5.4 – Anomalies in High-Dimensional Datasets

In recent times, datasets have become quite **large** – they may contain **hundreds or thousands of features** (or more).

Conventional **proximity-based** anomaly detection methods can only be expected to work reasonably well when the sample size n is larger than the dimension p ($n > p$).

In **high-dimensional data** ($n < p$), the main problem is an off-shoot of the **curse of dimensionality** (CoD): observations are often **isolated** and **scattered** (or sparse); the notion of proximity fails to maintain its relevance.

High-dimensional anomaly detection methods are linked with dimension reduction and feature selection methods.

5.4.1 – Definitions and Challenges

The challenges of anomaly and outlier detection in **high-dimensional data** (HDD) are due to:

- the notion of distance failing to retain its **relevance** due to the CoD ("the problem of detecting outliers is like finding a needle in a haystack");
- all points in HDDs **tend to be outliers**, and
- datasets become more **sparse** as the dimension of the feature space increases.

Good HDD anomaly detection methods should:

- allow for **effective management** of sparse data issues;
- provide **interpretability** of the discrepancies (i.e. how is the behaviour of such observations different than the behaviour from regular ones?);
- allow anomaly measurements to be **compared** ("apples-to-apples"), and
- consider the **local data behaviour** to determine whether an observation is abnormal or not.

5.4.2 – Projection-Based Methods

One approach to mitigate the effects of the CoD on conventional anomaly/outlier detection methods in **high dimension, low sample size** (HDLSS) datasets is to **reduce the dimensionality** of the dataset while preserving its essential characteristics.

Such projecion-based methods include:

- **principal component analysis**,
- **independent component analysis**,
- **feature selection**, etc. – see *Feature Selection and Dimension Reduction* module/report for more examples.

Principal Components Analysis

Principal component analysis (PCA) can be used to find the combinations of variables along which the data points are **most spread out**.

Geometrically, the procedure fits the "best" $p-$**ellipsoid** to a centered representation of the data.

The ellipsoid axes are the **principal components** of the data.

Small axes are components along which the variance is "small"; removing these components can lead to a "small" loss of information.

There are scenarios where it could be those "small" axes that are more interesting – such as the "pancake stack" problem.

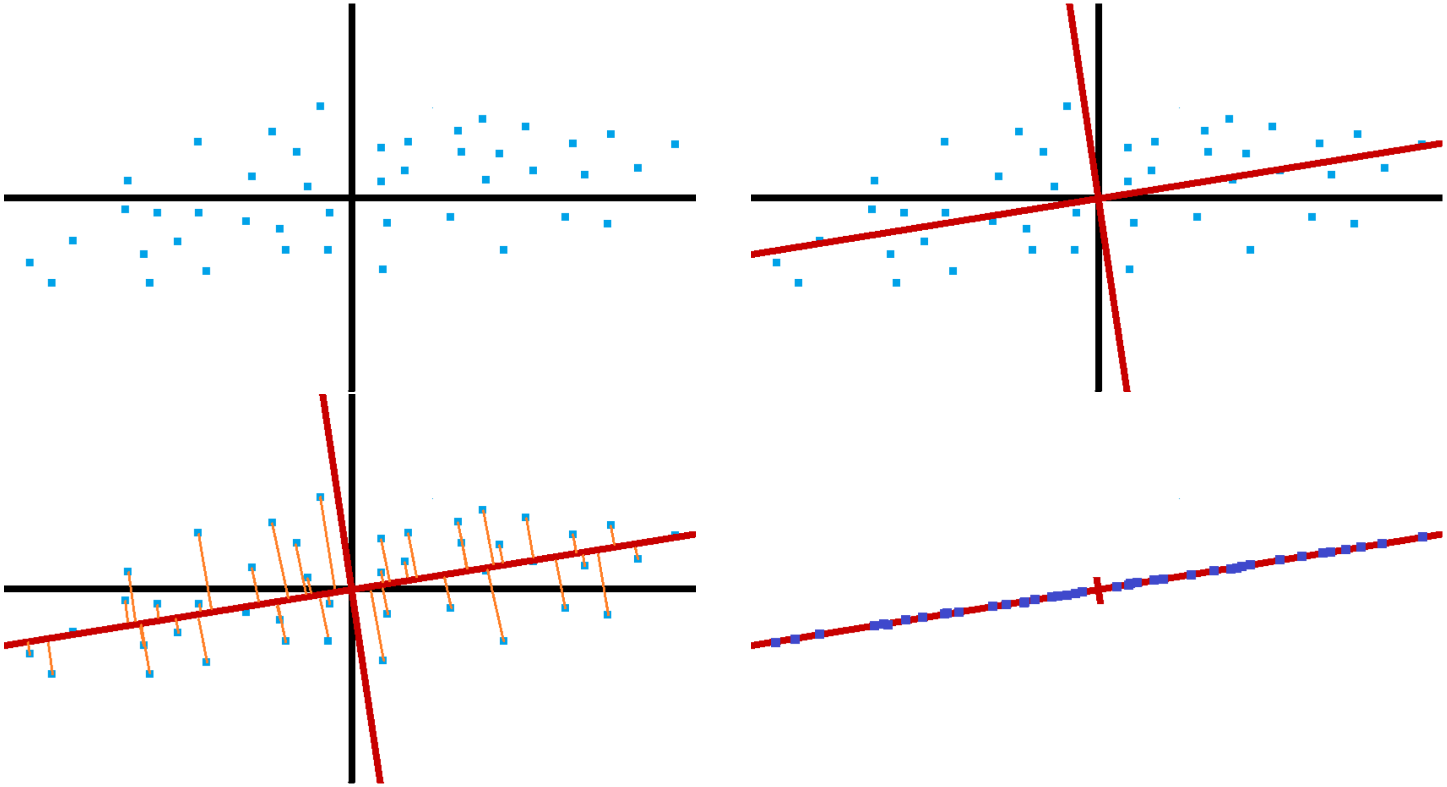


Illustration of PCA on an artificial 2D dataset. The red axes provide the best elliptic fit. Removing the minor axis by projecting the points on the major axis leads to dimension reduction and a (small) loss of information.

PCA Procedure:

1. centre and “scale” the data to obtain a matrix $\mathbf{X}$;

2. compute the data’s “covariance matrix” $\mathbf{K} = \mathbf{X}^\top \mathbf{X}$;

3. compute $\mathbf{K}$’s eigenvalues, $\mathbf{\Lambda}$ (ordered diagonal matrix), and its orthonormal eigenvectors matrix $\mathbf{W}$;

4. each eigenvector $\mathbf{w}$ (also known as **loading**) represents an axis, whose variance is given by the associated eigenvalue λ.

Note that $\mathbf{K} \geq 0 \implies \mathbf{\Lambda} \geq 0$.

The **first principal component** PC_1 is the eigenvector $\mathbf{w}_1$ of $\mathbf{K}$ associated to its largest eigenvalue λ_1, and the variance of the data along $\mathbf{w}_1$ is proportional to λ_1.

The **second principal component** PC_2 is the eigenvector $\mathbf{w}_2$ of $\mathbf{K}$ associated to its second largest eigenvalue $\lambda_2 \leq \lambda_1$, and the variance of the data along $\mathbf{w}_1$ is proportional to λ_2, and so on.

Final Result: $r = \mathrm{rank}(\mathbf{X})$ **orthonormal** principal components

$$\mathrm{PC}_1, \ldots, \mathrm{PC}_r.$$

If some of the eigenvalues are 0, $r < p$, and *vice-versa* $\Longrightarrow$ data is embedded in a $r-$dimensional subspace in the first place.

PCA can provide an avenue for dimension reduction by “removing” components with small eigenvalues.

The **proportion of the spread in the data** which can be explained by each PC can be placed in a **scree plot** (eigenvalues against ordered component indices), and retain the ordered PCs:

- for which the eigenvalue is above some threshold (say, 25%);
- for which the cumulative proportion of the spread falls below some threshold (say 95%), or
- prior to a **kink** in the scree plot.

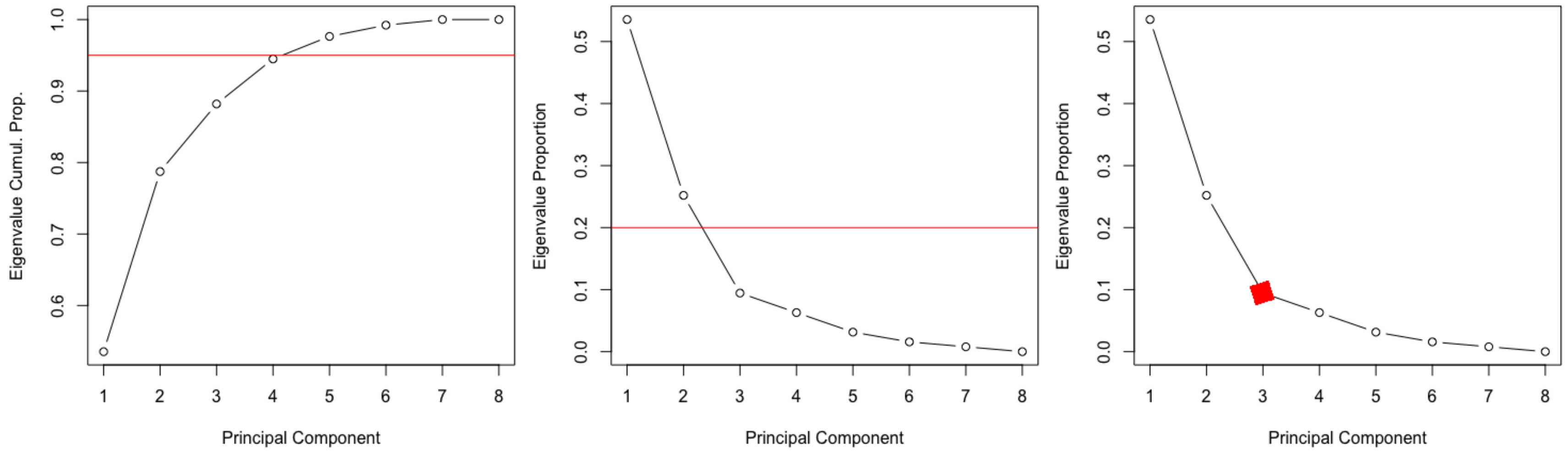
Example: consider an $8D$ dataset for which the ordered PCA eigenvalues are

PC	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Var	17	8	3	2	1	0.5	0.25	0
Prop	54	25	9	6	3	2	1	0
Cumul	54	79	88	94	98	99	100	100

If only the PCs that explain up to 95% of the **cumulative variance** are retained, the original dataset reduces to a $4D$ subset.

If only the PCs that **individually explain** more than 25% of the variance are retained, the original dataset reduces to a $2D$ subset.

If only the PCs that lead into the **first kink** in the scree plot are retained, the original dataset reduces to a $3D$ subset.

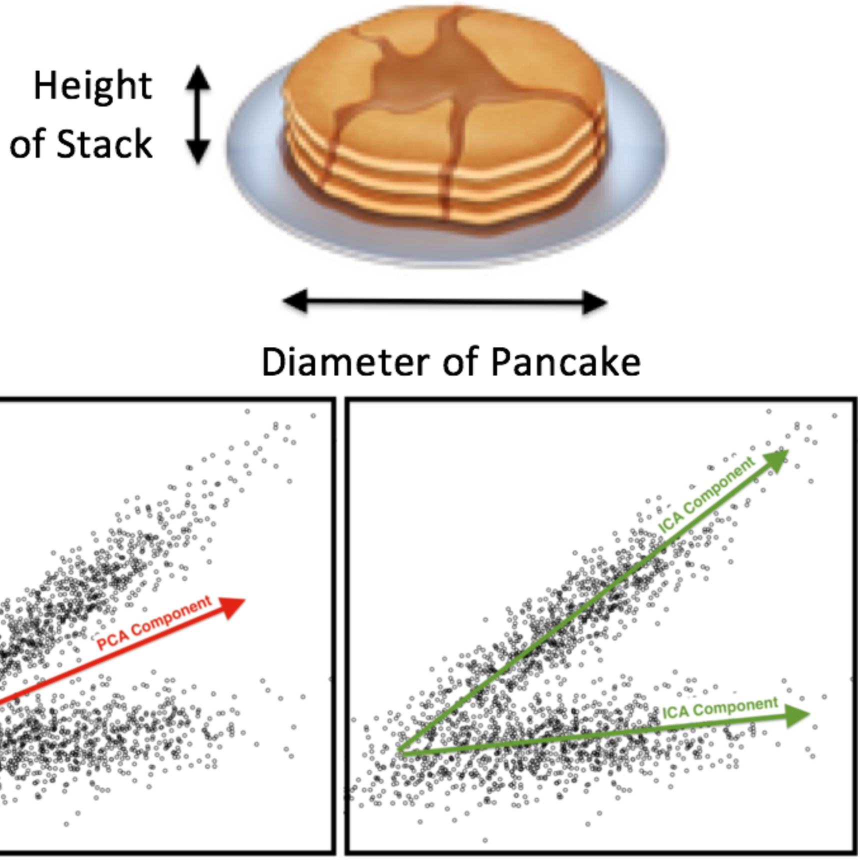


The proportion of the variance explained by each (ordered) component is shown in the first 3 charts; the cumulative proportion is shown in the last chart.

The cumulative proportion method is shown in the first image, the individual threshold method in the second, and the kink method in the third.

PCA Limitations:

- dependent on scaling, and so not unique;
- interpreting the PCs require domain expertise;
- (quite) sensitive to outliers;
- analysis goals not always aligned with the PCs, and
- data assumptions not always met – does it always make sense that important data structures and data spread be linked (see **counting pancakes** problem), or that the PCs be **orthogonal**?



(algobeans.com)

PCA is used in various contexts:

- as a dimension reduction method used during data pre-processing;
- as a data visualization aid, and
- as an anomaly and outlier detection method.

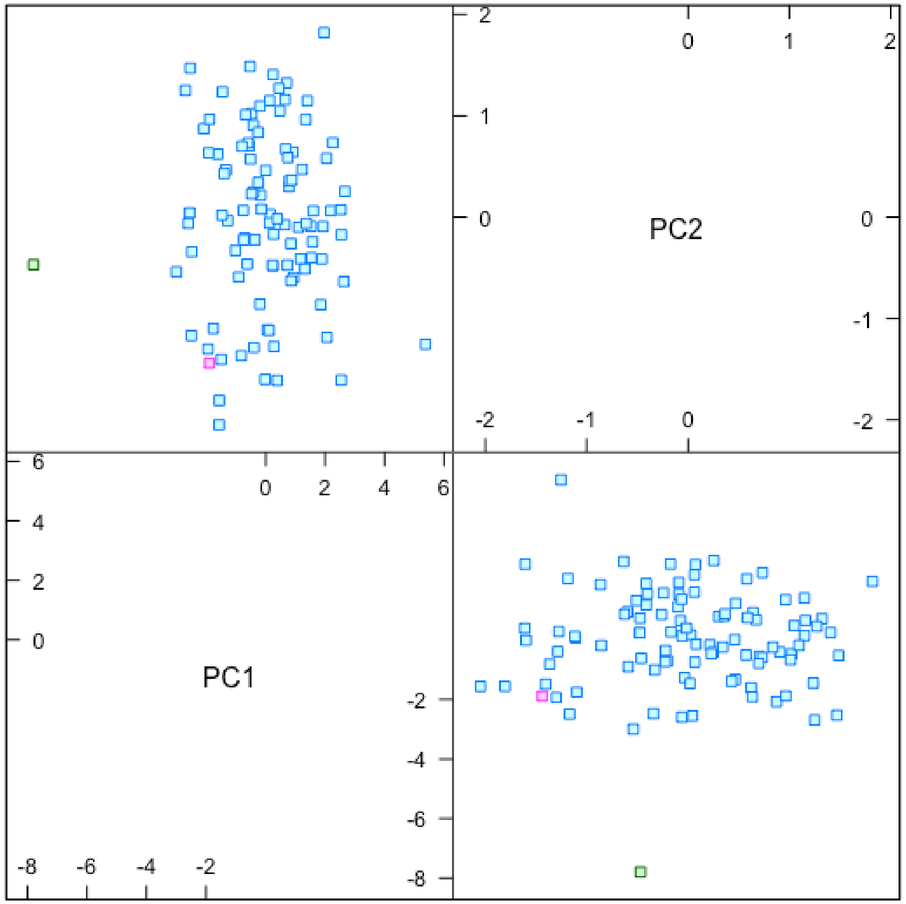
The **quality** of the PCA results is strongly dependent on the **number of retained principal components** ($=$ the dimension k of the subspace on which the observations are projected).

For anomaly detection purposes, it is not obvious that the methods shown prior to find an optimal k are appropriate $\Longrightarrow$ a good k is one which allows for good anomaly detection.

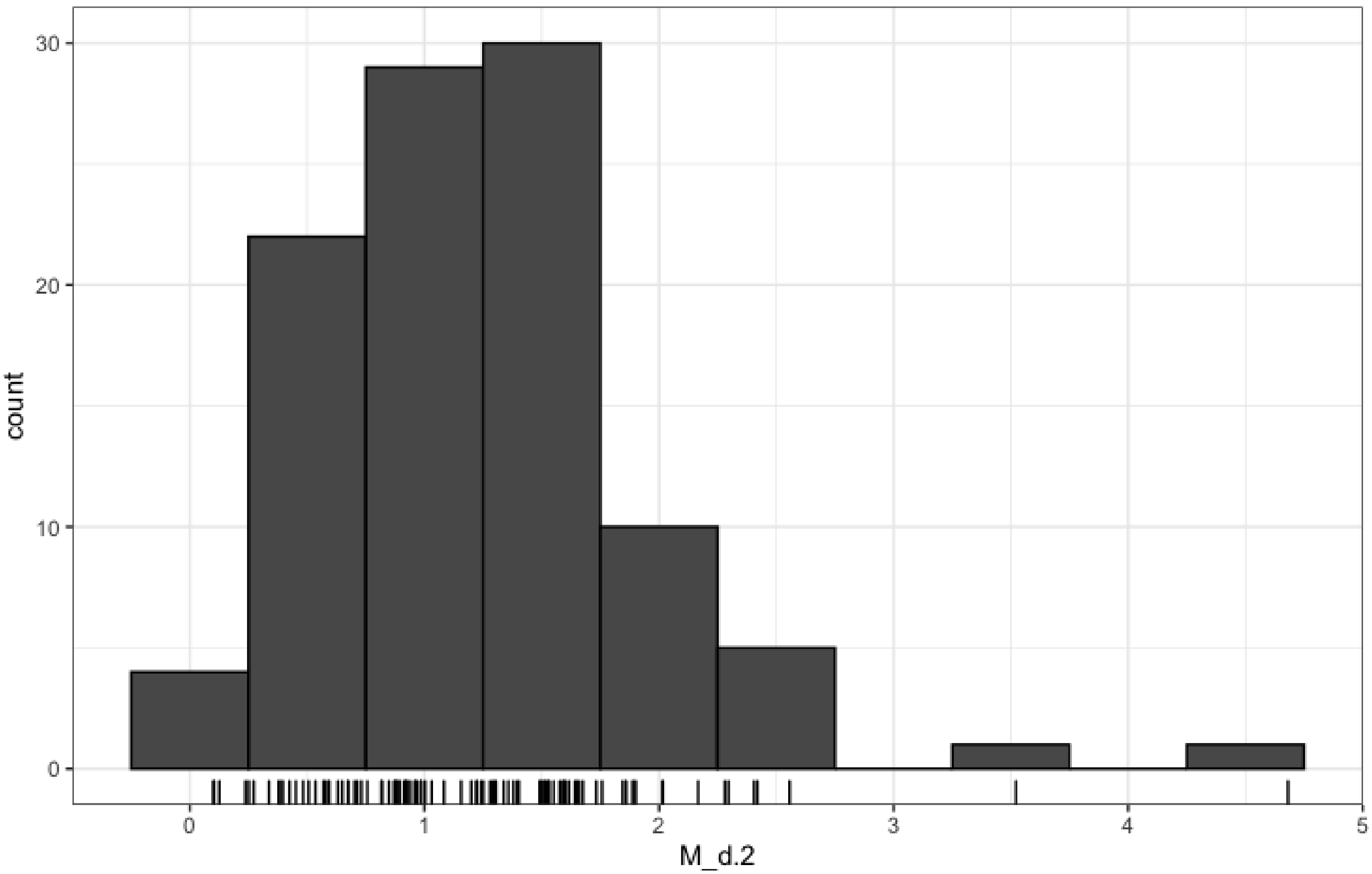
There are other PCA-associated dimension reduction methods: ICA, singular value decomposition, kernel PCA, etc.

What is the link with anomaly and/or outlier detection?

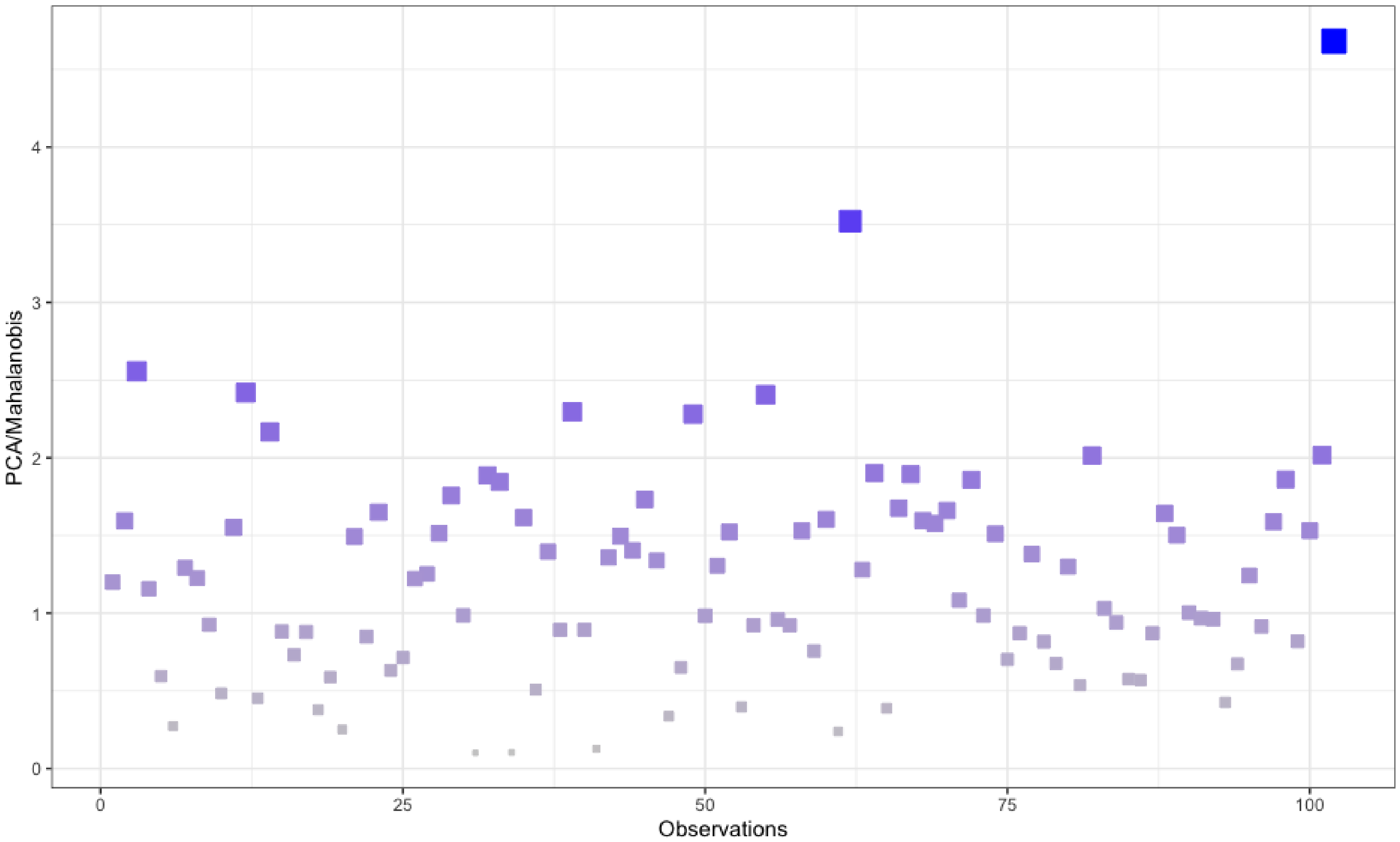
- Once the dataset has been projected on a lower-dimensional subspace, the curse of dimensionality is **mitigated** – traditional methods are applied to the **projected data**.
- Dimension reduction usually leads to a loss of information, which can affect the accuracy of the detection procedure – especially if the presence/absence of anomalies is **not aligned** with the dataset's principal components.



2D PCA projection.



Histogram of 2D PCA Mahalanobis scores.


PCA/Mahalanobis
0
1
2
3
4
0
25
50
75
100
Observations

Distance-Based Outlier Basis Using Neighbours

Main problem with using PCA-type projections for anomaly detection: there may not be a correlation between the axes of heightened variance and the presence or absence of anomalies.

The **distance-based outlier basis using neighbours** algorithm (DOBIN) builds a basis which is better suited for the eventual detection of outlying observations. DOBIN's main idea is to search for nearest neighbours that are in fact relatively distant from one another:

1. start by building a space $\mathbf{Y} = \{\mathbf{y}_\ell\}$ which contains $M \ll n(n+1)/2$ vectors of the form

$$\mathbf{y}_\ell = (\mathbf{x}_i - \mathbf{x}_j) \odot (\mathbf{x}_i - \mathbf{x}_j),$$

where $\odot$ is the element-by-element Hadamard multiplication, and for which the $1-$norm

$$\|\mathbf{y}_\ell\|_1 = (x_{1,1} - x_{2,1})^2 + \cdots + (x_{1,p} - x_{2,p})^2$$

is the square of the distance between $\mathbf{x}_i, \mathbf{x}_j \in \mathbf{X}$;

2. the selection of each of the M observation pairs is made according to a complex procedure which only picks $\mathbf{x}_i$ and $\mathbf{x}_j$ if they are part of one another's $k-$neighbourhood, for $k \in \{k_1, \ldots, k_2\}$;

3. the set $\mathbf{Y}$ thus contains points for which $\|\mathbf{y}_\ell\|_1$ is relatively large, which is to say that the observations $\mathbf{x}_i$ are $\mathbf{x}_j$ fairly distant from one another even if they are $k-$neighbours of each other;

4. next, a basis $\{\eta_1, \ldots, \eta_p\} \subset \mathbb{R}^p$ is built where each η_i is a unit vector given by a particular linear combination of points in $\mathbf{Y}$; they can be found using a Gram-Schmidt-like procedure:

$$\mathbf{y}_{\ell_0} = \mathbf{y}_\ell, \quad \ell = 1, \ldots, M$$

$$\mathbf{y}_{\ell_{b-1}} = \mathbf{y}_{\ell_{b-2}} - \langle \eta_{b-1} \mid \mathbf{y}_{\ell_{b-2}} \rangle, \quad \ell = 1, \ldots, M$$

$$\eta_b = \frac{\sum_{\ell=1}^{M} \mathbf{y}_{\ell_{b-1}}}{\left\| \sum_{\ell=1}^{M} \mathbf{y}_{\ell_{p-1}} \right\|_2},$$

for $b = 1, \ldots, p$;

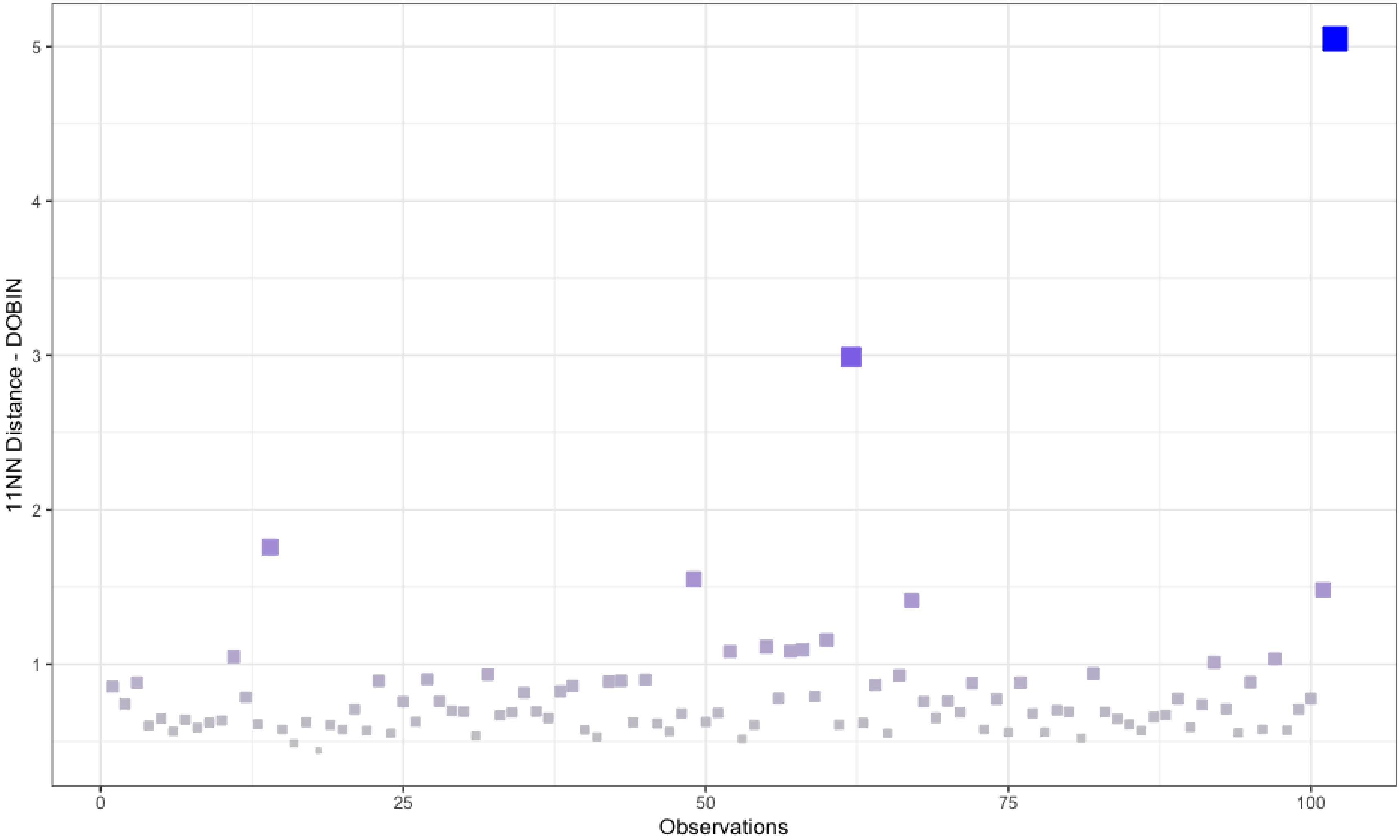
5. finally, transform the original dataset $\mathbf{X}$ according to $\hat{\mathbf{X}} = \mathcal{T}(\mathbf{X})\Theta$, where $\mathcal{T}(\mathbf{X})$ normalizes each feature of $\mathbf{X}$ according to a problem-specific scheme (Min-Max or Median-IQR, say), and

$$\Theta = [\eta_1 \mid \cdots \mid \eta_p]$$

   is an orthogonal $p \times p$ matrix.

$\hat{\mathbf{X}}$ plays an analogous role to the subspace projection of $\mathbf{X}$ in PCA – this is the object on which outlier and anomaly detection algorithms are applied.

In a nutshell, the first component provides the direction of largest kNN distance, instead of the direction of largest variance, and so forth. The paper by Kandanaarachchi and Hyndman provide more details.


11NN Distance - DOBIN
5
4
3
2
1
0
25
50
75
100
Observations

5.4.3 – Subspace Methods

Subspace methods have been used effectively for anomaly and outlier detection in high-dimensional datasets.

In this context, a **subspace** is obtained by projecting the original dataset D on some collection of its features (looking at D only along some of its axes).

Strengths:

- eliminates **additive noise effects** of HDD;
- leads to more robust outliers (identified as such even when using different methods).

Limitation:

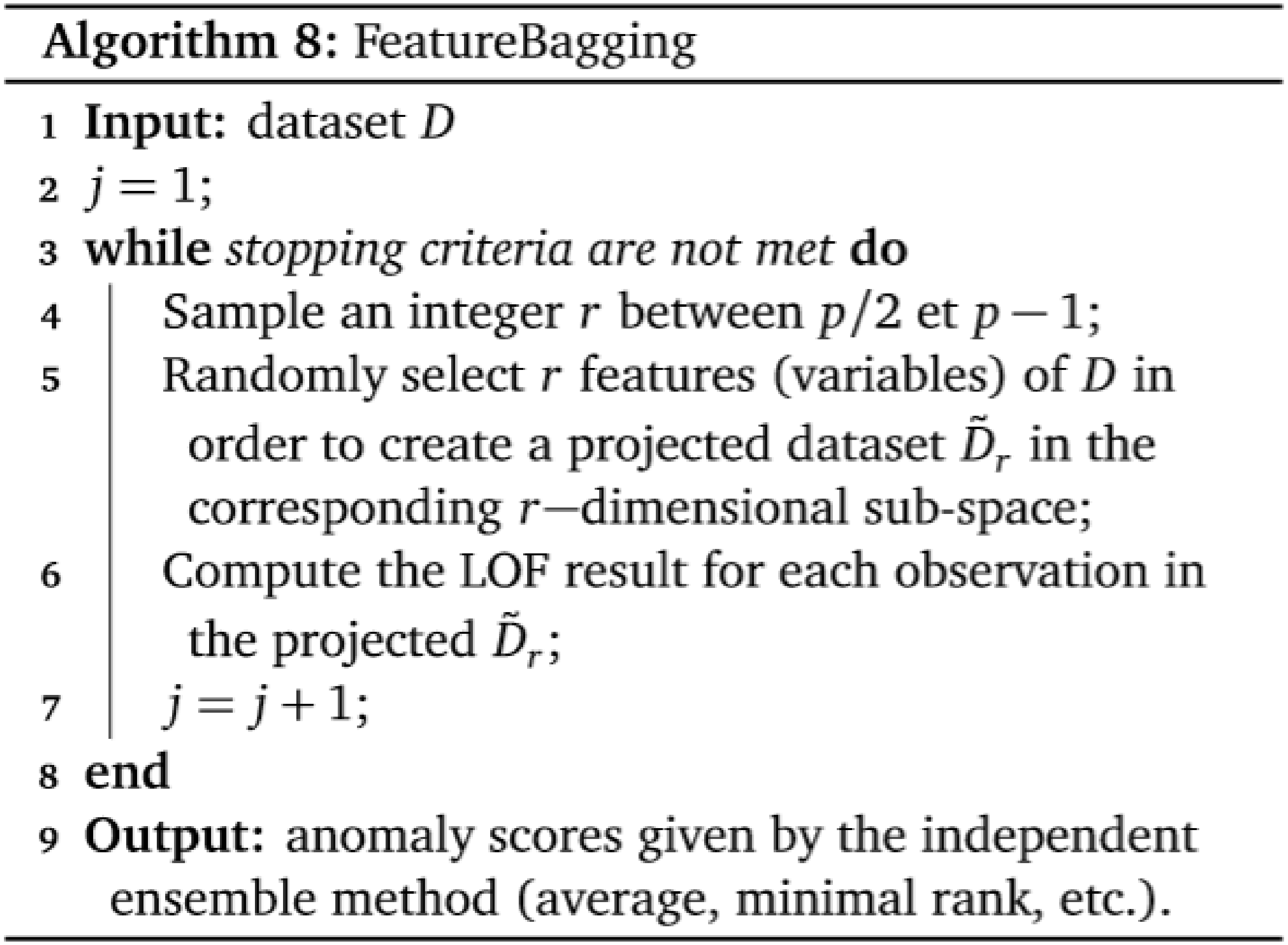
- difficult to solve effectively and efficiently $\Longrightarrow$ the potential # of subspace projections is exponential in # of features.

Feature bagging (FB) combines the results of the anomaly detection algorithm applied to **various subspaces** of the original data.

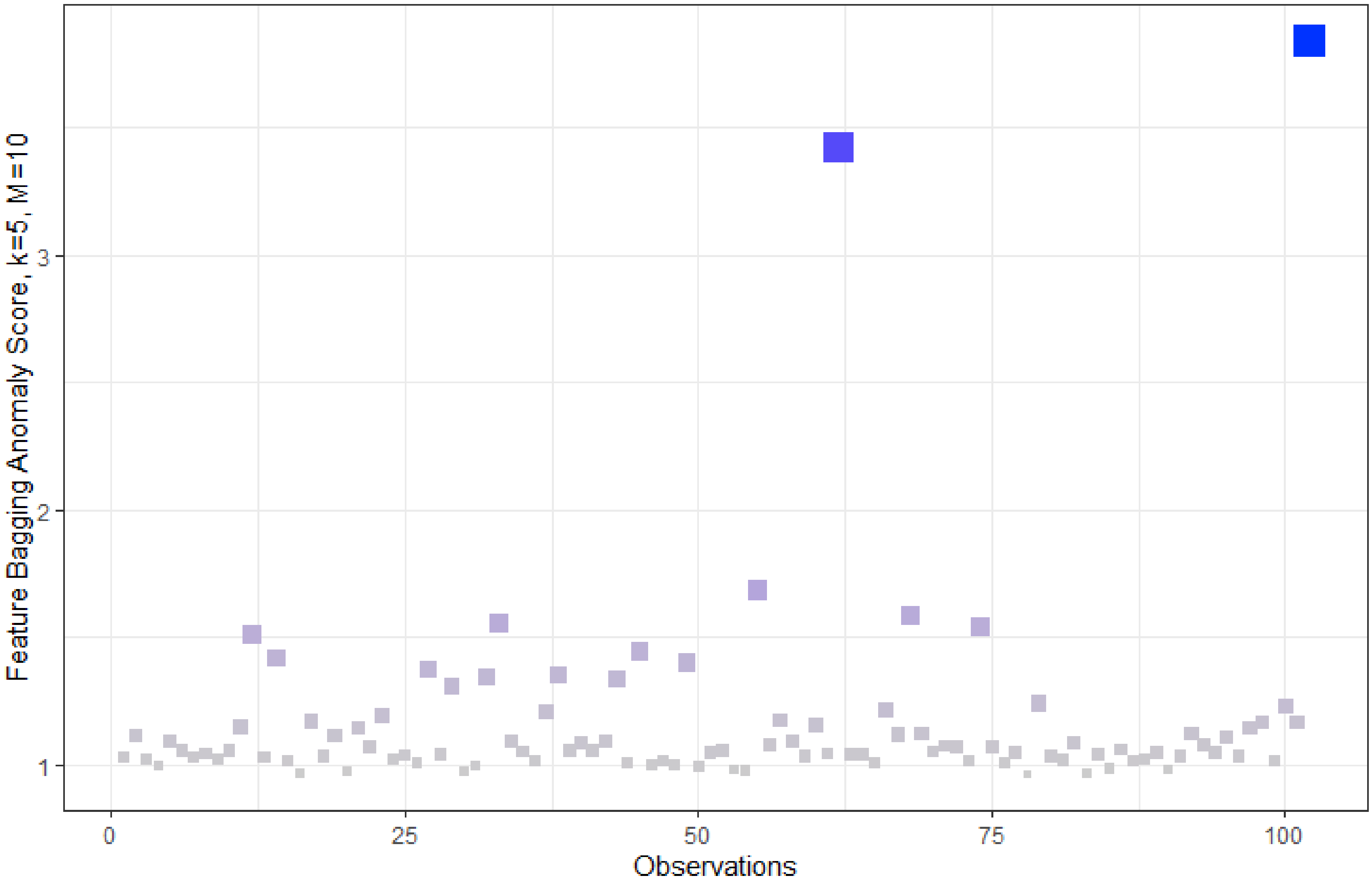
Officially, FB uses the Local Outlying Factor algorithm (LOF), but any fast anomaly detection algorithm can be used instead.

The anomaly scores and rankings from each run are aggregated as they are in the Independent Ensemble approach (cf. last Section of the slides).

The formal procedure is provided in Algorithm 8.



```
Algorithm 8: FeatureBagging
Input: dataset $D$
$j = 1$;
while stopping criteria are not met do
    Sample an integer $r$ between $p/2$ et $p-1$;
    Randomly select $r$ features (variables) of $D$ in order to create a projected dataset $\tilde{D}_r$ in the corresponding $r-$dimensional sub-space;
    Compute the LOF result for each observation in the projected $\tilde{D}_r$;
    $j = j + 1$;
end
Output: anomaly scores given by the independent ensemble method (average, minimal rank, etc.).
```


Feature Bagging Anomaly Score, k=5, M=10
1
2
3
0
25
50
75
100
Observations

Other, more sophisticated, subspace anomaly detection methods include:

- **High-dimensional Outlying Subspaces** (HOS);
- **Subspace Outlier Degree** (SOD), implemented in `HighDimOut`;
- **Projected Clustering Ensembles** (OutRank);
- **Local Selection of Subspace Projections** (OUTRES), etc.

⚠ “**No Free Lunch” Theorem:** there is no magic method – all methods have strengths and limitations, and the results depend heavily on the data.

Another list of algorithms is found at `https://pyod.readthedocs.io/en/latest/`
it is far from complete.