applied sciences

MDPI

Article

Outlier Recognition via Linguistic Aggregation of Graph Databases

Adam Niewiadomski *,†, Agnieszka Duraj †and Monika Bartczak †

Institute of Information Technology, Lodz University of Technology, Wólczańska 215, 90-924 Łódź, Poland; Agnieszka.Duraj@p.lodz.pl (A.D.); Monika.Bartczak@dokt.p.lodz.pl (M.B.)

* Correspondence: Adam.Niewiadomski@p.lodz.pl

† These authors contributed equally to this work.

Abstract: Datasets frequently contain uncertain data that, if not interpreted with care, may affect information analysis negatively. Such rare, strange, or imperfect data, here called "outliers" or "exceptions" can be ignored in further processing or, on the other hand, handled by dedicated algorithms to decide if they contain valuable, though very rare, information. There are different definitions and methods for handling outliers, and here, we are interested, in particular, in those based on linguistic quantification and fuzzy logic. In this paper, for the first time, we apply definitions of outliers and methods for recognizing them based on fuzzy sets and linguistically quantified statements to find outliers in non-relational, here graph-oriented, databases. These methods are proposed and exemplified to identify objects being outliers (e.g., to exclude them from processing). The novelty of this paper are the definitions and recognition algorithms for outliers using fuzzy logic and linguistic quantification, if traditional quantitative and/or measurable information is inaccessible, that frequently takes place in the graph nature of considered datasets.

Keywords: outliers in graph datasets; outlier recognition; fuzzy logic; outliers in terms of linguistic quantification

Citation: Niewiadomski, A.; Duraj, A.; Bartczak, M. Outlier Recognition via Linguistic Aggregation of Graph Databases. *Appl. Sci.* **2021**, *11*, 7434. https://doi.org/10.3390/app11167434

Academic Editor: Ki-Yong Oh

Received: 7 June 2021
Accepted: 6 August 2021
Published: 12 August 2021

1. Introduction

In collecting and processing information, there appear some uncertainties, mostly incomplete or imprecise data. Sources of uncertainty are usually measurements, probabilistic (stochastic uncertainty), lack of credibility, and linguistic descriptions (natural language uncertainty). To take care about such data, that, although strange, may contain valuable and exceptional information, one can consider them to be outliers . An "outlier" or "exception" (also: deviation, anomaly, aberration, etc.) means an observation that is rare, special, unique, unexampled, or infrequent. These terms mean that properties of interest possessed by outlying objects, are specific to recipients considering/processing them. Outliers are especially noticeable as highlighted or unusual observations on a background of numerous phenomena/objects similar one to another, typical, or ordinary. Unrecognized outliers in data exploration and mining, may decrease reliability of analysis, increase data imprecision and noise. In other words, outlying objects may distort or blur the final gists or meaning of collections analyzed. On the contrary, appropriately recognized outliers can bring unique information on change of activities and congestion in networks or intrusions into them, illegal use of debit/credit cards, serious damages of production lines, rapid changes of patients' health status and parameters of medical devices, etc.

The literature enumerates various definitions of outliers, mostly subjective, intuitive, and dependent on different numerical characteristic showing "how much" considered objects are atypical for analyzed databases or sets. For example, the definition of outlier by Hawkins, the most frequently quoted, is [1] "An outlier is an observation which deviates so much from the other observations as to arouse suspicions that it was generated by a different mechanism". Additionally, in [1], an outlier is defined as: "any object x in the

space $\mathcal{X}$, which has some abnormal, unusual characteristics in comparison to other objects from $\mathcal{X}$". Next, in [2], the authors say that outliers are "noise points outside the set which determine clusters", or, alternatively, as "points outside the clusters but separated from the noise" . Next, according to [3,4], "a point p in a data set is an outlier with respect to parameters k and λ, if no more than k points in the data set are at a distance λ or less from p", with $k \in \mathbb{N}, \lambda \in \mathbb{R}$. The λ parameter can be interpreted in terms of relations between objects in $\{x_1, \ldots, x_N\}$, $N \in \mathbb{N}$, not only as a metric, distance, a semantic connection (e.g., similarity as a binary fuzzy relation), etc. Moreover, another definition for outliers are worth mentioning from [5–7], and global and local outliers proposed in [8–10]. Interesting applications and techniques as far as approaches mixing outliers detection with clustering methods are proposed in [11,12]. Outliers in data streams (series or linear structures) are considered in [13]. Recognition of outliers is also the subject of consideration in [3,14,15] and many other.

The most important idea on the above literature review, is that it does not provide one objective or axiomatic definition of an outlier at all. This is the specificity of this domain of research: outliers can be defined in different manners, with different characteristics and parameters, and in specific relations to particular problems, issues, datasets, etc. Thus, in this contribution, we focus on outliers defined in terms of linguistic quantification and fuzzy sets [16–19], since no publications on outliers handled this way can be found. Recognizing outliers via fuzzy quantification and linguistic information can be useful when numerical information and traditional quantitative terms are inaccessible for a given set of objects. In such situations, the only information to detect anomalies is human experience and expert knowledge expressed linguistically (which is, in general, a common and obvious reason to apply fuzzy systems and techniques in various issues). Moreover, in our previous papers we focused on outliers in relational databases, and now, the main novelty of the paper is a successful attempt of using the methods for graph-oriented datasets. It is worth mentioning that graph datasets are frequently applied in circumstances in which relational structures are insufficient or unable to represent data and their meaning properly to the required context (e.g., in Customer Relationship Management Systems, CRM, or in social media).

The paper is organized as follows: Section 2 is a list of preliminary definitions and operations in fuzzy logic and the linguistic quantification of statements. Section 3 reminds our definitions for outliers in terms of linguistic information and, based on these definitions, algorithms for outlier detection/recognition. The specificity of preprocessing graph datasets to use fuzzy methods of recognition is illustrated in Section 4. An implemented example of outliers recognition is given to show how the proposed methods work on real graph datasets (here: a database on consumer complaints in a CRM system in a bank [20]) in Section 5. Finally, in Section 6, we discuss on future possibilities of works in the field presented.

2. Fuzzy Sets and Linguistic Quantification of Statements

In this section, we briefly review the basics of the linguistic quantification of statements in the sense of Zadeh [21]. A fuzzy set A in a finite non-empty $\mathcal{X} = \{x_1, \ldots, x_N\}$, $N \in \mathbb{N}$, is denoted as $A = \{\langle x, \mu_A(x)\rangle \colon x \in \mathcal{X}\}$, and $\mu_A \colon \mathcal{X} \to [0,1]$ is its membership function. The intersection of fuzzy sets A, B in $\mathcal{X}$ is a fuzzy set $A \cap B$ in $\mathcal{X}$:

$$\mu_{A\cap B}(x) = t(\mu_A(x), \mu_B(x)), \tag{1}$$

where t is a triangular norm, e.g., min or product. The cardinality of A, the so-called $\Sigma\text{count}(A)$, *sigma-count*, is defined as [22]:

$$\text{card}(A) = \Sigma\text{count}(A) =_{df} \sum_{x\in\mathcal{X}} \mu_A(x). \tag{2}$$

A *relative cardinality* of A with respect to a fuzzy set B is proposed:

$$\text{card}(A|B) =_{df} \frac{\text{card}(A \cap B)}{\text{card}(B)}. \tag{3}$$

For fuzzy set A in a continuous and uncountable universe of discourse $\mathcal{Y}$, the following counterpart of (2) is proposed:

$$\text{clm}(A) =_{df} \int_{y \in \mathcal{Y}} \mu_A(y) dy, \tag{4}$$

where "clm" comes from "cardinality-like measure". Finally, the support of A in $\mathcal{X}$ is a non-fuzzy set in $\mathcal{X}$:

$$\text{supp}(A) = \{x\colon \mu_A(x) > 0\}. \tag{5}$$

The generalized support of A is called *α-cut*, *alpha-cut*, and replaces the right side of inequality in Equation (5) with $\alpha \in [0,1]$: $A_\alpha = \{x\colon \mu_A(x) > \alpha\}$. Alpha cuts are non-fuzzy sets and they are necessary to define the convexity of fuzzy set A in $\mathcal{X}$: A is convex in $\mathcal{X}$ iff each of its alpha-cuts is convex non-fuzzy set. A in $\mathcal{X}$ is normal iff $\sup_{x \in \mathcal{X}} \mu_A(x) = 1$.

Assume, S, W are linguistically expressed characteristics of objects, represented by fuzzy sets in a finite $\mathcal{D} = \{d_1, d_2, \ldots, d_N\}$, $N \in \mathbb{N}$. Q is a relative fuzzy quantifier describing quantities of objects as ratios to a superset (usually, the universe of discourse of Q), e.g., "many of", "about 1/3", "less than half", "very few", and Q is modeled by a fuzzy set that is convex and normal in $[0,1]$ [23] (see Equations (23)–(25) as examples of relative fuzzy quantifiers). The first and the second form of linguistically quantified statements by Zadeh [21], are:

$$Q\ d\text{'s are } S, \tag{6}$$

$$Q\ d\text{'s being } W \text{ are } S, \tag{7}$$

respectively. Their degrees of truth in terms of fuzzy logic are evaluated via Equation (3) as:

$$T(Q\ d\text{'s are } S) = \mu_Q(\text{card}(S|\mathcal{D}), \tag{8}$$

$$T(Q\ d\text{'s being } W \text{ are } S) = \mu_Q(\text{card}(S|W). \tag{9}$$

It must be noticed that in the context of detecting outliers, only the so-called regular relative linguistic quantifier Q with the monotonically non-increasing μ_Q is taken into account, i.e.:

$$\mu_Q(0) = 1, \ \ \mu_Q(1) = 0, \tag{10}$$

$$\forall_{x_1, x_2 \in [0,1]} x_1 \leq x_2 \longrightarrow \mu_Q(x_1) \geq \mu_Q(x_2). \tag{11}$$

Moreover, quality of the fuzzy quantified statements can be additionally evaluated with the given measures of quality of fuzzy quantifier Q: T_{supp} and T_{clm} [23,24]. They are based on support (5) and on the clm measure (4), respectively, of a fuzzy set that represents the fuzzy quantifier Q:

$$T_{\text{supp}}(Q) = 1 - |\text{supp}(Q)|, \tag{12}$$

$$T_{\text{clm}}(Q) = 1 - \text{clm}(Q). \tag{13}$$

Both measures depend on the presented characteristics of Q and their meaning is: the closer to 1, the more precise quantifier Q is.

Naturally, the choice of representations for quantifier Q and characteristics (properties) S, W depends on the specificity of databases being analyzed, and, in particular, on linguistic information provided by experts. In Section 3, linguistically quantified statements (6) and

(7) are essential for proposed definitions of outliers, and for detecting/recognizing outliers in graph-oriented datasets.

3. Detecting and Recognizing Outliers with Linguistic Information

3.1. Outliers in Terms of Linguistically Aggregated Information

Now, we present the definitions of outliers based on fuzzy representations of linguistic information, i.e., by linguistically quantified statements with their degrees of truth.

Definition 1 (Outlier via the first form of linguistically quantified statement)**.** *Let* $\mathcal{D} = \{d_1, \ldots, d_N\}$, $N \in \mathbb{N}$, *be a finite non-empty set of objects. Let* S *be a linguistic expression characterizing objects in* $\mathcal{D}$ *and represented by a fuzzy set in* $\mathcal{D}$*. Let* Q *be a relative regular non-increasing linguistic quantifier (e.g., "almost none", "very few", "only several", or synonymous) represented by a fuzzy set in* $[0,1]$*, and* $\alpha \in [0,1]$*. An object* $d \in \mathcal{D}$ *is outlier iff:*

$$T(Q\ d\text{'s are } S) \geq \alpha. \tag{14}$$

The degree of truth of (14) is evaluated via (8):

$$T(Q\ d\text{'s are } S) = \mu_Q\left(\frac{\sum_{n=1}^{N} \mu_S(d_n)}{N}\right). \tag{15}$$

The definition of an outlier in terms of the second form of a linguistically quantified statement (7), i.e., taking into account two properties, S and W possessed by objects in $\mathcal{D}$, is introduced:

Definition 2 (Outlier via the second form of linguistically quantified statement)**.** *Let* $\mathcal{D}$ *is defined as in Definition 1, and* S, W *are linguistic expressions characterizing objects* $d \in \mathcal{D}$ *and represented by fuzzy sets in* $\mathcal{D}$*. Let* $\alpha \in [0,1]$*, and* Q*—a relative regular non-increasing linguistic quantifier as in Definition 1. An object* $d \in \mathcal{D}$ *is outlier iff:*

$$T(Q\ d\text{'s being } W \text{ are } S) \geq \alpha. \tag{16}$$

The degree of truth of (16) can be evaluated with Equation (9) as:

$$T\big(Q\ d\text{'s being } W \text{ are } S\big) = \mu_Q\left(\frac{\Sigma\text{count}(S \cap W)}{\Sigma\text{count}(W)}\right). \tag{17}$$

Two algorithms for detecting outliers are now presented, related to Definitions 1 and 2, respectively. They are designed as tools for detecting outliers in datasets (anomalies, exceptional data, etc.) in circumstances when only imprecise and linguistically formulated knowledge of their specificity is available. In particular, objects d in an analyzed dataset $\mathcal{D}$ are considered to be outliers, if they are intuitively characterized by expressions such as "small", "big", "hot", "very expensive" etc., represented by the S, W fuzzy sets, and the quantity of them is either not determined precisely, but linguistically expressed with statements as "very few", "almost none", represented by Q. Hence, the algorithms confirm that outliers exists in a dataset, iff statements "Q x's are S" or "Q x's being/having W are S" are of sufficiently large (larger than threshold α) degree of truth. The common assumptions for Algorithms 1 and 2 are:

1. $\mathcal{D} = \{d_1, d_2, \ldots, d_N\}$, $N \in \mathbb{N}$—a non-empty finite dataset,
2. S, W—linguistic labels for properties of d's in $\mathcal{D}$, represented by fuzzy sets,
3. $\{Q_1, Q_2, \ldots, Q_K\}$, $K \in \mathbb{N}$—relative regular monotonically non-increasing linguistic quantifiers, as in Definition 1, given as fuzzy sets in $[0,1]$,
4. $\alpha \in [0,1]$—an arbitrarily chosen threshold for degrees of truth for (14), (16),
5. Two possible results are produced: *true* = "THERE EXIST OUTLIERS IN $\mathcal{D}$", *false* = "NO OUTLIERS IN $\mathcal{D}$".

To detect outliers in $\mathcal{D}$ using Algorithm 1, we need the entry query in the form of:

$$\textit{How many } d\textit{'s are } S? \tag{18}$$

to detect outliers in $\mathcal{D}$ with respect to the S property and using linguistic quantifiers $Q_1, Q_2, \ldots, Q_K$.

Algorithm 1 Detecting outliers via the first form of linguistically quantified statement.

1: **for all** $k = 1, 2, \ldots, K$ **do**
2: $T_k \leftarrow 0, r \leftarrow 0$
3: **for all** $n = 1, 2, \ldots, N$ **do**
4: $r \leftarrow r + \mu_S(d_n)$
5: $T_k \leftarrow \mu_{Q_k}(r/N)$
6: **if not** $T_1 > \alpha$ **and not** $T_2 > \alpha$ **and** ... **and not** $T_K > \alpha$ **then return** "NO OUTLIERS IN $\mathcal{D}$"
7: else **return** "THERE EXIST OUTLIERS IN $\mathcal{D}$"

Comment: K linguistic expressions and their degrees of truth evaluated in Step 5:

$$Q_1 \ d\text{'s are } S \ [T_1], \ldots, Q_K \ d\text{'s are } S \ [T_K], \tag{19}$$

are side effects of the algorithm, and it is important in Algorithm 3 recognizing outlying d's (if detected) in $\mathcal{D}$, see Section 3.2.

Now, Algorithm 2 referring to Definition 2 is presented: outlying objects in $\mathcal{D}$ are here detected on the base of two, possibly overlapping, linguistic characteristics S, W. To detect outliers in $\mathcal{D}$ using Algorithm 2, the entry query is necessary:

$$\textit{How many } d\textit{'s being } W \textit{ are } S? \tag{20}$$

Algorithm 2 Detecting outliers via the second form of linguistically quantified statement.

1: **for all** $k = 1, 2, \ldots, K$ **do**
2: $T_k \leftarrow 0, rn \leftarrow 0, rd \leftarrow 0$
3: **for all** $n = 1, 2, \ldots, N$ **do**
4: $rn \leftarrow rn + \mu_{W \cap S}(d_n)$
5: $rd \leftarrow rd + \mu_W(d_n)$
6: $T_k \leftarrow \mu_{Q_k}(rn/rd)$
7: **if not** $T_1 > \alpha$ **and not** $T_2 > \alpha$ **and** ... **and not** $T_K > \alpha$ **then return** "NO OUTLIERS IN $\mathcal{D}$"
8: else **return** "THERE EXIST OUTLIERS IN $\mathcal{D}$"

As with Algorithm 1, the side effect of Algorithm 2 are K linguistically quantified statements with their degrees of truth:

$$Q_1 \ d\text{'s being } W \text{ are } S \ [T_1], \ldots, Q_K \ d\text{'s being } W \text{ are } S \ [T_K] \tag{21}$$

3.2. Recognizing Outliers via Linguistic Information

Here, the next two algorithms, now for recognizing and enumerating outlying observations $d \in \mathcal{D}$ possessing properties S, W, are introduced. The outlier detection tools, Algorithms 1 and 2, presented in Section 3.1, confirm only that some outliers **do exist** in dataset $\mathcal{D}$ (*true*) or **do not exist** (*false*). However, subsets of outliers $\mathcal{D}_{\text{out}} \subset \mathcal{D}$ remain unspecified. Hence, now we deal with algorithms accomplishing the following task: recognizing and enumerating particular objects in $\mathcal{D}$ that are outliers with respect to the S, W characteristics. in other words, via the algorithms presented here, the subsets of outliers $\mathcal{D}_{\text{out}} \subset \mathcal{D}$ with respect to S, W, are determined.

First, we take into account outliers according to Definition 1, and assumptions and symbols are as for Algorithm 1. Hence, for given dataset $\mathcal{D} = \{d_1, d_2, \ldots, d_N\}$, $N \in \mathbb{N}$, properties S, W, linguistic quantifiers Q_1, Q_2, ..., Q_K, $K \in \mathbb{N}$, and parameter $\alpha \in [0,1]$, Algorithm 3 is based on the query (18). Of course, Algorithm 3 is fired iff Algorithm 1 did detect anomalies in $\mathcal{D}$ (since no point to seek for objects in $\mathcal{D}_{\text{out}} = \emptyset$, otherwise). The result of firing Algorithm 3 is the collection of outlying observations in $\mathcal{D}$ selected with the S characteristics and the query in the form of (18).

Algorithm 3 Recognizing outliers detected with Algorithm 1.

1: declare $\mathcal{D}_{\text{out}} = \emptyset$
2: **for all** $n = 1, 2, \ldots, N$ **do**
3: **if** $\mu_S(d_n) > \alpha$ **then** $\mathcal{D}_{\text{out}} \leftarrow \mathcal{D}_{\text{out}} \cup \{d_n\}$
4: **return** $\mathcal{D}_{\text{out}}$

Per analogiam, subsets containing anomalies in $\mathcal{D}$ can be determined via Definition 2, with the same assumptions as for Algorithm 2. Hence, the Algorithm 4 is proposed with the query on its input given by (20). Its result is returned as an array of indices of found anomalies $d_n \in \mathcal{D}$, $n \in \{1, 2, \ldots, N\}$.

Algorithm 4 Recognizing outliers detected with Algorithm 2.

1: $i \leftarrow 0$
2: declare $outlierIndices[N]$
3: **for all** $n = 1, 2, \ldots, N$ **do**
4: **if** $\mu_{S \cap W}(d_n) > \alpha$ **then** $\{i \leftarrow i + 1; outlierIndices[i] \leftarrow n\}$
5: **if** $i = 0$ **then return** "NO OUTLIERS IN $\mathcal{D}$"
6: else **return** $outliersIndices$

4. Preprocessing Graph Databases to Use Linguistically Aggregated Information

Now, we briefly explain how filtering graph database is done. The aim is to select/filter objects of given types (e.g., clients, transactions, transfers) and represent them as a one-dimensional set (e.g., sequence, collection, list). It is important to notice that objects in a resulting sequence need not to be counterparts of every single vertex of a given type, e.g., a resulting object is not necessarily related to one "complaint" vertex, but rather to a set of data combined from several vertices/edges/properties describing a particular fact, here: a complaint. Next, selected objects are inputs for the method of outlier detection via linguistically quantified statements (described in Section 3). We assume that graph database $\mathcal{D}$ is represented by directed labeled graph G in which V (or $V(G)$) is the set of vertices of G, E (or $E(G)$) is the set of edges of G, and L (or $L(G)$) is the set of labels of G. It is t is important that both vertices and edges can have labels assigned, so G can be *vertex-labeled* or *edge-labeled*, respectively. In this case, $L(G)$ can be divided into two subsets $VL(G)$ (vertices' labels) and $EL(G)$ (edges' labels). A label of a vertex determines its type, and a label of an edge determines relation between two vertices. Finally, $P(V, E)$ is a set of properties that can be possessed by vertices or edges, e.g., amount of transfer, date of complaint, etc. (in fact, properties are counterparts of attributes in relational data models).

The graph database of Customer Relationship Management (CRM) system taken into account in the experiment, handles complaints submitted by clients of a bank [20]. The vertices of graph G represent data on customers' complaints submitted to CRM. G is the both edge- and vertex-labeled graph consisting of 355,371 unique vertices, $V(G)$, and 2,111,884 edges, $E(G)$. The vertices represent granules of information on complaints and are labeled with $VL(G) = \{$COMPLAINT, ISSUE, PRODUCT, SUBISSUE, ZIP CODE, RESPONSE, COMPANY, SUBMITTED, TAGS$\}$. The edges represent relations between data stored in vertices of different labels (types), and are labeled with $EL(G) = \{$WITH, ABOUT, VIA, TO, AGAINST, IN CATEGORY$\}$. Additionally, we have the set of properties possessed

by vertices in the base: $P(V,E) = \{\text{DAY, DISPUTED, ID, MONTH, NAME, TIMELY, YEAR}\}$. The sample structure of graph G is illustrated in Figure 1.

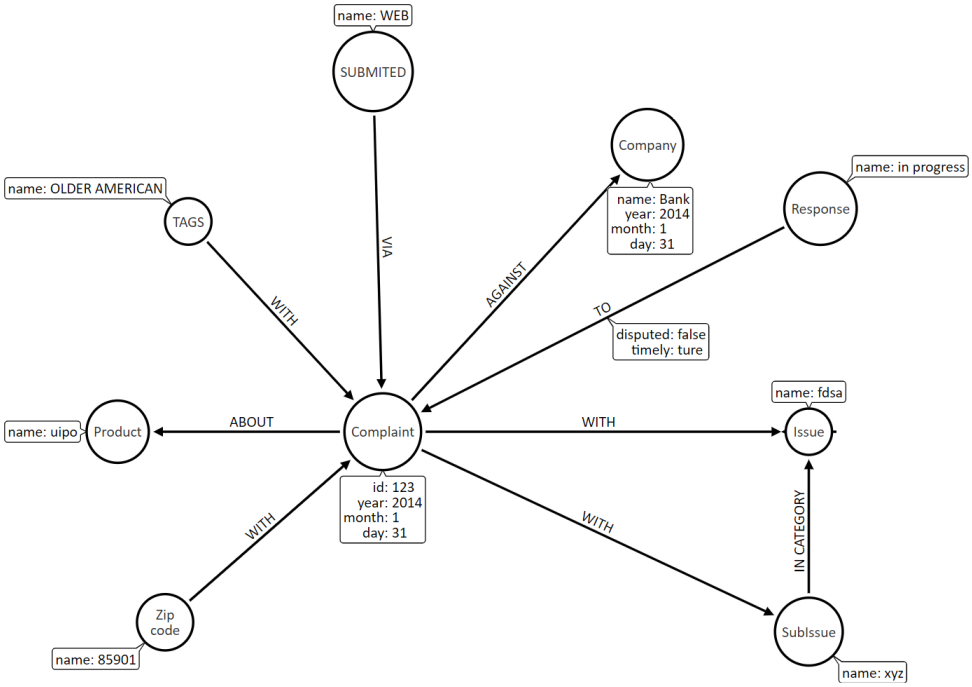

Figure 1. The structure of the processed graph database with labeled vertices (circles and their names), labeled edges (arrows and their names), and properties of vertices.

The keypoint of transforming selected nodes to a sequence is to construct and execute the query. Notice, it is not a simple serialization or graph searching, since objects in the resulting sequence may be described by properties of several different vertices and/or edges. The structure of a single object in the resulting sequence is determined by a specific query R in the general form of:

$$R(VL_R, EL_R, P_R), \tag{22}$$

where $VL_R \subseteq VL(G)$ is the subset of vertices' labels, $EL_R \subseteq EL(G)$ is the subset of edges' labels, and $P_R \subseteq P(V,E)$ is the subset of vertices'/labels' properties. The Neo4j database management system is used in the experiment [25], and one of queries executed to obtain selected vertices on complaints as sequence $\mathcal{D}$, is based on the MATCH clause (a counterpart of SELECT in SQL):

$R(VL_R, EL_R, P_R)$ = ("MATCH(r: Response) – [TO] – (complaint)" + "MATCH(company: Company) – [AGAINST] – (complaint)" + "MATCH(t: Tags) – [WITH] – (complaint)" + "MATCH(p: Product) – [ABOUT] – (complaint)" + "MATCH(s: Submitted) – [VIA] – (complaint)" + "RETURN complaint.id, complaint.year, complaint.day," + "complaint.month, r.name, company.name," + "TO.disputed, TO.timely, p.name, t.name, s.name");
where $VL_R = \{\text{COMPLAINT, PRODUCT, RESPONSE, COMPANY, SUBMITTED, TAGS}\}$, $EL_R = \{\text{WITH, ABOUT, VIA, TO, AGAINST}\}$, and $P_R = P(V,E)$, see (22).

As the result of executing this query on the given graph dataset, sequence $\mathcal{D} = \{d_1, \ldots, d_N\}$, $N = 40,083$, of objects representing complaints is selected. Sample records of the sequence are illustrated in Table 1.

Table 1. Sample records of dataset D obtained as a sequence from the graph database processed, see Section 4 and Figure 1.

ID	Company	Date Received	Sent by CFPB	Per Capita Income	Product	Zipcode
801691	STONELEIGH RECOVERY ASSOCIATES, LLC	11.04.2014	09.04.2014	$56,791	debt collection	94930
801371	NATIONSTAR mortgage	11.04.2014	09.04.2014	$56,791	mortgage	94947
305167	WELLS FARGO & COMPANY	07.02.2013	05.02.2013	$62,018	mortgage	22206
716577	WELLS FARGO & COMPANY	20.02.2014	15.02.2014	$62,018	mortgage	22203
809768	BARCLAYS BANK DELAWARE	15.04.2014	15.04.2014	$26,924	credit card	87124
941562	COMERICA	17.07.2014	17.07.2014	$23,023	bank account or service	48653
720703	Vision Financial Corp.	21.05.2012	21.05.2012	$60,929	debt collection	75089
…	…	…	…	…	…	…

5. Application Example

The proposed algorithms for outlier recognition via linguistic information are now implemented on the sequence obtained (see Table 1). Because of specificity of data and possible connections/relations between them, a graph representation has been chosen for the set of complaints submitted. The general schema of the experiment is to filter (select) the vertices representing complaints themselves and represent them as the sequence of objects, the parameters of which are inputs for algorithms detecting outliers (see Section 3).

Next, properties of interest of these objects are fuzzified, which means their crisp values are assigned to labels and corresponding fuzzy sets: "date received" to {early spring, middle spring, summer, autumn, early winter, winter} in $\mathcal{X}_1$, "county per capita income" to {poor, middle, rich} in $\mathcal{X}_2$, and "time of sending complaint" to {short, average, long} in $\mathcal{X}_3$, see sample linguistic values of chosen properties of objects in $\mathcal{D}$ (23)–(25). S_1 represents label "early spring" in $\mathcal{X}_1 = \{1, 2, \ldots, 366\}$—days in a year with μ_{S_1} given

$$\mu_{S_1}(x) = \begin{cases} \frac{x-79}{14}, & 79 \leq x \leq 93 \\ \frac{-x+107}{14}, & 93 \leq x \leq 107 \end{cases} \tag{23}$$

and 0 otherwise. S_2 represents label "rich county" in $\mathcal{X}_2 = \{0, 1, \ldots, 70\}$—per capita income (in USD thousands) in the county the submission comes from, with $\mu_{S_2}(x)$:

$$\mu_{S_2}(x) = \begin{cases} \frac{x-52}{6}, & 52 \leq x \leq 58 \\ \frac{-x+64}{6}, & 58 \leq x \leq 64 \end{cases} \tag{24}$$

and 0 otherwise. S_3 represents label "average time" in $\mathcal{X}_3 = \{0, 1, \ldots, 30\}$—numbers of days between receiving and sending the complaint to a company by CFPB (Consumer Financial Protection Bureau), with $\mu_{S_3}(x)$:

$$\mu_{S_3}(x) = \begin{cases} \frac{x-2}{4}, & 2 \leq x \leq 6 \\ \frac{-x+10}{4}, & 6 \leq x \in 10 \end{cases} \tag{25}$$

and 0 otherwise. S_4 is a non-fuzzy set representing one of the labels: {Older American, Servicemember, Older American and Servicemember, none}.

The relative linguistic quantifiers proposed to be applied in Algorithm 2, according to Definition 2, are Q_1 = "very few", Q_2 = "close to 0", Q_3 = "almost none". They are illustrated in Figure 2 and their membership functions for $r \in [0, 1] \subset \mathbb{R}$ are:

$$\mu_{Q_1}(r) = \begin{cases} 1 & r \leq 0.3 \\ \frac{-r+0.7}{0.4} & 0.3 \leq r < 0.7 \\ 0 & r \geq 0.7, \end{cases} \tag{26}$$

$$\mu_{Q_2}(r) = 9^{-r}, \tag{27}$$

$$\mu_{Q_3}(r) = \frac{1}{1 + \left(\frac{r}{0.3}\right)^4}. \tag{28}$$

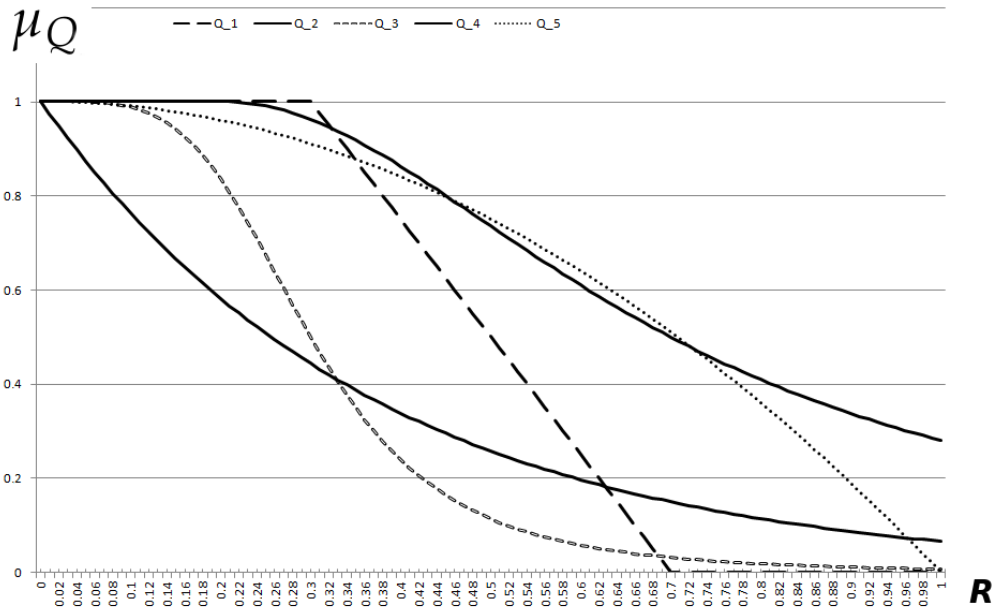

Figure 2. Membership functions of linguistic quantifiers Q_1 = "very few", Q_2 = "close to 0", Q_3 = "almost none", i.e., (26)–(28), respectively, used in the linguistic summaries (see Table 2).

Now we use Definitions 1 and 2 to detect outliers in $\mathcal{D}$. We use S and W as non-empty combinations of S_1, S_2, S_3, e.g., S_1 AND S_2, S_1 AND S_3, etc. Hence, queries R in the form of (18) and (20) are needed to fire Algorithms 1 and 2, e.g.:

$$\text{How many } (Q) \text{ complaints submitted in spring } (W) \text{ come from rich county } (S)? \tag{29}$$

Finally, 49 queries are formulated, and since we operate on 3 fuzzy quantifiers Q_1, Q_2, Q_3 substituting Q in (29), $3 \times 49 = 147$ linguistically quantified statements are generated see Table 2. The $\alpha = 0.9$ threshold is arbitrarily chosen to distinct statements

with the largest degree of truth (see Definition 2): statements 145. and 147. are found as sufficiently true to detect some outliers (lines bolded in Table 1). In both cases Algorithm 4 is used to determine particular outlying objects in $\mathcal{D}$, because the statements are in the form of (6). Two sets of outliers are finally recognized, $\mathcal{D}_{\text{out}_1}, \mathcal{D}_{\text{out}_2}$. Objects with IDs in $\mathcal{D}_{\text{out}_1} = \{801{,}691; 801{,}371; 375{,}975\}$ are outliers detected by statement 145, and objects with IDs in $\mathcal{D}_{\text{out}_2} = \{663{,}648; 210{,}516; 253{,}242; 673{,}669; 305{,}167; 716{,}577\}$ are outliers detected by statement 147. These choices were checked and confirmed by experts as outlying objects.

Table 2. Linguistically quantified statements 1.–147. generated with degrees of truth T and T_{supp}, T_{clm} measures (see (12) and (13)). The statements 145. and 147. that detected outliers, are bolded.

No.	Linguistically Quantified Statement	T	T_{supp}	T_{clm}
1.	Almost no complaints submitted in early spring come from rich county	0.75	0	0.33
2.	Almost no complaints submitted in middle spring come from rich county	0.19	0	0.33
42.	Almost none complaints submitted in winter are sent by CFPB in short time	0.58	0	0.33
60.	Very few complaints submitted in winter come from poor county	0.63	0.3	0.5
97.	Close to 0 complaints submitted in summer come from rich county	0.32	0	0.40
125.	Close to 0 complaints are submitted by Older American and Servicemember AND come from average county	0.25	0	0.40
...	...	...	...	...
144.	Close to 0 complaints submitted in early winter are sent by CFPB in long time	0.14	0	0.40
145.	**Almost no complaints submitted in early spring come from rich county AND are sent by CFPB in an average time**	**0.94**	**0**	**0.33**
146.	Close to 0 complaints submitted in early spring come from rich county AND are sent by CFPB in an average time	0.43	0	0.40
147.	**Very few complaints submitted in winter come from rich county AND are sent by CFPB in an average time**	**0.92**	**0.3**	**0.5**

Moreover, one interesting observation must be noticed here: in Table 2, linguistic expressions 145. and 146 have the same S ("come from rich county AND are sent by CFPB in an average time") and W ("submitted in early spring") properties, but different linguistic quantifiers ("almost none" and "close to 0", respectively) are used. As the result, the former is qualified as detecting possible outliers and the latter is not. Obviously, it depends on the membership functions of the quantifiers, so one may conclude that testing different fuzzy representations of expert knowledge appears crucial for final results of detection.

A Comparison to the LOF Algorithm

The $\mathcal{D}$ sequence containing 40,083 objects is now the entry for the LOF algorithm (Local Outlier Factor) for detecting outliers [26]. The Python libraries: scikit-learn [27] and pandas [28] are applied in computations. The sets of parameters for LOF and numbers of outliers detected are given in Table 3:

Table 3. Parameters of the LOF algorithm and number of outliers detected in the $\mathcal{D}$ dataset.

No.	Neighbors	Leafsize	Metric	Contamination	Number of Outliers
1.	20	30	Minkovsky	auto	3206
2.	50	30	Minkovsky	auto	2989
3.	50	100	Jaccard	auto	0
4.	20	30	dice	auto	0
5.	20	30	correlation	0.0050	132
6.	20	30	correlation	0.0012	33
7.	20	30	correlation	0.0004	11
8.	20	30	correlation	0.0002	6

Table 3 illustrates different parameters of LOF taken into account to analyze the given sequence of data. It must be underlined that only raw numerical data are analyzed, since there is no possibility to feed LOF with linguistically expressed knowledge. As it is seen, the number of outliers found by LOF varies from 0 to over 3000. Moreover, only the correlation used as a metric and very small contamination provide numbers of outliers similar to the fuzzy algorithms proposed. However, outliers found by the LOF algorithm (that does not use fuzzy sets) are different from the outliers detected by our algorithms, and the most probable explanation is that traditional algorithms do not use linguistically expressed knowledge. The conclusion is that mutual applying both methods, traditional and the one proposed, is worth considering, to recognize all outlying objects.

6. Conclusions

In this paper, we introduce a novel method of outlier detection and recognition in graph datasets, when only linguistic and/or imprecise knowledge is available to differ suspected objects on the background of regular, typical data. The method is applicable when no quantitative or measurable information is accessible (and, thus, outlier definitions by Aggarwal, Knorr, etc. would not work), but when it is possible to create fuzzy models, i.e., fuzzy representations of expert knowledge, based on raw numerical data (which is a common practice in fuzzy computations). Specific processing of graph databases is taken into account, to make it readable for fuzzy methods. An illustrative implementation example is provided to show how graph data can be processed by fuzzy representations of linguistic information and, finally, to point at particular objects as recognized outliers. In other words, we show how the question "which objects are outliers in $\mathcal{D}$?" can be answered, and not only "are there outliers in $\mathcal{D}$ or not?".

Algorithms 1 or 2 can confirm that outliers are present in $\mathcal{D}$, and the subsets of outlying observations $\mathcal{D}_{\text{out}}$ in the analyzed $\mathcal{D}$ are determined by Algorithms 3 or 4 taken into account the degrees of truth of linguistically quantified statements generated by Algorithms 1 or 2 as their side effects, see (19) and (21). Finally, we would like to underline that the approach proposed to the issue of detecting and recognizing outliers in datasets, especially, its novelty based on linguistically quantified statements interpreted in terms of fuzzy sets, were not applied, up to now, to graph datasets.

Currently, our further research on recognizing outliers is in progress, mostly using multi-subject linguistic summaries, cf. [29], and analyzing other non-relational databases.

Author Contributions: Conceptualization, A.N. and A.D.; methodology, A.D.; software, M.B.; validation, A.N., A.D. and M.B.; formal analysis, A.N. and A.D.; resources, M.B.; data curation and processing, M.B.; implementation, M.B.; writing—original draft preparation, A.N. and A.D.; writing—review and editing, A.N., A.D. and M.B.; supervision, A.N. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Publicly available datasets were analyzed in this study. This data can be found here: [20].

Acknowledgments: This publication was completed while the third author was the Doctoral Candidate in the International Doctoral School at Lodz University of Technology, Lodz, Poland.

Conflicts of Interest: The authors declare no conflict of interest.

Sample Availability: Samples of the compounds of the proposed algorithms are available from the authors.

References

1. Hawkins, D.M. *Identification of Outliers*; Springer: Cham, Switzerland, 1980; Volume 11.
2. Aggarwal, C.C.; Yu, P.S. Outlier detection for high dimensional data. In Proceedings of the ACM SIGMOD International Conference on Management of Data, Santa Barbara, CA, USA, 21–24 May 2001; Volume 30, pp. 37–46.
3. Knorr, E.M.; Ng, R.T.; Tucakov, V. Distance-based outliers: Algorithms and applications. *VLDB J.* **2000**, *8*, 237–253. [CrossRef]
4. Knox, E.M.; Ng, R.T. Algorithms for mining distancebased outliers in large datasets. In Proceedings of the International Conference on Very Large Data Bases, New York, NY, USA, 24–27 August 1998; pp. 392–403.
5. Aggarwal, C.C. *Outlier Analysis*; Springer: New York, NY, USA, 2013.
6. Barnett, V.; Lewis, T. *Outliers in Statistical Data*; Wiley: New York, NY, USA, 1994; Volume 3,
7. Knorr, E.M.; Ng, R.T. A Unified Notion of Outliers: Properties and Computation; In Proceedings of the KDD'97: Proceedings of the Third International Conference on Knowledge Discovery and Data Mining, Newport Beach, CA, USA, 14–17 August 1997; pp. 219–222.
8. Breunig, M.M.; Kriegel, H.P.; Ng, R.T.; Sander, J. LOF: Identifying density-based local outliers. In Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, Dallas, TX, USA, 16–18 May 2000; Volume 29, pp. 93–104.
9. Kriegel, H.P.; Kröger, P.; Schubert, E.; Zimek, A. LoOP: Local outlier probabilities. In Proceedings of the 18th ACM Conference on Information and Knowledge Management, Hong Kong, China, 2–6 November 2009; pp. 1649–1652.
10. Ramaswamy, S.; Rastogi, R.; Shim, K. Efficient algorithms for mining outliers from large data sets. In Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, Dallas, TX, USA, 16–18 May 2000; Volume 29, pp. 427–438.
11. Jiang, F.; Liu, G.; Du, J.; Sui, Y. Initialization of K-modes clustering using outlier detection techniques. *Inf. Sci.* **2016**, *332*, 167–183. [CrossRef]
12. Flanagan, K.; Fallon, E.; Connolly, P.; Awad, A. Network anomaly detection in time series using distance based outlier detection with cluster density analysis. In Proceedings of the 2017 Internet Technologies and Applications (ITA), Wrexham, UK, 12–15 September 2017; pp. 116–121.
13. Tran, L.; Fan, L.; Shahabi, C. Distance-based outlier detection in data streams. *VLDB Endow.* **2016**, *9*, 1089–1100. [CrossRef]
14. Aggarwal, C.C. Outlier Detection in Categorical, Text, and Mixed Attribute Data. In *Outlier Analysis*; Springer: New York, NY, USA, 2017; pp. 249–272.
15. Hodge, V.J.; Austin, J. A survey of outlier detection methodologies. *Artif. Intell. Rev.* **2004**, *22*, 85–126. [CrossRef]
16. Duraj, A. Outlier detection in medical data using linguistic summaries. In Proceedings of the 2017 IEEE International Conference on Innovations in Intelligent SysTems and Applications (INISTA), Gdynia, Poland, 3–5 July 2017; pp. 385–390.
17. Duraj, A.; Niewiadomski, A.; Szczepaniak, P.S. Outlier detection using linguistically quantified statements. *Int. J. Intell. Syst.* **2018**, *33*, 1858–1868. [CrossRef]
18. Duraj, A.; Niewiadomski, A.; Szczepaniak, P.S. Detection of outlier information by the use of linguistic summaries based on classic and interval-valued fuzzy sets. *Int. J. Intell. Syst.* **2019**, *34*, 415–438. [CrossRef]
19. Niewiadomski, A.; Duraj, A. Detecting and Recognizing Outliers in Datasets via Linguistic Information and Type-2 Fuzzy Logic. *Int. J. Fuzzy Syst.* **2020**, *23*, 878–889. [CrossRef]
20. Consumer Complaint Database. Available online: https://catalog.data.gov/dataset/consumer-complaint-database (accessed on 30 June 2020).
21. Zadeh, L.A. A computational approach to fuzzy quantifiers in natural languages. *Comput. Maths Appl.* **1983**, *9*, 149–184. [CrossRef]
22. De Luca, A.; Termini, S. A definition of the non-probabilistic entropy in the setting of fuzzy sets theory. *Inf. Control* **1972**, *20*, 301–312. [CrossRef]
23. Niewiadomski, A. *Methods for the Linguistic Summarization of Data: Applications of Fuzzy Sets and Their Extensions*; Academic Publishing House EXIT: Warsaw, Poland, 2008.
24. Niewiadomski, A. A Type-2 Fuzzy Approach to Linguistic Summarization of Data. *IEEE Trans. Fuzzy Syst.* **2008**, *16*, 198–212. [CrossRef]
25. Neo4j: Graph Database Platform | Graph Database. Available online: https://neo4j.com (accessed on 30 June 2021).
26. Schubert, E.; Zimek, A.; Kriegel, H.P. Local outlier detection reconsidered: A generalized view on locality with applications to spatial, video, and network outlier detection. *Data Min. Knowl. Discov.* **2012**, *28*, 190–237. [CrossRef]
27. Scikit-Learn: Machine Learning in Python. Available online: https://scikit-learn.org (accessed on 20 July 2021).

28. Pandas—Python Data Analysis Library. Available online: https://pandas.pydata.org (accessed on 20 July 2021).
29. Niewiadomski, A.; Superson, I. Multi-Subject Type-2 Linguistic Summaries of Relational Databases. In *Frontiers of Higher Order Fuzzy Sets*; Sadeghian, A., Tahayori, H., Eds.; Springer: New York, NY, USA, 2015; pp. 167–181.