PIF: Anomaly detection via preference embedding

Filippo Leveni
Politecnico di Milano (DEIB)
filippo.leveni@polimi.it

Luca Magri
Politecnico di Milano (DEIB)
luca.magri@polimi.it

Giacomo Boracchi
Politecnico di Milano (DEIB)
giacomo.boracchi@polimi.it

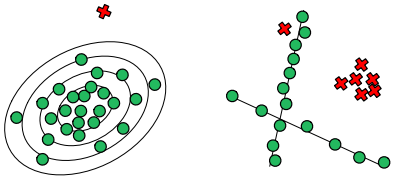
Cesare Alippi
Politecnico di Milano (DEIB)
cesare.alippi@polimi.it
Università della Svizzera italiana
cesare.alippi@usi.ch

***Abstract*—We address the problem of detecting anomalies with respect to structured patterns. To this end, we conceive a novel anomaly detection method called PIF, that combines the advantages of adaptive isolation methods with the flexibility of preference embedding. Specifically, we propose to embed the data in a high dimensional space where an efficient tree-based method, PI-FOREST, is employed to compute an anomaly score. Experiments on synthetic and real datasets demonstrate that PIF favorably compares with state-of-the-art anomaly detection techniques, and confirm that PI-FOREST is better at measuring arbitrary distances and isolate points in the preference space.**

I. INTRODUCTION

Anomaly detection deals with the problem of identifying data that do not conform to an expected behavior [1]. This task, sometimes referred to as outlier detection, finds numerous applications in fraud [2], [3] and intrusion [4] detection, health [5] and quality monitoring [6], to name a few examples. In the statistical and data-mining literature, *anomalies* are typically detected as samples falling in low-density regions of a probability density model describing the data [1]. On the contrary, *normal* data are samples that lie in denser regions. In this paper, we consider anomaly detection in a pattern-recognition setup, where anomalies are samples that deviate from certain structured patterns. Although *statistical anomalies* can also be seen as a particular case of these *pattern-recognition anomalies*, where the model describing normal data is a pdf, statistical-based and pattern-based approaches are traditionally treated separately in the literature as they employ different algorithms and methods.

Fig. 1 illustrates differences between statistical and pattern-recognition anomalies. In Fig. 1a a statistical anomaly can be easily identified as a sample falling in a low density area, in Fig. 1b anomalies are instead points that are not collinear, while normal data belong to two patterns described by line equations. Note that density by itself in this latter case is not meaningful to identify anomalies, unless further processing is considered. Although overly simplified, Fig. 1b illustrates a primary task that has to be successfully addressed in several computer vision and pattern recognition applications, where more general parametric models are used instead of straight lines to identify structures or regularities in data. Finding anomalies with respect to a parametric model is at the core of many low level vision tasks, like robust curve detection [7]. Moreover, this is a problem routinely addressed in Structure-from-Motion [8], where data consist in point-wise matching features between multiple images, and



(a) Density (b) Pattern

Fig. 1: Left: an anomaly (marked as $\times$) is recognized as a point in a low density area. Right: anomalies are defined with respect to their deviation from patterns described by line equations.

anomalies are wrong matches that cannot be described by consistent geometric transformations, such as homographies or fundamental matrices. Other examples include 3D registration [9], where anomalous matches are defined with respect to rototranslations, and object/template matching [10]–[12]. Anomaly detection in these settings is very challenging, since anomalies cannot be directly removed without having identified each and every structure first but, at the same time, anomalies hinder the identification of the existing structures. For this reason, anomalies are often detected as a byproduct of a multi-structure estimation process, which is performed through robust model fitting algorithms [13]–[16]. Within this framework, the structures underlying normal data are first identified, and then, all those points that do not conform with them are labeled as anomalous. We believe that model fitting is by far a more difficult problem than anomaly detection and that algorithms directly detecting anomalies would be preferable in those situations where anomaly detection is the primary goal (e.g., because anomalies convey relevant information on their own). Even in those situations where the focus is on the recovery of structures/models, it might be convenient to eliminate structure-less samples first, to ease the subsequent structure estimation task, as demonstrated by several domain-specific pre-filtering techniques employed before robust estimation (e.g., [17], [18]).

Here we present Preference Isolation Forest (PIF), a novel algorithm to directly detect anomalies among structures whose nature is described by a given parametric function of unknown parameters. To this purpose, we embed data into an high dimensional space, called *preference space* [14], [15], and then we perform anomaly detection by relying on PI-FOREST, an efficient tree-based method that reflects a suitable distance

metric for identifying anomalies. To the best of our knowledge, this is the first time anomaly detection is applied in the preference space. Extensive experiments show that (i) exploiting structure information allows to identify anomalies effectively and (ii) PIF outperforms state-of-the-art anomaly detectors like Local Outlier Factor [19] (LOF), Isolation Forest [20], [21] (IFOR) and Extended Isolation Forest [22] (EIFOR). Most remarkably, we show that straightforward solutions plugging-in anomaly detection algorithms in the preference space are often not successful. We speculate that this is due to the fact that these methods do not leverage an appropriate distance function for the preference space. On the contrary, PI-FOREST achieves superior performance thanks to a nested Voronoi tessellations constructed over the Tanimoto distance [23] that is specifically designed to capture preference agreements.

In summary our main contributions are:

- PIF, the first algorithm that identifies anomalies with respect to structured patterns by means of preference embedding.
- PI-FOREST, a novel tree-based anomaly detection tool that is very successful in the preference space, but that can be extended to any metric space.

II. PROBLEM FORMULATION

We assume we are given a noisy finite dataset $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\} \subset \mathbb{R}^d$ containing both normal[1] and anomalous data. Normal data can be described as belonging to a union of structures $S = S_1 \cup \ldots \cup S_k$, defined by a parametric model family $\mathcal{F}$, of which we assume to know the analytical expression. In particular, each structure S_i corresponds to the vanishing set of an instance of $\mathcal{F}$, described by a specific parameter vector $\boldsymbol{\theta}_i$. In a noisy free setup, all normal points must satisfy equation $\mathcal{F}(\mathbf{x}, \boldsymbol{\theta}_i) = 0$ for some parameter vector $\boldsymbol{\theta}_i$ but, because of noise, the previous equation is not necessarily satisfied, and we should rather expect $\mathcal{F}(\mathbf{x}, \boldsymbol{\theta}_i) \approx 0$ for every $\mathbf{x} \in S_i$. Anomalies instead do not refer to any structure, and form a subset $A \subset X$. For example, the inliers depicted in Fig. 1b can be described as a collection of two lines: in this case, each normal point $\mathbf{x} = (x, y) \in S \subset \mathbb{R}^2$ approximately satisfies a first degree equation $\mathcal{F}(\mathbf{x}, \boldsymbol{\theta}) = \theta_1 x + \theta_2 y + \theta_3 = 0$, where $\boldsymbol{\theta} = (\theta_1, \theta_2, \theta_3)$ represents the line coefficients. On the contrary, anomalies are generated by a different unknown process and are not coherent with lines.

We address the problem of automatically detecting all the anomalies $A \subset X$. Specifically, our aim is to derive an *anomaly score* $\alpha \colon X \to \mathbb{R}$ that ranks higher the anomalies, namely $\alpha(\mathbf{a}) \gg \alpha(\mathbf{s})$ for all $\mathbf{a} \in A$ and $\mathbf{s} \in S$, so that it is possible to detect them by setting an appropriate threshold. This task is particularly challenging since the overall number of structures k as well as their parameters $\{\boldsymbol{\theta}_i\}_{i=1,\ldots,k}$ are unknown. The nature and amount of noise determining how much normal data depart from the analytical equations of their corresponding structure is unknown, but we assume it can be directly estimated (e.g., [24]).

[1]Herein and through the paper by normal we do not mean Gaussian, but rather that conform to the normal state.

III. RELATED WORK

Several approaches have been proposed to identify anomalies, and for a comprehensive description refer to [1], [25]. A possible taxonomy envisages three main categories: distance-based, density-based and model-based. In distance-based anomaly detection [26] an instance is considered anomalous when its neighborhood does not contain a sufficient number of samples. Simplest methods of this category are based on the K-Nearest Neighbors [27] approach: the anomaly score of a data sample, is simply the distance to its k-th nearest neighbor. Better results can be obtained when data-dependent distance measures are employed [28]. Density-based anomaly detection methods (e.g., [19], [29], [30]) follow a similar idea, but density is used instead of distance. The key concept is that anomalous and normal instances differ in their local density. An important algorithm representative of this category is LOF [19]. The basic idea is that density around a normal instance is similar to the density around its k-neighbors, in contrast the density around an anomaly is significantly different from the local density of its k-neighbors.

In model-based anomaly detection it is assumed that normal data are generated from a model, thus, the more an instance deviates from the model, the higher its probability to be anomalous. Most existing model-based methods learn a model from the data, then identify anomalies as those data points that do not fit the model well. Notable examples of this approach are classification-based methods [31], reconstruction-based methods [32], [33], [34], and clustering-based methods [35]. Isolation Forest (IFOR) [20], [21] instead directly isolates anomalies by assuming that they are "few and different" [20] compared to normal instances. IFOR builds a forest of randomized trees [36] from data and [20], [21] show that, on average, anomalous points end up in leaves at shallower levels of tree height than normal data (herein and hereafter we refer to *height* as a synonym for *depth*). An effective extension of IFOR is Extended Isolation Forest (EIFOR) [22].

The above methods are usually applied in the *ambient space*, namely the space where data are given, and are effective to identify statistical anomalies. However, when dealing with pattern-recognition anomalies, this is not the best solution as pattern or group-level information is not exploited. In our approach we leverage on the principle that multiple normal points belong to a structure, shifting the problem in a preference space where anomalies can be easily separated from structured data. In the next section we recall the main concepts of preference analysis necessary for this construction.

IV. PREFERENCE ISOLATION FOREST

The proposed Preference Isolation Forest (PIF) computes anomaly scores α in two main steps: (i) embedding the data in the *preference space* and (ii) adopting a tree-based isolation approach to detect anomalies in the preference space.

A. Preference embedding

For the first time in the context of anomaly detection, we propose to use the preference embedding, a technique

Algorithm 1: PIF anomaly detection

Input: X - input data, t - number of trees, ψ - sub-sampling size, b - branching factor
Output: Anomaly scores $\{\alpha_\psi(\mathcal{E}(\mathbf{x}_i))\}_{i=1,\dots,n}$

```
/* Preference embedding */
```
1 Sample m models $\{\boldsymbol{\theta}_i\}_{i=1,\dots,m}$ from X
2 $P \leftarrow preferenceEmbedding(X, \{\boldsymbol{\theta}_i\}_{i=1,\dots,m})$
```
/* Training Preference Isolation Forest */
```
3 $F \leftarrow$ PI-FOREST(P, t, ψ, b)
```
/* Scoring input data */
```
4 **for** $i = 1$ *to* $|P|$ **do**
5 $\mathbf{h} \leftarrow [0, \dots, 0] \in \mathbb{R}^t$
6 **for** $j = 1$ *to* t **do**
7 $T \leftarrow j$-th PI-TREE in F
8 $[\mathbf{h}]_j \leftarrow$ PATHLENGTH$(\mathbf{p}_i, T, 0)$
9 $\alpha_\psi(\mathbf{p}_i) \leftarrow 2^{-\frac{E(\mathbf{h}(\mathbf{p_i}))}{c(\psi)}}$
10 return $\{\alpha_\psi(\mathbf{p}_i)\}_{i=1,\dots,n}$

previously used in the multi-model fitting literature [14]–[16] to which the interested reader is referred for further details. PIF starts by mapping each point $\mathbf{x} \in X$ to an m-dimensional vector having components in the unitary interval $[0, 1]$, via a mapping $\mathcal{E}\colon X \rightarrow [0,1]^m$. The space $[0,1]^m$ is called preference space. More precisely, the embedding depends on: a family $\mathcal{F}$ of models parametric in $\boldsymbol{\theta}$, a set of m model instances $\{\boldsymbol{\theta}_i\}_{i=1,\dots,m}$ and an estimate of the standard deviation σ of the noise affecting the data. A sample $\mathbf{x}_i \in X$ is then embedded to a vector $\mathbf{p}_i = \mathcal{E}(\mathbf{x}_i)$ whose j-th component is defined as

$$[\mathbf{p}_i]_j = \begin{cases} \phi(\delta_{ij}) & \text{if } \delta_{ij} = \mathcal{F}(\mathbf{x}_i, \boldsymbol{\theta}_j) \leq 3\sigma \\ 0 & \text{otherwise} \end{cases}, \tag{1}$$

where $\delta_{ij} = \mathcal{F}(\mathbf{x}_i, \boldsymbol{\theta}_j)$ measures the deviation of sample $\mathbf{x}_i$ with respect to the model $\boldsymbol{\theta}_j$, and ϕ is a monotonically decreasing function in $[0, 1]$ such that $\phi(0) = 1$. As in [16], we use a Gaussian function of the form

$$\phi(\delta) = \exp(-\delta^2/\sigma). \tag{2}$$

The j-th component of the preference vector $\mathbf{p}_i$, namely $[\mathbf{p}_i]_j$, is the preference granted by a point $\mathbf{x}_i$ to model $\boldsymbol{\theta}_j$: the closer $\mathbf{x}_i$ to $\boldsymbol{\theta}_j$, the higher the preference. The embedding function $\mathcal{E}$ maps dataset X to the set of preference vectors

$$P = \{\mathbf{p}_i = \mathcal{E}(\mathbf{x}_i) \,|\, \mathbf{x}_i \in X\}, \tag{3}$$

which represents the image of X through the embedding $\mathcal{E}$. The pool $\{\boldsymbol{\theta}_i\}_{i=1,\dots,m}$ of m models are sampled from the data using a RanSaC-like strategy (line 1, Algorithm 1): minimal sample set – composed by the minimal number of points necessary to constraint a parametric model – are extracted uniformly from the data, and are used to determine the model parameters. For example, two points are drawn to determine the equation of a line $\boldsymbol{\theta}_j$.

The preference space is equipped with the Tanimoto distance [23] to measure similarity between preferences: given two samples $\mathbf{p}_i = \mathcal{E}(\mathbf{x}_i)$ and $\mathbf{p}_j = \mathcal{E}(\mathbf{x}_j)$ their Tanimoto distance is

$$\tau(\mathbf{p}_i, \mathbf{p}_j) = 1 - \frac{\langle \mathbf{p}_i, \mathbf{p}_j \rangle}{\|\mathbf{p}_i\|^2 + \|\mathbf{p}_j\|^2 - \langle \mathbf{p}_i, \mathbf{p}_j \rangle}. \tag{4}$$

Our choice for the Tanimoto distance is motivated by the fact that points conforming to the same structures share similar preferences, yielding low distances. In contrast, anomalies would result in null preferences to the majority of structures, thus resulting in very sparse vectors that tend to have distance close to 1 with the majority of other samples.

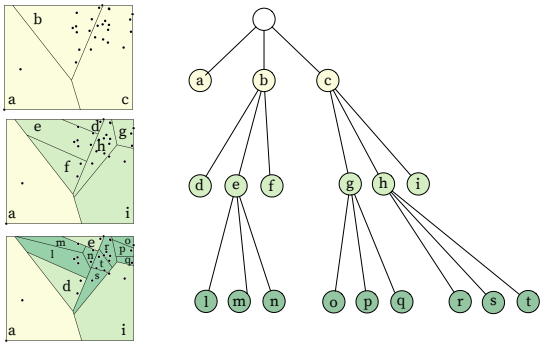


Fig. 2: A PI-TREE with branching factor $b = 3$ and height limit $l = 3$ constructed from a set of points in $\mathbb{R}^2$. Every region is recursively split in b sub-regions. The most isolated samples fall in leaves at lowest heights, such as 'a' and 'd' cells.

B. *PI*-FOREST

By moving from a Euclidean space to the preference space, state-of-the-art data driven methods such as IFOR and its variants perform poorly (as we will show in our experiments), since the splitting criterion used by these techniques implicitly measures point distances by the ℓ_2 norm. For this reason, we build our solution upon PI-TREE, a novel tree-based isolation technique where the splitting criterion is based on Voronoi tessellations. A Voronoi tessellation is simply a partition of a metric space into b regions defined by b samples, called seeds $\mathcal{S} = \{\mathbf{s}_i\}_{i=1,\dots,b}$. The i-th region produced by the tessellation contains all the points $\mathbf{p}$ of the space having $\mathbf{s}_i$ as the closest seed in $\mathcal{S}$. The proposed PI-TREE is a nested version of Voronoi tessellations, where each region is further split in b sub-regions; this procedure can be repeated recursively as illustrated in Fig. 2. Voronoi tessellation as partitioning criterion preserves, better than other splitting schemes, the notion of distance in the preference space. In fact, in our case, regions are defined by seeds $\mathcal{S}$ and Tanimoto distance (Eq. 4). In general, Voronoi tessellation naturally applies to any metric space, preference space included. The construction of a PI-TREE is described in Algorithm 2 and starts from a single region corresponding to the whole space $[0,1]^m$ that is then recursively split in b sub-regions by randomly selecting b seeds $\{\mathbf{s}_i\}_{i=1,\dots,b} \subset P$ (line 4). Thus the points are partitioned into b subsets $\mathcal{P} = \{P_i\}_{i=1,\dots,b}$, each $P_i \subset P$ collecting those points

Algorithm 2: PI-TREE

Input: P - preference representations, e - current tree height, l - height limit, b - branching factor
Output: A PI-TREE

1 **if** $e \geq l$ *or* $|P| < b$ **then**
2 return $exNode\{Size \leftarrow |P|\}$
3 **else**
4 randomly select a set of b seeds $\{\mathbf{s}_i\}_{i=1,\dots,b} \subset P$
5 $\mathcal{P} \leftarrow voronoiPartition(P, \{\mathbf{s}_i\}_{i=1,\dots,b})$
6 $chNodes \leftarrow \emptyset$
7 **for** $i = 1$ *to* b **do**
8 $chNodes \leftarrow chNodes \cup$ PI-TREE$(P_i, e+1,$
9 $l, b)$
10 return $inNode\{ChildNodes \leftarrow chNodes,$
11 $SplitPoints \leftarrow \{\mathbf{s}_i\}_{i=1,\dots,b}\}$

Algorithm 3: PI-FOREST

Input: P - preference representations, t - number of trees, ψ - sub-sampling size, b - branching factor
Output: A set of t PI-TREES

1 $F \leftarrow \emptyset$
2 set height limit $l = \log_b \psi$
3 **for** $i = 1$ *to* t **do**
4 $P' \leftarrow subSample(P, \psi)$
5 $F \leftarrow F \cup$ PI-TREE$(P', 0, l, b)$
6 return F

in P that have $\mathbf{s}_i$ as the closest seed according to Tanimoto distance (line 5). The number of seeds b is the branching factor of the tree associated to the splitting process. The partitioning process stops when it is not possible to further split a region (i.e., the number of points in the region is less than b), or the tree reaches a maximum height (lines 1-2), set by default at $l = \log_b \psi$ (an approximation for the average tree height [37]), where ψ is the number of points used to build the tree. The recursive process is outlined in lines 6-9, where a sub-tree is build for each subset $P_i \in \mathcal{P}$ (line 8).

The height of the leaf in which a point falls into is directly related to its separability: a lower height corresponds to points

Algorithm 4: PATHLENGTH

Input: $\mathbf{p}$ - a sample, T - a PI-TREE, e - current path length
Output: Path length of $\mathbf{p}$

1 **if** T *is an external node* **then**
2 return $e + c(T.size)$
3 $childNode \leftarrow voronoiLocate(\mathbf{p}, T.splitPoints,$
4 $T.childNodes)$
5 return $PathLength(\mathbf{p}, childNode, e+1)$

that can be separated with few splits from the rest of the data. To get an intuition of the concept of separability consider Fig. 2 where, for visualization purposes, the Euclidean distance is being considered. Here anomalies correspond to samples that fall in leaves with lower height. Conversely, samples in high-density regions fall in leaves with higher height since in denser regions the number of possible recursive splits is higher. In order to gain robustness and to decrease the variance due to randomness in PI-TREE realizations, this idea is extended to PI-FOREST, a forest of PI-TREES, and the *average* height of a point in this forest is used to compute its overall anomaly score α. Algorithm 3 details the construction of a PI-FOREST containing t PI-TREES. Each PI-TREE is instantiated on a subset $P' \subset P$ of preference representations of X (line 4). The subsampling factor is controlled by the parameter ψ.

C. Anomaly score

Anomaly scores are computed as in IFOR and other tree-based isolation methods [20], [21], [38]. With reference to Algorithm 1, after the samples are embedded in the preference space (lines 1-2) and the PI-FOREST is built (line 3), each instance $\mathbf{p} \in P$ is passed through all the PI-TREES of the PI-FOREST, and the heights reached in every tree are computed and collected in a vector $\mathbf{h}(\mathbf{p}) = [h_1(\mathbf{p}), \dots, h_t(\mathbf{p})]$ (lines 5-9). Then, (line 10) the anomaly score α is

$$\alpha_\psi(\mathbf{p}) = 2^{-\frac{E(\mathbf{h}(\mathbf{p}))}{c(\psi)}}, \tag{5}$$

where $E(\mathbf{h}(\mathbf{p}))$ is the mean value over the elements of $\mathbf{h}(\mathbf{p})$ and $c(\psi)$ is an adjustment factor.

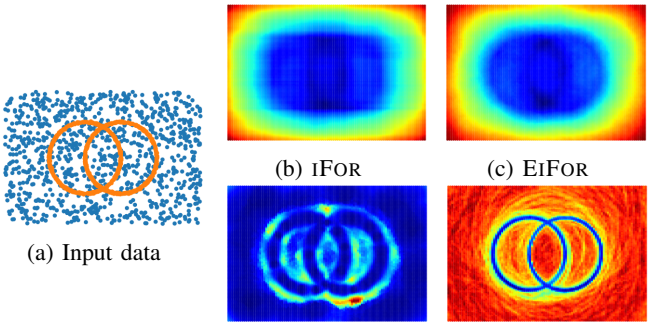
The heights are computed through PATHLENGTH function described in Algorithm 4. In particular, at line 3, the instance $\mathbf{p}$ is located in the corresponding region of the Voronoi partition having $T.splitPoints$ as seeds. Then the child node associated to the region is identified and used for the subsequent recursive process at line 5. At line 2, the height is computed as $h_i(\mathbf{p}) = e + c(T.size)$, that is the height e of the leaf where $\mathbf{p}$ falls in the i-th tree, plus an adjustment coefficient $c(n)$ that depends on the cardinality n of this leaf. The adjustment factor is necessary to take into account subtrees that stopped the construction process, having reached the height limit l. We assume that $b = 2$, in this case the adjustment factor becomes as [20], [21]:

$$c(n) = \begin{cases} 0 & \text{if } n = 1 \\ 1 & \text{if } n = 2 \\ 2H(n-1) - 2(n-1)/n & \text{if } n > 2 \end{cases}, \tag{6}$$

where $H(i)$ is the harmonic number and it can be estimated by $\ln(i) + \gamma$ (being γ the Euler's constant).

The complexity of PIF is $O(\psi \cdot t \cdot b \cdot \log_b \psi)$ as it regards PI-FOREST construction, and $O(n \cdot t \cdot b \cdot \log_b \psi)$ for the scoring phase, where n is the number of instances to be scored. With respect to IFOR we have an additional overhead due to the embedding $\mathcal{E}(\cdot)$, which however can be easily parallelized.

PIF has two main advantages over other isolation-based anomaly detection tools, like IFOR. First, the preference trick



(a) Input data (b) IFOR (c) EIFOR (d) LOF (e) PIF

Fig. 3: Color-coded anomaly scores produced by different algorithms: high scores in red, low scores in blue.

allows to integrate useful information about normal data and consequently to better characterize structure-less anomalies. Second, the Voronoi tessellation preserves the intrinsic distance of the preference space (i.e., Tanimoto distance) during the splitting process. These two features can be appreciated in the example reported in Fig. 3 where the problem of identifying anomalies with respect to two circles is addressed. By looking at the color-coded anomaly scores produced by IFOR, EIFOR, LOF, and PIF, it is possible to recognize that only PIF correctly identifies anomalies inside the circles. On the contrary IFOR, that works directly with X, does not reflect the structures of normal data, as it performs splits parallel to the axes, and consequently struggles to adapt to geometries that are not aligned to the main axes. With EIFOR the situation improves slightly, but anomalies still cannot be precisely identified. Also LOF, which heavily depends on the neighbourhood parameter k to estimate local density, has difficulties. LOF identifies as anomalous those regions where data density changes. The region outside the circles, although sparser, does not give rise to a change in density and it is thus erroneously identified as normal. Note that only PIF exploits the preference space, the other methods are confined to operate in terms of density and isolability. However, in the experiments reported in the next section, we will see that the advantages of PIF are not just motivated by the preference embedding, since other anomaly detection methods plugged in the preference space would yield lower detection performance than PIF. This further demonstrates the importance of PI-FOREST having regions defined by Tanimoto distance rather than Euclidean distance.

V. EXPERIMENTAL VALIDATION

We validate the advantages of PIF by assessing the benefits of coupling the preference trick with PI-FOREST, both in terms of detection accuracy and stability. To this purpose, we adopt both synthetic and real-world datasets.

A. Datasets

We consider the synthetic 2D datasets depicted in Fig. 4. The structures S characterizing normal data are lines in stair$[*]$ and star$[*]$, and circles in circle$[*]$, where $[*]$ indicates the number of structures. Anomalies have been sampled from a uniform distribution within the bounding box containing normal data. For all the datasets every structure has 50 normal points, with the exception of stair3 and circle3 that have unbalanced structures, as detailed in Table Ia. Experiments on real data are performed on the AdelaideRMF dataset [39], that consists in stereo images and annotated matching points. Erroneous matches are also annotated and correspond to anomalies. The first 19 sequences refer to static scenes containing several planes, each giving rise to a set of matches described by an homography. For the remaining 19 sequences the scene is not static: several objects move independently and give rise to a set of point correspondences described by a a fundamental matrix. In both these scenarios we want to recognize anomalous matches.

B. Methodology

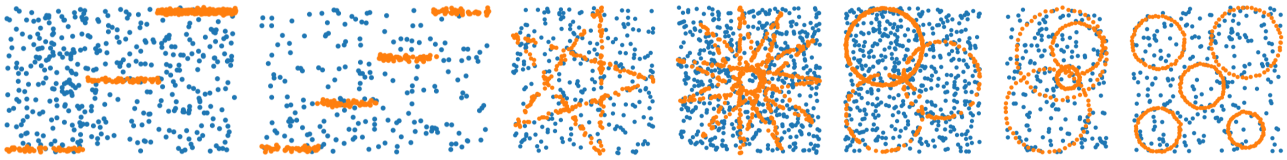
We compare PIF with IFOR, EIFOR and LOF. To assess the benefits of the preference trick, experiments on synthetic data are performed in the (i) ambient (Euclidean) space, (ii) preference space and (iii) binarized preference space, where the Tanimoto distance specializes exactly to the Jaccard distance [40] (binary vectors are used in Eq. (4)). We also explore the performance of PIF without the preference embedding (PIF ℓ_2), i.e. setting $P = X$, and with the binarized preference embedding (PIF jac). Preferences are computed with respect to a pool of $m = 10|X|$ model instances, being circles used in circle$[*]$ datasets and lines elsewhere.

To evaluate the stability of our approach, we perform an additional experiment on stair3 and circle5. These datasets have been modified so that they contain different percentages of anomalies. In particular, for both datasets, $|X|$ is kept fixed and equal to 1000, while $|S|$ and $|A|$ vary as: $\frac{|A|}{|X|} = \{0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$.

Experiments on real datasets are performed in the preference space. The pool of models is determined by sampling $m = 6|X|$ model instances, homographies for the first 19 datasets and fundamental matrices for the remaining 19.

The parameters of IFOR, EIFOR and PIF are kept fixed to $t = 100, \psi = 256$ and $b = 2$ in all the experiments. As regard LOF instead, given its sensitivity to the neighborhood size, various values of the parameter k are employed, ranging from $k = 10$ to $k = 500$.

Anomaly detection performance, collected in Tab. II, IIIa and IIIb, are evaluated in terms of AUC averaged over 10 runs. The highest value for each dataset is underlined, whereas a boldface indicates that the best AUC is statistically better than its competitors in the same embedding, according to a paired t-test with $\alpha = 0.05$. The LOF results refer to the k that maximizes the average AUC along the datasets of each model family, for all the embeddings (k parameter values are reported in Table Ib). Fig. 5 displays the AUC achieved in correspondence with different percentages of anomalies employed, averaged over 10 runs. Only the best 3 values of k for LOF have been reported.



(a) stair3 (b) stair4 (c) star5 (d) star11 (e) circle3 (f) circle4 (g) circle5

Fig. 4: Synthetic datasets. Orange dots represent normal data, while blue dots represent anomalies.

C. Discussion

Table II shows that all methods improve their performance when performed in the preference space. Therefore, it is always convenient to exploit the preference trick rather than working directly in the ambient space (i.e., in X). In addition, continuous preferences must be preferred over binary ones, as their greater expressiveness produces a space where anomalies can be identified more effectively. Note that LOF achieves a better AUC on star5, star11 and circle5 where normal data are evenly distributed among structures, and it is possible to get an optimal parameter of k. On the contrary, LOF performs poorly when structures have different cardinalities (stair3 and circle3). PIF instead works with fixed parameters and attains good results also for unbalanced structures; on average PIF is the best statistical significantly method. The critical dependence of LOF to k is also highlighted in Fig. 5 where stability is evaluated with respect to the rate of anomalies contamination. As the percentage of anomalies varies, the optimal value of k changes and it becomes difficult to guess a correct size for the neighborhoods of normal points. This is particularly evident in the unbalanced dataset stair3. As a consequence, the performance of LOF has a great variability according to the chosen k. Moreover, although IFOR and EIFOR achieve more stable performances, they are not as stable as PIF. In addition, there is a very apparent performance gap between IFOR, EIFOR and PIF, since only the latter is able to deal with the Tanimoto metric.

As real data are concerned, even if the high AUC values shown in Tab. IIIa and IIIb suggest that anomaly detection on these tasks is easier than in the synthetic case, the difference in performance between IFOR, EIFOR (Euclidean distance) and LOF, PIF (Tanimoto distance) remains evident.

Both LOF and PIF seem to be valid methods to perform anomaly detection in the preference space, although on average the performance of PIF is better: the difference between mean AUCs is always statistically in favor of PIF, except on Tab. IIIb where the methods are statistically equivalent. In addition, LOF must be tuned according to the expected size of the neighborhoods of normal points while PIF has been tested with fixed parameters.

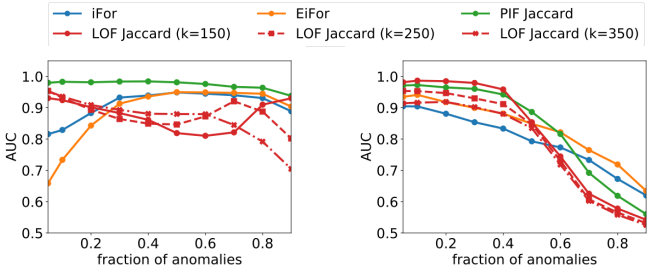
	$\lvert X\rvert$	$\lvert A\rvert$	$\lvert S\rvert$	$\lvert S_1\rvert,\ldots,\lvert S_k\rvert$
stair3	800	400	400	$\lvert S_1\rvert = 272, \lvert S_i\rvert = 64 \quad \forall i \in \{2,3\}$
stair4	400	200	200	$\lvert S_i\rvert = 50 \quad \forall i \in \{1,\ldots,4\}$
star5	500	250	250	$\lvert S_i\rvert = 50 \quad \forall i \in \{1,\ldots,5\}$
star11	1100	550	550	$\lvert S_i\rvert = 50 \quad \forall i \in \{1,\ldots,11\}$
circle3	1000	500	500	$\lvert S_1\rvert = 376, \lvert S_i\rvert = 62 \quad \forall i \in \{2,3\}$
circle4	400	200	200	$\lvert S_i\rvert = 50 \quad \forall i \in \{1,\ldots,4\}$
circle5	500	250	250	$\lvert S_i\rvert = 50 \quad \forall i \in \{1,\ldots,5\}$

(a) Synthetic datasets settings

	Euclidean	Preference binary	Preference
circle	75	25	25
line	25	150	75
homography	-	100	100
fundamental	-	75	80

(b) Optimal k parameters of LOF

TABLE I: Experiment settings



(a) stair3, unbalanced structures (b) circle5, balanced structures

Fig. 5: AUCs achieved at various percentages of anomalies.

VI. Conclusion and Future Directions

We proposed PIF, a new algorithm for identifying anomalous samples that do not conform to structured patterns. This was done by wisely coupling – for the fist time – two ingredients: (i) embedding input data in the preference space and, (ii) applying isolation-based anomaly detection tool (i.e., PI-FOREST) in the preference space. Our empirical evaluation demonstrated that preference embedding increases the separability between normal (structured) and anomalous (not structured) data, leading to superior performance than simply performing anomaly detection in the original space. Most remarkably, the proposed anomaly-detection method PI-FOREST outperforms all the alternatives where anomaly-detection methods are straightforwardly plugged in the preference space. This result highlights the effectiveness of nested Voronoi tessellations in the isolation process. We believe there are several research directions to investigate further, including use of non-parametric models for preference embedding (e.g., supervised trained models), and using PIF in real-world defect-detection applications. Furthermore, given the ability of PIF to deal with arbitrary distance metric, it could

	Euclidean				Preference binary				Preference			
	LOF ℓ_2	IFOR	EIFOR	PIF ℓ_2	LOF jac	IFOR	EIFOR	PIF jac	LOF tani	IFOR	EIFOR	PIF
stair3	0.737	0.925	0.920	0.918	0.904	0.885	0.864	0.958	0.815	0.923	0.925	**0.971**
stair4	0.814	0.889	0.874	0.871	0.849	0.855	0.860	0.941	0.881	0.912	0.908	**0.952**
star5	0.771	0.722	0.738	0.788	0.875	0.745	0.769	0.872	**0.929**	0.761	0.822	0.910
star11	0.671	0.728	0.727	0.738	0.830	0.739	0.741	0.771	**0.900**	0.738	0.774	0.796
circle3	0.761	0.698	0.732	0.779	0.719	0.842	0.854	0.900	0.731	0.854	0.891	**0.930**
circle4	0.640	0.641	0.665	0.679	0.827	0.686	0.699	0.860	0.906	0.667	0.720	0.897
circle5	0.543	0.569	0.570	0.633	0.699	0.597	0.617	0.672	**0.823**	0.573	0.593	0.780
Mean	0.705	0.739	0.747	0.772	0.815	0.764	0.772	0.853	0.855	0.775	0.805	**0.891**

TABLE II: Synthetic datasets AUCs

	LOF tani	IFOR	EIFOR	PIF
barrsmith	**0.969**	0.708	0.715	0.944
bonhall	0.918	0.969	0.967	0.949
bonython	**0.978**	0.679	0.691	0.954
elderhalla	0.999	0.925	0.909	**0.999**
elderhallb	0.986	0.924	0.943	**0.999**
hartley	0.963	0.749	0.793	**0.989**
johnsona	0.993	0.993	0.993	**0.998**
johnsonb	0.776	0.999	0.998	0.999
ladysymon	0.847	0.944	0.943	**0.997**
library	**1.000**	0.764	0.771	0.998
napiera	0.975	0.869	0.879	**0.983**
napierb	0.888	0.931	0.936	**0.953**
neem	0.985	0.896	0.906	**0.996**
nese	**0.996**	0.888	0.892	0.980
oldclassicswing	0.936	0.923	0.943	**0.987**
physics	0.670	0.858	0.787	**1.000**
sene	**0.997**	0.698	0.731	0.988
unihouse	0.785	0.998	0.998	**0.999**
unionhouse	**0.987**	0.639	0.664	0.968
Mean	0.929	0.861	0.866	**0.983**

(a) Homographies

	LOF tani	IFOR	EIFOR	PIF
biscuit	0.976	0.994	0.996	**1.000**
biscuitbook	1.000	0.987	0.988	1.000
biscuitbookbox	**1.000**	0.990	0.989	0.996
boardgame	**0.962**	0.400	0.304	0.949
book	0.996	1.000	1.000	1.000
breadcartoychips	**0.989**	0.978	0.971	0.976
breadcube	**1.000**	0.998	0.998	0.999
breadcubechips	**0.999**	0.985	0.985	0.998
breadtoy	0.984	0.999	0.998	0.999
breadtoycar	**0.998**	0.933	0.883	0.991
carchipscube	**0.993**	0.981	0.966	0.987
cube	0.999	0.970	0.982	0.999
cubebreadtoychips	0.990	0.962	0.958	0.989
cubechips	**1.000**	0.995	0.994	1.000
cubetoy	**1.000**	0.997	0.995	1.000
dinobooks	0.887	0.873	0.857	**0.899**
game	**1.000**	0.901	0.895	0.999
gamebiscuit	1.000	0.985	0.988	**1.000**
toycubecar	**0.973**	0.290	0.192	0.964
Mean	0.987	0.906	0.891	0.987

(b) Fundamental matrices

TABLE III: Real datasets AUCs

be interesting to apply PI-FOREST in spaces other than the preference space.

REFERENCES

[1] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM computing surveys*, vol. 41, no. 3, pp. 1–58, 2009.

[2] M. Ahmed, A. N. Mahmood, and M. R. Islam, "A survey of anomaly detection techniques in financial domain," *Future Generation Computer Systems*, vol. 55, pp. 278–288, 2016.

[3] A. Dal Pozzolo, G. Boracchi, O. Caelen, C. Alippi, and G. Bontempi, "Credit card fraud detection: A realistic modeling and a novel learning strategy," in *Transactions on Neural Networks and Learning*, IEEE, vol. 29, 2018, pp. 3784–3797.

[4] R. Bronte, H. Shahriar, and H. Haddad, "Information theoretic anomaly detection framework for web application," in *Computer Software and Applications Conference*, IEEE, vol. 2, 2016, pp. 394–399.

[5] H. Banaee, M. U. Ahmed, and A. Loutfi, "Data mining for wearable sensors in health monitoring systems: A review of recent trends and challenges," *Sensors*, vol. 13, no. 12, pp. 17 472–17 500, 2013.

[6] L. Stojanovic, M. Dinic, N. Stojanovic, and A. Stojadinovic, "Big-data-driven anomaly detection in industry (4.0): An approach and a case study," in *International Conference on Big Data*, IEEE, 2016, pp. 1647–1652.

[7] L. Xu, E. Oja, and P. Kultanen, "A new curve detection method: Randomized Hough transform (RHT)," *Pattern Recognition Letters*, vol. 11, no. 5, pp. 331–338, 1990.

[8] M. Farenzena, A. Fusiello, and R. Gherardi, "Structure-and-motion pipeline on a hierarchical cluster tree," in *IEEE International Workshop on 3-D Digital Imaging and Modeling*, Oct. 2009, pp. 1489–1496.

[9] P. J. Besl and N. D. McKay, "A method for registration of 3-d shapes," *Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239–256, 1992.

[10] R. Brunelli, *Template matching techniques in computer vision: theory and practice*. John Wiley & Sons, 2009.

[11] O. O. Sushkov and C. Sammut, "Local image feature matching for object recognition," in *International Conference on Control Automation Robotics & Vision*, IEEE, 2010, pp. 1598–1604.

[12] M. Cho, J. Lee, and K. M. Lee, "Feature correspondence and deformable object matching via agglomerative correspondence clustering," in *International Conference on Computer Vision*, IEEE, 2009, pp. 1280–1287.

[13] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[14] R. Toldo and A. Fusiello, "Robust multiple structures estimation with J-Linkage," in *European Conference on Computer Vision*, 2008, pp. 537–547.

[15] L. Magri and A. Fusiello, "T-Linkage: A continuous relaxation of J-Linkage for multi-model fitting," in *Conference on Computer Vision and Pattern Recognition*, Jun. 2014, pp. 3954–3961.

[16] L. Magri and A. Fusiello, "Multiple models fitting as a set coverage problem," in *Conference on Computer Vision and Pattern Recognition*, Jun. 2016, pp. 3318–3326.

[17] J. Cech, J. Matas, and M. Perdoch, "Efficient sequential correspondence selection by cosegmentation," *Transactions on pattern analysis and machine intelligence*, vol. 32, no. 9, pp. 1568–1581, 2010.

[18] T. Sattler, B. Leibe, and L. Kobbelt, "Scramsac: Improving ransacś efficiency with a spatial consistency filter," in *International Conference on Computer Vision*, IEEE, 2009, pp. 2090–2097.

[19] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "Lof: Identifying density-based local outliers," in *International Conference on Management of data*, 2000, pp. 93–104.

[20] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *International Conference on Data Mining*, IEEE, 2008, pp. 413–422.

[21] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation-based anomaly detection," *Transactions on Knowledge Discovery from Data*, vol. 6, no. 1, pp. 1–39, 2012.

[22] S. Hariri, M. C. Kind, and R. J. Brunner, "Extended isolation forest," *arXiv preprint arXiv:1811.02141*, 2018.

[23] T. Tanimoto, "An elementary mathematical theory of classification and prediction," IBM, Internal Technical Report, 1957.

[24] H. Wang and D. Suter, "Robust adaptive-scale parametric model estimation for computer vision," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1459–1474, 2004.

[25] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection for discrete sequences: A survey," *Transactions on knowledge and data engineering*, vol. 24, no. 5, pp. 823–839, 2010.

[26] E. M. Knorr, R. T. Ng, and V. Tucakov, "Distance-based outliers: Algorithms and applications," *The VLDB Journal*, vol. 8, no. 3-4, pp. 237–253, 2000.

[27] S. Ramaswamy, R. Rastogi, and K. Shim, "Efficient algorithms for mining outliers from large data sets," in *International conference on Management of data*, 2000, pp. 427–438.

[28] K. M. Ting, Y. Zhu, M. Carman, Y. Zhu, and Z.-H. Zhou, "Overcoming key weaknesses of distance-based neighbourhood methods using a data dependent dissimilarity measure," in *International conference on knowledge discovery and data mining*, 2016, pp. 1205–1214.

[29] H.-P. Kriegel, P. Kröger, E. Schubert, and A. Zimek, "Loop: Local outlier probabilities," in *Conference on Information and knowledge management*, 2009, pp. 1649–1652.

[30] S. Papadimitriou, H. Kitagawa, P. B. Gibbons, and C. Faloutsos, "Loci: Fast outlier detection using the local correlation integral," in *International conference on data engineering*, IEEE, 2003, pp. 315–326.

[31] N. Abe, B. Zadrozny, and J. Langford, "Outlier detection by active learning," in *International conference on Knowledge discovery and data mining*, 2006, pp. 504–509.

[32] J. Chen, S. Sathe, C. Aggarwal, and D. Turaga, "Outlier detection with autoencoder ensembles," in *International conference on data mining*, SIAM, 2017, pp. 90–98.

[33] D. Carrera, B. Rossi, P. Fragneto, and G. Boracchi, "Online anomaly detection for long-term ecg monitoring using wearable devices," *Pattern Recognition*, vol. 88, pp. 482–492, 2019.

[34] D. Carrera, F. Manganini, G. Boracchi, and E. Lanzarone, "Defect detection in sem images of nanofibrous materials," in *Transactions on Industrial Informatics*, IEEE, vol. 13, 2017, pp. 551–561.

[35] Z. He, X. Xu, and S. Deng, "Discovering cluster-based local outliers," *Pattern Recognition Letters*, vol. 24, no. 9-10, pp. 1641–1650, 2003.

[36] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," *Machine learning*, vol. 63, no. 1, pp. 3–42, 2006.

[37] D. E. Knuth, *The art of computer programming: Volume 3: Sorting and Searching*. Addison-Wesley Professional, 1998.

[38] A. Mensi and M. Bicego, "A novel anomaly score for isolation forests," in *International Conference on Image Analysis and Processing*, Springer, 2019, pp. 152–163.

[39] H. S. Wong, T.-J. Chin, J. Yu, and D. Suter, "Dynamic and hierarchical multi-structure geometric model fitting," in *International Conference on Computer Vision*, 2011.

[40] P. Jaccard, "Étude comparative de la distribution florale dans une portion des Alpes et des Jura," *Bulletin del la Société Vaudoise des Sciences Naturelles*, vol. 37, pp. 547–579, 1901.