Available online at www.sciencedirect.com
SCIENCE DIRECT®
Neurocomputing 69 (2006) 1608–1618
NEUROCOMPUTING
www.elsevier.com/locate/neucom

From outliers to prototypes: Ordering data

Stefan Harmeling[a,b,*], Guido Dornhege[a], David Tax[c], Frank Meinecke[a], Klaus-Robert Müller[a,b]

[a]*Fraunhofer FIRST.IDA, Kekuléstrasse 7, 12489 Berlin, Germany*
[b]*Department of Computer Science, University of Potsdam, August-Bebel-Strasse 89, 14482 Potsdam, Germany*
[c]*Delft University of Technology, Information and Communication Theory Group, P.O. Box 5031, 2600 GA, Delft, The Netherlands*

Received 1 July 2004; received in revised form 21 May 2005; accepted 24 May 2005
Available online 1 December 2005
Communicated by S. Hochreiter

Abstract

We propose simple and fast methods based on nearest neighbors that order objects from high-dimensional data sets from typical points to untypical points. On the one hand, we show that these easy-to-compute orderings allow us to detect outliers (i.e. very untypical points) with a performance comparable to or better than other often much more sophisticated methods. On the other hand, we show how to use these orderings to detect prototypes (very typical points) which facilitate exploratory data analysis algorithms such as noisy nonlinear dimensionality reduction and clustering. Comprehensive experiments demonstrate the validity of our approach.

Keywords: Outlier detection; Novelty detection; Ordering; Noisy dimensionality reduction; Clustering; Nearest neighbors

1. Introduction

Exploratory data analysis tries to find simple representations of high-dimensional data that best reflect the underlying structures. There are few robust methods that can be used for this purpose in high dimensions. In fact, most real-world data sets are spoiled by outliers arising from many different processes, e.g. measurement errors or miss-labeled samples. So it is necessary to remove outliers from the data to avoid erroneous results.

In the statistics literature, a large emphasis is put on the problem of outlier detection in *univariate* data [2,18]. In univariate data the objects are trivially ordered, which eliminates the problem of finding a one-dimensional measure for characterizing the typicality of an object. The main challenge is then to decide where to set the threshold to distinguish between genuine and outlier objects.

For the problem of outlier detection in multivariate data, more complicated models have to be applied in order to impose an ordering on the data. A well known method from the statistics community is the minimum covariance determinant (MCD) estimator [29]: find h observations out of n, such that its covariance matrix has the smallest determinant. The objects are then ordered according to their Mahalanobis distance to the data mean. Although this method is very robust, it is not very flexible because it only fits a Gaussian distribution to the data. More flexible density models include the Parzen density model [3] or the mixture of Gaussians [31]. The Parzen density can be approximated by defining the support of a data set by fitting balls of fixed size around the training set [12,1]. Unfortunately, density estimation in high-dimensional spaces is difficult, and in order to reliably estimate the free parameters, the models have to be restricted significantly in complexity.

From the pattern recognition/machine learning field more heuristic methods originate, for instance, neural

*Corresponding author. Fraunhofer FIRST.IDA, Kekuléstrasse. 7, 12489 Berlin, Germany.
E-mail addresses: harmeli@first.fhg.de (S. Harmeling), dornhege@first.fhg.de (G. Dornhege), davidt@first.fhg.de (D. Tax), meinecke@first.fhg.de (F. Meinecke), klaus@first.fhg.de (K.-R. Müller).

0925-2312/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2005.05.015

network models [23,19] or models which are inspired by the support vector classifiers [32,7,34]. They avoid performing the often very difficult density estimation, and directly fit a decision boundary around the data, but are often not simple to implement and optimize. Also the outputs of traditional two-class classifiers can be used for outlier detection [38], thus focusing on the outliers from the perspective of the classification problem.

Ref. [20] studies distance-based outliers which are defined with respect to two parameters p and D: a data point x is a distance-based outlier, abbr. $DB(p, D)$-outlier, if at least fraction p of the other points lies greater than distance D from x. The $DB(p, D)$-outlier are global outliers, because D and p are chosen for all data points. If the data consists of several clusters with different variances, it can be difficult to choose a single D which is appropriate. Thus, methods have been developed which focus on local properties of the data, e.g. local outlier factors (LOF, see [6]). LOF introduces an outlier index which is based on a sophisticated theory of "local reachability" and nearest neighbors, which gives rise to a somewhat convoluted index. The outlier indices proposed in this paper are defined in terms of nearest neighbors as well, but are designed to be as simple and straightforward as possible.

Most of the existing methods implicitly imply certain definitions of what an outlier actually is, which are often not explicitly stated. In this paper, we call data points outliers if their true probability density is very low. Obviously, the difficulty of this particular notion is that the true probability density is unknown and it is a challenge to obtain a reasonable estimate—especially in high dimensions. However, the indices proposed in this paper, some of which have been previously used for outlier detection in [28,14], coarsely approximate the probability density. Thus these indices are in principle applicable to all settings that assume outliers to be data points in sparse regions.

Notice that most methods mentioned above provide an ordering of objects in a data set, according to their typicality. Very untypical objects are candidates to be labeled as outliers. On the other hand, it is of similar importance to detect the most common or prototypical samples in a data set. The latter is often useful to gain a better *understanding* of the data. This idea will be elaborated in this paper as well.

Summing up, this paper proposes simple indices (see Section 2) based on nearest neighbors that allow an ordering of the data from outliers to prototypes. Once this representation is established we can use it for (1) prototype detection (see Section 3.1), (2) outlier removal, or accordingly novelty detection (see Section 3.2), and (3) robustification of unsupervised algorithms (see Section 3.3). Experiments on toy and real data sets and handwritten digits underline the practicability of our algorithm, in particular for high-dimensional data sets.

2. Indices for ordering

Consider a set of n data points from the d-dimensional Euclidean space,

$$\{x_1, \ldots, x_n\} \subset \mathfrak{R}^d,$$

with the Euclidean norm, $\|x\| = \sqrt{x^\top x}$, and the Euclidean metric. Other metrics (e.g. other Riemannian metrics or Mahalanobis distance) can be effortlessly incorporated in our framework. For a data point $x \in \mathfrak{R}^d$, let

$$z_1(x), \ldots, z_k(x) \in \{x_1, \ldots, x_n\} \subset \mathfrak{R}^d,$$

be its k nearest neighbors among the given data points $x_1, \ldots, x_n$ (with respect to the chosen metric). In terms of these neighbors, we define three indices for each point $x \in \mathfrak{R}^d$. We will later use them for the ordering process. As usual, the choice of k influences the perception of the data: if k is chosen too small the focus is too local, if k is too large it is too global.

2.1. Kappa

The k-nearest neighbor density estimator assesses the density at a particular point by calculating the volume of the smallest ball centered at that point which contains its k nearest neighbors and relating it to the quotient k/n. It can be proven that this density estimator is L_2-consistent (see [22]). Unfortunately, the estimate is not always very accurate if the number of data points is small or the dimensionality is high. However, outlier detection does not require the actual density. In order to decide whether a data point is an outlier or not, an approximate estimate is a sufficient indicator. Our first index thus represents the essence of the k nearest neighbor density estimator: $\kappa(x)$ is the radius of the smallest ball centered at x containing its k nearest neighbors, i.e. the distance between x and its kth nearest neighbor,

$$\kappa(x) = \|x - z_k(x)\|.$$

Obviously, in dense regions κ is small and in sparse regions κ is large, making it a good candidate for an outlier index, as the rationale is that outliers lie in sparse regions.

2.2. Gamma

The index κ, however, seems to be somewhat wasteful: it considers the distance to the kth nearest neighbor, but it ignores the distances to the closer neighbors. This suggests a refined index that takes the distances to all k nearest neighbors into account: $\gamma(x)$ is x's average distance to its k nearest neighbors,

$$\gamma(x) = \frac{1}{k} \sum_{j=1}^{k} \|x - z_j(x)\|.$$

This index enables us to distinguish the two situations depicted on the left panel of Fig. 1: the value of κ is the