ESANN 2018 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2018, i6doc.com publ., ISBN 978-287587047-6.
Available from http://www.i6doc.com/en/.

# Set point thresholds from topological data analysis and an outlier detector

Alessio Carrega

aizoOn - Technology Consulting
Torre San Vincenzo, Via San Vincenzo 2 16-th floor, 16121 Genova, Italy

**Abstract**. We provide an algorithm for unsupervised or semi-supervised learning to determine, once the input settings are given, a very easily described zone of optimal execution settings for a production. A region is very easily described if anyone can determine whether a point is inside it and select a point on it with a certain range of choice. This can be applied both in production optimization and in predictive maintenance. Part of the method is based on a topological data analysis tool: Mapper. We also provide a method to detect outliers on new data.

## 1 Introduction

Many unsupervised, or semi-supervised, learning algorithms divide the data set in sub-sets, then each new point of the space of data is associated with the sub-set that realizes the minimum distance from the point. This provides a decomposition of the space of data in regions. A first request can be to easily understand whether a point is inside one such region or not. Computing all distances from data set points can be computationally expensive. A further request is to easily select a point on one such region with a certain range of choice. To the best of the author knowledge, no algorithms in literature answer this request. Moreover it can required to consider only instances with some fixed coordinates.

This can be very useful in *production optimization* and *predictive maintenance*. In production optimization we are interested in getting products with an high quality. A target in predictive maintenance is to set machines so to have a lifetime bigger than a certain value. In these fields it is extremely useful to have one such very simple description of a region of *execution settings* (*e.g.* humidity, temperature) in order to get products as desired once the *input settings* (*e.g.* raw materials) are given.

The ideal description is a list of threshold for continuous features, and a list of values for non continuous features. In this paper we present an algorithm following the above line. As most learning algorithms, we require that data features are all numeric. In this case one such easy region is a multidimensional rectangle (parallel to the axis), namely a real interval for each execution setting. The number of input settings may be null.

Part of the algorithm is based on a *TDA* (*Topological Data Analysis*) tool, in particular it is based on the 1-dimensional *Mapper* [1]. See [2] as a general refer-

ESANN 2018 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2018, i6doc.com publ., ISBN 978-287587047-6.
Available from http://www.i6doc.com/en/.

ence for TDA. These methods are still little used but are very useful especially for unsupervised, or semi-supervised, learning.

As further motivation of this work, we can notice that most of machine learning algorithms applied in the mentioned fields are based on supervised learning, while we work in unsupervised, or semi-supervised, environment. For a survey on machine learning applications in production optimization and predictive maintenance we can refer on [3, 4].

Let $\Sigma \subset \mathbb{R}^d$ be the $(d-k)$-plane given fixing the $k \geq 0$ input settings. The general steps of the algorithm are: (1) Select an optimal sub-set $B_0$ of the data set (Section 3). (2) Get a *convex polyhedron* $P$ (Definition 2.1) represented by $B_0$ (Section 4). (3) Apply on $B_0$ an outlier detector considering outliers points outside $P_u$ or inside $P_l$ with $P_u$ and $P_l$ convex polyhedra (Section 2). (4) Get a multidimensional square inside the convex polyhedron $\Sigma \cap P \cap P_u$ as big as possible and representing data (Subsection 5.1, Fig. 1-(left)). (5) Get a multidimensional rectangle $R_u \subset \Sigma \cap P \cap P_u$ enlarging as much as possible the square (Subsection 5.2, Fig. 1-(center)). (6) Repeat steps (4) and (5) replacing $P \cap P_u$ with $P_l$ and getting a rectangle $R_l \subset \Sigma \cap P_L$. (7) Decompose $R_u \setminus R_l$ in smaller rectangles (Section 6).

A first approach for an algorithm to get a multidimensional rectangle enlarging a multidimensional square as much as possible would get an exponential complexity on the dimension. We provide a smarter algorithm with a linear complexity.

## 2 Outlier detector

In this section we describe a method to detect outliers on new data. No knowledge of TDA is needed. However the method would be much more precise if applied on every sub-set of the data set obtained after a *clustering* or a division in *bins* as in *Mapper*. Unfortunately there is not enough space to compare this method with better known ones and to provide helpful figures. The main reason why we adopt this method is that it returns convex polyhedra and this applies well to our purpose.

**Definition 2.1.** A *convex polyhedron* $P \subset \mathbb{R}^d$ the intersection of a finite number of half-spaces, namely it is a set of the form $\{x \in \mathbb{R}^d \mid \langle w^{(1)}, x\rangle \leq w_0^{(1)}, \ldots, \langle w^{(h)}, x\rangle \leq w_0^{(h)}\}$ for some $h \geq 1$, $w^{(1)}, \ldots, w^{(h)} \in \mathbb{R}^d$, $w^{(i)} \neq 0$, and $w_0^{(1)}, \ldots, w_0^{(h)} \in \mathbb{R}$, where $\langle v, w\rangle$ is the standard inner product in $\mathbb{R}^d$ of $v$ and $w$. The hyperplanes $\{x \in \mathbb{R}^d \mid \langle w^{(i)}, x\rangle = w_0^{(i)}\}$ that intersect $P$ are called *facets* of $P$. The *interior* of a convex polyhedron is the sub-set obtained substituting all the weak inequalities with strong inequalities. We consider also the empty set $\varnothing$ a convex polyhedron.

We start from a data set $\{x^{(1)}, \ldots, x^{(s)}\} \subset \mathbb{R}^d$ whose elements are considered not to be outlier. Two convex polyhedra, $P_u$ and $P_l$, are constructed, we call them respectively *upper polyhedron* and *lower polyhedron*. The basic idea is to consider as outliers the points too far away from a central point (construction

ESANN 2018 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2018, i6doc.com publ., ISBN 978-287587047-6.
Available from http://www.i6doc.com/en/.

of the upper polyhedron) and refine the result considering a hole around the central point (construction of the lower polyhedron). The upper polyhedron $P_u$ contains the data set while no points of the data set are in $P_l$. The instances outside $P_u$ and inside $P_l$ are considered outliers.

Let $\bar{m}$ be the mean of the data set: $\bar{m} := (\sum_i x^{(i)})/s$. Let $\varepsilon > 0$ be a fixed number, the more $\varepsilon$ is close to 0, the more points are considered outliers.

For $x, y \in \mathbb{R}^d$, $d(x,y) = \sqrt{\langle x-y, x-y\rangle}$ is the distance between $x$ and $y$. For $\lambda \geq -d(x^{(i)}, \bar{m})$ the half-space $H_\lambda(x^{(i)}) \subset \mathbb{R}^d$ is the one containing $\bar{m}$, with boundary hyperplane orthogonal to the line passing through $\bar{m}$ and $x^{(i)}$, and whose distance from $\bar{m}$ is $d(x^{(i)}, \bar{m}) + \lambda$.

Choose a positive integer $k_u$ as a parameter to smooth the rectangle. The higher is $k_u$ the sharper is the region to avoid outliers and the higher is the computational cost. The region $P_u$ is the convex polyhedron obtained by the following algorithm: start with $P_u = \mathbb{R}^d$; remove $\bar{m}$ from the data set; apply a cycle with at most $k_u$ steps; in each step select the point $x^{(i)}$ with the maximum distance from $\bar{m}$, if all the starting data set is contained in $P = P_u \cap H_\varepsilon(x^{(i)})$, substitute $P_u$ with $P$, in both cases remove the point $x^{(i)}$ from the list.

If there are points $x^{(i)}$ in the data set whose distance $d(x^{(i)}, \bar{m})$ from $\bar{m}$ is lower equal than $\varepsilon$ the convex polyhedron $P_l$ is empty. Otherwise $P_l$ is the intersection of all the half-spaces $H_{-\varepsilon}(x^{(i)})$.

## 3 Optimal sub-sets of the data set

Here we apply a method to divide the data set in sub-sets and make some assumptions that will be discussed in Section 4.

*Mapper* [1] is a TDA method which divides the data set in sub-sets called *bins*. These sub-sets can intersect themselves. Every such sub-set is a vertex of a graph and the intersections correspond to the connections between vertices.

Let $\{x^{(1)}, \ldots, x^{(n)}\} \subset \mathbb{R}^d$ be the data set. Apply 1-dimensional Mapper on the data set using the filter function $f : \mathbb{R}^d \to \mathbb{R}$.

**Assumption 3.1.** The filter function $f$ is linear: $f(x) = \langle w^f, x\rangle$ for some $w^f \in \mathbb{R}^d$.

By "optimal" points we mean points that are preferable than others, for instance they could be execution settings of a production giving good products. Let $B_0$ be a set whose elements are considered optimal and is the union $B_0 = \cup_j B_{0,j}$ of connected bins $B_{0,j}$ related to consecutive intervals of the used covering of $\mathbb{R}$. Let $(a,b) \subset \mathbb{R}$ be the open interval corresponding to $B_0$ in the covering of $\mathbb{R}$, and let $B_1, \ldots, B_m$, $m \geq 0$, be the bins such that for all $1 \leq i \leq m$, $f(B_i) \subset (a,b)$ and $B_0 \cap B_i = \varnothing$. Therefore all the elements $x^{(i)} \in B_0 \cup B_1 \cup \ldots \cup B_m$ satisfy $a < \langle w^f, x^{(i)}\rangle < b$.

**Assumption 3.2.** For every $i > 0$ the set $B_0$ is almost linearly separated from $B_i$, namely for each $i > 0$ there is a vector $w^{(i)} \in \mathbb{R}^d$ and a scalar $w_0^{(i)} \in \mathbb{R}$ such that $\langle w^{(i)}, x^{(j)}\rangle < w_0^{(i)}$ and $\langle w^{(i)}, x^{(h)}\rangle > w_0^{(i)}$ almost for every $x^{(j)} \in B_0$

ESANN 2018 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2018, i6doc.com publ., ISBN 978-287587047-6. Available from http://www.i6doc.com/en/.

and $x^{(h)} \in B_i$. This can be studied applying a linear SVM (Support Vector Machine).

## 4 First optimal region

In this section we describe the first optimal region of points in the space of data and discuss the assumptions.

For Section 3, all the points in the following convex polyhedron $P$ that are not outliers can be considered optimal: $P := \{x \in \mathbb{R}^d \mid \langle -w^f, x\rangle \leq -a,\ \langle w^f, x\rangle \leq b,\ \langle w^{(1)}, x\rangle \leq w_0^{(1)},\ \ldots,\ \langle w^{(m)}, x\rangle \leq w_0^{(m)}\}$.

Our assumptions are made to get the set $P$ and to ensure that it is a convex polyhedron and that $\{x^{(1)}, \ldots, x^{(n)}\} \cap P$ consists almost entirely of optimal points. In some situations this could be reached also by the application of a standard clustering algorithm. Mapper is preferable to standard clustering methods for several reasons: connection between bins, visualization, detection of zones, ... (see for instance [1, 5]). One of the most performing choice of filter function for Mapper is a principal component, this choice would make Assumption 3.1 satisfied. Scrolling through this list of filters, and applying Mapper in smaller data sets, we get both assumptions satisfied.

Apply on $B_0$ the method to detect outliers on new data described in Section 2 producing the convex polyhedra $P_u$ and $P_l$. We denote with $\Sigma := \{x \in \mathbb{R}^d \mid x_{d-k+i} = c_i \text{ for } i \in \{1, \ldots, k\}\}$ the $(d-k)$-plane defined by fixing the $k \geq 0$ input settings. Therefore we are interested in $\Sigma \cap P \cap P_u \cap (\mathbb{R}^d \setminus P_l)$. This set is composed just of optimal points, has no outliers, its points respect the input settings, and it is described just by a number of linear inequalities, hence it is easy to determine wheter a point is inside it, but it is still not easy to select a point inside it.

## 5 List of intervals of execution settings

This section explains how to get the multidimensional *rectangles* $R_u \subset \Sigma \cap P \cap P_u$.

### 5.1 The square

Here we aim to find a non empty $(d-k)$-dimensional *square* $C_u \subset \Sigma$ parallel to the axis such that all its elements are optimal and not outlier (see Fig. 1-(left)).

**Definition 5.1.** Let $L = ((a_1, b_1), \ldots, (a_d, b_d))$ be an ordered list of pairs such that for each $j$, $a_j, b_j \in \mathbb{R} \cup \{-\infty, \infty\}$ and either $-\infty \leq a_j < b_j \leq \infty$ or $a_j = b_j \in \mathbb{R}$. We call *rectangle* of $L$ the set

$$R = \{x \in \mathbb{R}^d \mid \text{ for each } j,\ a_j \leq x_j \leq b_j\}.$$

An ordered family $(v_1, \ldots, v_d) \in (\mathbb{R} \cup \{-\infty, \infty\})^d$ such that for every $j$, $v_j \in \{a_j, b_j\}$ is called *vertex* of $R$. The *dimension* of $R$ is the number of components $j$ such that $a_j < b_j$. A *square* is a rectangle such that $b_j - a_j = b_h - a_h$ for every

ESANN 2018 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2018, i6doc.com publ., ISBN 978-287587047-6. Available from http://www.i6doc.com/en/.

$j$ and $h$ such that $a_j < b_j$ and $a_h < b_h$. The quantity $b_j - a_j > 0$ is the *side length* of the square, while the point $m \in \mathbb{R}^d$, $m_j = (b_j + a_j)/2$, is its *center*.

The center $m_u$ of the square $C_u$ is a point in the interior of $\Sigma \cap P \cap P_u$ that represents well the elements of $B_0$ close to $\Sigma$. If it is not possible to get one such point, the algorithm stops, and a rectangle is not produced.

Take the maximum radius $r > 0$ for what $B(m_u, r) \cap \Sigma$ is contained in $P \cap P_u$, where $B(m_u, r)$ is the set of points with distance from $m_u$ at most $r$. This can be easily obtained applying a minimum on the distances $d(m_u, F_i \cap \Sigma)$ between $m_u$ and the hyperplanes $F_i$'s given by the inequalities that define $P \cap P_u$. If $F_i \cap \Sigma \neq \varnothing$ the distance $d(m_u, F_i \cap \Sigma)$ can be easily computed as point/hyperplane distance in $\mathbb{R}^{d-k}$ after the application on $m_u$ and on $F_i \cap \Sigma$ of the map $\mathbb{R}^d \to \mathbb{R}^{d-k}$ $(x_1, \ldots, x_d) \mapsto (x_1, \ldots, x_{d-k})$.

Define the square $C_u$ as the one centered in $m_u$ and with side length $\frac{2r_u}{\sqrt{d-k}}$. This is the biggest square $C$ centered in $m_u$ such that $C \subset \Sigma \cap P \cap P_u$.

### 5.2 Enlarging the square

In this subsection we enlarge the square $C_u$ as much as possible getting the $(d-k)$-dimensional *rectangle* $R_u \subset \Sigma \cap P \cap P_u$ (see Fig. 1-(center)). A first approach would get the result by a simple algorithm based on enlargements vertex by vertex, hence getting an algorithm with exponential complexity on the dimension $d-k$. We provide a smarter algorithm producing the same result and having linear complexity on the dimension $d-k$.

The idea is as follows: Suppose $k = 0$ ($\Sigma = \mathbb{R}^d$). Start taking the square $C_u$ as the rectangle $R_u$. Select a direction $\pm e^{(j)}$ parallel to the axis ($e^{(j)}$ is the $j$-th element of the standard basis of $\mathbb{R}^d$). Select a central point $\bar{m} \in F$ on the facet $F$ of $R_u$ corresponding to this direction $\pm e^{(j)}$. Let $F'$ be the hyperplane containing $F$. Push the facet $F$ and the hyperplane $F'$ along the direction $\pm e^{(j)}$ until the distance between $\bar{m}$ and $F' \cap \partial(P \cap P_u)$ is equal to the distance $d(\bar{m}, v)$, where $v$ is any vertex of $R_u \cap F'$. The central point $\bar{m}$ is constructed by cases according to the limitations of coordinates of the rectangle studying where the facet touches the boundary of the polyhedron for the first time. The distance between the central point $\bar{m}$ and a vertex $v \in F'$ of the rectangle does not depend on the choice of one such vertex and is easy to compute.

If $k > 0$, intersect $\Sigma$ with $P \cap P_u$, and with the hyperplanes $F_i$'s, apply the projection $\mathbb{R}^d \to \mathbb{R}^{d-k}$, $x \mapsto (x_1, \ldots, x_{d-k})$, then apply the method above.

## 6 Sharper list of intervals

Suppose that the lower polyhedron $P_l$, is not empty. Consider the point $m_u$ and its reflections along facets of $P_l$. If one of these points is in the interior of $P_l$ we take it as central point $m_l$ for $P_l$, then we repeat the procedure of Section 5 taking $P_l$ instead of $P \cap P_u$ and $m_l$ instead of $m_u$. Hence we get a rectangle $R_l \subset \Sigma \cap P_l$. Otherwise $R_l$ is empty. The set $R_u \setminus R_l$ can be described as union of rectangles.

ESANN 2018 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2018, i6doc.com publ., ISBN 978-287587047-6.
Available from http://www.i6doc.com/en/.

**Remark 6.1.** If we change the order of the variables $x_j$, $1 \leq j \leq d-k$, by a permutation $\sigma$, we get the same square $C_u$ but a different rectangle $R_{u,\sigma}$ (see Fig. 1-(right)). The algorithm to get a rectangle gives larger intervals to the first coordinates. We can take advantage of this observation to get bigger intervals for selected features or to get rectangles with bigger volumes.

## 7 Conclusions

We provided an algorithm for unsupervised, or semi-supervised, learning getting a very simple description of a region in the space of data consisting of optimal points that are not outliers and have some fixed coordinates. This can be very useful in *production optimization* and *predictive maintenance*.

This work has three main original contributes and combines them together: 1. A discussion of all the assumptions and a method to satisfy them based on TDA (Section 4, steps (1) and (2)). 2. An outlier detector for new data (TDA is not needed but is useful) (Section 2, step (3)). 3. An efficient algorithm to get a multidimensional rectangle enlarging a multidimensional square as much as possible (TDA is not needed) (Subsection 5.2, step (5)).

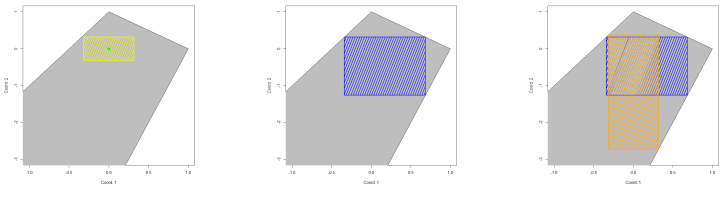

Fig. 1: $\mathbb{R}^d = \Sigma = \mathbb{R}^2$, (left) The biggest square inside $P \cap P_u$ centered in a representative point of the sub-set of the data set. (center) The rectangle inside $P \cap P_u$ obtained enlarging the square. (right) The rectangle obtained after the non trivial permutation.

## References

[1] G. Singh, F. Mémoli, and G. Carlsson. Topological methods for the analysis of high dimensional data sets and 3d object recognition. In *SPBG*, pages 91–100, 2007.

[2] G. Carlsson. Topology and data. *Bulletin of the American Mathematical Society*, 46(2):255–308, 2009.

[3] R. Dekker and P. A. Scarf. On the impact of optimisation models in maintenance decision making: the state of the art. *Reliability Engineering & System Safety*, 60(2):111–119, 1998.

[4] R. Ahmad and S. Kamaruddin. An overview of time-based and condition-based maintenance in industrial application. *Computers & Industrial Engineering*, 63(1):135–149, 2012.

[5] M. Nicolau, A. J. Levine, and G. Carlsson. Topology based data analysis identifies a subgroup of breast cancers with a unique mutational profile and excellent survival. *Proceedings of the National Academy of Sciences*, 108(17):7265–7270, 2011.