CoCo: Coding Cost For Parameter-Free Outlier Detection

Christian Böhm
University of Munich
Munich, Germany
boehm@dbs.ifi.lmu.de

Katrin Haegler
University of Munich
Munich, Germany
katrin.haegler@med.uni-muenchen.de

Nikola S Müller
Max Planck Institute of Biochemistry
Martinsried, Germany
nimuell@biochem.mpg.de

Claudia Plant
Technische Universität München
Munich, Germany
plant@lrz.tum.de

ABSTRACT

How can we automatically spot all outstanding observations in a data set? This question arises in a large variety of applications, e.g. in economy, biology and medicine. Existing approaches to outlier detection suffer from one or more of the following drawbacks: The results of many methods strongly depend on suitable parameter settings being very difficult to estimate without background knowledge on the data, e.g. the minimum cluster size or the number of desired outliers. Many methods implicitly assume Gaussian or uniformly distributed data, and/or their result is difficult to interpret. To cope with these problems, we propose CoCo, a technique for parameter-free outlier detection. The basic idea of our technique relates outlier detection to data compression: Outliers are objects which can not be effectively compressed given the data set. To avoid the assumption of a certain data distribution, CoCo relies on a very general data model combining the Exponential Power Distribution with Independent Components. We define an intuitive outlier factor based on the principle of the Minimum Description Length together with an novel algorithm for outlier detection. An extensive experimental evaluation on synthetic and real world data demonstrates the benefits of our technique. Availability: The source code of CoCo and the data sets used in the experiments are available at: `http://www.dbs.ifi.lmu.de/Forschung/KDD/Boehm/CoCo`.

Categories and Subject Descriptors

H.2.8 [**Database applications**]: Data mining

General Terms

Algorithms, Design, Reliability

KDD'09, June 28–July 1, 2009, Paris, France.

Keywords

Outlier Detection, Coding Costs, Minimum Description Length, Data Compression

1. INTRODUCTION

Automatic outlier detection in large data sets is often equally or even more important than the detection of regularities. In various application fields like economy, biology, or medicine, the detection of extraordinary observations is of great interest. For example, the identification of criminal activities, such as credit card fraud, is crucial in electronic commerce applications [9]. In biology, an automatic detection of outstanding measurements or noise is critical for high-throughput data generated with e.g. mass spectrometry or gene expression analysis. The wide range of application fields also includes entertainment, sports, e.g. performance analysis of athletes, and many more.

Today, many data mining publications are in the field of clustering or outlier detection. The first field searches for regularities in a data set whereby the second identifies irregular data. Closer inspection of both fields reveals a strong relationship, whereby one goes barely without the other: On one hand, most clustering algorithms are confronted with outliers which deteriorate the cluster quality and/or destabilize the algorithm. Thus, the outliers need to be removed beforehand. On the other hand, outlier detection algorithms require a definition of the underlying cluster structure although clusters are not explicitly identified. Only if the cluster structure (of the regular data) is known, outliers can be identified without any doubt. Following the definition of Hawkins [5]:

> An outlier is an observation that deviates so much from other observations as to arouse suspicion that it was generated by a different mechanism.

To formalize this definition, the ordinary and potentially clustered points as well as the outliers need to be differentiated with respect to a well-defined *distinction criterion.* In existing outlier detection approaches, the distinction criterion is quantified by a metric distance function and parameter settings. The results are only meaningful if the distance function is well-characterized with respect to the object similarity and suitable parameter settings. However,

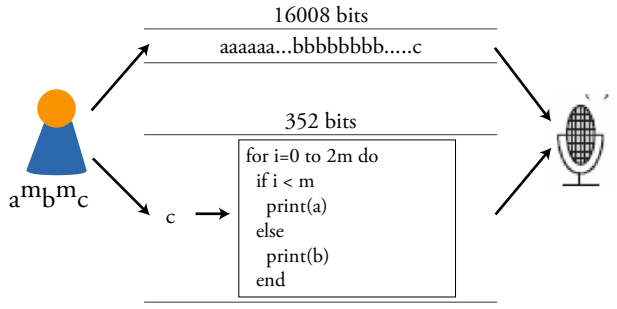

Figure 1: Data compression: The principle of MDL is to detect regularities in the data and compress accordingly.

these premises assume a prior characterization of the data set.

To cope with the problems of defining a distinct criterion and parametrization, we present CoCo, a parameter-free outlier detection method based on the ideas of data compression and coding costs. CoCo is able to identify the outliers in a data set based on a flexible definition of the *regular data*. The regular data is flexibly defined by a very general Probability Density Function (PDF), in our case a mixture model of the Exponential Power Distribution (EPD). The EPD is a family of distribution functions which contains the Gaussian distribution, the uniform distribution, the Laplacian distribution, and a great variety of other distribution functions. Compared to previous outlier detection approaches, the EPD is not restricted to either uniform or Gaussian distribution functions. We demonstrate with our experiments (cf. Section 4) that the EPD is powerful enough to model the regular data in a variety of applications.

CoCo considers a point P as outlier, unless it fits nicely in any of the distribution functions to be estimated of the points in the neighborhood of P, independent of the neighborhood size. To measure the quality of the fit of P we adopt the idea of data compression: If a point fits well into a distribution function, it can be compressed efficiently. To connect the data compression efficiency of P with the degree of P being an outlier, Figure 1 illustrates an intuitive example: Suppose, we want to transfer data via a communication channel. The sender wants to transfer the string $a^m b^m c$ to the receiver. A naive way would be to transfer each single character requiring in total 16,008 bits for $m = 1,000$ and 8 bits per character. To minimize the communication costs, a smart sender exploits regularities in the data. A little program could generate the first part of the string by printing 1,000 times the character a followed by 1,000 times b. An efficient coding in an arbitrary language requires e.g. 344 bits. The sender additionally transfers c as single character (8 bits) instead of adding a print statement to the little program (which would require 64 bits). Thus, 352 bits are required to transfer the string in total. This clever compression reduces the communication cost to 2.15%. In the example, object c is an outlier generated by a different mechanism than the other objects. The regular objects can be strongly compressed by formulating the underlying mechanism with a model (here, the little program) and require only a transfer of $344/2,000 = 0.17$ bits, each. Compared to the transfer of 8 bits for object c, the increase in coding costs can deposit c as an outlier. Unlike the character strings in the simple example, the object P is a point, i.e. a d-dimensional vector of continuous values. Inferred from the idea of Huffman coding, we can apply the data compression idea by assigning few bits to frequent values and many bits to rare values of the coordinates of P. Frequent and rare values can be clearly distinguished using the above mentioned EPD. This principle is generally called the *Minimum Description Length* (MDL).

CoCo effectively applies the MDL principle to parameter-free outlier detection. No *a-priori* information about the data set is required, like the number of clusters and outliers, the cluster size, a distance metric, or the cluster density. Furthermore, we define a CoCo outlier factor with the concept of coding costs of an object, given the entire data set. With the outlier factor we can clearly separate the cluster points from the outliers.

The paper is organized as follows: In the next section, we briefly survey the related work. In Section 3 we introduce CoCo by elaborating a flexible model for continuous data relying on two major building blocks: The Independent Component Analysis (ICA) and the EPD. Furthermore, we define our CoCo outlier factor. Section 4 provides an extensive experimental evaluation and Section 5 concludes the paper.

2. RELATED WORK

The most established approaches to outlier detection in databases can be classified into the two categories of distance- and density-based approaches. Additionally, a brief survey of the application of the information-theoretic MDL principle in data mining is given. For an extended survey on anomaly detection please refer to [4].

2.1 Distance-based Outlier Detection

Distance-based outlier detection is among the earliest approaches and has been proposed and further elaborated by E.M. Knorr and R.T. Ng [10, 11, 12]. An object o of a database DB is a distance-based outlier if at least a fraction β of the objects in DB have a distance greater than a previously specified distance d. This basic approach provides binary flagging of points as outliers or non-outliers. An extension [12] proposes algorithms to support semantic interpretation of distance-based outliers. However, without knowledge of the data distribution, it is difficult to specify suitable values for the parameters β and d. In addition, a fixed distance threshold d identifies only global outliers.

2.2 Density-based Outlier Detection

Density-based outlier detection introduces an outlier notion derived from density-based clustering and, therefore, detects not only global but also local outliers. A point is flagged as an outlier if it does not fit well into the objects neighborhood density.

The local outlier factor LOF [3] formalizes this idea by considering the $MinPts$ nearest neighbors of an object as its neighborhood. The LOF of an object is defined by the ratio of its $MinPts$-nearest neighbor distance and the mean $MinPts$-nearest neighbor distance in its neighborhood. However, the global parameter $MinPts$ strongly affects the outlier detection result: Arbitrary high or low values of $MinPts$

either regard small cluster points as outliers or do not detect outliers, respectively.

LOCI [14] is a density-based multi-granularity outlier factor. Similar to LOF, points are regarded as outliers if the object density in their local neighborhood significantly deviates from the average object density in the local neighborhood. The local neighborhood is specified by two parameters, which are called counting and sampling neighborhood. The counting neighborhood specifies some volume of the feature space which is used to estimate the local object density. The sampling neighborhood is larger than the counting neighborhood and contains all points which are used to compute the average object density in the neighborhood. LOCI differs from LOF by this decoupling of counting and sampling neighborhoods. It can be demonstrated that without this decoupling, density estimation leads to incorrect results in some specific cases. In addition, the decoupling allows for efficient algorithms for approximate computation of LOCI. However, the decoupling requires the specification of additional parameters. Together with the outlier factor, the LOCI approach proposes a visualization, the so-called LOCI plot which displays the LOCI of a point w.r.t. increasing sizes of the local neighborhood and, thereby, allows e.g. to identify micro-clusters. However, LOCI as well as LOF apply the Euclidean distance as a global metric distance function. In addition, the LOCI approach proposes to flag points as outliers which deviate in their local object density more than three times of the standard deviation of the overall object density of the sampling neighborhood. This flagging assumes a Gaussian distribution of the object densities.

2.3 Minimum Description Length in Data Mining

Information-theoretic concepts, especially the MDL principle and related ideas have been recently successfully applied to clustering [1, 2, 15], and are also established in the areas of regression [17], rule mining [19], classification [8], and anomaly detection [7]. The MDL principle relates learning and data compression, as already illustrated in Figure 1. Learning regularities from data allows to compress the data more efficiently. For model selection in clustering and classification, MDL allows to compare different candidate models achieving a natural balance between goodness of fit and model complexity. To the best of our knowledge, the MDL principle has not been applied to the problem of outlier detection so far.

Regarding the problem specification, clustering is most related to outlier detection. However, outliers are regarded as a problem for clustering, since they can severely affect the result of most algorithms. A parameter-free extension of K-Means clustering is X-Means [15]. However, the X-Means algorithm is restricted to spherical Gaussian clusters and very sensitive to outliers. RIC [1] has been designed as a post-processing step to improve an initial clustering of an arbitrary conventional clustering algorithm. After filtering the initial clusters from noise, for each cluster a model is determined. This model comprises a rotation matrix determined by PCA and a PDF assigned to each coordinate selected from a set of predefined PDFs. The recently proposed algorithm OCI [2] introduces a very general clustering notion based on the EPD and ICA. Also related are approaches to MDL-based de-noising of signals [16, 18]. However, these approaches are especially designed for time series and their goal is to reconstruct the signal as accurate as possible.

3. COCO - CODING COST OUTLIER DETECTION

With CoCo, we introduce an entirely parameter-free outlier detection method based on coding costs. Following Hawkins [5], we adapt the outlier definition to the MDL principle for data compression. A data point is considered as outlier, if its compression rate is unusually high. As reference to define a high compression rate, we consult a compression rate of a cluster point. This approach nicely avoids the definition of a distance metric which would require thresholding of an undefined and unknown neighborhood.

Data sets may be rotated or distorted with respect to the Cartesian coordinate system. The ICA enables us to process data sets which are not aligned to the orthogonal axes. However, the idea of an ordinary point needs to be clearly defined. In contrast to currently available outlier detection methods, we expect real life data to underly not only Gaussian distributions. Besides, we want to include several other distributions. A generalization of the Gaussian PDF is the EPD. The EPD includes, among others, the uniform, and the Laplacian PDF. By utilizing an EPD, any *a-priori* information on the type of distribution is required. Therefore, we do not create a bias towards Gaussian data models. Combining ICA with EPD as the description of a *regular* subset of the data set, we cover many real-world data sets without taking explicit care of cluster density, shape, and orientation.

Entirely automatic, CoCo detects outliers having high coding costs with respect to the ordinary points which can be effectively compressed. We implemented a bottom-up approach to identify all irregular data points while choosing the best compression model of ordinary points.

For each data point o, we initiate a set of nearest neighbors. Without prior knowledge of the underlying cluster shape, we extract a substantial number of nearest neighbors nn_o based on their Euclidean distance to o. We reliably center and whiten the set of nearest neighbors with ICA, ica_{nn_o}, and fit an EPD, epd_{nn_o}. Iteratively, we expand the nearest neighbor set with those remaining data points to be best compressed based on the current epd_{nn_o}. After each update of the set of nearest neighbors nn_o, we simply adjust the ica_{nn_o} and epd_{nn_o} since it is an expensive operation to estimate it anew. For each epd_{nn_o} estimate, we can calculate the coding costs $cost_o$ as compression rate of the object o under the given cluster description epd_{nn_o}. If the data is fully explored for each object o, we extract the most suitable EPD cluster model by selecting the minimum compression rate of any object included in $cost\_min_{nn_o}$. The outlier factor for the data object $cost_o(j)$ is determined by its corresponding compression excess to $cost\_min_{nn_o}(j)$.

The following defines the principles of ICA, EPD, data compression, and their link to the parameter-free outlier detection with CoCo.

3.1 Independent Component Analysis

It was observed that mixtures of signals get best de-mixed when searching for non-Gaussianity. A mixture of several signals originated from any distribution type is always more Gaussian than the originals. The entropy of a Gaussian dis-

Algorithm 1 CoCo

Input: Database D

$OF := \{\}$ // Outlier Factors
for data object $o \in D$ **do**
  $cost_o := \{\}$
  $cost\_min_{nn_o} := \{\}$
  $nn_o :=$ initial set of Nearest Neighbors
  $not\_nn_o := D \backslash nn_o$

  $ica_{nn_o} :=$ ICA(nn_o)
  $nn_{o,ica} :=$ transform(nn_o; ica_{nn_o})
  $epd_{nn_o} :=$ estimate EPD($nn_{o,ica}$)
  while $not\_nn_o \neq \{\}$ **do**
    $cost_o := cost_o \cup$ coding_cost(o_{ica}; epd_{nn_o})
    $cost\_min_{nn_o} := cost\_min_{nn_o} \cup$
                  min(coding_cost($nn_{o,ica}$; epd_{nn_o}))

    $not\_nn_{o,ica} :=$ transform(not_nn_o; ica_{nn_o})
    $cost_{not\_nn_{o,ica}} :=$ coding_cost($not\_nn_{o,ica}$; epd_{nn_o})

    $nn_o := nn_o \cup \{not\_nn_{o,ica}$ with lowest $cost_{not\_nn_{o,ica}}\}$
    $not\_nn_o := D \backslash nn_o$

    update ica_{nn_o}
    $nn_{o,ica} :=$ transform(nn_o; ica_{nn_o})
    update epd_{nn_o}
  end while
  $j :=$ min($cost\_min_{nn_o}$)// index best compressed cluster
  $OF := OF \cup (cost_o(j) - cost\_min_{nn_o}(j))$
end for

XMeans(OF) to obtain outlier & cluster points

tribution is maximal, whereby, all other distributions have a lower entropy. However, the coding costs, measured by the entropy, need to be minimized in order to guarantee a maximal compression efficiency. Thus, we apply the ICA to maximize non-Gaussianity as a measure of statistical independence. Its algorithm favors the directions in the data which are not similar to the Gaussian distribution.

We assume that most data sets in experimental data usually do not follow equally dense distributions. They are rather distorted data sets with respect to the Cartesian coordinate system. The ICA first transforms the data into a so-called white space. Whitening involves de-correlation and normalization of the data to unit variance which enables us to implicitly handle unequally dense clusters.

The Principal Component Analysis (PCA) identifies the directions of maximal variance $\vec{y}$ given a set of coordinates $\vec{x} \in C$ in a d-dimensional space. First, the data get centered $\vec{c} = \vec{x} - \vec{m}$ around the empirical mean

$$\vec{m} = \frac{1}{|C|} \sum_{\vec{x} \in C} \vec{x}$$

of the data set C. Second, the centered data $\vec{c}$ need to be normalized to unit variance in all directions. The eigenvalue decomposition of the covariance matrix Σ is $\Sigma := V \times \Lambda \times V^T$, where V and Λ are orthogonal matrices containing the eigenvectors and eigenvalues of Σ, respectively. Finally, the PCA transform of $\vec{x}$ is determined by

$$\vec{y} := \sqrt{\Lambda}^{-1} \times V^T \times \vec{c}.$$

Note, that $\Lambda = diag(\lambda_1, \ldots, \lambda_d)$ and $\sqrt{\Lambda}^{-1} = diag(\sqrt{1/\lambda_1}, \ldots, \sqrt{1/\lambda_d})$ are both diagonal matrices.

For optimal projection of the data we need to determine the directions of minimal entropy (generated with ICA) rather than the one of maximal variance (created by PCA). After transforming the data to white space, the FastICA algorithm [6] determines a weighting matrix W containing the independent components. Regarding the original space, the independent components are not orthonormal in contrast to the principal components. The iterative optimization of W expects the input data to be whitened. The fix point iteration optimizes $W = (\vec{w}_1, \ldots, \vec{w}_d)$, whereby the weight vectors are updated with the following rule:

$$\vec{w}_i := E\{\vec{y} \times g(\vec{w}_i{}^T \times \vec{y})\} - E\{g'(\vec{w}_i{}^T \times \vec{y})\} \times \vec{w}_i$$

We use $\tanh(s)$ for the non-linear contrast function $g(s)$. Note that $g'(s) = \frac{dg(s)}{ds}$ is the derivative of $g(s)$ and $E\{\ldots\}$ is the expected value. W is updated until convergence and then orthonormalized. The overall projection of the original data into the white space of independent components is achieved by the de-mixing matrix M^{-1}. With $M = V \times \sqrt{\Lambda} \times W$ we denote

$$M^{-1} = W^T \frac{1}{\sqrt{\Lambda}} V^T.$$

W and V are orthonormal matrices, thus the determinant is simply $det(M^{-1}) = \prod_{1 \leq i \leq d} \sqrt{1/\lambda_i}$. Recall that the rotation performed in the white space is expressed by W, and whitening is achieved by multiplying the coordinate vector by the scaled Eigenvector matrix.

After the independent components are determined, we can simply project the data $\vec{x}$ into the independent component space with

$$\vec{z} = M^{-1} \times (\vec{x} - \vec{m}).$$

3.2 Exponential Power Distribution

The EPD is a generalization of the Gaussian distribution in such a way, that it also includes the Laplacian and the uniform distribution, depending on the parameter setting. Its PDF has three different parameters. Beside the location parameter μ, and the scale parameter σ, a shape parameter p is introduced [13]. For a random variable X, the EPD is

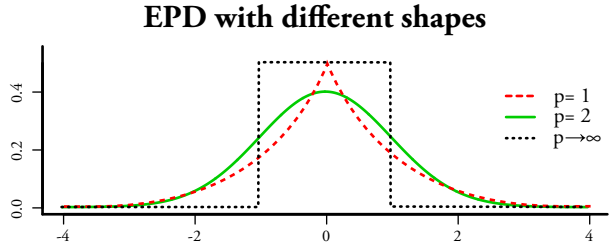

Figure 2: Different shapes of the Exponential Power Distribution for different choices of parameter p.

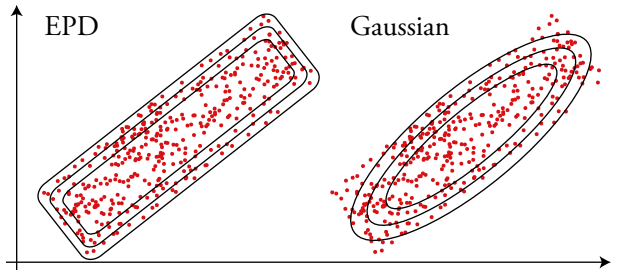

Figure 3: Data set approximated with an EPD and a Gaussian distribution.

defined as:

$$f_{EPD}(x;\mu,\sigma,p) = \frac{\exp(-\frac{|x-\mu|^p}{p\sigma^p})}{2\sigma p^{\frac{1}{p}}\Gamma(1+\frac{1}{p})}$$

Note that $\Gamma(s) = \int_0^\infty t^{s-1}\exp(-t)dt$ is the gamma function as an extension of the factorial operator for real numbers.

The shape parameter p determines kurtosis, or the sharpness of the distribution. For $p > 2$, the EPD is platykurtic, with $p \to \infty$ mimicking a uniform distribution. For $p = 2$, the EPD corresponds to a Gaussian distribution. If $2 < p < 0$, the EPD is leptokurtic, including a Laplacian distribution for $p = 1$ (Fig. 2).

3.3 EPD after ICA

After projection of the coordinates into the white space and ICA, the data $\vec{z}$ is de-correlated and independent. This allows us to describe each coordinate independently by an own EPD. Typically, a multi-dimensional data space contains d different PDF representations $f_{EPD}(z_i;\mu_i,\sigma_i,p_i)$ with $1 \le i \le d$. All d distributions are combined in a mixing matrix M, where the data points $\vec{x}$ correspond to $\vec{x} = M \times \vec{z} + \vec{m}$, with $\vec{m}$ being the shifting vector and M determined by PCA, as described above. M allows the independent components vectors to be not orthogonal. The EPD in a d-dimensional space (after ICA) is defined for a point $\vec{x}$ as

$$f_{EPD}(x;M^{-1},\vec{m},\mu,\sigma,p) = \frac{\prod_{1\le i\le d} f_{EPD}(z_i;\mu_i,\sigma_i,p_i)}{|\det(M^{-1})|}$$

Figure 3 illustrates the effect of the approximation of a data set with an EPD after ICA. While the approximation of the same data with a Gaussian distribution is rather inappropriate.

3.4 EPD Approximation

The estimation of the three parameters is a non-trivial problem. Although, $\mu_i = 0$ and $\sigma_i = 1$ are defined for p=2 (Gaussian distribution) after ICA, μ_i and σ_i are no longer identical to the empirical mean and standard deviation, respectively. All three parameters μ_i, σ_i and p_i can be optimized by estimating the maximum likelihood, given a data set C. Only a simultaneous approximation of all parameters ensures that the derivatives of the likelihood of the EPD vanish with respect to μ_i, σ_i and p_i.

Assuming μ_i and p_i to be given, the parameter σ_i can be simply calculated with the derivative of the likelihood function with respect to σ_i of the EPD $\sum_{\vec{z}\in C} f_{EPD}(\vec{z_i};\mu_i,\sigma_i,p_i)$:

$$\frac{df_{EPD}(C;\mu_i,\sigma_i,p_i)}{d\sigma_i} = -\frac{|C|}{\sigma_i} + \frac{\sum_{\vec{z}\in C}|z_i-\mu_i|^{p_i}}{\sigma_i^{p_i+1}} = 0.$$

$$\Rightarrow \sigma_i = \left(\frac{1}{|C|}\sum_{\vec{z}\in C}|z_i-\mu_i|^{\frac{1}{p_i}}\right)$$

The parameters μ_i and p_i need to be optimized explicitly. We use a nested bisection search as optimization technique to find p_i and μ_i in their parameter space. The direction to browse through the space is determined by the derivatives of the log-likelihood function with respect to μ_i

$$\frac{df_{EPD}(C;\mu_i,\sigma_i,p_i)}{d\mu_i} = -\frac{1}{\sigma_i^p}\sum_{\vec{z}\in C}|z_i-\mu_i|1^{p_i-1}sign(z_i-\mu_i)$$

and p_i

$$\frac{df_{EPD}(C;\mu_i,\sigma_i,p_i)}{dp_i} = -\frac{|C|}{p_i^2}\left(\log p_i + \Psi(1+\frac{1}{p_i}) - 1\right) + \frac{\sum_{\vec{z}\in C} s_i^{p_i} + p\log\sigma_i \sum_{\vec{z}\in C} s_i^{p_i} - p\sum_{\vec{z}\in C}(s_i^{p_i}-\log s_i)}{p_i^2\sigma_i^{p_i}},$$

with $s_i = |z_i - \mu_i|$. $\Psi(s) = \frac{d\ln\Gamma(s)}{ds}$ is the digamma function being the logarithmic derivative of the gamma function. The EPD is estimated by this maximum likelihood approach until convergence of p_i.

3.5 Coding Cost with MDL

After we estimated an exact representation $f_{EPD}(x;M^{-1},\vec{m},\mu,\sigma,p)$ of the data $\vec{x}$ with ICA and EPD, we need a reliable approach to judge the accuracy of the fit. We create the link of the concept of PDFs to the principle of data compression with the help of the MDL. Based on the Huffman coding, a number of bits are assigned to each object with the inverse logarithm of the probability of the object. This negative log-likelihood represents the coding costs c_{PDF} of an object $\vec{x}$, given any PDF, and is defined as:

$$c_{PDF}(\vec{x}) = \log_2\left(\frac{1}{f_{PDF}(\vec{x})}\right) = -\log_2(f_{PDF}(\vec{x})).$$

In order to represent the coding cost in the number of bits, the logarithm is typically used to a basis of 2. With CoCo, we underly an EPD as PDF. Thus, the relative coding cost of a data point $\vec{x}$ under a given EPD after ICA is:

$$c_{EPD}(\vec{x}) = \log_2\left(|det(M^{-1})|\right) - \sum_{1\le i\le d}\log_2\left(f_{EPD}(\vec{z};\mu_i,\sigma_i,p_i)\right).$$

We can neglect to determine the absolute coding costs depending on different PDFs and the coding of the PDF parameters. It is absolute crucial to determine statistically independent major directions with ICA to guarantee optimal data compression. Figure 4 clearly demonstrates that ICA transforms the data in such a way that it removes redundancy in the data with respect to the axes for best compression.

3.6 CoCo Outlier Factor and Detection

Putting everything together, for each set of coordinates $\vec{x}$ from the nearest neighbors nn_o generated with CoCo, we determine the rotation and the cluster description with EPD epd_{nn_o}. For each estimate epd_{nn_o}, the data compression

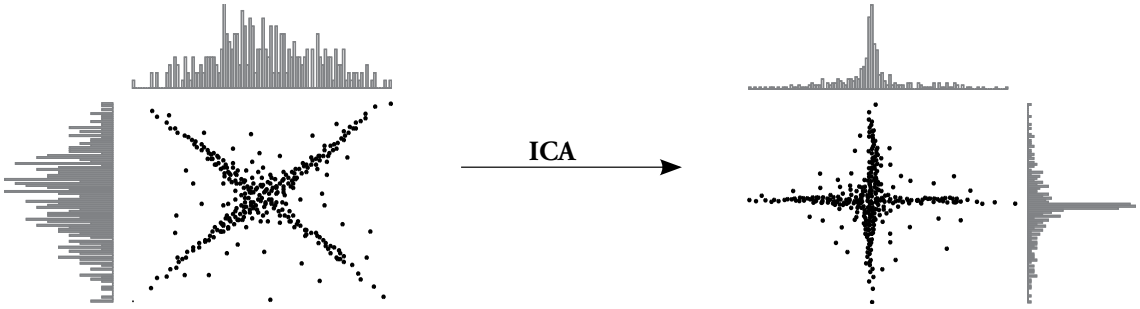

Figure 4: ICA creates redundancy in the data by centering and whitening.

rate is calculated with $c_{EPD}(\vec{o})$, $\vec{o}$ being the whitened coordinates of object o. We determine the efficiency to compress the data points nn_o, with an epd_{nn_o} estimate, with any object $p \in nn_o$ having minimal coding cost: We gather information of compression rates for each set of nn_o with increasing size. Ideally we need to know the optimal neighborhood cluster size of o to determine the perfect compression of o regarding C. Practically, we only have information for each epd_{nn_o} estimate throughout the data set. With it comes the information of any object (p) exhibiting the minimal coding cost in nn_o. The best compression rate ($\min(cost\_min_{nn_o})$, throughout all generated nn_o sets) represents the best epd_{nn_o} estimate for any nn_o. In order to obtain the factor of o being an outlier, the CoCo outlier factor is the absolute compression rate increase with respect to a minimal p.

The structure of a data set is usually unknown. We screen C coming from o iteratively by adding a set of neighbors; its size growing exponentially with respect to the size of C. To guarantee a stable estimate of EPD we initiate nn_o with a set of 20 neighbors. This screening approach of CoCo is however quadratic in the number of points n. In addition, the runtime is cubic in the dimensionality d due to PCA and EPD estimation.

After all CoCo outlier factors are obtained, we expect all outliers to exhibit unusually high costs in comparison to the ordinary, perhaps clustered points. The cluster points can be compressed very effectively and show outlier factors around 0. Flagging of outliers is difficult, since it involves to define a suitable threshold, which is a non-trivial task for an unknown data set. Instead, we simply apply an X-Means algorithm to determine the set of clustering points being the cluster closest to 0. Theoretically, we can establish an outlier order by simply organize the other CoCo outlier factor groups in ascending order. In practice, X-Means usually finds two clusters, one containing the clustering points, the other determining all outliers.

CoCo combines ICA with EPD as cluster description to determine outliers entirely parameter-free with the principle of data compression. No *a priori* knowledge of the number of outliers or the underlying cluster shape or density is required.

4. EVALUATION

In the following we evaluate our outlier factor CoCo in comparison to LOF [3] and LOCI [14] using one synthetic data set as well as NBA data. We implemented CoCo and LOF in Java and obtained the implementation of LOCI from the authors. The synthetic data set was created to exemplify the strength of CoCo.

4.1 Synthetic Data

We detected the outliers of a synthetic data set with our novel algorithm CoCo and compared them with outliers detected by LOF and LOCI. Figure 5 provides the results of CoCo, LOF, and LOCI for the synthetic data set. The synthetic data set consists of four clusters C1-4 containing 184 (C1), 154 (C2), 52 (C3), and 50 (C4) data points. Each

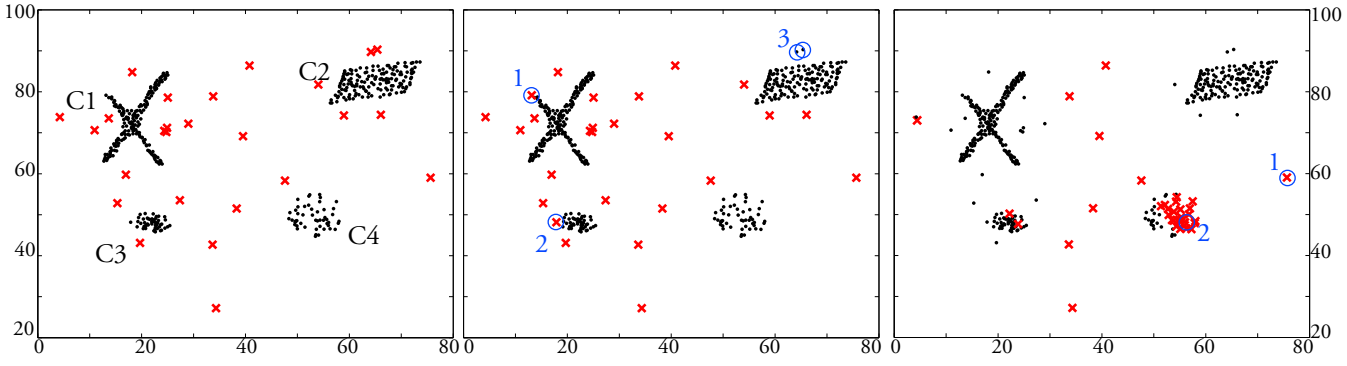

Figure 5: Outlier detection results from CoCo (left), LOF (middle, $MinPts = 50$ selecting only the top 26 outliers), and LOCI (right, $\alpha = 0.5$ and $r_{min} = 10$) for a synthetic data set consisting of four clusters (C1-4) and 26 outliers. Detected outliers are highlighted with red crosses.

cluster has different cluster properties and a non-orthogonal major orientation. Cluster C3 underlies a Gaussian PDF. All together 26 noise points were added to the data set.

CoCo correctly detects all 26 outlier points highlighted with red crosses (Fig. 5, left). All belong to one group of outliers, beside the group of cluster points shown in black. Note, that CoCo requires no input parameter in order to identify all noise points. It can handle different types of cluster shapes and orientations without expecting an explicit description of their distributions.

LOF was applied to identify the outliers based on a $MinPts$ neighborhood of 50 determined by the size of the smallest cluster in the set (Fig. 5, middle). We obtain the top 26 outliers (highlighted with red crosses) since we know how many outliers are present in the data set. There are 24 out of the 26 noise points assigned correctly. Two noise points next to cluster C2 (circled in blue as No. 3) are not detected, leading to two falsely identified cluster points as outlier (circled in blue as No.1&2). Note, that we collected the top 26 data points ranked by the LOF score. Setting the parameter $MinPts$ to a value smaller or equal than 10, LOF identifies more cluster points as outliers while leaving many true outliers undetected (data not shown). A $MinPts$ value of 20 to 50 leads to the result shown in Figure 5. If we have no a priori information about the number of outliers, it is only possible to determine an arbitrary number of outliers. In addition, an approximate cluster size needs to be known in advance to set $MinPts$, in order to get a meaningful output. These assumptions make it difficult to apply LOF to real world data.

LOCI was applied to our synthetic data set with $\alpha = 0.5$ and $r_{min} = 10$ (Fig. 5, right) and could identify 43 outlier points based on the suggested outlier flagging criteria. All together 17 true outliers were missed, while two points from within cluster C3 and 27 points from cluster C4 were labeled as outliers. Different parameter settings of r_{min} may detect more true outliers, but at the same time label more cluster points as outliers. Obviously, LOCI is not able to deal with clusters showing low density, like C4. In Figure 7, we have a closer look at the LOCI plot of an outlier point (circled in blue as No. 1 in Figure 5, right) and a cluster point (No. 2). The LOCI plots look very similar even though they are supposed to emphasize the difference between a cluster point and an outlier. We have to note, that although we applied the algorithm with the suggested parameter settings, the result was difficult to interpret even after correspondence with the authors.

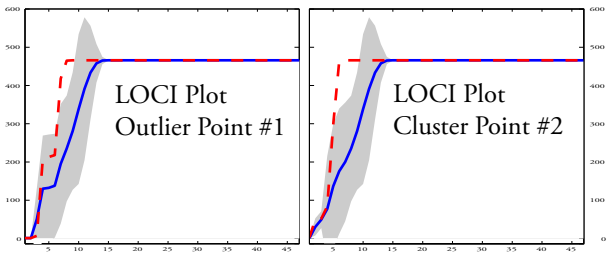

Figure 7: LOCI plot for two points detected as outliers. (1) True outlier. (2) Falsely labeled cluster point.

4.2 CoCo Outlier Factor Visualization

To emphasize the difference and strength of the CoCo outlier factor in comparison to the LOF score, we introduce a visualization of the "outlierness" (Fig. 6). A scatter plot of the data in x-y directions is combined with a bar representation of the outlier factors in the z-dimension. We can clearly show that the utilization of data compression is able to separate the outliers from the cluster points in comparison to the outlier factor of LOF. For the majority of the cluster points the CoCo coding costs are close to 0.0 which can be seen by the short, dark blue bars. Outliers are either light blue or even red indicating their extraordinariness, ranging from 6.4 up to 24.2. Due to the large range between cluster points and outliers it is possible to clearly differentiate them using CoCo. In contrast, LOF produces values ranging from 0.8 up to 2.3 which makes it almost impossible to clearly differentiate cluster points from outliers explicitly.

The visualization of the outlier-factors of LOF demonstrates, that the cluster structure is based on Euclidean dis-

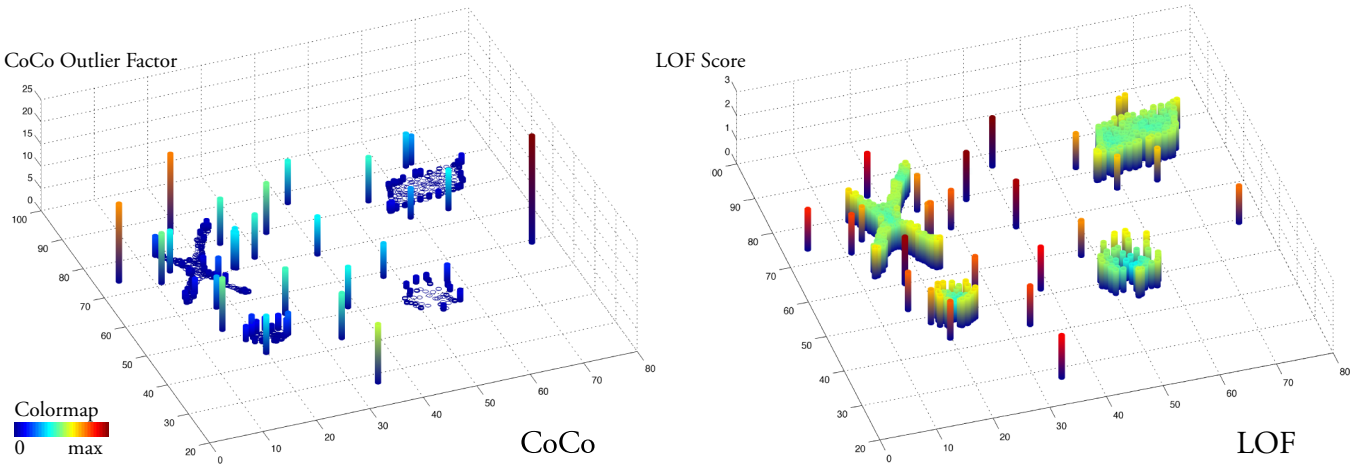

Figure 6: Visualization of the CoCo Outlier Factor and the LOF Score for the synthetic data set.

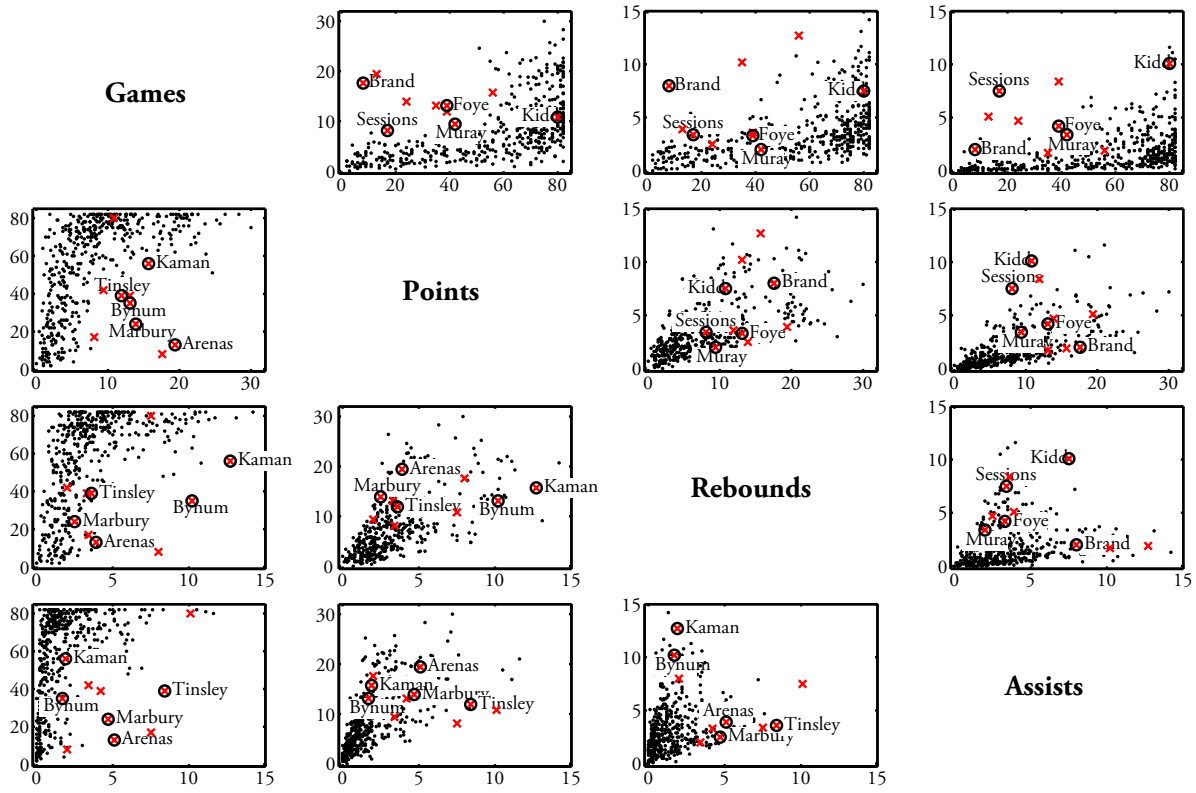

Figure 8: Outlier identification with CoCo for the NBA data set. Top-10 outlier are highlighted with red crosses and marked with player names.

tances: the outlier factors continuously increases circular from the cluster centers to the cluster margins. In contrast to LOF, the CoCo outlier factors are equally low throughout the entire cluster except for the cluster edge points. It is based on the flexible cluster structure description using ICA and EPD.

4.3 Experimental Data

After extensive evaluation of CoCo on synthetic data sets, we want to apply our novel parameter-free outlier detection method to experimental data. We used the NBA data available at the NBA website `http://www.nba.com`. In the Season 2007/08, 450 players are described with four attributes: the number of games played (GP), the number of points (PPG), the rebounds (RPG), and assists (APG) per game. CoCo was applied to this NBA data detecting 105 outliers. Figure 8 displays scatter plots of the data. For simplicity reasons, we highlight only the top 10 outliers in red as listed in Table 1. Obviously, the data distribution is non-Gaussian.

The top 10 outliers identified by CoCo, include outstanding players like Stephon Marbury with a coding cost of 19.6 being 12 times higher than the average coding costs. Marbury is an outstanding player with respect to all attributes. He played only 24 games out of 82 and was still able to achieve 13.9 points and additionally assisted in 4.7 points, resulting in being involved in 18.6 points per game. Jamaal Tinsley, has played 39 games in this season but was still able to assist in 8.4 game points. He was involved in 20.3 points and played more games than Marbury. Gilbert Arenas exhibits a rare combination of playing 13 games while achieving 19.4 points per game. Jason Kidd is outstanding in the number of rebounds having played in 80 out of 82 games. Elton Brand has played only few games but still was able to achieve an outstanding number of points. As evident from Figure 8, outstanding players such as Kidd or Brand are best characterized with the most general model with only one component.

To put the CoCo outlier detection method into a context, we applied LOF and LOCI to the NBA data set, as well. Table 2 displays the top 10 outliers identified by LOF. Highlighted in bold are all players that were identified as top 10 outlier of CoCo, like Marbury, Arenas, or Brand. Except for one, all players from the top 10 outlier of CoCo are at least under the top 20 of LOF. However, the outstanding player Kidd was missed by LOF ranked at the 50th position with a LOF score of 1.16. In addition, as observed for synthetic data, the result of LOF strongly depends on its parameterization. Only seven players are reproducibly detected as top 10 for a $MinPts = 40$ (players are marked with an asterix). All five players which were found to be under the top 10 of CoCo were also included in the intersect of $MinPts = 40$ and $MinPts = 50$ which strikes that they are strongly outstanding. The top 10 outliers found by LOCI are shown in Table 3. The intersect between LOCI and CoCo is again highlighted in bold.

CoCo o.f.	Name	GP	PPG	RPG	APG
19.6	Stephon Marbury	24	13.9	2.5	4.7
17.9	Jamaal Tinsley	39	11.9	3.6	8.4
16.1	Gilbert Arenas	13	19.4	3.9	5.1
15.4	Andrew Bynum	35	13.1	10.2	1.7
13.6	Elton Brand	8	17.6	8	2
12.9	Ronald Muray	73	9.1	4.5	1.3
12.8	Jason Kidd	80	10.8	7.5	10.1
12.5	Chris Kaman	56	15.7	12.7	1.9
12.3	Ramon Sessions	17	8.1	3.4	7.5
12.0	Randy Foye	39	13.1	3.3	4.2

Table 1: Top 10 outliers identified with CoCo on NBA data. (o.f. = outlier factor).

LOF	Name	GP	PPG	RPG	APG
1.43	**Elton Brand***	8	17.6	8	2
1.32	Steve Francis	10	5.5	2.3	3
1.31	Kasib Powell	11	7.6	4	1.6
1.28	**Gilbert Arenas***	13	19.4	3.9	5.1
1.28	Chris Webber*	9	3.9	3.6	2
1.27	**Stephon Marbury***	24	13.9	2.5	4.7
1.26	Dwyane Wade*	51	24.6	4.2	6.9
1.25	LeBron James	75	30	7.9	7.2
1.24	**Andrew Bynum***	35	13.1	10.2	1.7
1.24	**Chris Kaman***	56	15.7	12.7	1.9

Table 2: Top 10 outliers identified by LOF with $MinPts = 50$ on NBA data sorted by outlier-factor. Players also among the top 10 of CoCo are marked in bold. The asterix indicates players which are also among the top 10 using $MinPts = 40$. Note that all players found to be under the top 10 of CoCo and LOF $MinPts = 50$ are also found using $MinPts = 40$.

5. CONCLUSION

In this paper, we proposed CoCo, a parameter-free outlier detection. The perspective of data compression in outlier detection allows to define a notion of outliers, which is intuitive to interpret and requires no parameter settings. Our experiments demonstrate that CoCo is not restricted to Gaussian data but applicable to a wide range of data distributions.

In future work, we will further elaborate techniques to facilitate the interpretation of cost-based outliers. In addition, we will focus on online algorithms for cost-based outlier detection in data streams, since online monitoring is essential in many applications involving outlier detection.

Name	GP	PPG	RPG	APG
LeBron James	75	30	7.9	7.2
Kobe Bryant	82	28.3	6.3	5.4
Dwyane Wade	51	24.6	4.2	6.9
Chris Kaman	56	15.7	12.7	1.9
Elton Brand	8	17.6	8	2
Andrew Bynum	35	13.1	10.2	1.7
Jamaal Tinsley	39	11.9	3.6	8.4
Mike Bibby	48	13.9	3.3	6
Jermaine O'Neal	42	13.6	6.7	2.2
Udonis Haslem	49	12	9	1.4

Table 3: Top 10 outliers identified by LOCI on NBA data. Players also among the top 10 of CoCo are marked in bold.

6. REFERENCES

[1] C. Böhm, C. Faloutsos, J.-Y. Pan, and C. Plant. Robust information-theoretic clustering. In *KDD Conference*, pages 65–75, 2006.

[2] C. Böhm, C. Faloutsos, and C. Plant. Outlier-robust clustering using independent components. In *SIGMOD Conference*, pages 185–198, 2008.

[3] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander. Lof: Identifying density-based local outliers. In *SIGMOD Conference*, pages 93–104, 2000.

[4] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. *ACM Computing Surveys*, 2009.

[5] D. Hawkins. *Identification of Outliers*. Chapman and Hall, London, 1980.

[6] A. Hyvärinen, J. Karhunen, and E. Oja. *Independent Component Analysis*. 2001.

[7] E. Keogh, S. Lonardi, and C. A. Ratanamahatana. Towards parameter-free data mining. In *KDD '04: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 206–215, New York, NY, USA, 2004. ACM.

[8] S. Kim and I.-S. Kweon. Simultaneous classification and visualword selection using entropy-based minimum description length. In *ICPR (1)*, pages 650–653, 2006.

[9] E. M. Knorr. On digital money and card technologies. Technical Report Technical Report 97-02, University of British Columbia, 1997.

[10] E. M. Knorr and R. T. Ng. A unified notion of outliers: Properties and computation. In *KDD*, pages 219–222, 1997.

[11] E. M. Knorr and R. T. Ng. Algorithms for mining distance-based outliers in large datasets. In *VLDB*, pages 392–403, 1998.

[12] E. M. Knorr and R. T. Ng. Finding intensional knowledge of distance-based outliers. In *VLDB*, pages 211–222, 1999.

[13] A. Mineo and M. Ruggieri. A software tool for the exponential power distribution: The normalp package. *Journal of Statistical Software*, 12(4), 1 2005.

[14] S. Papadimitriou, H. Kitagawa, P. B. Gibbons, and C. Faloutsos. Loci: Fast outlier detection using the local correlation integral. In *ICDE*, pages 315–, 2003.

[15] D. Pelleg and A. Moore. X-means: Extending K-means with efficient estimation of the number of clusters. In *ICML Conference*, pages 727–734, 2000.

[16] J. Rissanen. Mdl denoising. *IEEE Transactions on Information Theory*, 46(7):2537–2543, 2000.

[17] M. Robnik-Sikonja and I. Kononenko. Pruning regression trees with mdl. In *ECAI*, pages 455–459, 1998.

[18] J. Xie, D. Zhang, and W. Xu. Spatially adaptive wavelet denoising using the minimum description length principle. *IEEE Transactions on Image Processing*, 13(2):179–187, 2004.

[19] T. Yoshida, H. Motoda, and T. Washio. Adaptive ripple down rules method based on minimum description length principle. In *ICDM*, pages 530–537, 2002.