Tilburg centre for Creative Computing
Tilburg University
http://www.uvt.nl/ticc

P.O. Box 90153
5000 LE Tilburg, The Netherlands
Email: ticc@uvt.nl

February 3, 2012

TiCC TR 2012–001

Stochastic Outlier Selection

Jeroen Janssens[*] **Ferenc Huszár**[◇]

Eric Postma[*] **Jaap van den Herik**[*]

∗ TiCC, Tilburg University, NL
◇ CBL, University of Cambridge, UK

Abstract

In this report we propose a novel, unsupervised algorithm for classifying data points as outliers. The algorithm is called Stochastic Outlier Selection (SOS). It applies the concept of affinity to the problem of outlier selection. We explain and motivate the use of affinity in Section 1. How the SOS algorithm selects outliers is described in Section 2. Section 3 presents four related unsupervised outlier-selection algorithms: K-Nearest Neighbour Data Description (KNNDD), Local Outlier Factor (LOF), Local Correlation Integral (LOCI), and Least Squares Outlier Detection (LSOD). Using outlier-score plots, we illustrate and discuss the qualitative performances of SOS and the four algorithms. In Section 4, we evaluate all five algorithms on eighteen real-world data sets, and by the Neményi statistical test we show that SOS performs significantly better. Moreover, seven synthetic data sets are used to gain insight into the behaviour of the algorithms. From our experiments we may conclude that SOS is more robust against data perturbations and varying densities than the other four algorithms. The results are discussed in Section 5. Finally, we give our conclusions in Section 6.

Stochastic Outlier Selection

Jeroen Janssens[*] **Ferenc Huszár**[◇] **Eric Postma**[*]

Jaap van den Herik[*]

∗ TiCC, Tilburg University, NL

◇ CBL, University of Cambridge, UK

1 An affinity-based approach to outlier selection

The objective is to classify automatically those data points as outliers that are labelled as anomalous by the expert. Each algorithm approaches this objective in a different way. The Stochastic Outlier Selection (SOS) algorithm, which is the topic of this report, selects outliers using an affinity-based approach.

The general idea of SOS is as follows. SOS employs the concept of affinity to quantify the relationship from one data point to another data point. Affinity is proportional to the similarity between two data points. So, a data point has little affinity with a dissimilar data point. A data point is selected as an outlier when all the other data points have insufficient affinity with it. The concept of affinity and the SOS algorithm are explained more precisely in the next section. First, in Subsection 1.1, we mention two problems to which affinity has been successfully applied. As a result of employing affinity, SOS computes outlier probabilities instead of outlier scores. The advantages of computing probabilities with respect to scores are discussed in Subsection 1.2.

1.1 Two successful applications of affinity

So far, affinity has been applied successfully to at least two other problems: (1) clustering and (2) dimensionality reduction. They will be discussed briefly below. To the best of our knowledge, SOS is the first algorithm that applies the concept of affinity to the problem of outlier selection.

First, affinity has been successfully applied to the problem of clustering. The goal of clustering is to select a (given) number of data points that serve as the representatives of the clusters (i.e., cluster exemplars) in a data set. The clustering algorithm 'Affinity Propagation' by Frey and Dueck (2007) updates iteratively the affinity that a data point has to a potential cluster exemplar by passing messages. In other words, the clustering algorithm employs affinity to quantify the relationships among data points.

Second, affinity has been successfully applied to the problem of dimension reduction. The goal of dimension reduction is to map a high-dimensional data set onto a low-dimensional space (Roweis and Saul, 2000; Tenenbaum, de Silva, and Langford, 2000). The challenge is to preserve the structure of the data set as much as possible. Several algorithms address this challenge successfully by concentrating on preserving the local relationships using affinity (Hinton and Roweis, 2003; van der Maaten and Hinton, 2008; van der Maaten, 2009). Again, as with clustering, affinity is used to quantify the relationships among data points.

Because of these two successful applications of affinity, we expect that affinity is also beneficial to outlier selection and in particular to the SOS algorithm. Here we note that affinity is calculated differently in the two applications. Our definition of affinity (which is presented in

Subsection 2.2) is based on the definition found in the works concerning dimension reduction because that allows us to compute outlier probabilities.

1.2 Outlier probabilities instead of scores

Current outlier-selection algorithms typically compute unbounded outlier scores (see Gao and Tan, 2006). The scores computed by such algorithms differ widely in their scale, range, and meaning. Moreover, the scores may also differ from data set to data set for the same algorithm (Kriegel, Kröger, Schubert, and Zimek, 2009).

SOS computes outlier *probabilities*, i.e., the probability that a data point is an outlier. Because an outlier probability is a number between 0 and 1, both the minimum and the maximum value of outlier probabilities are consistent from data set to data set. We state three reasons why outlier probabilities are favourable to outlier scores. The first reason is that outlier probabilities are easier to interpret by an expert than outlier scores (Kriegel et al., 2009). The second reason is that outlier probabilities allow to select an appropriate threshold for outlier selection (i.e., classification) using a Bayesian risk model (Gao and Tan, 2006). A Bayesian risk model takes into account the relative cost of misclassifications. Employing a Bayesian risk model is not possible with unbounded outlier scores. The third reason is that outlier probabilities provide a more robust approach for developing an ensemble outlier selection framework out of individual outlier-selection algorithms than unbounded outlier scores (Kriegel, Kröger, Schubert, and Zimek, 2011).

Whereas Gao and Tan (2006) and Kriegel et al. (2011) suggest converting the unbounded outlier scores from existing algorithms into calibrated probabilities, SOS computes the outlier probabilities directly from the data.

2 The Stochastic Outlier Selection algorithm

In this section we describe how SOS selects outliers. Stated more formally we describe how the outlier-selection algorithm f_{SOS} maps data points to the classifications 'outlier' and 'inlier'.

The data that is used by SOS, i.e., the input data, the intermediate data, and the output data, can be represented as five matrices. Figure 1 shows the five matrices and summarises SOS as a series of matrix transformations. The numbers above the arrow denote the subsections that discuss the matrix transformations. Next to each matrix we find a colour bar that maps a range of values to a range of colours. In the case of matrix $\mathbf{X}$, for example, 0 is mapped to white and 8 is mapped to dark blue. As such, Figure 1 may serve as an overview reference throughout our description of SOS.

Subsections 2.1—2.6 are concerned with the outlier scoring part φ_{SOS} that maps data points to outlier probabilities. We briefly mention the purpose of each subsection. In Subsection 2.1, we discuss the input data and the dissimilarity between data points (i.e., input matrix $\mathbf{X}$ and dissimilarity matrix $\mathbf{D}$). In Subsection 2.2, we explain how dissimilarities are transformed into affinities (i.e., affinity matrix $\mathbf{A}$). In Subsection 2.3 we continue our description by using graph theory, and generate stochastic neighbour graphs that are based on binding probabilities (i.e., binding matrix $\mathbf{B}$). We present three ways of computing the outlier probabilities (i.e., output matrix $\mathbf{\Phi}$). In Subsection 2.4 we show that by sampling stochastic neighbour graphs, we can estimate outlier probabilities. Through marginalisation we can compute exactly the outlier probabilities as is described in Subsection 2.5. With the help of probability theory, we observe in Subsection 2.6 that the outlier probabilities can also be computed in closed form.

In Subsection 2.7, the outlier scoring part φ_{SOS} is transformed into the outlier-selection algorithm f_{SOS}, so that SOS can classify data points as outliers. Finally, in Subsection 2.8, we explain in detail the concept of perplexity that allows SOS to create soft neighbourhoods.

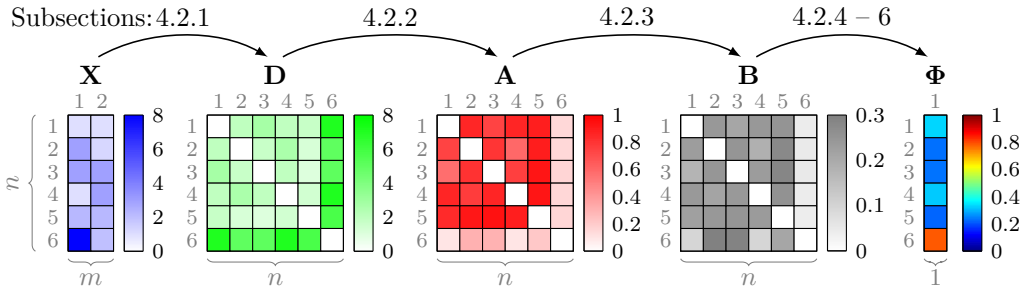

Figure 1: From input to output in five matrices: (1) the input matrix $\mathbf{X}$ containing the feature values of the data points, (2) the dissimilarity matrix $\mathbf{D}$, (3) the affinity matrix $\mathbf{A}$, (4) the binding probability matrix $\mathbf{B}$, and (5) the output matrix $\mathbf{\Phi}$ containing the outlier probabilities. The transformations from one matrix to the next matrix are explained in the subsections stated above the arrows.

2.1 Input to the algorithm

SOS is an unsupervised outlier-selection algorithm. Therefore, SOS requires as input an unlabelled data set $\mathcal{D}$, only. An unlabelled data set means that the labels 'anomaly' and 'normality' are unavailable. Therefore, SOS does not know whether a domain expert considers the real-world observations (that correspond to the data points) to be anomalous or normal. Nevertheless, SOS is able to classify the data points in the data set either as 'outlier' or 'inlier'.

The number of data points in the unlabelled data set $\mathcal{D}$ is denoted by n. In our description of SOS, each data point is represented by an m-dimensional, real-valued, feature vector $\mathbf{x} = [x_1, \ldots, x_m] \in \mathbb{R}^m$. As such, a data point can be regarded as a point in an m-dimensional Euclidean space. The data set $\mathcal{D}$ is represented by a matrix $\mathbf{X}$ of size $n \times m$, i.e., number of data points × number of features. The vector $\mathbf{x}_i$ denotes the i^{th} row of the matrix $\mathbf{X}$. In our description of SOS we do not distinguish between the data set $\mathcal{D}$ and the corresponding matrix $\mathbf{X}$, and often refer to the data set by $\mathbf{X}$ for readability.

Figure 2 shows the example data set that we use throughout our description of SOS. The data set contains six data points, and each data point has two features. The corresponding two-dimensional points are plotted on the left side of Figure 2. On the right side of the figure, the same data set is represented by matrix $\mathbf{X}$. We recall that the colour of the cells in the matrix $\mathbf{X}$ correspond to the feature values of the data points.

The features of the data points are used to measure the *dissimilarity* between pairs of data points. (Dissimilarity forms the basis for affinity, see the next subsection.) The dissimilarity between data point $\mathbf{x}_i$ and data point $\mathbf{x}_j$ is a non-negative scalar that is computed by a dissimilarity measure d. In our description and in our experiments we employ the Euclidean distance

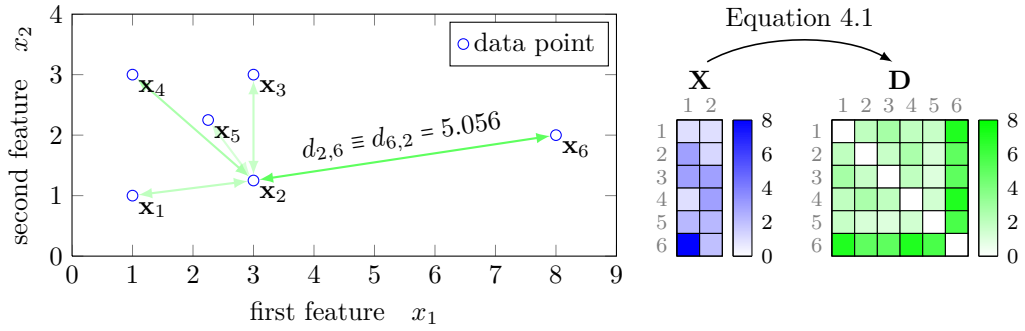

Figure 2: The example data set used for our SOS description. Each data point has two features and is a point in two-dimensional Euclidean space. The dissimilarity $d_{2,6}$ (and $d_{6,2}$) is the Euclidean distance between data points $\mathbf{x}_2$ and $\mathbf{x}_6$ (see Equation 1).

as the dissimilarity measure between pairs of data points. Let

$$d_{ij} = \sqrt{\sum_{k=1}^{m} \left(x_{jk} - x_{ik}\right)^2} \ , \tag{1}$$

where x_{ik} denotes the k^{th} feature value of i^{th} data point, i.e., cell i, k of matrix $\mathbf{X}$. From Equation 1 it follows (1) that our employed dissimilarity measure is symmetric, i.e., $d_{ij} \equiv d_{ji}$, and (2) that the dissimilarity between a data point and itself is zero, i.e., $d_{ii} = 0$. The data points on the left side of Figure 2 are connected by green lines with varying brightnesses. Both the length and the brightnesses of these green lines illustrate the dissimilarity between data point $\mathbf{x}_2$ and the other five data points. The right side of Figure 2 shows the dissimilarity matrix $\mathbf{D}$ that is obtained by applying Equation 1 to each pair of data points in the matrix $\mathbf{X}$. (The bold, upright letter '$\mathbf{D}$' should not be confused with the calligraphic letter '$\mathcal{D}$' that denotes a data set.) The brightnesses of the green lines are equal to the brightnesses of the cells in the second row of $\mathbf{D}$. In fact, because the Euclidean distance d is symmetric, the resulting dissimilarity matrix $\mathbf{D}$ is symmetric, meaning that the rows are equal to the columns. Therefore, the brightnesses of the green lines are also equal to the brightnesses of the cells of the second column.

In the next subsection, dissimilarities are used to compute *affinities* between data points. In other words, the dissimilarity matrix $\mathbf{D}$ is transformed into the affinity matrix $\mathbf{A}$.

2.2 Transforming dissimilarity into affinity

As mentioned in Section 1, we employ affinity in order to quantify the relationship from one data point to another data point. Our definition of affinity is based on the definitions used for the problem of dimension reduction (Hinton and Roweis, 2003; Goldberger, Roweis, Hinton, and Salakhutdinov, 2005; van der Maaten and Hinton, 2008).

Definition 1 (Affinity)**.** *Let d_{ij} denote the dissimilarity that data point $\mathbf{x}_j$ has to data point $\mathbf{x}_i$. Then the affinity that data point $\mathbf{x}_i$ has with data point $\mathbf{x}_j$ is given by*

$$a_{ij} = \begin{cases} \exp\left(-d_{ij}^2 \ / \ 2\sigma_i^2\right) & \text{if} \ \ i \neq j \\ 0 & \text{if} \ \ i = j \ , \end{cases} \tag{2}$$

where σ_i^2, known as the variance, is a scalar associated with data point $\mathbf{x}_i$.

We note that (1) the affinity that data point $\mathbf{x}_i$ has with data point $\mathbf{x}_j$ decays Gaussian-like with respect to the dissimilarity d_{ij}, and (2) a data point has no affinity with itself, i.e., $a_{ii} = 0$ (the justification is discussed below).

In words, Equation 2 states that the affinity that data point $\mathbf{x}_i$ has with data point $\mathbf{x}_j$ is proportional to the probability density at $\mathbf{x}_j$ under a Gaussian distribution that has mean $\mathbf{x}_i$ and variance σ_i^2. A graphical comparison of three variance values for Equation 2 is provided in Figure 3. A lower variance causes the affinity to decay faster. The higher a variance is, the less affinity is influenced by the dissimilarity. As an extreme example, an infinitely high variance yields an affinity of 1, no matter how high the dissimilarity is (because $\mathrm{e}^0 = 1$). Stated differently, the Gaussian distribution becomes a uniform distribution.

SOS has one parameter only, which is called the perplexity parameter and is denoted by h. The perplexity parameter h can be compared with the parameter k as in k-nearest neighbours, with two important differences. First, because affinity decays smoothly, 'being a neighbour' is not a binary property, but a smooth property. In fact, in the next subsection we formalise 'being a neighbour' into a probabilistic property using Stochastic Neighbour Graphs. Second, unlike the parameter k, perplexity is not restricted to be an integer, but can be any real number between 1 and $n-1$. Stated differently, a data point can have a minimum of 1 and a maximum of $n-1$ effective neighbours. Perplexity may therefore be interpreted as a smooth measure for

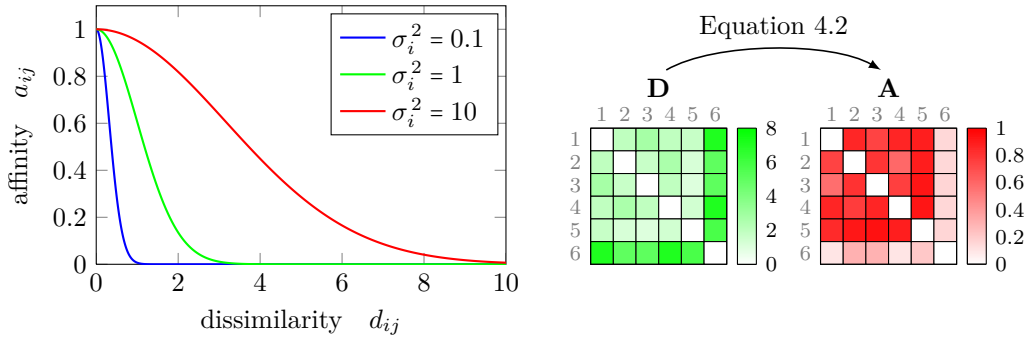

Figure 3: From dissimilarity to affinity. **Left:** Graphs of the affinity a_{ij} with respect to the dissimilarity d_{ij} as defined by Equation 2, for three values of the variance σ_i^2. **Right:** The affinity matrix **A** is obtained by applying Equation 2 to each cell in the dissimilarity matrix **D**.

the effective number of neighbours. Because a data point has no affinity with itself, i.e., $a_{ii} = 0$, it is never its own neighbour, which implies (as we will see later) that only *other* data points have influence on its outlier probability.

The value of each variance σ_i^2 is determined using an adaptive approach such that each data point has the same number of effective neighbours, i.e., h. The adaptive approach yields a different variance for each data point, causing the affinity to be asymmetric (Hinton and Roweis, 2003). To be precise, the affinities a_{ij} and a_{ji} are equal only when (1) the dissimilarities d_{ij} and d_{ji} are equal, which is always the case with the Euclidean distance but need not be the case with other dissimilarity measures, and (2) the variances σ_i^2 and σ_j^2 are equal, which is rarely the case unless the two data points are equally dissimilar to their own neighbours. As a counterexample to asymmetric affinity we mention the dimensionality reduction algorithm by van der Maaten and Hinton (2008), which symmetrises purposefully the affinities between data points (i.e., $a_{ij} = \frac{1}{2}a_{ij} + \frac{1}{2}a_{ji}$), such that dissimilar data points are not isolated but joined with one of the clusters in the low-dimensional mapping. The purpose of SOS, however, is to classify these dissimilar data points as outliers, and, therefore, does not symmetrise the affinities. We elaborate upon the details of assigning the variances with the adaptive approach in Subsection 2.8.

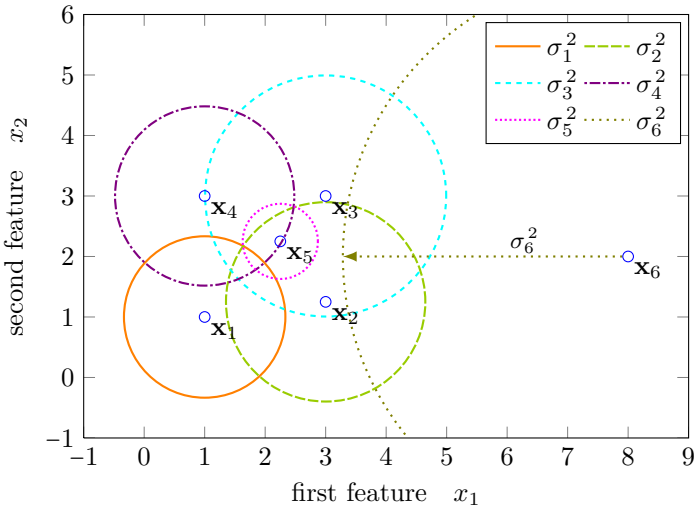

Figure 4: The radii of the circles correspond to the variance for the data points. The variance σ_i^2 adapts to the density of the data and is determined for each data point $\mathbf{x}_i$ separately, such that each data point has the same number of effective neighbours, i.e., perplexity. In this figure the perplexity is set to 3.5. The variance influences the amount of affinity that a data point has to the other data points (see Figure 3).

Figure 4 shows the variances for the data points in the example data set. The radii of the six circles correspond to the variances for the six data points (and please note that the circles do not represent a boundary). In all figures and calculations concerning the example data set, the perplexity h is set to 4.5, with Figure 4 being the only exception to this. In Figure 4, the perplexity h is set to 3.5 because otherwise, the radii of the six circles would be to be too large to create an illustrative figure (see Figure 12 for the variances that correspond to all possible settings of the perplexity parameter). Figure 4 shows, for example, that for data point $\mathbf{x}_6$ to have 3.5 effective neighbours, the variance σ_6^2 must be set higher than the other variances.

Using the dissimilarity matrix $\mathbf{D}$ and Equation 2, we compute the affinity that each data point has to every other data point, resulting in the affinity matrix $\mathbf{A}$ (illustrated in the right side of Figure 3). The affinity matrix $\mathbf{A}$ is in two aspects similar to the dissimilarity matrix $\mathbf{D}$, (1) the size is $n \times n$ and (2) the diagonal is 0, because data points have no affinity with themselves. However, unlike $\mathbf{D}$, $\mathbf{A}$ is not symmetric, because the affinity between two data points is not symmetric. The i^{th} row of the affinity matrix, denoted by $\mathbf{a}_i$, is called the *affinity distribution* of data point $\mathbf{x}_i$. Let

$$\mathbf{a}_i = [a_{i,1}, \ldots, a_{i,n}] \ , \tag{3}$$

where a_{ii} is 0. We note that the affinity distribution is not a probability distribution because it does not sum to 1. In the next subsection we continue with the affinity matrix $\mathbf{A}$ to generate a Stochastic Neighbour Graph.

2.3 Stochastic neighbour graphs based on affinities

In the remainder of our description of SOS, we employ graph theory, because (1) it provides a solid mathematical framework to perform calculations with affinities, and (2) it allows us to derive outlier probabilities. To model explicitly the data points and their relationships (i.e., affinities) using vertices and directed edges, we generate a Stochastic Neighbour Graph (SNG). The set of vertices $\mathcal{V}$ is associated with the data set $\mathbf{X}$. So, each vertex v_i corresponds to data point $\mathbf{x}_i$. Generating the directed edges between the vertices depends on *binding probabilities*. Therefore, we first introduce the concept of binding probabilities and subsequently define the generative process for an SNG. Finally, we introduce the binary property of being an outlier given one SNG.

2.3.1 Binding probabilities

The binding probability b_{ij} is the probability that vertex v_i binds to vertex v_j, i.e., the probability of generating a directed edge from v_i to v_j. We denote a directed edge from v_i to v_j by '$i \to j$'. The binding probability b_{ij} is proportional (denoted by $\propto$) to the affinity that data point $\mathbf{x}_i$ has with data point $\mathbf{x}_j$.

$$b_{ij} \equiv p\left(i \to j \in \mathcal{E}_G\right) \propto a_{ij} \ , \tag{4}$$

which is equal to the affinity a_{ij} normalised, such that $\sum_{k=1}^{n} b_{ik}$ sums to 1.

$$b_{ij} = \frac{a_{ij}}{\sum_{k=1}^{n} a_{ik}} \ . \tag{5}$$

We note that b_{ii} is always 0, because a_{ii} is always 0 (see Equation 2). By applying Equation 5 to every cell in the affinity matrix $\mathbf{A}$, we obtain the binding matrix $\mathbf{B}$. The binding probabilities for one vertex v_i form together a *binding distribution*, which is a discrete probability distribution

$$\mathbf{b}_i = [b_{i,1}, \ldots, b_{i,n}] . \tag{6}$$

Similar to the affinity distribution $\mathbf{a}_i$ from Equation 3, the binding distribution $\mathbf{b}_i$ is the i^{th} row of the binding matrix $\mathbf{B}$.

We can illustrate the binding probabilities by representing the data set as a graph, where each data point $\mathbf{x}_i$ is associated with a vertex v_i. Because each vertex has *some* probability

of binding to any other vertex, the graph is fully connected. Figure 5 shows this graph four times (these are not yet Stochastic Neighbour Graphs, those are shown later, in Figure 6). The brightness of each directed edge $i \to j$ is determined by the binding probability b_{ij}, as can be seen on the right side of Figure 5, where the binding matrix $\mathbf{B}$ is depicted four times.

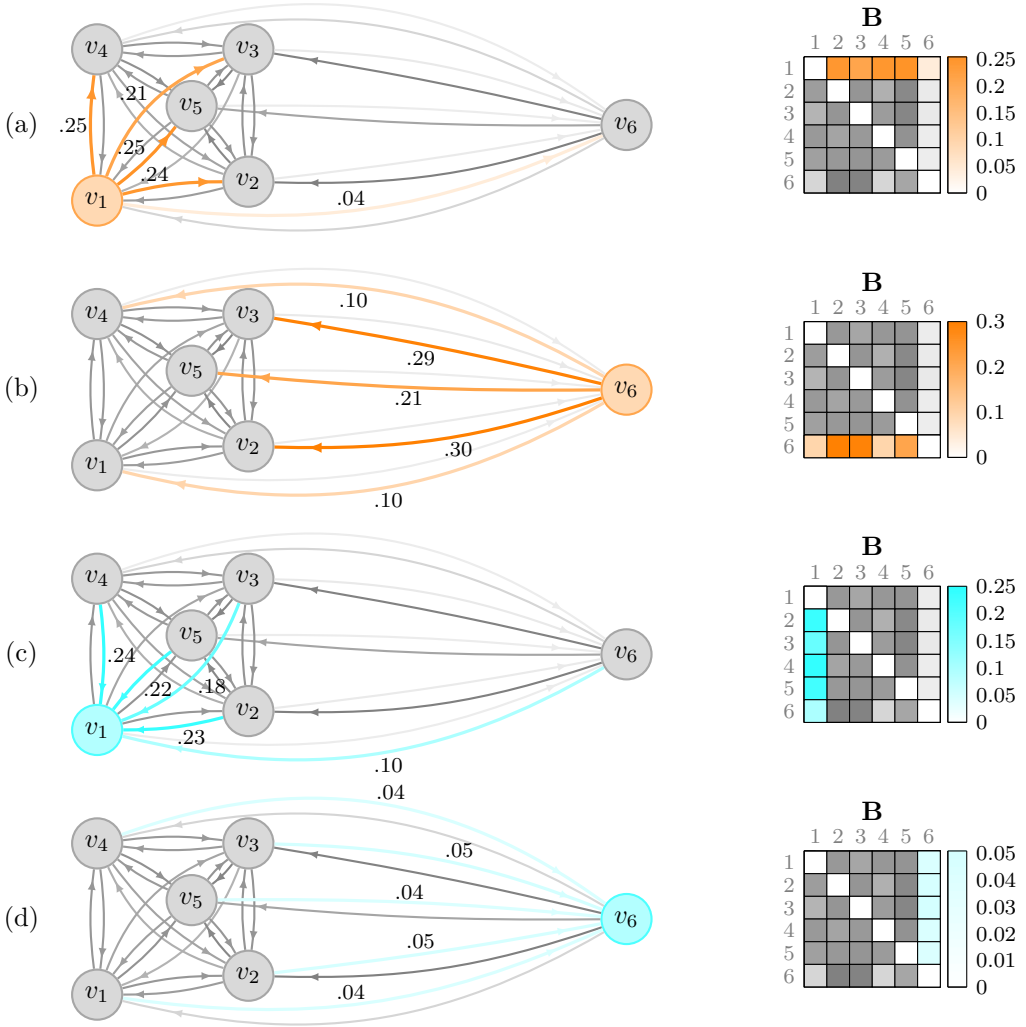

Figure 5: Illustration of binding probabilities. **Left:** Each vertex v_i in the graph is associated with a binding probability distribution $\mathbf{b}_i$. **Right:** The binding matrix $\mathbf{B}$. A darker cell in the matrix corresponds to a higher binding probability.

The same graph is shown four times, i.e., graph (a) to graph (d), where each graph illustrates a different aspect. In graph (a), five directed edges are coloured orange. The brightness of an orange edge is associated with the probability that vertex v_1 binds to another vertex. The corresponding binding distribution $\mathbf{b}_1$ sums to 1. The associated row in the binding matrix $\mathbf{B}$, i.e., the first row, is also coloured orange. From the graph we can see, for example, that the probability that v_1 binds to v_6 is relatively low ($\approx .04$). If we reconsider Figure 1 on page 3, then we can see that the low binding probability $b_{1,6}$ is due to the low affinity $a_{1,6}$, which is due to the high dissimilarity $d_{1,6}$, which is due mostly to the large difference in the values of the first feature of data points $\mathbf{x}_1$ and $\mathbf{x}_6$. In graph (b), the five edges from vertex v_6 are coloured orange. We can see that the binding distribution $\mathbf{b}_6$ is distributed more evenly than $\mathbf{b}_1$, because from the perspective of data point $\mathbf{x}_6$, the other five data points are roughly equally distant. In both graph (c) and graph (d), five edges are coloured cyan, to illustrate the probability that *other* vertices bind to vertex v_1 and vertex v_6, respectively. These binding probabilities do not necessarily sum to 1. From graph (d), (and similarly from the sixth column in the binding matrix $\mathbf{B}$), we can see that all vertices have a low probability to bind to vertex v_6. Please note that these four graphs only serve to illustrate the concept of binding probabilities, and will not

be used to compute outlier probabilities.

2.3.2 Generating a stochastic neighbour graph

After the introduction of the binding probabilities, we define formally the Stochastic Neighbour Graph (SNG) and discuss the generative process for an SNG. We do so by two assumptions and four subsequent consequences. After that we describe the generative process of the set of directed edges $\mathcal{E}_G$.

Definition 2 (Stochastic neighbour graph)**.** *A Stochastic Neighbour Graph (SNG) is a graph $G = (\mathcal{V}, \mathcal{E}_G)$ with a set of vertices $\mathcal{V}$ and a set of directed edges $\mathcal{E}_G$. Let matrix $\mathbf{X}$ be a data set containing n data points. For each data point $\mathbf{x}_i \in \mathbf{X}$, a vertex v_i is added to $\mathcal{V}$.*

Let d be the dissimilarity measure (e.g., the Euclidean distance as stated by Equation 1). Let h be the perplexity parameter, where $h \in [1, n-1]$. Then matrix $\mathbf{D}$ is the dissimilarity matrix of size $n \times n$ that is obtained by applying dissimilarity measure d on all pairs of data points in data set $\mathbf{X}$. Then $\{\sigma_1^2, \ldots, \sigma_n^2\}$ are n variances, whose values are determined in Subsection 2.8 using perplexity h. Then matrix $\mathbf{A}$ is the affinity matrix of size $n \times n$ as obtained by applying Equation 2 with the n variances to each cell of the dissimilarity matrix $\mathbf{D}$. Then matrix $\mathbf{B}$ is the binding matrix of size $n \times n$ as obtained by applying Equation 5 to each cell of the affinity matrix $\mathbf{A}$.

The set of directed edges $\mathcal{E}_G$ is generated by the following stochastic binding procedure. Let $\mathbf{b}_i$ be the binding distribution of vertex v_i, which is the i^{th} row of the binding matrix $\mathbf{B}$. Let each vertex $v_i \in \mathcal{V}$ bind independently to one other vertex v_j, where the index j is an integer sampled from the binding distribution $\mathbf{b}_i$. Let $i \to j$ denote a directed edge from vertex v_i to vertex v_j. The directed edge $i \to j$ is added to $\mathcal{E}_G$ if vertex v_i binds to vertex v_j.

Below we discuss seven properties of an SNG that are implied by Definition 2 and the two generative procedures. We illustrate our description by Figure 6.

Figure 6 shows three possible SNGs for the example data set with the binding matrix $\mathbf{B}$. (The right side of the figure is introduced below.)

First, an SNG always has n directed edges because there are n vertices that each connect to one vertex. Second, although the vertices of an SNG are fixed given a data set $\mathbf{X}$, each generated SNG may be different because the directed edges are generated using a stochastic binding procedure. Third, each binding distribution $\mathbf{b}_i$ is based on the affinity distribution $\mathbf{a}_i$, so an SNG reflects the relationship between data points. Fourth, an SNG has no self-loops, i.e., no vertex binds to itself because b_{ii} is 0, and as a consequence the index j that is sampled from the binding distribution will never be equal to i. Fifth, the vertices bind independently, which means that vertices do not influence each other's binding process. Sixth, each vertex v_i has an out-degree of 1, denoted by $\deg_G^+(v_i) = 1$, because each vertex binds to one other vertex. Seventh, because the binding process is stochastic, more than one vertex may bind to the same vertex. For example, both vertices v_1 and v_2 may bind to vertex v_3. Vertex v_3 now has an in-degree of 2, denoted by $\deg_G^-(v_3) = 2$. This implies that there is (at least) one vertex in the graph that no vertex binds to and thus has an in-degree of 0. In Figure 6 we can see, for instance, that in both graphs G_a and G_b, vertex v_6 has an in-degree of 0. In graph G_c, vertex v_6 has an in-degree of 3. It is possible (albeit improbable) to generate this graph, because each vertex (except vertex v_6 itself) has some probability of binding to vertex v_6. In the next subsection we elaborate on the probability of constructing a particular SNG.

2.3.3 Being an outlier given one SNG

If there is a directed edge from v_i to v_j, i.e., $i \to j \in \mathcal{E}_G$, then we say that $\mathbf{x}_j$ is a neighbour of $\mathbf{x}_i$. In other words, data point $\mathbf{x}_i$ chooses data point $\mathbf{x}_j$ as a neighbour. If there is no directed edge from v_j *back* to v_i, i.e., $j \to i \notin \mathcal{E}_G$, then $\mathbf{x}_j$ is not a neighbour of $\mathbf{x}_i$.

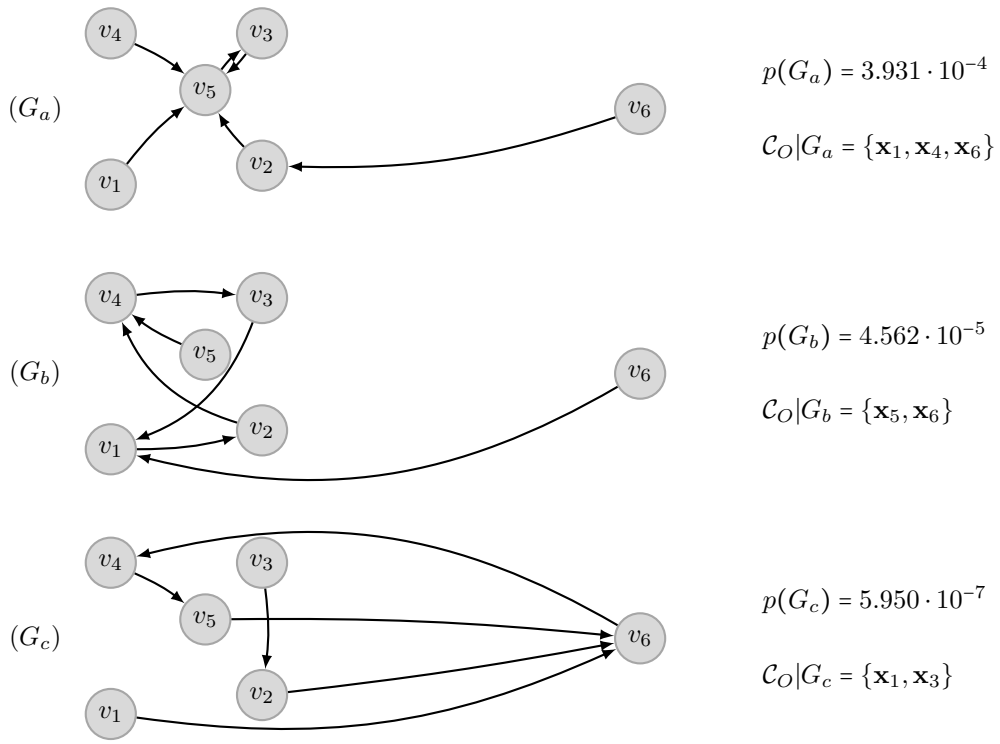

Figure 6: Three stochastic neighbour graphs (SNGs) generated with perplexity h set to 4.5. **Left:** The three SNGs G_a, G_b, and G_c are sampled from the discrete probability distribution $P(\mathcal{G})$. In Figure 7 it is indicated where the three SNGs are in the set of all graphs $\mathcal{G}$. **Right:** The probability $p(G_i)$ is the probability that graph G_i is generated (see Equation 10). The outlier class $\mathcal{C}_\mathrm{O}|G_i$ contains the data points that are outliers given graph G_i (see Equation 7).

We define that a data point is an outlier in graph G, when the data point has no neighbours in graph G. In graphical terms, a data point $\mathbf{x}_i$ belongs to the outlier class $\mathcal{C}_\mathrm{O}$ if its corresponding vertex v_i has no inbound edges in graph G, i.e., its in-degree $\deg^-_G$ is zero:

$$\mathcal{C}_\mathrm{O}\,|\,G = \{\mathbf{x}_i \in \mathbf{X} \mid \deg^-_G(v_i) = 0\} \ . \tag{7}$$

In words, there is no vertex v_j that binds to vertex v_i,

$$\mathcal{C}_\mathrm{O}\,|\,G = \{\mathbf{x}_i \in \mathbf{X} \mid \nexists v_j \in \mathcal{V} : j \to i \in \mathcal{E}_G\} \ , \tag{8}$$

or, similarly, all vertices v_j do not bind to vertex v_i,

$$\mathcal{C}_\mathrm{O}\,|\,G = \{\mathbf{x}_i \in \mathbf{X} \mid \forall v_j \in \mathcal{V} : j \to i \notin \mathcal{E}_G\} \ . \tag{9}$$

The right side of Figure 6, lists, for the three SNGs, the data points that belong to the outlier class $\mathcal{C}_\mathrm{O}\,|\,G$.

At this moment, being an outlier is a *binary* property; given one particular SNG G, a data point is either a member of the outlier class or not. The next subsection explains that by generating many SNGs, we can estimate the outlier *probability* .

2.4 Estimating outlier probabilities through sampling

In the previous subsection the outlier class $\mathcal{C}_\mathrm{O}$ is established deterministically given one particular SNG G. But because G itself is generated stochastically (using the binding probabilities), the outlier class $\mathcal{C}_\mathrm{O}$ becomes a stochastic subset of the data set. Therefore, data points are currently randomly selected as outlier, which is an obstacle.

We alleviate this obstacle by taking not one, but all possible SNGs into account. For a data set containing n data points, there are $(n-1)^n$ binding combinations possible, because each of the n vertices binds independently to one of $n-1$ vertices. We denote the set of all $(n-1)^n$ possible graphs by $\mathcal{G}$. For our example data set, the set $\mathcal{G}$ contains $5^6 = 15{,}625$ SNGs.

Because the binding probabilities are not distributed uniformly, certain edges are more probable to be generated than other edges. For instance, in our example data set, the edge $2 \to 1$ is more probable than the edge $2 \to 6$. As a consequence, certain SNGs are more probable to be generated than others. Since the vertices $\mathcal{V}$ are constant, the probability of generating a certain SNG G depends only on the binding probabilities.

$$p(G) = \prod_{i \to j \in \mathcal{E}_G} b_{ij} \ . \tag{10}$$

The sampling probabilities of the three SNGs are listed on the right side of Figure 6.

The set of all graphs $\mathcal{G}$ is thus associated with a discrete probability distribution $P(\mathcal{G})$. To sample an SNG from the probability distribution, denoted by $G \sim P(\mathcal{G})$, means to generate an SNG. Figure 7 shows the probability mass and the cumulative probability mass for $\mathcal{G}$, for three values of the perplexity h. A lower perplexity (e.g., $h = 4.0$, blue line) yields less uniform binding distributions, and consequently leads to more variation in the probabilities by which SNGs are sampled. Figure 7 is annotated by three arrows pointing to the red line. These three arrows indicate the 'positions' of the three SNGs of Figure 6 in $P(\mathcal{G})$. We can see that G_a is the most probable SNG to be generated, because each data point chooses as its neighbour the data point to which it has the most affinity. We can also see that G_c is one of the least probable SNGs.

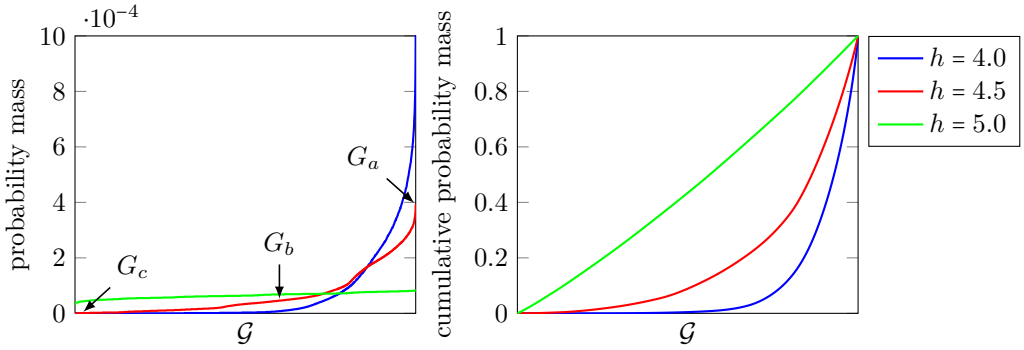

Figure 7: Discrete probability distribution for the set of all SNGs. **Left:** The probability mass for the discrete probability distribution for the set of all Stochastic Neighbour Graphs (SNGs) $\mathcal{G}$ for the example data set, for three values of the perplexity h. For this plot, the graphs in the set are ordered ascendingly by their probability of being generated. The annotations G_a, G_b, and G_c correspond to the probability of the three SNGs shown in Figure 6. **Right:** The cumulative probability mass for $\mathcal{G}$. The high sensitivity to the perplexity is due to the small number of data points in the example data set.

We are now ready to use a sampling procedure to estimate the probability that a data point belongs to the outlier class. Given a number of sampled SNGs G, we compute the *relative frequency* that the data point belongs to the outlier class. As the number of samples S approaches infinity, the relative frequency converges to the outlier probability.

$$p(\mathbf{x}_i \in \mathcal{C}_\mathrm{O}) = \lim_{S \to \infty} \frac{1}{S} \sum_{s=1}^{S} \mathbb{1}\left\{ \mathbf{x}_i \in \mathcal{C}_\mathrm{O} \,|\, G^{(s)} \right\} \quad , \quad G^{(s)} \sim P(\mathcal{G}) \ , \tag{11}$$

where $\mathbb{1}\{\cdot\}$ is an indicator random variable that has a value of 1 if data point $\mathbf{x}_i$ belongs to the outlier class $\mathcal{C}_\mathrm{O}$ given graph G, and 0 otherwise.

The sampling procedure implies that if a particular data point belongs to the outlier class for all possible graphs, then its outlier probability is 1 (since the sum of the probabilities of

all possible graphs is 1, i.e., $\sum_{G \in \mathcal{G}} p(G) = 1$). Similarly, if a data point is a member of the outlier class for 30% of the sampled graphs, then its outlier probability is 0.3. We note that the sampling procedure leads to an outlier probability that is a probability from a Frequentist point of view (Bayarri and Berger, 2004), and that it is not obtained by normalising an unbounded outlier score as in, for example, Gao and Tan (2006) and Kriegel et al. (2011).

The outlier probabilities during the first $1 \cdot 10^6$ iterations of one run of the sampling procedure are plotted in Figure 8. The figure reveals that when the perplexity is set to 4.5, the estimated outlier probabilities of data points $\mathbf{x}_1$ to $\mathbf{x}_6$ of the example data set, after $1 \cdot 10^6$ samples, are: 0.34, 0.23, 0.24, 0.32, 0.22, and 0.79, respectively. Because sampling SNGs is a stochastic process, each run produces in the beginning (say, the first 10 iterations) different outlier probabilities. Eventually (say, after 10,000 iterations), all runs produce outlier probabilities that converge to the same values.

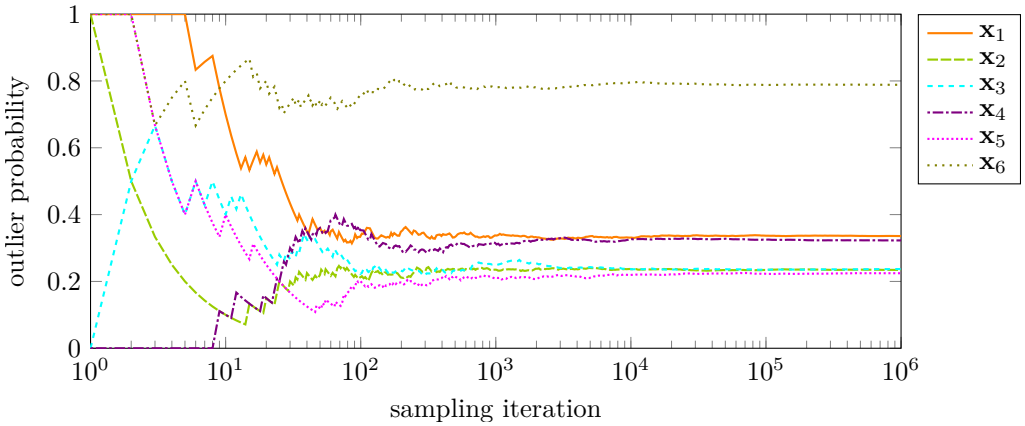

Figure 8: Convergence of the outlier probabilities by repeatedly sampling SNGs. The plot shows the first 1,000,000 sampling iterations of one run.

2.5 Computing outlier probabilities through marginalisation

The relative frequency computed by Equation 11 only converges to the outlier probability when the number of samples approaches infinity. It turns out that we can compute the outlier probability exactly, by enumerating once over all possible SNGs. If we state the enumeration procedure in more technical terms, then we say that we compute the marginal probability of any particular data point being an outlier, by marginalising out the stochastic graph G. Because one SNG is more probable than the other (due to the binding probabilities), it is important to take this into account as well.

$$p(\mathbf{x}_i \in \mathcal{C}_O) = \sum_{G \in \mathcal{G}} 1\{\mathbf{x}_i \in \mathcal{C}_O \,|\, G\} \cdot p(G) \tag{12}$$

$$= \sum_{G \in \mathcal{G}} 1\{\mathbf{x}_i \in \mathcal{C}_O \,|\, G\} \cdot \prod_{q \to r \in \mathcal{E}_G} b_{qr} \ . \tag{13}$$

The exact outlier probabilities for the data points in the example data set, as computed by marginalisation, are: $0.335, 0.235, 0.237, 0.323, 0.224$, and 0.788, respectively.

So, instead of *estimating* the outlier probabilities by sampling, say, 1,000,000 SNGs, we can *compute exactly* the outlier probabilities by marginalising over 15,625 SNGs. However, such a gain holds only for small data sets, such as our example data set. To illustrate how quickly the size of $\mathcal{G}$, i.e., $|\mathcal{G}|$, grows with respect to n, consider $n = 5$, $n = 10$, and $n = 100$. These data set sizes correspond to $|\mathcal{G}| = 1024$, $|\mathcal{G}| = 3.5 \cdot 10^9$, and $|\mathcal{G}| = 3.7 \cdot 10^{199}$, respectively. So, even small data sets lead to a combinatorial explosion, making Equation 12 intractable to compute. In the next subsection we present a way to avoid this problem.

2.6 Computing outlier probabilities in closed form

Because each vertex binds to exactly one other vertex, the outlier probability can be computed in closed form, without actually enumerating all the SNGs in $\mathcal{G}$. Here we note that if a vertex would have been allowed to bind to *multiple* vertices, the outlier probability could not be computed in closed form.

We observe that the probability that data point $\mathbf{x}_i$ belongs to the outlier class, is equal to the probability that its in-degree is zero, i.e., the probability that none of the other vertices bind to vertex v_i. Without using graph-theoretical terms, the outlier probability of data point $\mathbf{x}_i$ can be reformulated as the joint probability that data point $\mathbf{x}_i$ is never chosen as a neighbour by the other data points. As a consequence, we can compute the outlier probabilities directly, without generating any SNG.

Theorem 1 (Outlier probability)**.** *Let* $\mathbf{X}$ *be a data set containing* n *data points. Let* $\mathcal{C}_\mathrm{O}$ *denote the outlier class. If* a_{ij} *is the affinity that data point* $\mathbf{x}_i$ *has with data point* $\mathbf{x}_j$*, then by Equation 5 we have that* b_{ij} *is the normalised affinity, i.e., the probability that* $\mathbf{x}_i$ *chooses* $\mathbf{x}_j$ *as its neighbour. Then the probability that data point* $\mathbf{x}_i$ *belongs to the outlier class is given by*

$$p(\mathbf{x}_i \in \mathcal{C}_\mathrm{O}) = \prod_{j \neq i} (1 - b_{ji}) \ . \tag{14}$$

Proof. We recall from Equation 7 that a data point $\mathbf{x}_i$ belongs to the set of outliers $\mathcal{C}_\mathrm{O}$, given one SNG graph G, when the corresponding vertex v_i has an in-degree of zero:

$$\mathbf{x}_i \in \mathcal{C}_\mathrm{O} \,|\, G \iff \deg^-_G(v_i) = 0 \ . \tag{15}$$

We aim to compute the marginal probability that a data point is an outlier, given all SNGs. By associating the right-hand side of Equation 15 with an indicator random value $\mathbb{1}\{\cdot\}$, which has a value of 1 if v_i has an in-degree of zero and has a value of 0 otherwise, we may rewrite the probability as the expected value of the indicator random variable (cf. Cormen et al., 2009, p. 118),

$$p(\mathbf{x}_i \in \mathcal{C}_\mathrm{O}) = \mathbb{E}_\mathcal{G}\left[\mathbb{1}\{\deg^-_G(v_i) = 0\}\right] \ , \tag{16}$$

where the subscripted $\mathcal{G}$ indicates the sample space. By rewriting Equation 9, which states that the in-degree of v_i is zero if none of the vertices bind to v_i, as a product, we obtain,

$$p(\mathbf{x}_i \in \mathcal{C}_\mathrm{O}) = \mathbb{E}_\mathcal{G}\left[\prod_{j \neq i} \mathbb{1}\{\, j \to i \notin \mathcal{E}_G\}\right] \ . \tag{17}$$

Substituting the indicator random variable by its complement yields,

$$p(\mathbf{x}_i \in \mathcal{C}_\mathrm{O}) = \mathbb{E}_\mathcal{G}\left[\prod_{j \neq i} (1 - \mathbb{1}\{\, j \to i \in \mathcal{E}_G\})\right] \ . \tag{18}$$

Because the vertices bind independently, the expected value operator is multiplicative (Ross, 2007, p. 52), which allows us to move the expected value operator inside the product,

$$p(\mathbf{x}_i \in \mathcal{C}_\mathrm{O}) = \prod_{j \neq i} (1 - \mathbb{E}_\mathcal{G}\left[\mathbb{1}\{\, j \to i \in \mathcal{E}_G\}\right]) \ . \tag{19}$$

We employ the same argument that we used for Equation 16, and we rewrite the expected value of the indicator random variable as the binding probability,

$$p(\mathbf{x}_i \in \mathcal{C}_\mathrm{O}) = \prod_{j \neq i} (1 - p(\, j \to i \in \mathcal{E}_G)) \ , \tag{20}$$

which is equal to the probability that data point $\mathbf{x}_j$ chooses data point $\mathbf{x}_i$ as its neighbour (see Equation 4). Hence, the probability that data point $\mathbf{x}_i$ belongs to the outlier class is

$$p(\mathbf{x}_i \in \mathcal{C}_O) = \prod_{j \neq i} (1 - b_{ji}) \ , \tag{21}$$

which, in words, states that the outlier probability of data point $\mathbf{x}_i$ is the probability that data point $\mathbf{x}_i$ is never chosen as a neighbour by the other data points. □

Figure 9 illustrates, for completeness, the final matrix transformation. The output matrix , which holds the outlier probabilities, is obtained by applying Equation 14 to the binding matrix $\mathbf{B}$. The right side of Figure 9 contains a plot of our example data set with the data points filled with the colour corresponding to their outlier probability (see the colour bar next to matrix $\mathbf{B}$). By Equation 22 we formally conclude our description of how SOS computes outlier probabilities: the scores produced by the outlier scoring algorithm φ_{SOS}, are equivalent to the outlier probabilities.

$$\varphi_{\text{SOS}}(\mathbf{x}_i) \equiv p(\mathbf{x}_i \in \mathcal{C}_O) \ . \tag{22}$$

In the next subsection we transform the outlier scoring algorithm φ_{SOS} into an outlier-selection algorithm, f_{SOS}, i.e., we transform the outlier probabilities into the classifications 'outlier' and 'inlier'.

2.7 Classifying outliers

In the previous six subsections, we have presented SOS as an outlier *scoring* algorithm φ_{SOS}, because it maps data points onto outlier scores. It is time for SOS to fulfil its name and transform it into an outlier *selection* algorithm f_{SOS}. By thresholding the computed outlier scores, classifications of the form 'outlier' and 'inlier' are obtained. Although SOS computes outlier probabilities instead of outlier scores, classifications are obtained in the same way.

$$f_{\text{SOS}}(\mathbf{x}) = \begin{cases} \text{outlier} & \text{if } \varphi_{\text{SOS}}(\mathbf{x}) > \theta, \\ \text{inlier} & \text{if } \varphi_{\text{SOS}}(\mathbf{x}) \leq \theta. \end{cases} \tag{23}$$

If the expert sets the threshold θ to 0.5, then applying Equation 23 to the outlier probabilities of the example data set results in the classifications as shown in Figure 10. The figure reveals that the first five data points, i.e., $\{\mathbf{x}_1, \ldots, \mathbf{x}_5\}$, are classified as inlier and that data point $\mathbf{x}_6$ is classified as outlier. We can indeed verify that only the outlier probability of $\mathbf{x}_6$, i.e., $\varphi_{\text{SOS}}(\mathbf{x}_6) = 0.788$, exceeds the threshold of 0.5. The selection boundary is obtained using the first five data points. So, the selection boundary indicates the region where a sixth data point would be classified as inlier. Because in the example data set, data point $\mathbf{x}_6$ lies outside the selection boundary, it is classified as outlier.

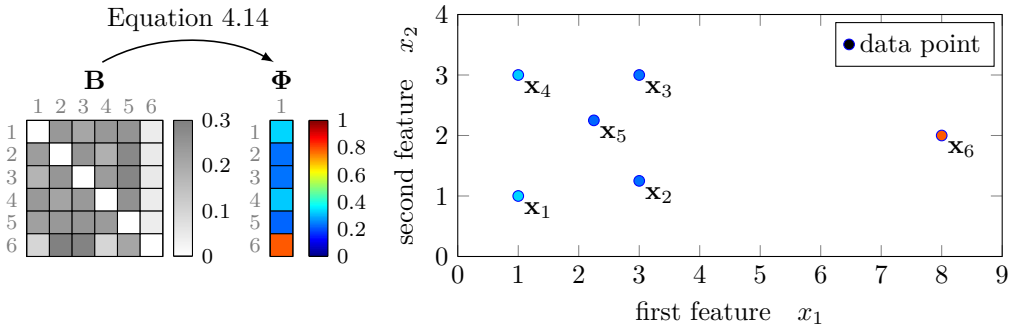

Figure 9: Outlier probabilities of the example data set. **Left:** The outlier probabilities in the output matrix $\mathbf{\Phi}$ are obtained by applying Equation 14 to the binding matrix $\mathbf{B}$. **Right:** A plot of our example data set with the data points filled by the colour corresponding to their outlier probability as computed by SOS.

Choosing a proper threshold for a certain real-world application can be challenging. If we are equipped with a loss associated with misclassifications, the Bayesian risk framework may be used to set the threshold so that the average expected loss of our decisions is minimised (Zellner, 1986).

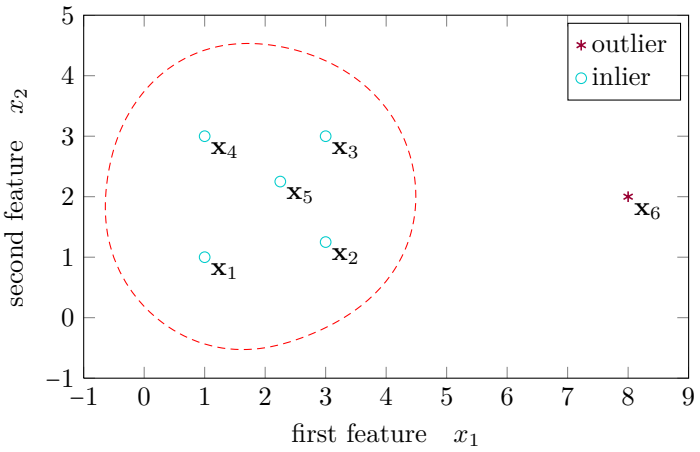

Figure 10: Classifications made by SOS on the example data set. Data point $\mathbf{x}_6$ is selected as outlier. The red, dashed line illustrates the selection boundary that corresponds to a threshold θ of 0.5.

2.8 Adaptive variances via the perplexity parameter

A possible challenge with the outlier probability in Equation 14 is that two data points that are similar to each other, but dissimilar from the remaining data points may have a low outlier probability, because the two data points have sufficient affinity with each other. We avoid this challenge by setting adaptively the variances σ_i^2 that are used for computing affinities in Equation 2.

We recall from Subsection 2.2 that SOS has one parameter h, called the perplexity. So far, we have treated the perplexity parameter h as a smooth measure for the effective number of neighbours of a data point, set by the expert. In fact, perplexity is a measurement that stems from the field of information theory. Perplexity may be computed for a probability distribution, for example, to compare it with another probability distribution (Jelinek, Mercer, Bahl, and Baker, 1977). In SOS, perplexity is employed to set adaptively the variances in such a way that each data point has h effective neighbours (Hinton and Roweis, 2003). To be precise, we require that the binding distribution $\mathbf{b}_i$ of each data point $\mathbf{x}_i$ has a perplexity that is equal to the perplexity parameter set by the expert,

$$h(\mathbf{b}_i) = 2^{H(\mathbf{b}_i)} \ , \tag{24}$$

where $H(\mathbf{b}_i)$ is the Shannon entropy of $\mathbf{b}_i$ (Shannon, 1948; MacKay, 2003),

$$H(\mathbf{b}_i) = -\sum_{\substack{j=1 \\ j \neq i}}^{n} b_{ij} \log_2 (b_{ij}) \ . \tag{25}$$

We remark that b_{ii} is not taken into account, because a data point is never its own neighbour. As a consequence of this requirement, the variances adapt to the local density of data points, in such a way that a higher density leads to a lower variance, causing the affinity a_{ij} to decay faster. The effect of such an adaptive variance is that $\mathbf{x}_i$ distributes, in general, around 90% of its affinity to its h nearest neighbours. So, indeed, the value of perplexity h may be interpreted as a smooth measure for the effective number of neighbours of a data point (van der Maaten and Hinton, 2008).

Figure 11 shows the influence that the perplexity parameter h has on the outlier probabilities. Having a fixed perplexity h, rather than a fixed variance σ^2 (cf. bandwidth in kernel density estimation (Parzen, 1962)), allows the SOS algorithm (1) to classify accurately data points in data sets with varying densities, and (2) to avoid the challenge with small clusters of outliers.

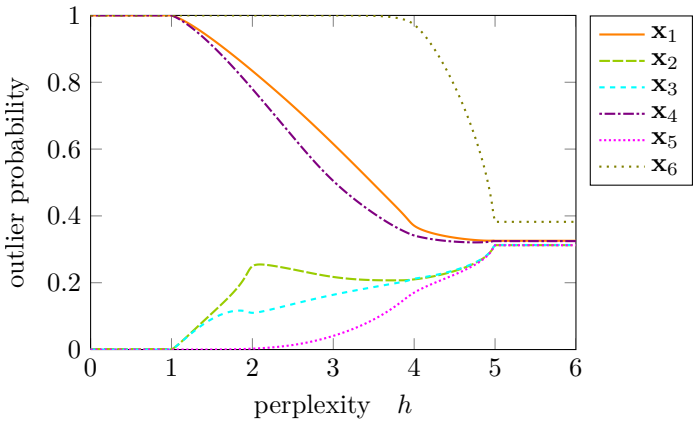

Figure 11: Influence of the perplexity h on the outlier probabilities of the six data points in the example data set.

Figure 12 shows how the variance influences the perplexity of the binding distribution. The values of the variances that correspond to the desired perplexity (which is set by the expert) are found using a binary search. The binary search starts with a sufficiently large interval (e.g., from 0.1 to 30) in which the desired variances lie. (This initial interval is derived from the distances between the data points.) In each iteration, the binary search bisects the interval and then selects a subinterval until the desired variances for each data point are found. Figure 13 shows the first 10 iterations of a binary search for the example data set. The figure shows that the perplexity of each binding distribution converges to the desired perplexity ($h = 4.5$).

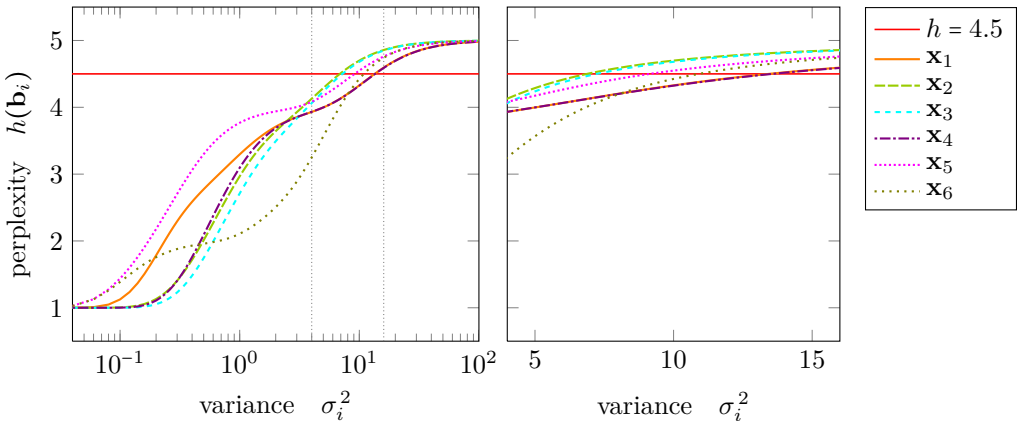

Figure 12: Six graphs of the perplexity $h(\mathbf{b}_i)$ with respect to the variance σ_i^2 for the six data points in the example data set. For each data point, a different variance is required such that the binding probability distribution $\mathbf{b}_i$ has the desired perplexity h of 4.5 (denoted by the horizontal, red line). **Left:** Semi-log plot with a logarithmic scale on the x-axis. **Right:** Linear plot with the same graphs, zoomed in on the range of values where each variance corresponds to the desired perplexity.

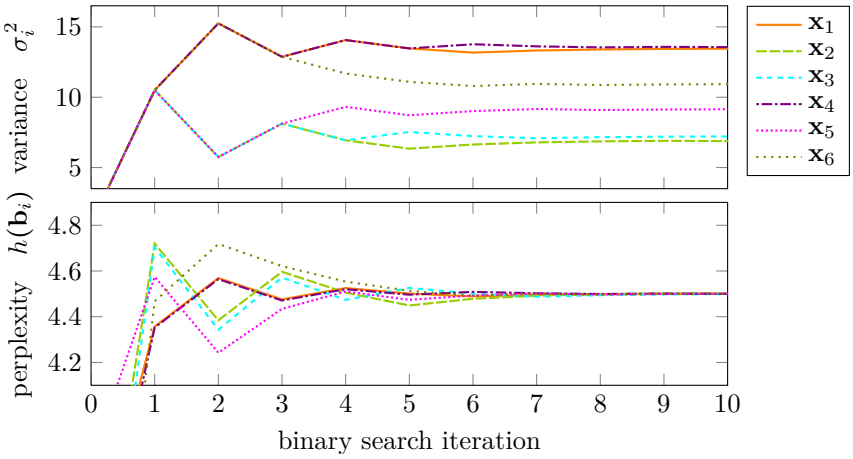

Figure 13: The first 10 iterations of the binary search that sets adaptively the variances. The desired perplexity h is set to 4.5. **Top:** The current variance for each of the six data points in the example data set. **Bottom:** The current perplexities given the current values of the variances.

3 Qualitative evaluation of SOS and four related algorithms

In this section we introduce and discuss four related outlier-selection algorithms and evaluate them, together with SOS, in a qualitative manner. The aim is to make this qualitative evaluation complementary to the quantitative evaluation (of the same algorithms) presented in Section 4. The four related algorithms are: K-Nearest Neighbour Data Description (KNNDD) by Tax (2001), Local Outlier Factor (LOF) by Breunig, Kriegel, Ng, and Sander (2000), Local Correlation Integral (LOCI) by Papadimitriou, Kitagawa, Gibbons, and Faloutsos (2003), and Least Squares Outlier Detection (LSOD) by Hido, Tsuboi, Kashima, Sugiyama, and Kanamori (2008) and Kanamori, Hido, and Sugiyama (2009).

To achieve our aim, we first explain outlier-score plots in Subsection 3.1. Subsequently, for each algorithm (SOS included) we (1) provide a brief description and (2) discuss several of its strong and weak points using the outlier-score plots shown in Figure 14 (Subsections 3.2–3.6).

3.1 Outlier-score plots

We qualitatively evaluate SOS and the four related algorithms using outlier-score plots on three small example data sets. The fifteen corresponding outlier-score plots are shown in Figure 14. An outlier-score plot is a two-dimensional plot that shows how well the algorithm captures the structure of a particular data set. (They are related to the plots employed in Aha, Kibler, and Albert (1991), which illustrate the decision boundaries of various instance-based learning algorithms.) So, an outlier-score plot may increase our understanding of outlier-selection algorithms.

In practice, an outlier-score plot is obtained as follows. First, we assume that we have a two-dimensional data set $\mathbf{X}$. Second, we generate a set of data points $\mathbf{Z}$ whose feature vectors correspond to the pixels of the resulting outlier-score plot. For instance, generating a plot of size 100 by 100 pixels requires $|\mathbf{Z}| = 10{,}000$ data points. Third, we remove one data point ($\mathbf{x}_{\text{new}}$) of $\mathbf{Z}$ and add it to $\mathbf{X}$. Fourth, we apply the outlier-selection algorithm to the data set $\mathbf{X}$ and record the outlier score (or probability in the case of SOS) for data point $\mathbf{x}_{\text{new}}$. Fifth, we remove $\mathbf{x}_{\text{new}}$ from $\mathbf{X}$. Steps 3, 4, and 5 are repeated until $\mathbf{Z}$ is empty. Sixth, the pixels of the plot are coloured by mapping the recorded outlier scores onto a colour map (cf. the colour map next to the output matrix in Figure 1 on page 3). The mapping of colours to outlier probabilities / scores varies by plot, since the possible minimum and maximum scores may differ by algorithm and by data set (except for SOS). Finally, in order to see the data set $\mathbf{X}$, its data points are

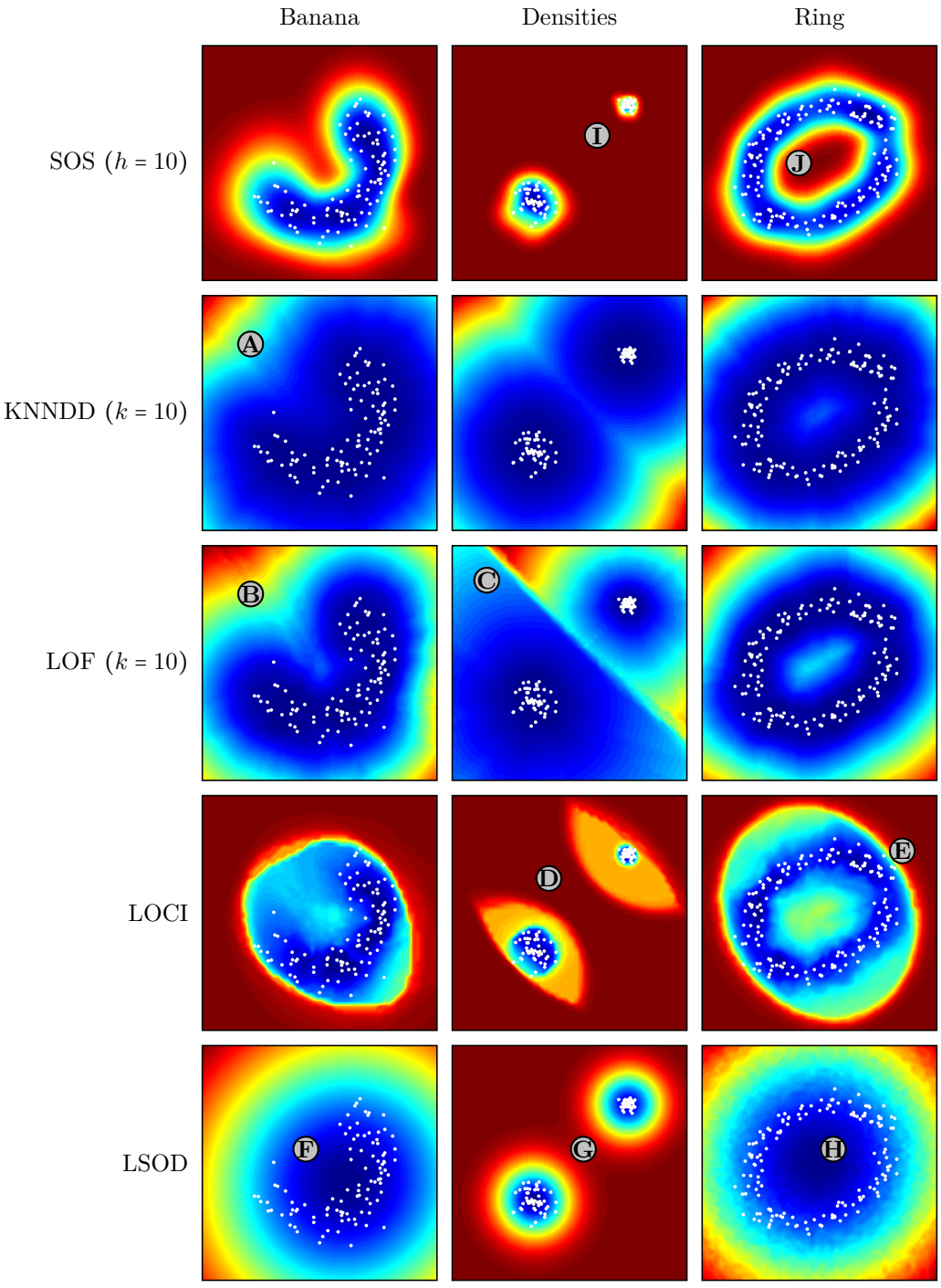

Figure 14: Outlier-score plots for SOS and four related outlier-selection algorithms as applied to three small example data sets. The colour at a certain location corresponds to the outlier probability / score that would be assigned to a new data point, should it appear at that location.

plotted as white dots.[1]

The first data set in Figure 14, *Banana*, contains a banana-shaped cluster with 100 data points. The asymmetric shape allows us to investigate how algorithms cope with irregular

[1]We restrict ourselves to two dimensions because it is not practical to visualise higher-dimensional outlier-score plots. For instance, a three-dimensional data set would require us to show the colour (outlier score) of all the $1 \cdot 10^7$ data points that lie in the cube of 100 by 100 by 100 voxels. The goal of outlier-score plots to gain an intuitive understanding of the outlier-selection algorithms. However, because the outlier scores computed by each of the five algorithms are determined by the Euclidean distance between the data points, we may expect that the performance of each algorithm to be affected similarly by higher-dimensional data sets. In Section 4.4 we report on the performances obtained on real-world data sets, which have a variety of dimensionalities.

distributions. The second data set, *Densities*, contains two Gaussian-distributed clusters with 50 data points each. The second cluster is denser than the first, which allows us to investigate how the algorithms cope with varying densities. The third data set, *Ring*, contains a rotated ellipse-shaped cluster with 150 data points. The ring allows us to investigate how algorithms cope with low density regions that are enclosed by data points. We note that each of the five algorithms is applied to exactly the same three data sets and that its parameter settings are kept constant. The number of data points has no influence on the outlier-score plots. We are now ready for a description and qualitative evaluation of SOS and the four related algorithms.

3.2 SOS

The results of SOS are shown in the top row of Figure 14. We see that SOS has a smooth boundary around the data points. For the Banana data set, the shape is captured well. For the Densities data set, we see that the boundary around the denser cluster is tighter than around the big, sparse cluster (see I). This indicates that SOS takes the relative density of the data well into account. For the Ring data set, the outlier probability assigned to a data point in the middle of the ring would be roughly equal to those appearing outside the ring (see J). The transition from a low to high outlier probability seems often smoother than with the other algorithms, which can be explained by the use of affinities that causes the property of 'being a neighbour' to be a smooth property.

3.3 KNNDD

The K-Nearest Neighbour Data Description (KNNDD) by Tax (2001) is an algorithm with one free parameter k. KNNDD defines the outlier score for a data point $\mathbf{x}$ as the ratio of the distance between $\mathbf{x}$ and its k-nearest neighbour $\mathbf{x}'$, and the distance between $\mathbf{x}'$ and *its* k-nearest neighbour. KNNDD differs from SOS in that it employs discrete rather than soft neighbourhood boundaries (this holds for LOF and LOCI too).

For the Banana data set, KNNDD generalises the shape of the data too much. The outlier scores for the Densities data set increase similarly for both clusters, which means that KNNDD is not so much influenced by the density of the data. Data points appearing in the middle of the Ring data set get a moderately higher outlier score. KNNDD has the drawback that outlier scores are unbounded (see A).

3.4 LOF

The Local Outlier Factor (Breunig et al., 2000) computes an outlier score by estimating the relative density in the neighbourhood of the data point. A data point, whose nearest neighbours have a smaller neighbourhood with an equal number of data points, is assigned a higher outlier score. The neighbourhood size is determined by a free parameter k.

LOF's outlier scores are also unbounded (see B). It captures the shape of the data in the Banana data set better than KNNDD. At the boundary between the two clusters in the Density data set, the outlier scores exhibit a discontinuity (see C), which is due to the use of discrete neighbourhood boundaries, and the fact that LOF takes densities explicitly into account, as opposed to KNNDD. The outlier scores in the middle of the Ring data set are more increased than with KNNDD.

3.5 LOCI

The Local Correlation Integral (Papadimitriou et al., 2003) also estimates the relative density in the neighbourhood of the data point, but then for a whole range of neighbourhood sizes. The outlier score is based on the maximum ratio between the local and global densities that is found in the range. Although some values in the actual algorithm can be adjusted, Papadimitriou et al. claim that LOCI has no free parameters, and will therefore be treated as such. The same holds for LSOD.

The outlier scores computed by LOCI do have a maximum, but this maximum may be different per data set. For the Banana data set, LOCI forms a strange shape around the data points. The strange shapes (see D) in the Densities data set, mostly include higher outlier scores, and are probably a result of the fact that the neighbourhoods of the constituent data points include data points from both clusters. The shape in the Ring data set seems to be orthogonal with the cluster (see E) and have very tight boundaries at certain places.

3.6 LSOD

Least Squares Outlier Detection (Kanamori et al., 2009; Hido et al., 2008) is an 'inlier-based' outlier-selection algorithm. Unlike the other algorithms, it is supervised, in that it uses a given set of data points, labelled as normality. The outlier scores of the remaining data points are given by the ratio of probability densities between the normalities and the remaining data points. For our experiments, we slightly alter LSOD by computing the outlier score of one data point at a time, and treating all other data points as normal. This way, LSOD can be considered as an unsupervised outlier-selection algorithm and thus ensures a fair comparison with the other algorithms. According to Kanamori et al. LSOD, like LOCI, has no free parameters.

For the Banana data set, LSOD seems to model the data as a spherical structure (see F), such that the outlier scores increase linearly with distance from the cluster centre. For the Densities data set, the outlier scores of LSOD seem to be less influenced by the density of clusters (see G). Data points appearing in the middle of the Ring data set would not be selected as outliers (see H).

4 Experiments and results

In this section we present our experiments and the corresponding results.

In Subsection 4.1, we discuss how binary- and multi-classification data sets can be transformed into one-class data sets such that they are usable for evaluating outlier-selection algorithms. In Subsection 4.2, we describe a procedure that simulates anomalies using one-class data sets. In Subsection 4.3, we describe the weighted version of Area Under the ROC Curve (AUC), which is appropriate for aggregating over multiple data sets.

We evaluate SOS and the four algorithms discussed in Section 3 (i.e., KNNDD, LOCI, LOF, and LSOD) on eighteen real-world data sets (Section 4.4) and on seven synthetic data sets (Section 4.5).

4.1 Constructing one-class data sets from a multi-class data set

To evaluate the performance of an outlier-selection algorithm, we need data sets where the data points are labelled as normal and anomalous. Such a data set is called a one-class data set $\mathcal{D}$. In order to obtain a good sense of the characteristics of an outlier-selection algorithm we need a large number of varied one-class data sets.

One-class data sets are not as abundant as multi-class data sets ($\mathcal{D}_{\mathrm{M}}$). A multi-class data set contains two or more classes $\mathcal{C}$ that are not necessarily labelled as normal or anomalous;

$$\mathcal{D}_{\mathrm{M}} = \bigcup_{i=1}^{m} \mathcal{C}_i \ , \tag{26}$$

where m is the number of classes. For example, the Iris flower data set (Fisher, 1936) consists of 50 data points from each of the three classes: (1) Setosa, (2) Versicolor, and (3) Virginica. Figure 15(top) shows a scatter plot of the Iris flower data set. Normally, multi-class data sets are used for binary classification (Asuncion and Frank, 2010).

We can construct a one-class data set from a multi-class data set by relabelling one class as the normal class ($\mathcal{C}_{\mathrm{N}}$) and the remaining $m-1$ classes as anomalous ($\mathcal{C}_{\mathrm{A}}$). Let

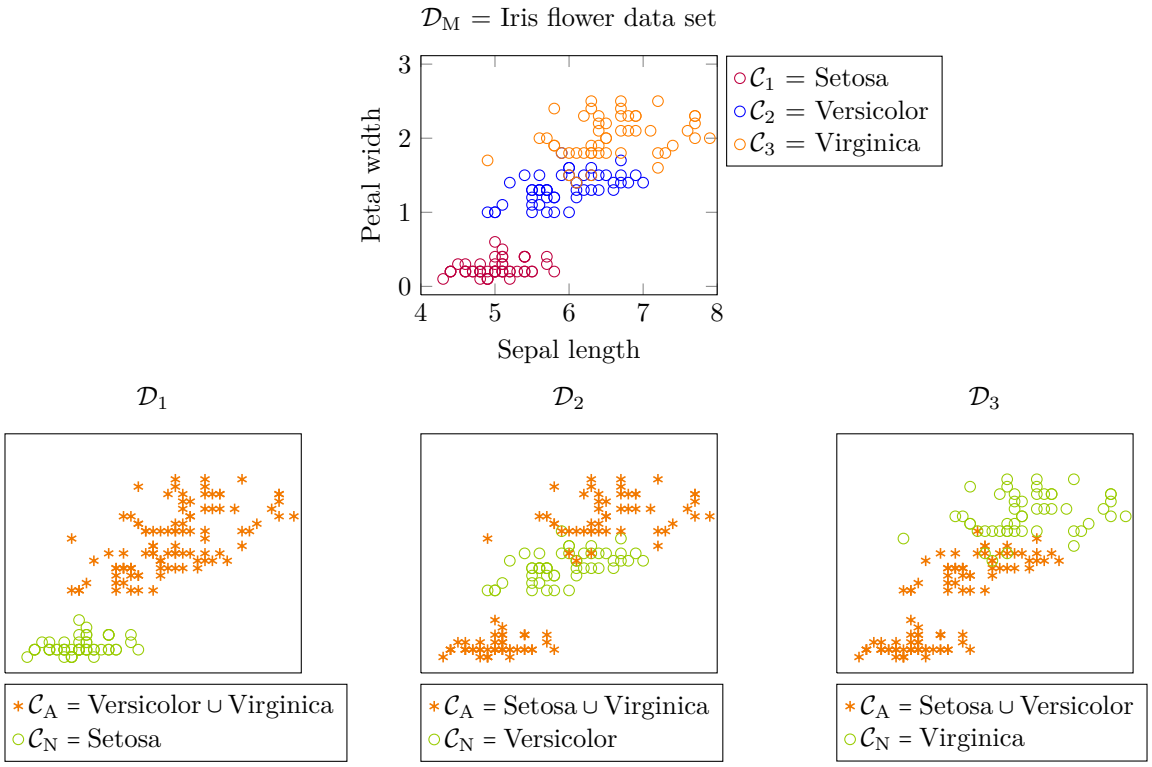

Figure 15: Illustration of relabelling a multi-class data set into multiple one-class data sets. **Top:** The Iris flower data set is a multi-class data set that consists of three classes: Setosa, Versicolor, and Virginica. **Bottom:** By relabelling the data points, three one-class data sets are obtained. Each of the three classes is made once the normal class.

$$\mathcal{D}_i = \mathcal{C}_{\mathrm{A}} \cup \mathcal{C}_{\mathrm{N}} \text{ such that } \mathcal{C}_{\mathrm{A}} = \bigcup_{\substack{j=1 \\ j \neq i}}^{m} \mathcal{C}_j \text{ and } \mathcal{C}_{\mathrm{N}} = \mathcal{C}_i. \tag{27}$$

For a multi-class data set containing m classes, we can repeat this m times, where each class is relabelled as the normal class once (Tax, 2001). We remark that a one-class data set contains the same data points as the multi-class data set, but with different labels. The bottom row of Figure 15 shows three one-class datasets: $\mathcal{D}_1$, $\mathcal{D}_2$, and $\mathcal{D}_3$ that are constructed from the Iris flower data set. These one-class data sets are suitable for evaluating an outlier-selection algorithm.

4.2 Simulating anomalies

In data sets that are obtained from multi-class data sets, as described in Subsection 4.1, usually both the normal and the anomalous class are well-represented, i.e., clustered. As such, an unsupervised outlier-selection algorithm would not classify any anomalous data points as outliers.

In order to use such a data set for the evaluation of outlier-selection algorithms, we employ a three-step procedure that simulates the anomalies to be rare. First, all data points of the anomalous class are removed from the data set. Second, the outlier-selection algorithm computes the outlier scores for all normal data points. Third, we add one of the anomalous data points and compute its outlier score, and remove it thereafter. The third step is repeated until all anomalous data points have been processed by the outlier-selection algorithm. The result is an outlier score for each normality and anomaly.

4.3 Weighted AUC

When applying an outlier-selection algorithm to multiple one-class data sets that are constructed from the same multi-class data set, we report performances with the help of the weighted AUC.

To compute the weighted AUC, i.e., multi-class AUC (cf. Fawcett, 2004), the AUCs of the one-class data sets are averaged, where each one-class data set is weighted according to the prevalence of the normal class, $\mathcal{C}_{\mathrm{N}i}$.

$$\mathrm{AUC}_{\mathrm{M}}(\varphi, \mathcal{D}_{\mathrm{M}}) = \sum_{i=1}^{m} \mathrm{AUC}(\varphi, \mathcal{D}_i) \cdot \frac{|\mathcal{C}_{\mathrm{N}i}|}{|\mathcal{D}_{\mathrm{M}}|} \tag{28}$$

The use of a weighted average prevents one-class data sets containing few normalities from dominating the results (Hempstalk and Frank, 2008).

4.4 Real-world data sets

We evaluated SOS and the four related outlier-selection algorithms on eighteen real-world data sets (see Figure 16 for the list of data sets). Except for the Delft Pump data set (Ypma, 2001) and the Colon Gene data set (Alon, Barkai, Notterman, Gish, Ybarra, Mack, and Levine, 1999), all data sets come from the UCI Machine Learning Repository (Asuncion and Frank, 2010). Because these data sets contain multiple classes that are not necessarily defined as either normal or anomalous, they are relabelled into multiple one-class data sets using the procedure from Subsection 4.1.

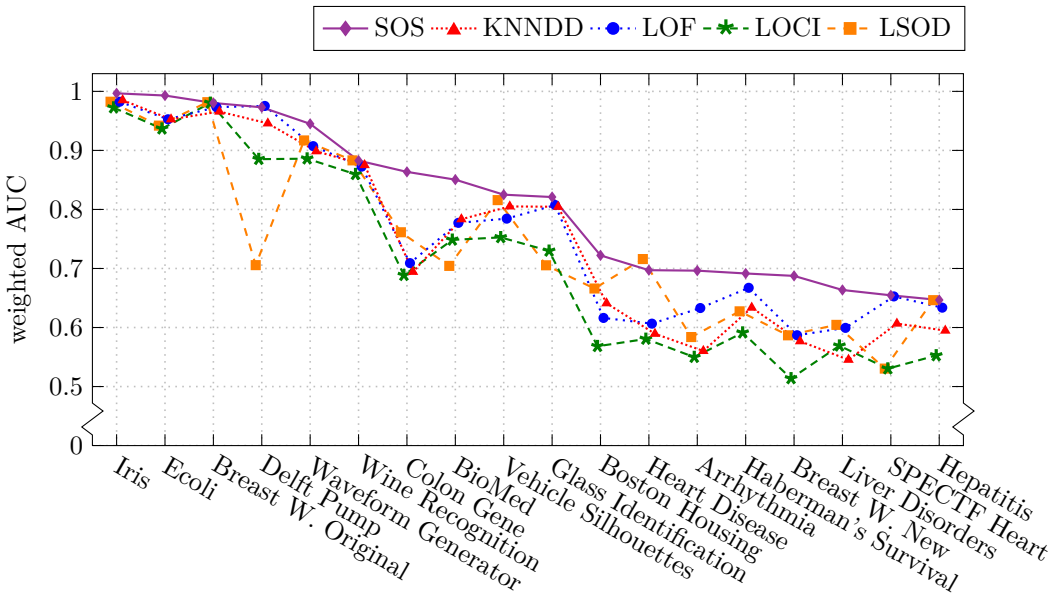

Figure 16: The weighted AUC performances of the five outlier-selection algorithms on eighteen real-world data sets. The data sets are ordered by the performances of SOS.

Figure 16 shows the weighted AUC performances of the five outlier-selection algorithms on the real-world data sets. The performance of SOS is illustrated by a solid (purple) line, while the other outlier-selection algorithms are illustrated by dashed and dotted lines. For clarity, the data sets are ordered according to the performance of SOS.

The figure reveals that SOS has a superior performance on twelve data sets. On the other six data sets its performance was at least 98% of the best performing algorithm.

For completeness, the AUC performances of SOS, KNNDD, LOF, LOCI, and LSOD on the 47 real-world one-class data sets are stated in Table 2 in the appendix for various parameter settings.

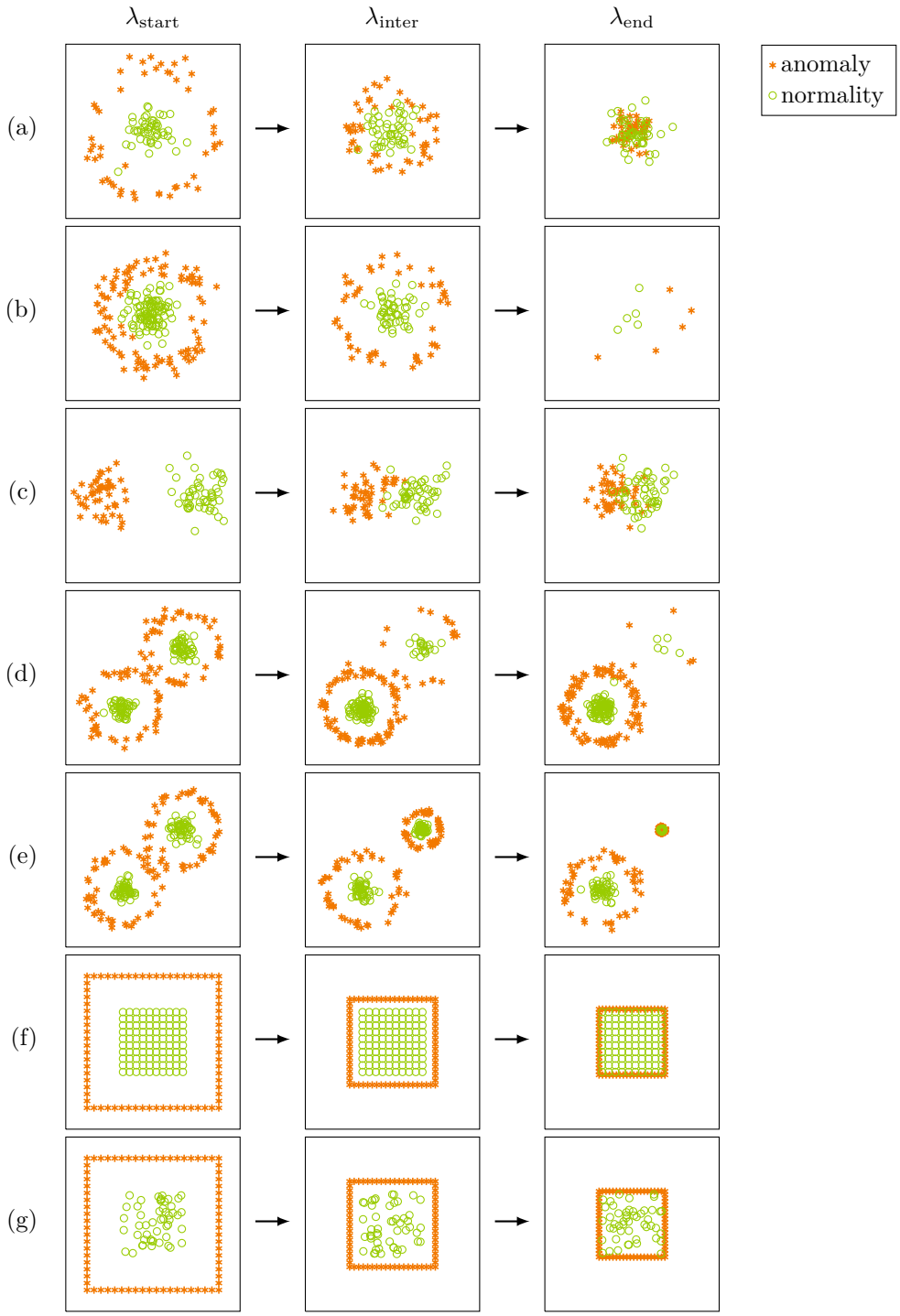

Figure 17: From left to right, the three columns illustrate, for each synthetic data set, three instantiations where λ is set to (1) the start value λ_{start}, (2) an intermediate value λ_{inter}, and (3) the end value λ_{end}, respectively.

4.5 Synthetic data sets

Although real-world data sets give a serious indication of how well the five algorithms will perform in a real-world setting, we may gain additional insight into their behaviour using synthetic data sets. Therefore, we designed seven synthetic data sets that contain data points from both the normal and the anomalous class. The synthetic data sets are two-dimensional. For each synthetic data set we introduced a single parameter, λ, that determines one property of the data set. In one data set, for example, λ corresponds to the distance between two Gaussian clusters. If we gradually adjust λ, i.e., adjust the distance between the two clusters

Table 1: The seven synthetic data sets controlled by parameter λ.

	Parameter λ			
Data set	Determines	λ_{start}	step size	λ_{end}
(a)	Radius of ring	5	0.1	2
(b)	Cardinality of cluster and ring	100	5	5
(c)	Distance between clusters	4	0.1	0
(d)	Cardinality of one cluster and ring	0	5	45
(e)	Density of one cluster and ring	1	0.05	0
(f)	Radius of square ring	2	0.05	0.8
(g)	Radius of square ring	2	0.05	0.8

while keeping the other properties constant, then we observe how cluster-overlap influences the performances of the algorithms under consideration. In general, the purpose of the synthetic data sets is to measure the resilience of each algorithm for the different data-set properties.

Table 1 lists all seven synthetic data sets and the function of parameter λ, viz. what property it determines. Besides cluster overlap (data sets (a), (c), (f), and (g)), we also evaluate the influence of cluster densities (e) and cluster cardinality ((b) and (d)).

Figure 17 illustrates for each synthetic data set three instantiations with three different values of λ, namely the start value, an intermediate value, and the end value. The reader should note that, similar to the real-world data sets, the true class-labels indicate whether a data point is anomalous or normal, and not whether it is an outlier or inlier. Consequently, due to the unsupervised nature of the outlier-selection algorithms, anomalous data points might not be selected as outliers (corresponding to a *miss*), especially as λ reaches its end value, λ_{end}.

Because the synthetic data sets contain random samples from various distributions and are generated anew for each evaluation, we applied each algorithm to 100 instantiations of each data set per value of λ, and computed the average AUC performances. Figure 18 displays the performances of the five algorithms on the seven synthetic data sets. Again, the performance of SOS is illustrated by a solid (purple) line and the other algorithms by dashed lines.

5 Discussion of the results

To compare the performances of multiple algorithms on multiple data sets, Demšar (2006) suggests the Neményi test (Neményi, 1963). The Neményi test checks for significant difference by ranking each algorithm for each data set, where the best performing algorithm is assigned the rank of 1, the second best the rank of 2, and so forth. Two algorithms are significantly different when their average ranks differ more than the critical distance, which is in our case 1.438 for $p = .05$.

We apply the Neményi test on the performances obtained by the five outlier-selection algorithms on the eighteen real-world data sets. The outcome is shown in the top part of Figure 19 by a critical difference diagram. Groups of methods that are not significantly different (at $p = .05$) are connected by a horizontal bar. For the real-world data sets, three (partially overlapping) groups of algorithms, ranked from low to high performance, are identified: (1) LOCI and KNNDD, (2) KNNDD, LSOD, and LOF, and (3) SOS. The performance of SOS is significantly higher than the other algorithms.

From the results on synthetic data sets (b) and (d), we may conclude that SOS has a superior performance with data sets that contain clusters with low cardinality. The performance on data set (e) indicates that SOS copes best with data sets containing clusters with varying densities. Although on data set (c), SOS is outperformed by the other algorithms for intermediate values of λ , the performance increases for $\lambda \leq 1$. This observation, combined with the superior results on data sets (a), (f), and (g), implies that SOS is, in general, less sensitive to cluster overlap than the other algorithms.

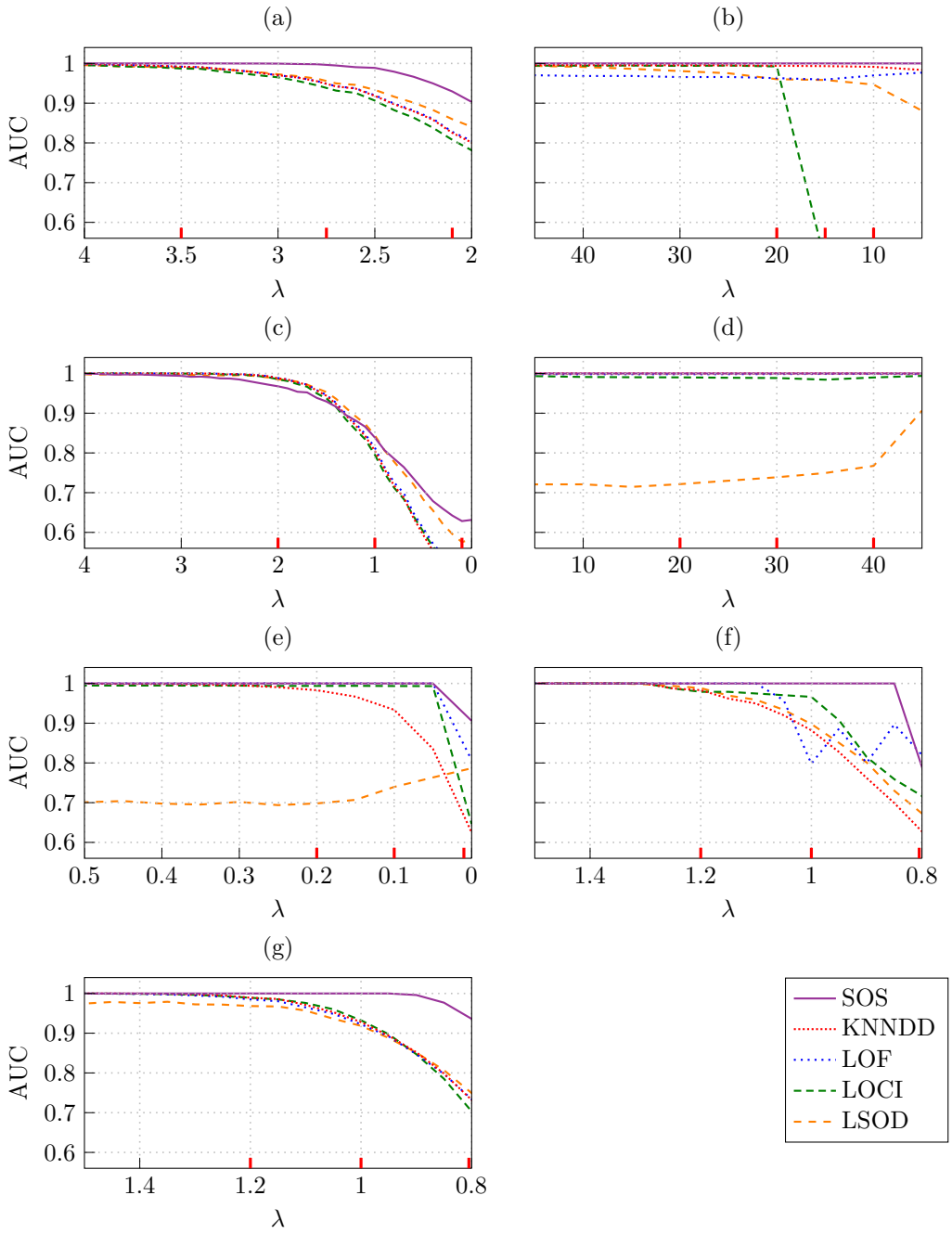

Figure 18: The AUC performances of the five outlier-selection algorithms on the seven synthetic data sets.

Apart from a few exceptions, the algorithms KNNDD, LOF, and LOCI show similar trends among the seven synthetic data sets. This was expected since they are all based on the k-nearest neighbour algorithm. For data set (b), LOCI shows a poor performance when the cardinality of the cluster is below 20. This is due to the requirement of LOCI to have a sample size of at least 20 (Papadimitriou et al., 2003). We expect that without this constraint, LOCI will perform comparably to KNNDD and LOF for $\lambda < 20$. Regarding LSOD, the results on the real-world data sets already showed that it has a relatively poor performance. Its performance on the synthetic data sets (d) and (e) confirm that LSOD is unable to handle data sets containing clusters of different cardinality or densities.

The bottom part of Figure 19 shows the critical difference diagram for the synthetic data sets. There are two groups of algorithms, ranked from low to high performance: (1) LOCI, LSOD, KNNDD, and LOF; and (2) SOS. The performance of SOS is significantly higher (at

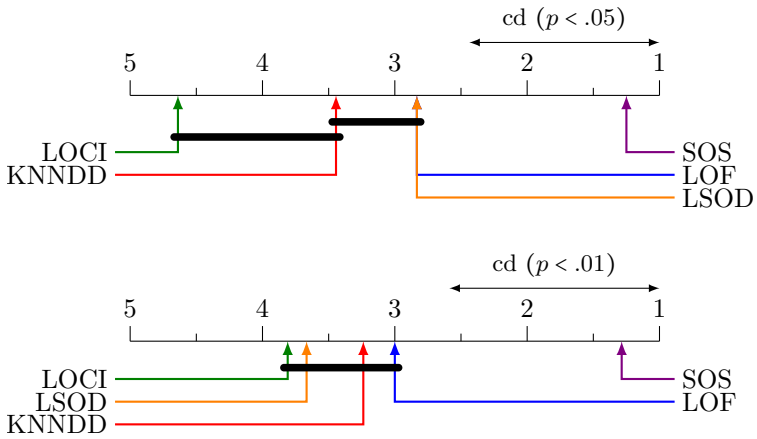

Figure 19: Critical difference diagrams. **Top:** From applying the outlier-selection algorithms on the eighteen real-world data sets. The critical distance is 1.438. **Bottom:** From applying the algorithms on the seven synthetic data sets. The critical distance is 1.588. Groups of algorithms that are connected are not significantly different.

$p = 0.01$) than the other algorithms.

6 Conclusion

In this report we developed and evaluated Stochastic Outlier Selection (SOS), a novel unsupervised algorithm for classifying data points as outliers in a data set. SOS computes for each data point an outlier probability, using affinities. The outlier probabilities provide three advantages with respect to unbounded outlier scores as computed by existing outlier-selection algorithms (cf. Subsection 1.2). First, a Bayesian risk model can be used to find an appropriate threshold for classifying data points as outliers (see Gao and Tan, 2006). Second, an ensemble outlier selection framework can be built by aggregating probabilities from individual outlier-selection algorithms. Third, we expect that outlier probabilities are easier to interpret by domain experts than outlier scores.

We described an evaluation procedure that enables us to evaluate unsupervised outlier-selection algorithms with standard benchmark data sets. We introduced the concept of an outlier-score plot, which allowed us to inspect visually how well an algorithm captures the structure of a data set (cf. Subsection 3.1). Using both real-world data sets and synthetic data sets, we have shown that SOS has an outstanding performance when compared to the current outlier-selection algorithms, KNNDD, LOF, LOCI, and LSOD. The Neményi statistical test revealed that SOS's performance is significantly higher. The seven synthetic data sets were parametrised by λ, such that the outlier-selection algorithms could be evaluated on individual data-set properties.

From our empirical results we observe that (1) SOS is an effective algorithm for classifying data points as outliers in a data set and that (2) SOS compares favourably to state-of-the-art outlier-selection algorithms. We may therefore conclude that the concept of affinity, which forms the basis SOS, is successfully applied to the problem of outlier selection.

References

D.W. Aha, D. Kibler, and M.K. Albert. Instance-based learning algorithms. *Machine Learning*, 6:37–66, 1991.

U. Alon, N. Barkai, D.A. Notterman, K. Gish, S. Ybarra, D. Mack, and A.J. Levine. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues

probed by oligonucleotide arrays. *Proceedings of the National Academy of Sciences*, 96(12): 6745–50, June 1999. ISSN 0027-8424.

A. Asuncion and A. Frank. UCI Machine learning repository, 2010. URL `http://archive.ics.uci.edu/ml`.

M.J. Bayarri and J.O. Berger. The interplay of Bayesian and Frequentist analysis. *Statistical Science*, 19(1):58–80, Feb. 2004. ISSN 0883-4237.

M.M. Breunig, H.P. Kriegel, R.T. Ng, and J. Sander. LOF: Identifying density-based local outliers. *ACM SIGMOD Record*, 29(2):93–104, 2000.

T.H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein. *Introduction to algorithms*. MIT Press, third edit edition, Sept. 2009.

J. Demšar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7:1–30, 2006.

T. Fawcett. ROC graphs: Notes and practical considerations for researchers. *Machine Learning*, 31:1–38, Mar. 2004.

R.A. Fisher. The use of measurements in taxonomic problems. *Annals of Eugenics*, 7(2): 179–188, 1936.

B.J. Frey and D. Dueck. Clustering by passing messages between data points. *Science (New York, N.Y.)*, 315(5814):972–6, Feb. 2007. ISSN 1095-9203.

J. Gao and P.N. Tan. Converting output scores from outlier detection algorithms into probability estimates. In *Proceedings of the 6th International Conference on Data Mining*, volume 6, pages 212–221, Hong Kong, China, Dec. 2006. Ieee. ISBN 0-7695-2701-7. doi: 10.1109/ICDM.2006.43.

J. Goldberger, S.T. Roweis, G.E. Hinton, and R. Salakhutdinov. Neighbourhood components analysis. In L.K. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems*, pages 513–520, Cambridge, MA, 2005. MIT Press.

K. Hempstalk and E. Frank. Discriminating against new classes: One-class versus multi-class classification. In *Proceedings of the 21st Australian Joint Conference on Artificial Intelligence*, pages 325–336, Auckland, New Zealand, 2008. Springer.

S. Hido, Y. Tsuboi, H. Kashima, M. Sugiyama, and T. Kanamori. Inlier-based outlier detection via direct density ratio estimation. In *Proceedings of the 8th International Conference on Data Mining*, pages 223–232, Los Alamitos, CA, 2008. IEEE Computer Society.

G.E. Hinton and S. Roweis. Stochastic Neighbor Embedding. In *Advances in Neural Information Processing Systems*, volume 17, pages 857–864, 2003.

F. Jelinek, R.L. Mercer, L.R. Bahl, and J.K Baker. Perplexity–a measure of the difficulty of speech recognition tasks. *Journal of the Acoustal Society of America*, 62(S1):S63, 1977.

T. Kanamori, S. Hido, and M. Sugiyama. A least-squares approach to direct importance estimation. *Journal of Machine Learning Research*, 10:1391–1445, 2009.

H.P. Kriegel, P. Kröger, E. Schubert, and S.A. Zimek. LoOP: Local outlier probabilities. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, pages 1649–1652. ACM, 2009. ISBN 9781605585123.

H.P. Kriegel, P. Kröger, E. Schubert, and S.A. Zimek. Interpreting and unifying outlier scores. In *Proceedings of the 11th SIAM International Conference On Data Mining*, 2011.

D.J.C. MacKay. *Information theory, inference, and learning algorithms*. Cambridge University Press, 2003. ISBN 0521642981.

P.F. Neményi. *Distribution-free multiple comparisons*. Ph.D. Thesis, Princeton, 1963.

S. Papadimitriou, H. Kitagawa, P.B. Gibbons, and C. Faloutsos. LOCI: Fast outlier detection using the local correlation integral. In *Proceedings of the 19th International Conference on Data Engineering*, pages 315–326, Bangalore, India, Mar. 2003.

E. Parzen. On estimation of a probability density function and mode. *The Annals of Mathematical Statistics*, pages 1065–1076, 1962.

S.M. Ross. *Introduction to probability models*. Elsevier, 9th editio edition, 2007.

S.T. Roweis and L.K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–6, Dec. 2000. ISSN 0036-8075.

C.E. Shannon. The mathematical theory of communication. *Bell System Technical Journal*, 27:379–423, 623–656, 1948. ISSN 0724-6811.

D.M.J. Tax. *One-class classification: Concept-learning in the absence of counter-examples*. Ph.D. Thesis, Delft University of Technology, Delft, The Netherlands, June 2001.

J.B. Tenenbaum, V. de Silva, and J.C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–23, Dec. 2000. ISSN 0036-8075.

L.J.P. van der Maaten. Learning a parametric embedding by preserving local structure. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, volume 5, pages 384–391, 2009.

L.J.P. van der Maaten and G.E. Hinton. Visualizing Data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

A. Ypma. *Learning methods for machine vibration analysis and health monitoring*. Ph.D. Thesis, Delft University of Technology, Delft, The Netherlands, 2001.

A. Zellner. Bayesian estimation and prediction using asymmetric loss functions. *Journal of the American Statistical Association*, 81(394):446–451, June 1986. ISSN 01621459.

A Performances on real-world one-class data sets

In Section 4.4, we evaluated SOS and four other outlier-selection algorithms on 18 real-world datasets. Since these datasets contained multiple, m, classes, m one-class data sets were generated. The reported performance measure was the weighted average AUC performance on these m one-class data sets (see Figure 16).

Table 2 shows the AUC performances on the 47 one-class datasets. The one-class data set is stated as the dataset name, as used in Figure 16, followed by the name of the normal class in italics and in brackets, e.g., "Iris *(Setosa)*". In addition to the maximum achieved AUC performance, Table 2 shows the performances for various other parameter values.

In the case of SOS, h_5, for example, is short-hand notation for $h = 5$. The maximum AUC performance is given by h_b. The parameter value corresponding to the best AUC performance is given by h and is indicated in gray. For KNNDD and LOF, the columns are k_b and k, respectively. Note that LOCI and LSOD have no free parameters.

Table 2: AUC performances on real-world one-class data sets.

	SOS							KNNDD							LOF							LOCI	LSOD
Dataset *(normal class)*	h_5	h_{10}	h_{20}	h_{50}	h_{100}	h_b	h	k_5	k_{10}	k_{20}	k_{50}	k_{100}	k_b	k	k_5	k_{10}	k_{20}	k_{50}	k_{100}	k_b	k	–	–
Iris *(Setosa)*	1	1	1	1	1	1	10	1	1	1	–	–	1	1	1	1	1	–	–	1	10	1	1
Iris *(Versicolor)*	.97	.98	.98	.98	.98	1	49	.98	.98	.97	–	–	.99	3	.97	.97	.96	–	–	.98	7	.97	.96
Iris *(Virginica)*	.94	.96	.97	.97	.97	.99	49	.95	.95	.94	–	–	.97	1	.95	.96	.93	–	–	.97	3	.95	.99
Breast W. Original *(malignant)*	.81	.85	.88	.91	.92	.98	250	.68	.8	.88	.94	.96	.97	150	.7	.74	.85	.96	.97	.97	149	.98	.98
Breast W. Original *(benign)*	.81	.85	.88	.91	.92	.98	250	.68	.8	.88	.94	.96	.97	150	.7	.74	.85	.96	.97	.97	149	.98	.98
Heart Disease *(diseased)*	.62	.58	.56	.55	.54	.67	2	.51	.5	.5	.49	.5	.51	3	.53	.55	.54	.48	.5	.56	11	.53	.68
Heart Disease *(healthy)*	.67	.64	.61	.6	.6	.72	2	.65	.65	.65	.64	.62	.66	16	.59	.61	.64	.63	.6	.65	18	.62	.75
BioMed *(healthy)*	.87	.89	.89	.88	.88	.91	122	.9	.9	.9	.89	.88	.91	4	.88	.87	.88	.89	.87	.89	4	.88	.9
BioMed *(diseased)*	.71	.67	.65	.64	.02	.73	2	.33	.33	.36	.51	–	.55	61	.52	.47	.44	.37	–	.56	42	.49	.33
Arrhythmia *(normal)*	.75	.76	.77	.79	.81	.85	200	.77	.77	.77	.77	.77	.78	16	.77	.78	.77	.77	.76	.78	9	.76	.76
Arrhythmia *(abnormal)*	.36	.33	.31	.3	.3	.5	1	.26	.25	.24	.25	.25	.28	1	.3	.27	.25	.26	.29	.45	1	.27	.35
Hepatitis *(normal)*	.6	.57	.53	.51	.55	.65	2	.59	.59	.59	.59	.57	.59	79	.54	.58	.57	.6	.53	.63	109	.55	.65
Ecoli *(periplasm)*	.9	.92	.91	.99	.99	.99	51	.94	.95	.95	.95	–	.95	47	.93	.94	.95	.72	–	.95	39	.94	.94
Delft Pump *(AR app.)*	1	1	1	1	1	1	100	1	1	.99	.99	.96	1	1	1	1	.99	.98	.84	1	4	.98	.95
Delft Pump *(5x3)*	.93	.96	.98	.99	.99	.99	70	.94	.91	.82	.71	.62	.95	1	.98	.97	.8	.83	.47	.99	3	.93	.66
Delft Pump *(5x1)*	.89	.93	.94	.94	.93	.97	132	.88	.76	.68	.56	.54	.94	1	.92	.79	.75	.46	.47	.95	1	.87	.68
Delft Pump *(3x2)*	.88	.96	.98	.96	.95	.99	136	.9	.83	.72	.59	.58	.94	1	.98	.77	.7	.5	.58	.99	4	.87	.69
Delft Pump *(2x2)*	.79	.88	.91	.88	.44	.92	99	.89	.81	.69	.52	–	.9	1	.94	.73	.66	.5	–	.97	2	.84	.66
Delft Pump *(1x3)*	.86	.92	.97	.92	.74	1	70	.95	.91	.7	.65	–	.97	1	.93	.89	.47	.64	–	.96	3	.9	.74
Delft Pump *(5x3 noisy)*	.9	.91	.88	.84	.82	.91	7	.75	.6	.54	.51	.49	.9	1	.61	.35	.46	.53	.48	.93	3	.74	.55
Delft Pump *(5x1 noisy)*	.94	.96	.95	.91	.45	.96	10	.83	.73	.73	.65	–	.94	1	.74	.78	.79	.56	–	.98	2	.85	.75
Delft Pump *(3x2 noisy)*	.86	.97	.99	.95	.41	.99	82	.92	.82	.73	.65	–	.97	1	.94	.67	.66	.53	–	1	2	.92	.72
Delft Pump *(2x2 noisy)*	.86	.95	.97	.92	.49	.98	63	.89	.81	.7	.58	–	.94	1	.97	.8	.53	.54	–	1	2	.92	.74
Delft Pump *(1x3 noisy)*	.93	.99	.99	.71	.71	1	40	.9	.87	.8	–	–	.96	1	1	.97	.66	–	–	1	7	.94	.79

Table 2: (Continued)

	SOS							KNNDD							LOF							LOCI	LSOD
Dataset *(normal class)*	h_5	h_{10}	h_{20}	h_{50}	h_{100}	h_b	h	k_5	k_{10}	k_{20}	k_{50}	k_{100}	k_b	k	k_5	k_{10}	k_{20}	k_{50}	k_{100}	k_b	k	–	–
Breast W. New *(non-ret)*	.61	.55	.53	.48	.49	.7	2	.58	.57	.58	.58	.58	.59	1	.51	.55	.54	.57	.56	.59	64	.51	.61
Breast W. New *(ret)*	.63	.55	.51	.36	.36	.65	3	.43	.4	.44	–	–	.52	1	.53	.41	.48	–	–	.57	2	.52	.53
SPECTF Heart *(0)*	.94	.94	.93	.92	.85	.95	93	.9	.88	.86	.85	–	.97	1	.83	.84	.84	.85	–	.92	1	.91	.98
SPECTF Heart *(1)*	.53	.48	.46	.43	.44	.54	4	.26	.22	.2	.2	.2	.47	1	.31	.25	.21	.2	.22	.55	1	.39	.36
Colon Gene *(2)*	.61	.63	.68	.82	.82	.86	39	.68	.69	.68	–	–	.69	12	.67	.7	.69	–	–	.71	9	.69	.76
Glass Identification *(float)*	.86	.86	.86	.87	.64	.9	68	.82	.79	.76	.73	–	.86	1	.85	.87	.85	.63	–	.87	12	.78	.86
Glass Identification *(nonfloat)*	.72	.72	.71	.72	.31	.75	75	.73	.71	.69	.65	–	.75	1	.73	.74	.69	.67	–	.75	9	.68	.56
Liver *(1)*	.61	.59	.57	.55	.58	.68	2	.59	.59	.59	.59	.58	.6	2	.56	.57	.59	.59	.56	.6	144	.58	.67
Liver *(2)*	.64	.62	.61	.59	.59	.65	2	.51	.5	.5	.49	.48	.51	4	.54	.55	.52	.49	.44	.6	2	.56	.56
Wine Recognition *(1)*	.95	.95	.96	.95	.98	.98	100	.93	.94	.96	.95	–	.96	20	.87	.82	.96	.93	–	.96	22	.95	.92
Wine Recognition *(2)*	.81	.77	.75	.75	.71	.81	5	.83	.82	.82	.82	–	.83	4	.73	.79	.79	.76	–	.82	34	.81	.85
Wine Recognition *(3)*	.87	.85	.83	.82	.82	.87	5	.8	.8	.81	–	–	.84	1	.79	.81	.81	–	–	.83	38	.83	.89
Waveform Generator *(0)*	.8	.82	.83	.85	.87	.92	300	.84	.84	.85	.85	.86	.86	127	.82	.84	.84	.86	.87	.87	94	.84	.87
Waveform Generator *(1)*	.85	.87	.88	.9	.91	.97	300	.9	.9	.91	.91	.92	.92	150	.88	.89	.9	.91	.93	.93	144	.92	.95
Waveform Generator *(2)*	.86	.87	.89	.9	.92	.95	300	.9	.9	.91	.91	.91	.91	147	.87	.89	.89	.9	.91	.91	112	.9	.93
Vehicle Silhouettes *(van)*	.95	.96	.97	.97	.97	.97	190	.96	.95	.93	.87	.81	.96	4	.95	.93	.91	.89	.73	.95	3	.92	.96
Vehicle Silhouettes *(Saab)*	.7	.72	.72	.73	.74	.74	99	.71	.7	.68	.64	.61	.71	5	.65	.65	.57	.45	.62	.67	3	.68	.74
Vehicle Silhouettes *(bus)*	.84	.85	.85	.87	.84	.9	200	.85	.81	.77	.71	.68	.91	1	.72	.68	.78	.75	.67	.88	1	.8	.87
Vehicle Silhouettes *(Opel)*	.67	.66	.65	.65	.65	.69	2	.65	.64	.63	.6	.59	.65	4	.59	.58	.55	.42	.62	.65	123	.62	.7
Haberman's Survival *(>5yr)*	.71	.69	.65	.62	.61	.72	4	.67	.68	.68	.66	.62	.68	10	.62	.67	.7	.65	.59	.71	18	.64	.64
Haberman's Survival *(<5yr)*	.62	.56	.53	.5	.36	.63	4	.41	.4	.43	.47	–	.5	74	.47	.45	.45	.5	–	.56	71	.45	.59
Boston Housing *(MEDV<35)*	.68	.66	.63	.59	.58	.72	2	.61	.61	.56	.51	.46	.62	1	.52	.59	.56	.52	.57	.59	11	.56	.65
Boston Housing *(MEDV>35)*	.69	.64	.6	.33	.33	.72	3	.66	.66	.63	–	–	.83	1	.42	.64	.63	–	–	.82	3	.62	.85