \leq m$ (representing an upper bound to the size of an acceptable explanation):

1. For each attribute $a_i \in \mathbf{A}$, the interval I_{a_i} and, hence, the associated condition $a_i \in I_{a_i}$, is determined by means of the function *ComputeInterval*;
2. Given the set of conditions $S = \{a_1 \in I_{a_1}, \ldots, a_m \in I_{a_m}\}$ on the m attributes in $\mathbf{A}$, we exhaustively enumerate all the pairs (E, p), with $E \subseteq I$, $|E| \leq$

Function *ComputeInterval*(o, a, DB)

Input: o : an outlier object
a : a dataset attribute
DB : a dataset
Output: *out* : the natural interval of the attribute a in o w.r.t. DB

```
1  set x to DB[a];
2  set j* to √n;
3  initialize γ_ij randomly, ∀i ∈ [1..n] and ∀j ∈ [1..j*];
4  repeat
5      for j = 1 to j* do
6          update w_j // Equation (4.8)
7          if w_j > 0 then
8              update m_j, b_j // Equation (4.6)
9              update γ_ij, ∀i ∈ [1..n] // Equation (4.7)
10         else
11             eliminate the jth component;
12             set j* to j* − 1;
13 until increase in likelihood is negligible;
14 assign x_i to the interval I_a^{j_i} s.t. j_i = arg max_j γ_ij;
15 let I_a^{j_o} be the interval which o[a] belongs to;
16 set l_a to min_i{x_i | x_i ∈ I_a^{j_o}};
17 set u_a to max_i{x_i | x_i ∈ I_a^{j_o}};
18 return [l_a, u_a]
```

k_θ, and $p \in \mathbf{A} \setminus E$ and maintain in the set $\mathcal{OP}$ those pairs such that

(a) E is a σ_θ-explanation, and

(b) the outlierness $\text{out}_p^E(o, DB)$ is greater than Ω_θ; the outlierness is measured by means of the function *ComputeOutlierness*;

3. The set $\mathcal{OP}$ is returned.

As for the cost of the above procedure, the first step is basically depends on the rate of convergence of the EM algorithm. The basic iteration (see lines 4-12 of function *ComputeInterval*) is $O(n^{3/2})$. Notice, however, that interval components annihilate early in the first iterations, so practically we can assume that the number of intervals j^* is bounded to a constant value. Thus, the overall complexity of the first step is linear in the size of the data and the number of iterations. Clearly, the rate of convergence of the algorithm is of practical interest, and it is usually slower than the quadratic convergence typically available with Newton-type methods. [10] shows that the rate of convergence of the EM algorithm is linear and the it depends on the proportion of information in the observed data.

As far as the second step is concerned, point (a) costs at most nk_θ, while we have already seen that point (b) costs $O(n \log n)$. Since these two sub-steps are executed at most $O(m^{k_\theta})$ times, the overall cost of step 2 is $O(m^{k_\theta} n \log n)$.

As for the parameter k_θ, it is needed in order to bound the size of an acceptable explanation. As a matter of fact, allowing more than a few conditions will lead to unintelligible explanations. Hence, in the experimental results section we will set it to the value $k_\theta = 3$.

5 Experimental results

In this section, experimental results conducted by employing the proposed methodology are described. Specifically, first, Section 5.1 evaluates the technique on some datasets from the UCI Machine Learning repository and, then, Section 5.2 describes a specific real-life case study where the technique was profitably exploited.

In both cases, The ground truth is represented by outlier tuples, detected by resorting to the feature bagging algorithm described in [21]. Briefly, the technique detects outliers by iteratively running a base outlier detection algorithm on a subset of the available attributes. Outlier detected in the various runs are then scored by adopting a *combine* function which assigns a score to each outlier.

The bagging technique was instantiated by explot-

ing the base OD method described in [2], where the parameters are set to produce just a single outlier. Further, the *combine* technique adopted simply scores outliers on the basis of the positive responses they get within the iterations.[2]

Notice that the feature bagging technique boosts the robustness of base outlier detection techniques. at the same time, it makes quite difficult to manually infer (e.g., by means of visualization techniques) the reasons why a specific tuple was deceted as an outlier. In fact, a tuple can be reputed an outlier for a combination of factors which in turn depend on different subsets of the attributes. As a consequence, the analysis of the outliers produced with such a technique provides a significant benchmark on the effectiveness of the outlier explanation technique.

5.1 Evaluation and execution time. We employ three real datasets from the UCI Machine Learning repository [13]. The first and the second databases, called *Ecoli* (with 336 instances and 7 attributes) and *Yeast* (with 1,484 instances and 8 attributes) respectively, contain information about protein localization sites. The third database, called *Cloud*, contains information about cloud cover and includes 1,024 instances with 10 attributes.

The support threshold σ_θ has been set to 0.2 and the maximum number k_θ of conditions in the explanation to 3. The following table reports the explanation-property pairs scoring the maximum value of outlierness.

DB	o	$\mathbf{out_p^E(o)}$	p	E
Ecoli	223	1.000	a_4	$\emptyset$
Yeast	990	0.997	a_3	$\{\ a_2 \in [0.13, 0.38]\ \}$
Cloud	354	1.000	a_6	$\{\ a_1 \in [1.0, 6.7]$, $a_2 \in [134.9, 255.0]$, $a_5 \in [2{,}450.5, 3{,}211.5]\ \}$

In the third column, we report the outlierness value, in the fourth column the attribute associated with the property, and in the fifth column the explanation. Figure 3 reports the functions $G_a(f)$ associated with the objects considered in the experiments.

Figure 3 at the top left reports the area associated with the property a_4 and empty explanation for the object 223 in the *Ecoli* database. The property a_4 is the attribute *Presence of charge on N-terminus of predicted lipoproteins*. The object 223 is the only object assuming value 0.5 on this attribute, while all the other objects assume value 1.0. As a consequence, this attribute is a clear outlying property with respect to the whole database and, in fact, the associated explanation is empty.

Figure 3 at the top right reports the area associated with the property a_3 for the object 990 in the *Yeast* database. The attribute a_3 is *Score of the ALOM membrane spanning region prediction program*. The solid line represents the curve $G_{a_3}(f)$ obtained when the explanation $\{a_2\}$ is taken into account, while the dashed line represents the curve $G_{a_3}(f)$ obtained for the empty explanation. We note that, by taking the explanation into account, an improvement of the outlierness value is achieved, even if the property a_3 is quite interesting also with respect to the whole database.

Finally, Figure 3 at the bottom left reports the area associated with the property a_6 and the explanation $\{a_1, a_2, a_5\}$ for the object 354 in the *Cloud* database. The attribute a_6 is the *Visible entropy*, while the explanation attributes are *Visible mean*, *Visible max* and *Contrast*. Figure 3 on the bottom right reports the area associated with the same property, but for the empty explanation. In this case, it is worth noting that the property a_6 is not exceptional with respect to the whole database (the outlierness value is approximatively 0.3) but it becomes very exceptional with respect to the subpopulation selected by the explanation.

The following table reports the execution times associated with the experiments.[3]

DB	Condition Building	Total Outlier Computation	Mean Outlier Computation
Ecoli	6.39 *sec*	16.76 *sec*	0.04 *sec*
Yeast	54.38 *sec*	138.51 *sec*	0.19 *sec*
Cloud	702.67 *sec*	91.08 *sec*	0.05 *sec*

The second column shows the total time required to compute the proper conditions, the third one the time required to compute the outlierness of all the explanation-property pairs, and the fourth the corresponding mean time.

5.2 A case Study. We tested the above methodology on a real-life dataset about doctors and their associated medical prescriptions. In the scenario under considerations, each doctor is associated with a group of patients, and can prescribe drugs to people belonging to that group. There are several respects in which the detection of anomalous prescriptions can be of interest in this scenario: from fraud detection (doctors prescribing

[2] In practice, if a tuple is detected as an outlier in a given iteration, it gets a positive score. Scores are then summarized in the combine function, and tuples are sorted according to the scores.

[3] Experiments have been performed on a Intel Xeon 2.33GHz based computer.

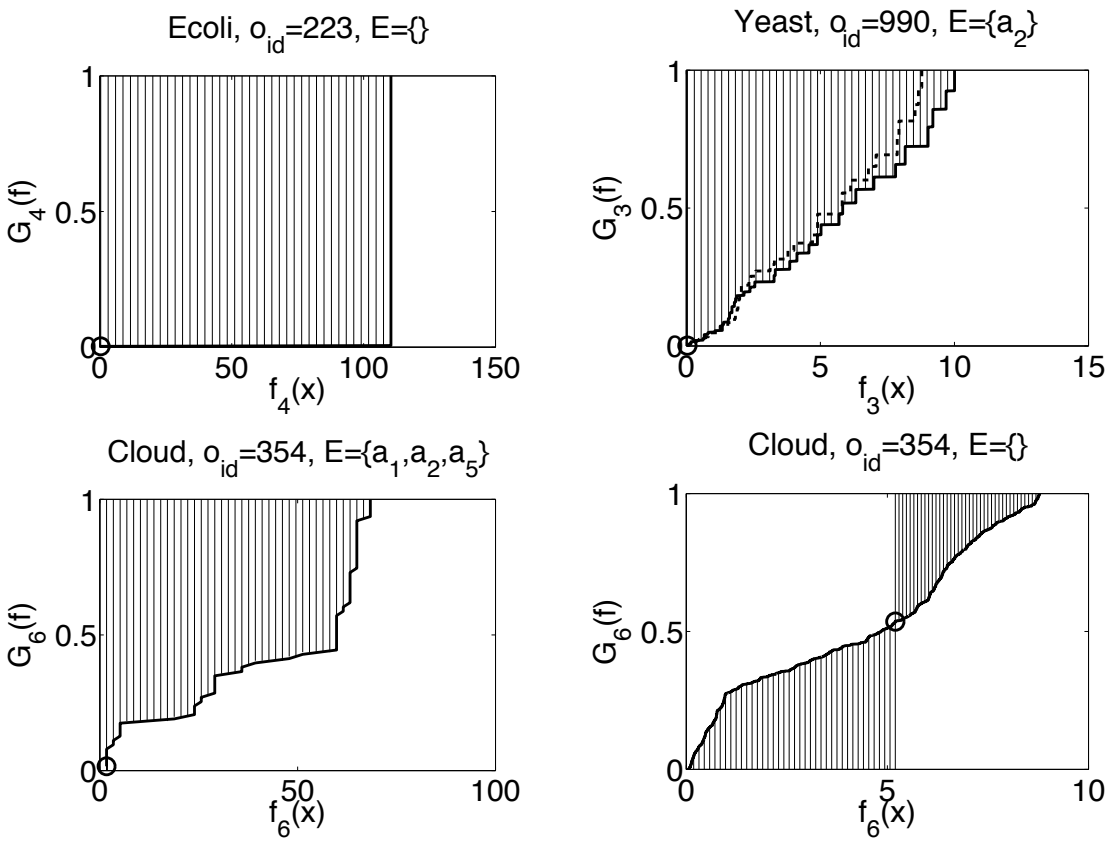


Figure 3: Experimental results on the *Ecoli*, *Yeast*, and *Cloud* datasets.

more than expected, e.g., with regards to a specific pharmaceutical company) to the diagnosis unknown health issues. The specific goal here is to find doctors whose behaviour is different than expected. Outlier explanation plays a crucial role here, since we are interested in knowing both the reference population of doctors with similar prescribing behavior, and the reason why a doctor is considered anomalous in that population. For example, a doctor can be considered anomalous because its number of prescriptions for a given drug is significantly higher than average or he/she he is prone to prescribe drugs from a particular company.

The data we analyze contains information about three different entities:

- *doctors*: demographic information, along with with information about its patients;
- *drugs*: the active element and the pharmaceutical company that produces the drug
- *prescriptions*: this is the facts table containing information about prescriptions made by doctors to their patients

The resulting table contains 2020 tuples, where each tuple represents the number of prescriptions that a specific doctor made on 106 drugs. To better model patients' influence on prescriptions, prescriptions were weighted according to their age and sex. In practice, tuples are normalized in order to make fair comparisons among doctors exhibiting different classes of patients.

By analyzing the data with the aforementioned algorithm we found 5 top outliers exhibiting a significant outlierness score. Two of these outliers are particularly interesting to analyze with the explanation techniques, namely tuple 34 and 651.

In particular, the outlierns of tuple 34 is characterized by attributes a_{26}, a_{102} and a_{103}. Within the population detected by the intervals for the attributes a_{102} and a_{103}, the tuple however exhibits a significantly low value for a_{26}. This is clearly shown in Figure 4.

A different behavior is instead exhibited by tuple 651, characterized by attributes a_1, a_2, a_5, a_6. selecting the population by means of attributes a_1, a_2, a_5 and studying the distribution for attribute a_6 in this population, we can notice that the value exhibited by tuple

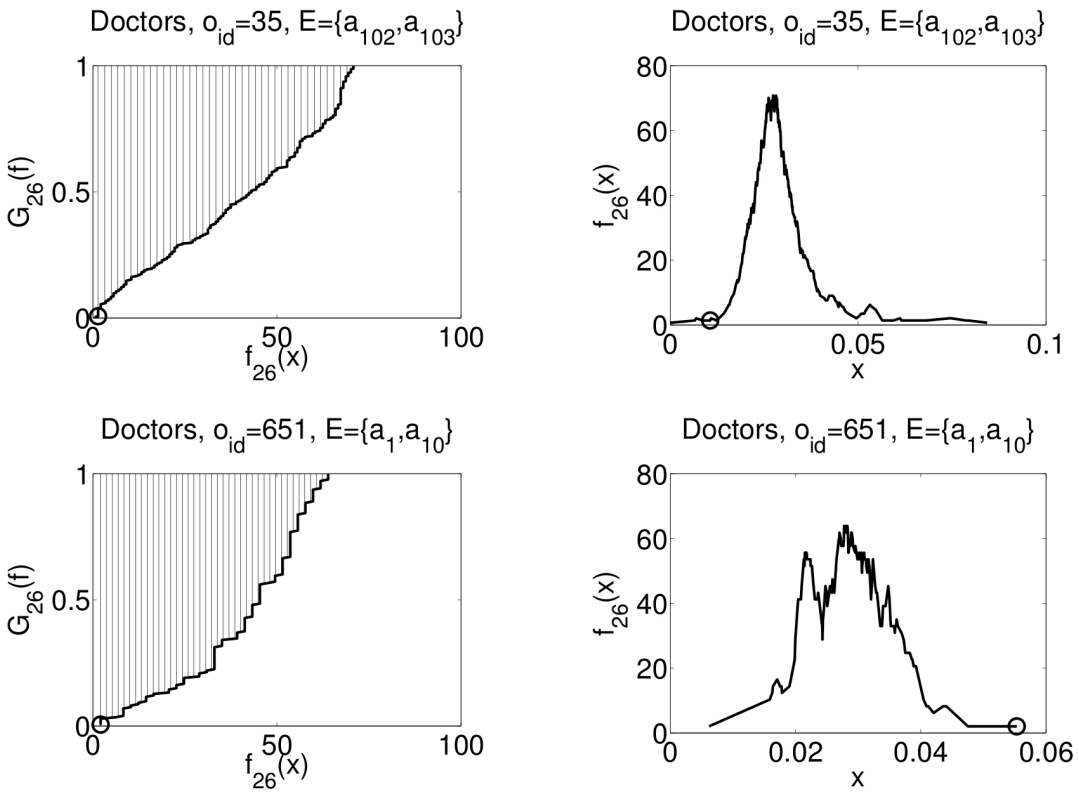


Figure 4: Outlier explanations in the `Doctors` Dataset, for tuples a_{34} and a_{651}.

651 is at the upper extreme. Again, this represents a deviation from the normal behavior in that population, as shown in the leftmost graph of Figure 4.

6 Conclusions and Future Work

The purpose of this paper has been that of devising techniques by which the outlying properties detection problem can be solved in the presence of both categorical and numerical attributes, which represents a step forward with respect to available literature. The core of our approach has been the definition of a sensible outlierness measure, representing a refined generalization of that proposed in [3], which is able to quantify the exceptionality of a given property featured by the given input anomalous object with respect to a reference data population. Also, we have developed algorithms to detect properties characterizing the anomalous object provided in input. The experimental results we have obtained confirm that the presented approach is more than promising.

As a matter of fact, there are several application scenarios where the proposed technique can be profitably applied. In the *doctors* scenario, for example, it can be used to find explanations for anomalous or frauding behavior. Further scenarios include rank learning problems like in [9]: there, we investigate the problem of detecting rules for characterizing individuals who are scored as exceptional according to a specific scoring function (like, e.g., the amount of fraud they commit in a fraud detection scenario). It is clear that if exceptional objects are reputed as outliers, then the outlier explanation technique described in this paper is a basic building block for rule learning in that domain.

Also, it is worth highlighting the importance of dealing with numerical attributes other than categorical one in outlier explanation, especially from an application viewpoint. In the aforementioned scenarios, for example, data express basically measurements on empirical situations, and the underlying data is made of several numerical attributes describing such measurements.

As future work, we are interested in designing an efficient mining algorithm exploiting suitable pruning rules, in exploring other strategies for generating proper conditions, and in performing a more extensive experimental campaign.

References

[1] C. C. Aggarwal and P.S. Yu. Outlier detection for high dimensional data. In *Proc. of the International Conference on Managment of Data (SIGMOD)*, pages 37–46, 2001.

[2] F. Angiulli and F. Fassetti. Dolphin: an efficient algorithm for mining distance-based outliers in very large datasets. *ACM Transactions on Knowledge Discovery from Data*, 3(1):Article 4, 2009.

[3] F. Angiulli, F. Fassetti, and L. Palopoli. Detecting outlying properties of exceptional objects. *ACM Transactions on Database Systems*, 34(1):Article 7, 2009.

[4] A. Arning, C. Aggarwal, and P. Raghavan. A linear method for deviation detection in large databases. In *Proc. ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 164–169, Portland, OR, USA, 1996.

[5] V. Barnett and T. Lewis. *Outliers in Statistical Data*. John Wiley & Sons, 1994.

[6] M. M. Breunig, H. Kriegel, R.T. Ng, and J. Sander. Lof: Identifying density-based local outliers. In *Proc. International Conference on Managment of Data (SIGMOD)*, pages 93–104, Dallas, TX, USA, 2000.

[7] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. *ACM Computing Surveys*, 41(3), 2009.

[8] N. V. Chawla, N. Japkowicz, and A. Kotcz. Editorial: special issue on learning from imbalanced data sets. *SIGKDD Explorations*, 6(1):1–6, 2004.

[9] Gianni Costa, Fabio Fassetti, Massimo Guarascio, Giuseppe Manco, and Riccardo Ortale. Mining models of exceptional objects through rule learning. In *SAC*, pages 1078–1082, 2010.

[10] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, B*, 39, 1977.

[11] M. A. T. Figueiredo and A. K. Jain. Unsupervised learning of finite mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24:381–396, 2002.

[12] J. Fox. Describing univariate distributions. In J. Fox and J. S. Long, editors, *Modern Methods of Data Analysis*, pages 58–125. CA: Sage Publications, 1990.

[13] A. Frank and A. Asuncion. Uci machine learning repository [`archive.ics.uci.edu/ml`], 2010.

[14] D.M. Hawkins. *Identification of Outliers*. Monographs on Applied Probability and Statistics. Chapman & Hall, May 1980.

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

[16] W. Jin, A. K. H. Tung, and J. Han. Mining top-n local outliers in large databases. In *Proc. of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 293–298, San Francisco, CA, USA, 2001.

[17] M. C. Jones and D. A. Henderson. Maximum likelihood kernel density estimation: On the potential of convolution sieves. *Computational Statistics & Data Analysis*, 53:3726–3733, 2009.

[18] E. Knorr and R. Ng. Algorithms for mining distance-based outliers in large datasets. In *Proc. of the International Conference on Very Large Databases (VLDB)*, pages 392–403, New York, NY, USA, 1998.

[19] H.-P. Kriegel, P. Kröger, E. Schubert, and A. Zimek. Loop: local outlier probabilities. In *Proc. of the ACM International Conference on Information and Knowledge Management (CIKM)*, pages 1649–1652, Hong Kong, China, 2009.

[20] H.-P. Kriegel, M. Schubert, and A. Zimek. Angle-based outlier detection in high-dimensional data. In *Proc. of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 444–452, Las Vegas, USA, 2008.

[21] Aleksandar Lazarevic and Vipin Kumar. Feature bagging for outlier detection. In *Proc. of ACM SIGKDD Conf (KDD'05)*, pages 157–166.

[22] F.T. Liu, K.M. Ting, and Z.-H. Zhou. Isolation forest. In *Proc. of the IEEE International Conference on Data Mining (ICDM)*, pages 413–422, Pisa, Italy, 2008.

[23] T.M. Mitchell. *Machine Learning*. Mac Graw Hill, 1997.

[24] S. Papadimitriou, H. Kitagawa, P.B. Gibbons, and C. Faloutsos. Loci: Fast outlier detection using the local correlation integral. In *Proc. of the International Conference on Data Enginnering (ICDE)*, pages 315–326, Bangalore, India, 2003.

[25] I. H. Salgado-Ugarte and M. A. Pérez-Hernández. Exploring the use of variable bandwidth kernel density estimators. *Stata Journal*, 3(2):133–147, 2003.

[26] B. Schölkopf, C. Burges, and V. Vapnik. Extracting support data for a given task. In *Proc. of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 252–257, Montreal, Canada, 1995.

[27] B. W. Silverman. *Density Estimation for Statistics and Data Analysis*. Chapman and Hall, 1986.

[28] P.-N. Tan, M. Steinbach, and V. Kumar. *Introduction to Data Mining*. Addison-Wesley Longman, 2005.