Networks of Sensors

Onderwater, M.

2016

document version
Publisher's PDF, also known as Version of record

3

Outlier Preservation by Dimensionality Reduction Techniques

In the previous chapter we discussed how middleware components bridge the gap between sensor networks and applications that rely on the data produced by these networks. With sensors playing an increasing role in technologies and in our lives, applications can choose many types and sources of data that are available at middleware platforms. How can all this information be transformed into actionable insight? Providing a short insightful summary that helps users identify events and take appropriate action is essential. Inevitably though, some information is always lost when providing a summary, so the technique used to create it should be chosen carefully. In this chapter we focus our attention on *Dimensionality Reduction* (DR), a family of techniques often used for creating short summaries. We study the effect of such techniques on outliers – measurements in the data that do not conform to regular patterns. We demonstrate that dimensionality reduction can indeed have a large impact on outliers. To that end we apply three dimensionality reduction techniques to three real-world data sets, and inspect how well they preserve outliers. We use several performance measures to demonstrate how well these techniques are capable of preserving outliers, and we discuss the results.

This chapter is based on the results presented in [4].

3.1 Introduction

Recent technological developments have resulted in a broad range of cheap and powerful sensor nodes, enabling companies to use sensor networks in a cost-effective way. Sensor networks will increasingly become part of our daily life – envision, e.g., a house with sensors related to smoke detection, lighting control, motion detection, environmental information, security issues, and structural monitoring. Combining all this information to actionable insights is a challenging problem. For instance, in the event of a burglary in a house, the sensors involved in motion detection, environmental monitoring, and security all yield useful information. Providing a short insightful summary that helps users identify the event and take appropriate action is essential. Dimensionality reduction is a family of techniques aimed at reducing the number of variables (dimensions) in the data and thus at making the data set smaller. In essence, it helps identify what is important, and what is not.

Dimensionality reduction often results in some loss of information, and applications might be affected by this loss. For instance, the burglary mentioned before is a (hopefully) rare event that is different from normal patterns in the sensor data (i.e., a so-called *outlier*). Unfortunately, DR-methods often lose outliers among the regular sensor data. Figure 3.1 illustrates this situation using a two-dimensional data set with an outlier near the top-left corner. When dimensionality is reduced by projecting all points onto a line, the outlier is mapped into the center of the reduced data set (the middle arrow in Figure 3.1), and is thus no longer an outlier. So dimensionality reduction might lose outliers among regular points, causing problems for applications relying on the detection of outliers.

A solution to this problem is to identify outliers prior to applying DR. This is, however, not always computationally feasible due to the high dimensionality of the data, particularly when an outlier involves multiple dimensions. The point in the top-left corner of Figure 3.1 is an example of such an outlier: it is not an outlier in either the x- or y-dimension, but clearly is an outlier in the (x, y)-plane. In such a computationally challenging situation, it might be more efficient to apply DR first, followed by the detection of outliers.

Motivated by this, in this chapter we experimentally determine how well DR-techniques preserve outliers. To this end, we describe three well-known DR-techniques that are relevant for a broad audience, and apply them to several real-world data sets from a sensor-related context. For each DR-technique we capture its capability to preserve outliers in three performance measures, and

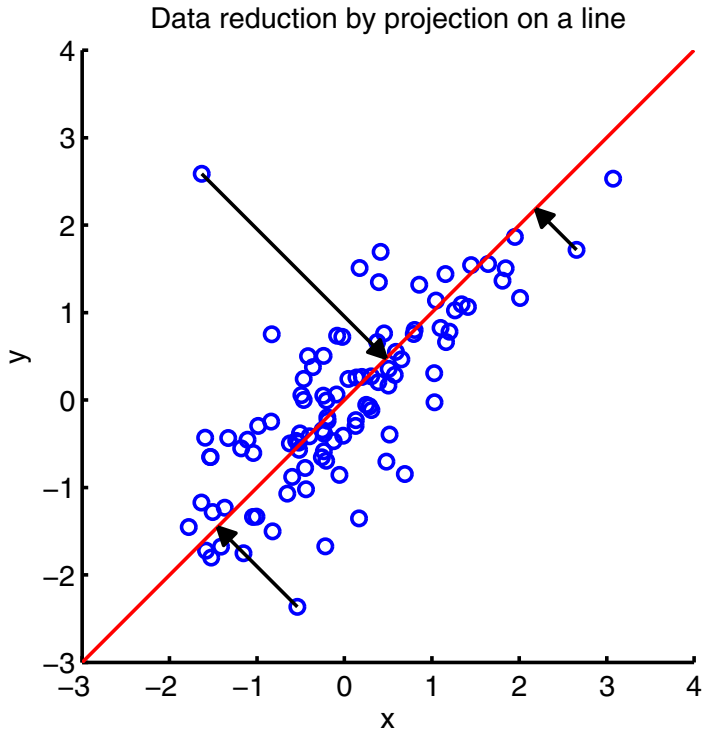

FIGURE 3.1: A two-dimensional data set reduced to one dimension, with an outlier (middle arrow) mapped to the center of the reduced data set.

compare the results. From the three techniques we identify the one with the best performance, and discuss the intuitions behind the scores.

A large body of literature exists on dimensionality reduction, and an overview of techniques from this field can be obtained from [33, 48, 59, 78, 120, 157]. DR-techniques are typically used for visualization [91, 152], as a preprocessing step for further analysis [52, 131, 153], and for increasing computational efficiency [42, 62]. Outlier detection is a popular research topic as well, and is comprehensively reviewed in survey papers [16, 70, 97, 163, 164]. Certain specific topics, such as intrusion detection [56] and fraud detection [26, 124], are closely related to outlier detection. In [114] the authors consider Kohonen's Self-Organizing Maps (SOM, [79]) and how this DR-technique can be used to identify outliers. [64] illustrates the effect of outlier-removal on Isomap [150], another DR-technique. [34] looks at local DR, where reduction is applied to previously identified clusters. Outlier detection occurs as part of the cluster-identification phase.

These papers do not, however, look at outlier *preservation* by DR-techniques, as discussed in this chapter. In [45], the authors compare multiple outlier detection methods on various data sets, including one data set with its dimensionality reduced. As in this chapter, their analysis also suggests that outlier detection is affected by dimensionality reduction, although they only use one DR-method and one performance measure. In [115], a setup is used that is close to our approach: four DR-methods are applied to three data sets, and the performance (using one score measure) is inspected for two outlier detection methods. However, the DR-methods in [115] are selected from the feature extraction domain, and are not well-known in the DR-community.

The structure of this chapter is as follows: Section 3.2 describes the DR-techniques, Section 3.3 contains the outlier detection method as well as the performance measures. Then, in Section 3.4 we describe the data sets that we use in the experiments. Section 3.5 demonstrates the output of the experiments and discusses the results, followed by conclusions, recommendations, and ideas for further research in Section 3.6.

3.2 Dimensionality reduction techniques

Denote by n the number of measurements and by d the number of sensors producing the measurements. The number of sensors is known as the *dimension* of the data, and DR-techniques aim to lower this dimension to a smaller value. More formally, if the measurements are vectors $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^d$, then DR-techniques try to find points $\mathbf{y}_1, \ldots, \mathbf{y}_n \in \mathbb{R}^{d'}$ with $d' < d$. This section describes three well-known and often used DR-techniques: *Principal Component Analysis* (PCA), *Multidimensional Scaling* (MDS), and *t-Stochastic Neighbourhood Embedding* (t-SNE).

3.2.1 Principal Component Analysis

Principal Component Analysis was initially proposed in [122]. It finds a low-dimensional representation of the data with minimal loss of variation of the data set. Suppose that we have n data points $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^d$ (corresponding to n measurements from each of d sensors in this chapter), and that they are placed in the $n \times d$ matrix X. We denote the $d \times d$ correlation matrix of X by C, its eigenvalues by $\lambda_1, \ldots, \lambda_d$, and its eigenvectors by $\mathbf{u}_1, \ldots, \mathbf{u}_d \in \mathbb{R}^d$. Typically, the eigenvalues are ordered s.t. $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_d$, and the

eigenvectors are orthogonal to each other. The eigenvalues reflect the amount of variance in the data set explained by the corresponding eigenvectors. To be precise, the first d' eigenvalues explain a fraction $\left(\sum_{k=1}^{d'} \lambda_k\right)/\left(\sum_{k=1}^{d} \lambda_k\right)$ of the variance.

PCA achieves dimensionality reduction by omitting eigenvectors $\mathbf{u}_{d'+1}, \ldots \mathbf{u}_d$, with d' the smallest integer such that the fraction of explained variance exceeds a threshold $\tau \in [0, 1]$. This threshold is a parameter of PCA. Summarized, the process works as follows:

1. Construct the data matrix X.
2. Compute the correlation matrix C.
3. Find the n eigenvalues λ_k and eigenvectors $\mathbf{u}_k$ of C.
4. Determine d' such that $\left(\sum_{k=1}^{d'} \lambda_k\right)/\left(\sum_{k=1}^{d} \lambda_k\right) < \tau$.
5. Construct matrix $\hat{U} = [\mathbf{u}_1 \ldots \mathbf{u}_{d'}]$.
6. Reduce dimension by computing $\hat{X} = X\hat{U}^T$.

The $n \times d'$ matrix $\hat{X}$ matrix contains n data points $\hat{\mathbf{x}}_\mathbf{1}, \ldots, \hat{\mathbf{x}}_\mathbf{n} \in \mathbb{R}^{d'}$ that form the reduced data set. The vectors $\mathbf{u}_k$ $(1 \leq k \leq d)$ vectors are called the *Principal Components*, and give PCA its name. A more detailed description and examples of PCA can be found in, for instance, [63, 74, 84, 149].

3.2.2 Multidimensional Scaling

Multidimensional Scaling is the name of a family of dimensionality reduction techniques that preserve distances in the data set. The *classical* version of MDS finds points $\mathbf{y}_1, \ldots, \mathbf{y}_n \in \mathbb{R}^{d'}$ in a low-dimensional space that minimize

$$\min_{\mathbf{y}_1,\ldots,\mathbf{y}_n} \sum_{i=1}^{n}\sum_{j=1}^{n} \left(||\mathbf{x}_i - \mathbf{x}_j|| - ||\mathbf{y}_i - \mathbf{y}_j||\right)^2. \tag{3.1}$$

Here $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^d$ are the high-dimensional points, and $||\cdot||$ is the Euclidean distance in the respective space. The classical version of MDS is equivalent to PCA, see for instance [54]. Other members of the MDS family use a different distance measure or a different quantity to optimize than Eq. (3.1). We use a version of MDS with the so-called *squared stress* criterion

$$\min_{\mathbf{y}_1,\ldots,\mathbf{y}_n} \frac{\sum_{i=1}^{n}\sum_{j=1}^{n} \left(||\mathbf{x}_i - \mathbf{x}_j||^2 - ||\mathbf{y}_i - \mathbf{y}_j||^2\right)^2}{\sum_{i=1}^{n}\sum_{j=1}^{n} ||\mathbf{x}_i - \mathbf{x}_j||^4}. \tag{3.2}$$

For the distance measure $||\mathbf{x}_i - \mathbf{x}_j||$ we do not use the Euclidean distance measure as in the classical version of MDS. To see why, note that MDS with the Euclidean distance is sensitive to natural variations in the data. Consider, for instance, a data set consisting of two columns, one with values uniformly drawn from the interval $[1000, 2000]$ and one with values drawn from $[0, 1]$. Clearly, all values in the first column are several orders of magnitude larger than those in the second column. When minimizing the quantity in Eq. (3.1) the procedure focuses on the elements of the first column, since that brings it closest to the minimum. In essence, the second column is ignored and MDS is biased towards the first column.

To overcome this problem, the Euclidean distance is typically replaced by the *Mahalanobis* distance [100]:

$$||x_i - x_j||_M = \sqrt{(\mathbf{x_i} - \mathbf{x_j})\Sigma^{-1}(\mathbf{x_i} - \mathbf{x_j})^T}, \tag{3.3}$$

where Σ is the covariance matrix. By including the covariance matrix in the distance measure, the natural variations in the data are removed and thus MDS is unbiased with respect to dimensions. Eq. (3.1) then becomes

$$\min_{\mathbf{y}_1,\ldots,\mathbf{y}_n} \frac{\sum_{i=1}^{n}\sum_{j=1}^{n}\left(||\mathbf{x}_i - \mathbf{x}_j||_M^2 - ||\mathbf{y}_i - \mathbf{y}_j||^2\right)^2}{\sum_{i=1}^{n}\sum_{j=1}^{n}||\mathbf{x}_i - \mathbf{x}_j||_M^4}. \tag{3.4}$$

Note that the Mahalanobis distance is only used for the high-dimension points $\mathbf{x}_i$, because the low-dimensional points $\mathbf{y}_i$ are found by the minimization.

3.2.3 t-Stochastic Neighbourhood Embedding

Stochastic Neighbourhood Embedding

t-Stochastic Neighbourhood Embedding is a variation on *Stochastic Neighbourhood Embedding* (SNE), first proposed in [68]. SNE presents the novel idea of defining a probability that two points are neighbours. If the distance between two points is small, SNE assigns a high 'probability of being a neighbour' to this pair. Similarly, points that are far apart are assigned a low 'probability of being a neighbour'. SNE reduces dimensionality by looking for low-dimensional points that preserve the assigned probabilities.

In SNE, the probability assigned to two points $\mathbf{x}_i$ and $\mathbf{x}_j$ is

$$p_{i|j} = \frac{e^{-||\mathbf{x}_i - \mathbf{x}_j||_M^2/2\sigma_i^2}}{\sum_{k=1,k\neq i}^{n} e^{-||\mathbf{x}_i - \mathbf{x}_k||_M^2/2\sigma_i^2}}. \tag{3.5}$$

The parameter σ_i is set by hand or determined with a special search algorithm. Note how we again employ the Mahalanobis distance from Eq. (3.3) for the high-dimensional points in Eq. (3.5). Also, observe that points that are close together result in a large value for $p_{i|j}$, and that points that are far away from each other yield a low value for $p_{i|j}$.

In low-dimensional space, probabilities similar to those in Eq. (3.5) are defined as

$$q_{i|j} = \frac{e^{-||\mathbf{y}_i - \mathbf{y}_j||^2}}{\sum_{k=1,k\neq i}^{n} e^{-||\mathbf{y}_i - \mathbf{y}_k||^2}}. \tag{3.6}$$

The parameter σ_i is not necessary here, because it would only lead to a rescaling of the resulting low-dimensional points $\mathbf{y}_i$. The $\mathbf{y}_i$ are then found by minimizing the Kullback-Leibler divergence of these two probability distributions

$$\min_{\mathbf{y}_1,\cdots,\mathbf{y}_n} \sum_{i=1}^{n}\sum_{j=1}^{n} p_{i|j} \log \frac{p_{i|j}}{q_{i|j}}. \tag{3.7}$$

Minimization of Eq. (3.7) can be done with, e.g., the gradient descent algorithm, or the scaled conjugate gradients procedure.

t-SNE

In [156] the authors propose t-SNE, which differs from SNE in two aspects. First, note that the probabilities in Eq. (3.5) are not necessarily symmetric, i.e., $p_{i|j}$ and $p_{j|i}$ do not need to be equal. This complicates minimization of Eq. (3.7), because it has twice as many variables as in the symmetric case. In t-SNE, the $p_{i|j}$ in Eq. (3.7) are replaced by p_{ij}:

$$p_{ij} = \frac{p_{i|j} + p_{j|i}}{2n},$$

with $p_{i|j}$ still computed from Eq. (3.5). Note that p_{ij} is symmetric in i and j, and thus reduces the number of variables in the minimization of the Kullback-Leibler divergence by a factor two. Additionally, this change ensures that $\sum_{j=1}^{n} p_{ij} > 1/(2n)$ so that each point (including outliers) has a significant contribution to the cost function.

The second change proposed for t-SNE concerns the q_{ij}. Instead of using Gaussian-style probabilities as in Eq. (3.6), t-SNE uses probabilities inspired

by the Student t-distribution (with one degree of freedom):

$$q_{ij} = \frac{(1 + ||\mathbf{y}_i - \mathbf{y}_j||^2)^{-1}}{\sum_{k=1,k\neq i}^{n}(1 + ||\mathbf{y}_i - \mathbf{y}_k||^2)^{-1}}.$$

This distribution has heavier tails than the Gaussian used by SNE, so it maps nearby high-dimensional points less nearby in low-dimensional space than SNE. A justification for this approach comes from the so-called *Crowding problem*: there is much more room in high-dimensional space for points, so in a low-dimensional representation data points tend to be 'squeezed' together. By using the Student t-distribution, these crowded points are placed just a bit further apart.

Low-dimensional points are still found by optimizing the Kullback-Leibler divergence from Eq. (3.7), but with $p_{i|j}$ replaced by p_{ij} and $q_{i|j}$ by q_{ij}:

$$\min_{\mathbf{y}_1,\cdots,\mathbf{y}_n} \sum_{i=1}^{n}\sum_{j=1}^{n} p_{ij} \log \frac{p_{ij}}{q_{ij}}. \tag{3.8}$$

3.3 Experimental setup

We adopt the following experimental setup when investigating dimensionality reduction for outlier preservation:

1. Normalize each data set such that it has zero mean and unit variance. This is a common preprocessing step for experimental data.
2. Find outliers in the high-dimensional (centered and scaled) data set.
3. Reduce the data set to two dimensions.
4. Again look for outliers, this time in the low-dimensional data.
5. Compute a score reflecting the performance of each DR-method on the data set.

We apply this setup to the DR-techniques from Section 3.2 and to a number of real-world data sets, described later in Section 3.4. Prior to that, the sections below describe the technique that we use for outlier detection, and three performance measures that we use to assess how well outliers are preserved. For the DR-techniques we used Matlab implementations available in the *Dimensionality Reduction Toolbox* [155].

Algorithm 3.1 Peeling

1. Calculate the convex hull around all the points in the data set.
2. Find the point on the hull with the largest (Mahalanobis) distance to all other points in the data set.
3. Remember the outlier and remove it from the data set.
4. Calculate the new convex hull, and check if the stop criterion is reached. If so, stop, otherwise continue with step 2.

3.3.1 Onion peeling

The idea of *Onion Peeling*, or Peeling in short, is to construct a convex hull around all the points in the data set and then find the points that are on the convex hull. These points form the first 'peel' and are removed from the data set. Repeating the process gives more peels, each containing a number of points. This technique can be utilized for finding outliers, if we consider a point in the data set to be an outlier if they have a large distance to the other points in the data set. With this intuitive interpretation of an outlier, the largest outlier in the data set is on the first peel. By inspecting the total distance of each point on the hull to all other points in the data set, we can find the one with the largest total distance. Removing this point from the data set and repeating the process gives new outliers. The decrease in volume of the convex hull after removing an outlier is used as a stop criterion. Once the volume decreases by a fraction less than α ($0 \leq \alpha \leq 1$), we stop looking for outliers. In our experiments we set $\alpha = 0.005$. Although with this procedure there is no guarantee that all outliers are found, it is sufficient for the data sets in this chapter. Peeling is outlined in Algorithm 3.1.

3.3.2 Measuring performance

After running the experiment for one data set and one DR-method, we need to quantify the performance of this method with respect to the preservation of outliers. In order to do so, we assign each point to one of four groups:

- True Positive (TP). The point is an outlier both before and after DR.
- False Positive (FP). The point is not an outlier before DR, but is one after.
- False Negative (FN). The point is an outlier before DR, but not after.
- True Negative (TN). The point is not an outlier before DR, nor after.

		Outlier before DR?	
		Yes	**No**
After DR?	**Yes**	TP	FP
	No	FN	TN

FIGURE 3.2: Confusion matrix indicating what happened to outliers after DR.

We can summarize these quantities in a *confusion matrix*, as demonstrated in Figure 3.2. In an ideal scenario the confusion matrix would be diagonal (i.e., 0 FPs and FNs), indicating that all outliers and non-outliers were correctly retained by the DR-methods. However, in practice the matrix often contains some FPs and FNs, and the performance of a DR-method is judged by all four quantities. Confusion matrices are used in several research communities to assess the performance of, e.g., binary classifiers and statistical tests. Often a single number is needed to capture performance, which subsequently results in a combination of the four quantities in the table. Several such combinations exist and are used in various fields of research (see the overview in [127] for more information).

We describe three performance measures that are often used in the literature, but before we do so we highlight one complicating aspect of our problem scenario. Most practical data sets have a significantly larger number of non-outliers than outliers, so in the confusion matrix the TN is usually the largest number. As an example of a performance measure that is affected by this, we look at *accuracy*, defined as

$$\text{accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{FN} + \text{TN}}.$$

Since TN is the dominating number in this expression, accuracy is always close to 1, making it difficult to identify small differences in performance. The three performance measures described below are selected because they are capable of handling this issue.

F1-score

The *F1-score* is a combination of *recall* and *precision*:

- Recall: the fraction of high-dimensional outliers that is retained by the DR-method (i.e., $\text{TP}/(\text{TP}+\text{FN})$), which is maximized when FN equals 0.
- Precision: the fraction of low-dimensional outliers that were also high-dimensional outliers (i.e., $\text{TP}/(\text{TP}+\text{FP})$), which is maximized when FP equals 0.

The F1-score takes the harmonic mean of precision and recall, resulting in a number between 0 (when TP=0) and 1 (when FP=FN=0):

$$\begin{aligned} F1 &= 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \\ &= 2 \cdot \frac{\text{TP}/(\text{TP}+\text{FP}) \cdot \text{TP}/(\text{TP}+\text{FN})}{\text{TP}/(\text{TP}+\text{FP}) + \text{TP}/(\text{TP}+\text{FN})} \\ &= \frac{2\text{TP}}{(2\text{TP}+\text{FN}+\text{FP})}. \end{aligned} \tag{3.9}$$

If $\text{TP}+\text{FN}=0$ or $\text{TP}+\text{FP}=0$ then the F1-score is defined as 0. Note that the element TN of the confusion table does not affect the score, and it is therefore not affected by the sparsity of outliers. The F1-score is used in, e.g., Information Retrieval [32, 103] and Machine Learning [45, 136, 154].

Matthews correlation

The *Matthews Correlation* [107] computes a correlation coefficient between the class labels (i.e., outlier or non-outlier) in high and low dimension of each point in the data sets. It results in a number between -1 (perfect anti-correlation) and 1 (perfect correlation), with 0 indicating the absence of correlation. Below we derive an expression for the Matthews Correlation in terms of the elements of the confusion matrix. Denote the class labels in low-dimensional space by $l_1 \cdots l_n$ and those in high-dimensional space by $h_1 \cdots h_n$, i.e.,

$$h_i = \begin{cases} 1 & \text{if point } i \text{ is an outlier in high dimension,} \\ 0 & \text{otherwise,} \end{cases}$$

and

$$l_i = \begin{cases} 1 & \text{if point } i \text{ is an outlier in low dimension,} \\ 0 & \text{otherwise.} \end{cases}$$

Here, n is still the total number of points in the data set. The Matthews correlation can be interpreted as a measure of how well outliers are preserved. It is denoted by ρ, and computed from

$$\rho = \frac{1}{n-1} \frac{\sum_{i=1}^{n}(l_i - \bar{l})(h_i - \bar{h})}{\sigma_l \sigma_h}, \tag{3.10}$$

where

$$\bar{l} = \frac{1}{n}\sum_{i=1}^{n} l_i = \frac{\text{TP} + \text{FP}}{n}, \qquad \bar{h} = \frac{1}{n}\sum_{i=1}^{n} h_i = \frac{\text{TP} + \text{FN}}{n}, \tag{3.11}$$

using notation from the confusion matrix. The σ_l is the standard deviation of the l_i, i.e.,

$$\begin{aligned}
\sigma_l &= \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(l_i - \bar{l})^2} = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(l_i^2 - 2l_i\bar{l} + \bar{l}^2)} \\
&= \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(l_i - 2l_i\bar{l} + \bar{l}^2)} = \sqrt{\frac{n\bar{l} - 2n\bar{l}^2 + n\bar{l}^2}{n-1}} \\
&= \sqrt{\frac{n}{n-1}}\sqrt{\bar{l}(1-\bar{l})}.
\end{aligned}$$

Similarly, the standard deviation of the h_i becomes $\sigma_h = \sqrt{\frac{n}{n-1}}\sqrt{\bar{h}(1-\bar{h})}$. Substituting these quantities into Eq. (3.10) yields

$$\begin{aligned}
\rho &= \frac{\sum_{i=1}^{n}(l_i - \bar{l})(h_i - \bar{h})}{\sigma_l \sigma_h} = \frac{\sum_{i=1}^{n}(l_i - \bar{l})(h_i - \bar{h})}{n\sqrt{\bar{l}\bar{h}(1-\bar{l})(1-\bar{h})}} \\
&= \frac{\sum_{i=1}^{n}(l_i h_i - \bar{l}h_i - l_i\bar{h} + \bar{l}\bar{h})}{n\sqrt{\bar{l}\bar{h}(1-\bar{l})(1-\bar{h})}} = \frac{\sum_{i=1}^{n}(l_i h_i) - n\bar{l}\bar{h} - n\bar{l}\bar{h} + n\bar{l}\bar{h}}{n\sqrt{\bar{l}\bar{h}(1-\bar{l})(1-\bar{h})}} \\
&= \frac{\sum_{i=1}^{n}(l_i h_i) - n\bar{l}\bar{h}}{n\sqrt{\bar{l}\bar{h}(1-\bar{l})(1-\bar{h})}}.
\end{aligned}$$

Using $\sum_{i=1}^{n}(l_i h_i) = \text{TP}$ and Eq. (3.11), some algebra yields

$$\rho = \frac{\text{TP}\cdot\text{TN} - \text{FP}\cdot\text{FN}}{\sqrt{(\text{TP}+\text{FN})(\text{TP}+\text{FP})(\text{TN}+\text{FP})(\text{TN}+\text{FN})}}. \tag{3.12}$$

If any of $\text{TP}+\text{FN}, \text{TP}+\text{FP}, \text{TN}+\text{FP}$, or $\text{TN}+\text{FN}$ are 0, then ρ is defined as 0. Note that, since ρ is a correlation, it is not affected by the large number of non-outliers. The Matthews Correlation is often used in Bioinformatics to assess the performance of classifiers, see, e.g., [75, 111, 139].

Relative information score

The Relative Information score was proposed in [80] and relies on ideas from the Information Theory field. In this section we derive an expression for the Relative Information score based on the confusion matrix. Suppose we consider one particular point, then a priori we can compute the probability that it is an outlier from the confusion matrix

$$\mathbb{P}(\text{outlier in high dimension}) = \frac{\text{TP} + \text{FN}}{n}.$$

After DR, we can compute this same probability for the same point as

$$\mathbb{P}(\text{outlier in low dimension} \mid \text{outlier in high dimension}) = \frac{\text{TP}}{\text{TP} + \text{FN}}.$$

The authors of [80] argue that any well-performing classifier (DR-method) should at least result in a confusion table with $\frac{\text{TP}}{\text{TP}+\text{FN}} > \frac{\text{TP}+\text{FN}}{n}$, otherwise it has lost information from the original data. This forms the basis for their Relative Information score.

We introduce some notation and denote by $\mathbb{P}(C_i = c)$ the probability that point i in the data set has class c, with $c = 1$ indicating that it is an outlier in high dimension, and $c = 0$ that it is a non-outlier. From the confusion matrix, we know that

$$\mathbb{P}(C_i = 1) = \frac{\text{TP} + \text{FN}}{n}, \tag{3.13}$$

$$\mathbb{P}(C_i = 0) = \frac{\text{FP} + \text{TN}}{n}. \tag{3.14}$$

After DR each point is again an outlier or non-outlier, but this time in low dimension. We denote the probability that point i in low dimension has class c, given that it also had class c in high dimension, by $\mathbb{P}(C'_i = c|C_i = c)$. From the confusion matrix, we find that

$$\mathbb{P}(C'_i = 1|C_i = 1) = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{3.15}$$

$$\mathbb{P}(C'_i = 0|C_i = 0) = \frac{\text{TN}}{\text{FP} + \text{TN}}. \tag{3.16}$$

In [80], the amount of information (as defined by [137]) necessary to correctly classify point i is measured as

$$-\log_2(\mathbb{P}(C'_i = c|C_i = c)).$$

A DR-method that satisfies $\mathbb{P}(C'_i = c|C_i = c) > \mathbb{P}(C_i = c)$ for point i then gets a positive score on point i of

$$\log_2\big(\mathbb{P}(C'_i = c|C_i = c)\big) - \log_2\big(\mathbb{P}(C_i = c)\big).$$

If $\mathbb{P}(C'_i = c|C_i = c) < \mathbb{P}(C_i = c)$ the score is

$$\log_2\big(1 - \mathbb{P}(C_i = c)\big) - \log_2\big(1 - \mathbb{P}(C'_i = c|C_i = c)\big),$$

which is negative. When $\mathbb{P}(C'_i = c|C_i = c) = \mathbb{P}(C_i = c)$ the score is defined as 0. The total score I of a DR-method is then

$$I = \sum_{i=1}^{n} \Big[\mathbb{1}_{\{\mathbb{P}(C'_i=c|C_i=c) > \mathbb{P}(C_i=c)\}} \cdot \log_2 \frac{\mathbb{P}(C'_i = c|C_i = c)}{\mathbb{P}(C_i = c)} + \mathbb{1}_{\{\mathbb{P}(C'_i=c|C_i=c) < \mathbb{P}(C_i=c)\}} \cdot \log_2 \frac{1 - \mathbb{P}(C_i = c)}{1 - \mathbb{P}(C'_i = c|C_i = c)} \Big].$$

Here, we used the equality $\log_2 x - \log_2 y = \log_2(x/y)$ for compactness. Usually, when comparing classifiers, I is reported relative to the expected information E needed to correctly classify each point:

$$E = -\sum_{i=1}^{n} \mathbb{P}(C'_i = c|C_i = c) \cdot \log_2(\mathbb{P}(C'_i = c|C_i = c)). \tag{3.17}$$

The Relative Information score I_r is then

$$I_r = \frac{I}{E} \cdot 100\%. \tag{3.18}$$

Note that I_r can become negative because I can also be negative. Inserting Eqs. (3.13)-(3.16) into Eq. (3.17) and Eq. (3.18) yields an expression in terms of the elements of the confusion matrix.

3.4 Data sets

In the previous sections we described the setup of our experiments, the DR-techniques, and how we measure their performance. The experiments use three real-world data sets which we describe below.

Radiant light energy measurements. The measurements in this data set are from sensor nodes deployed in several office buildings in New York City, as part of Columbia University's EnHANTs project. Each node has one sensor

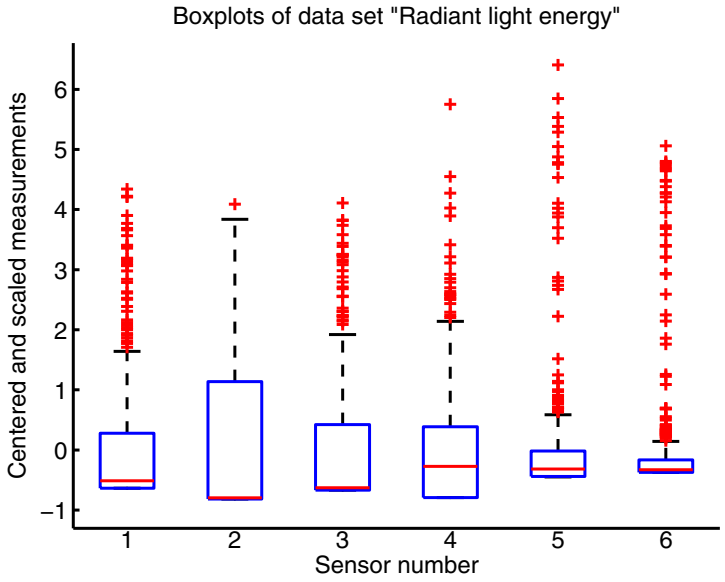

FIGURE 3.3: Boxplots of the sensors in the "Radiant light energy" data set.

measuring *irradiance* (radiant light energy), and this data set contains values measured during about one year. Figure 3.3 shows boxplots of each of the six sensors in this data set, with the measurements centered and scaled as discussed in Section 3.3. Each boxplot reflects the distribution of the 500 measurements by one sensor, and highlights possible outliers. Each sensor contains 40-60 possible outliers, except for the second sensor which has just one. The Peeling algorithm from Section 3.3.1 selects which of these points we use as outliers in our experiments. Note also that the median of each sensor's values (except sensor 4) is close to the .25 quantile, indicating that those distributions are skewed towards the smaller values.

More detailed information on the data set can be found in [57], or from the CRAWDAD website [58] where the data is available for downloading. For computational reasons, we do not use all the data for the experiments in this chapter, but select 500 random measurements from each of the six sensors.

Signal strength data. This data originates from a wireless sensor network deployed in a library building, where sensors measure radio frequency energy level (RSSI) on all 802.15.4 channels in the 2.4 GHz band [116, 117]. In essence, RSSI is an indication of the power level of a signal received by the antenna on the sensor node. The building has several collocated WIFI networks in normal

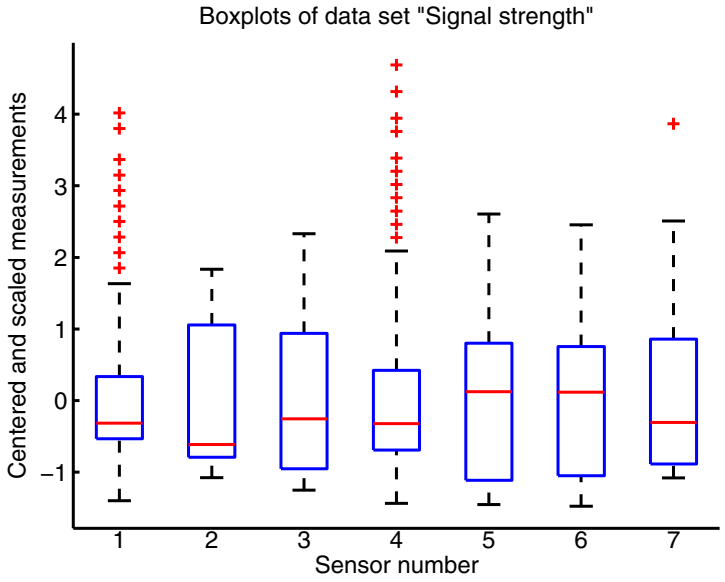

FIGURE 3.4: Boxplots of the seven sensors in the "Signal strength" data set.

operation that cause interference, so the sensor network is used to monitor the signal strengths on one location in this WIFI network. The sensor network consists of sixteen sensor nodes (each monitoring a single WIFI channel) of which we used only seven for computational reasons [25].

Again, we took 500 randomly selected measurements of each node to form this data set. The boxplots of the sensor values in this data set are in Figure 3.4. In contrast to the "Radiant light energy" data set, the measurements of the sensors in the "Signal strength" data set contain fewer possible outliers and are more evenly distributed. All possible outliers are positive values, corresponding to a strong incoming WIFI signal.

Decibel levels. This (proprietary) data set consists of five sensor nodes deployed in a kindergarten, one in each room of a single-story building, that are used to monitor the indoor climate. Among other parameters, the nodes measure decibel levels, and report these regularly to a central base station. We took 500 measurements from each decibel sensor on a day in May 2011 and included them in this data set. Figure 3.5 demonstrates that most sensors have fairly evenly distributed values, with several outliers on both sides of the median. However, kindergartens tend to be noisy rather than quiet, so most outliers are on the positive side of the median.

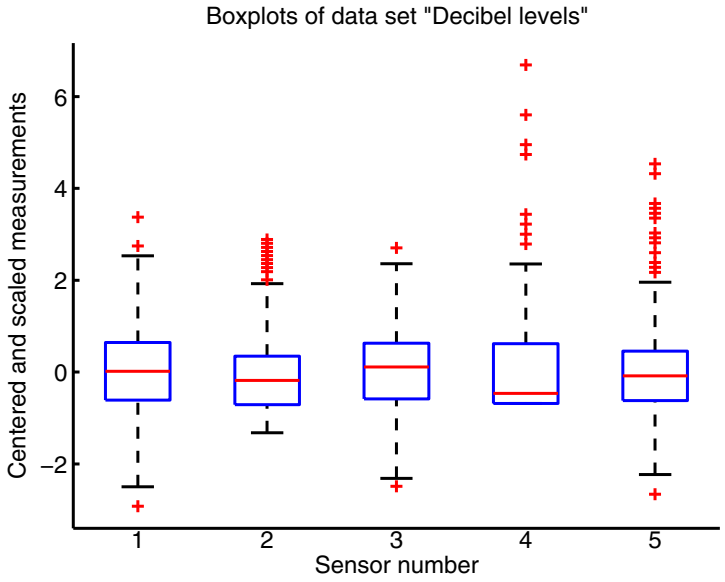

Figure 3.5: Boxplots of the five sensors in the "Decibel levels" data set.

3.5 Results and discussion

We apply the experimental setup of Section 3.3 to the DR-techniques of Section 3.2 and summarize the results in Table 3.1. This table contains the F1-score, Matthews Correlation, and Relative Information score for each combination of DR-technique and data set, where a high score implies that the technique preserves outliers well on that data set. Since MDS and t-SNE rely on a random initialization, we repeated the computation of scores 25 times, and reported the average and standard deviation of the results in Table 3.1. For the F1-scores, MDS achieves the best results, with values more than twice as large as those of PCA on the first and third data set. The scores of t-SNE are low, and suggest that it does not preserve outliers well. With the Matthews Correlation and Relative Information score we see similar results: MDS consistently attains high scores, PCA performs reasonably well on the first and third data set, and t-SNE has overall low scores.

We can visually inspect what happens to outliers after applying the three DR-methods. In Figures 3.6-3.8 we plot the low-dimensional version of the second data set "Signal Strength" (in circles), with outliers in the original high-dimensional data set marked by a triangle. Figure 3.6 demonstrates that

DR-technique	Light	Signal	Decibel
F1-score	*Avg(std)*	*Avg(std)*	*Avg(std)*
PCA	0.3333 (0.0000)	0.0000 (0.0000)	0.3529 (0.0000)
MDS	0.8316 (0.1076)	0.8705 (0.0828)	0.7212 (0.0739)
t-SNE	0.1067 (0.1587)	0.0073 (0.0364)	0.0352 (0.0718)
Matthews corr.			
PCA	0.3302 (0.0000)	-0.0149 (0.0000)	0.3477 (0.0000)
MDS	0.8439 (0.0907)	0.8767 (0.0754)	0.7374 (0.0617)
t-SNE	0.1389 (0.2146)	0.0033 (0.0472)	0.0497 (0.1170)
Rel. Inf. score			
PCA	0.7261 (0.0000)	-0.1295 (0.0000)	0.6398 (0.0000)
MDS	1.0197 (0.0465)	0.9118 (0.0299)	0.8409 (0.0238)
t-SNE	0.2583 (0.4720)	0.0020 (0.1364)	0.1228 (0.3021)

TABLE 3.1: F1-score ($\in [0,1]$), Matthews Correlation ($\in [-1,1]$), and Relative Information Score ($\in (-\infty,\infty)$) of each combination of DR-technique and data set. The reported values are the mean and standard deviation of 25 runs.

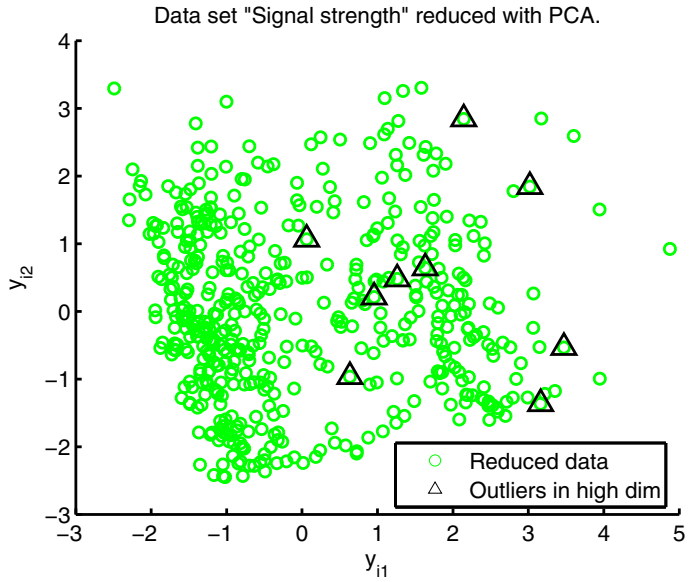

FIGURE 3.6: Data set "Signal strength" after dimensionality reduction with PCA (circles). The triangles mark the outliers that were found in the original high-dimensional data set.

most of the outliers are mapped to the interior of the reduced data set by PCA. In contrast, the low-dimensional data set in Figure 3.7 created by MDS has all high-dimensional outliers close to the boundary. Lastly, t-SNE also maps most outliers to the interior of the low-dimensional data set (shown in Figure 3.8), which illustrates its low scores.

By analyzing the objective of the three DR-techniques, we can explain the observed differences in performance. Firstly, PCA is a technique that focuses on preserving variance, so it only preserves outliers if they happen to be in a direction of high variance. Figure 3.1 from the introduction provides another illustration of what can happen to an outlier that is in a direction with low variance. The figure corresponds to reducing a two-dimensional data set to one dimension (the line) with PCA, and clearly demonstrates how the top-left outlier ends up in the center of the reduced data set.

MDS optimizes the squared stress optimization criterion in Eq. (3.4), which includes the term $||\mathbf{x}_i - \mathbf{x}_j||_M$. This term is the distance between two points $\mathbf{x}_i$ and $\mathbf{x}_j$, which is typically large when one of the points is an outlier. The criterion uses these distances to the power 4, so the outliers have a large effect on the squared stress criterion. Hence, minimizing these distances has a large positive effect on this criterion and thus MDS preserves outliers well.

The t-SNE technique optimizes the Kullback-Leibler divergence (3.8), which attaches high costs to nearby points in high-dimensional space (large p_{ij}) that are mapped to far away points in low-dimensional space (small q_{ij}). Hence, nearby points in high-dimensional space are kept nearby in low-dimensional space. This does not hold for points that are far away in high-dimensional space – outliers, which have low p_{ij} – as they are mapped to nearby points (with high q_{ij}) with low costs. So t-SNE tries to keep nearby points nearby and is therefore more suitable for preserving clusters than for preserving outliers.

From the analysis above we see that from the three selected methods, MDS achieves the highest scores and is best capable of preserving outliers. However, it is not necessarily the best DR-technique available, since many others exist in literature. In particular, the class of *supervised* DR-techniques (PCA, MDS, t-SNE are unsupervised) might provide methods with better performance than MDS. These techniques aim to reduce dimensionality while simultaneously trying to retain "sufficient information" for a classification task (which, in our case, would be retaining outliers). Hence, they could be applied to the scenario in this chapter, and possibly have good performance. Nevertheless, supervised DR-techniques are not included here, because we assume that the DR-techniques have no apriori knowledge about the outliers, and thus they are not suitable

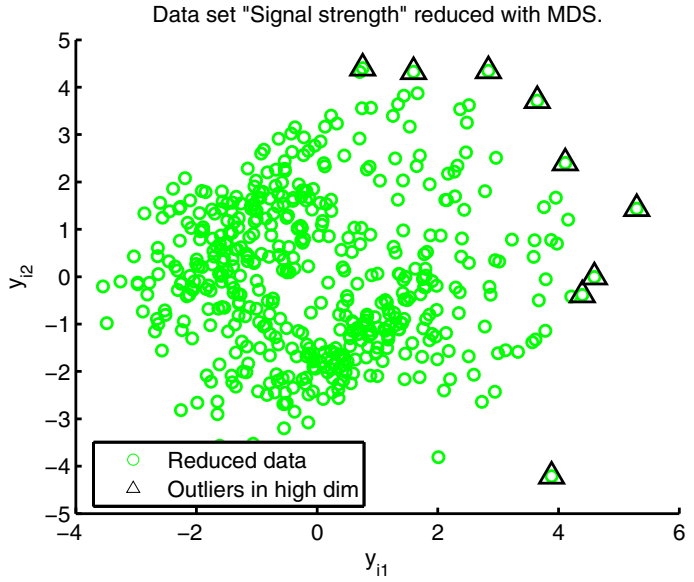

FIGURE 3.7: Data set "Signal strength" after dimensionality reduction with MDS (circles). The triangles mark the outliers that were found in the original high-dimensional data set.

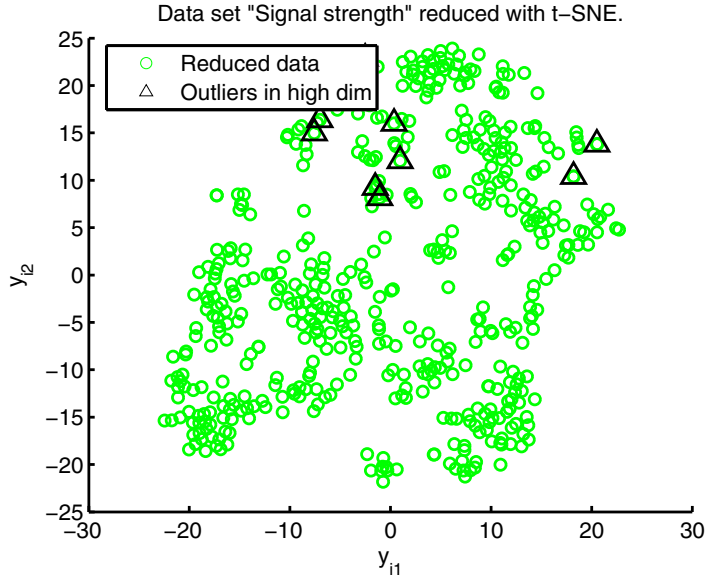

FIGURE 3.8: Data set "Signal strength" after dimensionality reduction with t-SNE (circles). The triangles mark the outliers that were found in the original high-dimensional data set.

for this chapter. Readers interested in supervised DR-techniques are referred to, e.g., [141].

The performance measures in this chapter are all based on the elements of the confusion matrix, which do not contain information about whether a point is a 'large' or 'small' outlier. Hence, with these scores we are not able to, e.g., find out which outlier has the large effect on a score. This 'binary' view of an outlier is, however, important for the scenario in the current chapter. Our motivation comes from applications where it is of critical importance to correctly identify an outlier after DR. If an outlier is no longer an outlier after DR, then it is useless for the application. Nevertheless, if this 'binary' approach can be relaxed from the point of view of the application, other scores might be more appropriate (see, e.g., [31]).

3.6 Conclusion

In this chapter we described three well-known DR-techniques (PCA, MDS, and t-SNE) and analyzed how well they are capable of preserving outliers. Based on three scores (F1-score, Matthews Correlation, and Relative Information score), and using three real-world data sets, we assessed the performance of each method on each data set. The resulting analysis demonstrates that, among the three described DR-methods, MDS is best at preserving outliers. It consistently achieves the highest scores, and performs significantly better than both PCA and t-SNE. In the discussion, we explain that this difference in performance is caused by the specific objectives of the techniques: PCA tries to preserve variance, MDS preserves large distances (i.e., outliers), and t-SNE preserves clusters. In general, we recommend that the dimensionality reduction technique is chosen with the intended application in mind. For outlier detection MDS is a good choice, PCA is designed for preserving variance, and for preserving clusters t-SNE is a good choice. Future research includes investigating specific types of dimensionality reduction (e.g., supervised DR-methods, real-time DR-methods), and how they are affected by outliers.