

## 5.5 – Advanced Topics

Anomaly detection and outlier analysis is still a young field, with a very active research community.

The challenges are numerous (we have highlighted some of them), and new algorithms come out nearly monthly.

An application to time series (using the S&P 500) is provided in the accompanying report, as are suggested exercises and projects (Airline Data, Distracted Driving Fatalities Data, Houseprice Data, etc.).

We wrap up this module with a discussion of outlier ensembles and of anomalies in text data.

## 5.5.1 – Outlier Ensembles

We have looked at various anomaly detection algorithms whose relative performance varies with the type of data being considered.

The **No Free Lunch theorem** reminds us that there is no specific algorithm that is guaranteed to outperform every other algorithm for all datasets.

The impact of algorithmic mismatch can be mitigated by using **ensemble methods**, where the results of several algorithms are considered before making a final decision.

We discuss two types of ensemble methods: **sequential ensembles** (boosting) and **independent ensembles**.

## Sequential Ensembles

In **sequential ensembles**, a baseline algorithm is applied to a dataset in a sequential manner.

At each step, the weight associated with each observation is modified according to the preceding results using some “boosting” method (such as AdaBoost or XGBoost, for instance).

The final result is either some weighted combination of all preceding results, or simply the outputs of the last step in the sequence (see *Boosting with AdaBoost and Gradient Boosting*, on the Data Action Lab blog).

The formal procedure is provided in Algorithm 6.

---

**Algorithm 6: SequentialEnsemble**

---

- 1 **Inputs:** dataset  $D$ , base algorithms  $A_1, \dots, A_r$
  - 2  $j = 1$ ;
  - 3 **while** *stopping criteria are not met* **do**
  - 4     Select an algorithm  $A_j$  based on the results from the preceding steps;
  - 5     Create a new dataset  $D_j$  from  $D$  by modifying the weight of each observation based on the results from the preceding steps;
  - 6     Apply  $A_j$  to  $D_j$ ;
  - 7      $j = j + 1$ ;
  - 8 **end**
  - 9 **Output:** anomalous observations obtained by weighing the results of all previous steps
-

## Independent Ensembles

In **independent ensembles**, different algorithms (or different instantiations of one algorithm) to the dataset (or some **resampled** dataset).

Choices made at the data and algorithm level are **independent** of preceding runs results (in comparison with sequential ensembles). The results are then combined to obtain more robust outliers

Every base anomaly detection algorithm provides an **anomaly score** (or an abnormal/regular classification) for each observation in  $D$ ; observations with higher scores are considered more anomalous than observations with lower scores.

The formal procedure is provided in Algorithm 7.

---

**Algorithm 7: IndependantEnsemble**

---

```
1 Inputs: dataset  $D$ , base algorithms  $A_1, \dots, A_r$ 
2  $j = 1$ ;
3 while stopping criteria are not met do
4   |   Select an algorithm  $A_j$ ;
5   |   Create a new dataset  $D_j$  from  $D$  by (potential)
   |   re-sampling, but independently of the
   |   preceding steps' results;
6   |   Apply  $A_j$  to  $D_j$ ;
7   |    $j = j + 1$ ;
8 end
9 Output: anomalous observations obtained by
   combining the results of all previous steps
```

---

Many combination techniques are used in practice:

- **majority vote,**
- **average,**
- **minimal rank, etc.**

Let  $\alpha_i(\mathbf{p})$  and  $r_i(\mathbf{p})$  represent the (normalized) **anomaly score** and the **anomaly rank** of  $\mathbf{p} \in D$  according to algorithm  $A_i$ , respectively. The smaller the anomaly score, the smaller the anomaly rank, and *vice-versa*.

Anomaly scores lie between 0 (unlikely to be an anomaly) to 1 (likely to be an anomaly); ranks range from 1 to  $n$  (the number of observations, with ties allowed).

If the base detection algorithms are  $A_1, \dots, A_m$ , the average anomaly score and the minimal rank of an observation  $\mathbf{p} \in D$  according to the independent ensemble method, say, are respectively

$$\alpha(\mathbf{p}) = \frac{1}{m} \sum_{i=1}^m \alpha_i(\mathbf{p}) \quad \text{and} \quad r(\mathbf{p}) = \min_{1 \leq i \leq m} \{r_i(\mathbf{p})\}.$$

If  $n = m = 3$ , for instance, we could end up with

$$\alpha_1(\mathbf{p}_1) = 1.0, \alpha_1(\mathbf{p}_2) = 0.9, \alpha_1(\mathbf{p}_3) = 0.0;$$

$$\alpha_2(\mathbf{p}_1) = 1.0, \alpha_2(\mathbf{p}_2) = 0.8, \alpha_2(\mathbf{p}_3) = 0.0;$$

$$\alpha_3(\mathbf{p}_1) = 0.1, \alpha_3(\mathbf{p}_2) = 1.0, \alpha_3(\mathbf{p}_3) = 0.0.$$



Using the mean as the combination techniques, we obtain

$$\alpha(\mathbf{p}_1) = 0.7, \alpha(\mathbf{p}_2) = 0.9, \alpha(\mathbf{p}_3) = 0.0, \implies \mathbf{p}_2 \succeq \mathbf{p}_1 \succeq \mathbf{p}_3 :$$

$\mathbf{p}_2$  is more anomalous than  $\mathbf{p}_1$ , which is itself more anomalous than  $\mathbf{p}_3$ .

Consequently,


$$r_1(\mathbf{p}_1) = 1, r_1(\mathbf{p}_2) = 2, r_1(\mathbf{p}_3) = 3;$$

$$r_2(\mathbf{p}_1) = 1, r_2(\mathbf{p}_2) = 2, r_2(\mathbf{p}_3) = 3;$$

$$r_3(\mathbf{p}_1) = 2, r_3(\mathbf{p}_2) = 1, r_3(\mathbf{p}_3) = 3,$$

and under the minimal rank method, we obtain

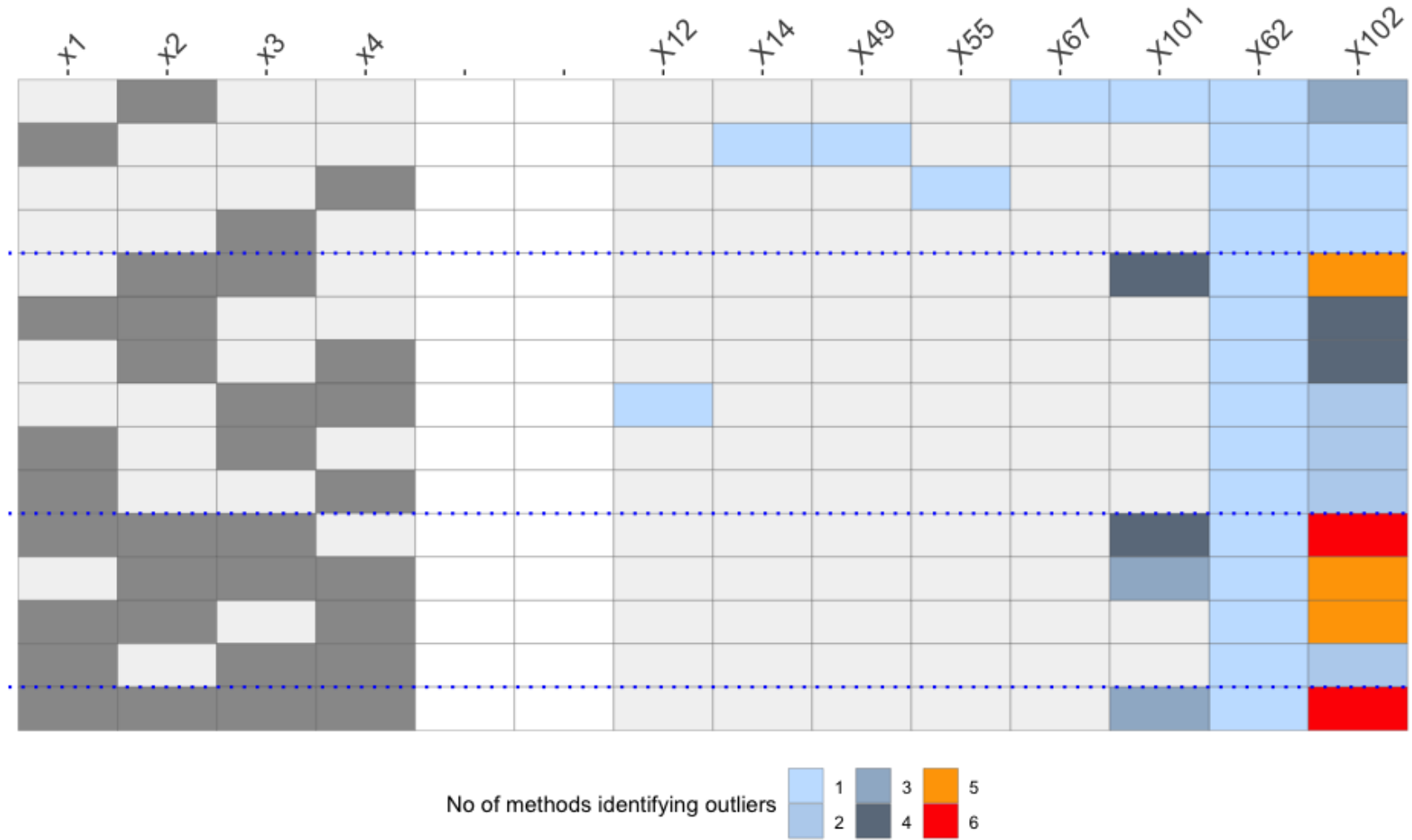
$$r(\mathbf{p}_1) = r(\mathbf{p}_2) = 1, r(\mathbf{p}_3) = 3, \implies \mathbf{p}_1 \succeq \mathbf{p}_3 \text{ and } \mathbf{p}_2 \succeq \mathbf{p}_3.$$

 In general, the results not only depend on the dataset under consideration and on the base algorithms that are used in the ensemble, **but also on how they are combined.**

For HDLSS data, ensemble methods can sometimes allow the analyst to mitigate some of the effects of the curse of dimensionality by selecting **fast** baseline algorithms (which can be run efficiently multiple times) and focusing on building robust relative anomaly scores through combination.

Another suggestion: use a **different sub-collection** of the original dataset's features at each step, in order to **de-correlate** the base detection models.

Even without combining the results, it may be useful to run multiple algorithms on different subspaces to produce an **Overview of Outliers (O3)**.



The columns on the left indicate the **subspace variables** (see row colouring).

The columns on the right indicate which **observations were identified as an outlier** by at least 1 method (HDoutliers, FastPCS, mvBACON, adjOutlyingness, DectectDeviatingCells, covMCD) in at least 1 subspace.

The **colours** depict the number of methods that identify each observation in each subspace as an outlier.

Observation 102 is identified as an outlier by 6 methods in 2 subspaces, 5 methods in 3 subspaces, 4 methods in 2 subspaces, 3 methods in 1 subspace, 2 methods in 4 subspaces, and 1 method in 3 subspaces – it is clearly the **most anomalous** observation in the dataset.

Observations 62 and 101 are also commonly identified as outliers.