MODEL-BASED OUTLIER DETECTION FOR OBJECT-RELATIONAL DATA

by

Fatemeh Riahi
Bachelor's degree, Sharif University of Technology, 2009
Master's degree, Dalhousie University, 2012

a Thesis submitted in partial fulfillment
of the requirements for the degree of

Doctor of Philosophy

in the
School of Computing Science
Faculty of Applied Sciences

SIMON FRASER UNIVERSITY
Spring 2016

APPROVAL

Name: Fatemeh Riahi

Degree: Doctor of Philosophy

Title of Thesis: Model-based Outlier Detection for Object-Relational Data

Examining Committee: Dr. TBD, Associate Professor, Computer Science
Chair

Dr. Oliver Schulte, Professor, Computing Science
Senior Supervisor

Dr. Jian Pei, Professor, Computing Science
Supervisor

Dr. Christos Faloutsos, Professor, Computer Science
Carnegie Mellon University
External Examiner

Dr. David Mitchell
Associate Professor, Computing Science
Internal Examiner

Date Approved:

Abstract

Outliers are anomalous and interesting objects that are notably different from the rest of the data. Outliers often reveal useful information about the unusual behaviour of the systems and entities. The identification of such abnormal behaviour provides valuable application-specific insights. Outlier Detection task has sometimes been considered as removing noise from the data. However, it is usually the significantly interesting deviations that are of most interest. There are many applications, such as detecting criminal activities and identifying aircraft engine rotation defects, where finding outliers is crucial and does not seem to be a straightforward task.

Different outlier detection techniques work with various data formats. For example, the data may be purely multidimensional or sequential with temporal ordering or may be defined in the form of a network with arbitrary relationships among data points. In addition, the attributes in the data may be numerical, categorical or maybe mixed. The outlier detection process needs to be sensitive to the nature of the underlying data. Most of the previous work on outlier detection was designed for propositional data. This dissertation focuses on developing outlier detection methods for structured data, more specifically object-relational data. Object-relational data can be viewed as a heterogeneous network with different classes of objects and links.

We develop two new approaches to unsupervised outlier detection; both approaches leverage the statistical information obtained from a statistical-relational model. The first method develops a propositionalization approach to summarize information from object-relational data in a single data table. We use Markove Logic Network (MLN) structure learning to construct the features for the single data table and to mitigate the loss of information that usually happens when features are generated by the manual aggregation. By using propositionalization as a pipeline, we can apply many previous outlier detection methods that were designed for single-table data.

Our second outlier detection method ranks the objects as potential outliers in an object-oriented data model. Our key idea is to compare the feature distribution of a potential outlier object with the feature distribution of the objects class. We introduce a novel distribution divergence concept that is suitable for outlier detection. Our methods are validated on synthetic datasets and on real-world data sets about soccer matches and movies.

To Ali, and to my parents

"I wish I had an answer to that question because I'm tired of answering that question".

– Yogi Berra

Acknowledgments

First I want to thank my Ph.D. senior supervisor, Dr. Oliver Schulte. I appreciate all his contributions of time, ideas, and funding that made my Ph.D. possible. I am very thankful for his excellent guidance and knowledge and patience throughout my Ph.D. I would also like to thank my supervisor, Dr. Jian Pei for his guidance and support. I am grateful to Dr. Christos Faloutsos and Dr. David Mitchell for participating in my final defense as external and internal examiner.

I would like to thank my family for all their love and support. Especially my father and mother for all the sacrifices that they have made on my behalf. And most of all I would like to thank my loving, supportive and patient husband Ali. He has been my motivation to move my career forward.

I gratefully acknowledge my collaborators and other students that I had a chance to work with during my Ph.D., Nicole Li, Zhensong Qian, Sajjad Gholami, Yan Sun and Mahmoud Khademi.

I would like to thank my friends that made my student life more enjoyable: Anahita, Elaheh, Mojtaba, Monir, Ali M., Shabnam, Atieh, Ali R. and Faezeh.

Lastly, I would like to thank the authors whose books inspired, entertained and educated me during my Ph.D. journey. I would like to thank Margaret Mitchell for teaching me that “after all, tomorrow is another day”; George R.R. Martin for teaching me that “fear cuts deeper than sword”; and Ernest Hemingway for teaching me that “there’s no one thing that’s true. It’s all true’.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Detection of outliers is an essential part of knowledge discovery in databases (KDD) and has been used to identify anomalous entities from data for decades. System faults or changes, human or mechanical error, or any sort of deviations in population caused by unknown reasons, may result in outliers in the data [35].
Many techniques have been employed in Data Mining, Machine Learning, and Statistics to find outliers in different domains such as Network Performance, Motion Segmentation, Medical Condition Monitoring and Pharmaceutical Research. Demand for efficient analysis methods to detect outliers has been increased due to the large amount of data collected in databases. In this dissertation, we developed two generative model-based methods for the case of object-relational data.

1.1 Outlier Definition

Many definitions have been proposed for an outlier and there is no generally accepted definition. Grubbs *et al.* define outliers as an observation that appears to deviate considerably from other members of the sample in which it occurs [31]. Hawkings *et al.* describe outliers as an observation which deviates so much from the other observations as to arouse suspicions that it was generated by a different mechanism [33]. These are only a few examples of definitions proposed in the previous outlier detection work. Basically, the outlier definition is context related and depends on the type of the application that employs the outlier detection. For example, Akoglu *et. al* define outliers in graphs as rare graph objects that differ significantly from the majority of the reference objects in the graph [5]. This definition is the basis of our proposed outlier detection methods.

The output of an outlier detection algorithm can be one of these two types [3]:

1. Outlier score assigned to each individual that shows the degree of "outlierness" of each data point.

2. A binary label indicating whether a data point is an outlier or not. By imposing thresholds on outlier scores, based on their statistical distribution, the outlier scores can be converted into binary labels.

1.1.1 Outlier Detection Challenges

Outlier Detection applications must overcome many challenges such as:

- Modeling normality is hard: there is not often a clear line separating data normality and abnormality since it is hard to define all possible normal behaviours. Therefore, many outlier techniques measure the degree of outlier-ness of each data point instead of firmly labelling it as either outlier or normal.

- Designing an outlier detection method depends on the type of application: one of the earliest steps in designing a model to identify outliers is choosing similarity or distance measure. However, different applications require different sensibility in terms of similarity or difference. For example, in medical data analysis, a tiny deviation may be a sign of an outlier; while marketing analysis, for example, allows larger fluctuations between its normal data points.

- Separating noise and error from the outliers: outliers and noise are different, however, they are similar in the sense that they both deviate from the normal behaviour. This similarity can make the distinction between normal and outlier and noise objects even harder.

- Understandability and interpretability: in some applications, detected outliers should be justified and the features that moved the data points from normal to outlier should be identified. In this case, outlier detection method must provide some explanations.

- Class imbalance: the imbalanced nature of outlier detection makes accurate detection hard to achieve.

1.1.2 Approach

Many outlier detection methods have been developed for data that are represented in a propositional format (i.e., as a flat feature vector) or unstructured data.

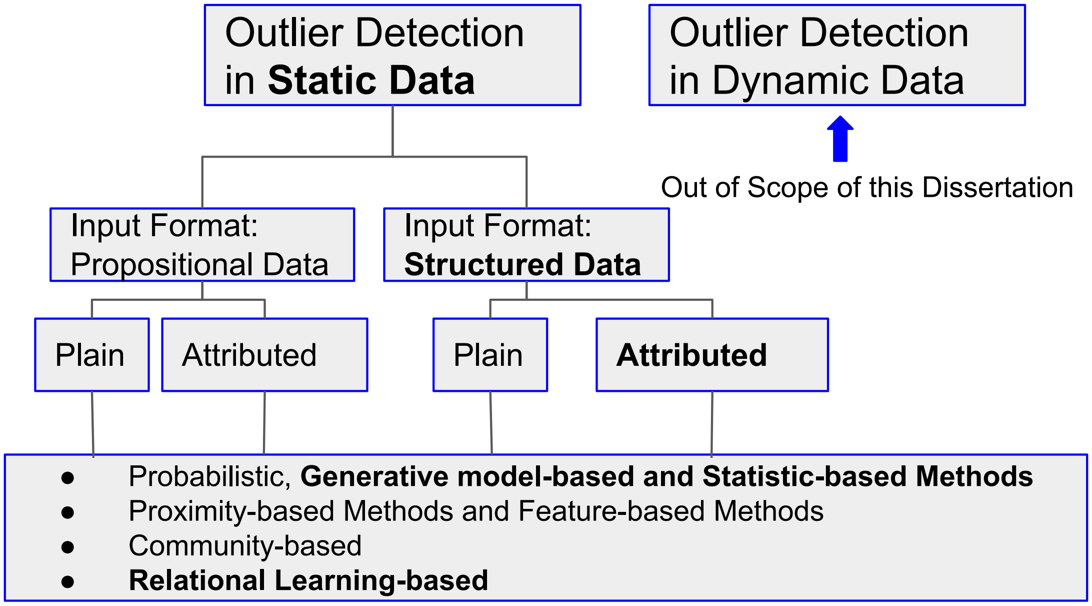

Outlier Detection in **Static Data**
Outlier Detection in Dynamic Data
Out of Scope of this Dissertation
Input Format: Propositional Data
Input Format: **Structured Data**
Plain
Attributed
Plain
Attributed
Probabilistic, **Generative model-based and Statistic-based Methods**
Proximity-based Methods and Feature-based Methods
Community-based
Relational Learning-based

Figure 1.1: Categorization of outlier detection methods. The bold font indicates where our methods stand in this categorization.

In a propositional data table, a row represents a data point, a column represents an attribute of a data point, and a table entry represents an attribute value for a data point. This dissertation extends unsupervised statistical outlier detection to the case of structured data, more specifically object-relational data. Object-relational data represents a complex heterogeneous network [27], which comprises objects of different types, links among these objects, also of different types, and attributes of these links. Given the prevalence of object relational data in organizations, outlier detection for such data is an important problem in practice. However, applying standard outlier detection methods designed for single data tables on object-relational data runs into impedance mismatch, since object-relational data are represented in multiple interrelated tables.

In order to detect outliers in object-relational data, we have developed two generative model-based approaches. The advantages of employing model-based approach for outlier detection task are as follows: 1) we can apply many statistical relational learning methods for building the model. 2) We can leverage statistical concepts such as divergence metrics to measure outlierness of the data points. 3) We can employ outlier detection methods designed for the propositional data.

Based on Figure 1.1 categorization, our proposed outlier detection methods fall into the category of unsupervised, relational learning-based, attributed models which can be applied to both static and dynamic datasets.

1. In chapter 4 a model-based method has been proposed to generate conjunctive features for outlier detection task. This method leverages outlier detection tools that are designed for the single table via a pipeline data preprocessing approach by converting the object-relational data into a single attribute-value table, then applying the data analysis tools. Since the attribute value representation corresponds to propositional logic, the conversion is called propositionalization. Propositionalization has been used to detect outliers in the literature. For example, a technique called ODDBALL introduced by Akoglue *et al.* extracts graph-centric feature to detect anomalies in graph structure [5]. However, to the best of our knowledge, this work is the first model-based propositionalization approach for outlier detection. In chapter 4 we show that conjunctive features for outlier detection can be learned from data by using statistical-relational methods. Specifically, we apply Markov Logic Network structure learning. Compared to baseline propositionalization methods, Markov Logic propositionalization produces the most compact data tables, whose attributes capture the most complex relational correlations. We apply three representative outlier detection methods (*LOF*, *KNNOutlier*, *OutRank*) to the data tables constructed by propositionalization. This research was published in the proceedings of the Florida Artificial Intelligence Association (FLAIRS2016) conference [24].

2. Chapter 5 introduces a model-based method to define outlierness metric. We first 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 likelihood-based model achieved the best detection accuracy on all datasets except one.

Model-based methods have been previously used for outlier detection tasks. Loglikelihood has been used to identify outliers [13]. Rule mining and sub-group mining are another examples of model-based outlier detection [4, 45]. However, none of these methods are based on a joint distribution distance metric.

This work was published in the proceedings of IEEE Symposium series on Computational Intelligence (SSCI 2015) conference and won the best student paper award [75].

In chapter 6 we compare the log-likelihood distance to metrics of success for a given domain. Success rankings are one of the most interesting features to users. Our reasoning is that high success is an independent metric that indicates an unusual individual. So a correlation between log-likelihood distance and success is an independent validation of the log-likelihood distance, and also shows that it points to meaningful and interesting outliers. A version of this work was submitted to the Journal of Data mining and Knowledge discovery.

Figure 1.2 provides a tree picture of where our methods are situated with respect to other outlier detection methods and other data models.

1.1.3 Contributions

The main contributions of this dissertation are the followings:

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.

3. A novel task for relational learning: MLN-propositionalization for outlier detection.

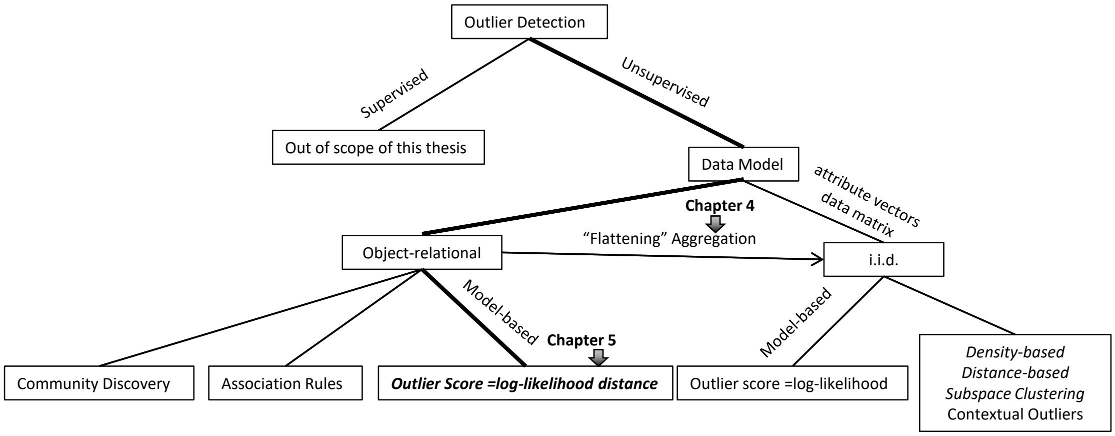


Figure 1.2: A tree structure for research on outlier detection for structured data. A path specifies an outlier detection problem, the leaves list major approaches to the problem.

This facilitates leveraging standard single-table outlier analysis methods for object-relational data. This task is also a novel application of Markov Logic Network structure learning.

4. A novel task for relational learning: propositionalization for outlier detection. This facilitates leveraging standard single-table outlier analysis methods for object-relational data. We use Markove Logic Network (MLN) structure learning for propositionalization task. This is a novel application of MLN structure learning.

1.1.4 Limitations and Directions for Future Work

The main limitations of the work presented in this dissertation are the following:

1. **Limitation of Approach:**

   (a) Our proposed methods rank potential outliers, but do not set a threshold for a binary identification of outlier vs. non-outlier.

   (b) Our current Bayesian Network Learning method can only be applied to discrete data. Prior to learning the model, we take an extra step in data preprocessing and convert the continuous data into discrete which naturally causes some information loss.

(c) Our generative model-based methods learn a generic Bayesian network structure for the entire population, ignoring the subgroups that inherently exist in the real datasets, as a result, the detected outliers are global outliers. However, there are more complex outliers that locally deviate from their subgroups and can be detected only by subgroup comparison. One direction for future work is to first detect subgroups in the population and then perform the outlier detection task.

2. **Limitation of Data Analysis:**

   In this dissertation,to simplify the outlier detection task, we used only part of the full information available in our rich datasets. The model-based outlier detection can be extended in future work to take advantage of the full information.

   (a) In the Premier League dataset, players are naturally related to one another and modelling the interaction between players can be another way to detect anomalous players. The graph-based features, such as detecting near-clique nodes and star nodes, proved to be efficient in discovering patterns for anomaly detection task as shown in ODDBALL [5].

   (b) In this dissertation, we did not use the temporal information available in the data. In the learning process we do not give a higher weight (importance) to the more recent action (performance) of an individual. This point is especially important when applying the methods to dynamic data or the data that are collected from long periods of time.

Chapter 2

Literature Review

This chapter provides a literature review of the state-of-the-art methods in the field of outlier detection. Outlier detection is a very well-explored area and there are many surveys to overview the state-of-the-art methods. Each survey categorized these methods differently. The categorization can be based on datatype (e.g. graph data) or type of methods that have been used to detect outlier (e.g. structured-based methods). In this chapter we group outlier methods based on the format of the input data, whether it is presented in a single data table or it has a higher level of organization such as data presented in a relational database, or XML format or OLAP. Since the focus of our work is on structured data, we explore more methods designed for that format .

We conclude this chapter with Section 2.5 which addresses the limitations of current outlier detection methods. Figure 2.1 shows the organization of this chapter.

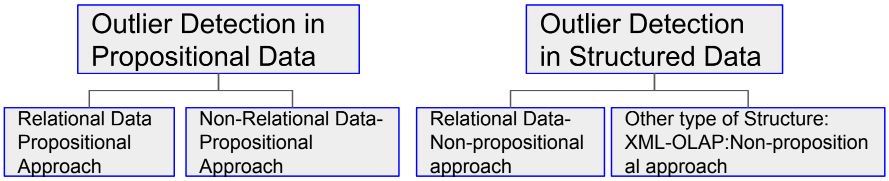

Outlier Detection in Propositional Data
Relational Data Propositional Approach
Non-Relational Data-Propositional Approach
Outlier Detection in Structured Data
Relational Data-Non-propositional approach
Other type of Structure: XML-OLAP:Non-proposition al approach

Figure 2.1: Related work categorization

2.1 Outlier Detection Methods for Propositional Data

In this section, we explore outlier detection methods that take *propositional* data. One interpretation of propositional data is that the attributes describe characteristics of one object-class. For example, $\mathit{shotEfficiency}(\mathit{Player})$ shows shot efficiency of a player in general, while in a more structured data the attributes are more complex; for example in the Object-relational data model, the attributes are shown in this format: $\mathit{shotEfficiency}(\mathit{Player}, \mathit{Match})$ which represents shot efficiency of a player in a match and involves two object-classes. Throughout this dissertation, we call the methods designed for single data table propositional methods. In this section, we further categorize these methods into supervised and unsupervised based on whether the sample of data has been provided with labels and domain expert information to build an outlier detection model.

2.1.1 Supervised Methods for Propositional Data

These methods model both normality and abnormality and requires pre-labelled data. Normal points may belong to a single class or be divided among different classes. Supervised outlier detection is a special case of the classification where the labels are extremely unbalanced in terms of occurrence [14]. Normal data points are easily available while outlier examples are very sparse and it is the rarity that makes these data points outliers. In this sense, outlier detection can also be viewed as *rareclass* detection problem. The imbalanced nature of outlier detection makes the accurate classifications quite hard to achieve and it might result in over-training [3].

When the purpose of classification is outlier detection, cost-sensitive variations of machine learning algorithms can be used in order to make the classification of anomalies more accurate [3]. Class imbalance is one of the common problems in supervised outlier detection. The standard evaluation techniques in classifications cannot simply be applied to outlier detection. For example, in the case of breast cancer, where 99% of scans are identified to be normal and only 1% abnormal, the trivial classification algorithm which returns all the instances in the test cases as normal would have a high accuracy of 99%. However, it is not useful in the context of detecting breast cancer. Cost-sensitive learning is one way that has been used to handle the class imbalance problem in outlier detection. The objective function of the classification in this type of learning has been designed in a way that it weights the errors in classification differently for different classes. In this case, classifiers are tuned

so the errors in classification of outliers are more penalized compared to the misclassified normal classes [23]. In other words, methods are forced to predict the outlier class far better than the normal class. This trade-off is characterized either by the precision-recall curve or a receiver operating characteristics (ROC) curve.
Active re-sampling is another way to tackle class imbalance problem. The relative proportion of the rare classes is magnified through re-sampling. This approach can be considered as an indirect form of cost-sensitive learning.
Classification outlier methods can be grouped into two categories:

- Multi-class: Training data in this group contains the instances from multiple normal classes [83]. First, a classifier is learned to distinguish between instances from different normal classes. If a test instance is not classified as normal by any of the classifiers, then it is labeled as an outlier.
- One-class: All training instances belong to a single class label. If any test case does not fall into the normal boundary, it is identified as an outlier. Examples of well-known algorithms are:
    - One-class SVMs [79]
    - One-class Kernel Fisher Discriminant [77]

In the following subsections, we provide examples of supervised methods in different areas.

Neural Network

Neural networks are applied to outlier detection in one-class learning as well as multi-class scenarios. At the first step, a neural network is trained on normal training data in order to learn the normal behavior of data points. Then, test instances are presented to the neural network. If the test is accepted, the data point is normal, otherwise, it is an outlier. Different neural network techniques have been proposed to tackle outlier detection problem. Ghosh *et al.* [30] apply multi-layered perceptrons and focus on detecting attacks on computer systems. They perform intrusion detection on software programs instead of what most intrusion detection systems do by analyzing network traffic and host system logs. They build a profile of software behaviour to distinguish between a normal software behaviour and a malicious one.

Support Vector Machine

Support Vector Machines have been applied to outlier detection mostly in a one-class setting. These techniques first learn a region that includes the training data points. If a test instance falls into the learned region, it is considered as normal, otherwise, it is an outlier [17, 72].

2.1.2 Unsupervised Methods for Propositional Data

In the datasets where data points are not labeled, unsupervised approaches make some assumptions about the behaviour of outliers. Based on the assumptions made, these methods can be categorized.

Probabilistic and Statistical Models

In the probabilistic and statistical model, the data is assumed to be derived from a closed form of probability distribution and the goal is to learn the parameters of this model. Therefore, the main challenge is to choose the data distribution. The parameter of the distribution can be learned by using different algorithms, such as Expectation Maximization, the key output of this method is that the membership probability of data points to the distribution and the ones that have a very low fit will be considered as outliers. Extreme Value Test can also determine the outliers in this stage [3].

The most popular statistical modeling is detecting extreme values that determine data values at the tails of a uni-variate distribution. However, these methods were not designed to focus on issues such as data representation or computational efficiency. Also, most of the multidimensional outliers cannot be determined through extreme data values and are usually defined by the relative positions of data points with respect to each other. While extreme value analysis may be applicable to only a specific type of data, they have many applications beyond the univariate case since the final step in most outlier detection methods is to identify extreme values to assigned scores. Gao *et al.* have worked on the problem of identifying extreme values from the outlier scores [28].

Laurikkala *et al.* describe one of the simplest statistical outlier detection methods where an information box plot has been used to identify outliers in both uni-variate and multivariate datasets [50]. For multivariate datasets the authors claimed that there is not any clear ordering, but they suggested using reduced sub-ordering based on the generalized distance metric using the *Mahalanobis* distance. Mahalanobis distance is similar to the Euclidean distance, except that it normalizes the data on the basis of the inter-attribute correlation and scales the distance values by local cluster variances along the directions of correlation. Consider a dataset containing k clusters. Assume that the rth cluster in d-dimensional space has a corresponding d-dimensional mean vector $\bar{\mu_r}$ and a $d \times d$ co-variance matrix Σ_r. The (i, j)th entry of this matrix is the co-variance between dimension i and j in that cluster. Then, the Mahalanobis distance $MB(\bar{X}, \bar{\mu_r})$ between a data point $\bar{X}$ and the cluster centroid $\bar{\mu_r}$ is as follows:

$$MB(\bar{X}, \bar{\mu_r}) = (\bar{X} - \bar{\mu_r}) \cdot \Sigma_r^{-1} \cdot (\bar{X} - \bar{\mu_r})^T \tag{2.1}$$

Intuitively, this metric scales the square distances by the cluster variances along the different directions of correlation.

Bayesian Network In order to detect disease outbreak early, Wong *et al.* [93], compared the distribution of data against a baseline distribution. A different environmental attribute, such as trends caused by the day of a week and by seasonal variations in temperature and weather, makes defining such a baseline hard, if not impossible. By using Bayesian network that takes the joint distribution of the data and conditioning on attributes that are responsible for the trends, they were able to define such a baseline.

Babbar *et al.* [11] used joint probability distribution and knowledge of the domain driven by Bayes net to identify low probable data points with intrinsic anomalous patterns and they treat them as potential outliers.

Cansado *et al.* [13] followed a probabilistic approach and modeled the joint probability density function of the attributes of data points in the database and ranked the records according to their oddness. In order to estimate the joint probability density function and handle its complexity, they used a Bayesian Network.

Proximity-Based Models

Proximity based approach are based on the calculation of the distance between all records and make no assumptions about the data distribution. The most common approaches for defining proximity for outlier detection are:

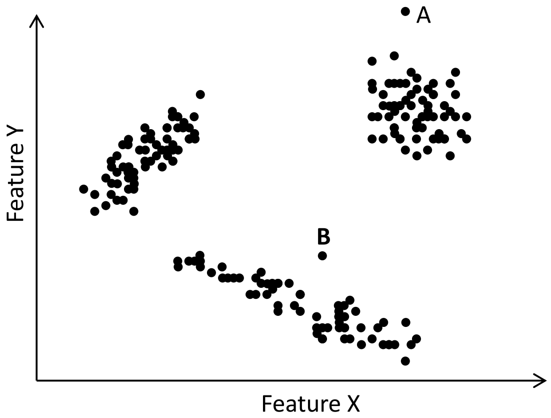


Figure 2.2: The Mahalanobis distance function can detect better outliers: When using Euclidean distance, the distance between data point B and the closest cluster centroid will be smaller than A and its cluster centroid; while data point 'B' is more obviously an outlier than data point 'A', because it does not follow the direction of the correlation of its cluster.

1: Cluster-Based methods These methods score outliers based on whether they belong to any predefined cluster and also the distance of the data points from clusters. Therefore, the performance of these methods has a high correlation with the efficiency of the clustering algorithm that they use [3]. However, outliers that are chosen based on their complementary membership to a cluster are often weak outliers or noise and not necessarily interesting to analyze. For example, a data point that is located at the margin of a large cluster is very different from a point that is completely away from all other clusters. In addition, all data points in a small cluster may sometime actually be outliers [3]. Therefore, a measure is needed to quantify the degree of abnormality of data points. Many cluster-based methods try to assign a score to the outliers, most of the time by a simple definition as the distance of data points to cluster centroids. As clusters may have different shapes, Mahalanobis Distance is the best way to compute the distance that scales the square distances by the cluster variances along the different directions of correlation and it is used for effective statistical normalization. In other words, large distances in clusters with high variance may not be statistically significant within that data locality. It is possible that a data point that is closer to one of the clusters has a higher Mahalanobis distance than a data point which is away

on the basis of Euclidean distance. In Figure 2.2 data point ‘B’ is more obviously an outlier than data point ‘A’.
Mahalanobis distance can be used as distance measure in many clustering algorithms, such as k-means [92, 15].

One advantage of cluster-based outlier detection methods is that they are based on global analysis of the data and small groups that do not fit within the major patterns can be easily detected using cluster-based methods.

Muller *et al.* propose a novel outlier scoring concept based on subspace clustering [60]. Their hypothesis is that regular objects show clustered behaviour in multiple subspaces even if the subspaces are very dissimilar to each other. On the other hand, outliers are clustered in some subspaces but deviate from these clusters if one considers other subspaces. Figure 2.3 shows object o_2 is clustered in two views, but not in the social view. Although there is a very similar clustering structure of the black objects in the “Sports View”, we see that this object deviates from its common grouping. Their outlier score, *outrank*, takes the similarity of subspaces into account and computes the outlierness degree based on the information available from subspace analysis. They rely on the general assumption that outliers are objects that do not agree with other data in at least a few of the attributes:

1. Outliers may be regular in some subspaces
2. They deviate in at least some subspaces.

They provide an abstract definition of a scoring function, given a subspace clustering as follows: let $SCR = (C_1, S_1), ..., (C_k, S_k)$ be a subspace clustering, a set of clusters C_i in their associated subspaces S_i. A scoring function on SCR is then defined as:

$$score(o) = \sum_{\{C,S) \in SCR | o \in C\}} evidence(o, (C, S), SCR) \tag{2.2}$$

where *evidence* computes a value of regularity for o being clustered in subspace cluster (C, S) given the entire subspace clustering result SCR. In their paper, they introduce three instantiations for the evidence function in equation 2.2.
Duan *et al.* tackle the problem of mining contrast subspaces [49]. Given a multi-dimensional data set $D = \{D_1, ..., D_d\}$ of two classes, and a query object q and a target class, their goal is to find the subspace $S = \{D_{i_1}, ..., D_{i_t}\} \subseteq D$ where the query

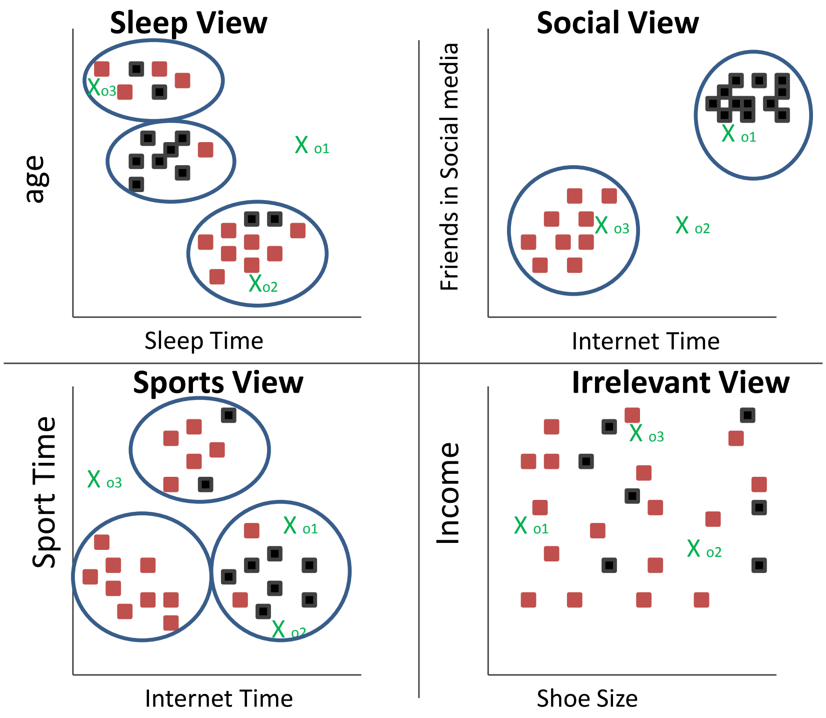


Figure 2.3: Outliers with respect to subspace views.

object most likely belongs to; given a set of objects, O, they assume that a latent distribution Z generates the objects in O. For a query object q, denote by $L_D(q|Z)$ the likelihood of q being generated by Z in full space D is computed. They assume that the objects in O belong to two classes, C_+ and C_- exclusively. Given a query object q, they aim to find the likelihood of q being a member of C_+ and not a member of C_-.

2: Distance-based methods In order to define proximity, distance-based methods use the distance of a data point and other data points in the dataset (or k-nearest neighbour of each point) and the most isolated data points are considered as outliers. However, they suffer from computational growth. The complexity of computation is a function of the dimensionality of the data (m) and the number of records(n). Therefore, methods such as k-nearest-neighbours with $O(n^2m)$ runtime are not feasible for high-dimensionality datasets.

However, a lot of approaches have been proposed in order to optimize k-nearest-neighbours and to produce a ranked list of potential outliers in a less complex way. [71] includes techniques for speeding k-nearest-neighbours algorithm, such as partitioning the data into cells and only considering a cell and its directly adjacent neighbours. If that cell contains more than k points then the cell has laid in a dense area and it is not likely to contain any outlier. They have improved the running speed of k-nearest-neighbours by using an efficient indexing structure with linear running time.

3: Density-based methods Local density is defined as the number of other points within a specified region of a data point. The difference between clustering and density-based methods is that clustering methods partition data points, while density based methods partition data space [3]. Figure 2.4 shows the cases that cannot be discovered by distance-based outlier techniques unless a small threshold is used. However, smaller distance threshold may result in incorrectly identifying many data points as outliers in the sparser clusters. It means that ranking returned by a distance-based method might be incorrect if there is significant heterogeneity in the local distribution of data. The most popular density based outlier methods are as follows:

- *LOF*: The Local Outlier Factor was originally presented in [12] as a measure to

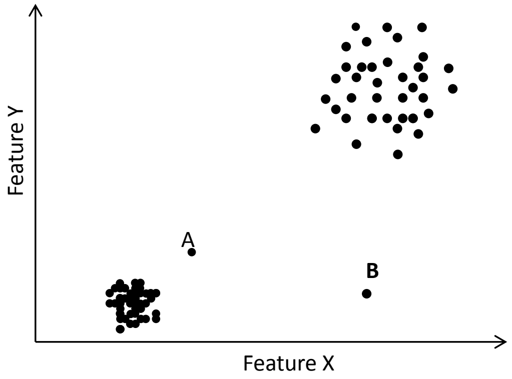


Figure 2.4: Impact of local density on outliers. If the threshold of a chosen distance-based methods is larger than the distance of A and the cluster centroid then the data point A will not be chosen as an outlier. If it is smaller then most of the points in the sparse cluster will be identified as outliers.

quantify the outlier-ness of the data points relative to regions of different densities. Therefore, the score is defined based on local density instead of the nearest neighbour distance. In simple words, *LOF* compares the density of area around an object to the densities of the areas of the surrounding objects. However, *LOF* defines density as the inverse of the average of the smoothed reach-ability distances in a neighbourhood; this definition is not the precise definition of density in terms of the number of data points within a specific region. Furthermore, *LOF* is only sensitive to the density of the area and ignores the orientation and the shape of the area [38]. Figure 2.5 shows the basic idea of *LOF*.

- *LOCI*: Local Correlation Integral is a variation of a local density-based method that uses the precise definition of density $M(\bar{x}, \epsilon)$ of a data point $\bar{x}$ in terms of the number of data points within a predefined radius epsilon around a point. In other aspects, this method is similar to *LOF* in terms of using the local relations while defining the score of a data point [64].

The difference between proximity based methods is the way proximity is defined. However, the main difference between distance-based and the other two methods is the level of granularity of analysis [3]. In particular, clustering and density based methods abstract the data by different forms of summarization; in order to compute

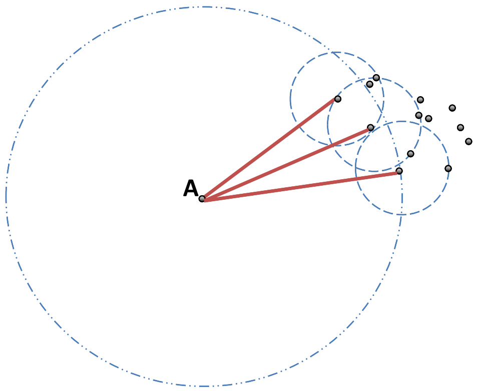


Figure 2.5: Point A has a high LOF score because its density is lower than its neighbours densities. Dotted circles show the distance to each point's third nearest neighbour.

the outlier score, only the distance of a point from its cluster centroid or the points in its local density area is computed. On the other hand, a distance-based algorithm with full granularity computes the distance of a point from all other points in the dataset.

In clustering and density-based methods, the partitioning of the points and space is predefined and data points are compared with these predefined aggregation. This makes distance based methods better fit to distinguish between noise and anomalies because the noisy data points will be included in the distance evaluations rather than the cluster centroids. However, it is possible to modify cluster-based approaches to include the effects of noise. In this case, the two approaches will have very similar schemes.

2.2 Outlier Detection Methods for Structured Data with Propositional Approach

The data used in this type of methods is structured and the idea is to use to extract structured and meaningful features and employ those features in propositional outlier detection task.

2.2.1 Feature-based Methods

Feature-based Methods use the graph representation of data to extract graph-centric features for outlier detection task. An example of this type is a technique called ODDBALL, introduced by Akoglu *et al.* [5]. In this work they define *egonet* as the immediate neighbourhood around a node, in other words, an egonet is the induced 1-step sub-graph for each node as shown in Figure 2.6. They extract egonet-based features to discover patterns from the graph structure in order to define normality. Outliers are the nodes that deviate from those patterns.

In previous applications of single-table outlier analysis methods to structured data, the data were manually propositionalized by aggregating information about single attributes. For example, Breunig *et al.* counted the total number of goals scored by a player in a season as an attribute for outlier analysis [12]. Counts for single attributes in isolation are basically equivalent to the unigram-term frequency method. Manual propositionalization is limited because it becomes very difficult for attributes that represent interactions between features (e.g., the bigram method below).

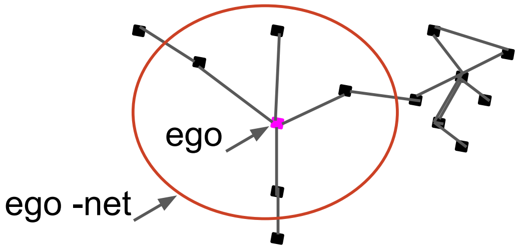


Figure 2.6: ego and ego-net in a toy graph.

2.3 Outlier Detection Methods for Relational Data with non-propositional Approach

In the previous section, we explored a few of the many outlier detection methods that are designed for “flat” data. However, many real-world datasets have some sort of a structure. For example, social network data consists of individuals of different types where each individual is characterized by various sets of attributes. There are many applications of outlier detection that have a structured characteristic where the data consist of several interrelated data types. Therefore, instead of looking for individuals with values that deviate from specific variables, the focus is to find individuals with deviating structures. This section focuses on methods that have been designed for relational data.

2.3.1 Supervised

The main idea in this type of methods is to use the structure of the data to assign objects to normal and abnormal classes. In the following, we overview a few examples of this type of methods.

Relational Classifier

By extending the Markov networks to the relational setting, Taskar *et al.* drive a conditional distribution over the labels of all the individuals given the relational structure and the content attributes [89]. By using the conditional likelihood of the labels given the features, they could improve the classification accuracy.

Rule-Based Classifiers

Rule-based outlier detection methods extract rules that define the normal behaviour of the general population. At a given level of support and confidence, a test instance not consistent with any of the rules is identified outlier. An associated confidence has been assigned to each rule which is propositional to the ratio between the number of correctly classified training instances and a total number of training instances. Decision Trees are commonly used for rule learning [22].

Mahoney *et al.* try to overcome the common problem of intrusion detection techniques: the inability to detect novel attacks. They use Prediction by Partial Matching (*PPMC*) to model normal behavior from attack free network traffic [55] and extract a set of attributes for each event. They then induce a set of conditional rules that have a very low probability of being violated, according to a model learned from normal traffic in the training data.

2.3.2 Unsupervised

Maervoet *et al.* applied a relational frequent pattern miner to automatically discover a set of rules from data; exceptions from these rules are considered potential outliers and are passed to the next step for human expert evaluation [54]. They built a tool to look for regularities in geographical data using the WARMR algorithm. WARMR is based on a breath-first search of the pattern space and searches the space beginning from the most general patterns. First, it searches for rules that describe the regularities and all violations are defined as outliers.

There is other research that employs rule mining to identify outliers. The rule $X \rightarrow Y$ holds in the transaction set D with *confidence* $c\%$ if $c\%$ of transactions in D that contain X also contain Y. The rule $X \rightarrow Y$ has *support* of $s\%$ in transaction set D, if $s\%$ of transactions in D contain $X \cup Y$.

Based on the way they use the identified rules they can be categorized as follows:

- **Rules with minimum support, minimum confidence**: that generates *frequent* itemsets (i.e. those that produce rules with support higher than minsup) by joining the frequent itemsets of the previous path and pruning those subsets that have a support lower than minimum support. Therefore, in order to generate the rules that have low support minsup must be set very low, this increases the running time of the algorithm and generates a lot of redundant rules [4].

- **Rules with low support but high confidence** [45] define the sporadic rules as the ones with low support and high confidence to find a rare but strong association. For example, a rare association of two symptoms indicating a rare fatal disease. In order to do so they adopt an Apriori-Inverse approach which similar to Apriori algorithm and is based on a level-wise search. However, they invert the downward-closure principle of the Apriori algorithm and instead of all subsets of rules being over minsup, it returns all subsets that are under maxup.

- **Exception rule mining**: There are lots of methods to extract exceptional rule from data. Suzuki *et al.*, propose a method to discover a set of interesting rule pairs from a dataset [88]. Hussain *et al.* define interestingness with respect to already mined rules and evaluate the rule's interestingness with respect to its support and confidence. In this area a similar problem of low supports exists [36].

Outlier Detection in ILP

Inductive Logic Programming is an important field at the intersection of machine learning and logic programming and its goal is to induce relation descriptions of data in the form of logic programs. ILP has been used for relational outlier detection. One approach views an example as anomalous if it is not covered by a learned set of rules [10]. It logically harmonizes the background theory with the observations at hand and the main interest is singling out the set of observations that do not behave as predicted from the background knowledge. Their definition of the outlier is based on a set of observations rather than a single observation. They defined outliers as given a set of P^{rls} which encode the general knowledge about the world and P^{obs} to be a set of facts encoding some observed aspects of the current status of the world. The structure P is defined as $P =< P^{rls}, P^{obs} >$. A set of O of observations is anomalous according to the general theory P^{rls} and the other facts in $P^{obs} \backslash O$.

Another work measures the difference in generality between a rule set learned with the anomalous examples, and a rule set learned without [9]. Intuitively, if a subset of examples does not comply with a background theory and the whole set of examples, then this means that the hypothesis induced in the absence of this subset is significantly more general than the hypothesis induced when the examples are seen. Given a subset of examples O of ε, they argue that the compliance of these examples with $\varepsilon \cup \beta$ and $\bar{\varepsilon} \cup \beta$ can be exploited in order to understand if the set O contains abnormal observation.

Community-based methods

Community-based methods are based on finding well-connected groups of individuals. Outliers are the individuals that do not clearly belong to one community. Sun *et al.* use proximity of nodes in the graph to detect anomalies in bipartite graphs. They define outliers as "bridge" nodes/edges that do not fit into any community. They first find the community of a node, also referred to as "neighborhood" of a node, by using random-walk-with-restart Personalized PageRank (PPR) score. The neighbourhood of the given node consists of nodes with high PPR scores and then they define a metric to quantify the level of a given node to be a bridge node [85].

Gao *et al.* proposed a probabilistic model to interpret normal objects and outliers where the object information is described by some generative mixture model. They use k

components to describe the normal behaviour and one component for outliers. Community components are assumed to be drawn from Gaussian or multinomial distribution, while distribution for outlier component is uniform [27].

Muller *et al.* developed an outlier detection technique called GOutRank for heterogeneous databases and attributed graphs [61]. Their main insight is that complex anomalies could be revealed in a subset of relevant attributes. Their previous work, OutRank, focused on high dimensional data and did not take into account graph substructure in assigning outlierness score to the nodes. However, GOutRank extracts the hidden potential of graph clustering and detects complex outliers which deviate only with respect to a local sub-graph and subset of relevant attributes.

2.4 Outlier Detection Methods for Other Types of Structured Data

Multi-dimensional OLAP

The multi-dimensional data model defines numeric measures for a set of dimensions. A seminal approach to explore a multi-dimensional data-cube was presented by Sarawagi *et al.* They perform an analyst's search for anomalies guided by pre-computed indicators of exceptions at various levels of detail in the cubes [78]. This enables users to notice abnormal patterns in the data at any level of aggregation. They annotate every cell in all possible aggregations of a data cube with a value that shows the degree of surprise that the quantity in the cell holds. They define a different degree of surprises with respect to the position of the cells and find exceptions at all levels of aggregation.

Attribute Outlier

Koh *et al.* introduce the notion of correlated subspaces that leverage the hierarchical structure of XML to derive groups of attributes that are logically correlated in XML [44]. In order to define the extent of outlierness of a target attribute, they define two correlation-based outlier metrics. One metric quantifies the co-occurrence of a target attribute with its neighbours and the co-occurrence of its neighbours in its absence. The lower the value of this metric is, the higher the degree of outlierness will be. The other metric measures the conditional probability: given the neighbours of a target attribute the probability that target value is introduced in the dataset is computed. Depending on the metric specification of users, the outlier score is computed based on the first or second metric.

2.5 Limitations of Current Outlier Detection Methods

Parametric models assume a specific distribution of the data and then learn the parameters to fit the data. Most of the time one of two scenarios occurs: The assumed generative model is too restrictive and the data is not likely to fit the model well, therefore, many data points will be reported as outliers. The second scenario occurs when the model is too complex and the number of parameters is large. This case often results in overfitting the data. Parametric methods are not efficient when datasets are large since these methods use iterative EM algorithm that scans the entire data in each iteration of E and M steps.
Most of the parametric methods lack interpretability, however, this issue may not be a problem for all parametric methods. For example, a simple version of Gaussian model may be described simply and intuitively in terms of features of the original data. Most proximity-based methods use distance to define outliers. Methods that summarize the data will not perform well in identifying true anomalies from noisy regions with low density. These methods need to combine global and local analysis in order to find the true outliers. In proximity based methods particularly, the higher level of granularity results in greater accuracy. However increasing the granularity causes curse of dimensionality and makes the algorithm inefficient (in the worst case a distance based methods with full granularity can require $O(N^2)$ in a dataset with N records). Indexing can be used in order to prune the search for the outliers but it cannot be very effective when data is sparse. Another limitation of these methods is the quality of the outlier detection in high dimensional data where all points become almost equidistant from one another and contrast in distance is lost [34].

Chapter 3

Background, Data Model, Datasets and Statistical Models

In this chapter we first provide an introduction to object-relational data model, which is one of the main data models for structured data and has been used to represent our data throughout this dissertation. In section 3.2 we discuss three synthetic and two real-world datasets that have been designed to evaluate the performance of our outlier detection methods. In section 3.4 we review the necessary background on statistical models that have been employed in this dissertation. Section 3.5 explains the evaluation techniques we have utilized to examine the outlier detection methods that will be introduced in chapter 4 and 5.

3.1 Object-relational Data Model

The main characteristics of objects that we utilize in this dissertation 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 Relationship.* 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 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.* Object-relational data can be represented as a network as shown in Figure **??**.

Example A query for computing object data for Arsenal includes the selection condition $TeamID = Arsenal$. Note that object data features all the data of the object as well as the data from more complex objects within that object.

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 3.1 that contains only rows where TeamID = *WA*; see Table 3.2. In database terminology, an object database is like a view centered on the object.

3.1.1 Notation and Definition

We adopt a term-based notation for combining logical and statistical concepts [70, 43]. A functor is a function or a 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 **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 $f(\tau_1, \ldots, \tau_k)$ where f is a functor and each τ_i is a first-order variable or a constant denoting an object. A term/literal/formula 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) [43]. A **grounding** replaces each first-order variable in a term/literal/formula 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.

Consider a joint assignment (also known as conjunctive formula or conjunction in logic) $P(\boldsymbol{V} = \mathbf{v})$ 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. 3.3). The basic populations are teams, players, and matches with corresponding first-order variables T, P, M. As shown in Table 3.1, the groundings count can be visualized in terms of a groundings table [81], also called a universal schema [76]. The first three column headers show first-order variables ranging over different populations. The remaining columns represent terms. Each row represents a single grounding and the values of the ground terms are defined by the grounding. In terms of the grounding table, the grounding count of a joint assignment is the number of rows that satisfy the conditions in the joint assignment. In the network view representation of data, grounding count represents the number of subgraphs that satisfy a given conjunction as shown in Figure **??**. The database frequency is the grounding count divided by the total number of rows in the groundings table. Counts are based on the 2011-2012 Premier League Season. We count only groundings $(\mathit{team}, \mathit{match})$ such that *team* plays in *match*. Each team, including Wigan Athletics, appears in 38 matches. The total number of team-match pairs is $38 \times 20 = 760$.

Example Figure 3.1 shows an example database. The ground literal

$$(\mathit{ShotEff}(P, M) = \mathit{Low})\{P\backslash 123, M\backslash 1\} = (\mathit{ShotEff}(\mathit{123}, \mathit{1}) = \mathit{Low})$$

evaluates as true in this database. For the grounding count we have

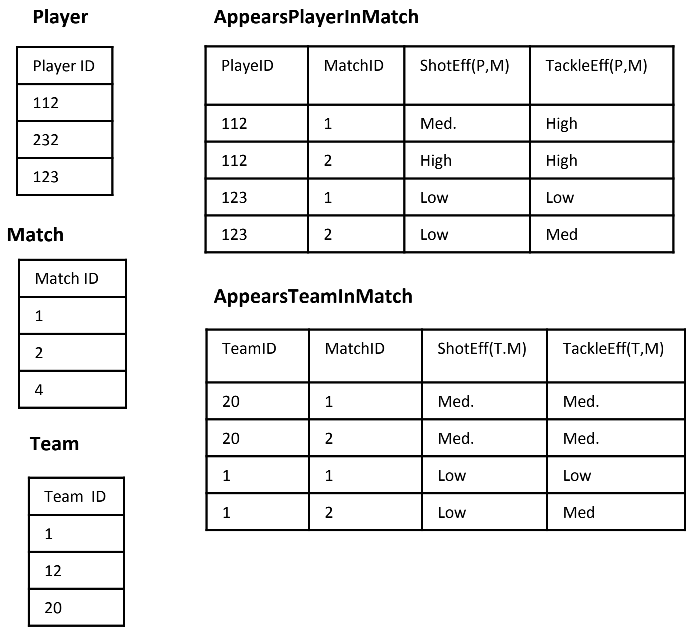
$$\#_{\mathcal{D}}(\mathit{ShotEff}(P, M) = \mathit{Low})\{P\backslash 123\}) = 2.$$

Table 3.1: 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 3.2: Sample object data table, for team $T = \mathit{WA}$.

MatchId M	TeamId $T = \mathit{WA}$	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	...	...	...	...	



Player

Player ID
112
232
123

AppearsPlayerInMatch

PlayeID	MatchID	ShotEff(P,M)	TackleEff(P,M)
112	1	Med.	High
112	2	High	High
123	1	Low	Low
123	2	Low	Med

Match

Match ID
1
2
4

AppearsTeamInMatch

TeamID	MatchID	ShotEff(T.M)	TackleEff(T,M)
20	1	Med.	Med.
20	2	Med.	Med.
1	1	Low	Low
1	2	Low	Med

Team

Team ID
1
12
20

Figure 3.1: An example database

Table 3.3: Example of grounding count and frequency in the Premier League, for the conjunction $passEff(T,M) = high$, $shotEff(T,M) = high$,$Result(T,M) = win$.

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	$7/38 = 0.18$

Table 3.4: Instances of the conjunction: $passEff(T,M) = high$, $shotEff(T,M) = high$, $Result(T,M) = win$ in the network representation of Figure 3.2.

Team	Player	MatchID	$shotEff(T,M)$	$passEff(T,M)$	$Result(T,M)$
Manchester United	Javier Hernandez	119	high	high	win
Manchester United	Anderson	119	high	high	win

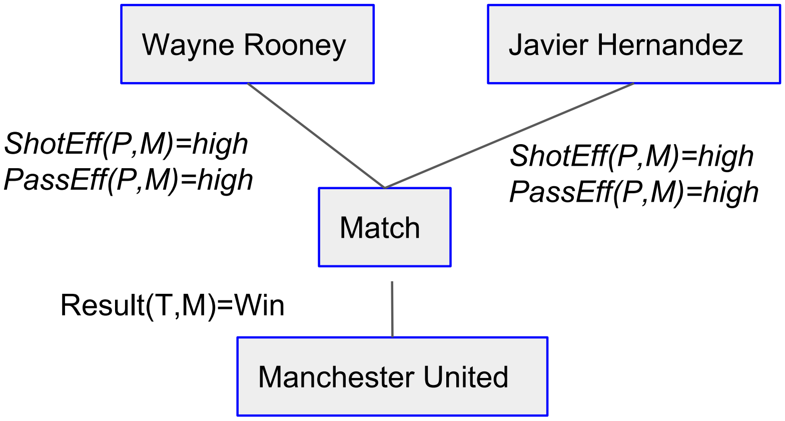


Figure 3.2: In the network representation, We have shown two instances of the conjunction: $\textit{passEff}(T, M) = hi$, $\textit{shotEff}(T, M) = hi$, $\textit{Result}(T, M) = win$. We use the conjunctions to define subgraphs.

3.2 Synthetic Datasets

We generated three synthetic datasets with normal and outlier players using the distributions represented in the three Bayesian networks of Figure 3.3. 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 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. See Figure 3.3(a).

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. See Figure 3.3(b).

Single features 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. See Figure 3.3(c).

We used the *mlbench* package in *R* to generate synthetic features in matche and followed 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.

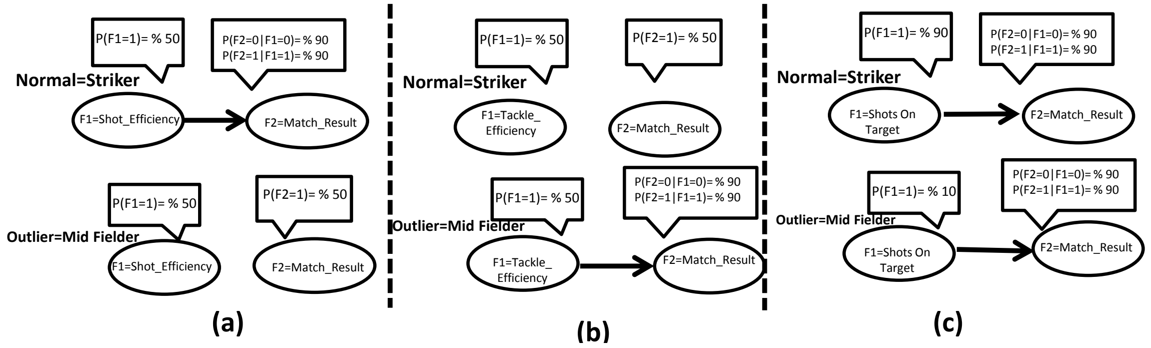


Figure 3.3: Illustrative Bayesian networks. The networks are not learned from data, but hand-constructed to be plausible for the soccer domain. (a) High Correlation; (b) Low Correlation; (c) Single Attributes.

3.3 Real-world Datasets

In this dissertation, real world data tables are prepared from Opta data [56] and IMDb [37].Table 3.5 lists the populations and features. Table 3.6 shows summary statistics for the datasets.

Soccer Data The Opta data were released by Manchester City. It lists all the ball actions within each game, by each player, for the 2011-2012 season. Number of goals, passes, fouls, tackles, saves and blocks, and also the position assigned to a player in a match, are examples of the information associated with each player. For each player in a match, our dataset 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, $Appears\_Player(P, M)$, $Appears\_Team(T, M)$. We refer to the Premier League data as the *Soccer* dataset. Table 3.6 shows summary statistics for the datasets.

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 in a set of

Individuals	Features
Soccer-Player per Match	$TimePlayed$,$Goals$,$SavesMade$, $ShotEff$,$PassEff$,$WinningGoal$, $FirstGoal$,$PositionID$, $TackleEff$,$DribbleEff$, $ShotsOnTarget$
Soccer-Team per Match	$Result$,$TeamFormation$, $\sum Goals$,$\mu ShotEff$,$\mu PassEff$, $\mu TackleEff$,$\mu DribbleEff$.
IMDB-Actor	$Quality$, $Gender$
IMDB-Director	$Quality$,$avgRevenue$
IMDB-Movie	$year$,$isEnglish$,$Genre$,$Country$, $RunningTime$, $Rating$ by User
IMDB-User	$Gender$, $Occupation$.

Table 3.5: Attribute features.

Premier League Statistics		IMDB Statistics	
Number Teams	20	Number Movies	3060
Number Players	484	Number Directors	220
Number Matches	380	Number Actors	98690
avg player per match	26.01	avg actor per movie	36.42

Table 3.6: Summary statistics for the IMDb and the Premier League datasets

Table 3.7: Outlier/normal objects in real-world datasets.

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

compressed text files. We preprocessed the data like, [67], to obtain a database with seven tables, one for each population and one for the three relationships: $\mathit{Rated}(\mathit{User}, \mathit{Movie})$, $\mathit{Directs}(\mathit{Director}, \mathit{Movie})$, and $\mathit{ActsIn}(\mathit{Actor}, \mathit{Movie})$.

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 only that class. We then rank all individuals from the normal class, together with all objects from a contrast class treated as outliers, in order to test whether an outlier score recognizes objects from the contrast class as outliers. Table 3.7 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. Chapter 5 describes a few in-class outliers. In the soccer data we considered only individuals who played more than 5 matches out of a maximum 38.

3.4 Statistical Models

We use notation and terminology from previous work [70, 16, 53, 20]. While we do not introduce any new terminology, we combine concepts from different areas, such as propositionalization and log-linear models.

3.4.1 Bayesian Network

We adopt the Parametrized Bayes net (PBN) formalism [70] that combines Bayes nets with logical syntax for expressing relational concepts.

A **Bayesian Network (BN) structure** B is a directed acyclic graph (DAG) whose nodes comprise a set of random variables [65]. Depending on the context, we interchangeably refer to the nodes and variables of a BN. Fix a set of variables $\boldsymbol{V} = \{f_1, \ldots, f_n\}$. The possible values of f_i are enumerated as $\{v_{i1}, \ldots, v_{ir_i}\}$. The notation $P(f_i = v) \equiv P(v)$ denotes the probability of variable f_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 f_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

$$\ln P(\boldsymbol{V} = \mathbf{v}; B, \boldsymbol{\theta}) = \sum_{i=1}^{n} \ln \theta(\mathbf{v}_i | \mathbf{v}_{\mathbf{pa}_i}) \tag{3.1}$$

where $\mathbf{v}_i$ resp. $\mathbf{v}_{\mathbf{pa}_i}$ is the assignment of values to node f_i resp. the parents of f_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 $\boldsymbol{\theta}$ 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 f has value v as $P(f = v; B, \boldsymbol{\theta}) \equiv \theta(v)$. In the following chapters, we use the term Bayesian network model to refer to a network structure with parameters (i.e., a pair $(B, \boldsymbol{\theta})$); for brevity, we also use the terms "Bayesian network" or "model". A **Parametrized Bayesian Network Structure** (PBN) is a Bayesian network structure whose nodes are PRVs [70]. The relationships and features in an object database define a set of nodes for Bayes net learning; see Figure 3.4.

Example. Figure 3.4 shows an example of a Bayesian network model and associated joint and marginal probabilities.

3.4.2 Markov Logic Network

A Markov Logic Network (MLN) [20] is a set $\{(\phi_1, w_1), \ldots, (\phi_m, w_m)\}$ where ϕ_i is a formula, and each w_i is a real number called the weight of ϕ_i. The MLN semantics views a formula with logical variables as a feature template that is instantiated by ground formulas. The number m of formulas is independent of the size of the instantiated MLN. The log-linear likelihood of a possible world/database is proportional to the weighted sum, over all formulas, of the grounding count of each formula in the given database:

$$P(D) \propto \exp(\sum_{i=1}^{m} w_i \cdot \#_{\mathcal{D}}(\phi_i)) \tag{3.2}$$

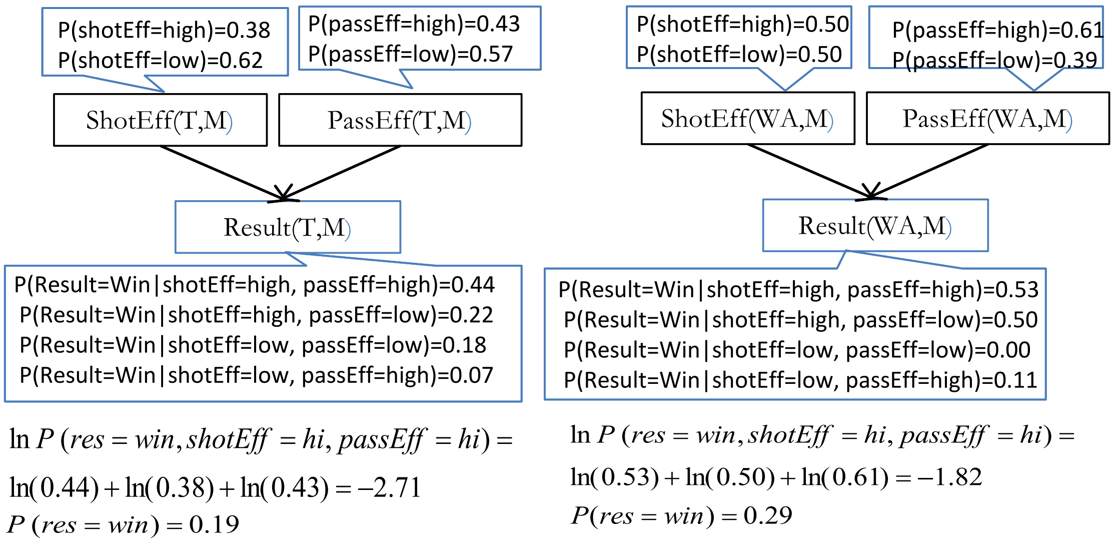

P(shotEff=high)=0.38
P(shotEff=low)=0.62
P(passEff=high)=0.43
P(passEff=low)=0.57
ShotEff(T,M)
PassEff(T,M)
Result(T,M)
P(Result=Win|shotEff=high, passEff=high)=0.44
P(Result=Win|shotEff=high, passEff=low)=0.22
P(Result=Win|shotEff=low, passEff=low)=0.18
P(Result=Win|shotEff=low, passEff=high)=0.07
$\ln P\,(res = win, shotEff = hi, passEff = hi) =$
$\ln(0.44) + \ln(0.38) + \ln(0.43) = -2.71$
$P\,(res = win) = 0.19$
P(shotEff=high)=0.50
P(shotEff=low)=0.50
P(passEff=high)=0.61
P(passEff=low)=0.39
ShotEff(WA,M)
PassEff(WA,M)
Result(WA,M)
P(Result=Win|shotEff=high, passEff=high)=0.53
P(Result=Win|shotEff=high, passEff=low)=0.50
P(Result=Win|shotEff=low, passEff=low)=0.00
P(Result=Win|shotEff=low, passEff=high)=0.11
$\ln P\,(res = win, shotEff = hi, passEff = hi) =$
$\ln(0.53) + \ln(0.50) + \ln(0.61) = -1.82$
$P(res = win) = 0.29$

Figure 3.4: 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.

This semantics defines a joint distribution over descriptive attributes of entities, links between entities, and attributes of those links. Domingos *et al.* discuss the representational power of this semantics [20].

3.4.3 Structure Learning

In this dissertation, for Bayesian network structure learning we employ the Learn-and-Join (LAJ) algorithm. This is a state-of-the-art structure learning algorithm, especially well-suited for datasets with many descriptive attributes such as those we used in our evaluation [41, 81].

The LAJ algorithm employs an iterative deepening strategy, which can be described as follows for object data. The algorithm learns a set of interrelated BNs. The initial step is to learn one BN for each object class. The BN for one class represents associations among the features of the objects in the class only. The algorithm then learns a BN for each pair

Table 3.8: MLN formulas derived from the toy Bayesnet shown in Figure 3.4

Formula
$Result(T,M) = win \wedge ShotEff(P,M) = high \wedge PassEff(P,M) = high$
$Result(T,M) = win \wedge ShotEff(P,M) = high \wedge PassEff(P,M) = low$
$Result(T,M) = win \wedge ShotEff(P,M) = low \wedge PassEff(P,M) = low$
$Result(T,M) = win \wedge ShotEff(P,M) = low \wedge PassEff(P,M) = high$
$ShotEff(P,M) = high$
$ShotEff(P,M) = low$
$PassEff(P,M) = high$
$PassEff(P,M) = low$

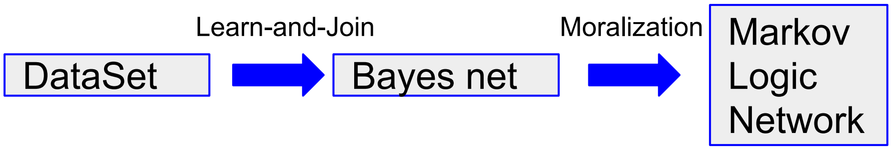


Figure 3.5: Learning an Markov Logic Network from an input relational database.

of linked components, such that the BN for the pair inherits the edges of the BNs for the objects. Next, the algorithm learns a BN for each triple of objects related by a path of length 2, where edges are inherited from the BNs for the relevant pairs, etc. The algorithm stops when increasing the path length leads to no new edges being learned. The multiple Bayesian networks are then merged into a complete Bayesian networks for all objects. The LAJ algorithm takes as input a relational database and outputs a Parametrized Bayes net structure.

The Parametrized Bayes net learned from LAJ algorithm can then be converted to an MLN set of clauses using moralization method described by Domingos and Richardson[21]. Moralization converts the probabilistic clauses defined by Bayes net to conjunctions of literals as shown in Figure 3.5. An example of this conversion is shown in Table 3.8.

3.5 Evaluation Techniques in Outlier Detection

Measuring the effectiveness of outlier detection methods is not often an easy task. Most of the time ground truth information, that shows which data points are outliers, is unavailable.

Several techniques have been employed in literature to evaluate the performance of outlier detection methods:

1. Intuitive evaluation: case studies have been extensively used in literature to evaluate outliers [3]. In chapter 4 we use this method of evaluation for top n ranked detected outliers and we try to explain and make sense of the detected outliers.

2. Synthetic data generation: another approach to evaluate anomaly detection methods is generating synthetic data and inject synthetic outliers into the data [3]. We have designed and developed three synthetic datasets as it was discussed in Section 3.2.

3. Anomaly injection: anomalies are injected into the real-world datasets. Outlier detection methods are expected to detect the injected data points as outliers [6]. We employ this approach in our real world datasets. The disadvantage of this evaluation metric is that the real world data may also contain certain type of anomalies, known as in-class outlier. However, this metric treats only the injected data points as true positive and will score anything other than those as false positive.

4. Internal Evaluation: in this type of evaluation outlier score has been used to quantify the extremity of data points [69].

Chapter 4

Propositionalization for Unsupervised Outlier Detection in Object-relational Data

In this chapter, we develop a novel propositionalization approach to unsupervised outlier detection for object-relational data. Propositionalization summarizes the information from relational data, that is typically stored in multiple tables, into a single data table. An advantage of propositionalization is that it facilitates leveraging the many previous outlier detection methods that were designed for single-table data. Previous work has employed propositionalization for various applications; Anderson *et. al* [8] use propositionalization to apply clustering algorithms like, KMeans, to multi-relational data. Propositionalization for classification has been extensively explored [46, 68, 51, 48]. ODDBALL extracts patterns from large weighted graphs and then uses those patterns as features to discover anomalous nodes in graph [5].

In this work we develop propositionalization for outlier detection for the case of object-relational data. A novel application of Markov Logic Network structure learning is the basis of our propositionalization method for outlier detection. Alternative propositionalization methods that we evaluate in this work are based on enumerating all conjunctive formulas with, at most, two literals (unigrams and bigrams). Compared to baseline propositionalization methods, Markov Logic propositionalization produces the most compact data tables whose attributes capture the most complex object-relational correlations. (More complex correlations are represented by longer logical formulas). We apply three representative outlier detection methods (*LOF*, *KNNOutlier*, *OutRank*) to the data tables constructed by propositionalization. For each outlier detection method, Markov Logic propositionalization provided the best average accuracy over all datasets compared to the baseline propositionalization methods.

4.1 Introduction

Many outlier detection methods have been developed for data that is represented in an attribute-value format [3]. This work addresses outlier detection for object-relational data. In a single data table a row represents a data point, a column represents an attribute of a data point, and a table entry represents an attribute value for a data point. Data analysis tools that are built for single data tables, can be leveraged for multiple relational data tables via a pipeline approach: first, convert the object-relational data to a single attribute-value table, then apply the data analysis tool. Since the attribute value representation corresponds to propositional logic, the conversion process is called *propositionalization* [46]. While propositionalization for classification has been extensively explored [46, 68, 51, 48], to our knowledge propositionalization for outlier detection is a new research problem.

Approach. We use Markov Logic Network (MLN) structure learning to construct a single data table from object-relational data. This is a novel application of MLN learning. The format of the resulting data table is an individual-centric representation [53, 68]: we assume that there is a target class of individuals to be ranked as potential outliers (e.g. soccer players or movies). A row in the data table represents the attributes of an individual. Attributes are defined by logical first-order formulas [53]. The more complex the formula, the more relational information is represented by the formula. A feature function maps an individual and a first-order formula to a real value that is the value of the attribute for the individual. For example, we use the number of instantiations or groundings of a formula as such a function. A Markov Logic Network structure is a set of formulas. Our Markov Logic propositionalization method applies the MLN structure learning method, introduced in Chapter 3, to produce a set of formulas, these formulas define attributes for propositionalization. Our approach can be summarized by this equation:

$$\textit{Markov Logic Network Structure} = \textit{Set of Formulas} = \textit{Set of Attributes} \quad (4.1)$$

A baseline comparison method is to enumerate all conjunctive formulas up to a fixed length n as attributes for propositionalization. This is an instance of the recent Wordification approach to propositionalization [68]. Wordification is based on an analogy between text data and relational data, where an n-gram in text data corresponds to a conjunctive formula with n literals. In text analysis, n-grams are often treated as features of a document. Analogously, wordification uses conjunctive formulas up to a fixed length n as features for propositionalization. The disadvantage of this approach is that the number of such formulas grows exponentially with n.

Evaluation. We use synthetic and real-world datasets that were introduced in Chapter 3. Markov Logic propositionalization produces significantly fewer attributes, leading to much smaller data tables for outlier analysis compared to the baseline wordification approach. The MLN attributes capture more complex relational associations (with over 3 literals on average compared to 2 literals for wordification). MLN propositionalization is competitive with wordification: for a given outlier analysis method, the average AUC score over all datasets is best for MLN propositionalization.

We believe that propositionalization for outlier detection is a fruitful application area for other statistical-relational learning generative models in addition to Markov Logic Networks. Our approach can utilize any model class whose structure is represented by logical formulas, or can be easily converted to logical formulas, which includes many statistical-relational models [73, 7, 40, 29].

Contributions. The contributions of this chapter may be summarized as follows.

1. A novel task for relational learning: propositionalization for outlier detection. This facilitates leveraging standard single-table outlier analysis methods for object-relational data.

2. A novel application of Markov Logic Network structure learning to perform this task. MLN structure learning generates a compact yet expressive set of features from object-relational data.

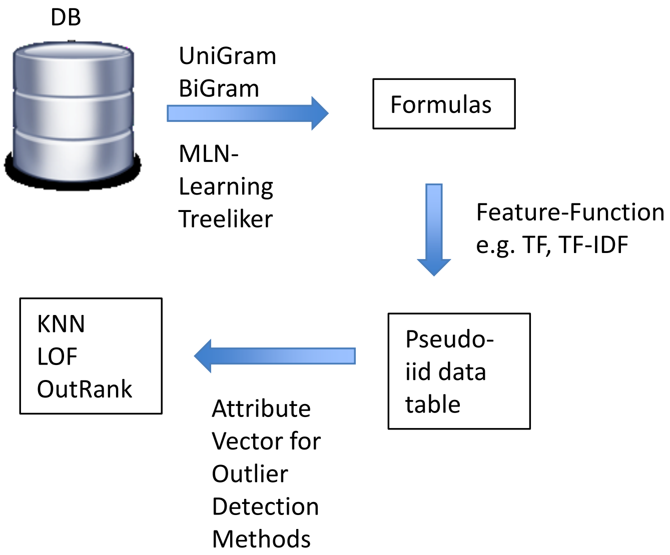


Figure 4.1: System Flow

Table 4.1: An example pseudo-iid data view. For definitions please see text.

Formula →	$SavesMade(P, M) = med \wedge shotsOnTarget(P, M) = lo \wedge ShotEff(P, M) = lo$		$SavesMade(P, M) = med \wedge shotsOnTarget(P, M) = hi \wedge ShotEff(P, M) = hi$	
Feature Function → Player ↓	*TF*	*TF-IDF*	*TF*	*TF-IDF*
Wayne Rooney	4	1.99	12	33.83
David Silva	6	2.99	19	53.57
Robin VanPersie	2	0.99	24	67.67

4.2 Propositionalization, Pseudo-iid data views, and Markov Logic Networks

Figure 4.1 provides an overview of our propositionalization system. Lippe *et al.* [53] describe propositionalization in terms of a pseudo-iid (p-iid) data view. A p-iid data view is a data table in which one row specifies attribute values for one example. Statistical analysis tools, such as outlier analysis methods, that take as input single-table data are applied to the p-iid data view. Since in the relational case the attribute values in different rows are often not independent, Lippi *et al.* coined the term “pseudo-iid ”.

Definition 1 (based on [53]). *Let $\mathcal{D}$ be a relational database. A pseudo-iid (p-iid) data view of $\mathcal{D}$ comprises*

1. *a logical variable E, called the **example variable***

2. a set of examples, where each example consists of a constant in the domain of E

3. a set of ***attributes*** *$F_1, F_2, \ldots, F_d$. An attribute specifies a real number given an example and the database $\mathcal{D}$.*

Lippi *et al.* give a more general definition of p-iid views where examples may consist of tuples rather than a single constant. In our experiments, we used only single individual examples (=constants). In the framework of Lippi *et al.*, an attribute is derived from two components. (1) A conjunctive formula, called a query. The formula can be viewed as a *template* that can be instantiated multiple times for a single example. (2) A function that aggregates the multiple instantiations to derive a real number that is the value of the attribute. Lippi *et al.* introduce two basic feature functions: the instantiation count (how many times the query formula is instantiated) and existence, a 0/1-valued attribute that indicates whether there is some instantiation of the feature for the given individual.

Propositionalization via Markov Logic Networks. Propositionalization is usually applied as a technique for *discriminative* learning in relational data [46]. A new idea in this work is that pseudo-iid data views can also be derived from *generative* models. The generative model we employ is Markov Logic Networks [20]. Markov Logic Network learning provides a way to learn formulas for constructing pseudo-iid views, we refer to this as *MLN propositionalization.* For each example individual, the value of an attribute is determined by a feature function that aggregates the multiple instantiations of the attribute query for the individual in order to derive a real number.

Formula + Feature Function = Attribute Values

The motivation for Markov Logic propositionalization is as follows.

1. Constructing a generative model is one of the major approaches to unsupervised outlier detection [3]. Intuitively, the generative model represents normal behavior in the population.

2. The formulas in the MLN indicate which relations are normally associated, and which are normally independent of each other.

3. Relevant formulas are learned from the data rather than constructed from a fixed a priori set of templates.

Algorithm 1 describes how this propositionalization schema can be applied with Markov Logic Networks.

```
Algorithm 1: Markov Logic Network Propositionalization
    Input: An MLN $\{(\phi_1, w_1), \ldots, (\phi_m, w_m)\}$; Database $\mathcal{D}$; Example logical variable $E$.
    Output: A data matrix $D$. (Pseudo-iid data view.)
    Calls: Feature Function $F$. $F(a, \phi, \mathcal{D})$ returns a number.
  For each individual $a_1, \ldots, a_n$ in the domain of the example variable $E$, add a row to the data matrix $D$.
  For each formula $\phi_1, \ldots, \phi_m$ in the MLN that contains the example variable, add a column to the data matrix $D$.
  for all individuals $a_i$ and formulas $\phi_j$ do
      $D_{ij} := F(a_i, \phi_j, \mathcal{D})$.
  end for
  Return $D$.
```

4.3 Wordification: n-gram Methods

As a baseline for empirical comparison, we present an alternative approach to generating formulas in a pseudo-iid view based on the wordification analogy between relational and text data that is introduced by Lavrac *et al.* [68]. The wordification analogy is as follows:

- A document corresponds to an example individual.
- A word in a document corresponds to a literal.
- An n-gram in a document (i.e., a sequence of n words) corresponds to a conjunction of n literals. In our datasets this was computationally feasible for $n < 3$.
- The term frequency (TF) of an n-gram in a document corresponds to the conjunction grounding count.

Just as a formula can have multiple groundings for an individual in a database, an n-gram can occur multiple times in a document. The wordification analogy suggests using the analog of n-grams in text mining. A range of functions for defining attribute values have been explored in NLP research; perhaps the most widely used is term frequency/inverse document frequency ($TF - IDF$), which down-weights terms that are frequent across documents [68]. The two feature functions we employ in this paper are analogs of TF and $TF - IDF$. For a given w in document d from corpus D, the $TF - IDF$ measure is defined as follows:

Table 4.2: Generating pseudo-iid data views using Feature Functions and Formulas

Feature Function → Formula ↓	*TF*	*TF-IDF*
Unigram	Unigram-TF	Unigram-IDF
Bigram	Bigram-TF	Bigram-IDF
MLN	MLN-TF	MLN-IDF

$$TF - IDF(w, d) = TF(w, d) \times log \frac{|D|}{d \in D : w \in d} \tag{4.1}$$

In sum, we use the following methods for generating formulas in a p-iid data view. All generated formulas are constrained to contain the example variable.

MLN Learn a Markov Logic Network for the given database, then use the learned formulas.

Unigram All single literals.

Bigram All conjunctions of two literals that share at least one first-order variable.

Combining our three formula generating methods with two feature functions defines a space of six methods for constructing a p-iid data view for outlier detection, as illustrated in Table 4.2. Table 4.1 presents an example of a pseudo-iid view with two trigram formulas learned from data and attribute values computed from the real-world data.

4.4 Experimental Design: Methods Used

We evaluate the six methods shown in Table 4.2. The Unigram-IDF approach produced substantially weaker results than Unigram-TF on all datasets, so we omit this method to simplify the presentation. Generating unigrams and bigrams is straightforward given a predicate language. Instantiation counts for term frequencies and inverse document frequencies were computed using MySQL Server version 5.5.34.. The most complex computation is structure learning for MLNs. We use a previously existing algorithm that we briefly review.

MLN Structure Learning. In principle our propositionalization method can employ any MLN structure learning algorithm. In this work, we employ the Learn-and-Join (LAJ) algorithm that was discussed in Chapter 3. This is a state-of-the-art MLN structure learning algorithm, especially well-suited for datasets with many descriptive attributes such as those in our empirical evaluation [41, 81]. Our emphasis is on comparing *learning* formulas with the baseline of *enumerating* all n-grams, so we leave evaluating other MLN structure learning algorithms, such as MLN-Boost [42], for future work.

Outlier Analysis Methods. We applied the following three standard matrix-based outlier analysis methods to the pseudo-iid data views: log, *KNNOutlier* and *OutRank*. These methods have been explained in details in Chapter 2.

These methods represent three fundamental approaches to outlier detection. Both *LOF* and *KNNOutlier* require specifying the value of a k parameter. Following the recommendation of the *LOF* creators [12], we employed the three k-values 10,15,20. Our experiments report the best results. The *OutRank* research of [60] suggest using *DISH* or *PRO-CLUS* as clustering subroutines. Our experiments applied *DISH* [1]. Outrank requires three parameters to be specified, α, ϵ and μ. For these parameters we tested different values in the suggested range and the experiments reports the best results. We used the available implementation of all three data matrix methods from the state of the art data mining software *ELKI* [2].

4.5 Evaluation Results

4.5.1 Performance Metrics Used

We report several properties of the pseudo-iid data views produced by the different methods.

Dimensionality The number of attributes in the pseudo-iid data view.

Attribute Complexity The length of the conjunctions that define the attributes.

Outlier Analysis Run Time How long it takes each outlier method to rank outliers, given the pseudo-iid data view.

Attribute Construction Time How long it takes to build the pseudo-iid view from an input relational database.

Our performance accuracy score for outlier rankings is the area under curve (AUC) of the well-established receiver operating characteristic *ROC* curve. This has been widely used to measure the performance of outlier ranking methods [60]. The relationship between false positive rate (1- Specificity) and true positive rate (Sensitivity) is captured by the ROC curve. Ideally, the best performance is achieved when we have the highest sensitivity and the highest specificity. The maximum values for AUC is 1.0 indicating a perfect ranking with 100% sensitivity and 100% specificity. In order to compute the AUC value, we used the *R* package *ROCR* [82]. Given a set of outlier scores, one for each object, this package returns an AUC value.

The summary of our findings is that MLN propositionalization shows the following advantages and disadvantages. The details follow.

- For a fixed outlier detection method, competitive accuracy over all datasets (the best for *LOF* and *KNNOutlier* tie with Bigram-idf for *OutRank*).
- Compact pseudo-iid data views: substantially fewer attributes (columns) than bigrams, yet average formula length 3.27 or greater.
- Faster outlier analysis due to this compactness.
- There is learning overhead for discovering relevant formulas, but it is small (e.g. 5 minutes for MLN learning vs. 1 minute for bigram construction).

4.5.2 Dimensionality of Pseudo-iid Data Views

Figure 4.2 provides information about the formulas constructed by the different propositionalization methods, and the size of the resulting data table. For unigram resp. bigram methods, the formulas have length 1 resp. 2 by definition. The average formula length for MLNs is above 3 for the soccer data, for the IMDb data above 4. This shows that MLN structure learning finds more complex formulas beyond length 2. For the dimensionality of the resulting pseudo-iid views, there is a big increase from unigrams to bigrams (e.g. from 63 to 1825 for Strikers vs. Goalies). The dimensionality of MLN pseudo-iid data views lies between that of unigrams and bigrams, (e.g. 331 for Strikers vs. Goalies). This shows that MLN structure learning can find complex longer formulas with a relatively small increase in the dimensionality of the resulting pseudo-iid data view, compared to bigrams. The trade-off is that learning a compact set of relevant formulas takes more time than enumerating all formulas up to a fixed length. However, the learning overhead is small (e.g. 5.24 min vs. 1.2 min for Strikers vs. Goalies). The smaller dimensionality can decrease the running time of

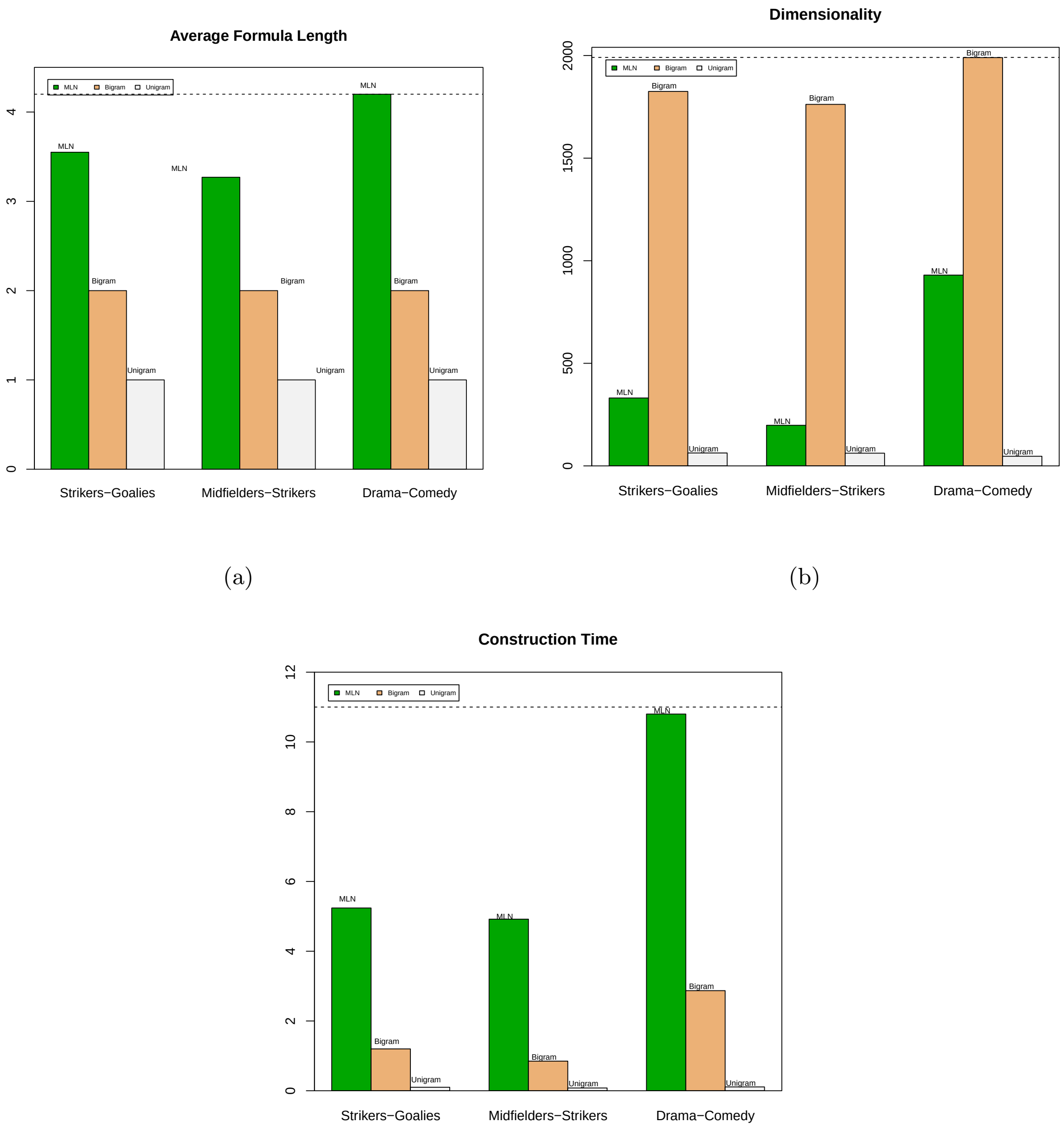
Table 4.3: OutRank running time (ms) given different attribute vectors. Running time for other outlier analysis methods were very similar.

Dataset	$Unigram - TF$	$Bigram - TF$	$Bigram - IDF$	$MLN - TF$	$MLN - IDF$
Drama vs. Comedy	945	855,714	898,438	389,765	397,371
MidFielders vs. Strikers	486	642,261	631,813	18,737	21,466
Strikers vs. Goalies	578	814,807	861,870	55,448	64,837

the outlier detection methods, as shown in Table 4.3. For example, the running time of the Outrank method for Strikers vs. Goalies is 861,870 ms given the Bigram $TF - IDF$ data view, vs. 64,837 for the MLN $TF - IDF$ data view. For the other two outlier detection methods the run-time difference was negligible.

4.5.3 Accuracy

Figures 4.3 and 4.4 present detailed measurements of the AUC-ROC for different outlier propositionalization methods. There is no single propositionalization method that always leads to the best accuracy for all three outlier analysis methods. MLN propositionalization produces the best results on two datasets. It is always close to the maximum AUC score (never less than 0.1 AUC units away). Table 4.4 summarizes the performance of the propositionalization methods for a fixed outlier detection algorithm. The $\mu(AUC)$ column reports the average AUC score over different datsets. A propositionalization method "wins" on a dataset if its AUC is at least 0.01 greater than that of others. A "tie" for first place earns 0.5 points. The total number of points is shown in the Wins columns. MLN-*TF* is revealed to be the best method in terms of average AUC, for all outlier detection methods. *TF* is, in a sense, the natural feature function for MLNs since the likelihood function of MLNs is defined in terms of formula grounding counts (equation 4.1). MLN-*TF* propositionalization scores the most wins when applied with *LOF* or *KNNOutlier* and a tie when applied with *OutRank*. Thus, methods that tend to treat attributes independently, such as *LOF* and *KNNOutlier*, benefit from being provided complex attributes that summarize complex associations. Subspace analysis can utilize complex associations from bigram data, but requires much more time to do so than MLN propositionalization (see Table 4.3).

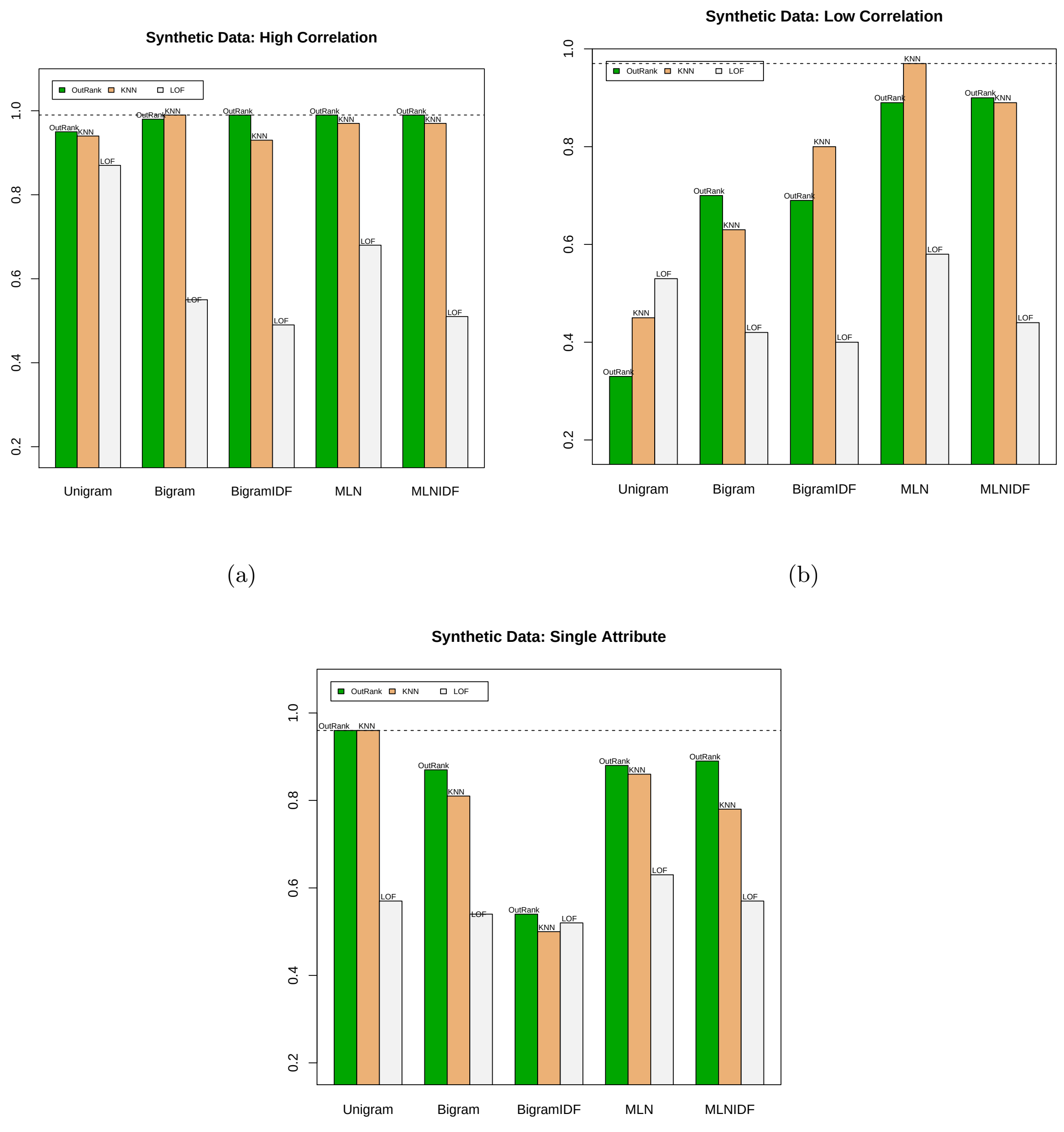


(a)

(b)

(c)

Figure 4.2: Comparison of complexity, dimensionality and construction time for the attributes produced by different propositionalization methods


Synthetic Data: High Correlation

Synthetic Data: Low Correlation

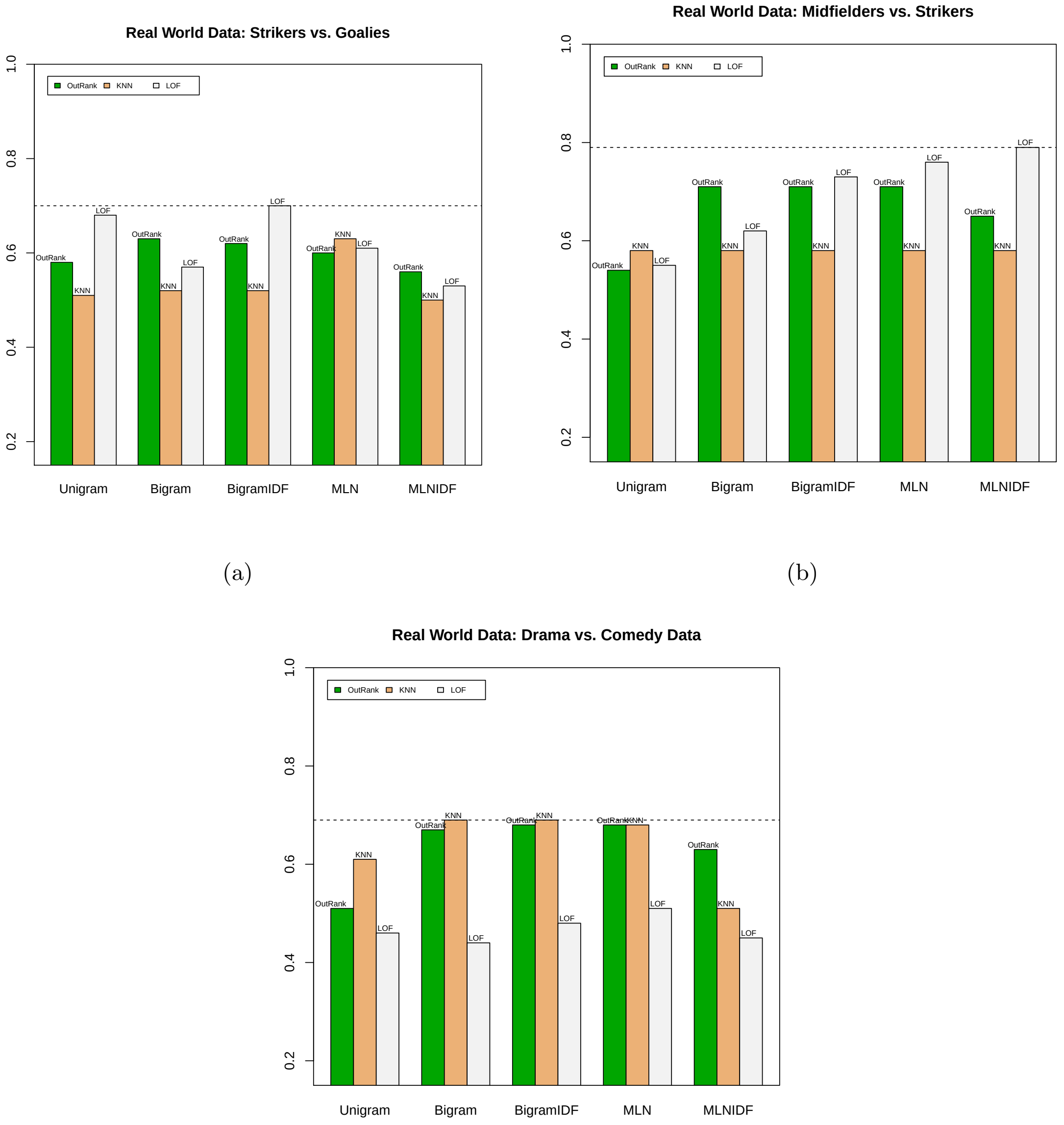
(a)

(b)

Synthetic Data: Single Attribute

(c)

Figure 4.3: Accuracy for different Methods/Attribute Vector in the Synthetic datasets



(a)

(b)

(c)

Figure 4.4: Accuracy for different Methods/Attribute Vector in the Real World datasets

Table 4.4: Summarizing the accuracy results of Figures 4.4 and 4.3: A propositionalization method is scored 1 point if it produces the best accuracy on a dataset, and 0.5 points if it ties. The table shows the total number of wins and average of AUC over all datasets.

Propositionalization → Outlier Detection Method ↓	MLN-TF		Bigram-IDF		Unigram-TF	
	Wins	$\mu(AUC)$	Wins	$\mu(AUC)$	Wins	$\mu(AUC)$
OutRank	**2.50**	**0.79**	**2.50**	0.70	1.00	0.64
KNN	**3.50**	**0.78**	1.50	0.67	1.50	0.67
LOF	**4.00**	**0.63**	1.00	0.55	1.00	0.61

4.6 Comparison With Propositionalization for Supervised Outlier Detection and Log-Likelihood

In this section we compare our novel MLN propositionalization method with a previous propositionalization method that was developed for supervised classification problems. Since, to our knowledge, all previous propositionalization methods are for classification problems, in this section only we consider *supervised* outlier detection, where examples are labelled as "normal" and "abnormal". Given the ground truth labels, supervised outlier detection can be treated as a special case of classification [35]. Supervised outlier detection serves as a benchmark of the accuracy of unsupervised outlier detection: if the unsupervised method comes close to the accuracy of the supervised method, this indicates a good performance of the unsupervised method.

We report experiments with the state-of-the-art Treeliker propositionalization method [48]. We used the implementation of Treeliker available in the ClowdFlows platform [47], which supports the MySQL data format. The HiFi algorithm from [48] has been used with minimum frequency specified as 0.2, maximum size of features to be 10 and sample size as 5. We train and test Treeliker on the same dataset with ground truth labels as classification target. So the way we use Treeliker gives it two advantages over MLN propositionalization: It sees the ground truth labels, and it is tested on the training data. On almost all real-world datasets, this translates into a higher AUC score, except for the Striker-Midfielder problem using *LOF* (0.61 for MLN vs. 0.56 for Treeliker). MLN propositionalization comes close to the Treeliker score, the only substantial difference occurs with *LOF* on the Striker-Goalie problem (0.76 for MLN vs. 0.84 for Treeliker). On the synthetic data, the MLN method performs even better, beating the Treeliker propositionalization by a substantial margin.

Database	Outlier Method	Treeliker AUC value	MLN-TF AUC value
Single Attribute	KNN	0.65	**0.86**
Single Attribute	LOF	0.53	**0.63**
High Correlation	KNN	0.66	**0.97**
High Correlation	LOF	0.57	**0.68**
Low Correlation	KNN	0.65	**0.97**
Low Correlation	LOF	0.56	**0.58**
Striker Goalie	KNN	**0.6**	0.58
Striker Goalie	LOF	**0.84**	0.76
Midfielder Striker	KNN	**0.65**	0.63
Midfielder Striker	LOF	0.56	**0.61**

Table 4.5: Accuracy of Treeliker for different databases and outlier techniques. Bold values represent the cases where Treeliker outperforms other methods.

Given the advantages for the supervised setting, this is a very good performance for MLN propositionalization. We emphasize that this is not a criticism of Treeliker as a propositionalization method, because it is not designed for outlier detection problems. Rather, our conclusion is that new methods provide value for the problem of propositionalization for outlier detection.

4.7 Conclusion

In this chapter we developed a pipeline propositionalization approach where the information from multiple data tables is summarized in a single data table. The key step is to find a set of relevant logical formulas that define conjunctive attributes of potential outlier individuals as sum. We utilized Markov Logic Network learning for this task. In an empirical comparison with the baseline wordification approach of enumerating all conjunctive formulas up to length 2, Markov Logic propositionalization showed several advantages: 1) The set of formulas learned was substantially smaller, leading to smaller data tables and faster outlier detection. 2) The formulas learned were longer, representing more complex relational patterns. 3) For a fixed single-table outlier analysis method, the average detection accuracy was higher.

We view this work as an initial step in the topic of propositionalization for outlier detection, there are several fruitful directions for future work. While Markov Logic networks are a prominent generative model class for relational data, our approach can be used with other generative models; this opens a new application area for statistical-relational learning. Dimensionality reduction techniques can be employed after propositionalization to reduce the size of the data tables before outlier detection methods are used. Propositionalization algorithms that were developed for classification could be adapted for unsupervised outlier detection by using a feature selection score that is relevant for outlier detection, rather than supervised classification.

Another direction for future work is to leverage graph-based descriptive features in our generative model learning process. These features proved to be efficient in discovering patterns for anomaly detection task in ODDBALL [5]. Examples of such features in our datasets include: number of matches a player has played, number of reviews a movie has received, and features related to the extent of interaction between players.

Chapter 5

Metric-based Outlier Detection

In chapter 4 we introduced a pipeline propositionalization method to convert object-relational data to a single data table. By summarizing the information from relational data in one data table, we showed that we can leverage the many previous outlier detection methods that were designed for single data table. The goal of this chapter is to introduce an unsupervised statistical outlier detection method to the case of object-relational data without converting the data to i.i.d. propositional format. For each object there is a probability distribution over the features of related objects. For example, for each soccer team there is a distribution over the features of its players. 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. The likelihood ratio can be improved for outlier detection by applying two transformations: (1) a mutual information decomposition and (2) replacing log-likelihood differences by log-likelihood distances. 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.

5.1 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 [3]. 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.

Approach Figure 5.1 illustrates these concepts and the system flow for computing an outlier score. 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 [29, 81, 20]. 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.*

Evaluation Our code and datasets are available on-line at [74]. Our performance evaluation follows the design of previous outlier detection studies [27, 3], 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).

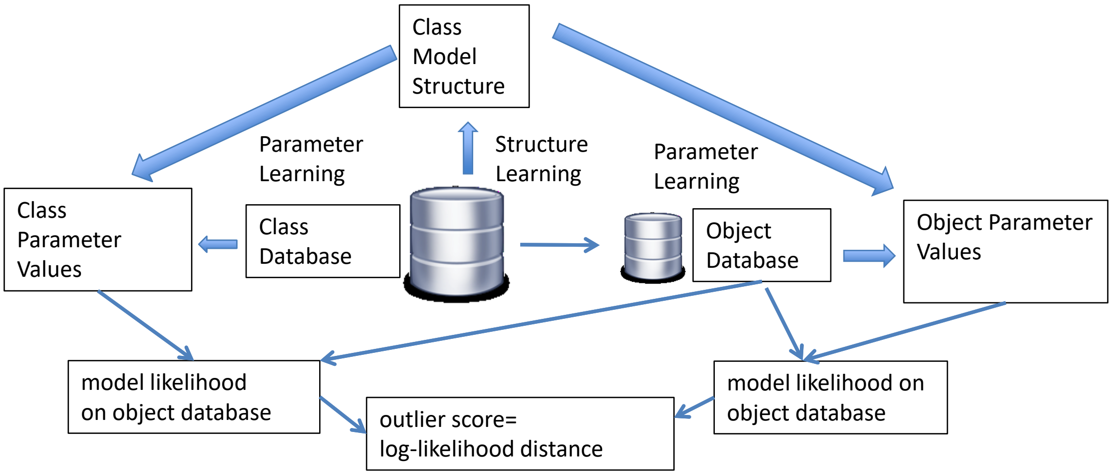


Figure 5.1: Computation of outlier score.

On the synthetic data we have known ground truth. For the real-world data, we use a anomaly injection method discussed in Chapter 2, 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.

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.

Contributions Our main contributions in this chapter 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.

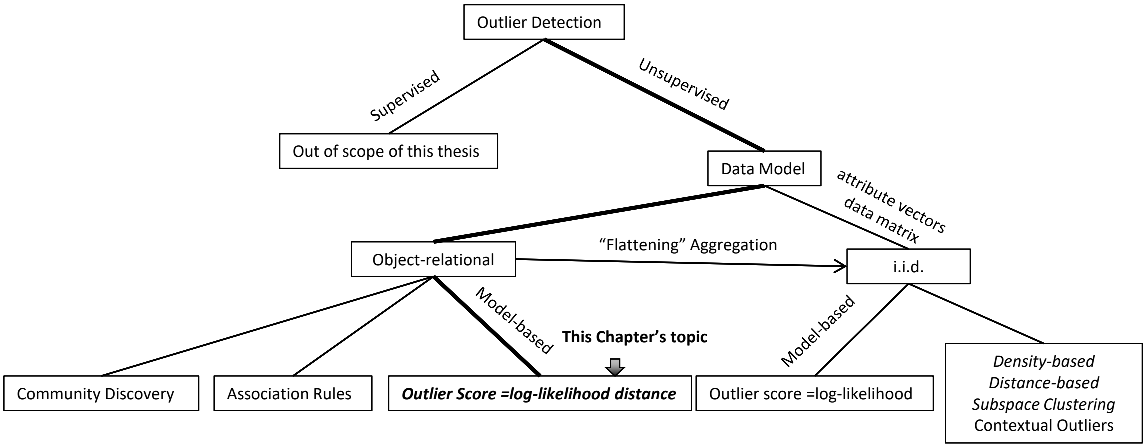


Figure 5.2: A tree structure for related work on outlier detection for structured data. A path specifies an outlier detection problem, the leaves list major approaches to the problem. Approaches in italics appear in experiments.

Table 5.1: Example of grounding count and frequency in the 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 $P_D(\boldsymbol{V} = \mathbf{v})$
A novel aspect of our paper is that we learn model Population	76	$76/760 = 0.10$
A novel aspect of our paper is that we learn model Wigan Athletics	7	$7/38 = 0.18$

5.2 Related Work

In Chapter 2 we reviewed some outlier detection methods designed for the structured data. 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. Figure 5.2 provides a tree picture of where our method is situated with respect to other outlier detection methods and other data models. In the following we review a generative model-based method that has been used as a baseline to our work.

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 [13]. 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 [80], which can be computed as follows for a given Bayesian network and database.

$$LOG(\mathcal{D}, B, \boldsymbol{\theta}) = \sum_{i=1}^{n} \sum_{j=1}^{r_i} \sum_{\mathbf{pa}_i} P_{\mathcal{D}}(v_{ij}, \mathbf{pa}_i) \ln \theta(v_{ij}|\mathbf{pa}_i) \tag{5.1}$$

Equation (5.1) represents the standard BN log-likelihood function for the object data [18], except that parent-child instantiation counts are standardized to be proportions [80]. 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.

The maximum of the pseudo-likelihood (5.1) is given by the empirical database frequencies [80, Prop.3.1.]. In all our experiments we use these maximum likelihood parameter estimates.

Example. The family configuration

$$\mathit{passEff}(T, M) = \mathit{hi}, \mathit{shotEff}(T, M) = \mathit{hi}, \mathit{Result}(T, M) = \mathit{win}$$

contributes one term to the pseudo log-likelihood for the BN Figure 3.3 of Chapter 3. For the population database, this term is $0.1 \times \ln(0.44) = -0.08$. For the Wigan Athletics database, the term is $0.18 \times \ln(0.44) = -0.14$.

5.3 Likelihood-Distance Object Outlier score

We introduce a novel model-based outlier score, that extends the log-likelihood (5.1), using the following notation.

- $\mathcal{D}_C$ is the database for the entire class of objects; cf. Table 3.1 of Chapter 3. 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 3.2 of Chapter 3. This database defines the **object distribution** $P_o \equiv P_{\mathcal{D}_o}$.
- B_C is a Bayesian network structure learned with $\mathcal{D}_{\mathcal{P}}$ as the input database; cf. Figure 3.3(a) of Chapter 3.
- $\boldsymbol{\theta}_C$ resp. $\boldsymbol{\theta}_o$ are parameters learned for B_C using $\mathcal{D}_c$ resp. $\mathcal{D}_o$ as the input database.

Figure 5.1 illustrates these concepts and the system flow for computing an outlier score. First, we learn a Bayesian network structure B_C for the entire population using a previous learning algorithm (see Section 5.5.1 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 (5.1) of the target *database*:

$$LOG(\mathcal{D}_o, B_C, \boldsymbol{\theta}_C). \tag{5.2}$$

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

$$LR(\mathcal{D}_o, B_C, \boldsymbol{\theta}_o) \equiv LOG(\mathcal{D}_o, B_C, \boldsymbol{\theta}_o) - LOG(\mathcal{D}_o, B_C, \boldsymbol{\theta}_C). \tag{5.3}$$

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 [18]. 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 5.4 below provides example computations.

$$ELD_i = \sum_{j=1}^{r_i} P_o(v_{ij}) \left| \ln \frac{\theta_o(v_{ij})}{\theta_C(v_{ij})} \right| + \tag{5.4}$$

$$\sum_{j=1}^{r_i} \sum_{\mathbf{pa}_i} P_o(v_{ij}, \mathbf{pa}_i) \left| \ln \frac{\theta_o(v_{ij}|\mathbf{pa}_i)}{\theta_o(v_{ij})} - \ln \frac{\theta_C(v_{ij}|\mathbf{pa}_i)}{\theta_C(v_{ij})} \right|. \tag{5.5}$$

The first sum (5.4) 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 (5.5) 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.

5.3.1 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

$$\ln \frac{\theta(v_{ij}|\mathbf{pa}_i)}{\theta(v_{ij})}.$$

This relevance term is basically the same as the widely used lift measure [91], 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.* The **mutual information decomposition** shows that each term in the log-likelihood difference (5.3) decomposes into a relevance difference and a marginal difference:

$$\ln \frac{\theta_o(v_{ij}|\mathbf{pa}_i)}{\theta_C(v_{ij}|\mathbf{pa}_i)} = \ln \frac{\theta_o(v_{ij}|\mathbf{pa}_i)}{\theta_o(v_{ij})} - \ln \frac{\theta_C(v_{ij}|\mathbf{pa}_i)}{\theta_C(v_{ij})} + \ln \frac{\theta_o(v_{ij})}{\theta_C(v_{ij})}. \tag{5.6}$$

These differences can have different signs for different child-parent configurations and cancel each other out; see Table 5.2 below for an example. Since our goal is to assess the distinctness of an object, *we do not want differences to cancel out.* Taking distances ,as in Equations (5.4) and (5.5), avoids the undesirable cancellation. 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 component-wise difference of the vectors (0,0) and (1,-1) is 0, but their distance is not.

5.3.2 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 5.2 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).

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

Log-Likelihood Difference *LR* is the log-likelihood difference (5.3) between the class-level and object-level parameters.

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

The next proposition shows that the outlier scores have the standard properties of a divergence measure between probability distributions: they are nonnegative, and 0 if and only if the class and object distributions are the same. Also, the triangle inequality entails that the scores can be ordered by *dominance:* one is guaranteed to be at least as great as another. Dominance means that a divergence potentially provides more discrimination among objects as it maps the set of objects onto a larger range of scores. Our *ELD* score dominates all others. We provide empirical evidence that dominance leads to greater discrimination.

Table 5.2: Baseline comparison outlier scores

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

Proposition 1. *The following hold for any class and object distributions, and each node v_i.*

1. *For any class and object distribution, we have $ELD_i \geq |LR_i| \geq LR_i = LR_i^+ \geq 0$. Also, $ELD_i \geq FD_i$.*
2. *All divergences $ELD_i, |LR_i|, LR_i, LR_i^+, FD_i$ are nonnegative. The divergences are 0 if and only if the object parameters $\boldsymbol{\theta}_o$ and class parameters $\boldsymbol{\theta}_C$ are the same.*

These properties also hold for the divergences $ELD, |LR|, LR, LR^+, FD$ summed over all nodes.

Proof. (Part 1) It is immediate that $ELD_i \geq FD_i$. We show that $LR = LR^+$. Using the marginalization

$$P_o(v_{ij}) \ln \frac{\theta_o(v_{ij})}{\theta_C(v_{ij})} = \sum_{\mathbf{pa}_i} P_o(v_{ij}, \mathbf{pa}_i) \ln \frac{\theta_o(v_{ij})}{\theta_C(v_{ij})} \tag{5.7}$$

and the mutual information decomposition (5.6) it is easy to verify that LR_i^+ simplies to LR. Next, $|LR_i| \geq LR_i$ holds because $a - b \leq |a - b|$ for any numbers a, b. The inequality $ELD_i \geq |LR_i|$ is established as follows.

$$
\begin{aligned}
ELD &= \sum_{j=1}^{r_i}\sum_{\mathbf{pa}_i} P_o(v_{ij},\mathbf{pa}_i)\left|\ln\frac{\theta_o(v_{ij})}{\theta_C(v_{ij})}\right| + \sum_{j=1}^{r_i}\sum_{\mathbf{pa}_i} P_o(v_{ij},\mathbf{pa}_i)\left|\ln\frac{\theta_o(v_{ij}|\mathbf{pa}_i)}{\theta_o(v_{ij})} - \ln\frac{\theta_C(v_{ij}|\mathbf{pa}_i)}{\theta_C(v_{ij})}\right| \\
&= \sum_{j=1}^{r_i}\sum_{\mathbf{pa}_i} P_o(v_{ij},\mathbf{pa}_i)\left|\ln\frac{\theta_o(v_{ij})}{\theta_C(v_{ij})}\right| + \left|\ln\frac{\theta_o(v_{ij}|\mathbf{pa}_i)}{\theta_o(v_{ij})} - \ln\frac{\theta_C(v_{ij}|\mathbf{pa}_i)}{\theta_C(v_{ij})}\right| \qquad (5.8) \\
&\geq \sum_{j=1}^{r_i}\sum_{\mathbf{pa}_i} P_o(v_{ij},\mathbf{pa}_i)\left|\ln\frac{\theta_o(v_{ij})}{\theta_C(v_{ij})} + \ln\frac{\theta_o(v_{ij}|\mathbf{pa}_i)}{\theta_o(v_{ij})} - \ln\frac{\theta_C(v_{ij}|\mathbf{pa}_i)}{\theta_C(v_{ij})}\right| \qquad (5.9) \\
&= \sum_{j=1}^{r_i}\sum_{\mathbf{pa}_i} P_o(v_{ij},\mathbf{pa}_i)\left|\ln\theta_o(v_{ij}|\mathbf{pa}_i) - \ln\theta_C(v_{ij}|\mathbf{pa}_i)\right| \qquad (5.10) \\
&= |LR_i| \qquad (5.11)
\end{aligned}
$$

Here Equation (5.8) follows from Equation (5.7), inequality 5.9 follows from the triangle inequality $|a| + |b| \geq |a + b|$, and Equation (5.10) from Equation (5.6).

(Part 2) The claim is immediate for FD_i. We show that LR_i is nonnegative and 0 only if the object and class parameters associated with node i are the same. Consider a simple Bayes net structure B' comprising the parents of node i, node i, and no other nodes or links. Then LR_i is the log-likelihood difference

$$LR_i = LOG(\mathcal{D}_o, B', \boldsymbol{\theta}_o) - LOG(\mathcal{D}_o, B', \boldsymbol{\theta}_C).$$

The empirical frequency parameters $\boldsymbol{\theta}_o$ uniquely maximize the function $LOG(\mathcal{D}_o, B', \cdot)$ [81], so the difference LR_i is nonnegative, and equals 0 if and only if $\theta_C = \theta_{\boldsymbol{\theta}}$. □

5.4 Two-Node Examples

We provide three simple examples with only two variables/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 3.3 describes the BN representation of the examples. 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.

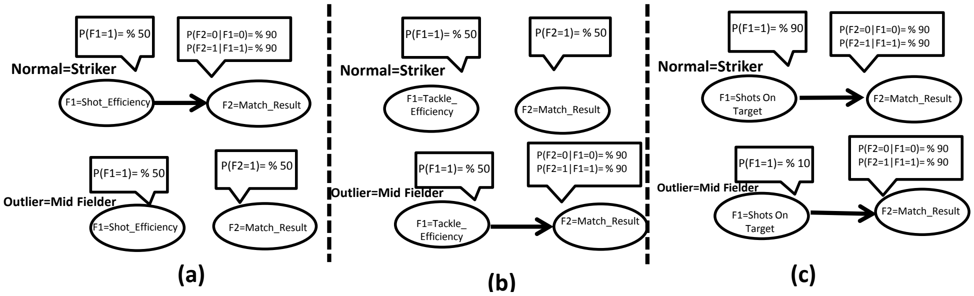


Figure 5.3: Illustrative Bayesian networks with two nodes. The networks are not learned from data, but hand-constructed to be plausible for the soccer domain. (a) *High Correlation:* Normal individuals exhibit a strong association between their features, outliers no association. Both normals and outliers 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.

5.4.1 Computations

Table 5.3 provides the computation of the scores. 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 (5.4) is 0 for each node in both examples. The table illustrates the undesirable cancelling effects in *LR*. In the high correlation scenario 3.3(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 5.3 the log-ratios ln(0.5/0.9) and ln(0.5/0.1) therefore have different signs. In the low correlation scenario 3.3(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.

5.4.2 Visualization

Figure 5.4 provides scatter plots for each synthetic dataset and each comparison outlier metric. The figure is best viewed on screen. As entailed by Proposition 1(Part 1), the *ELD*

Table 5.3: Example computation of different outlier scores for outliers given the distributions of Figure 5.3 (a),(b).

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

(a) High Correlation Case. Figure 5.3(a).

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

(b): Low Correlation Case. Figure 5.3(b).

metric maps players to the largest range of outlier scores. It also provides the best separation of normal from abnormal players: The normal players receive low anomaly scores and hence are clustered to the left of the *ELD* scatter plot, whereas the abnormal players receive high scores and hence are clustered on the right. The $|LR|$ metric also shows a larger range of scores and a better discrimination compared to the LR metric. This illustrates the value of using distances rather than differences. In Section 5.6 we provide an aggregate detection accuracy score that quantifies this values.

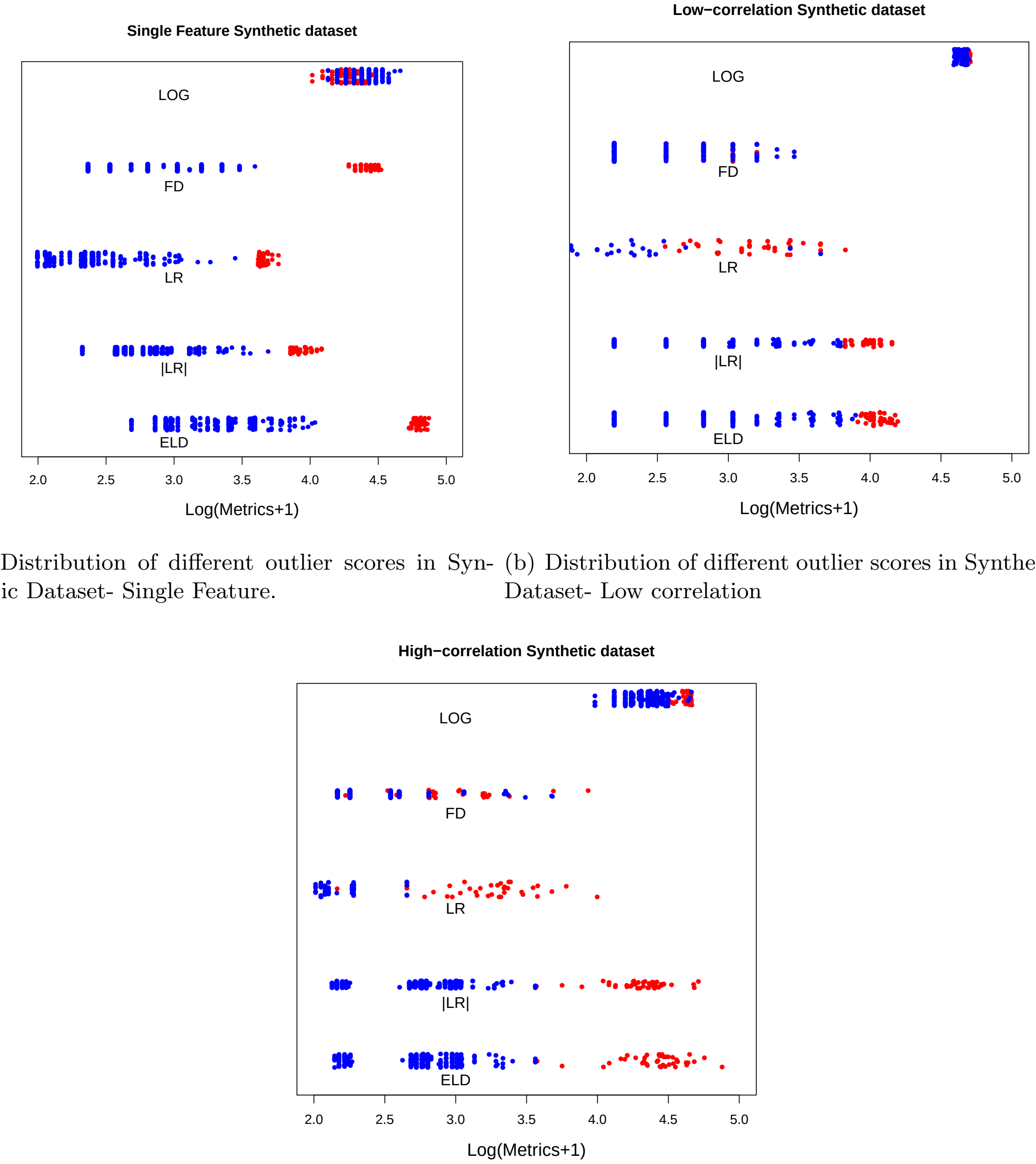
5.5 Experimental Design

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.

5.5.1 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 5.3. For relational Bayesian network structure learning we utilize the previous learn-and-join algorithm (LAJ) that was introduced in Chapter 3.

The second approach is to compare the loglikelihood metric with some of the standards matrix-based outlier analysis methods. One way to convert relational data to a single data matrix is to 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 **propositional-based** (cf. Figures 5.2). For example, this was the approach taken by Breunig *et al.* for identifying anomalous players in sports data [12]. However, as discussed in chapter 4, aggregation tends to lose information about correlations and MLN propositionalization is a better candidate for this conversion. Therefore, instead of aggregating over features, we used features generated by MLN-TF. In chapter 4 we showed that in most datasets, MLN-TF produce better quality features compare to the other propositionalization methods. We evaluated three standard propositional-based outlier detection methods: Density-based *LOF* [12], distance-based *KNNOutlier* [71] and subspace analysis *OutRank* [60]. These represent common, fundamental approaches for propositional data.



(a) Distribution of different outlier scores in Synthetic Dataset- Single Feature.

(b) Distribution of different outlier scores in Synthetic Dataset- Low correlation

(c) Distribution of different outlier scores in Synthetic Dataset- High correlation

Figure 5.4: Visualizing likelihood-based outlier metrics on our three synthetic datasets. We employed log scale to to show the score values of different range in a single plot. To avoid the negative numbers for the values between 0 and 1, we used metric+1. The figure is best viewed in colors. Score values are shown on the x-axis; higher values should indicate more anomalous players. For each dataset and each metric, we provide a 1D scatterplot of the 280 synthetic player scores. In the scatterplot, blue dots represent normal players, and red dots represent anomalous players.

5.6 Empirical Results

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

Computational Cost of the *ELD* Score.

Table 5.4 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 5.5 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 5.4: 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 5.5: 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

Dataset	*ELD*	*|LR|*	*LR*	*FD*	*LOG*
High Correlation	**1.00**	0.99	0.95	0.88	0.98
Low Correlation	**1.00**	0.99	0.98	0.53	0.81
Single Feature	**1.00**	**1.00**	**1.00**	**1.00**	0.79
Strikers vs. Goalies	**0.89**	0.69	0.61	0.73	0.58
Midfielders vs. Strikers	**0.66**	0.64	0.64	0.52	0.64
Drama vs. Comedy	**0.70**	0.67	0.65	0.61	0.63

Table 5.6: AUC of *ELD* vs. other probabilistic scores.

Dataset	*ELD*	*LOF*	OutRank	*KNN Outlier*
High Correlation	**1.00**	0.68	0.99	0.97
Low Correlation	**1.00**	0.58	0.83	0.97
Single Feature	**1.00**	0.63	0.88	0.86
Strikers vs. Goalies	**0.89**	0.61	0.60	0.63
Midfielders vs. Strikers	0.66	**0.76**	0.71	0.58
Drama vs. Comedy	**0.70**	0.51	0.68	0.68

Table 5.7: AUC of *ELD* vs. propositional-based outlier detection methods. The single table used as input for *OutRank*, *LOF* and *KNNOutlier* was generated using the MLN-TF, a propositionalization approach introduced in Chapter 3.

Detection Accuracy

Our experiments provide empirical evidence that in practice *ELD* generally works better than other scores for object outlier detection.

Our performance score for outlier rankings is the area under curve (AUC) of the well-established receiver operating characteristic ROC curve. [25]. This has been widely used to measure the performance of outlier ranking methods [13, 60]. The relationship between false positive rate (1- Specificity) and true positive rate (Sensitivity) is captured by the ROC curve. Ideally, the best performance is achieved when we have the highest sensitivity and the highest specificity. The maximum values for AUC is, 1.0 indicating a perfect ranking with 100% sensitivity and 100% specificity. In order to compute the AUC value, we used the *R* package *ROCR* [82]. Given a set of outlier scores, one for each object, this package returns an AUC value.

Probabilistic Structured methods: Table 5.6 shows the *AUC* values for each probabilistic ranking. On the synthetic data, it ought to be easy to distinguish the outliers. Single feature is the easiest dataset and most metrics except log are successful in perfectly detecting outliers. However, *ELD* is the only score that achieves the perfect detection across all three synthetic datasets.

Propositionalization-Based Methods vs. *ELD* Table 5.7 shows the precision values for propositional-based methods compared to *ELD*. *Our ELD score outperforms all Propositional-based methods on all datasets*, except for one real-world dataset that *LOF* outperform *ELD* and that is with the help of conjunctive features generated by the propositionalization method introduced in Chapter 4. If we use aggregated features the performance of *LOF* drops to 0.53. In general, if we use aggregated features, as it was used in the literature to generate features for the baseline methods, the performances of propositional-based methods are most like that of the probabilistic score *FD*, which does not consider the correlation among the features.

Case Studies

For a case study, we examine three top outliers as ranked by *ELD*, shown in Table 5.8. 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 f_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.

Strikers vs. Goalies *ELD* separates goalies from Strikers better compared to the other methods. 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 his 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

$$\mathit{ShotEff} = \mathit{high}, \mathit{TackleEff} = \mathit{medium} \rightarrow \mathit{DribbleEff} = \mathit{low}$$

is 50%, whereas in the general striker class it is 38%. The *ELD* divergence also ranks Edin Dzeko as unusual. But because it allows feature and joint information divergence to cancel, his rank is somewhat lower. The likelihood metric does not recognize him as unusual at all.

The next two outliers according to *ELD* are goalies Paul Robinson and Michel Vorm. Their rank is based only on feature divergence, with zero mutual information distinction. The maximum feature divergence is obtained by the *SavesMade* feature. This makes intuitive sense since strikers basically never make saves. In other words, feature divergence with respect to *SavesMade* is a good way to distinguish goalies from strikers.

The *ELD* divergence also ranks Paul Robinson and Michel Vorm as clear goalies. The likelihood metric does not recognize Paul Robinson as unusual at all.

Midfielders vs. Strikers The *ELD* metric separates midfielders from strikers better compared than the other methods. The single feature divergence does not discriminate these two classes of objects. Intuitively, this is because strikers and midfielders are generally similar with respect to single features. The distance metrics have a better TOR rate than the averaging metrics.

The decomposition analysis for the top three *ELD* outliers proceeds as follows. 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

$$\mathit{ShotEff} = \mathit{high}, \mathit{TimePlayed} = \mathit{high} \rightarrow \mathit{ShotsOnTarget} = \mathit{high}$$

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

Wayne Rooney is recognized as a striker for similar reasons, but less clearly because he achieves a high number of shots on target less frequently. 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%.

Drama vs. Comedy As with the other datasets, the *ELD* metric separates normal objects from the contrast class better than the other methods. 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.

5.7 Conclusion

In this chapter, we presented a new approach for applying Bayes nets to object-relational outlier detection. 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 except for one dataset. The alternative of

Table 5.8: 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
Edin Dzeko	Striker	1	DribbleEfficiency	83.84	DE=low
Paul Robinson	Goalie	2	SavesMade	49.4	SM=Medium4
Michel Vorm	Goalie	3	SavesMade	85.9	SM=Medium
Midfielders (Normal) vs. Strikers (Outlier)					
PlayerName	Position	*ELD* Rank	*ELD* Max Node	*ELD* Node Score	*FD* Max feature Value
Robin Van Persie	Striker	1	ShotsOnTarget	153.18	ST=high
Wayne Rooney	Striker	2	ShotsOnTarget	113.14	ST=high
Scott Sinclair	Midfielder	6	DribbleEfficiency	71.9	DE=high
Drama (Normal) vs. Comedy (Outlier)					
MovieTitle	Genre	*ELD* Rank	*ELD* Max Node	*ELD* Node Score	*FD* Max feature Value
Brave Heart	Drama	1	ActorQuality	89995.4	a_quality=4
Austin Powers	Comedy	2	Cast_Position	61021.28	Cast_Num=3
Blue Brothers	Comedy	3	Cast_Position	24432.21	Cast_num=3

converting the structured data to a flat data matrix via Propositionalization had a negative impact. Case studies showed that the log-distance score leads to easily interpreted rankings. Overall, our new log-likelihood distance metric provides a promising new approach for applying machine learning techniques to outlier detection for object-relational data, a challenging and practically important topic.

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.

Chapter 6

Success and Outlierness

In chapter 5 we introduced the *ELD* metric that quantifies the extent to which the individual association pattern of a potential outlier deviates from that of the whole population. The aim of this chapter is to compare the *ELD* metric with other meaningful metrics for comparing individuals. The goal is to use the *ELD* to estimate the value of the individuals and rank them. An empirical evaluation on soccer and movie data shows a strong correlation between the *ELD* score and success metrics: individuals that our metric identifies as unusual tend to have unusual success.

6.1 Introduction

The appearance of professional soccer statistics websites has made it possible to extend statistical studies to the sports domain. One of the interesting problems in this domain is predicting success and providing true estimates of players' abilities. An intuitive way to estimate the value of the players is to manually aggregate information about features of individuals over time and then rank them based on their performance in those features. For example, we can compare players based on the total number of goals they have scored or the average of their shot efficiency. However, this comparison may be unfair to most players because not all players are in the position to shoot or score a goal (e.g. goalies or defenders). One may argue that defenders (or goalies) have some other characteristics that are a lot stronger in their group compared to other groups. However, detecting important and distinctive features of each group of individuals requires domain knowledge and it is not often an easy task.

Another disadvantage of ranking based on manual aggregation of features is that it causes loss of information.

In chapter 4 and 5 we showed the advantage of using a generative model is to learn complex, and at the same time, informative features for individuals. In this chapter we propose a method to rank individuals which is based on *ELD* metric introduced in chapter 5. We compare the *ELD* metric to other metrics of success for a given domain. Our reasoning is that high success is an independent metric that indicates an unusual individual. Therefore, a correlation between log-likelihood distance and success is an independent validation of the log likelihood distance and shows that it points to meaningful and interesting outliers.

Approach Individuals are grouped into categories. A class-model Bayesian network (BN) structure is learned with data for the entire category of the individuals. An individual-model Bayesian network structured is learned from the individual data. *ELD* metric is computed based on the approach introduced in chapter 5 and is used to rank the individuals.

Evaluation We analyze two real-world data sets, from the UK Premier League and the Internet Movie Database (IMDb). Success metrics, such as the player's salary, provide an independent score for comparison with the ELD score. The empirical distributions of the ELD metric show a strong correlation with independent success metrics.

6.2 Preliminary Analysis

Market value of a player is not solely based on the player's performance and is often influenced by some other factors, such as the player's age and nationality.
In this section we study a few of these factors that are known to affect the market value of soccer players in the literature. We manually collected salary, nationality and age of 120 players of the Premier League in order to investigate the effect of each of these factors on the success of players.

Fact #1: Some teams tend to pay more: Table 6.1 shows the average salaries of players of different teams in the Premier League. Some teams have much larger budgets and are able to pay higher wages compared to less wealthy teams. A player in Manchester United may not necessarily be performing substantially better than a player in the same position in Tottenham, while there is a substantial difference between the average salary of players in Tottenham and Manchester United. For this reason, we normalize the salaries of the player in order to decrease this effect.

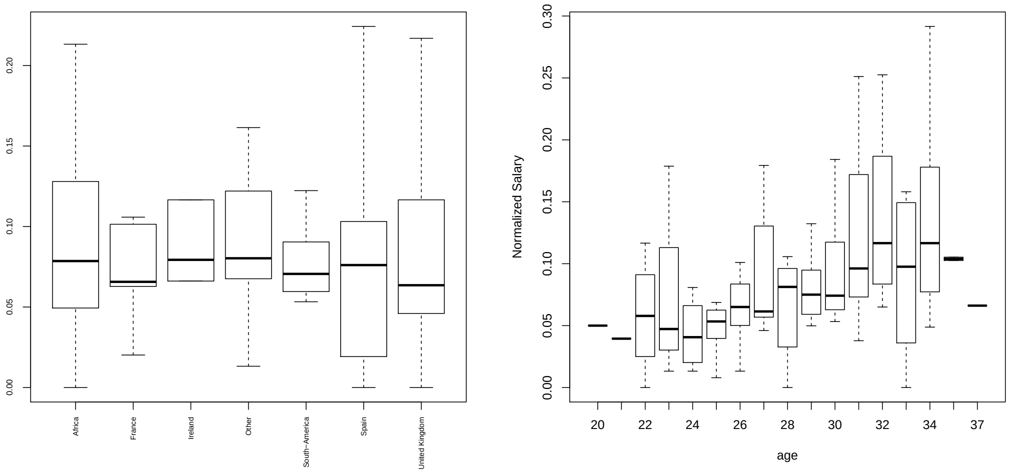
Fact #2 Player's nationality has little effect on player's salary: Previous research on soccer shows that the nationality of a player sometimes affects his salary, regardless of his performance [94].
"The phrase 'Brazilian soccer player' is like the phrases 'French chef' or 'Tibetan monk.' The nationality expresses an authority, an innate vocation for the job, whatever the natural ability." [62]
We investigated this phenomenon in our domain and showed that it has very little effect on the players market value. Figure 6.1(a) shows the distribution of salaries of the players across different nationalities.

Team	μ(Salary of the players of PL teams in €)
Arsenal	82307.69
Aston Villa	23625.00
Chelsea	88500.00
Everton	44090.91
Fullham	30372.57
Liverpool	66666.67
Manchester City	111076.90
Manchester United	81384.62
New Castle	41000.00
Sunderland	31857.14
Tottenham	65428.57

Table 6.1: Average salary of players in different teams .

Fact #3 Player's salary increases as age increases: Older players, from ages 30 to 33, tend to earn higher wages compared to other age groups because it takes time to accumulate fame and experience. A famous older player and a young player may play equally well, but the famous player may have a higher salary due to his reputation.
For each player, salary distribution has a different peak but it is between 30-33 for most players [90]. Figure 3.7 shows the salary of the players in different age groups.

Based on the discussion above, we know that a method that is solely based on players' performance will not be 100% successful in ranking the players.



(a) Salaries of players of different nationalities. (b) Salaries of players of different age group.

Figure 6.1: Salary comparison of Premier League Players

6.3 Related Work

6.3.1 Analyzing sports data

Analyzing sports data can make a significant difference in scoring players, signing contracts and preventing injuries. Pei *et al.*[66] propose a reference-based method that uses relative degree of density with respect to a fixed set of reference points to calculate the neighbourhood density of a data point. They aim to find outstanding players based on two test settings: 1) The total number of games played, goals scored and shooting percentages. 2) Points scored, plus/minus statistics and penalty minus.

Schwartz [90] *et al.* focused on the valuation of draft order in the SuperDraft. The valuation of draft order was first introduced in the National Football League and proved very useful to the coach in trading players. They first estimate career trajectories of players and then assess the value of the draft position by introducing some performance measure. They used time played and salary of the player as ground truth in order to validate their method.

6.3.2 Ranking system in Sports domain

Ranking individuals is a useful task for many applications in Information Retrieval, Natural Language Processing and Data mining. In sports, performance is usually interpreted as a rating system or ranking system. Ranking players is especially important in this domain because teams with lower budgets are usually looking for ways to detect undervalued players to be able to compete with wealthier teams for lower costs. Lewis *et. al* [52] used a quantitative analysis to evaluate the value of baseball players.

In the individual sports (e.g. tennis), ranking is straightforward and can be driven from the results of past tournaments, as it has been done for years by the Association of Tennis Professionals (ATP). However, this simple framework for ranking has been questioned and claimed to perform poorly in predicting the results of future games [58]. Other sports association, such as soccer and cricket, also have official rankings of teams and players, which is the basis of many important decisions. For example, FIFA's world ranking plays an important part in awarding work permits to players outside the European Union in the Premier League [59]. The problem of ranking teams and players is not trivial and the need for a better analytical system should not be understated. Although the world ranking performs poorly in predicting match outcomes, it has been used to determine qualifications for tournaments [57].

In team sports, rating individuals is a more complex task due to team structure; players have different positions. Keri *et. al* have developed a rating system for each speciality in baseball [39].

The analysis becomes more complicated when the goal is to compare players with different specialities. [84] investigates the metrics that attempt such an analysis in baseball and value players regardless of their position. McHale [59] *et. al* developed an index to rate players regardless of their playing speciality, based on their contributions to wining performances.

6.4 Correlation with Success

The aim of this section is to compare the *ELD* metric with other meaningful metrics for comparing individuals. Our reference metrics are success rankings of individuals selected for a specific domain, shown in table 6.2. We use the same data as in our other experiments, as described in Chapter 3.

Success rankings are one of the most interesting features to users. Strong correlations between the *ELD* metric and meaningful success metrics provide evidence that the *ELD* metric is also meaningful. We measure correlation strength by the standard correlation coefficient ρ. The coefficient ranges from -1 to 1, where 0 means no correlation and 1 or -1 indicates maximum strength [26].

The observed correlations are remarkable in at least two respects: 1) The strength of the correlation between the *ELD* metric and the success ranking are high; coefficients range from 0.45 to 0.82. 2) We observe this phenomenon across different domains, different types of individuals and different success metrics.

Dataset	Success Metric	Min	Max	Standard Dev.	Mean
IMDb	Sum of Rating	1	14795	1600.22	1057.58
Soccer-Player	TimePlayed	5.0	3420	1015.69	1484.0
Soccer-Player	Normalized Salary	0.007	0.28	0.620	0.100
Soccer-Player	Sum of Shot Efficiency	0	82	9.87	6.53
Soccer-Team	Standing	1.0	20	5.91	10.5

Table 6.2: Success metrics and their distributions.

For a population with a diverse set of skills and resources, being different from the generic class can be interpreted as both exceptionally better or worse than normal population. In the domains we study in this data, we found that higher *ELD* scores indicate exceptionally good individuals. Our interpretation of this positive correlation between *ELD* and success is that our domains featured skilled individuals, such that the average is already quite successful. For example, in the Premier League we expect most players to be in the range of good players. Therefore, deviating from the rest of the population is a signal for detecting exceptionally good players. Our *ELD*-success scatterplots below provide empirical evidence for this interpretation; we typically see a large cluster of individuals around the origin, meaning that their success is normal and their *ELD* score is low.

6.4.1 Methodology

We report the correlations between the *ELD* metric and metrics of success for a specific domain. We also focus on some unusually successful individuals as case studies. In considering the correlation between *ELD* and success, it is useful to investigate subgroups of individuals to ensure an apples-to-apples comparison [87]. For instance, the attributes that lead to success are different for strikers and goalies. Accordingly, we report correlations for

Table 6.3: Correlation between *ELD* metric and standing of Teams. The best standing is place 1.

Team	Standing
Top Teams	-0.71
Bottom Team	-0.33
All Team	-0.20

Table 6.4: Correlation between *ELD* metric and success metrics of Players.

Class	TimePlayed	Salary	SavesMade	ShotsOntarget	Passeff
Strikers	0.76	0.79	NA	0.72	NA
Midfielders	0.73	0.45	NA	NA	0.89
Goalies	0.69	NA	0.71	NA	NA
All players	0.81	0.56	NA	NA	NA

subgroups as well as entire classes of individuals.

6.4.2 Correlations between the *ELD* outlier metric and success

The next three tables summarize the observed correlations between success and *ELD* metrics: Teams in Table 6.3, Players in Table 6.4, Movies in Table 6.5.

Teams

Team Standing The most successful team has Standing=1 and the least successful team has Standing=20 in the 2011-2012 Season. For the top teams, a very strong negative correlation emerges between *ELD* and standing: teams with higher *ELD* achieve a better (lower) standing.

Figure 6.2 shows the correlation of *ELD* with team success metrics in a scatter plot.

Table 6.5: Correlation between *ELD* metric and success metric of Movies.

Genre	Sum of Rating	Average of Rating	Number of Rating
Action	0.68	0.30	0.72
Drama	0.76	0.32	0.77
Comedy	0.85	0.41	0.84
All Movies	0.56	0.17	0.60

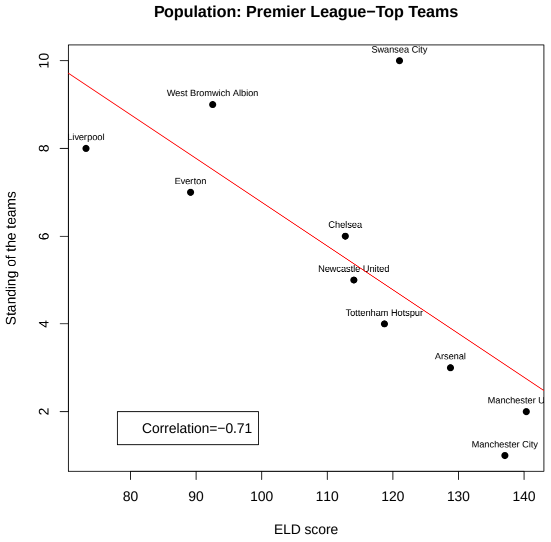


Figure 6.2: Teams: Team Standing vs. *ELD* for the top teams in the Premier League.

The top two teams, Manchester City and Manchester United, stand out strongly in terms of the *ELD* metric (bottom right corner).

Players

Players Time Played is the total time that a player played over all matches in the season. This metric was shown to correlate strongly with other success metrics, such as salary, on MLS soccer data [90]. For each subgroup there is a strong positive correlation with *ELD*, meaning that atypical players with higher *ELD* tend to play more minutes.

Salary is probably the most obvious, and at the same time often the most misleading way to measure success of the players. Previous studies suggest that salary of the players does not always follow their performance in many sports, such as baseball and soccer [32, 19]. They show that pay cannot be explained only by past performance and there are other factors that are hard to quantify and have a great effect on the salaries.

We manually collected the salaries of 120 players that we could find on-line. Table 6.4 and Figures 6.3 and 6.4 show the correlation between *ELD* and this success metric. The correlation is high, especially for Strikers. We discuss the relatively weaker salary correlation for midfielders in more detail below.

Shots on Target applies to strikers only. This is defined as any shot attempt that would, or does, enter the goal if left unblocked. We record the total number of these shots over all matches of the strikers only. This metric was shown to correlate strongly with *ELD* (see Table 6.4, Figure 6.3(b)).

Figure 6.3 plots *ELD* against striker success metrics. We observe a large cluster around the origin, which points to a large base of normal strikers with both salaries and low *ELD* scores.

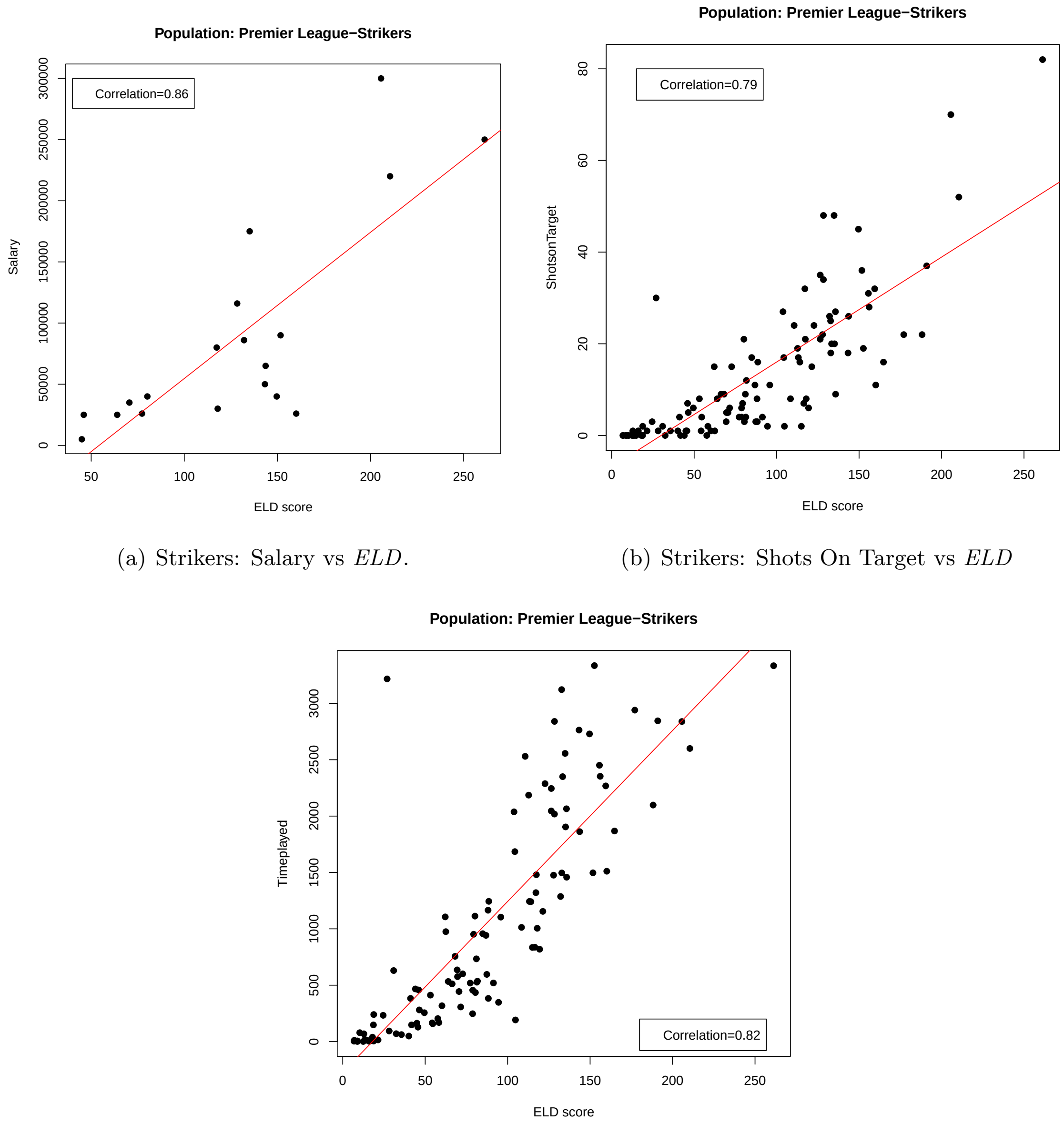
Saves Made applies to goalies only; it is defined as the total number of saves that goalies made over all the matches. This metric shows a strong correlation with *ELD* as well (see Table 6.4, Figure 6.4(b)).

Figure 6.4 shows the correlation of *ELD* with goalie success metrics in a scatter plot. Goalies do not vary much in terms of the time they play. Wayne Hennessey has the highest number of Saves Made and also an unusually high *ELD* score, although not the highest.

Midfielder Salary We omit a scatterplot for midfielder salary vs. *ELD* because it is less informative due to the weaker correlation (0.45). To investigate the reason for the weaker correlation, we chose two midfielders: 1) Stephane Sessegnon who has been ranked second in the *ELD* ranking but does not draw a large salary. 2) Steven Gerrard is a very well known player and ranked second in the Salary ranking, but according to the *ELD* score, he has been ranked 21. Based on domain knowledge, we chose some of the features from the raw data that are relevant to midfield performance and compared the feature statistics for these two players. Table 6.6 shows the details of their appearances in different matches. Sessegnon scored higher than Gerrard in three out of the four categories (Passes and Time Played). However, his salary was much lower than Gerrard's. This is an example of how weak the correlation is between salary and the observed box scores, which is the basis for the *ELD* metric.

Name	Team	age	Salary Ranking	*ELD* Ranking	Time Played	Unsuccessful Passes	Successful Long Passes	Successful corners
Steven Gerrard	Liverpool	31	2	21	1212 min	244	52	25
Stephane Sessegnon	Sunderland	26	22	2	3133 min	231	82	15

Table 6.6: Comparison of two midfielders.

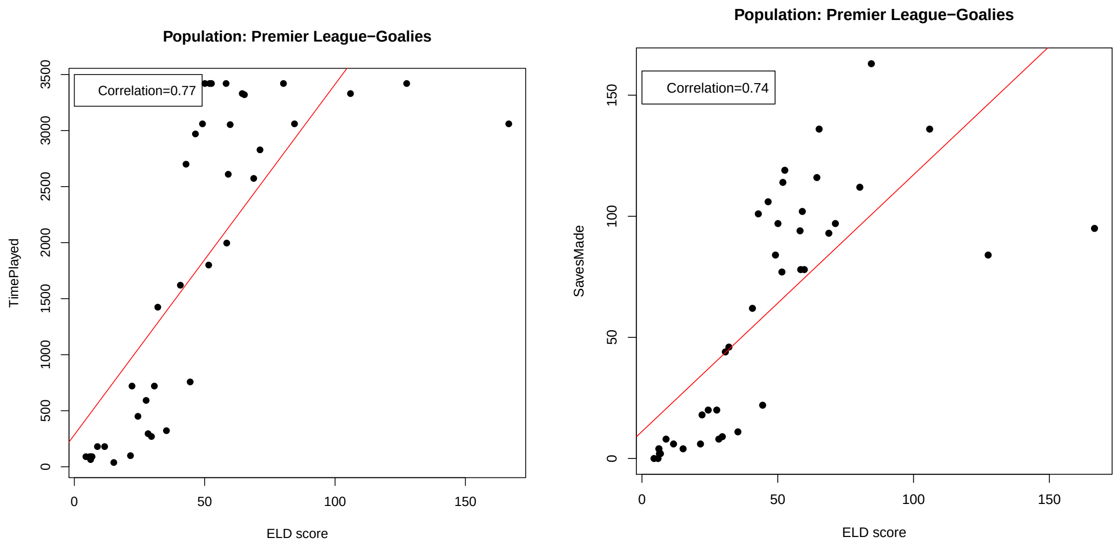


(a) Strikers: Salary vs *ELD*.

(b) Strikers: Shots On Target vs *ELD*

(c) Strikers: Time played vs *ELD*.

Figure 6.3: Correlations in the strikers population



(a) Goalies: sum of time played vs *ELD*.

(b) Goalies: sum of saves made vs *ELD*

Figure 6.4: Correlations in the goalies population

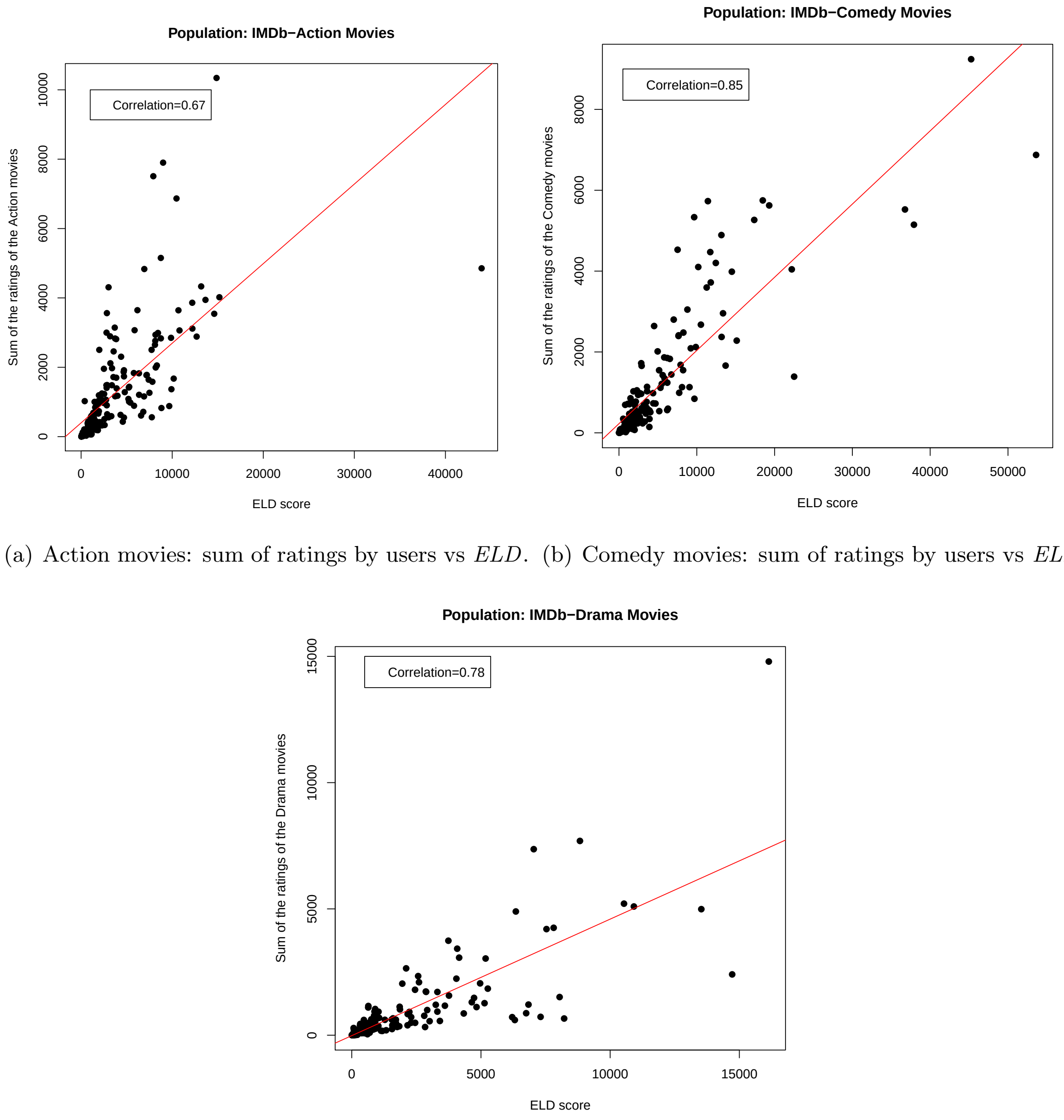
Movies

Movie Sum of Ratings is the number of user ratings of a movie. Table 6.5 shows a high correlation with the *ELD* metric. The highest correlation obtained is for the Comedy genre (0.84). The correlation between a movie and the sum of its ratings is equally strong, but the correlation with its average rating is much weaker. Thus, the *ELD* score is mainly related to how many users have rated the movie rather than with how they have rated it. The number of ratings is a meaningful success metric as it indicates the number of people who have gone to see a movie.

Figure 6.5 shows the correlation of *ELD* with movie success metrics in a scatter plot. We again observe a large cluster of movies around the origin. For drama and comedy movies, the top rated movies are ("American Beauty" resp. "Being John Malkovich"); these also stand out in the *ELD* metric.

6.5 Conclusion

In this chapter we used the *ELD* metric, that was introduced in Chapter 5, to rank the individuals. We compared the *ELD* metric to other metrics of success for a given domain.



(a) Action movies: sum of ratings by users vs *ELD*.

(b) Comedy movies: sum of ratings by users vs *ELD*

(c) Drama movies: sum of ratings by users vs *ELD*.

Figure 6.5: Correlation in the movies population

In our experimental results we showed that the *ELD* metric correlates with success metrics, to a surprising degree, across different domains and classes of individuals. Since high success is an independent metric that indicates an unusual individuals, this correlation shows that *ELD* marks meaningful and interesting outliers.

We investigated some independent factors that affect the success of individuals in the Premier League. We showed that there are factors other than the performance of players that affect their ranking and for this reason the methods that are solely based on evaluation of the performance of players can never achieve 100% accuracy in predicting the ranking.

Chapter 7

Summary and Conclusion

Outlier detection is an important task in data mining and has many applications in areas such as health care, security and finance. While many outlier analysis techniques have been developed for i.i.d. propositional data, there are not many methods designed for structured data. In this dissertation, we developed two model-based outlier detection methods for object relational data model.

In Chapter 4 we developed a pipeline propositionalization approach where the information from multiple data tables is summarized in a single data table. We utilized Markov Logic Network learning for this task. In an empirical comparison with the baseline wordification approach of enumerating all conjunctive formulas up to length 2, Markov Logic propositionalization showed several advantages: 1) The set of formulas learned was substantially smaller, leading to smaller data tables and faster outlier detection. 2) The formulas learned were longer, representing more complex relational patterns. 3) For a fixed single-table outlier analysis method, the average detection accuracy was higher.

In Chapter 5 we presented a new approach for applying Bayes nets to object-relational outlier detection. 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 Chapter 6 we used the *ELD* metric, that was introduced in Chapter 5, to rank the individuals. We compared the *ELD* metric to other metrics of success for a given domain. In our experimental results we showed that the *ELD* metric correlates to a surprising degree with success metrics across different domains and classes of individuals. Since high success is an independent metric that indicates an unusual individual, this correlation shows that *ELD* marks meaningful and interesting outliers.

Bibliography

[1] Elke Achtert, Christian Bohm, Hans-Peter Kriegel, Peer Kroger, Ina Muller-Gorman, and Arthur Zimek. Detection and visualization of subspace cluster hierarchies. In *DASFAA*. Springer, 2007.

[2] Elke Achtert, Hans-Peter Kriegel, Erich Schubert, and Arthur Zimek. Interactive data mining with 3d-parallel coordinate trees. In *Proceedings of the 2013 ACM SIGMOD*, New York, NY, USA, 2013.

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

[4] R. Agrawal and R. Srikant. Fast algorithms for mining association rules in large databases. In *Proc International Conference on Very Large Databases*, pages 478–499, Santiage, Chile, 1994. Morgan Kaufmann, Los Altos, CA.

[5] Leman Akoglu, Mary McGlohon, and Christos Faloutsos. Oddball: Spotting anomalies in weighted graphs. In *Proceedings of the 14th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining - Volume Part II*, PAKDD'10, pages 410–421, Berlin, Heidelberg, 2010. Springer-Verlag.

[6] Leman Akoglu, Hanghang Tong, and Danai Koutra. Graph based anomaly detection and description: A survey. *Data Min. Knowl. Discov.*, 29(3), May 2015.

[7] Waleed Alsanie and James Cussens. Learning recursive prism programs with observed outcomes. In *Proceedings of the ICML-2012 Workshop on SRL*, 2012.

[8] Grant Anderson and Bernhard Pfahringer. Clustering relational data based on randomized propositionalization. In *Inductive Logic Programming, 17th International Conference, ILP 2007, Corvallis, OR, USA, June 19-21, 2007, Revised Selected Papers*, pages 39–48, 2007.

[9] Fabrizio Angiulli and Fabio Fassetti. Outlier detection using inductive logic programming. In Wei Wang 0010, Hillol Kargupta, Sanjay Ranka, Philip S. Yu, and Xindong Wu, editors, *ICDM*, pages 693–698. IEEE Computer Society, 2009.

[10] Fabrizio Angiulli, Gianluigi Greco, and Luigi Palopoli. Outlier detection by logic programming. *ACM Trans. Comput. Logic*, 9, 2007.

[11] Sakshi Babbar and Sanjay Chawla. On bayesian network and outlier detection. In P. Sreenivasa Kumar, Srinivasan Parthasarathy, and Shantanu Godbole, editors, *COMAD*, page 125. Allied Publishers, 2010.

[12] Markus Breunig, Hans-Peter Kriegel, Raymond T. Ng, and Jrg Sander. Lof: Identifying density-based local outliers. In *Proceedings of ACM SIGMOD*, 2000.

[13] Antonio Cansado and Alvaro Soto. Unsupervised anomaly detection in large databases using bayesian networks. *Appl. Artif. Intell.*, 22(4):309–330, April 2008.

[14] Nitesh V. Chawla, Nathalie Japkowicz, and Aleksander Kotcz. Editorial: Special issue on learning from imbalanced data sets. *SIGKDD Explor. Newsl.*, 6(1):1–6, June 2004.

[15] Sanjay Chawla and Aristides Gionis. k-means-: A unified approach to clustering and outlier detection. In *SDM*, pages 189–197. SIAM, 2013.

[16] Michael Chiang and David Poole. Reference classes and relational learning. *Journal of Approximation Reasoning*, 53(3), 2012.

[17] Manuel Davy and Simon J. Godsill. Detection of abrupt spectral changes using support vector machines an application to audio signal segmentation. In *ICASSP*, pages 1313–1316. IEEE, 2002.

[18] L. de Campos. A scoring function for learning bayesian networks based on mutual info. and cond. indep. tests. *JMLR*, pages 2149–2187, 2006.

[19] Pedro Garcia del Barrio and Francesc Pujol. Pay and Performance in the Spanish Soccer League: Who Gets the Expected Monopsony Rents? Faculty Working Papers 05/04, School of Economics and Business Administration, University of Navarra, March 2004.

[20] Pedro Domingos and Daniel Lowd. *Markov Logic: An Interface Layer for Artificial Intelligence.* Morgan and Claypool Publishers, 2009.

[21] Pedro Domingos and Matthew Richardson. Markov logic: A unifying framework for statistical relational learning. In *Introduction to Statistical Relational Learning* [29].

[22] Karanjit Singh. Dr. Shuchita Upadhyaya. Anomaly pattern detection in categorical datasets. In *Proceedings of the International Journal of Computer Trends and Technology (*, 2012.

[23] Charles Elkan. The foundations of cost-sensitive learning. In *In Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence*, pages 973–978, 2001.

[24] Oliver Schulte Fatemeh Riahi. Propositionalization for unsupervised outlier detection in multi-relational data. In *The Florida Association of Artificial Intelligence*, 2016.

[25] Tom Fawcett. An introduction to ROC analysis. *Pattern Recogn. Lett.*, 2006.

[26] R. A. Fisher. On the probable error of a coefficient of correlation deduced from a small sample. *Metron*, 1:3–32, 1921.

[27] Jing Gao, Feng Liang, Wei Fan, Chi Wang, Yizhou Sun, and Jiawei Han. On community outliers and their efficient detection in information networks. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '10, pages 813–822, New York, NY, USA, 2010. ACM.

[28] Jing Gao and Pang-Ning Tan. Converting output scores from outlier detection algorithms into probability estimates. In *ICDM*, pages 212–221. IEEE Computer Society, 2006.

[29] Lise Getoor and Ben Tasker. *Introduction to statistical relational learning.* MIT Press, 2007.

[30] Anup K. Ghosh, James Wanken, and Frank Charron. Detecting anomalous and unknown intrusions against programs. In *In Proceedings of the Annual Computer Security Application Conference (ACSAC98*, pages 259–267, 1998.

[31] F. E. Grubbs. Procedures for detecting outlying observations in samples. *Technometrics*, 11:1–21, 1969.

[32] Stephen Hall, Stefan Szymanski, and Andrew S. Zimbalist. Testing causality between team performance and payroll: The cases of major league baseball and english soccer. *Journal of Sports Economics*, 3(2):149–168, 2002.

[33] D. M. Hawkins. *Identification of Outliers.* Chapman and Hall, London [u.a.], 1980.

[34] Alexander Hinneburg, Charu C. Aggarwal, and Daniel A. Keim. What is the nearest neighbor in high dimensional spaces? In *Proceedings of the 26th International Conference on Very Large Data Bases*, VLDB '00, pages 506–515, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc.

[35] Victoria Hodge and Jim Austin. A survey of outlier detection methodologies. *Artif. Intell. Rev.*, 22(2):85–126, October 2004.

[36] Farhad Hussain, Huan Liu, Einoshin Suzuki, and Hongjun Lu. Exception rule mining with a relative interestingness measure. In *In Proceedings of Paci Asia Conference on Knowledge Discovery in DataBases (PAKDD-2000*, pages 86–97, 2000.

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

[38] Wen Jin, Anthony K. H. Tung, Jiawei Han, and Wang Wei. Ranking outliers using symmetric neighborhood relationship. In *Springer*, volume 3918 of *Lecture Notes in Computer Science*, pages 577–593, 2006.

[39] James Click Jonah Keri. *Baseball Between the Numbers.* Basic Book, 2006.

[40] Kristian Kersting and Luc De Raedt. Bayesian logic programming: Theory and tool. In *Introduction to Statistical Relational Learning* [29], chapter 10, pages 291–318.

[41] Hassan Khosravi, Oliver Schulte, Tong Man, Xiaoyuan Xu, and Bahareh Bina. Structure learning for Markov logic networks with many descriptive attributes. In *AAAI*, 2010.

[42] Tushar Khot, Sriraam Natarajan, Kristian Kersting, and Jude W. Shavlik. Learning Markov logic networks via functional gradient boosting. In *ICDM*. Computer Society, 2011.

[43] Angelika Kimmig, Lilyana Mihalkova, and Lise Getoor. Lifted graphical models: a survey. *Computing Research Repository*, 2014.

[44] Judice LY Koh, Mong Li Lee, Wynne Hsu, and Wee Tiong Ang. Correlation-based attribute outlier detection in xml. In *ICDE 2008. IEEE 24th*, 2008.

[45] Yun Sing Koh and Nathan Rountree. Finding sporadic rules using apriori-inverse. In *Proceedings of the 9th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining*, PAKDD'05, pages 97–106, Berlin, Heidelberg, 2005. Springer-Verlag.

[46] Stefan Kramer, Nada Lavravc, and Peter Flach. Propositionalization approaches to relational data mining. pages 262–286, 2000.

[47] Janez Kranjc, Vid Podpecan, and Nada Lavrac. Clowdflows: A cloud based scientific workflow platform. In *ECML/PKDD*, Lecture Notes in CS. Springer, 2012.

[48] Ondvrej Kuvzelka and Filip Zelezny. Hifi: Tractable propositionalization through hierarchical feature construction. In *Late Breaking Papers, ILP*, 2008.

[49] J. Pei J. Bailey G. Dong A. Campbell L. Duan, G. Tang and C. Tang. Mining contrast subspaces. *Proceedings of the 18th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD14).*

[50] Jorma Laurikkala, Martti Juhola, and Erna Kentala. Informal identification of outliers in medical data, 2000.

[51] Nada Lavravc, Matic Perovvsek, and Anvze Vavpetivc. Propositionalization online. In *ECML*, pages 456–459. Springer, 2014.

[52] David Lewis. (spam vs.) forty years of machine learning for text classification. In *Proceedings of the Spam Conference*, 2003. Available: `http://www.daviddlewis.com/publications/slides/lewis-2003-0117-spamconf-slides.pdf`.

[53] Marco Lippi, Manfred Jaeger, Paolo Frasconi, and Andrea Passerini. Relational information gain. *Machine Learning*, 83(2):219–239, 2011.

[54] Joris Maervoet, Celine Vens, Greet Vanden Berghe, Hendrik Blockeel, and Patrick De Causmaecker. Outlier detection in relational data: A case study in geographical information systems. *Expert Syst. Appl.*, 39(5):4718–4728, April 2012.

[55] Matthew V. Mahoney and Philip K. Chan. Learning models of network traffic for detecting novel attacks.

[56] MCFC Analytics. The premier league dataset, 2012. URL = `http://www.mcfc.co.uk/Home/The%20Club/MCFC%20Analytics`.

[57] Ian McHale and Alex Morton. *Statistical analysis of the FIFA world rankings.* Statistical Thinking in Sport, Chapman and Hall, 2007.

[58] Ian McHale and Alex Morton. A bradley-terry type model for forecasting tennis match results. *International Journal of Forecasting*, 27(2):619–630, 2011.

[59] Ian G. McHale, Philip A. Scarf, and David E. Folker. On the development of a soccer player performance rating system for the english premier league. *Interfaces*, 42(4):339–351, July 2012.

[60] Emmanuel Muller, Ira Assent, Patricia Iglesias, Yvonne Mulle, and Klemens Bohm. Outlier ranking via subspace analysis in multiple views of the data. In *IEEE ICDM*, 2012.

[61] Emmanuel Müller, Patricia Iglesias Sánchez, Yvonne Mülle, and Klemens Böhm. Ranking outlier nodes in subspaces of attributed graphs. In *Workshops Proceedings of the 29th IEEE International Conference on Data Engineering, ICDE 2013, Brisbane, Australia, April 8-12, 2013*, pages 216–222, 2013.

[62] PAPASTERGIADIS Nikos. *Futebol and Myths of the Brazilian Way of Life.* Cultural Studies Review, 2013.

[63] Petra Kralj Novak, Geoffrey I. Webb, and Stefan Wrobel. Supervised descriptive rule discovery: A unifying survey of contrast set, emerging pattern and subgroup mining. *Journal of Machine Learning Research*, 2009.

[64] Spiros Papadimitriou, Hiroyuki Kitagawa, Phillip B. Gibbons, and Christos Faloutsos. Loci: Fast outlier detection using the local correlation integral. In *ICDE*, pages 315–326. IEEE Computer Society, 2003.

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

[66] Yaling Pei, Osmar R. Zaiane, and Yong Gao. An efficient reference-based approach to outlier detection in large datasets. In *Proceedings of the Sixth International Conference on Data Mining*, ICDM '06, pages 478–487, Washington, DC, USA, 2006. IEEE Computer Society.

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

[68] Matic Perovsek, Anze Vavpetic, Bojan Cestnik, and Nada Lavrac. A wordification approach to relational data mining. In *Discovery Science*, Lecture Notes in Computer Science. Springer, 2013.

[69] James III Pickands. Statistical inference using extreme order statistics. *The Annals of Statistics*, (1):119–131, 1975.

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

[71] Sridhar Ramaswamy, Rajeev Rastogi, and Kyuseok Shim. Efficient algorithms for mining outliers from large data sets. In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, SIGMOD '00, pages 427–438, New York, NY, USA, 2000. ACM.

[72] G Ratsch, S Mika, B Scholkopf, and K.-R Muller. Constructing boosting algorithms from svms: an application to one-class classification. *IEEE PAMI*, 2002. In press. Earlier version is GMD TechReport No. 119, 2000.

[73] Irma Ravkic, Jan Ramon, and Jesse Davis. Learning relational dependency networks in hybrid domains. *Machine Learning*, 2015.

[74] Fatemeh Riahi and Oliver Schulte. Codes and Datasets. [Online]. Available:. `ftp://ftp.fas.sfu.ca/pub/cs/oschulte/CodesAndDatasets/`, 2015.

[75] Fatemeh Riahi and Oliver Schulte. Model-based outlier detection for object-relational data. In *Computational Intelligence, 2015 IEEE Symposium Series on*, pages 1590–1598. IEEE, 2015.

[76] Sebastian Riedel, Limin Yao, Andrew McCallum, and Benjamin M. Marlin. Relation extraction with matrix factorization and universal schemas. In *Human Language Technologies: NAA*, pages 74–84, 2013.

[77] Volker Roth. Outlier detection with one-class kernel fisher discriminants. In *Advances in Neural Information Processing Systems 17*, pages 1169–1176. MIT Press, 2005.

[78] Sunita Sarawagi, Rakesh Agrawal, and Nimrod Megiddo. Discovery-driven exploration of OLAP data cubes. In *In Proc. Int. Conf. of Extending Database Technology (EDBT'98*, pages 168–182. Springer-Verlag, 1998.

[79] Bernhard Schölkopf, John C. Platt, John C. Shawe-Taylor, Alex J. Smola, and Robert C. Williamson. Estimating the support of a high-dimensional distribution. *Neural Comput.*, 13(7), July 2001.

[80] Oliver Schulte. A tractable pseudo-likelihood function for Bayes nets applied to relational data. In *SIAM SDM*, 2011.

[81] Oliver Schulte and Hassan Khosravi. Learning graphical models for relational data via lattice search. *Journal of Machine Learning*, 2012.

[82] Tobias Sing, Oliver Sander, Niko Beerenwinkel, and Thomas Lengauer. *ROCR: Visualizing the performance of scoring classifiers.*, 2012. R package version 1.0-4.

[83] Claudio De Stefano, Carlo Sansone, and Mario Vento. To reject or not to reject: that is the question-an answer in case of neural classifiers. *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, pages 84–94, 2000.

[84] Christina Kahrl Steven Goldman. *Baseball Prospectus 2010: The Essential Guide to the 2010 Baseball Season.* John Wiley & Sons, Inc.,Hoboken, 2010.

[85] Jimeng Sun, Huiming Qu, Deepayan Chakrabarti, and Christos Faloutsos. Neighborhood formation and anomaly detection in bipartite graphs. In *Proceedings of the Fifth IEEE International Conference on Data Mining*, ICDM '05, pages 418–425, Washington, DC, USA, 2005. IEEE Computer Society.

[86] Y. Sun, J. Han, Peixiang Zhao, Zhijun Yin, Hong Cheng, and Tianyi Wu. Community distribution outlier detection in heterogeneous network. In *ECML*, 2013.

[87] Yizhou Sun, Han Jiawei, and Peixiang Zhao. Rankclus: Integrating clustering with ranking for heterogeneous information network analysis. In *Proceedings of the International Conference on Extending Database Technology: Advances in Database Technology*, pages 565–576, New York, NY, USA, 2009. ACM.

[88] Einoshin Suzuki. Discovering interesting exception rules with rule pair. In *In J. Fuernkranz (Ed.), Proceedings of the ECML/PKDD Workshop on Advances in Inductive Rule Learning*, pages 163–178, 2004.

[89] B. Taskar, P. Abbeel, and D. Koller. Discriminative probabilistic models for relational data. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, 2002.

[90] Mohan Parameswaran T.Swartz, Adriano Arce. Assessing value of the draft positions in major league soccer's superdraft. *The Sport Journal*, 2013.

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

[92] Kilian Q. Weinberger and Lawrence K. Saul. Distance metric learning for large margin nearest neighbor classification. *J. Mach. Learn. Res.*, 10:207–244, June 2009.

[93] Gregory Cooper Michael Wagner Weng-Keen Wong, Andrew Moore. Bayesian network anomaly pattern detection for disease outbreaks. In Tom Fawcett and Nina Mishra, editors, *Proceedings of the Twentieth International Conference on Machine Learning*, pages 808–815, Menlo Park, California, 2003. AAAI Press.

[94] Jadrian James Wooten. Can ranking nationalities explain the salary discrepancies in major league soccer. *Social Science Research Network*, 2013.