0. $\alpha \in A$, Z^α for $t = T + 1, ..., T + L$:

For each $\alpha \in A$:

1. $\tilde{z}^\alpha_t$ – form truncated z vector for point $T + 1$ for pattern α
2. $S^\alpha_t = \{z_i[-1] \mid \rho(\tilde{z}^\alpha_i,\ \tilde{z}^\alpha_t) < \varepsilon\}$
3. $S_t = \cup_\alpha S^\alpha_t$
4. $\hat{y}^\omega_t = \text{mean}(S_t) + \mathcal{N}(0, \sigma^2)$

Note that steps 1 and 2 generate a permuted trajectory.

Repeat this M times, where M is the number of new trajectories

For $t = T + 1, ..., T + L$:

1. For set of possible predicted values $S_t = \left\{\hat{y}^{(i)}_t\right\}^M_{i=1}$
2. Apply an algorithm for point classification

For $t = T + 1, ..., T + L$:

For each $\alpha \in A$:

1. $\tilde{z}^\alpha_t$
2. $S^\alpha_t = \{z^\alpha_i[-1] \mid \rho(\tilde{z}_i,\ \tilde{z}^\alpha_t) < \varepsilon\}$
3. $S_t = \cup_\alpha S^\alpha_t$
4. Cluster S_t into $C_1, C_2, ..., C_l$. To overcome computational problems use l^* largest clusters.
5. Set predictions $\left\{\hat{y}^{(1)}_t, ..., y^{(l)}_t\right\}$, where $\hat{y}^{(i)}_t = \text{mean}(C_i)$.

IMPORT THE PICTURE (abount training T_1, T_2, etc.), VERY IMPORTANT!!!

DBSCAN algorithm

1. Parameters

- ε

- minPts – minimum number of points in ε-neighbourhood to determine the core point.

2. Point types

- core point – a point that hase at least minPts points inside of its ε-neighbourhood including itself.

- border point – a point with less than minPts points in its ε-neighbourhood but is reachable from a core point

- noise point – every other point.

3. Reachability

Def. q is directly reachable from p if q is in its ε-neighbourhood.

Def. q is reachable from p if there exists a chain of points that includes p and q such that each next point is directly reachable from a previous one.

4. Algorithm

1. Initialization.

- All points are marked as unvisisted
- An empty list of clusters is created.

2. Marking the points one-by-one.

- For each point p:

    - if p is visited → skip

    - mark p as visited

    - calculate the number of points in ε-neighbourhood of p (including p itself).

    - if the # of neighbours $<$ minPts → mark p as noisy point

    - otherwise:

        - create a new cluster;

- add all neighbours to this cluster;

- recuresively grow the cluster by adding points reachable from the core points.

3. CLuster expansion.

- For each point q in the current cluster

  - if q is not visited → mark it as visited, find its ε-neighbourhood and if it contains at least minPts points, add them to the cluster too.

5. Advantages

- DBSCAN can find clusters of any shape
- It does not require setting the number of clusters
- It is robust to outliers

6. Disadvantages

- This algorithm is very sensitive to its hyperparameter values
- The results vary depending in distance metric you choose
- Does not work well if data has differing cluster densities

7. Hyperparameter selection

- minPts is usually selected to be $\geq \#$ of features $+ 1$
- ε can be selected base on graph of distances to k-th nearest neighbours: it is mostly selected as the point of a sharp bend in the graph.

Density and graph clusterization algorithm

Wishart algorithm modified by Lapko nad Chentsov.

1. Density estimation

$$p(x) = \frac{K}{V_{K(x)} \cdot n},$$

where:

$V_{K(x)}$ is the volume of K-dimensional hypersphere with center at x and containting K points,

$d_{K(x)}$ – radius of the spehere (a.k.a. dist. to the Kth nearest neighbour),

n – the total # of points.

2. Connectivity graph $G(Z_n, U_n)$

Here Z_n is the set of all vertices and U_n – the set of edges. So X_i is connected to X_j if

$$d(X_i, X_j) \leq d_K(x_i),\ i \neq j.$$

2.1. Connectivity subgraph $G(Z_i, U_i)$

Here, $Z_i = \{X_1, X_2, ..., X_i\}$ and U_i is the set of edges connecting points in Z_i.

3. Height significance of a cluster

Some cluster C_l is height significant w.r.t. some height $h > 0$ if

$$\max\left\{|p(X_i) - p(X_j)| \mid \forall X_i,\ X_j \in C_l\right\} \geq h.$$

4. Algorithm

1. Prepare the data.

- Calculate $d_K(X)\ \forall X \in \text{data}$
- Sort the data based n $d_K(X)$ in ascending order

2. Initialization.

- Set counter $i = 1$
- Denote cluster label for i-th point as $\omega(X_i)$

3. Process points in sorted order.

For each point in subgraph $G(Z_i, U_i)$ do the following depending on the scenario:

A. Isolated vertex.

- If X is not connected to any other vertex create new cluster.

B. Connected to only one cluster C_l.

- If cluster is complete $\omega(X_i) = 0$ or in layman's terms we label X_i as noise.
- If cluster is not complete we label this point with class label.

C. Connected to multiple clusters.

- If all clusters as complete we lable the point as noise.
- Determine the number of significant cluters (in terms of height significance) $Z(h)$. If $Z(h) > 1$ then label all significant clusters as complete and delete insignificant clusters (by assigning them to noise) and assign the point to noise too: $\omega(X_i) = 0$. If there is only signle significant cluster combine all clusters into one $C_{l_1}, \omega(x) = l_1$

4. Update the counter $i = i + 1$, if $i \leq n$ goto step 3.

5. Advantages

- Algorithm itself determines the # of clusters
- Tends to label more points as noise the other algorithms do*
- Can find clusters of arbitrary form
- Not sensitive to noise

6. Disadvantages

- Sensitive to choice of hyperparameters
- Optimal hyperparameter values are hard to determine
- Slow (but we dgaf)

7. Hyperparameter selection

Obviously weuse grid search, but we need to compare clusterization results to each other. Let us discuss some metrics we can use.

1. Silhouette score $\mathcal{O}(n^2)$

Denote the avg. distance from point i to all other points in a cluster as $a(i)$.

Denote minimum avg. distance between i and other cluster(s?) as $b(i)$.

$$S(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}},$$
$$S = \text{mean}(S(i)).$$

2. Halinski-Harabaz index $\mathcal{O}(n)$

$$\text{SSB} = \sum |C_k| \cdot || \; C_k - C \; ||^2,$$
$$\text{SSW} = \sum || \; X - C_k \; ||^2,$$
$$\text{CH} = \frac{\frac{\text{SSB}}{K-1}}{\frac{\text{SSW}}{N-K}},$$

where:

SSB – intercluster dispersion,

SSW – intracluster dispersion,

K – the number of clusters,

N – the number of points,

$|C_k|$ – the number of elements in the cluster C_k,

C – center of all data.

3. Davies-Bondeu index $\mathcal{O}(n)$

$$s_i = \frac{1}{|C_i|} \sum || \; X - C_i \; ||$$

$$R_{ij} = \frac{s_i + s_j}{|| \; C_i - C_j \; ||}$$

$$D_i = \max_{j, \; i \neq j} R_{ij}$$

$$\mathrm{DB} = \frac{1}{K} \sum D_i, \quad \mathrm{DB} \in [0, \infty)$$

The lower DB is the better.

4. Vaname ratio criterion.

$$\mathrm{TSS} = \sum || \; X_i - C \; ||^2$$

$$\mathrm{BSS} = \sum | \; C_k \; | \; || \; C_k - C \; ||^2$$

$$\mathrm{VR} = \frac{\mathrm{TSS}}{\mathrm{BSS}} \in [0, 1]$$

The closer to 0, worse the clustering is.