Model-based Outlier Detection for Object-Relational Data

Fatemeh Riahi
School of Computing Science
Simon Fraser University
Burnaby, Canada
sriahi@sfu.ca

Oliver Schulte
School of Computing Science
Simon Fraser University
Burnaby, Canada
oschulte@cs.sfu.ca

***Abstract*—This paper extends unsupervised statistical outlier detection to the case of object-relational data. Object-relational data represent a complex heterogeneous network [9], which comprises objects of different types, links among these objects, also of different types, and attributes of these links. This special structure prohibits a direct vectorial data representation. We apply state-of-the-art probabilistic modelling techniques for object-relational data that construct a graphical model (Bayesian network), which compactly represents probabilistic associations in the data. We propose a new metric, based on the learned object-relational model, that quantifies the extent to which the individual association pattern of a potential outlier deviates from that of the whole population. The metric is based on *the likelihood ratio* of two parameter vectors: One that represents the population associations, and another that represents the individual associations. Our method is validated on synthetic datasets and on real-world data sets about soccer matches and movies. Compared to baseline methods, our novel transformed likelihood ratio achieved the best detection accuracy on all datasets.**

I. Introduction

Outlier detection is an important data analysis task in many domains. Statistical approaches to unsupervised outlier detection are based on a generative model of the data [2]. The generative model represents normal behavior. An individual object is deemed an outlier if the model assigns sufficiently low likelihood to generating it. We propose a new method for extending statistical outlier detection to the case of object-relational data using a novel likelihood-ratio comparison for probabilistic models.

The object-relational data model is one of the main data models for structured data [15]. The main characteristics of objects that we utilize in this paper are the following. (1) *Object Identity.* Each object has a unique identifier that is the same across contexts. For example, a player has a name that identifies him in different matches. (2) *Class Membership.* An object is an instance of a class, which is a collection of similar objects. Objects in the same class share a set of attributes. For example, van Persie is a player object that belongs to the class striker, which is a subclass of the class player. (3) *Object Relationships.* Objects are linked to other objects. Both objects and their links have attributes. A common type of object relationship is a component relationship between a complex object and its parts. For example, a match links two teams, and each team comprises a set of players for that match. A difference between relational and vectorial data is therefore that an individual object is characterized not only by a list of attributes, but also by its links and by attributes of the object linked to it. We refer to the substructure comprising this information as the *object data*.

a) Approach: A class-model Bayesian network (BN) structure is learned with data for the entire population. The nodes in the BN represent attributes for links, of multiple types, and attributes of objects, also of multiple types. To learn the BN model, we apply techniques from statistical-relational learning, a recent field that combines AI and machine learning [10], [27], [7]. Given a set of parameter values and an input database, it is possible to compute a *class model likelihood* that quantifies how well the BN fits the object data. The class model likelihood uses BN parameter values *estimated from the entire class data.* This is a relational extension of the standard log-likelihood method for i.i.d. vectorial data, which uses the likelihood of a data point as its outlier score. While the class model likelihood is a good baseline score, it can be improved by comparing it to *the object model likelihood*, which uses BN parameter values *estimated from the object data.* The *model log-likelihood ratio* (LR) is the log-ratio of the object model likelihood to the class model likelihood. This ratio quantifies how the probabilistic associations that hold in the general population deviate from the associations in the object data substructure. While the likelihood ratio discriminates relational outliers better than the class model likelihood alone, it can be improved further by applying two transformations: (1) a mutual information decomposition, and (2) replacing log-likelihood differences by log-likelihood distances. We refer to the resulting novel score as the *log-likelihood distance*.

b) Evaluation: Our code and datasets are available online at [24]. Our performance evaluation follows the design of previous outlier detection studies [9], [2], where the methods are scored against a test set of known outliers. We use three synthetic and two real-world datasets, from the UK Premier Soccer League and the Internet Movie Database (IMDb). On the synthetic data we have known ground truth. For the real-world data, we use a one-class design, where one object class is designated as normal and objects from outside the class are the outliers. For example, we compare goalies as outliers against the class of strikers as normal objects. On all datasets, the log-likelihood distance metric achieves the best detection accuracy compared to baseline methods.

978-1-4799-7560-0/15/$31

We also offer case studies where we assess whether individuals that our score ranks as highly unusual in their class are indeed unusual. The case studies illustrate that our outlier score is *easy to interpret*, because the Bayesian network provides a sum decomposition of the data distributions by features. Interpretability is very important for users of an outlier detection method as there is often no ground truth to evaluate outliers suggested by the method.

c) Contributions: Our main contributions may be summarized as follows.

1) The first approach to outlier detection for structured data that is based on a probabilistic model.
2) A new model-based outlier score based on a novel model likelihood comparison, the log-likelihood distance.

d) Paper Organization: We review background about Bayesian networks for relational data. Then we introduce our novel log-likelihood distance outlier score. After presenting the details of our approach, we review related work. Empirical evaluation compares model-based and aggregation-based approaches to relational outlier detection, with respect to three synthetic and three real-world problems.

II. Background: Bayesian Networks for Relational Data

We adopt the Parametrized Bayes net (PBN) formalism [22] that combines Bayes nets with logical syntax for expressing relational concepts.

A. Bayesian Networks

A **Bayesian Network (BN)** is a directed acyclic graph (DAG) whose nodes comprise a set of random variables [20]. Depending on context, we interchangeably refer to the nodes and variables of a BN. Fix a set of variables $\boldsymbol{V} = \{V_1, \ldots, V_n\}$. The possible values of V_i are enumerated as $\{v_{i1}, \ldots, v_{ir_i}\}$. The notation $P(V_i = v) \equiv P(v)$ denotes the probability of variable V_i taking on value v. We also use the vector notation $P(\boldsymbol{V} = \mathbf{v}) \equiv P(\mathbf{v})$ to denote the joint probability that each variable V_i takes on value $\mathbf{v}_i$.

The conditional probability parameters of a Bayesian network specify the distribution of a child node given an assignment of values to its parent node. For an assignment of values to its nodes, a BN defines the joint probability as the product of the conditional probability of the child node value given its parent values, for each child node in the network. This means that the log-joint probability can be *decomposed* as the node-wise sum

$$\ln P(\boldsymbol{V} = \mathbf{v}; B, \boldsymbol{\theta}) = \sum_{i=1}^{n} \ln \theta(\mathbf{v}_i | \mathbf{v}_{\mathbf{pa}_i}) \qquad (1)$$

where $\mathbf{v}_i$ resp. $\mathbf{v}_{\mathbf{pa}_i}$ is the assignment of values to node V_i resp. the parents of V_i determined by the assignment $\mathbf{v}$. To avoid difficulties with $\ln(0)$, here and below we assume that joint distributions are positive everywhere. Since the parameter values for a Bayes net define a joint distribution over its nodes, they therefore entail a marginal, or unconditional, probability for a single node. We denote the **marginal probability** that node V has value v as $P(V = v; B, \boldsymbol{\theta}) \equiv \theta(v)$.

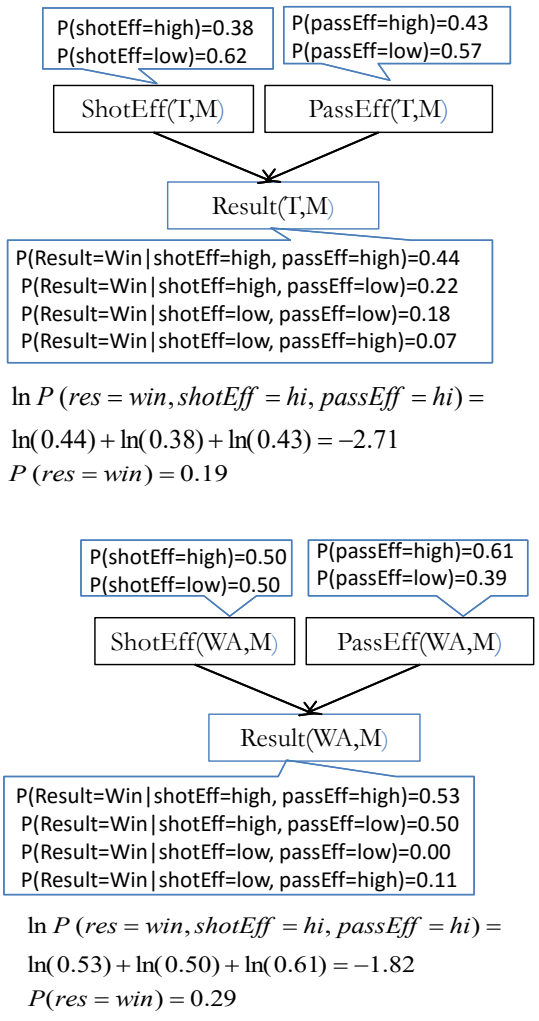

Fig. 1. Example of joint and marginal probabilities computed from a toy Bayesian network structure. The parameters were estimated from the Premier League dataset. (Top): A class model Bayesian network B_c for all teams with class parameters $\boldsymbol{\theta}_c$. (Bottom): The same Bayesian network structure with object parameters $\boldsymbol{\theta}_o$ learned for Wigan Athletics ($T = WA$). Our model-based methods outlier scores compare the data likelihood of the class parameters and the object parameters.

e) Example.: Figure 1 shows an example of a Bayesian network and associated joint and marginal probabilities.

B. Relational Data

We adopt a functor-based notation for combining logical and statistical concepts [22], [13]. A functor is a function or predicate symbol. Each functor has a set of values (constants) called the **domain** of the functor. The domain of a **predicate** is $\{T, F\}$. Predicates are usually written with uppercase Roman letters, other terms with lowercase letters. A predicate of arity at least two is a **relationship** functor. Relationship functors specify which objects are linked. Other functors represent

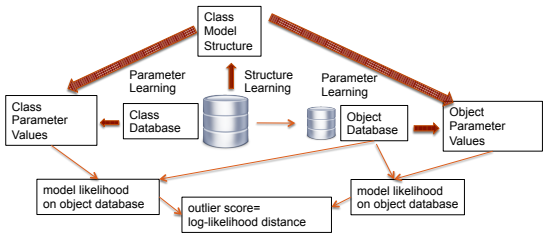

Fig. 2. Computation of outlier score.

features or **attributes** of an object or a tuple of objects (i.e., of a relationship). A **population** is a set of objects. A **term** is of the form $f(\sigma_1, \ldots, \sigma_k)$ where f is a functor and each σ_i is a first-order variable or a constant denoting an object. A term is **ground** if it contains no first-order variables; otherwise it is a first-order term. In the context of a statistical model, we refer to first-order terms as **Parametrized Random Variables** (PRVs) [13]. A **grounding** replaces each first-order variable in a term by a constant; the result is a ground term. A grounding may be applied simultaneously to a set of terms. A relational database $\mathcal{D}$ specifies the values of all ground terms, which can be listed in data tables.

Consider a joint assignment $P(\boldsymbol{V} = \mathbf{v})$ of values to a set of PRVs $\boldsymbol{V}$. The *grounding space* of the PRVs is the set of all possible grounding substitutions, each applied to all PRVs in $\boldsymbol{V}$. The *count* of groundings that satisfy the assignment with respect to a database $\mathcal{D}$ is denoted by $\#_{\mathcal{D}}(\boldsymbol{V} = \mathbf{v})$. The **database frequency** $P_{\mathcal{D}}(\boldsymbol{V} = \mathbf{v})$ is the grounding count divided by the number of all possible groundings.

Example. The Opta dataset represents information about premier league data (Sec. VI-B). The basic populations are teams, players, matches, with corresponding first-order variables T, P, M. Table I specifies values for some ground terms. The first three column headers show first-order variables ranging over different populations. The remaining columns represent features. Table III illustrates grounding counts. Counts are based on the 2011-2012 Premier League Season. We count only groundings $(team, match)$ such that $team$ plays in $match$. Each team, including Wigan Athletics, appears in 38 matches. The total number of team-match pairs is $38 \times 20 = 760$.

TABLE I. SAMPLE POPULATION DATA TABLE (SOCCER).

MatchId *M*	TeamId *T*	PlayerId *P*	First_goal(*P,M*)	TimePlayed(*P,M*)	ShotEff(*T,M*)	result(*T,M*)
117	WA	McCarthy	0	90	0.53	*win*
148	WA	McCarthy	0	85	0.57	*loss*
15	MC	Silva	1	90	0.59	*win*
…	…	…	…	…	…	

TABLE II. SAMPLE OBJECT DATA TABLE, FOR TEAM $T = WA$.

MatchId *M*	TeamId $T = WA$	PlayerId *P*	First_goal(*P,M*)	TimePlayed(*P,M*)	ShotEff(*WA,M*)	result(*WA,M*)
117	WA	McCarthy	0	90	0.53	*win*
148	WA	McCarthy	0	85	0.57	*loss*
…	WA	…	…	…	…	

A novel aspect of our paper is that we learn model parameters for specific objects as well as for the entire population. The appropriate **object data table** is formed from the population data table by restricting the relevant first-order variable to the target object. For example, the object database for target Team $WiganAthletic$, forms a subtable of the data table of Table I that contains only rows where TeamID = WA; see Table II. In database terminology, an object database is like a view centered on the object. The object database is an individual-centered representation [8].

TABLE III. EXAMPLE OF GROUNDING COUNT AND FREQUENCY IN PREMIER LEAGUE DATA, FOR THE CONJUNCTION $passEff(T, M) = hi, shotEff(T, M) = hi, Result(T, M) = win$.

Database	Count or $\#_D(\boldsymbol{V} = \mathbf{v})$	Frequency or $P_D(\boldsymbol{V} = \mathbf{v})$
Population	76	$76/760 = 0.10$
Wigan Athletics	7	$7/38 = 0.18$

C. Bayesian Networks for Relational Data

A **Parametrized Bayesian Network Structure** (PBN) is a Bayesian network structure whose nodes are PRVs. The relationships and features in an object database define a set of nodes for Bayes net learning; see Figure 1.

1) Model Likelihood for Parametrized Bayesian Networks: A standard method for applying a generative model assumes that the generative model represents normal behavior since it was learned from the entire population. An object is deemed an outlier if the model assigns sufficiently low likelihood to generating its features [5]. This likelihood method is an important baseline for our investigation. Defining a likelihood for relational data is more complicated than for i.i.d. data, because an object is characterized not only by a feature vector, but by an object database. We employ the relational pseudo log-likelihood [26], which can be computed as follows for a given Bayesian network and database.

$$LOG(\mathcal{D}, B, \boldsymbol{\theta}) = \sum_{i=1}^{n} \sum_{j=1}^{r_i} \sum_{\mathbf{pa}_i} P_{\mathcal{D}}(v_{ij}, \mathbf{pa}_i) \ln \theta(v_{ij}|\mathbf{pa}_i) \quad (2)$$

Equation (2) represents the standard BN log-likelihood function for the object data [6], except that parent-child instantiation counts are standardized to be proportions [26]. The equation can be read as follows.

1) For each parent-child configuration, use the conditional probability of the child given the parent.
2) Multiply the logarithm of the conditional probability by the database frequency of the parent-child configuration.
3) Sum this product over all parent-child configurations and all nodes.

Schulte proves that the maximum of the pseudo-likelihood (2) is given by the empirical database frequencies [26, Prop.3.1.]. In all our experiments we use these maximum likelihood parameter estimates.

Example. The family configuration

$$passEff(T, M) = hi, shotEff(T, M) = hi, Result(T, M) = win$$

contributes one term to the pseudo log-likelihood for the BN of Figure 1. For the population database, this term is $0.1 \times \ln(0.44) = -0.08$. For the Wigan Athletics database, the term is $0.18 \times \ln(0.44) = -0.14$.

III. Likelihood-Distance Object Outlier score

We introduce a novel model-based outlier score, that extends the log-likelihood, using the following notation.

- $\mathcal{D}_C$ is the database for the entire class of objects; cf. Table I. This database defines the **class distribution** $P_C \equiv P_{\mathcal{D}_c}$.
- $\mathcal{D}_o$ is the restriction of the input database to the target object; cf. Table II. This database defines the **object distribution** $P_o \equiv P_{\mathcal{D}_o}$.
- B_C is a model (e.g., Bayesian network) learned with $\mathcal{D}_\mathcal{P}$ as the input database; cf. Figure 1(a).
- $\boldsymbol{\theta}_C$ resp. $\boldsymbol{\theta}_o$ are parameters learned for B_C using $\mathcal{D}_c$ resp. $\mathcal{D}_o$ as the input database.

Figure 2 illustrates these concepts and the system flow for computing an outlier score. First, we learn a Bayesian network B_C for the entire population using a previous learning algorithm (see Section VI-C below). We then evaluate *how well the class model fits the target object data.* For vectorial data, the standard model fit metric is the log-likelihood of the target *datapoint*. For relational data, the counterpart is the relational log-likelihood (2) of the target *database*:

$$LOG(\mathcal{D}_o, B_C, \boldsymbol{\theta}_C). \tag{3}$$

While this is a good baseline outlier score, it can be improved by considering scores based on the likelihood ratio, or **log-likelihood difference**:

$$LR(\mathcal{D}_o, B_C, \boldsymbol{\theta}_o) \equiv LOG(\mathcal{D}_o, B_C, \boldsymbol{\theta}_o) - LOG(\mathcal{D}_o, B_C, \boldsymbol{\theta}_C). \tag{4}$$

The log-likelihood difference compares how well the class-level parameters fit the object data, vs. how well the object parameters fit the object data. In terms of the conditional probability parameters, it measures how much the log-conditional probabilities in the class distribution differ from those in the object distribution. Note that this definition applies only for relational data where an individual is characterized by a substructure rather than a "flat" feature vector. Assuming maximum likelihood parameter estimation, LR is equivalent to the Kullback-Leibler divergence between the class-level and object-level parameters [6]. While the LR score provides more outlier information than the model log-likelihood, it can be improved further by two transformations as follows. (1) Decompose the joint probability into a single-feature component and a mutual information component. (2) Replace log-likelihood differences by log-likelihood distances. The resulting score is the **log-likelihood distance** (ELD), which is the main novel score we propose in this paper. Formally it is defined as follows for each feature i. The total score is the sum of feature-wise scores. Section IV below provides example computations.

$$\begin{aligned} ELD_i = &\textstyle\sum_{j=1}^{r_i} P_o(v_{ij}) \left|\ln \frac{\theta_o(v_{ij})}{\theta_C(v_{ij})}\right| + \\ &\textstyle\sum_{j=1}^{r_i} \sum_{\mathbf{pa}_i} P_o(v_{ij}, \mathbf{pa}_i) \left|\ln \frac{\theta_o(v_{ij}|\mathbf{pa}_i)}{\theta_o(v_{ij})} - \ln \frac{\theta_C(v_{ij}|\mathbf{pa}_i)}{\theta_C(v_{ij})}\right|. \end{aligned} \tag{5}$$

The first sum is the **single-feature** component, where each feature is considered independently of all others. It computes the expected log-distance with respect to the singe feature value probabilities between the object and the class models. The second ELD sum is the **mutual information component**, based on the mutual information among all features. It computes the expected log-distance between the object and the class models with respect to the mutual information of feature value assignments. Intuitively, the first sum measures how the models differ if we treat each feature in isolation. The second sum measures how the models differ in terms of how strongly parent and child features are associated with each other.

A. Motivation

The motivation for the mutual information decomposition is two-fold.

(1) *Interpretability*, which is very important for outlier detection. The single-feature components are easy to interpret since they involve no feature interactions. Each parent-child local factor is based on the average relevance of parent values for predicting the value of the child node, where relevance is measured by $\ln(\theta(v_{ij}|\mathbf{pa}_i)/\theta(v_{ij}))$. This relevance term is basically the same as the widely used lift measure [29], therefore an intuitively meaningful quantity. The ELD score compares how relevant a given parent condition is in the object data with how relevant it is in the general class.

(2) *Avoiding cancellations.* Each term in the log-likelihood difference (4) decomposes into a relevance difference and a marginal difference:

$$\ln \frac{\theta_o(v_{ij}|\mathbf{pa}_i)}{\theta_C(v_{ij}|\mathbf{pa}_i)} = \ln \frac{\theta_o(v_{ij}|\mathbf{pa}_i)}{\theta_o(v_{ij})} - \ln \frac{\theta_C(v_{ij}|\mathbf{pa}_i)}{\theta_C(v_{ij})} + \ln \frac{\theta_o(v_{ij})}{\theta_C(v_{ij})}. \tag{6}$$

These differences can have different signs for different child-parent configurations and cancel each other out; see Table IV. Taking distances as in Equation 5 avoids this undesirable cancellation. Since our goal is to assess the distinctness of an object, *we do not want differences to cancel out.* The general point is that averaging differences is appropriate when considering costs, or utilities, but not appropriate for assessing the distinctness of an object. For instance, the average of both vectors (0,0) and (1,-1) is 0, but their distance is not.

B. Comparison Outlier Scores

Our lesion study compares our log-likelihood distance ELD score to baselines that are defined by omitting a component of ELD. In this section we define these scores. The scores increase in sophistication in the sense that they apply more transformations of the log-likelihood ratio. More sophisticated scores provide more information about outliers. Table IV defines local feature scores; the total score is the sum of feature-wise scores. All metrics are defined such that *a higher score indicates a greater anomaly.* The metrics are as follows.

Feature Divergence FD is the first component of the ELD score. It considers each feature independently (no feature correlations).

TABLE IV. BASELINE OUTLIER SCORES FOR BAYESIAN NETWORKS

Method	Formula
FD_i	$\sum_{i=1}^{n}\sum_{j=1}^{r_i} P_o(v_{ij})\left\lvert\ln\frac{\theta_o(v_{ij})}{\theta_C(v_{ij})}\right\rvert$
$-LOG_i$	$-\sum_{i=1}^{n}\sum_{j=1}^{r_i}\sum_{\mathbf{pa}_i} P_o(v_{ij},\mathbf{pa}_i)\ln\theta_C(v_{ij}\lvert\mathbf{pa}_i)$
LR_i	$\sum_{j=1}^{r_i}\sum_{\mathbf{pa}_i} P_o(v_{ij},\mathbf{pa}_i)\ln\frac{\theta_o(v_{ij}\lvert\mathbf{pa}_i)}{\theta_C(v_{ij}\lvert\mathbf{pa}_i)}$.
$\lvert LR_i\rvert$	$\sum_{j=1}^{r_i}\sum_{\mathbf{pa}_i} P_o(v_{ij},\mathbf{pa}_i)\lvert\ln\frac{\theta_o(v_{ij}\lvert\mathbf{pa}_i)}{\theta_C(v_{ij}\lvert\mathbf{pa}_i)}\rvert$.
LR_i^+	$\sum_{j=1}^{r_i} P_o(v_{ij})\ln\frac{\theta_o(v_{ij})}{\theta_C(v_{ij})}+$ $\sum_{j=1}^{r_i}\sum_{\mathbf{pa}_i} P_o(v_{ij},\mathbf{pa}_i)\ln\frac{\theta_o(v_{ij}\lvert\mathbf{pa}_i)}{\theta_o(v_{ij})}-\ln\frac{\theta_C(v_{ij}\lvert\mathbf{pa}_i)}{\theta_C(v_{ij})}$.

Log-Likelihood Score *LOG* is the standard model-based outlier detection score using data likelihood.

Log-Likelihood Difference *LR* is the log-likelihood difference (4) between the class-level and object-level parameters.

Log-Likelihood Difference with absolute value $|LR|$ replaces differences in LR by distances.

Log-Likelihood Difference with decomposition LR^+ applies a mutual information decomposition to LR.

IV. EXAMPLES

We provide three simple examples with only two features that illustrate the computation of the outlier scores. They are designed so that outliers and normal objects are easy to distinguish, and so that it is easy to trace the behavior of an outlier score. The examples therefore serve as thought experiments that bring out the strengths and weaknesses of model-based outlier scores. Figure 3 describes the BN representation of the examples. Table V provides the computation of the scores. For intuition, we can think of a soccer setting, where each match assigns a value to each attribute $F_i, i = 1, 2$ for each player. Scores for the F_2 feature are computed conditional on $F_1 = 1$. Expectation terms are computed first for $F_2 = 1$, then $F_2 = 0$.

The single feature distributions are uniform, so the feature component ELD_1 is 0 for each node in both examples. The table illustrates the undesirable cancelling effects in LR. In the high correlation scenario 3(a), the outlier object has a lower probability than the normal class distribution of $Match\_Result = 0$ given that $Shot\_Efficiency = 1$. Specifically, 0.5 vs. 0.9. The outlier object exhibits a higher probability $Match\_Result = 1$ than the normal class distribution, conditional on $Shot\_Efficiency = 1$; specifically, 0.5 vs. 0.1. In line 1, column 2 of Table V the log-ratios $\ln(0.5/0.9)$ and $\ln(0.5/0.1)$ therefore have different signs. In the low correlation scenario 3(b), the cancelling occurs in the same way, but with the normal and outlier probabilities reversed. The cancelling effect is even stronger for attributes with more than two possible values.

V. RELATED WORK

Outlier detection is a densely researched field, for a survey please see [2]. Figure 4 provides a tree picture of where our method is situated with respect to other outlier detection methods and other data models. Our method falls in the category of *unsupervised* statistical model-based approaches. To our knowledge, ours is the first model-based method tailored for object-relational data. Like other model-based approaches, it detects *global outliers.* Aggarwal [2] defines a global outlier to be a data point that notably deviates from the rest of the population. We review relevant approaches from different data models, the most common atomic object model—where data is represented by vectors—and structured data models.

TABLE V. EXAMPLE COMPUTATION OF DIFFERENT OUTLIER SCORES.

Score	$F1 = 1$ Computation	$F2 \mid F1 = 1$ Computation	Result
LR	$1/2\ln(0.5/0.5) = 0$	$1/4\ln(0.5/0.9) + 1/4\ln(0.5/0.1)$	0.36
$\lvert LR\rvert$	0 (no parents)	$1/4\lvert\ln(0.5/0.5) - \ln(0.9/0.5)\rvert + 1/4\lvert\ln(0.5/0.5) - \ln(0.1/0.5)\rvert$	0.79
FD	$\lvert\ln(0.5/0.5)\rvert = 0$	$1/2\lvert\ln(0.5/0.5)\rvert + 1/2\lvert\ln(0.5/0.5)\rvert$	0
ELD	$0 + 0$	$0.79 + FD$	0.79

Table V(a): High Correlation Case, Figure 3(a).

Score	$F1 = 1$ Computation	$F2 \mid F1 = 1$ Computation	Result
LR	$1/2\ln(0.5/0.5) = 0$	$0.5 \cdot 0.9\ln(0.9/0.5) + 0.5 \cdot 0.1\ln(0.1/0.5)$	0.26
$\lvert LR\rvert$	0 (no parents)	$0.5 \cdot 0.9\lvert\ln(0.9/0.5) - \ln(0.5/0.5)\rvert + 0.5 \cdot 0.1\lvert\ln(0.1/0.5) - \ln(0.5/0.5)\rvert$	0.50
FD	$\lvert\ln(0.5/0.5)\rvert = 0$	$1/2\lvert\ln(0.5/0.5)\rvert + 1/2\lvert\ln(0.5/0.5)\rvert$	0
ELD	$0 + 0$	$0.5 + FD$	0.5

Table V(b): Low Correlation Case. Figure 3(b).

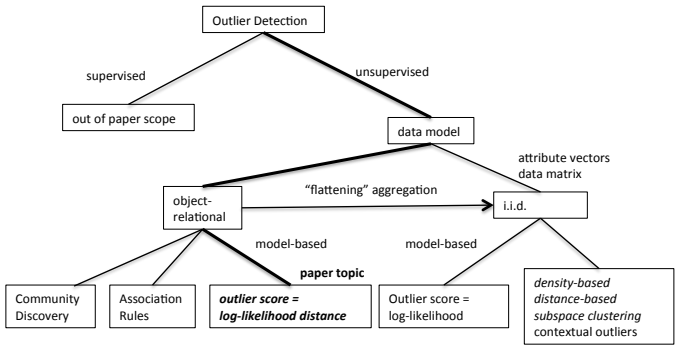

Fig. 4. A tree structure for related work on outlier detection for structured data. A path specifies an outlier detection problem, the leaves list major approaches to the problem. Approaches in italics appear in experiments.

a) Attribute Vector Data Model: By far most work on outlier detection considers atomic objects with flat feature vectors. This leads to an impedance mismatch: The required input format for these outlier detection methods is a single data matrix, not a structured dataset. For example, one cannot provide a relational database as input. This mismatch is not simply a question of choosing a file format, but instead reflects a different underlying data model: complex objects with both attributes and component objects vs. atomic objects with attributes only. It is possible to "flatten" structured data by converting it to unstructured feature vectors, for instance by using aggregate functions. We evaluated the aggregation approach in this paper by applying three standard methods for outlier detection.

Work on atomic contextual outliers [28] is like ours in that it considers the distinctness of a target individual from a reference class. A reference class is not specified for each object, but is constructed as part of outlier detection. Our work could be combined with a class discovery approach by providing a score of how informative the inferred classes are.

b) Structured Data Models: We discuss related techniques in three types of structured data models: SQL (relational), XML (hierarchical), and OLAP (multi-dimensional).

For relational data, many outlier detection approaches aim to discover rules that represent the presence of anomalous

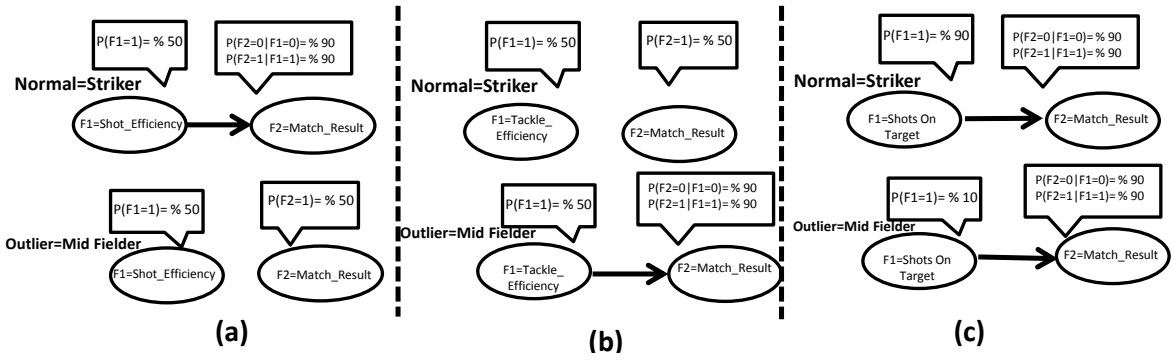

Fig. 3. Illustrative Bayesian networks. The networks are not learned from data, but hand-constructed to be plausible for the soccer domain. (a) High Correlation: Normal individuals exhibit a strong association between their features, outliers no association. Both normals and outliers have a close to uniform distribution over single features. (b) Low Correlation: Normal individuals exhibit no association between their features, outliers have a strong association. Both normals and outliers have a close to uniform distribution over single features. (c) Single Attributes: Both normal and outlier individuals exhibit a strong association between their features. In normals, 90% of the time, feature 1 has value 0. For outliers, feature 1 has value 0 only 10% of the time.

associations for an individual or the absence of normal associations [16], [9]. The survey by [19] unifies within a general rule search framework related tasks such as exception mining, which looks for associations that characterize unusual cases, subgroup mining, which looks for associations characterizing important subgroups, and contrast space mining, which looks for differences between classes. Another rule-based approach uses Inductive Logic Programming techniques [3]. While local rules are informative, they are not based on a global statistical model and do not provide a single outlier score for each individual.

A latent variable approach in information networks ranks potential outliers in reference to the latent communities inferred by network analysis [9]. Our model aggregates information from entities and links of different types, but does not assume that different communities have been identified.

Koh *et al.* [14] propose a method for hierarchical structures represented in XML document trees. Their aim is to identify feature outliers, not class outliers as in our work. Also, they use aggregate functions to convert the object hierarchy into feature vectors. Their outlier score is based on local correlations, and they do not construct a model.

The multi-dimensional data model defines numeric measures for a set of dimensions. The differences in the two data models mean that multi-dimensional outlier detection models [25] do not carry over to object-relational outlier detection. (1) The object data model allows but does not require any numeric measures. In our datasets, all features are discrete. Nor do we assume that it is possible to aggregate numeric measures to summarize lower-level data at higher levels. (2) In scoring a potential outlier object, our method considers other objects *both* below and above the target object in the component hierarchy. OLAP exploration methods consider only cells below or at the same level as the target cell. For example, in scoring a player, our method would consider features of the player's team. Also, the ELD outlier score of an object is not determined by the outlier scores of its components, in contrast to the approach of Sarawagi *et al.*. (3) Our approach models a joint distribution over features, exploiting correlations among features. Most of the OLAP-based methods consider only a single numeric measure at a time, not a joint model.

VI. Experimental Design

All the experiments were performed on a 64-bit Centos machine with 4GB RAM and an Intel Core i5-480 M processor. The likelihood-based outlier scores were computed with SQL queries using JDBC, JRE 1.7.0. and MySQL Server version 5.5.34. We describe the datasets used in our experiments.

A. Synthetic Datasets

We generated three synthetic datasets with normal and outlier players using the distributions represented in the three Bayesian networks of Figure 3. Each player participates in 38 matches, similar to the real-world data. The main goal of designing synthetic experiments is to test the methods on easy to detect outliers. Each match assigns a value to each feature $F_i, i = 1, 2$ for each player.

High Correlation See Figure 3(a).
Low Correlation See Figure 3(b).
Single features See Figure 3(c).

We used the $mlbench$ package in R to generate synthetic features in matches, following these distributions for 240 normal players and 40 outliers. We followed the real-world Opta data in terms of number of normal and outlier individuals. The scores are used to rank all 280 players.

B. Real-World Datasets

Data tables are prepared from Opta data [17] and IMDb [11]. Our datasets and code are available on-line [12].

a) Soccer Data: The Opta data were released by Manchester City. It lists box scores, that is, counts of all the ball actions within each game by each player, for the 2011-2012 season. For each player in a match, our data set contains eleven player features. For each team in a match, there are five features computed as player feature aggregates, as well as the team formation and the result (win, tie, loss). There are two relationships, $Appears\_Player(P, M)$, $Appears\_Team(T, M)$.

TABLE VI. OUTLIER/NORMAL OBJECTS IN REAL-WORLD DATASETS.

Normal	#Normal	Outlier	#Outlier
Striker	153	Goalie	22
Midfielder	155	Striker	74
Drama	197	Comedy	47

b) IMDB Data: The Internet Movie Database (IMDB) is an on-line database of information related to films, television programs, and video games. The IMDB website offers a dataset containing information on cast, crew, titles, technical details and biographies into a set of compressed text files. We preprocessed the data like [21] to obtain a database with seven tables: one for each population and one for the three relationships $Rated(User, Movie)$, $Directs(Director, Movie)$, and $ActsIn(Actor, Movie)$.

In real-world data, there is no ground truth about which objects are outliers. To address this issue, we employ a one-class design: we learn a model for the class distribution, with data from that class only. Then we rank all individuals from the normal class together with all objects from a contrast class treated as outliers, to test whether an outlier score recognizes objects from the contrast class as outliers. Table VI shows the normal and contrast classes for three different datasets. In-class outliers are possible, e.g. unusual strikers are still members of the striker class. Our case studies describe a few in-class outliers. In the soccer data, we considered only individuals who played more than 5 matches out of a maximum 38.

C. Methods Compared

We compare two types of approaches, and within each approach several outlier detection methods. The first approach evaluates the likelihood-based outlier scores described in Section III. For relational Bayesian network structure learning we utilize the previous learn-and-join algorithm (LAJ), which is a state-of-the-art BN structure learning method for relational data [27]. The LAJ algorithm employs an iterative deepening strategy, which can be described as as search through a lattice of table joins. For each table join, different BNs are learned and the learned edges are propagated from smaller to larger table joins. For a full description, complexity analysis, and learning time measurements, please see [27]. We used the implementation of the LAJ algorithm due to its creators [12].

The second approach first "flattens" the structured data into a matrix of feature vectors, then applies standard matrix-based outlier detection methods. We refer to such methods as **aggregation-based** (cf. Figures 4). For example, this was the approach taken by Breunig *et al.* for identifying anomalous players in sports data [4]. Following their paper, for each continuous feature in the object data, we use the average over its values, and for each discrete feature, we use the occurrence count of each feature value in the object data. Aggregation tends to lose information about correlations. Our experiments address the empirical question of whether this loss of information affects outlier detection. We evaluated three standard matrix-based outlier detection methods: Density-based LOF [4], distance-based $KNNOutlier$ [23] and subspace analysis $OutRank$ [18]. These represent common, fundamental approaches for vectorial data. Like ELD, subspace analysis is sensitive to correlations among features. We used the available implementation of all three data matrix methods from the state of the art data mining software *ELKI* [1]. We used *PRO-CLUS* as the clustering function for $OutRank$, recommended by [18].

VII. EMPIRICAL RESULTS

We present results regarding computational feasibility, predictive performance, and case studies.

1) Computational Cost of the ELD Score.: Table VII shows that the computation of the ELD value for a given target object is feasible. On average, it takes a quarter of a minute for each soccer player, and one minute for each movie. This includes the time for parameter learning from the object database. Learning the class model BN takes longer, but needs to be done only once for the entire object class. *The BN model provides a crucial low-dimensional representation of the distribution information in the data.* Table VIII compares the number of terms required to compute the ELD score in the BN representation to the number of terms in an unfactored representation with one parameter for each joint probability.

TABLE VII. TIME (MIN) FOR COMPUTING THE ELD SCORE.

Dataset	Class Model	Average per Object
Strikers vs. Goalies	4.14	0.25
Midfielder vs. Goalies	4.02	0.25
Drama vs. Comedy	8.30	1.00

TABLE VIII. THE BAYESIAN NETWORK REPRESENTATION DECREASES THE NUMBER OF TERMS REQUIRED FOR COMPUTING THE ELD SCORE.

Dataset	#Terms Using BN	#Terms without Using BN
Strikers vs. Goalies	1,430	114,633,792
Midfielders vs. Goalies	1,376	43,670,016
Drama vs. Comedy	50,802	215,040,000

2) Detection Accuracy: We follow the evaluation design of [9] and make each baseline methods detect the same percentage of objects as outliers: Sort the outlier scores obtained by the three baseline methods in descending order, and take the top r percent as outliers. Then we use **precision**, a.k.a. **true positive rate** as the evaluation metric which is the percentage of correct ones in the set of outliers identified by the algorithm. As in [9], we set the percentages of outlier to be 1% and 5%. In the one-class design, precision measures how many members of the outlier class were correctly recognized. We also report some AUC measurements [2], which aggregate precision values at different percentage cutoffs.[1]

a) Likelihood-Based Methods: Table IX shows the $Precision$ values for each probabilistic ranking. Our ELD score achieves the top score on each dataset. On the synthetic data, ELD and $|LR|$ are the only scores with 100% precision at 1% and 5%. This confirms the value of using distances rather than differences. While it ought to be easy to distinguish the outliers, Table X shows that *ELD is the only score that achieves perfect detection*, that is AUC = 1.0.

[1]Our ELD score performs the best also with other metrics such as recall, to a similar degree; we omit the details due to space constraints.

TABLE IX. PRECISION OF OUTLIER SCORES IN DIFFERENT DATASETS.

Dataset	percentage	Model-based models					Aggregation-based models		
		ELD	$\|LR\|$	*LR*	*FD*	*LOG*	*LOF*	$OutRank$	*KNNOutlier*
High-Correlation	1%	**1.00**	**1.00**	0.73	0.47	0.91	0.11	0.53	0.48
	5%	**1.00**	**1.00**	0.85	0.65	0.95	0.22	0.50	0.65
Low-Correlation	1%	**1.00**	**1.00**	0.87	0.14	0.93	0.10	0.00	0.06
	5%	**1.00**	**1.00**	0.90	0.25	0.95	0.25	0.10	0.14
Single-Feature	1%	**1.00**	**1.00**	0.39	0.53	0.81	0.46	**1.00**	0.51
	5%	**1.00**	**1.00**	0.55	0.62	0.92	0.55	**1.00**	0.54
Striker-Goalie	5%	**0.57**	0.27	0.22	0.51	0.36	0.19	0.47	0.42
	15%	**0.63**	0.36	0.31	0.58	0.40	0.32	0.50	0.52
Midfielder-Striker	1%	**0.49**	0.42	0.25	0.41	0.46	0.29	0.44	0.16
	5%	**0.52**	0.48	0.39	0.44	0.50	0.38	0.48	0.35
Drama-Comedy	1%	**0.44**	0.38	0.39	0.15	0.22	0.29	0.07	0.014
	5%	**0.47**	0.45	0.44	0.40	0.28	0.36	0.17	0.20

TABLE X. *AUC* OF ELD VS. $|LR|$.

Score	High-Cor.	Low-Cor.	Single-F.	Striker	Midfielder	Drama
ELD	1.00	1.00	1.00	0.89	0.66	0.70
$\|LR\|$	0.95	0.95	0.89	0.61	0.64	0.65

b) Aggregation-Based Methods vs. ELD: Table IX shows the precision values for aggregation-based methods compared to ELD. *Our ELD score outperforms all aggregation-based methods on all datasets*, except for a tie with $OutRank$(ProClus) on the relatively easy problem of distinguishing strikers from goalies. The performances of aggregation-based methods are most like that of the probabilistic score FD, which does not consider the correlation among the features. This finding reflects the fact that aggregation tends to lose information about correlations. The aggregation-based methods achieve their highest performance on the Strikers vs. Goalies dataset. In this dataset action count features such as ShotsOnTarget, ShotEfficiency point to strikers and the feature SavesMade points to goalies. Therefore, outliers in this dataset are easy to find by considering features in isolation.

3) Case Studies: For a case study, we examine three top outliers as ranked by ELD, shown in Table XI. The aim of the case study is to provide a qualitative sense of the outliers indicated by the scores. Also, we illustrate how the BN representation leads to an interpretable ranking. Specifically, we employ a *feature-wise decomposition* of the score combined with a *drill down* analysis:

1) Find the node V_i that has the highest ELD_i divergence score for the outlier object.
2) Find the parent-child combination that contributes the most to the ELD_i score for that node.
3) Decompose the ELD score for the parent-child combination into feature and mutual information component.

We present strong associations—indicated by the ELD's mutual information component—in the intuitive format of association rules.

a) Strikers vs. Goalies: In real-world data, a rare object may be a *within-class outlier*, i.e., highly anomalous even within its class. In an unsupervised setting without class labels, we do not expect an outlier score to distinguish such an in-class outlier from outliers outside the class. An example is the striker Edin Dzeko. He is a highly anomalous striker who obtains the top ELD divergence score among both strikers and goalies. His ELD score is highest for the Dribble Efficiency feature. The highest ELD score for that feature occurs when Dribble Efficiency is low, and its parents have the following values: Shot Efficiency high, Tackle Efficiency medium. Looking at the single feature divergence, we see that Edin Dzeko is indeed an outlier in the Dribble Efficiency subspace: His dribble efficiency is low in 16% of his matches, whereas a randomly selected striker has low dribble efficiency in 50% of their matches. Thus, Edin Dzeko is an unusually good dribbler. Looking at the mutual information component of ELD, i.e., the parent-child correlations, for Edin Dzeko the confidence of the rule

$$ShotEff = high, TackleEff = medium \rightarrow DribbleEff = low$$

is 50%, whereas in the general striker class it is 38%.

b) Midfielders vs. Strikers: For the single feature score, Robin van Persie is recognized as a clear striker because of the $ShotsOnTarget$ feature. It makes sense that strikers shoot on target more often than midfielders. Robin van Persie achieves a high number of shots on targets in 34% of his matches, compared to 3% for a random midfielder. The mutual information component shows that he also exhibits unusual correlations. For example, the confidence of the rule

$$ShotEff = high, TimePlayed = high \rightarrow ShotsOnTarget = high$$

is 70% for van Persie, whereas for strikers overall it is 52%.

The most anomalous midfielder is Scott Sinclair. His most unusual feature is $DribbleEfficiency$: For feature divergence, he achieves a high dribble efficiency 50% of the time, compared to a random midfielder with 30%.

c) Drama vs. Comedy: The top outlier rank is assigned to the within-class outlier $BraveHeart$. Its most unusual feature is $ActorQuality$: In a random drama movie, 42% of actors have the highest quality level 4, whereas for $BraveHeart$ 93% of actors achieve the highest quality level.

The ELD score identifies the comedies $BluesBrothers$ and $AustinPowers$ as the top out-of-class outliers. In a random drama movie, 49% of actors have casting position 3, whereas for $AustinPowers$ 78% of actors have this casting position, and for $BluesBrothers$ 88% of actors do.

VIII. CONCLUSION

We presented a new approach for applying Bayes nets to object-relational outlier detection, a challenging and practically important topic for machine learning. The key idea is to learn one set of parameter values that represent class-level associations, another set to represent object-level associations, and compare how well each parametrization fits the relational data that characterize the target object. The classic metric for comparing two parametrized models is their log-likelihood ratio; we refined this concept to define a new relational log-likelihood distance metric via two transformations: (1) a mutual information decomposition, and (2) replacing log-likelihood differences by log-likelihood distances. This metric combines a single feature component, where features are treated as independent, with a correlation component that measures the deviation in the features' mutual information.

In experiments on three synthetic and three real-world outlier sets, the log-likelihood distance achieved the best detection accuracy. The alternative of converting the structured data to a flat data matrix via aggregation had a negative impact. Case studies showed that the log-distance score leads to easily interpreted rankings.

TABLE XI. CASE STUDY FOR THE TOP OUTLIERS RETURNED BY THE LOG-LIKELIHOOD DISTANCE SCORE *ELD*

Strikers (Normal) vs. Goalies (Outlier)							
PlayerName	Position	ELD Rank	ELD Max Node	ELD Node Score	FD Max feature Value	Object Probability	Class Probability
Edin Dzeko	Striker	1	DribbleEfficiency	83.84	DE=low	0.16	0.5
Paul Robinson	Goalie	2	SavesMade	49.4	SM=Medium	0.3	0.04
Michel Vorm	Goalie	3	SavesMade	85.9	SM=Medium	0.37	0.04
Midfielders (Normal) vs. Strikers (Outlier)							
PlayerName	Position	ELD Rank	ELD Max Node	ELD Node Score	FD Max feature Value	Object Probability	Class Probability
Robin Van Persie	Striker	1	ShotsOnTarget	153.18	ST=high	0.34	0.03
Wayne Rooney	Striker	2	ShotsOnTarget	113.14	ST=high	0.26	0.03
Scott Sinclair	Midfielder	6	DribbleEfficiency	71.9	DE=high	0.5	0.3
Drama (Normal) vs. Comedy (Outlier)							
MovieTitle	Genre	ELD Rank	ELD Max Node	ELD Node Score	FD Max feature Value	Object Probability	Class Probability
Brave Heart	Drama	1	ActorQuality	89995.4	a_quality=4	0.93	0.42
Austin Powers	Comedy	2	Cast_Position	61021.28	Cast_Num=3	0.78	0.49
Blue Brothers	Comedy	3	Cast_Position	24432.21	Cast_num=3	0.88	0.49

There are several avenues for future work. (i) A limitation of our current approach is that it ranks potential outliers, but does not set a threshold for a binary identification of outlier vs. non-outlier. (ii) Our divergence uses expected L1-distance for interpretability, but other distance scores like L2 could be investigated as well. (iii) Extending the expected L1-distance for continuous features is a useful addition.

In sum, outlier metrics based on model likelihoods are a new type of structured outlier score for object-relational data. Our evaluation indicates that this model-based score provides informative, interpretable, and accurate rankings of objects as potential outliers.

ACKNOWLEDGEMENT

This work was supported by a Discovery Grant from the National Sciences and Engineering Council of Canada.

REFERENCES

[1] E. Achtert, H. Kriegel, E. Schubert, and A. Zimek. Interactive data mining with 3d-parallel coordinate trees. In *Proceedings of the 2013 ACM SIGMOD*, New York, NY, USA, 2013.

[2] C. Aggarwal. *Outlier Analysis*. Springer New York, 2013.

[3] F. Angiulli, G. Greco, and L. Palopoli. Outlier detection by logic programming. *ACM Transactions on Computer Logic*, 2004.

[4] M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander. Lof: Identifying density-based local outliers. In *Proceedings of ACM SIGMOD*, 2000.

[5] A. Cansado and A. Soto. Unsupervised anomaly detection in large databases using Bayes Nets. *Appllied Artificial Intelligence*, 2008.

[6] L. de Campos. A scoring function for learning Bayes nets based on mutual information and conditional independence tests. *Journal of Machine learning Research*, 2006.

[7] P. Domingos and D. Lowd. *Markov Logic: An Interface Layer for Artificial Intelligence*. Morgan and Claypool Publishers, 2009.

[8] P. A. Flach. Knowledge representation for inductive learning. In *Symbolic and Quantitative Approaches to Reasoning and Uncertainty*, pages 160–167. Springer, 1999.

[9] J. Gao, F. Liang, W. Fan, Y. Wang, and J. Han. On community outliers and their detection in information network. In *Proceedings of ACM SIGKDD*, 2010.

[10] L. Getoor and B. Taskar. *Introduction to statistical relational learning*. MIT Press, 2007.

[11] Internet Movie Database. Internet movie database. [Online]. Available: URL = http://www.imdb.com/.

[12] H. Khosravi, T. Man, J. Hu, E. Gao, and O. Schulte. Learn and join algorithm code. [Online]. Available: URL = http://www.cs.sfu.ca/~oschulte/jbn/.

[13] A. Kimmig, L. Mihalkova, and L. Getoor. Lifted graphical models: a survey. *Computing Research Repository*, 2014.

[14] J. L. Koh, M. L. Lee, W. Hsu, and W. T. Ang. Correlation-based attribute outlier detection in XML. In *Proceedings of ICDE. IEEE 24th*, 2008.

[15] D. Koller and A. Pfeffer. Object-oriented Bayes nets. In *Proceedings of UAI*, 1997.

[16] J. Maervoet, C. Vens, G. Vanden Berghe, H. Blockeel, and P. De Causmaecker. Outlier detection in relational data: A case study. *Expert System Applications*, 2012.

[17] MCFC Analytics. The premier league dataset. [Online]. Available: URL = http://www.mcfc.co.uk/Home/MCFCAnalytics.

[18] E. Muller, I. Assent, P. Iglesias, Y. Mulle, and K. Bohm. Outlier ranking via subspace analysis in multiple views of the data. In *Proceedings of ICDM*, 2012.

[19] P. K. Novak, G. I. Webb, and S. Wrobel. Supervised descriptive rule discovery: A unifying survey of contrast set, emerging pattern and subgroup mining. *Journal of Machine Learning Research*, 2009.

[20] J. Pearl. *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann, 1988.

[21] V. Peralta. Extraction and Integration of MovieLens and IMDb. Technical report, APDM project, 2007.

[22] D. Poole. First-order probabilistic inference. In *Proceedings of IJCAI*, 2003.

[23] S. Ramaswamy, R. Rastogi, and K. Shim. Efficient algorithms for mining outliers from large data sets. *SIGMOD*, 2000.

[24] F. Riahi and O. Schulte. Codes and Datasets. [Online]. Available:. ftp://ftp.fas.sfu.ca/pub/cs/oschulte/CodesAndDatasets/, 2015.

[25] S. Sarawagi, R. Agrawal, and N. Megiddo. Discovery-driven exploration of OLAP data cubes. In *Proceedings of International Conference on Extending Database Technology*. Springer-Verlag, 1998.

[26] O. Schulte. A tractable pseudo-likelihood function for Bayes nets applied to relational data. In *Proceedings of SIAM SDM*, 2011.

[27] O. Schulte and H. Khosravi. Learning graphical models for relational data via lattice search. *Journal of Machine Learning*, 2012.

[28] G. Tang, J. Bailey, J. Pei, and G. Dong. Mining multidimensional contextual outliers from categorical relational data. In *Proceedings of SSDBM*, 2013.

[29] S. Tuffery. *Data Mining and Statistics for Decision Making*. Wiley Series in Computational Statistics, 2011.