Data Min Knowl Disc (2014) 28:190–237
DOI 10.1007/s10618-012-0300-z

Local outlier detection reconsidered: a generalized view on locality with applications to spatial, video, and network outlier detection

Erich Schubert · Arthur Zimek · Hans-Peter Kriegel

Received: 5 December 2011 / Accepted: 27 November 2012 / Published online: 18 December 2012

Abstract Outlier detection research has been seeing many new algorithms every year that often appear to be only slightly different from existing methods along with some experiments that show them to "clearly outperform" the others. However, few approaches come along with a clear analysis of existing methods and a solid theoretical differentiation. Here, we provide a formalized method of analysis to allow for a theoretical comparison and generalization of many existing methods. Our unified view improves understanding of the shared properties and of the differences of outlier detection models. By abstracting the notion of locality from the classic distance-based notion, our framework facilitates the construction of abstract methods for many special data types that are usually handled with specialized algorithms. In particular, spatial neighborhood can be seen as a special case of locality. Here we therefore compare and generalize approaches to spatial outlier detection in a detailed manner. We also discuss temporal data like video streams, or graph data such as community networks. Since we reproduce results of specialized approaches with our general framework, and even improve upon them, our framework provides reasonable baselines to evaluate the true merits of specialized approaches. At the same time, seeing spatial outlier detection as a special case of local outlier detection, opens up new potentials for analysis and advancement of methods.

Responsible editor: M. J. Zaki.

E. Schubert · H.-P. Kriegel
Ludwig-Maximilians-Universität München, Oettingenstr. 67, 80538 Munich, Germany
e-mail: schube@dbs.ifi.lmu.de

H.-P. Kriegel
e-mail: kriegel@dbs.ifi.lmu.de

A. Zimek (✉)
Department of Computing Science, University of Alberta, Edmonton, AB, T6G 2E8, Canada
e-mail: zimek@ualberta.ca

Keywords Local outlier · Spatial outlier · Video outlier · Network outlier

1 Introduction

A well-renowned definition states an outlier be "*an observation* (*or subset of observations*) *which appears to be inconsistent with the remainder of that set of data*" (Barnett and Lewis 1994). To grasp the meaning of "apparent inconsistency" is probably the most difficult part and heavily depends on the application area. Finding outliers (i.e., data objects that do not fit well to the general data distribution) is very important in many practical applications, including, e.g., credit card abuse detection in financial transaction data, the removal of measurement errors in scientific data, or the analysis of sports statistics data. There are many more application areas, and often similar algorithmic approaches to outlier detection have been proposed independently in the different areas that regardless of the algorithmic similarities grasp a different meaning of "outlierness". Or, vice versa, quite different algorithmic approaches sometimes implicitly pursue the same meaning of outlierness, but the definition of outlierness is hidden in the design of the algorithm and not obvious. As different as the questions that should be answered by outlier detection methods are the approaches to the problem of identifying outliers. In this study, we focus on unsupervised outlier detection, where different approaches have been categorized as *global* versus *local* approaches, i.e., the decision on the outlierness of some data object is based on the complete (global) database or only on a (local) selection of data objects. Here, we will scrutinize the true meaning of *locality* as in so-called "local" approaches.

The motivation behind this study is, on the one hand, that we find only subtle differences between several well-known approaches. We even see major similarities of different approaches in quite different application areas such as spatial/geographical data, video-sequences, or community-networks (to name just the examples we will actually be analyzing in this paper). Often, the meaning of "outlierness" is defined in a way typical for the application domain, but the algorithmic means, though more or less identical to approaches developed in different research areas, are re-developed (allegedly independently).

On the other hand, acclaimed similarities do not always hold when analyzed more thoroughly. In fact, we found confused statements concerning locality in the literature. For example, as we will discuss in the paper, though the method local distance-based outlier factor (DB-outlier) (LDOF, Zhang et al. 2009) names itself "local" and claims to be motivated by LOF (Breunig et al. 2000), LDOF is not local in the same sense as LOF. To clarify the notion of locality is therefore the second main motivation for this study. By studying LOF and LDOF in the framework for analysis which we are going to propose here, it becomes immediately clear, that both use different notions of locality. Overall, we argue that there are different meanings and different degrees of locality. Researchers and users should be aware of these differences. Proposals of new methods should probably clarify which degree and specific notion of locality is adopted.

Hence, we aim at generalizing many existing methods to a unified view that allows a better understanding of their shared properties and of the true meaning of their

differences by identification of the essential building blocks. This allows easier combination of existing algorithms and the realization that some are not fundamentally different (and thus are expected to be correlated strongly). A welcome side effect is the convenient construction of new combinations of these methods or the extension of existing basic approaches to specialized application domains. We will demonstrate that, using general methods re-designed by means of our framework of analysis, we can efficiently reproduce the results of exemplary, rather specialized state-of-the-art approaches which could raise the question of how highly specialized approaches were justified. We suggest that using a general outlier model to tackle special problems provides a reasonable baseline. We demonstrate how fundamental basic outlier models can be easily adapted to special problems. Specialized approaches should hence possibly elaborate on their outlier semantics as well as their justification or superiority in comparison with such a general yet efficient setting. In particular, we revisit spatial outlier detection as a special case of local outlier detection. This facilitates the comparison of different approaches, helps to understand their similarities and differences, and opens up new possibilities for advancement of specialized spatial methods.

In the following, we first present a broad overview of existing methods for outlier detection (Sect. 2). Considering different notions of "locality" in outlier detection methods, we then derive a dominant algorithmic structure and identify in an informal way common building blocks of many of these methods in a more detailed survey (Sect. 3). In a more abstract approach, we then formalize the analysis of outlier detection methods, their building blocks, and their notion of locality, in a novel model function based notion for outlier detection methods (Sect. 4). This framework for formal analysis allows for a straightforward comparison of similarities or differences between outlier detection methods, as we demonstrate in analyzing many existing approaches as well as in focused case studies. As a more practical application scenario, we show that a formalized approach to analyze "locality" in outlier detection in particular makes it possible to perform a thorough analysis and theoretical comparison of different approaches to spatial outlier detection wrt their outlier model, regardless of their algorithmic merits. We will find the results of several methods being correlated to a certain degree. We discuss in detail the notion of locality that is typical for spatial outlier detection (Sect. 5). As further example areas, we consider applications to video data as an example for temporal data (Sect. 6), and analyze DBLP data as an example for graph data (network outlier), where we also compare with a recent specialized method (Sect. 7). Finally, Sect. 8 concludes the article, discussing in summary the findings and the possibilities opened up for future research.

2 Related work

Existing outlier detection methods differ in the way they model and find the outliers and, thus, in the assumptions they, implicitly or explicitly, rely on. In general, statistical methods for outlier detection (identification, rejection) are based on presumed distributions of objects. The classical textbook of Barnett and Lewis (1994) discusses numerous tests for different distributions. The tests are optimized for each distribution dependent on the specific parameters of the corresponding distribution, the number of

expected outliers, and the space where to expect an outlier. A commonly used rule of thumb, known as the "$3 \cdot \sigma$-rule", is that points deviating more than three times the standard deviation from the mean of a normal distribution may be considered outliers.

The work of Knorr and Ng on the distance-based notion of outliers (DB-outlier, Knorr and Ng 1997, 1998; Knorr et al. 2000) unifies statistical distribution-based approaches and triggered the data mining community to develop many different approaches that have a less statistically oriented but more spatially oriented notion to model outliers. This model relies on the choice of two thresholds, D and p. In a set of objects (for these methods, usually a database of real-valued feature vectors) O, an object $o \in O$ is an outlier if at least a fraction p of all data objects in O has a distance greater than D from o. This idea is based on statistical reasoning but simplifies the approach to outlier detection considerably motivated by the need for scalable methods handling huge datasets. Work following this approach later on was primarily interested in algorithmic merits improving efficiency, for example based on approximations or improved pruning techniques for mining the top-n outliers only (e.g. Bay and Schwabacher 2003; Kollios et al. 2003; Vu and Gopalkrishnan 2009; Angiulli and Fassetti 2009). Ramaswamy et al. (2000) use the distances to the k nearest neighbors (NNs) and rank the objects according to their distances to their kth NN. A partition-based algorithm is then used to efficiently mine top-n outliers. As a variant, Angiulli and Pizzuti (2002) propose to use the sum of distances to all points within the set of kNNs (called the weight) as an outlier degree along with an approximation solution based on space filling curves to enable scalability with increasing data dimensionality. Another approximation based on reference points was proposed by Pei et al. (2006). Several efficient or approximate algorithms for mining DB-outliers have been studied by Orair et al. (2010). They identify common algorithmic techniques but do not discuss model properties, restricting themselves to the distance-based models of Knorr and Ng (1997), Ramaswamy et al. (2000), Angiulli and Pizzuti (2002). Density-based approaches consider ratios between the local density around an object and the local density around its neighboring objects. These approaches introduce the notion of local outliers. The basic idea is to assign a density-based local outlier factor (LOF) to each object of the database denoting a degree of outlierness (Breunig et al. 2000). The LOF compares the density of each object o of a database O with the density of the kNNs of o. An LOF value of approximately 1 indicates that the corresponding object is located within a region of homogeneous density (i.e., a cluster). If the difference between the density in the local neighborhood of o and the density around the kNNs of o is higher, o gets assigned a higher LOF value. The higher the LOF value of an object o is, the more distinctly is o considered an outlier. Several extensions and refinements of the basic LOF model have been proposed, e.g. a connectivity-based outlier factor (Tang et al. 2002). Using the concept of micro-clusters to efficiently mine the top-n density-based local outliers in large databases (i.e., those n objects having the highest LOF value) is proposed by Jin et al. (2001). A similar algorithm, named INFLO, is presented by Jin et al. (2006) for an extension of the LOF model using also the reverse NNs additionally to the NNs and considering a symmetric relationship between both values as a measure of outlierness. Papadimitriou et al. (2003) propose another local outlier detection schema named local outlier integral (LOCI) based on the concept of a multi-granularity deviation factor (MDEF). The main difference between the LOF and

the LOCI outlier model is that the MDEF of LOCI uses ε-neighborhoods rather than kNNs. The authors propose an approximate algorithm computing the LOCI values of each database object for any ε value. The results are displayed as a rather intuitive outlier plot. This way, the approach becomes much less sensitive to input parameters. The local DB-outlier detection (LDOF) approach (Zhang et al. 2009) is comparable in performance to classical kNN or LOF-based outlier detection but allegedly less sensitive to parameter values. Kriegel et al. (2009a, 2011) define the local outlier score as a probability also resulting in a more stable and reliable performance.

Some approaches specifically address the special needs in high dimensional data. Angle-based outlier detection (ABOD) (Kriegel et al. 2008) assesses the variance in angles between an outlier candidate and all other pairs of points, de Vries et al. (2010) use random projections to find anomalies approximately with a certain probability. Others try to account for a local feature relevance and search outliers in meaningful axis-parallel subspaces of the data space (Aggarwal and Yu 2001; Zhang et al. 2004; Müller et al. 2008; Kriegel et al. 2009b; Müller et al. 2010a,b, 2011; Nguyen et al. 2011; Keller et al. 2012) or even in arbitrarily-oriented subspaces, accounting for local correlations (Kriegel et al. 2012). In the area of spatial data mining (Roddick and Spiliopoulou 1999), the topic of spatial outliers has found quite some interest over the last decade (Anselin 1995; Shekhar et al. 2003; Lu et al. 2003; Kou et al. 2006; Sun and Chawla 2004; Chawla and Sun 2006; Liu et al. 2010; Chen et al. 2010). These approaches discern between spatial attributes (relevant for defining a neighborhood) and other attributes (usually only one additional attribute) where outliers deviate considerably from the corresponding attribute value of their spatial neighbors. How to derive spatial neighborhood and how to define "considerable deviation", however, differs from approach to approach. We will discuss this in detail below (Sect. 5). Other specialized approaches tackle for example outliers in time series (Takeuchi and Yamanishi 2006; Jagadish et al. 1999), outliers in graphs (e.g. in social networks or in DBLP) (Gao et al. 2010), outlying trajectories (Lee et al. 2008), outliers in stream data (Yamanishi et al. 2004; Pokrajac et al. 2007), in categorical or ordinal data (Yu et al. 2006), or in uncertain data (Aggarwal and Yu 2008). To combine different outlier models into an ensemble for outlier detection has been addressed in different ways (Lazarevic and Kumar 2005; Gao and Tan 2006; Nguyen et al. 2010; Kriegel et al. 2011; Schubert et al. 2012).

Although several survey papers have been published over the last decade (e.g. Hodge and Austin 2004; Agyemang et al. 2006; Hadi et al. 2009; Chandola et al. 2009, 2012; Su and Tsai 2011; Zimek et al. 2012), covering a broad selection of algorithmic approaches to outlier detection and quite different application scenarios, there has been no thorough attempt to derive or even to formalize the common properties and the shared techniques and building blocks.

3 On different notions of locality in outlier detection approaches

In a rather general sense, the very nature of outlier detection requires the comparison of an object with a set of other objects wrt some property (e.g. the kNN distance or a density model). When comparing different outlier detection methods, we find

different levels of restriction of the set to compare with. Furthermore, the property to be compared is usually also derived from the dataset, taking into account, again, a set of other objects. Both sets, set A from which to derive the property for an object, and set B to compare with, need not be identical. We can name set A the *context set* for model building, and set B the *reference set* for model comparison.

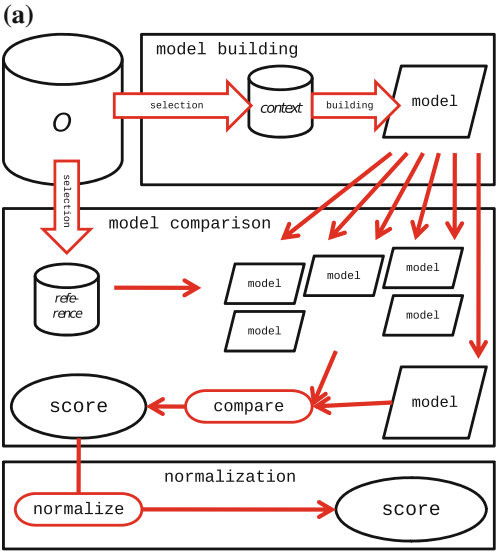
This decomposition has been implemented gradually (and probably only to a certain extent intentionally) during the development of outlier detection methods as surveyed in the previous section. Consider the fundamental statistical methods. They are modelling the complete dataset by a single distribution and judging an object basically by the probability of whether it could have been generated by the corresponding model. In this case, both the model building set and the reference set are the complete dataset. The first approach to DB-outlier detection already considers the local neighborhood by means of a range-query but compares the property thus derived with the complete dataset. The same is true for kNN-related outlier models: the model building set are the kNNs while the derived property is compared with the properties of the complete dataset as a reference. Thus the meaning of "locality" introduced in LOF (Breunig et al. 2000) relates to the locality of the reference set as well as the model building set. LOF uses the same neighborhood for both situations, but it could easily be abstracted to use different neighborhoods.

As opposed to the methods reflecting kNN distances, in truly local methods the resulting outlier score is adaptive to fluctuations in the local density and, hence, intended to be comparable over a dataset with varying densities. The central contribution of LOF and related methods is hence to enhance the comparability of outlier scores for a given dataset.

Based on this fundamental distinction of (i) the context used for model building and (ii) the context used as a reference for model comparison, we additionally identify the basic building blocks of (iii) the method used for learning a model (note that we are only interested in unsupervised learning procedures, to highlight this restriction, we name this also "building" or "computing" a model) and (iv) the method used for comparison of models of different objects. Finally, (v) the values describing the relations between different models (i.e., the outlier scores) can be normalized in some way.

This general algorithmic scheme, as visualized in Fig. 1, can accommodate many different outlier detection methods. Essentially, as the figure suggests, we can group these five elements in three algorithmically separable steps. First, the model building step assigns a model (or some simple property as, e.g., a distance value) to each object $o \in O$ based on some set context$(o) \subseteq O$. Second, the model comparison step compares the model of each object $o \in O$ with the models (built in the first step) of some set reference$(o) \subseteq O$. This step can be figured as omitted by simple methods just performing a ranking of the "models" (in this case usually consisting of one-dimensional values like distances) retrieved in the first step—but the ranking procedure can also be seen as comparison step with a global reference set. Finally, the score retrieved in step 2 can be normalized (step 3). All these steps are performed on the given dataset in order to identify (possible) outliers in an unsupervised manner.

In order to describe the exact methodological approach which some outlier detection method pursues, we therefore need to identify the procedures or definitions behind the



(b)

Input: database O
Output: (normalized) outlier score for each $o \in O$
  {STEP 1: model building}
  for all $o \in O$ **do**
    select context(o)
    model(o) := buildModel(o, context(o))
  end for
  {STEP 2: model comparison}
  for all $o \in O$ **do**
    select reference(o)
    score(o) := compare(model(o), $\{\text{model}(r)\}_{r \in \text{reference}(o)}$)
  end for
  {STEP 3: normalization}
  if normalization required **then**
    for all $o \in O$ **do**
      normalizedScore(o) := normalize(score(o))
    end for
  end if

Fig. 1 Typical schema of local outlier detection. **a** Workflow, **b** algorithmic schema

method names used in the framework (Fig. 1a): *context*, *buildModel*, *reference*, *compare*, and *normalize*. Let us exemplify these methods considering the LOF (Breunig et al. 2000), because it is a well-known outlier detection algorithm and many variants have been based on this basic approach. Also it uses most of these components.

LOF uses the kNNs for both context(o) and reference(o) of an object o. It uses a density model called "local reachability density" (lrd) based on the local context (however a more sophisticated estimation than just dividing the number of objects by the volume). This model is defined as follows (Breunig et al. 2000), reading kNNs of o for both, context(o) and reference(o):

$$\text{lrd}(o) := 1 \Big/ \frac{\sum_{p \in \text{context}(o)} \text{reachability-distance}_k(o,\ p)}{|\text{context}(o)|} \tag{1}$$

where the reachability-distance is given by:

$$\text{reachability-distance}_k(o,\ p) := \max\{k\text{NN-}dist(p),\ dist(o,\ p)\} \tag{2}$$

with some distance measure *dist* and kNN-$dist(p)$ being the distance between p and the kth NN of p (i.e., a value derived from the local context of p). The final score is then obtained using the comparison method

$$\text{LOF}(o) := \text{avg}_{n \in \text{reference}(o)} \frac{\text{lrd}(n)}{\text{lrd}(o)} \tag{3}$$

The only step not used in LOF is a global normalization. Local Outlier Probabilities model (LoOP, Kriegel et al. 2009a), for example, is a LOF variation that also uses this step.

Some variants define and use these building blocks in different ways. INFLO (Jin et al. 2006) uses more or less the same elements as LOF except for the reference set, which is defined as the intersection of the kNNs and the k reverse NNs (i.e., the set of those objects that list the query point among their kNNs). In the approach based on reference points (Pei et al. 2006), the model is computed in a similar way as in LOF but based on the context of approximated kNNs (based on the neighbors of reference points) and compared over the complete set of reference points. Also, they are implementing a normalization. LOCI (Papadimitriou et al. 2003) is explicitly using two different radii for the context and the reference set. Approaches for outlier mining in high dimensional data (e.g. Aggarwal and Yu 2001; Kriegel et al. 2009b, 2012; Müller et al. 2010b) usually assess neighborhoods based on distances in subspaces.

Aside from choosing different contexts and references for model building and model comparison, also the procedures of model building and model comparison can be quite diverse.

The model building step often is something as simple as using the object count in a particular radius (as a simple estimation of density). However, there are also much more complex models possible. A statistical baseline approach is an EM-like fitting of a Gaussian (inliers) and a uniform (outliers) distribution on the complete dataset (described by Tan et al. 2006). Here, context and reference set are global but the model definition is statistically refined. Opposed to that, database-oriented approaches often try to simplify the model building for the sake of efficiency. ABOD (Kriegel et al. 2008), for example, computes the pairwise angles that local objects appear under. This model is then condensed to a single feature, the variance of the angle spectrum. LOF (Breunig et al. 2000) assumes an OPTICS-like model (Ankerst et al. 1999) and estimates the density level at which the point became a cluster member. LOCI (Papadimitriou et al. 2003) compares object counts for different regions. LoOP (Kriegel et al. 2009a) models local Gaussian distributions and estimates density using the variance of the resulting Gaussian distribution. LDOF (Zhang et al. 2009) computes the pairwise distances of local neighbors, and reduces this to the mean value

as single-valued feature. The definitions of outliers of Ramaswamy et al. (2000) and Angiulli and Pizzuti (2002) (regardless of their algorithmic merits) differ only marginally in the model definition (distance to the kth NN vs. aggregated distances for the first k neighbors—this sum of distances provides a certain smoothing effect on the outlier scores but essentially measures the same property).

While single-dimensional values such as distance or density can easily be used for ranking directly, methods such as LOF (Breunig et al. 2000) very successfully use more complex comparison methods. While the simple methods usually just take the maximum or minimum global value as the most prominent outlier, the advanced methods usually use a "local" context again as reference. LOF, for example, computes the quotient of the object's reachability density and the average reachability density of its neighbors. It thus no longer detects the globally least dense point as outlier, but those that are significantly less dense than their neighbors. This advanced model comparison marks some of the so-called "local" methods as truly local. Examples for local methods, aside from LOF, are INFLO (Jin et al. 2006), reference point-based (Pei et al. 2006), LOCI (Papadimitriou et al. 2003), and LoOP (Kriegel et al. 2009a). Let us note that LDOF (Zhang et al. 2009), though acclaimed to be local, is actually local only in the context but global in the model comparison reference set. This example of a possible misunderstanding of locality already demonstrates that a detailed scrutiny of the meaning of "local" in outlier detection could be rather useful in order to better understand the scope and contribution of different methods. These different notions—and complexities—of locality are the core interest in the present study.

Table 1 gives an overview on some well-known outlier detection methods represented in this framework. We list here only keywords or short terms reminding on the basic ideas of the listed approaches, as discussed earlier in this section and in Sect. 2, to informally identify the fundamental algorithmic building blocks of these methods for a first overview. We will discuss the models behind these algorithmic building blocks in a more formal manner in the subsequent section. As the normalization step is not present in many of these examples, let us note again that some methods to provide a normalization for those outlier detection models have been proposed by Kriegel et al. (2011). Some methods (e.g., DB-outlier) directly derive a label (outlier vs. inlier) instead of performing some score normalization.

In summary, the identified common algorithmic scheme for local outlier detection consists of the following components:

(1) Context: a "local" context of an object o for model building (context(o))
(2) Model: the method used for building the model
(3) Reference: a "reference" context of object o for model comparison (reference(o))
(4) Comparison: the method used for model comparison
(5) Normalization: a (global) normalization procedure

These are the common algorithmic building blocks of local outlier detection approaches and they enable us at the same time to see global approaches as special cases of local approaches (and, as we will discuss in the following, spatial outlier detection and even outlier detection in special data like video streams or network data can be seen as a specialization of this framework as well). Though not every

Table 1 Overview: local outlier methods

Method	Context Reference	Model Comparison Normalization
Outlier EM (Tan et al. 2006)	Global	EM-fitting of two models (Gaussian vs. uniform)
	Global	Conditional probability
DB-outlier (Knorr et al. 2000)	Range	Object count
	Global	Threshold
kNN outlier (Ramaswamy et al. 2000)	kNN	Maximum distance
	Global	Descending
Aggregated kNN (Angiulli and Pizzuti 2002)	kNN	Sum of distances
	Global	Descending
LOF (Breunig et al. 2000)	kNN	Reachability density
	kNN	Average quotient
INFLO (Jin et al. 2006)	kNN	Reachability density
	kNN $\cap$ rev.kNN	Average quotient
Reference points (Pei et al. 2006)	k approx. NN	Density estimate
	Reference points	Descending $1 - \frac{\text{Value}}{\max\{\text{values}\}}$
LOCI (Papadimitriou et al. 2003)	Range r_1	Objective count at any $r < r_1$
	Range r_2	Quotient with average
LDOF (Zhang et al. 2009)	kNN	Distances quotient
	Global	Descending
LoOP (Kriegel et al. 2009a)	kNN	1/RMSD
	kNN	1/RMSD Erf
ABOD (Kriegel et al. 2008)	kNN	Angle variance
	Global	Ascending
High.-dim. (Aggarwal and Yu 2001)	Subspaces	Object count
	Global	Threshold
SOD (Kriegel et al. 2009b)	SNN-based kNN	Subspace model
	Global	Descending
Subspace outlier ranking (Müller et al. 2010b)	Adaptive range in subspace	Subspace density model
	Global, but subspace	Deviation from exp. density $\frac{\text{Dens.}}{\text{dev.}}$ if dev. > 1

component is actually used or present in every instance of outlier detection methods, many existing methods can be unified using this algorithmic framework.

4 Formalized analysis of outlier detection models and locality

Before we inspect specializations (Sects. 5–7) of the algorithmic framework, we now take a higher perspective and discuss a framework for formal analysis of outlier detection models (Sect. 4.1). We discuss common context functions, that are used to derive context sets or reference sets (Sect. 4.2). Applying the framework, we then derive formal descriptions of outlier detection algorithms, based on the successive execution of model functions (Sect. 4.3). Based on this framework for formal analysis, we discuss in two case studies similarities and differences among variants of LOF (Sect. 4.4) and improved understanding of a complex method (Sect. 4.5). Finally, we show by means

of dependency graphs in a formal way the level of locality actually used in "local" outlier detection algorithms (Sect. 4.6).

4.1 Generalized outlier detection model framework

As we have seen in the analysis and generalization of existing work (Sect. 3), there are reoccurring patterns in outlier detection. The most prominent pattern is the computation of a model, based on a reference set of objects. Many established methods can be formulated to apply this pattern twice, and the normalization step can also be formatted to follow this pattern. As we will show here, this results in a general formal framework for analysis of outlier detection models. Within the framework, we focus on the properties of outliers, not on the actual computation of the scores. The formalization into multiple steps however allows the construction of generic algorithms that usually are not of higher complexity than the originally proposed algorithms.

In order to define the general model, we first need to define its basic components and building blocks. Let O be the objects in the database uniquely identified.

Definition 1 (*Context Function*)
A context function c_i is a function to the powerset $\mathcal{P}$ of O

$$c_i : O \to \mathcal{P}(O)$$

that maps objects o to their context set $c_i(o) \subseteq O$, usually objects that are considered to be relevant to judging the outlierness of o.

Definition 2 (*Intermediate Data*)
For some value domain V_i, let $D_i(o) \in V_i$ with $i = 0, \ldots, n$ be the **intermediate data** of step i for object o, with $D_0(o) = o$ the initial data (identity map).

Definition 3 (*Available Data*)
Let the collected intermediate data, $\mathfrak{D}_i(o) := \{D_k(o) | k \leq i\}$ be the **available data** after step i.

Definition 4 (*Model Function*)
A model function f_i is a function

$$f_i\,(o,\ c_i(o),\ \mathfrak{D}_{i-1}) =: D_i(o)$$

where o is the current object, $c_i(o)$ is the context set of the object, $\mathfrak{D}_{i-1}(o)$ is the available data before executing function f_i, and $D_i(o)$ is the output (model) data for object o.

Definition 5 (*Algorithm step*)
An algorithm step p_i is the task of computing a model function for the whole database, and can be formalized as:

$$p_i : \mathfrak{D}_{i-1} \mapsto \{D_0, \ldots, D_i\} = \mathfrak{D}_i,$$

where the intermediate data D_i is the output of f_i for all objects:

$$D_i := \{o \mapsto f_i\,(o,\, c_i(o),\, \mathfrak{D}_{i-1})\}$$

Each step computes a new intermediate dataset based on the existing intermediate maps and the new data obtained by computing f_i on all objects to obtain the new dataset D_i. When executed in sequence, they transform the data as follows:

$$\underbrace{\{D_0\}}_{=\mathfrak{D}_0} \mapsto_{p_1} \underbrace{\{D_0,\, D_1\}}_{=\mathfrak{D}_1} \mapsto_{p_2} \underbrace{\{D_0,\, D_1,\, D_2\}}_{=\mathfrak{D}_2} \ldots \mapsto_{p_i} \underbrace{\{D_0, \ldots, D_i\}}_{=\mathfrak{D}_i}$$

Executing the steps p_i one after the other is called the **canonical algorithm** for computing the outlier result.

Definition 6 (*Generalized Outlier Detection Model*)
A generalized outlier detection model is a series of model functions and context functions

$$[(f_0,\, c_0)\,, \ldots, (f_i,\, c_i)]$$

such that $D_i : O \to \mathbb{R}$ is the map onto the objects' outlier score.

Any outlier detection can trivially be captured in this model by using the outlier detection as an arbitrarily complex function f_0 with $c_0 = \mathit{global}$ being the full dataset. In fact, the formalization allows to use any computable function this way. However, in the following we will show that we are able to model many well-known methods using much more primitive functions. Usually functions of complexity $\mathcal{O}(1)$ or $\mathcal{O}(|c_i(o)|)$, with a focus on analyzing the notion of locality used in the analyzed methods.

Definition 7 (*Linear Generalized Outlier Detection Model*)
A Generalized Outlier Detection Model is called linear, if and only if for each step $p_i = (f_i,\, c_i)$ the complexity of f_i is at most in $\mathcal{O}(|c_i(o)|)$.

The canonical algorithm can then compute a linear generalized outlier detection model in $\mathcal{O}(i \cdot |O| \cdot |c_i(o)|)$ plus the time needed to compute the context sets. This definition rules out using an existing complete, possibly complex, outlier detection algorithm as algorithm step. Note that we do not impose a constraint on computing the c_i, and indeed many of the popular vector space methods will take $\mathcal{O}(|O|^2)$ time without index support to compute all context sets and reference sets. However for graph data, the c_i are part of the input data, so it makes sense to treat the context set computation separately. To fully control complexity, it may be desirable to put a limit on the context set size, for example by assuming $|c_i| \ll |O|$. However we will see a number of cases where it is more understandable to specify the context set as global, even when the total computation is still linear in the number of objects—for example when outlier scores are normalized. Therefore, this can only be used as a rough estimation of the total complexity of the canonical algorithm, or this optimization of sharing a computation among multiple observations as an optimization of the canonical

Table 2 Common context definitions

Context function	Definition
$\text{range}_{d,\varepsilon}$	Range query with distance function d and radius ε
$k\text{NN}_{d,k}$	kNN query with distance function d
$k\text{NN}'_{d,k}$	kNN query with d, excluding query object
$\text{r}k\text{NN}_{d,k}$	Reverse kNN query with distance function d
$\text{SNN}_{d,s,k}$	k best with respect to the shared $k\text{NN}_{d,s}$
global	O (the complete database)
$\emptyset$	The empty set
spatial	Spatial neighborhood (predefined)
prev_k	Previous k objects (temporal)

algorithm. From a mathematical point of view, in such a normalization step we do have a dependency of each object to every other object. This captures the fact that a single object change in the dataset may change the normalization (for example by changing the minimum or maximum value). Hence, we do not propose this formalization as a generalization of whatsoever outlier detection models. Rather, the formalization allows to decompose many existing outlier detection models in their simple steps and, by means of this decomposition, it allows to analyze many existing methods and to state their similarities to each other or their essential differences and individual merits. By extracting the simple building blocks, using the formalism also allows for simple complexity analysis of the baseline algorithm given implicitly by executing the algorithm steps one after the other.

Let us therefore emphasize that we are not actually proposing this general model as a *new method* that is able to express everything, but as a means of analysis of existing (and future) methods. Accordingly, in the remainder of this paper, we analyze existing methods by this formalization, in order to survey important abstract outlier detection methods (this section) and to relate also to specialized methods by demonstrating the applicability of the model to specialized notions of locality (spatial data, video sequences, graph data) in Sects. 5–7, reproducing the results of state-of-the-art methods in these quite diverse fields with a straightforward baseline method (built as composition in the formalized general outlier detection model).

In the following, we introduce a number of example functions that can be used to define many well-known outlier detection models in a uniform manner. The functions are summarized in Table 2 (context functions) and Table 3 (model functions).

4.2 Fundamental context functions

As important as the local context is for local outlier detection, as much is it essentially an input parameter. Locality is commonly defined using the k NNs for a given distance function d, a range query with a radius of ε, a spatial neighborhood based on graph adjacency or polygon adjacency, or a temporal context, e.g. in terms of a sliding window. Sometimes, there are slight variations. For example the k nearest neighbors may or may not include the query object itself, may consist of exactly k neighbors (which might not be unique) or may include additional neighbors that share

Table 3 Common model functions

Model function	Definition
$\text{count}(o,\ c(o),\ \mathfrak{D})$	$\lvert c(o)\rvert$
$\text{maxdist}_{i,d}(o,\ c(o),\ \mathfrak{D})$	$\max_{n\in c(o)} d(D_i(o),\ D_i(n))$
$\text{avgdist}_{i,d}(o,\ c(o),\ \mathfrak{D})$	$\text{mean}_{n\in c(o)} d(D_i(o),\ D_i(n))$
$\text{pairdist}_{i,d}(o,\ c(o),\ \mathfrak{D})$	$\frac{1}{\lvert c(o)\rvert\cdot(\lvert c(o)\rvert-1)} \sum_{n\in c(o)} \sum_{m\in c(o), m\neq n} d(D_i(n),\ D_i(m))$
$\text{lrd}_{i,j,d}(o,\ c(o),\ \mathfrak{D})$	$1/\text{mean}_{n\in c(o)} \max\{D_j(n),\ d(D_i(o),\ D_i(n))\}$
$\text{mean}_i(o,\ c(o),\ \mathfrak{D})$	$\text{mean}_{n\in c(o)} D_i(n)$
$\text{stddev}_i(o,\ c(o),\ \mathfrak{D})$	$\text{stddev}_{n\in c(o)} D_i(n)$
$\text{frac}_{i,j}(o,\ c(o),\ \mathfrak{D})$	$\frac{D_i(o)}{D_j(n)}$
$\text{pdist}_{i,d}(o,\ c(o),\ \mathfrak{D})$	$\lambda\sqrt{\text{mean}_{n\in c(o)} d(D_i(o),\ D_i(n))^2}$
$\text{erf}'_{i,\lambda}(o,\ c(o),\ \mathfrak{D})$	$\max\left\{0,\ \text{erf}\left(\frac{1}{\sqrt{2}} \frac{D_i(o)}{\lambda\cdot\sqrt{\text{mean}_{n\in c(o)} D_i(n)^2}}\right)\right\}$
Utility function	**Definition**
$\text{mean}_{o\in O} f(o)$	$\frac{1}{\lvert O\rvert} \sum_{o\in O} f(o)$ (arithmetic mean of f in O)
$z(p,\ f,\ O)$	$(f(p) - \text{mean}_{o\in O} f(o))/\text{stddev}_{o\in O} f(o)$ (standard score)

the identical distance with the kth neighbor. We do not cover all of these variations here. Some methods implicitly assume that there are no objects with a distance of 0, and may even divide by 0 when there are more than k objects at distance 0.

Table 2 lists these contexts without a detailed formalization (which is trivial in these cases). For completeness, we also include the complete database (denoted as global) or no objects (denoted as $\emptyset$) as trivial contexts. This allows for an improved reuse of model functions and restricts the number of required specializations.

4.3 Fundamental model functions

The key building blocks of an outlier detection model are the model functions that compute key properties. Many will output into the real number domain, although complex models such as covariance matrices are possible. Here, we define a number of commonly used functions. Additional functions are summarized in Table 3. We loosely follow chronological order for these methods, which largely reflects their complexity as well. When these blocks are then combined into outlier detection models, overlaps and similarities between models will become visible.

The initial DB-outlier definition by Knorr et al. (2000) did not yet address "local" outlier detection, but provided a binary decision based on a threshold on the relative number of objects outside a given radius. By turning the density threshold at which a point would become a DB-outlier into a score it becomes a ranking outlier detection method, as introduced by Kriegel et al. (2011).[1] While locality was not discussed explicitly, the dependence on the distance function implies a certain degree of locality.

[1] This adaptation could also be considered "Schönfinkeling" (or, tastier, "Currying") of the original DB-outlier model.

When formalizing DB-outliers, the essential building block is to count the number of objects within the query range (the context of the object), which will be the first example for a model function:

Definition 8 (*Object Count Model Function*)

$$\text{count}(o,\ c(o),\ \mathfrak{D}) := |c(o)|$$

Definition 9 (*Scoring DB-outlier*)
Scoring DB-outlier (Knorr et al. 2000; Kriegel et al. 2011) is a linear generalized outlier detection model for distance function d and range ε with

$$\text{DB-outlier}(d,\ \varepsilon) = \left[\left(\text{count},\ \text{range}_{d,\varepsilon}\right)\right]$$

Instead of using the number of objects as a score, a different way of turning DB-outliers into a scoring method is to use the radius at which the number of neighbors would suffice the DB-outlier definition as score. For outliers, a much larger neighborhood would be required, for inliers a smaller distance would be sufficient. Ramaswamy et al. (2000) formalized this notion of outliers, out of which we extract the next model function, which computes the maximum distance to an object of the context set:

Definition 10 (*Maximum Distance Model Function*)

$$\text{maxdist}_{i,d}(o,\ c(o),\ \mathfrak{D}) := \max_{n \in c(o)} d\left(D_i(o),\ D_i(n)\right)$$

Definition 11 (*kNN Outlier*)
kNN outlier (Ramaswamy et al. 2000) is a linear generalized outlier detection model for distance function d and neighborhood size k with

$$k\text{NN}(d,\ k) = \left[\left(\text{maxdist}_{0,d},\ k\text{NN}_{d,k}\right)\right]$$

This work was then again generalized and extended by Angiulli and Pizzuti (2002) to improve stability by taking the average (or sum) instead of the maximum distance of the neighborhood.

Definition 12 (*Average Distance Model Function*)

$$\text{avgdist}_{i,d}(o,\ c(o),\ \mathfrak{D}) := \text{mean}_{n \in c(o)} d\left(D_i(o),\ D_i(n)\right)$$

Definition 13 (*Aggregate kNN Outlier*)
Aggregate kNN outlier (Angiulli and Pizzuti 2002) is a linear generalized outlier detection model for distance function d and neighborhood size k with

$$Ak\text{NN}(d,\ k) = \left[\left(\text{avgdist}_{0,d},\ k\text{NN}_{d,k}\right)\right]$$

Up to now, our framework was only able to represent the known two methods. The additional combinations—computing the number of objects in the k-neighborhood and computing the maximum distance within a fixed radius—were of little interest. This model function actually allows us to consider an interesting new combination: computing the average distance within a fixed radius around an object could be a reasonable score, assuming that an outlier will likely have less close and more distant neighbors. This method is however not very useful in practice, since the radius parameter is particularly hard to choose, and the value becomes unstable when there are only few neighbors available. The combination by Angiulli and Pizzuti (2002) with a fixed size neighborhood is much more reasonable.

So far, the algorithms were essentially identical to the application of the model function onto the context of an object. The first method known to use a more complex approach is the LOF (Breunig et al. 2000). Instead of just computing a local score on the object itself, it in fact computes a particular property—a density estimation—for each object, then again compares these values within the local neighborhood. For modelling LOF we need a total of four model functions, one of which we have seen before in Definition 10: what is called k-distance in LOF is essentially the $\text{maxdist}_{i,d}$ function that the kNN-Outlier method by Ramaswamy et al. (2000) used. It serves a stabilizing role in LOF, and we will then discuss how it can be removed to obtain a "Simplified-LOF" method (which has actually been used—probably unintentionally—in many approaches that allegedly were based on the original LOF idea, see the case study in Sect. 4.4). The second model function of LOF computes a density model known as lrd and is the key component of LOF:

Definition 14 (*Lrd Model Function*)

$$\text{lrd}_{i,j,d}(o,\ c(o),\ \mathfrak{D}) := 1/\text{mean}_{n\in c(o)} \max\left\{D_j(n),\ d\left(D_i(o),\ D_i(n)\right)\right\}$$

where mean denotes the arithmetic mean operator and D_j is the $\text{maxdist}_{i,d}$ result obtained before, while d is the distance function.

The other two model functions required for the definition of LOF are very basic operations that we will however see in many of the following methods, the computation of a mean value over the neighborhood and a simple, context-free comparison step for simple numeric models by computing the fraction.

Definition 15 (*Mean Model Function*)

$$\text{mean}_i(o,\ c(o),\ \mathfrak{D}) :=\ \text{mean}_{n\in c(o)} D_i(n)$$

Definition 16 (*Fraction Model Function*)

$$\text{frac}_{i,j}(o,\ \_,\ \mathfrak{D}) := \frac{D_i(o)}{D_j(o)}$$

These four model functions (Definitions 10, 14–16) can now be connected together to form the LOF model.

Definition 17 (*Local Outlier Factor*)
LOF (Breunig et al. 2000) is a linear generalized outlier detection model for distance function d and neighborhood size k with

$$\begin{aligned}\mathrm{LOF}(d,\,k) = \big[&\left(\mathrm{maxdist}_{0,d},\,k\mathrm{NN}_{d,k}\right),\\ &\left(\mathrm{lrd}_{0,1,d},\,k\mathrm{NN}_{d,k}\right),\,\left(\mathrm{mean}_{2},\,k\mathrm{NN}_{d,k}\right),\,\left(\mathrm{frac}_{3,2},\,\emptyset\right)\big]\end{aligned}$$

Note the chaining of operations given by the indices on the operators: the maximum distance is computed on the original data, the lrd uses the original data and this maximum distance, the final step puts the density models only into relation with each other. We will use this later to obtain a dependency graph representation of the models.

4.4 Case study: variants of LOF

For LOF, we have pointed out the often overlooked detail of the reachability distance. A theme commonly seen in LOF extensions is to drop the second model function of LOF and instead use a much simpler density estimation, resulting in the following base model of a density-quotient outlier model.

Definition 18 (*Simplified-LOF*)
Simplified-LOF is a linear generalized outlier detection model for distance function d and neighborhood size k with

$$\mathrm{Simplified\text{-}LOF}(d,\,k) = \left[\left(1/\mathrm{maxdist}_{0,d},\,k\mathrm{NN}_{d,k}\right),\,\left(\mathrm{mean}_{1},\,k\mathrm{NN}_{d,k}\right),\left(\mathrm{frac}_{2,1},\,\emptyset\right)\right]$$

In this baseline method, local density is estimated by the inverse of the k-distance, and the combination of mean and frac forms the core of most algorithms that reference LOF.

Given that LOF is clearly distance-based and the article introducing the LDOF (Zhang et al. 2009) compares the algorithm to LOF, one would expect a strong overlap of these methods. Our representation shows that it actually is a variation of the Simplified-LOF: instead of taking the maximum distance, LDOF uses the avgdist model for estimating the local density, and instead of the mean density it uses pairwise distances for estimating a neighborhood density.

Definition 19 (*Pairwise Distance Model Function*)

$$\mathrm{pairdist}_{i,d}(o,\,c(o),\,\mathfrak{D}) := \frac{1}{|c(o)|\cdot(|c(o)|-1)}\sum_{n,m\in c(o),\,m\neq n} d\left(D_i(n),\,D_i(m)\right)$$

We can now combine these to form LDOF.

Definition 20 (*Local DB-outlier Factor*)
LDOF (Zhang et al. 2009) is a linear generalized outlier detection model:

$$\mathrm{LDOF}(d,\,k) = \left[\left(\mathrm{avgdist}_{0,d},\,k\mathrm{NN}'_{d,k}\right),\,\left(\mathrm{pairdist}_{0,d},\,k\mathrm{NN}'_{d,k}\right),\,\left(\mathrm{frac}_{1,2},\,\emptyset\right),\,\right]$$

A different kind of variation of Simplified-LOF is Influenced Outlierness (INFLO, Jin et al. 2006), which diverges from Simplified-LOF by using a different context set for its second model function.

Definition 21 (*Influenced Outlierness*)
INFLO (Jin et al. 2006) is a generalized outlier detection model for distance function d and neighborhood size k with

$$\begin{aligned}&\mathrm{INFLO}(d,\ k) = \\ &\quad \left[\left(1/\mathrm{maxdist}_{0,d},\ k\mathrm{NN}_{d,k}\right),\ \left(\mathrm{mean}_1,\ k\mathrm{NN}_{d,k} \cap \mathrm{r}k\mathrm{NN}_{d,k}\right),\ \left(\mathrm{frac}_{2,1}, \emptyset\right)\right]\end{aligned}$$

Another Simplified-LOF variation that is more interesting for our framework since it introduces a new kind of model function is the LoOP model (Kriegel et al. 2009a), which includes an additional normalization step based on the assumption that the quotient scores are normally distributed (with a fixed mean). This step is interesting, because it involves the complete dataset as context for estimating the distribution parameter. Alternative normalization functions that can directly replace this model function have been discussed by Kriegel et al. (2011).

Definition 22 (*Error Function Normalization Model Function*)

$$\mathrm{erf}'_{i,\lambda}(o,\ c(o),\ \mathfrak{D}) := \max\left\{0,\ \mathrm{erf}\left(\frac{1}{\sqrt{2}}\frac{D_i(o)}{\lambda \cdot \sqrt{\mathrm{mean}_{n\in c(o)} D_i(n)^2}}\right)\right\}$$

Additionally, LoOP uses a different density estimation function, that assumes a half-Gaussian distribution of the local distances.

Definition 23 (*Probability Density Model Function*)

$$\mathrm{pdist}_{i,d}(o,\ c(o),\ \mathfrak{D}) := \lambda\sqrt{\mathrm{mean}_{n\in c(o)} d\left(D_i(o),\ D_i(n)\right)^2}$$

Definition 24 (*Local Outlier Probabilities*)
LoOP (Kriegel et al. 2009a) is a linear generalized outlier detection model:

$$\begin{aligned}&\mathrm{LoOP}(d,\ k,\ \lambda) \\ &\quad = \left[\left(\mathrm{pdist}_{0,d},\ k\mathrm{NN}_{d,k}\right),\ \left(\mathrm{mean}_1,\ k\mathrm{NN}_{d,k}\right),\ \left(\mathrm{frac}_{1,2} - 1,\ \emptyset\right),\ \left(\mathrm{erf}'_{2,\lambda},\ \mathrm{global}\right)\right]\end{aligned}$$

Again, the second and third model functions are an easily recognizable pattern of Simplified-LOF. The first model function (a statistically more robust density estimation) and the fourth model function, which serves as global normalization step, are the key contributions of this method.

Another recent example for (unintentional?) use of Simplified-LOF instead of the actual LOF model is projection-indexed NNs (de Vries et al. 2010). Thus, overall, this case study may demonstrate that a better understanding of the actually used outlier model and the adopted notion of locality in some algorithm may help to reveal the relationships, similarities, and differences between some approaches in the literature.

4.5 Case study: plot models as used by LOCI

A well known method that uses a more complex model—so far, all model functions in fact were mappings onto the real numbers—is the local correlation integral (LOCI, Papadimitriou et al. 2003). Where LOF only used the maxdist function (Definition 10) with a parameter k, LOCI uses a plot consisting of radius/count pairs that give the number of neighbors within the given radius. We represent these plots as a map with the signature $r \mapsto v$, mapping a radius r to a value v. Again we present a slight generalization of LOCI that instead of producing a binary result for a given threshold (k_σ in LOCI) produces a score representing the threshold value of k_σ where the point would become an outlier. Some fine details such as the minimum radius $r_{\min}$ and minimum neighborhood size $\hat{n}_{\min}$ were also omitted for brevity, as was the computation of interesting values for r.

LOCI uses a—difficult to grasp—interplay of two radii, r and αr. In general, the radius of αr is used for density estimation, the radius of r is used as reference set. The first model function for LOCI is the density estimation using the modified radius. We deliberately assign it to the radius of r to simplify the whole LOCI model, reducing the use of α to this single occurrence.

Definition 25 (*Density Plot Model Function*)

$$\text{denplot}'_{d,\alpha}(o,\ c(o),\ \mathfrak{D}) := r \mapsto |\{n \in c(o) \wedge d(n,\ o) < \alpha r\}|$$

Similar to the mean function, these plots are averaged over their neighbor sets to obtain a type of mean count integral, taking only those neighbors into account that are within the given radius (the use of α now is hidden in D_i).

Definition 26 (*Plot Mean Model Function*)

$$\text{plotmean}_{i,d}(o,\ c(o),\ \mathfrak{D}) := r \mapsto \text{mean}_{p \in c(o) \wedge d(p,\ o) < r} D_i(p)(r)$$

The quantity denoted as σ_{MDEF} in LOCI is the corresponding standard deviation, normalized additionally by the mean.

Definition 27 (*MDEF Standard Deviation Model Function*)

$$\text{sigmdef}_{i,j,d}(o,\ c(o),\ \mathfrak{D}) := r \mapsto \frac{\text{stddev}_{p \in c(o) \wedge d(p,\ o) < r} D_i(p)(r)}{D_j(o)(r)}$$

As we will be able to see below, the normalization is not needed, at which point we just have the common standard deviation formula.

Definition 28 (*Plot Standard Deviation Model Function*)

$$\text{plotstddev}_{i,d}(o,\ c(o),\ \mathfrak{D}) := r \mapsto \text{stddev}_{p \in c(o) \wedge d(p,\ o) < r} D_j(o)(r)$$

The comparison step is then another quotient function, resulting in the MDEF plot, by computing the quotient of the object count to the mean object count, where lower values than 1 indicate outlierness.

Definition 29 (*MDEF Plot Model Function*)

$$\begin{aligned}\text{plotmdef}_{i,j}(o,\ \_,\ \mathfrak{D}) :=&\quad r \mapsto 1 - \tfrac{D_i(o)(r)}{D_j(o)(r)}\\ \equiv&\ r \mapsto \tfrac{D_j(o)(r) - D_i(o)(r)}{D_j(o)(r)}\end{aligned}$$

We carry out the equivalent simplification as we did in Definition 28:

Definition 30 (*Plot Delta Model Function*)

$$\text{plotdelta}_{i,j}(o,\ \_,\ \mathfrak{D}) := r \mapsto D_j(o)(r) - D_i(o)(r)$$

The value is finally normalized by taking the local MDEF standard deviation into account. Applying this function to the results of Definitions 27 and 29 is equivalent to applying it to the results of Definitions 28 and 30.

Definition 31 (*Plot Fraction Model Function*)

$$\text{plotfrac}_{i,j}(o,\ \_,\ \mathfrak{D}) := r \mapsto D_i(o)(r)/D_j(o)(r)$$

LOCI considers points to be outliers based on their highest MDEF score, which can be turned into a model function reducing the plot to a single score

Definition 32 (*Plot Maximum Model Function*)

$$\text{plotmax}_i(o,\ \_,\ \mathfrak{D}) := \max_r D_i(o)(r)$$

Definition 33 (*Local Correlation Integral*)
LOCI (Papadimitriou et al. 2003) is a generalized outlier detection model:

$$\begin{aligned}\text{LOCI}(d,\ r,\ \alpha) = \big[&\left(\text{denplot}'_{d,\alpha},\ \text{range}_{d,\alpha r_{\max}}\right),\ \left(\text{plotmean}_{1,d},\ \text{range}_{d,r_{\max}}\right),\\ &\left(\text{sigmdef}_{1,2,d},\ \text{range}_{d,r_{\max}}\right),\ \left(\text{plotmdef}_{1,2},\ \emptyset\right),\\ &\left(\text{plotfrac}_{4,3},\ \emptyset\right),\ \left(\text{plotmax}_5,\ \emptyset\right)\big]\end{aligned}$$

An equivalent definition can be given using the simplified formulas and standard functions.

Definition 34 (*Local Correlation Integral* (*equiv.*))
LOCI (Papadimitriou et al. 2003) is a generalized outlier detection model:

$$\begin{aligned}\text{LOCI}(d,\ r,\ \alpha) \equiv \big[&\left(\text{denplot}'_{d,\alpha},\ \text{range}_{d,\alpha r_{\max}}\right),\ \left(\text{plotmean}_{1,d},\ \text{range}_{d,r_{\max}}\right),\\ &\left(\text{plotstddev}_{1,d},\ \text{range}_{d,r_{\max}}\right),\ \left(\text{plotdelta}_{1,2},\ \emptyset\right),\ \left(\text{plotfrac}_{4,3},\ \emptyset\right),\ \left(\text{plotmax}_5,\ \emptyset\right)\big]\end{aligned}$$

At this point, we do not only have a clear understanding what LOCI actually computes which is much more difficult from the formulas in the original publication. We can also see that it does follow the common pattern of firstly computing a local

feature—here the mass of the neighborhood for a radius αr, as per Definition 25—and secondly comparing the deviation of this value from the value of its neighbors (this time with radius r) and normalizing it. Finally, it does all this for a variety of radius values, using the maximum score obtained.

4.6 Dependency graph and order of locality

Analyzing the outlier models we have seen so far in this uniform framework leads to the obvious question on how to make use of this structural knowledge. We have seen that the model functions are shared among various algorithms, that we can do new recombinations such as Simplified-LOF, and that we can easily identify building blocks used in different methods (which is not always clear from the original publications). For example, kNN-Outlier detection is the first model function of LOF, so when we are computing LOF we implicitly also compute the kNN-Outlier score. Context sets such as the kNN context are used throughout the algorithms and can sensibly be precomputed. In fact, all model functions (with the sole exception of LOCI) that we have seen so far can then be computed in essentially linear time of the database size (more precisely, most algorithms are in $|\mathcal{O}| \cdot \|$ then), the expensive step in all of these algorithms is the computation of the context set: computing the kNN of an object has the complexity $\mathcal{O}(|O|^2)$ when done naïvely and $\mathcal{O}(|O| \log |O|)$ when an appropriate index structure is available.

In the model functions as written in our definitions, we also implicitly denoted a dependency graph that can help us analyzing the corresponding algorithms for efficient implementation. The functions often have integer indexes that reference previous results, which directly encodes the dependency graph. A dependency in this graph means that a model function f_i uses the intermediate data D_j produced by a model function $j < i$. In addition to the dependencies between model functions, we also include dependency edges to the input data and context functions. Figure 2 contains explicit dependency graphs of the methods LOF, LDOF, LoOP and Simplified-LOF. Model functions are represented as rounded boxes, while input data D_0 and context functions are represented as ellipses. Let us define the dependency graph formally.

Definition 35 (*Dependency Graph*) Let G be the dependency graph of the algorithm defined as

$$(j \longleftarrow i) \in G \Leftrightarrow f_i \text{ depends on } D_j$$
$$(j \leftarrow\!\!- c_j) \in G \Leftrightarrow f_i \text{ depends on } c_j$$

Steps that do not depend on each other can trivially be computed in parallel (an example for this situation can be found in Fig. 2b for LDOF, where the avgdist and pairdist functions can be computed in parallel).

From the dependency graphs depicted in Fig. 2 it can be clearly seen that LoOP extends Simplified-LOF with the normalization step, while the similarity of LDOF to LOF is mostly in using the kNN and the frac function (as elaborated in Sect. 4.4). In Simplified-LOF for example, the frac function depends on the results of both the first and second model function (which in turn depended on the first function). Since the

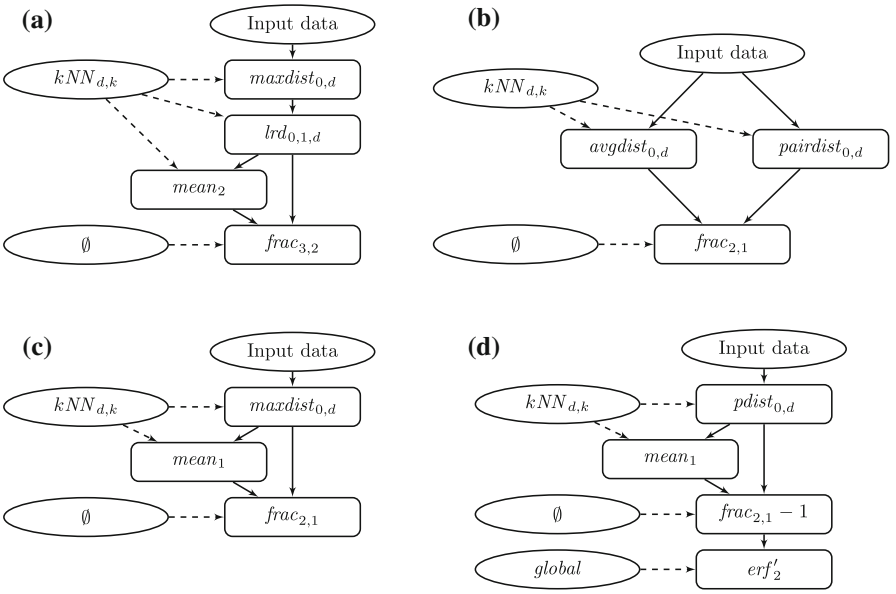


Fig. 2 Graphical dependency graphs for LOF variations. The relationship of LOF, Simplified-LOF and LoOP are easy to recognize, while LDOF differs much more. **a** LOF, **b** LDOF, **c** Simplified-LOF, **d** LoOP

fraction however has an empty set, when implementing this method we can compute the second and third model function in a single iteration over the data, without the need to store the intermediate result of the second model function. We can unfortunately not apply the same trick to the first model function: the second function needs its value for more than one object at a time. When we choose to combine the two functions into one, we will have to invoke the first function much more often (if the kNN contexts are precomputed, it of course is affordable to recompute the maximum function). This analysis shows us two different strategies for evaluating Simplified-LOF (and most similar methods): If we precompute the kNN contexts, we can essentially compute the LOF outlier score in a single pass over the dataset. If we cannot afford the precomputation due to storage costs but are able to store the intermediate results of the model functions, we can still avoid extensive kNN computations. Only when we have extremely little memory available, we may need to compute the kNNs of kNN by evaluating the whole outlier model for each point independently.

Another analysis that we can perform on this graph is to determine a notion of locality complexity of an outlier detection model:

Definition 36 (*Order of Locality of a Path*)
Given a Generalized Outlier Detection Model $[(f_0, c_0), \ldots, (f_n, c_n)]$ and its dependency graph $G = \{(i \leftarrow j)\}$. Then the order of locality of a path $L(a_k \leftarrow a_{k-1} \leftarrow \ldots a_1)$ with $a_i \in \{1 \ldots n\}$ is defined as:

$$L(a_j) := \begin{cases} 0 & \text{iff } c_{a_j} \in \{\emptyset, \text{ global}\} \\ 1 & \text{otherwise} \end{cases}$$
$$L\left(a_k \leftarrow a_{k-1} \leftarrow \ldots a_1\right) := L\left(a_k\right) + L\left(a_{k-1} \leftarrow \ldots a_1\right)$$

Definition 37 (*Order of Locality of a Model*)
Given a Generalized Outlier Detection Model $[(f_0, c_0), \ldots, (f_n, c_n)]$ and its dependency graph $G = \{(i \leftarrow j)\}$, the **Order of Locality** of the model is defined as the maximum order of locality of all paths in the model.

Based on this, we can obtain the order of locality of the analyzed outlier models. We give some examples here:

Proposition 1 (Order of Locality of kNN Outlier)
The order of locality *of the kNN outlier model* (*Ramaswamy et al. 2000*) *is 1.*

Proof Proposition 1 follows directly from Definitions 37 and 11. □

Proposition 2 (Order of Locality of LDOF)
The order of locality *of the LDOF outlier model* (*Zhang et al. 2009*) *is 1.*

Proof Proposition 2 follows directly from Definitions 37 and 20. □

Proposition 3 (Order of Locality of Simplified-LOF)
The order of locality *of the Simplified-LOF outlier model* (*Definition 18*) *is 2.*

Proof Proposition 3 follows directly from Definitions 37 and 18. □

Proposition 4 (Order of Locality of LoOP)
The order of locality *of the LoOP outlier models* (*Kriegel et al. 2009a*) *is 2.*

Proof Proposition 4 follows directly from Definitions 37 and 24. □

Proposition 5 (Order of Locality of LOF)
The order of locality *of the LOF outlier model* (*Breunig et al. 2000*) *is 3.*

Proof Proposition 5 follows directly from Definitions 37 and 17. □

These findings align with our intuition that LOF takes locality more into account than the other models, and reflects the simplification of Simplified-LOF. This bears repercussions for the order of locality of all outlier models that pretend to be a variant of LOF but actually are based on Simplified-LOF (as discussed in Sect. 4.4).

Let us note that these findings do not state any superiority of LOF. We do not imply that "more" locality is "better" in any way. But we state that methods of different orders of locality model outliers in truly different ways. This should be recognized and taken into account when using these models (be it for applications or as role models for adapted outlier detection models). In the following, we show application scenarios, where the notion of locality is adapted to make basic outlier models suitable for complex data.

5 Locality and spatial outliers

Spatial outlier detection has grown as a field of its own interest over several years (Anselin 1995; Shekhar et al. 2003; Lu et al. 2003; Kou et al. 2006; Sun and Chawla 2004; Chawla and Sun 2006; Liu et al. 2010; Chen et al. 2010). A key result of Anselin (1995) is the generalization of the global spatial association statistics Moran's I (1950) and Geary's C (1954) to individual contributions denoted as the local Moran I_i and local Geary C_i, then using a statistical test to identify strong contributions. Additionally the Moran scatterplot was introduced, which plots the locally standardized attribute value against the globally standardized attribute. In this plot, objects close to the regression line indicate consistency with the trend, whereas objects in the upper left and bottom right areas appear different on local and global scales. This concept of comparing local with global scores can be found in many newer methods in slight variations, others however just use the local scores proposed here directly or with only slight modifications.

The idea is to separate spatial attributes from other attributes, compute the neighborhood wrt the spatial attributes solely but compare the non-spatial attributes only to derive a notion of outlierness. Most of these methods use a local neighborhood based on the spatial attributes solely in order to extract a score (the simplest type of model) for the object using the non-spatial attributes only. Many methods can just process a single non-spatial attribute, and there are rather few methods that use a model more complex than a preliminary score or a non-trivial comparison step, but we will highlight some examples in this section.

With our framework, this notion of spatial outliers can be implemented quite easily and straightforward. Actually, even some methods that at first appear to follow an entirely different approach can be rewritten to follow this schema, sometimes even significantly improving both the understanding and performance of the algorithm.

5.1 Data

For the experiments we used the US Census data,[2] which are available at various granularities. For the population data, we used the Census County Subdivisions level, which often subdivides counties in particular in densely populated areas resulting in 36,351 objects. The partitioning is based on administrative regions which occasionally do form a grid but also often follow geographical boundaries such as rivers. We analyzed the ethnic group attributes consisting of five dimensions representing ethnic groups in the population that approximately sum up to 1.0 (people may belong to multiple ethnic groups). There are many more attributes available, but these may require specialized algorithms and extra normalization; five-dimensional normal vectors is a data type algorithms can be expected to work well with. Since some algorithms are not performing that well and may only be able to analyze single attributes, we additionally used a dataset representing land use homogeneity at a county level containing 3,109

[2] Available at the US Census Bureau, United States Department of Commerce. http://www.census.gov/.

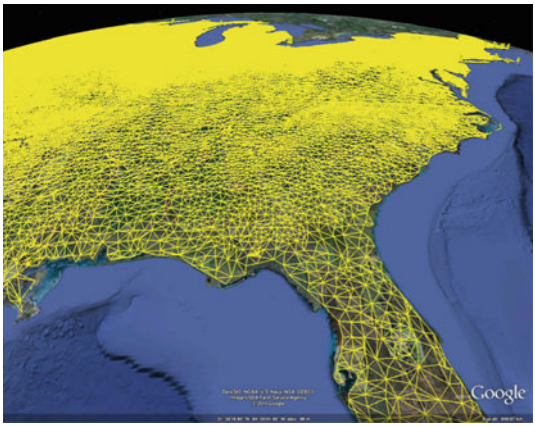


Fig. 3 Neighborhood graph in Google Earth

Background © 2011 Google.

records derived from satellite imaging.[3] For the third dataset we use unemployment rates in Germany[4] again at a county level. The dataset has a single dimension, 412 records ranging from 2.2 (full employment) to 18.3 with a median value of 7.0 and a skewed distribution.

5.2 Neighborhood in spatial outlier detection

Most algorithms use the k nearest neighbors in the spatial attributes as neighborhood set, because of the uniform size. When polygon data are available, polygon intersection and shared borders (also known as the "meets" relation, in line with Allen's Interval Algebra) are obvious choices, that may however result in objects such as islands not having any neighbors at all, or just a single neighbor when e.g. a city is contained within a single county polygon. Rarely, more complex notions of neighborhood are encountered such as Voronoi cells as proposed e.g. by Liu et al. (2010). In the latter two cases, it can be desirable to expand the neighborhood to the t-fold transitive of the original relation to obtain a sensible neighborhood size. Figure 3 shows a one-neighborhood graph for the high resolution census dataset. Neighborhoods can for example be defined using length-limited paths in this graph. Occasionally, measures obtained during computation of the neighborhood (such as shared border length and distance) can be reused for weighting in the model generation phase.

Often, the definition of the actual neighborhood is part of a proposed method. It is, however, obvious that the choice of a concrete method for neighborhood computation will have a huge impact on the results of any spatial data mining method and hence should be considered independently from the algorithmic approach and the outlier model. In the following, we discuss the properties of models and comparison methods as well as the impact of using a local or a global context and reference irrespective of the

[3] Available at the Arizona State University GeoDa Center, http://geodacenter.asu.edu/.

[4] Available at Statistisches Bundesamt Deutschland, http://www.destatis.de.

definition of spatial neighborhood (we use the same definition of spatial neighborhood for all methods in the evaluation). For the US Census data, neighbors are defined using polygon adjacency expanded to five steps for Census County Subdivisions (which often are very fine-grained) and three steps for county level. For the dataset on Germany, we also used three steps at a county level.

Neighborhood computations, all compared methods, and our generalized and adapted methods are implemented in the ELKI framework (Achtert et al. 2011, 2012).

5.3 Models in spatial outlier detection

The common model used in spatial outlier detection is a deviation score, where the non-spatial attribute value(s) of an object are compared to the attributes of the neighbors. For example, Shekhar et al. (2003), Lu et al. (2003) propose various simple statistics such as the quotient of the attribute value and the neighbors' attribute mean, the difference of the attribute value to this mean, and the standardized deviation of the attribute value from the neighbors' attribute median. Kou et al. (2006) extend the model to the difference from a weighted mean normalized as z-score and the weighted mean difference (without normalization). However, the model is then (in some of the proposed methods) iteratively refined by replacing attribute values with their expected value, making the result of the algorithm much less predictable. As such, there is no formal definition of what actually constitutes an outlier for these methods, except the given algorithm.

In other cases, the model is less obvious: POD (Kou et al. 2007) is described as a graph-based method. For each object, the k nearest neighbors are determined, which are used as edges of a graph. The weight of each edge is based on the similarity of the non-spatial attribute. The edges are managed in a global priority queue, and are successively removed until some objects become isolated, which are returned as outliers. The complexity of the algorithm is given as $\mathcal{O}(kn \log(kn))$ due to managing the priority queue of size kn.

However, this algorithm can be transformed into our framework easily. In essence, the edges are processed by their length, longest first. An object becomes an outlier, when the last edge is removed. The other edges essentially do not play any role, and do not need to be managed. Therefore, we chose to model each object by its shortest edge—a very simple local score—and obtain the same ranking. Instead of managing the complete priority queue, we can directly compute the length at which an object becomes an outlier. By comparing these lengths globally we obtain the identical ranking, and the top m results equal those of POD. The runtime however is reduced to $\mathcal{O}(n \log n)$, dominated by performing one kNN query for each object. Most of the outlier detection algorithms have this complexity, since the model generation and comparison steps usually are much faster than a kNN query.

These examples show how using the framework allows for a deeper understanding of the algorithms' actual results (including a more formal definition of what constitutes an outlier) as well as a canonical way of computing the results that is usually cheaper than computing the neighborhoods.

Table 4 Locality and models in spatial outlier detection

Method	Context Reference	Model function Comparison function Normalization (optional)	Notes
Local Moran (Anselin 1995)	Matrix	$\sum_{\mathcal{N}} \omega_n \pi(p)\pi(n)$	Spatial autocorrelation
	Global	With $\sum \pi(p)\pi(n)$ z-Score	
Local Geary (Anselin 1995)	Matrix	$\sum_{\mathcal{N}} \omega_n (\pi(p) - \pi(n))^2$	Spatial autocorrelation
	Global	With $\sum (\pi(p) - \pi(n))^2$ z-score	
z-Statistic (Shekhar et al. 2003)	kNN	$\pi(p) - E_{\mathcal{N}}[\pi(n)]$	
	Global	Threshold $\lvert z\text{-score}\rvert$	
Iterative r (Lu et al. 2003)	kNN	$\pi(p)/E_{\mathcal{N}}[\pi(n)]$ or inverse	Iteratively updated: $\pi(p) = E(\pi(n))$
	Global	Threshold	
Iterative z (Lu et al. 2003)	kNN	$\pi(p) - E_{\mathcal{N}}[\pi(n)]$	Iteratively updated: $\pi(p) = E(\pi(n))$
	Global	Threshold $\lvert z\text{-score}\rvert$	
Median (Lu et al. 2003)	kNN	$\pi(p) - \text{med}_{\mathcal{N}}[\pi(n)]$	
	Global	Absolute descending $\lvert z\text{-score}\rvert$	
Weighted z (Kou et al. 2006)	kNN	$\pi(p) - \sum_{\mathcal{N}} \omega_n \pi(n)$	
	Global	Absolute descending $\lvert z\text{-score}\rvert$	
Average difference (Kou et al. 2006)	kNN	$\sum_{\mathcal{N}} \omega_n \lvert\pi(p) - \pi(n)\rvert$	
	Global	Descending	
POD (Kou et al. 2007)	kNN	$\min_{\mathcal{N}}\{\lvert\pi(p) - \pi(n)\rvert\}$	Inefficient graph-based algorithm
	Global	Top-k Label	
SLOM (Chawla and Sun 2006)	Spatial	Trimmed mean distance $\hat{d}$	Largest distance ignored
	Spatial	$\frac{\hat{d}}{\text{mean}(\hat{d})+1} * \beta$	β oscillation measure
	Global	Descending	

In Table 4, we give a summarized overview of the models used by various spatial outlier detection methods proposed. We use a number of shorthand notions for brevity such as $\pi(p)$ for the non-spatial projection of an object, ω_n for a weight assigned to a neighbor, $E_{\mathcal{N}}(\ldots)$ for the expected value (usually the mean), and z-score, i.e., $z(x) := \frac{x-\mu_X}{\sigma_X}$, for the standard score normalization that assumes a normal distribution.

It is fairly obvious that most methods compute a simple local statistic such as the mean deviation, then try to normalize and compare these values on a global level out of necessity. However, they often lack motivation for their choices, and the methods in

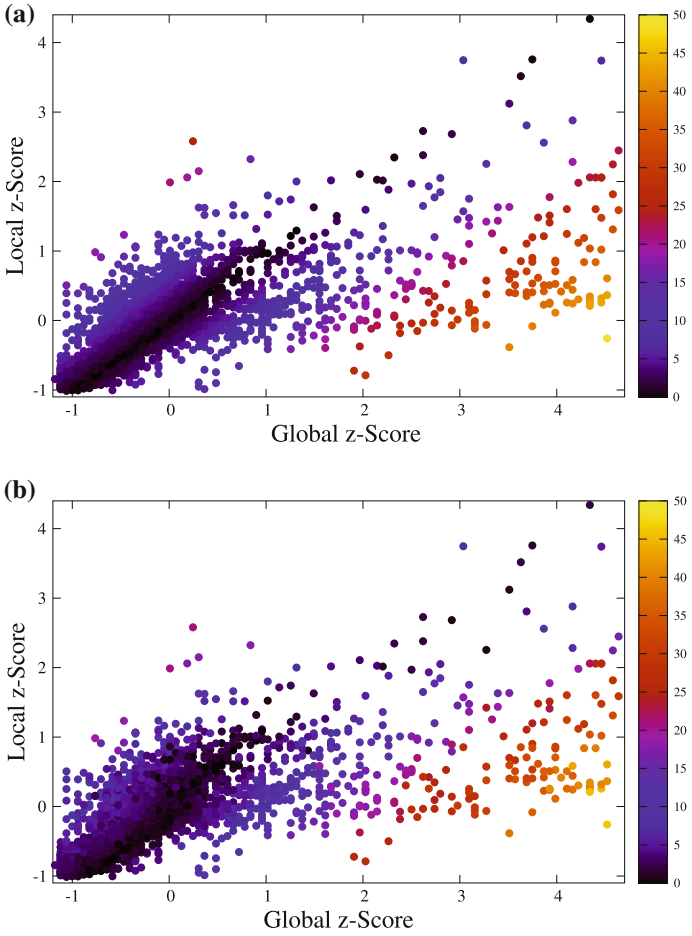


Fig. 4 Moran scatterplots with outlier scores for US Census land use data—basic methods. **a** *Z*-statistic (Anselin 1995; Shekhar et al. 2003), **b** median (Lu et al. 2003)

total offer little beyond the classic work of Anselin (1995). The similarities between the methods have not been properly studied before and the users are essentially facing a variety of algorithms without indication of which is the most appropriate for them. The evaluation which just lists a number of example outliers detected is not very helpful here either, especially since many of these outliers may well be global outliers as well (such as Soldier Summit, Utah, which is a ghost town and thus obviously has uncommon rent and population values already on a global level).

5.4 Experimental comparison of spatial outlier scores

In Fig. 4, we visualize the results for z-statistic-based methods (comparing local and global z-scores) of Anselin (1995), Shekhar et al. (2003), Lu et al. (2003) in relation to

the raw z-scores in a Moran scatterplot. The x axis is the global z-score, the y axis the local z-score. Globally unusual objects are on the left and right side, locally unusual objects (by their z-score) are on the top and bottom. The color is determined by the outlier score. The results of both methods are closely related and are essentially the deviation from the diagonal. The imbalanced nature of the actual data distribution—which apparently is not normal but skewed—seems to affect the algorithm performance. Figure 5 contains some of the more advanced methods, including SLOM (Sun and Chawla 2004; Chawla and Sun 2006) but also, using our framework, the canonical adaptations of LOF (Breunig et al. 2000) and LDOF (Zhang et al. 2009) to the spatial domain. The results differ much more from the z-score than the median-based method in Fig. 4b, which produced mostly the same results as the original z-score. SLOM results here show noise in the range of low scores from 0 to 0.2, and no objects are assigned a particularly high score. The adaptations of the traditional outlier detection methods offer a much better contrast and seem to find more interesting outliers, such as the objects around $(0.4, -1)$ that are slightly above the global mean, but around one standard deviation below their neighbors mean. By definition, LOF and LDOF only detect outliers that are towards the bottom right area, but can trivially be adapted to detecting the type of outliers in the upper left or both by using the inverse ratio.

Figure 6 compares different algorithms on this dataset, assessing the correlation between the resulting outlier scores. The relationship of LOF and LDOF is surprisingly linear, while z-statistic and LOF diverge much more. SLOM interestingly seems to differ mostly from z-statistic for low scores. These probably are the points in the upper left area of the Moran scatterplots that are not outliers by the LOF-style definition used.

In summary, we see here that the implementation of a spatial neighborhood does not make a method local in the strict sense of using locality not only for model building but also for model comparison. Making spatial outlier detection truly local remains as a possible improvement for a broad range of existing methods.

In Fig. 7, we show some example analysis of the unemployment rates for 2009 in Germany on a county level. The classic z-statistic is not very convincing on this dataset. It detects only two strong (both of which are global outliers) and some less strong outliers that cannot be easily explained. SLOM performs better and detects some additional outliers. However, as with z-statistic, only two of them are significant: the harbor city Bremerhafen next to Bremen, and the small city of Pirmasens close to France, where the ongoing demise of the shoe producing industries along with the closing down of a US military base have caused the unemployment rates to skyrocket. The LOF adaptation in our framework performs very well, detecting much more interesting additional outliers (and all of the outliers mentioned above). For example, it detects the city of Munich as an outlier. While the unemployment rate for Munich was just 6 %—an excellent value for a big city, the median in this dataset is 7 %—this is twice as much as that of the surrounding counties. This is a prime example of a local outlier, with a very normal value on the global scale, but a strong divergence compared to the local neighbors. It is this kind of outliers that we expected all the algorithms to discover.

Figure 8 gives a detail view of the outliers detected for Bavaria. Most city regions here show up as outliers, which is correct as they usually have a significantly higher unemployment rate than the rural areas in Bavaria (largely because unemployed people

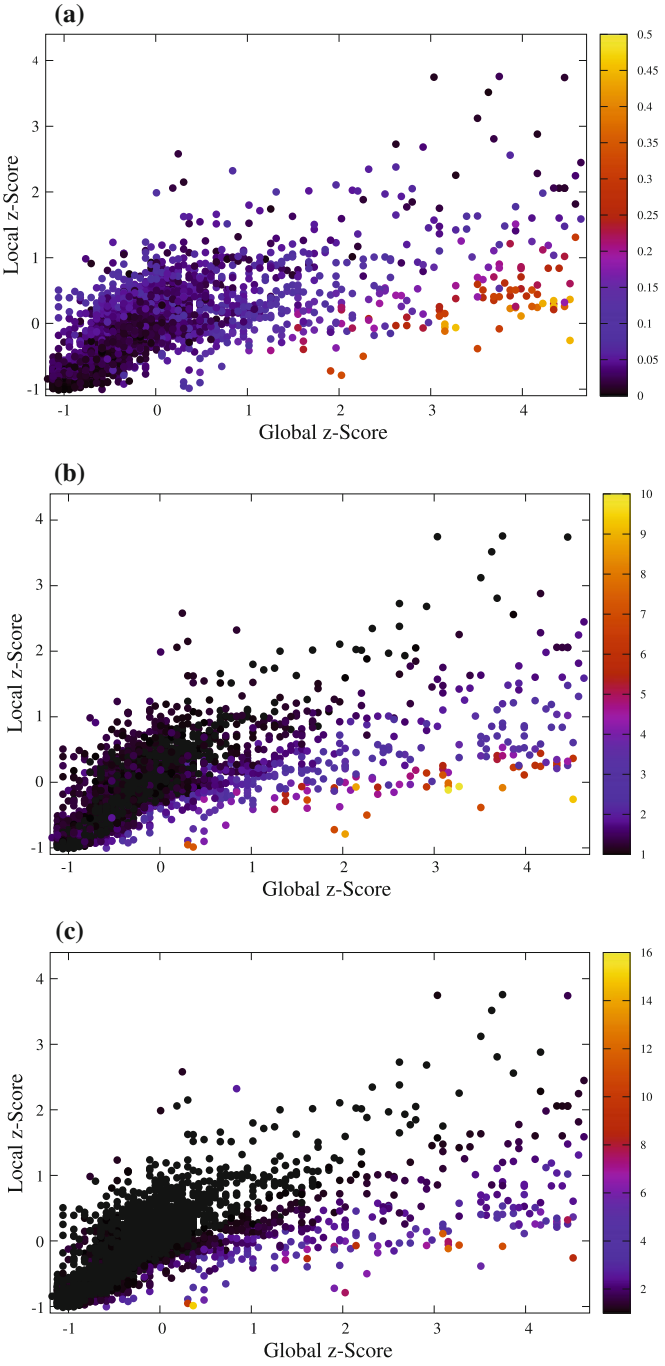


Fig. 5 Moran scatterplots with outlier scores for US Census land use data—advanced methods. **a** SLOM (Sun and Chawla 2004; Chawla and Sun 2006), **b** LOF (Breunig et al. 2000) adaptation, **c** LDOF (Zhang et al. 2009) adaptation

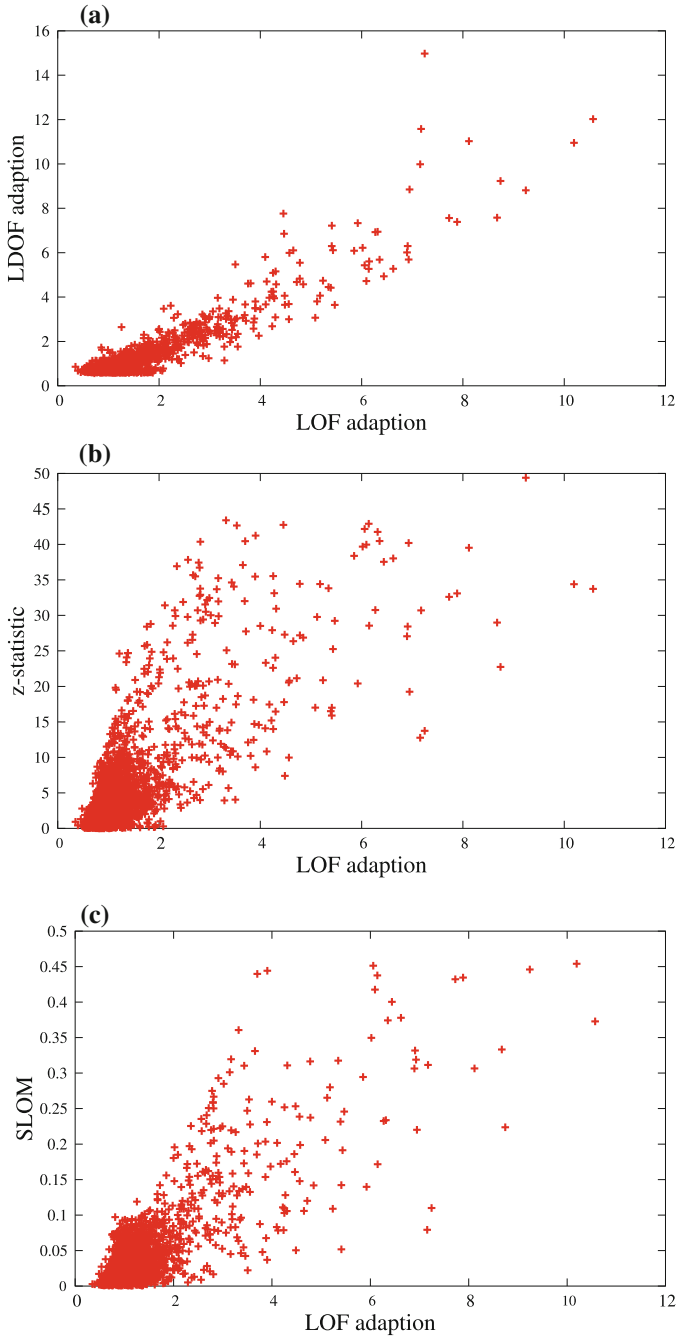
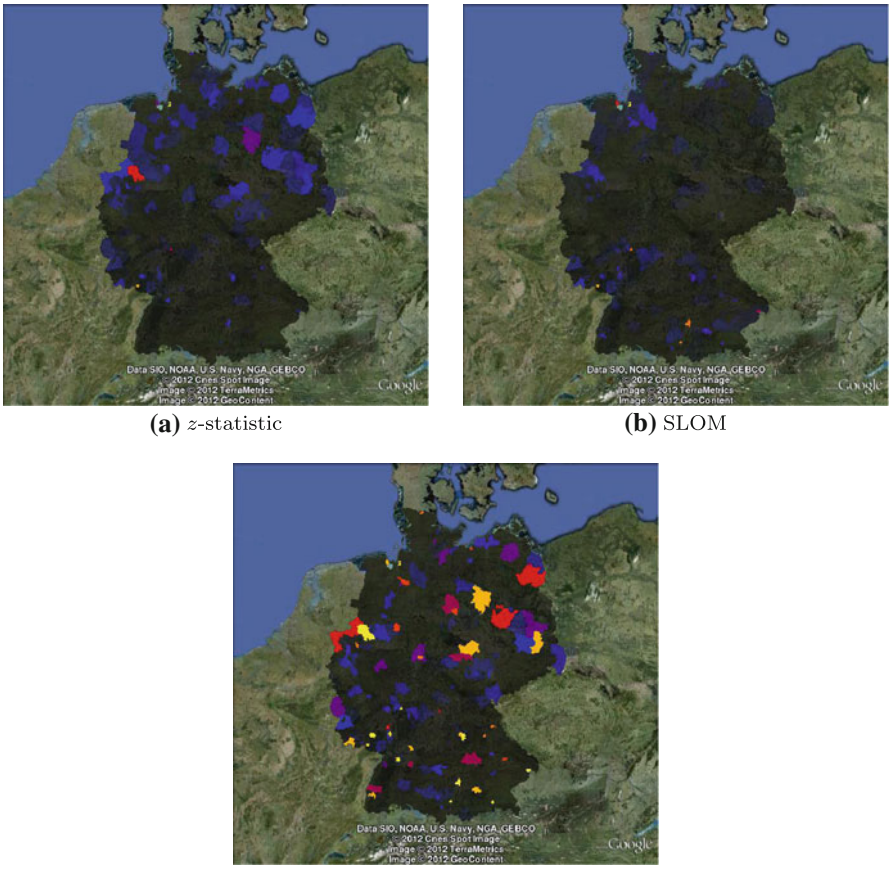


Fig. 6 Comparing different methods. **a** LOF versus LDOF, **b** LOF versus z-statistic, **c** LOF versus SLOM



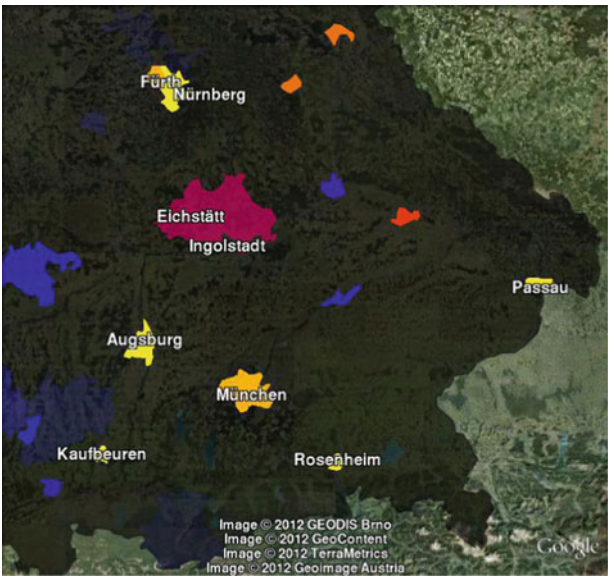
(a) z-statistic **(b)** SLOM

(c) LOF adaptation

Background © 2012 Cnes/Spot Image, TerraMetrics, GeoContent, Google

Fig. 7 Outlier scores in unemployment rates in Germany. **a** *Z*-statistic, **b** SLOM, **c** LOF adaptation

tend to move to the cities where much more new job opportunities are created, while people with safe jobs tend to buy houses in the rural areas outside of the cities). But there also is an interesting case where this rule of thumb does not hold: the city of Ingolstadt is with 4.8 % rather close to the Bavarian average of 4.5 %, and indeed it does not achieve a high outlier score. The region of Eichstätt—a much larger, rural area not far north of Ingolstadt—however has so-called full employment with just 2.2 % unemployment rate (and as many open positions as unemployed people). The reason for this excellent score however is located in Ingolstadt: the car manufacturer Audi is doing very well and hiring many people, which can afford to move to nice homes outside of the city in Eichstätt. So while this gives Ingolstadt a score typical for this area of Bavaria, it brings Eichstätt to this unusually low score. Note that none of the other methods managed to rank Eichstätt highly. The observed trend of cities showing up as outliers does not hold for all of Germany. In Eastern Germany, Dresden and



Background © 2012 GEODIS Brno, GeoContent, TerraMetrics, Geoimage Austria, Google

Fig. 8 Bavaria detail for LOF adaptation

Chemnitz for example are well-aligned with the surrounding areas at around 12 %. Hannover also is with 9.3 % just slightly higher than the surroundings at around 8.5 %.

Overall, we see again that the concept of locality (that does not necessarily come along with selecting spatial neighbors for model building) can considerably improve the results of outlier detection.

5.5 SLOM as a special case of local outlier

To return to our formal analysis, we also show the dependency graph for the spatial LOF adaptation in Fig. 9. Compared to Fig. 2a the only differences are that we are using spatial neighbors as reference set, and we added a normalization step from Kriegel et al. (2011) using a Gamma normalization. A similar normalization was also added to the comparison method for visualization purposes.

This adaptation of the classic LOF method to spatial data is leading to interesting observations on outlier rates, while the raw numbers just depicted the well known difference between Western and Eastern Germany and that the southern states are doing much better economically. The other spatial outlier detection methods primarily produced the counties with the highest unemployment rates, which are global outliers.

To study the impact of the use of locality in the reference set for model comparison, we design two baseline methods: For a first, simple method, we measure the deviation from the mean vector of the neighbors and compare this on a global level. For the advanced method, we add the second type of locality and subtract the mean deviation

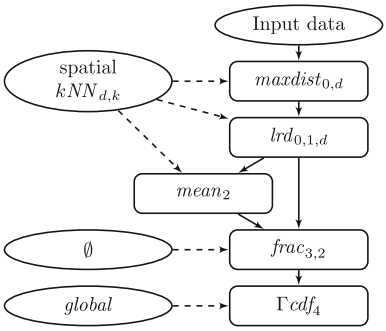


Fig. 9 Spatial adaptation of LOF, including normalization

Table 5 Spatial outlier as special case of local outlier

Method	Context Reference	Model Comparison Normalization
SLOM (Chawla and Sun 2006)	Spatial	Trimmed mean distance $\hat{d}$
	Spatial	$\frac{\hat{d}}{\text{mean}(\hat{d})+1} * \beta$
	Global	Descending
Simple spatial	Spatial	Deviation from mean vector
	Global	Descending Erf
Advanced spatial	Spatial	Deviation from mean vector
	Spatial	Deviation from mean deviation Erf

of the neighbors.The setup is also detailed in Table 5 and their dependency graphs are shown in Fig. 10. As a reference, we compare these two straightforward formulations of the general framework for spatial data with SLOM (Sun and Chawla 2004; Chawla and Sun 2006), one of the more renowned approaches in this family which analyzes the spatial neighborhood beyond a simple statistic by computing the so-called oscillating parameter β, which grows when the local distance distribution is skewed and the mean is not central. Figure 11 is the dependency graph visualization of this method. In consists of a trimmed-mean average distance (the maximum distance is not included) denoted as $\hat{d}$ and the aforementioned stability parameter β. These functions are given as pseudocodes in the SLOM article, and for some of the terms in the formulas little reason is given, except, for example, to avoid division by 0.

We compare the results of these baseline methods and SLOM on the US Census ethnic groups data (using five attributes), as visualized in Fig. 12 using a Google Earth overlay. While the strongest outliers are not affected much by using the locality in model comparison, it clearly stabilizes the results in the Mississippi delta area and improves contrast in general. The contrast of outlier scores in our method is also a lot better than in SLOM, where the highest achieved score is just 0.91, and we had to boost the contrast manually for the visualization.

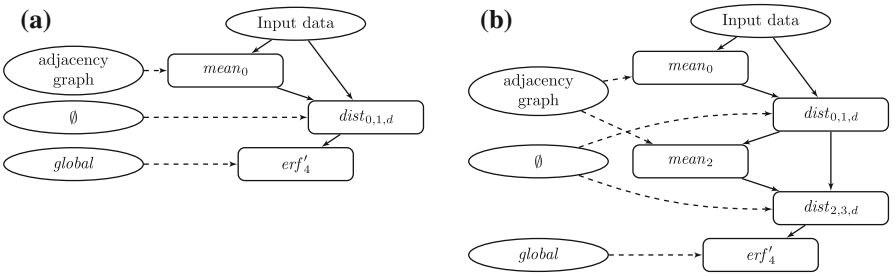


Fig. 10 Dependency graphs for baseline spatial outlier detection methods. **a** First order deviation outlier, **b** second order deviation outlier

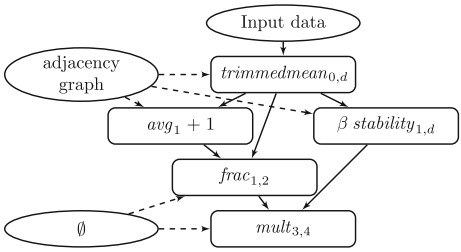


Fig. 11 Dependency graph for SLOM

The main outliers detected by all three methods are comparable, and not very surprising. They usually fall into one of three categories: Indian reservations, invalid or incomplete data (there are records with no inhabitants), and small towns. Larger cities are usually not considered outliers. Instead they are split into multiple Census subdivisions that in turn are similar. Only occasionally, one of these subdivisions is recognized as being dissimilar, e.g., when there is a large (and local) Asian community. While the Indian reservations can be seen as global outliers and are recognized easily on a global level, outliers such as the "Space Center CCD" in Florida are not atypical on a global scale. However it is a local spatial outlier, since it contains a wide mixture of ethnic groups while it is located in a predominantly Caucasian area of Florida.

To summarize: we basically reproduce the results of SLOM with a rather simple and efficient setting (advanced method) in terms of the identified outliers and we even improved the stability and contrast of the actual outlier scores. However, the model constructed is a simple two-stage "difference from mean" approach that is very straightforward and easily comprehensible compared to the SLOM pseudocode. Regarding the order of locality, we also see the impact of using second order locality where the advanced method produces much more stable and useful results than the simple (first order) method. As well, we see the impact of third order of locality (the spatial adaptation of LOF) over the second order of locality used in SLOM.

5.6 Univariate versus multivariate outlier analysis

In comparison with recent non-spatial outlier detection algorithms, what is noticeable about spatial outlier detection algorithms is that they are in most cases only able to perform a univariate, not a multivariate statistical analysis of outliers. Most approaches

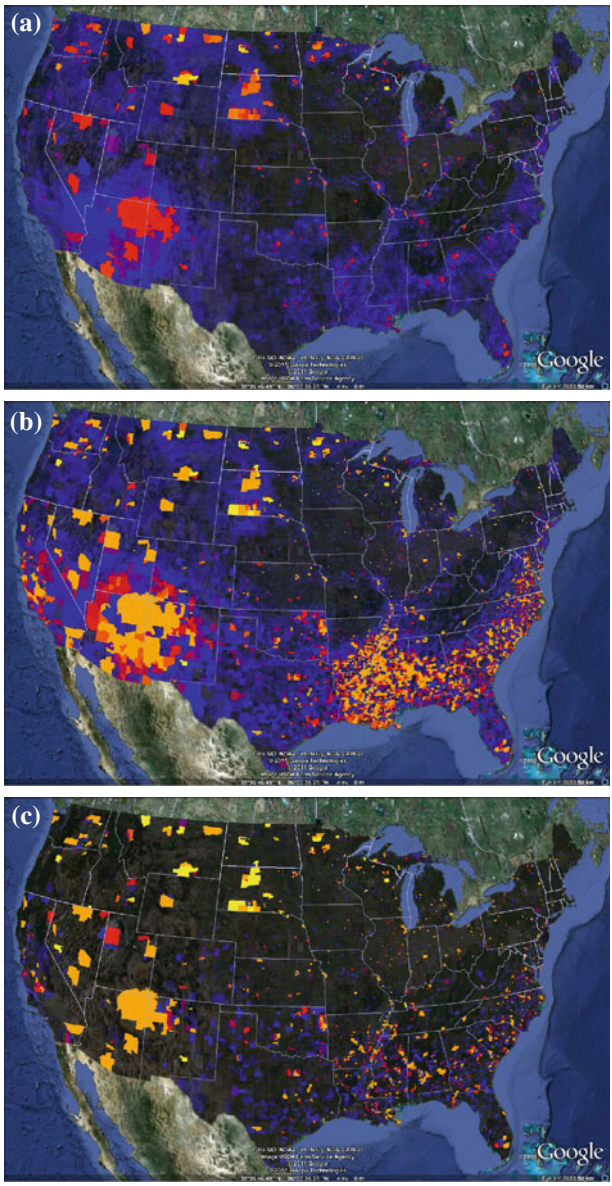


Fig. 12 Spatial outliers on US Census data on five-dimensional ethnic groups. *Bright colors* indicate outliers, *dark colors* inliers. **a** SLOM (Sun and Chawla 2004; Chawla and Sun 2006), **b** simple method, **c** advanced method (Color figure online)

focus on data that consist, aside from the spatial coordinates, of a single non-spatial attribute. For many methods it is obvious or conceded that they are only capable of performing a univariate analysis (e.g. Kou et al. 2006, 2007; Shekhar et al. 2003; Lu et al. 2003). Furthermore, the univariate analysis used in most of these methods is rather simple and is covered by classic statistics as discussed in Sect. 2. Only some

methods (Sun and Chawla 2004; Chawla and Sun 2006) are obviously applicable in the multivariate case though they do not care to demonstrate or even to state that they are.

Let us finally point out that the application of our general framework to the task of spatial outlier detection allows to transfer all the achievements of multivariate outlier detection given in traditional outlier detection research since it is easy to apply a traditional model to spatial data as a special case (as we demonstrated) by means of an adapted notion of locality.

6 Locality in video streams

The multimedia community analyzing video sequences is interested in a lot of different questions like, e.g., key-frame extraction and storyboard display (Money and Agius 2008; Kim and Kim 2010), shot or scene change detection (Hong et al. 2003; Lee et al. 2004), or detection of continuity errors (Pickup and Zisserman 2009). In this application example, we do not aim at covering such a wide range of goals but just to demonstrate the flexibility and usability of the general outlier detection framework to adapt to highly specialized tasks. To this end, we examine video sequences, defining for each video frame the previous 12 frames (0.5 s) both as local context and as reference set as you would do in a streaming context. Frame similarity is measured using HSB color histograms with quadratic form distance to capture color similarity (texture and edge features are not used for this experiment).

From the local context, the root mean square distance (RMSD) to the previous frames is computed:

Definition 38 (*Root-Mean-Squared-Deviation Model Function*)

$$\mathrm{RMSD}_{i,d}(o,\ c(o),\ \mathfrak{D}) := \sqrt{\mathrm{mean}_{n\in c(o)} d\left(D_i(o),\ D_i(n)\right)^2}$$

This model function is meant to capture "image instability" and is visualized in the first row of Fig. 13.

Our intended video outlier definition is based on a sudden increase of instability. This can now be modeled using two simple additional model functions: the first computes the maximum over the context set, the second one the increase over this maximum. These two can trivially be combined into one model function, we chose however to split them to show reusability of components. Finally, we also add a normalization using *erf*.

Definition 39 (*Maximum Model Function*)

$$\mathrm{max}_i(o,\ c(o),\ \mathfrak{D}) := \max_{n\in c(o)} D_i(n)$$

Definition 40 (*Increase Model Function*)

$$\mathrm{inc}_{i,j}(o,\ c(o),\ \mathfrak{D}) := \max\left\{0,\ D_i(n) - D_j(n)\right\}$$

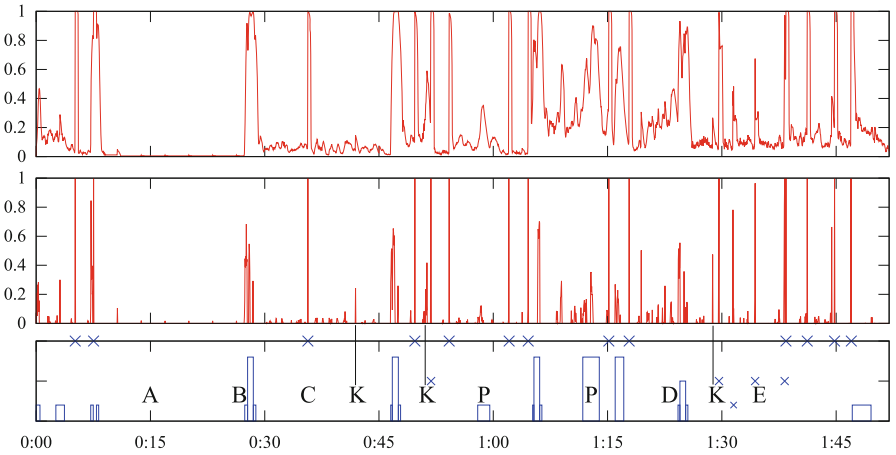


Fig. 13 Outlier scores in a news clip. The *first plot* uses the RMSD directly, the *second plot* uses the increase in RMS. In the *third plot*, *crosses* indicate interesting events, while *boxes* indicate transitions of varying significance

Frames captured from the news video clip at outlier locations B, C, D, and E.
© 2011 ARD

Fig. 14 Frames captured from the news video clip at outlier locations B, C, D, and E.

Definition 41 (*Video Outlier Detection*)
Video outlier detection is a generalized outlier detection model for distance function d and neighborhood size k with

$$\text{VideoOutlier}(d,\ k) = \left[\left(\text{RMSD}_{0,d},\ \text{prev}_k\right),\ \left(\max_1,\ \text{prev}_k\right),\ \left(\text{inc}_{1,2},\ \emptyset\right),\ \left(\text{erf}_3,\ \text{global}\right)\right]$$

The advanced method is adaptive to very different situations. The results were not refined and could further be improved by locating local maxima only. However, this naïve approach was surprisingly effective, in particular since the only parameter is the size of the time window used.

Figure 13 consists of three rows. The first row shows the local RMSD, the second row shows the derived outlier scores. The third row contains human-made annotations to the video that indicate single-frame events (crosses) and multi-frame transitions (boxes) with varying severity. The news clip is from a public television news summary,

consisting of various short scenes. Most of the annotated events capture switches between different news items, sometimes with transitions or the occasional fast camera pan. Let us discuss some of these events in detail. The region marked as *A* is actually a still image (a black and white archive picture of the movie director Bernd Eichinger who deceased 24 January 2011). When zoomed in closely, there is a periodic signal to be seen here, which is caused by the keyframes of the video stream compressions. Many of the non-annotated spikes for example at the locations marked with *K* can be attributed to image quality changes due to keyframes. The first *K* is in a press conference setting. At the last two marked locations, the keyframe event is clearly less significant than the true event shortly after. We marked some camera pans with a *P*. While they show a strong variance in the upper plot, they are not recognized as outliers in the second plot. These camera pans start slowly, therefore the increase in variance is only gradual. The three adjacent boxes labeled as *B* form a complex three-part transition. First the heading is removed, then the scene transition happens, then the new heading is added. Two frames of the main transition are shown in Fig. 14. *C* is a typical scene change within the news item, causing an abrupt change with only a "ghost image" in the first frame, a typical example of the top outliers detected. Two consecutive frames are shown in Fig. 14. A fast camera pan event with light changes can be seen at *D*. Additionally the high level of detail blurs with the video compression, and again keyframes show up as outliers within this high-activity region. The simpler variance detection shows the high overall variance, the second row shows that the improved method actually recognizes individual keyframes. Two frames less than a second apart are shown in Fig. 14. A true outlier we only annotated in the second pass is found at *E*. We had first only skipped through the clip to mark scene changes. But when investigating this outlier, there is a clear reason to acknowledge it: the scene is a press conference, and the detected outlier frame is caused by a photographers flash. There is a clear color change, but it is still less severe than in a scene change. Again, two consecutive frames are shown in Fig. 14.

7 Locality in network outliers

As another application example, we refer to the ideas and modeling of community outliers as presented by Gao et al. (2010). For their community outlier detection algorithm (CODA), they used EM to both learn community assignments and outliers in the dataset at the same time, whereas we focus on detecting the outliers directly within their neighborhood context. In the dataset based on a selection of conferences and journals listed in DBLP[5] data, the goal is to identify outlying conferences and authors.

For the selected conferences and journals, we extracted all publication titles, applied the tokenizer and stemmer from the text search engine software Xapian[6] and produced term frequency vectors. Orthogonally, we extracted all the participating authors, considering every author a term directly. Since DBLP data are normalized, we do not

[5] http://www.dblp.org/db/.

[6] Xapian search engine library, http://xapian.org/, GPL.

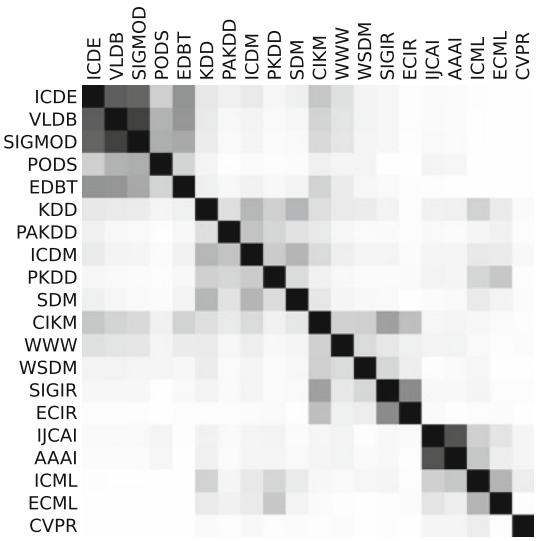


Fig. 15 Author similarity matrices for DBLP 20 data set

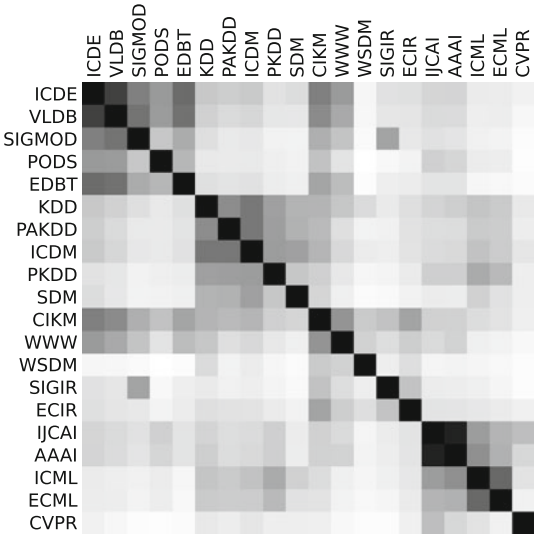


Fig. 16 Title similarity matrices for DBLP 20 data set

have to take care of different spellings ourselves. We used the common TF-IDF normalization to weight down common words and ubiquitous authors as well as cosine similarity to compute the similarities.

With the small data selection of 20 conferences (DBLP 20) as described by Gao et al. (2010), we were able to retrieve the same results as CODA: the conferences CVPR and CIKM were identified as the top outliers by a simple LOF-based approach. We use the same components as LOF, but this time only modify the context functions (using $r_1 =$ titles, $r_2 =$ authors):

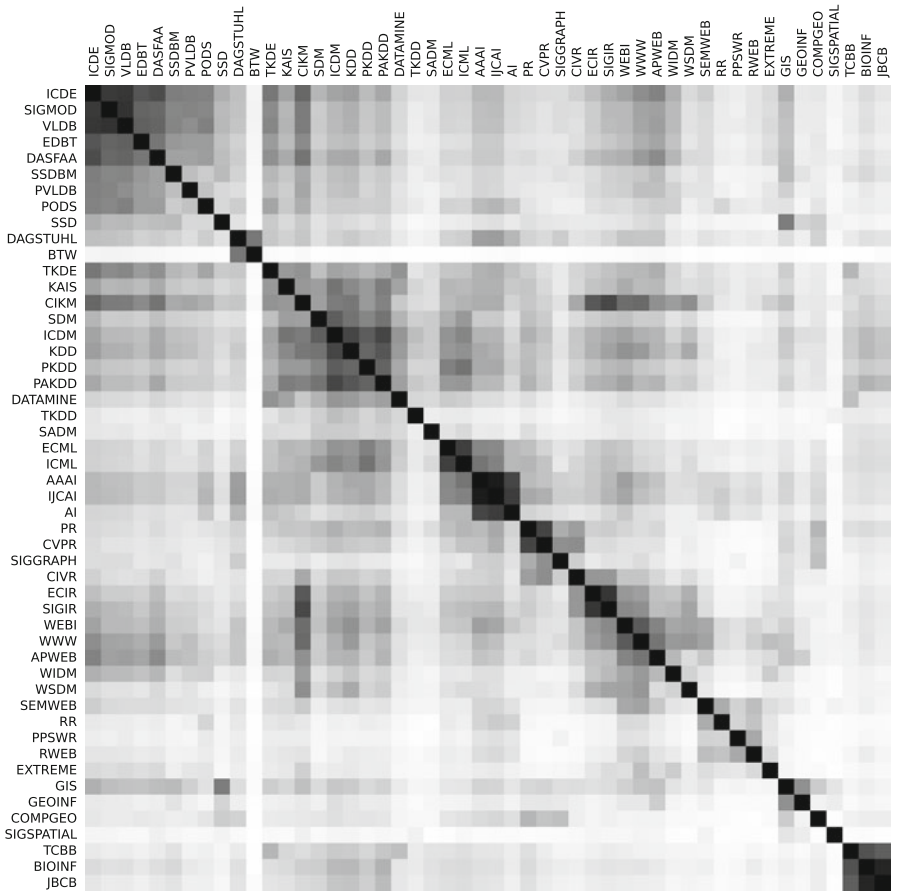


Fig. 17 Title similarity for DBLP 50 data set

Definition 42 (*Bipartite LOF*)
Bipartite LOF is a generalized outlier detection model for distance function d and neighborhood size k and two relations r_1 and r_2 with

$$\begin{aligned}\text{BipartiteLOF} = \big[&\left(\text{maxdist}_{0,d},\ r_1 k\text{NN}_{d,k}\right),\\ &\left(\text{lrd}_{0,1,d},\ r_1 k\text{NN}_{d,k}\right),\ \left(\text{mean}_2,\ r_2 k\text{NN}_{d,k}\right),\ (\text{frac}_3, 2,\ \emptyset)\big]\end{aligned}$$

The same outliers were already analyzed by Gao et al. (2010) as being community-outliers since both conferences bring together different research communities: CVPR draws on computer vision, artificial intelligence, and machine learning, while CIKM attracts data mining, information retrieval, and database people. Unlike the other conferences, it is hard to pin them down to one research community or the other. Figures 15 and 16 give the similarity matrices on authors and titles, respectively. The similarity matrices support the interpretation that CIKM is connected to multiple communities

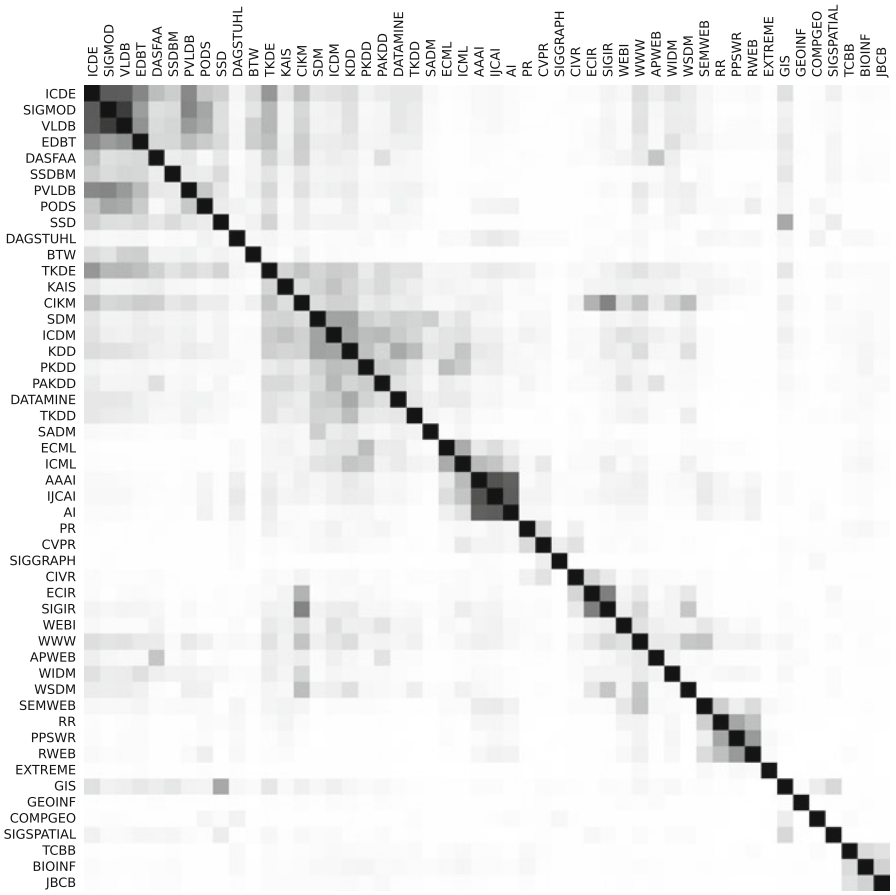


Fig. 18 Author similarity for DBLP 50 data set

both in titles and authors. CVPR is loosely connected on the titles similarity, but is even a global outlier when it comes to the authors involved.

Interestingly, CODA is not a local approach but learns globally c classes (using EM-clustering), where c is an important parameter. The effect of EM is that certain local structures are learned. This may be the reason why the results of CODA can be also retrieved with an approach that uses locality in the first place.

We also performed experiments on a broader data selection using 50 conferences and journals from DBLP (DBLP 50). The corresponding similarity matrices are depicted in Figs. 17 and 18. The difference between both ways to assess the similarity can be highlighted by BTW, a German conference that attracts German database researchers (hence showing a certain similarity level to international database conferences ICDE, SIGMOD, VLDB, EDBT in terms of common authors) but comprises many papers written in German (hence the low overall similarity in terms of title words). The interesting aspect of community outliers as discussed by Gao et al. (2010) is that both similarities are not used in isolation but complementary.

Table 6 Top-10 outliers in DBLP 50 dataset

Conference	Score	Notes
EXTREME	0.970	Small XML conference
DAGSTUHL	0.968	Multi-topic, multi-language
APWEB	0.965	Asia-Pacific community
BTW	0.903	German-language database community
WIDM	0.902	Web data mining
SSDBM	0.891	Sub-community
JBCB	0.875	Bioinformatics journal
TCBB	0.863	Bioinformatics journal
SADM	0.843	Data mining and statistics community
GEOINF.	0.818	Geo-informatics

Correspondingly, with our settings, BTW is not the utmost outlier although still a prominent one. The Top-10 community outliers based on a generalized local outlier approach using building blocks from LoOP (Kriegel et al. 2009a) (largely for the normalization benefits) are listed in Table 6. Interestingly, in this larger dataset, CIKM is not a prominent outlier any more. Including web-conferences and other special topics like bioinformatics and geo-spatial as well as a broader selection of IR conferences leads to more cross-community-links overall (as, e.g., the prominent outliers DAGSTUHL and SADM).

8 Conclusions

Much of the data mining research on outlier detection is interested in (A) gearing an existing outlier detection approach to some specialized domain or in (B) improving efficiency of existing outlier models by approximations or some algorithmic means. The literature interested in the second issue usually is not that interested in the meaning of "outlier" while in the first topic it usually is not easy to compare the meaning of "outlier" developed for a specialized area with outlier models established in different areas. Here, we identified a structure common to many different outlier detection approaches. Defining the appropriate framework, in turn, should support both issues (A) and (B) while being aware of the implemented semantics of the outlier model. Improvements in either area can more easily be combined with improvements in the other.

As essential building blocks or workflow elements, we find (i) deriving a (local) context for the definition of some model, (ii) the model building procedure, (iii) deriving a reference set for comparison of different models, (iv) the model comparison procedure, and (v) a normalization for improved comparability of outlier scores. We identified these elements in a couple of prominent outlier detection approaches which already allows to see quite different usage of "locality" in different "local" approaches.

Based on this observation, we provided a rigorous formalization of analysis and applied our formal analysis to a broad selection of outlier detection models from the literature to present all these models (that are usually hidden or rather defined implicitly only within some algorithmic reasoning) in a common formal framework.

This way, similarities and differences can be easily identified, such as different degrees of locality.

We demonstrated two directions of possible impact of such a generalization on data mining research:

First, we made the model of outlierness used by different algorithms (usually implicitly) comparable, independent of the algorithmic merits of different approaches. Most commonly, new outlier detection methods are evaluated in a way that demonstrates the superiority of the new algorithm in (surprisingly?) finding exactly those outliers that are (usually implicitly) defined by the algorithm itself. We hope to stimulate a non-exclusive agreement in the research community on some (possibly several quite different) models of outlierness to allow the evaluation of different algorithms wrt the approximation quality of the independently given model or wrt the efficiency in retrieving outliers as defined by the independently given model. With saying "non-exclusive", we want to point out, that, of course, new models are expected to emerge with new application areas or problems or even with just better understanding of the nature of allegedly known data. But for the same model, several algorithmic solutions may be possible. Hence, comparing different algorithmic techniques to enhance the efficiency, focusing on one outlier model (as recently presented by Orair et al. 2010 in an exemplary manner for DB-outliers), is complementary to our approach and makes perfectly sense as well.

Second, we used the identified framework to adapt in a simple and easy way to some example application scenarios, namely spatial data, video streams, and information networks. In each of these quite different areas, our framework established easily a meaningful baseline approach and facilitates the identification of outliers as well as their interpretation. This allows an easier assessment of the true merits of specialized approaches. Using a simple and straightforward setup of our framework, we were able to reproduce results of highly specialized state-of-the-art approaches on some example datasets.

Aside from these concrete directions of research, our results may also raise a question of possibly broad interest to the data mining community: How is a complex approach to a specialized problem justified if a simple, general approach efficiently retrieves the same result? Please note that by no means we want to suggest that the approaches e.g. of Gao et al. (2010) or of Sun and Chawla (2004), Chawla and Sun (2006) where not justified. These are recognized state-of-the-art approaches that we used as examples only and our findings (on some datasets only!) for these examples where we reproduced their results would not support such a statement. But we would like to point out that using a general outlier method (as it is easy to construct by means of our framework) to tackle special problems should probably serve as a reasonable baseline and specialized approaches should possibly elaborate on their outlier semantics as well as their justification or superiority in comparison with such a general yet efficient setting using established outlier models just adapted to the special problem at hand.

We actually see a big potential in transferring abstract outlier detection methodology to the application areas we sketched and, as future work, would like to explore the possibilities for outlier detection in these areas in more detail, such as the transfer of the lessons learned in abstract outlier detection methods to specialized application

areas like spatial data (e.g., wrt the mostly unattended problems of locality and multivariate outliers) and video data. While the formalization worked for our video outlier experiment, a focus on streaming data may for example bring new requirements such as stronger (e.g. real-time) restrictions on the model functions that we have not yet considered. On a more technical side, the framework we sketched can be put into software, probably in form of a rapid prototyping system for outlier detection. While the raw principles sketched are easy to translate into software they also bring along new considerations with respect to efficiency, memory management, and parallelization (considering both, multi-threading and distributed computing, as in MapReduce clusters).

Let us finally note, once again, that we do not propose this framework as a general method for outlier detection in order to prevent new approaches from being proposed. Far from that we hope, however, that informally identifying context, reference, model, and comparison functions (as well as normalization steps) as building blocks or even formally analyzing existing and new methods helps to actually identify open questions and unattended problems in the research on outlier detection. In this study, we analyzed a selection of existing methods. The formalization should also help researchers to analyze new methods they are proposing and to state clearly their relationships to existing methods and their genuine merits. Our general model could therefore serve as a means of analysis of existing (and future) methods and should rather stimulate than suppress the design and development of new, improved or specialized methods for outlier detection. The formalization and the canonical algorithm outlined should be seen as a blueprint to start new algorithms with in order to emphasize on the one hand the use of improved outlier detection models, and on the other hand computational improvements over the baseline of computing the model for each object.

Acknowledgements Arthur Zimek is partially supported by NSERC.

References

Achtert E, Hettab A, Kriegel HP, Schubert E, Zimek A (2011) Spatial outlier detection: data, algorithms, visualizations. In: Proceedings of the 12th international symposium on spatial and temporal databases (SSTD), Minneapolis, MN, pp 512–516. doi:10.1007/978-3-642-22922-0_41

Achtert E, Goldhofer S, Kriegel HP, Schubert E, Zimek A (2012) Evaluation of clusterings—metrics and visual support. In: Proceedings of the 28th international conference on data engineering (ICDE), Washington, DC, pp 1285–1288. doi:10.1109/ICDE.2012.128

Aggarwal CC, Yu PS (2001) Outlier detection for high dimensional data. In: Proceedings of the ACM international conference on management of data (SIGMOD), Santa Barbara, CA, pp 37–46

Aggarwal CC, Yu PS (2008) Outlier detection with uncertain data. In: Proceedings of the 8th SIAM international conference on data mining (SDM), Atlanta, GA, pp 483–493

Agyemang M, Barker K, Alhajj R (2006) A comprehensive survey of numeric and symbolic outlier mining techniques. Intell Data Anal 10:521–538

Angiulli F, Fassetti F (2009) DOLPHIN: an efficient algorithm for mining distance-based outliers in very large datasets. ACM Trans Knowl Discov Data 3(1):4:1–4:57. doi:10.1145/1497577.1497581

Angiulli F, Pizzuti C (2002) Fast outlier detection in high dimensional spaces. In: Proceedings of the 6th European conference on principles of data mining and knowledge discovery (PKDD), Helsinki, Finland, pp 15–26. doi:10.1007/3-540-45681-3_2

Ankerst M, Breunig MM, Kriegel HP, Sander J (1999) OPTICS: ordering points to identify the clustering structure. In: Proceedings of the ACM international conference on management of data (SIGMOD), Philadelphia, PA, pp 49–60

Anselin L (1995) Local indicators of spatial association—LISA. Geogr Anal 27(2):93–115

Barnett V, Lewis T (1994) Outliers in statistical data, 3rd edn. Wiley, New York

Bay SD, Schwabacher M (2003) Mining distance-based outliers in near linear time with randomization and a simple pruning rule. In: Proceedings of the 9th ACM international conference on knowledge discovery and data mining (SIGKDD), Washington, DC, pp 29–38. doi:10.1145/956750.956758

Breunig MM, Kriegel HP, Ng R, Sander J (2000) LOF: identifying density-based local outliers. In: Proceedings of the ACM international conference on management of data (SIGMOD), Dallas, TX, pp 93–104

Chandola V, Banerjee A, Kumar V (2009) Anomaly detection: a survey. ACM Comput Surv 41(3):Article 15, 1–58. doi:10.1145/1541880.1541882

Chandola V, Banerjee A, Kumar V (2012) Anomaly detection for discrete sequences: a survey. IEEE Trans Knowl Data Eng 24(5):823–839

Chawla S, Sun P (2006) SLOM: a new measure for local spatial outliers. Knowl Inf Syst 9(4):412–429. doi:10.1007/s10115-005-0200-2

Chen F, Lu CT, Boedihardjo AP (2010) GLS-SOD: a generalized local statistical approach for spatial outlier detection. In: Proceedings of the 16th ACM international conference on knowledge discovery and data mining (SIGKDD), Washington, DC, pp 1069–1078. doi:10.1145/1835804.1835939

de Vries T, Chawla S, Houle ME (2010) Finding local anomalies in very high dimensional space. In: Proceedings of the 10th IEEE international conference on data mining (ICDM), Sydney, Australia, pp 128–137. doi:10.1109/ICDM.2010.151

Gao J, Tan PN (2006) Converting output scores from outlier detection algorithms into probability estimates. In: Proceedings of the 6th IEEE international conference on data mining (ICDM), Hong Kong, China, pp 212–221. doi:10.1109/ICDM.2006.43

Gao J, Liang F, Fan W, Wang C, Sun Y, Han J (2010) On community outliers and their efficient detection in information networks. In: Proceedings of the 16th ACM international conference on knowledge discovery and data mining (SIGKDD), Washington, DC, pp 813–822. doi:10.1145/1835804.1835907

Geary RC (1954) The contiguity ratio and statistical mapping. Inc Stat 5(3):115–146

Hadi AS, Rahmatullah Imon AHM, Werner M (2009) Detection of outliers. Wiley Interdiscip Rev Comput Stat 1(1):57–70

Hodge VJ, Austin J (2004) A survey of outlier detection methodologies. Artif Intell Rev 22:85–126. doi:10.1023/B:AIRE.0000045502.10941.a9

Hong SB, Nah W, Baek JH (2003) Abrupt shot change detection using multiple features and classification tree. In: Proceedings of the 4th international conference on intelligent data engineering and automated learning (IDEAL), Hong Kong, China, pp 553–560. doi:10.1007/978-3-540-45080-1_76

Jagadish HV, Koudas N, Muthukrishnan S (1999) Mining deviants in a time series database. In: Proceedings of the 25th international conference on very large databases (VLDB), Edinburgh, Scotland, pp 102–113

Jin W, Tung A, Han J (2001) Mining top-*n* local outliers in large databases. In: Proceedings of the 7th ACM international conference on knowledge discovery and data mining (SIGKDD), San Francisco, CA, pp 293–298. doi:10.1145/502512.502554

Jin W, Tung AKH, Han J, Wang W (2006) Ranking outliers using symmetric neighborhood relationship. In: Proceedings of the 10th Pacific-Asia conference on knowledge discovery and data mining (PAKDD), Singapore, pp 577–593. doi:10.1007/11731139_68

Keller F, Müller E, Böhm K (2012) HiCS: high contrast subspaces for density-based outlier ranking. In: Proceedings of the 28th international conference on data engineering (ICDE), Washington, DC

Kim HH, Kim YH (2010) Toward a conceptual framework of key-frame extraction and storyboard display for video summarization. J Am Soc Inf Sc Technol 61(5):927–939. doi:10.1002/asi.21317

Knorr EM, Ng RT (1997) A unified notion of outliers: properties and computation. In: Proceedings of the 3rd ACM international conference on knowledge discovery and data mining (KDD), Newport Beach, CA, pp 219–222

Knorr EM, Ng RT (1998) Algorithms for mining distance-based outliers in large datasets. In: Proceedings of the 24th international conference on very large databases (VLDB), New York City, NY, pp 392–403

Knorr EM, Ng RT, Tucanov V (2000) Distance-based outliers: algorithms and applications. VLDB J 8(3–4):237–253

Kollios G, Gunopulos D, Koudas N, Berchthold S (2003) Efficient biased sampling for approximate clustering and outlier detection in large datasets. IEEE Trans Knowl Data Eng 15(5):1170–1187. doi:10.1109/TKDE.2003.1232271

Kou Y, Lu CT, Chen D (2006) Spatial weighted outlier detection. In: Proceedings of the 6th SIAM international conference on data mining (SDM), Bethesda, MD

Kou Y, Lu CT, Dos Santos RF (2007) Spatial outlier detection: a graph-based approach. In: Proceedings of the 19th IEEE international conference on tools with artificial intelligence (ICTAI), Patras, Greece, pp 281–288. doi:10.1109/ICTAI.2007.169

Kriegel HP, Schubert M, Zimek A (2008) Angle-based outlier detection in high-dimensional data. In: Proceedings of the 14th ACM international conference on knowledge discovery and data mining (SIGKDD), Las Vegas, NV, pp 444–452. doi:10.1145/1401890.1401946

Kriegel HP, Kröger P, Schubert E, Zimek A (2009a) LoOP: local outlier probabilities. In: Proceedings of the 18th ACM conference on information and knowledge management (CIKM), Hong Kong, China, pp 1649–1652. doi:10.1145/1645953.1646195

Kriegel HP, Kröger P, Schubert E, Zimek A (2009b) Outlier detection in axis-parallel subspaces of high dimensional data. In: Proceedings of the 13th Pacific-Asia conference on knowledge discovery and data mining (PAKDD), Bangkok. Thailand, pp 831–838. doi:10.1007/978-3-642-01307-2_86

Kriegel HP, Kröger P, Schubert E, Zimek A (2011) Interpreting and unifying outlier scores. In: Proceedings of the 11th SIAM international conference on data mining (SDM), Mesa, AZ, pp 13–24

Kriegel HP, Kröger P, Schubert E, Zimek A (2012) Outlier detection in arbitrarily oriented subspaces. In: Proceedings of the 12th IEEE international conference on data mining (ICDM), Brussels, Belgium

Lazarevic A, Kumar V (2005) Feature bagging for outlier detection. In: Proceedings of the 11th ACM international conference on knowledge discovery and data mining (SIGKDD), Chicago, IL, pp 157–166. doi:10.1145/1081870.1081891

Lee W, Kim H, Kang H, Lee J, Kim Y, Jeon S (2004) Video cataloging system for real-time scene change detection of news video. In: Proceedings of Teh 10th international workshop on combinatorial image analysis (IWCIA), Auckland, New Zealand, pp 705–715

Lee JG, Han J, Li X (2008) Trajectory outlier detection: a partition-and-detect framework. In: Proceedings of the 24th international conference on data engineering (ICDE), Cancun, Mexico, pp 140–149. doi:10.1109/ICDE.2008.4497422

Liu X, Lu CT, Chen F (2010) Spatial outlier detection: Random walk based approaches. In: Proceedings of the 18th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (ACM GIS), San Jose, CA, pp 370–379, doi:10.1145/1869790.1869841

Lu CT, Chen D, Kou Y (2003) Algorithms for spatial outlier detection. In: Proceedings of the 3rd IEEE international conference on data mining (ICDM), Melbourne, FL, pp 597–600. doi:10.1109/ICDM.2003.1250986

Money AG, Agius H (2008) Video summarisation: a conceptual framework and survey of the state of the art. J Vis Commun Image Represent 19(2):121–143. doi:10.1016/j.jvcir.2007.04.002

Moran P (1950) Notes on continuous stochastic phenomena. Biometrika 37(1/2):17–23

Müller E, Assent I, Steinhausen U, Seidl T (2008) OutRank: ranking outliers in high dimensional data. In: Proceedings of the 24th international conference on data engineering (ICDE) workshop on ranking in databases (DBRank), Cancun, Mexico, pp 600–603. doi:10.1109/ICDEW.2008.4498387

Müller E, Schiffer M, Gerwert P, Hannen M, Jansen T, Seidl T (2010a) SOREX: subspace outlier ranking exploration toolkit. In: Proceedings of the European conference on machine learning and knowledge discovery in databases (ECML PKDD), Barcelona, Spain, pp 607–610. doi:10.1007/978-3-642-15939-8_44

Müller E, Schiffer M, Seidl T (2010b) Adaptive outlierness for subspace outlier ranking. In: Proceedings of the 19th ACM conference on information and knowledge management (CIKM), Toronto, ON, Canada, pp 1629–1632. doi:10.1145/1871437.1871690

Müller E, Schiffer M, Seidl T (2011) Statistical selection of relevant subspace projections for outlier ranking. In: Proceedings of the 27th international conference on data engineering (ICDE), Hannover, Germany, pp 434–445. doi:10.1109/ICDE.2011.5767916

Nguyen HV, Ang HH, Gopalkrishnan V (2010) Mining outliers with ensemble of heterogeneous detectors on random subspaces. In: Proceedings of the 15th international conference on database systems for advanced applications (DASFAA), Tsukuba, Japan, pp 368–383. doi:10.1007/978-3-642-12026-8_29

Nguyen HV, Gopalkrishnan V, Assent I (2011) An unbiased distance-based outlier detection approach for high-dimensional data. In: Proceedings of the 16th international conference on database

systems for advanced applications (DASFAA), Hong Kong, China, pp 138–152. doi:10.1007/978-3-642-20149-3_12

Orair GH, Teixeira C, Wang Y, Meira W Jr, Parthasarathy S (2010) Distance-based outlier detection: consolidation and renewed bearing. Proc VLDB Endow 3(2):1469–1480

Papadimitriou S, Kitagawa H, Gibbons P, Faloutsos C (2003) LOCI: fast outlier detection using the local correlation integral. In: Proceedings of the 19th international conference on data engineering (ICDE), Bangalore, India, pp 315–326. doi:10.1109/ICDE.2003.1260802

Pei Y, Zaïane O, Gao Y (2006) An efficient reference-based approach to outlier detection in large datasets. In: Proceedings of the 6th IEEE international conference on data mining (ICDM), Hong Kong, China, pp 478–487. doi:10.1109/ICDM.2006.17

Pickup L, Zisserman A (2009) Automatic retrieval of visual continuity errors in movies. In: Proceedings of the 8th ACM international conference on image and video retrieval (CIVR), Santorini, Greece. doi:10.1145/1646396.1646406

Pokrajac D, Lazarevic A, Latecki LJ (2007) Incremental local outlier detection for data streams. In: Proceedings of the IEEE symposium on computational intelligence and data mining (CIDM), Honolulu, HI, pp 504–515. doi:10.1109/CIDM.2007.368917

Ramaswamy S, Rastogi R, Shim K (2000) Efficient algorithms for mining outliers from large data sets. In: Proceedings of the ACM international conference on management of data (SIGMOD), Dallas, TX, pp 427–438

Roddick JF, Spiliopoulou M (1999) A bibliography of temporal, spatial and spatio-temporal data mining research. ACM SIGKDD Explor 1(1):34–38. doi:10.1145/846170.846173

Schubert E, Wojdanowski R, Zimek A, Kriegel HP (2012) On evaluation of outlier rankings and outlier scores. In: Proceedings of the 12th SIAM international conference on data mining (SDM), Anaheim, CA, pp 1047–1058

Shekhar S, Lu CT, Zhang P (2003) A unified approach to detecting spatial outliers. GeoInformatica 7(2):139–166. doi:10.1023/A:1023455925009

Su X, Tsai CL (2011) Outlier detection. Wiley Interdiscip Rev Data Min Knowl Discov 1(3):261–268. doi:10.1002/widm.19

Sun P, Chawla S (2004) On local spatial outliers. In: Proceedings of the 4th IEEE international conference on data mining (ICDM), Brighton, UK, pp 209–216. doi:10.1109/ICDM.2004.10097

Takeuchi J, Yamanishi K (2006) A unifying framework for detecting outliers and change points from time series. IEEE Trans Knowl Data Eng 18(4):482–492. doi:10.1109/TKDE.2006.1599387

Tan PN, Steinbach M, Kumar V (2006) Introduction to Data Mining. Addison Wesley, Upper Saddle River

Tang J, Chen Z, Fu AWC, Cheung DW (2002) Enhancing effectiveness of outlier detections for low density patterns. In: Proceedings of the 6th Pacific-Asia conference on knowledge discovery and data mining (PAKDD), Taipei, Taiwan, pp 535–548. doi:10.1007/3-540-47887-6_53

Vu NH, Gopalkrishnan V (2009) Efficient pruning schemes for distance-based outlier detection. In: Proceedings of the European conference on machine learning and knowledge discovery in databases (ECML PKDD), Bled, Slovenia, pp 160–175. doi:10.1007/978-3-642-04174-7_11

Yamanishi K, Takeuchi JI, Williams G, Milne P (2004) On-line unsupervised outlier detection using finite mixture with discounting learning algorithms. Data Min Knowl Discov 8:275–300. doi:10.1023/B:DAMI.0000023676.72185.7c

Yu JX, Qian W, Lu H, Zhou A (2006) Finding centric local outliers in categorical/numerical spaces. Knowl Inf Syst 9(3):309–338. doi:10.1007/s10115-005-0197-6

Zhang J, Lou M, Ling TW, Wang H (2004) HOS-miner: a system for detecting outlying subspaces of high-dimensional data. In: Proceedings of the 30th international conference on very large databases (VLDB), Toronto, Canada, pp 1265–1268

Zhang K, Hutter M, Jin H (2009) A new local distance-based outlier detection approach for scattered real-world data. In: Proceedings of the 13th Pacific-Asia conference on knowledge discovery and data mining (PAKDD), Bangkok, Thailand, pp 813–822. doi:10.1007/978-3-642-01307-2_84

Zimek A, Schubert E, Kriegel HP (2012) A survey on unsupervised outlier detection in high-dimensional numerical data. Stat Anal Data Min 5(5):363–387. doi:10.1002/sam.11161