Anomaly Detection

UW ECE 657A - Core Topic

Mark Crowley

Lecture Outline

1. What is Anomaly Detection?
   - Definitions
   - Solution Approaches
   - Datasets
2. Anomaly Detection By Density
   - Local Outlier Factor (LOF)
   - Oversampling PCA
3. Anomaly Detection By Isolation : Isolation Forests
4. Anomaly Detection By Isolation : iMondrian Forests
   - iMondrian Forests Concept
   - Experiments
   - Comparison Amongst All Algorithms
5. Anomaly Detection By Classification : One-Class SVM
6. Comparing Algorithms
7. References and Further Reading

Lecture Outline

1. What is Anomaly Detection?
   - Definitions
   - Solution Approaches
   - Datasets
2. Anomaly Detection By Density
3. Anomaly Detection By Isolation : Isolation Forests
4. Anomaly Detection By Isolation : iMondrian Forests
5. Anomaly Detection By Classification : One-Class SVM

What is an Anomaly?

When someone says they want to detect or deal with anomalies they could be saying the data that is ...

- "normal" (be careful people don't mean Gaussian!)
- inlier (as oppossed to an *outlier*)
- usual, **regular**, standard (as a pattern in space, in time?, in some other dimension?)
- expected (as opposed to "rare")
  - what assumptions underlie this expectation?
- conforming to what has been seen before
- ...really *anything* they decide that normal means.

What is an Anomaly?

When someone says they want to detect or deal with anomalies they could be saying the data that is ...

- "normal" (be careful people don't mean Gaussian!)
- inlier (as oppossed to an *outlier*)
- usual, **regular**, standard (as a pattern in space, in time?, in some other dimension?)
- expected (as opposed to "rare") nominal vs anomaly
  - what assumptions underlie this expectation?
- conforming to what has been seen before
- ...really *anything* they decide that normal means.

in the end, it is subjective, it needs to be defined for the specific problem at hand

Applications: Where can Anomalies Occur?

- intrusion detection
- fraud detection
- fault detection
- system health monitoring
- event detection in sensor networks
- detecting ecosystem disturbances
- more...

What is Anomaly Detection?

Ideally[1], *Anomaly Detection* is a subset of **Supervised Learning** where

[1] what do I mean *ideally*? well, AD *can* be done in a supervised manner...but if you always have the labels, it's not really anomaly detection.

What is Anomaly Detection?

Ideally[1], *Anomaly Detection* is a subset of **Supervised Learning** where

Definition: Anomaly Detection

The task of labelling the datapoints in a dataset which are "significantly different" from the "regular" pattern of data in structure, in behaviour over time, or some other property.

[1] what do I mean *ideally*? well, AD *can* be done in a supervised manner...but if you always have the labels, it's not really anomaly detection. what is it then? outlier detection?

What is Anomaly Detection?

Ideally[1], *Anomaly Detection* is a subset of **Supervised Learning** where

Definition: Anomaly Detection

The task of labelling the datapoints in a dataset which are "significantly different" from the "regular" pattern of data in structure, in behaviour over time, or some other property.

Of course this just kicks the can down the road...

- how do we define significantly different?
- how do we define regular?

fair questions, just like clustering

[1] what do I mean *ideally*? well, AD *can* be done in a supervised manner...but if you always have the labels, it's not really anomaly detection. what is it then? outlier detection?

What is Anomaly Detection?

Ideally[1], *Anomaly Detection* is a subset of **Supervised Learning** where

Definition: Anomaly Detection

The task of labelling the datapoints in a dataset which are "significantly different" from the "regular" pattern of data in structure, in behaviour over time, or some other property.

Of course this just kicks the can down the road...

- how do we define significantly different?
- how do we define regular?

fair questions, just like clustering Just like **clustering** this can be seen as an *inherently subjective* task. some examples of subjective clusters from clustering?

[1] what do I mean *ideally*? well, AD *can* be done in a supervised manner...but if you always have the labels, it's not really anomaly detection. what is it then? outlier detection?

General Approach Towards Anomalies

Outlier Detection:

- Anomalies are outliers, datapoints which are far from the regular datapoints and not representative of the underlying distritbuion.
- This could be noise, errors in measurement, or some minor other distritbuion which you are not interested in (a kind of noise).
- Outliers are labelled in the training data.
- The detector is an estimation algorithm which learns a model for the concentrated, regular points.
- This model can be used to label other points as anomalies.

Novelty Detection:

- Training data does not contain outliers
- Goal is to detect whether a new observation is different than previous points, is it new or *novel*?

Different Ways To Detect Anomalies

- Supervised Anomaly Detection:
  - **Outlier Detection** fits here.
  - Learn a Classifier for the normal cases, for datapoints with bad fit for any class, label as anomaly
  - 1 vs All
  - **The Tricky Part:** The data is inherently *unbalanced*, relatively small number of outliers.
  - Common methods: One-Class SVM, Random Forests
- Unsupervised or Semi-Supervised Anomaly Detection:
  - **kmeans**, **DBScan,**, etc
  - **Local Outlier Factor (LOF)** - density based
  - **Oversampling PCS (osPCA)** - online method uses oversampling around points to estimate expected direction and classify *deviations* as anomalous
  - Isolation Forests - tree-based ensemble methods for anomaly detection

Different Ways To Detect Anomalies

- Supervised Anomaly Detection:
  - **Outlier Detection** fits here.
  - Learn a Classifier for the normal cases, for datapoints with bad fit for any class, label as anomaly
  - 1 vs All
  - **The Tricky Part:** The data is inherently *unbalanced*, relatively small number of outliers. otherwise they wouldn't be outliers would they?
  - Common methods: One-Class SVM, Random Forests
- Unsupervised or Semi-Supervised Anomaly Detection:
  - **kmeans**, **DBScan,**, etc
  - **Local Outlier Factor (LOF)** - density based
  - **Oversampling PCS (osPCA)** - online method uses oversampling around points to estimate expected direction and classify *deviations* as anomalous
  - Isolation Forests - tree-based ensemble methods for anomaly detection

Other Definitions

- Supervised anomaly detection techniques require a data set that has been labeled as "normal" and "abnormal" and involves training a classifier (the key difference to many other statistical classification problems is the inherent unbalanced nature of outlier detection).
- Semi-supervised anomaly detection techniques construct a model representing normal behavior from a given normal training data set, and then test the likelihood of a test instance to be generated by the learnt model.
- Unsupervised anomaly detection techniques detect anomalies in an unlabeled test data set under the assumption that the majority of the instances in the data set are normal by looking for instances that seem to fit least to the remainder of the data set.

Online vs. Offline

To give us even more options, Anomaly Detection could can be performed :

- **offline**, where the data are processed as a **batch**
- or **online**, or where the data are processed as a**stream**

Motivation for Better Anomaly Detection

Anomaly detection refers to the task of detecting the outliers in a dataset where the anomalies are different from the regular pattern of data.

- security, embedded systems
- medical technology
- autonomous driving

Much work done already.
Why aren't we done yet?

Simple Statistical Approaches

A simple way to see anomalies is as *irregularities* in your data[2]

- deviations from mean, median, mode, quantile measures of data
- for a time-series, you could use a rolling window to compute statistic then use a **low-pass filter** or a **Kalman filter** to detect deviations as sudden prediction changes.

Drawbacks: This simple technique won't work in some siutations, such as:

- The data contains noise similar to abnormal behavior
- It the definition of abnormal is not static, such as in security as attacks strategies evolve and adapt. A static threshold based on moving average will not work here.
- The pattern could arise from seasonality. This involves more sophisticated methods, then you need to seperate out each of the underlying patterns or cluster/seperate the data by season/period to see the true, simpler pattern.

[2] This slide is based on description in this nice blog post on Anomaly Detection by Pramit Choudhary, `https://blogs.oracle.com/datascience/introduction-to-anomaly-detection`.

Simple Statistical Approaches

A simple way to see anomalies is as *irregularities* in your data[2]

- deviations from mean, median, mode, quantile measures of data
- for a time-series, you could use a rolling window to compute statistic then use a **low-pass filter** or a **Kalman filter** to detect deviations as sudden prediction changes.

Drawbacks: This simple technique won't work in some siutations, such as:

- The data contains noise similar to abnormal behavior because the boundary between normal and abnormal behavior is often not precise
- It the definition of abnormal is not static, such as in security as attacks strategies evolve and adapt. A static threshold based on moving average will not work here.
- The pattern could arise from seasonality. This involves more sophisticated methods, then you need to seperate out each of the underlying patterns or cluster/seperate the data by season/period to see the true, simpler pattern.

[2]This slide is based on description in this nice blog post on Anomaly Detection by Pramit Choudhary, `https://blogs.oracle.com/datascience/introduction-to-anomaly-detection`.

Anomaly Detection Datasets

We will use a standard set of datasets used in AD literature.
CICDS Dataset (CICDS, 2017) This is a larger dataset captures network 80 network features on the duration of a transmitted (or received) message, called a "flow", from an established set of network attacks on a network.

Features Include:

- total packets
- min/max/mean/std packet length/duration
- inter-arrival time between packets
- various network flags set status (eg. FIN,SYN, ACK, CWR)
- min/max/mean/std active/idle period durations
- upload/download ratio
- foward and backward versions of all these

Anomaly Detection Datasets

We will use a standard set of datasets used in AD literature.

CICDS Dataset (CICDS, 2017) This is a larger dataset captures network 80 network features on the duration of a transmitted (or received) message, called a "flow", from an established set of network attacks on a network.

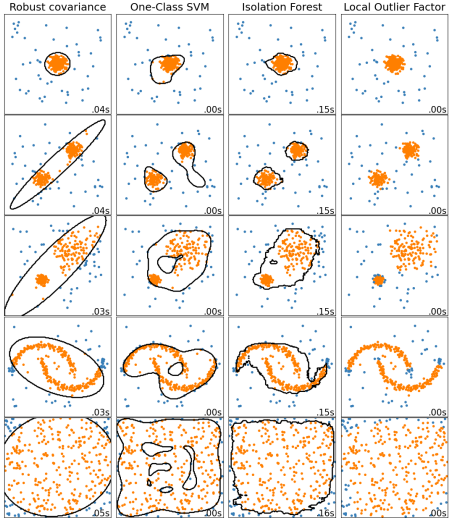
CICFlowMeter : Extraction done by -

https://www.unb.ca/cic/research/applications.html#CICFlowMeter

Features Include:

- total packets
- min/max/mean/std packet length/duration
- inter-arrival time between packets
- various network flags set status (eg. FIN,SYN, ACK, CWR)
- min/max/mean/std active/idle period durations
- upload/download ratio
- foward and backward versions of all these

Comparison of Algorithms



A Review of Tree-Based Approaches for Anomaly Detection

Table 2 ROC AUC score of a selection of methods from literature.

	Code available	HTTP	SMTP	Forest C.	Shuttle	Mammogr.	Satellite	Pima	Breastw	Arrhytmia	Ionosph.
IF	✓	**1.00**	0.88	0.88	**1.00**	0.86	0.71	0.67	**0.99**	0.80	0.85
SCIForest	✓	**1.00**	-	0.74 [98]	**1.00** [98]	0.59 [11]	0.71 [98]	0.65 [98]	0.98 [98]	0.72 [98]	0.91 [93]
EIF	✓	*0.99*	0.85 [37]	0.92	0.99 [37]	0.86	0.78	0.70	***0.99***	0.80 [37]	0.91
RRCF	✓	0.99 [29]	0.89 [29]	*0.91*	0.91 [18]	0.83 [18]	0.68 [18]	0.59 [18]	0.64 [18]	0.74 [18]	0.90 [18]
IMF	✓	***1.00***	0.87	*0.90*	*0.99*	*0.74*	0.74	0.64	*0.97*	*0.80*	0.86
MPF		**1.00**	0.84	0.77	0.51	0.87	0.70	0.66	0.97	0.81	0.88
PIDForest	✓	0.99	**0.92**	0.84	**0.99**	0.84	0.70	0.70 [18]	0.99	-	0.84 [18]
OPHiForest		-	-	-	0.99	-	0.77	0.72	0.96	0.78	0.93
LSHiForest	✓	-	-	0.94	0.97	-	0.77	0.71	0.98	0.78	0.91
HIF	✓	-	0.90	-	**1.00**	**0.88**	0.74	0.70	*0.98*	0.80	*0.86*
HEIF		-	0.90	-	0.99	0.83	0.73	0.72	-	0.80	-
OneClassRF	✓	0.98	**0.92**	0.85	0.95	-	-	0.71	-	0.70	0.90
T-Forest		0.99	-	-	0.99	-	0.68	0.71	-	**0.84**	0.94
EGiTree		-	-	**0.97**	0.94	-	0.73	-	-	-	0.94
GIF	✓	-	-	0.94	-	0.87	**0.86**	**0.84**	-	-	-
dForest		**1.00**	-	-	**1.00**	-	-	0.75	**0.99**	-	**0.97**
ReMass IF		**1.00**	0.88	0.96	**1.00**	0.86	0.71	-	**0.99**	0.80	0.89
HSF	✓	**1.00**	0.90	0.89	**1.00**	0.86	-	0.69	**0.99**	**0.84**	0.80

Selected algorithms are: Isolation Forest (IF) [53], Split-Criteria Isolation Forest (SCIForest) [54], Extended Isolation Forest (EIF) [33], Robust Random Cut Forest (RRCF) [30], Isolation Mondrian Forest (IMF) [60], Mondrian Polya Forest (MPF) [18], Partial Identification Forest (PIDForest) [29], Order Preserving Hashing Based Isolation Forest (OPHIF) [93], Locality Sensitive Hashing Isolation Forest (LSHiForest) [98], Hybrid Isolation Forest (HIF) [64], Hybrid Extended Isolation Forest (HEIF) [37], One-class Random Forest (OneClassRF) [27], Trident Forest (T-Forest) [97], Entropy-based Greedy Isolation Tree (EGiTree) [50], Generalized Isolation Forest (GIF) [11], Distribution Forest (dForest) [95], Re-Mass Isolation Forest (ReMass IF) [6], Half-Spaces Forest (HSF) [84].

Figure: From `https://link.springer.com/chapter/10.1007/978-3-030-83819-5_7`

Lecture Outline

1 What is Anomaly Detection?

2 Anomaly Detection By Density
- Local Outlier Factor (LOF)
- Oversampling PCA

3 Anomaly Detection By Isolation : Isolation Forests

4 Anomaly Detection By Isolation : iMondrian Forests

5 Anomaly Detection By Classification : One-Class SVM

6 Comparing Algorithms

Local Outlier Factor (LOF)

Summary:

This algorithm[BKNS00]is an **unsupervised anomaly** approach that uses *local density variations* for each datapoint to detect anomalous points.

- Input Parameter: **Number of Neigbours** n_{neigh} (good default = 20?)
  - $n_{neigh} >$ mininum size of a useful cluster
  - $n_{neigh} <$ largest group points that could still be considered outliers

Definitions of LOF Components

Definitions:

$k - dis(A)$: k-distance(A) is the distance to from A to the kth nearest neighbour

$N_k(A)$: is the set of all neighbours at or nearer than k-dist(A)

- LOF uses some common concepts from DBScan and Optics.
- Note: distance is not predefined, it is *relative to the density* of the **k nearest neighbours** locally to each point.

$rd(A, B, k)$: reachability distance is either:

- the true distance between A and B : dist(A,B)
- OR the k-dist(B)
- *In essence this treats $N_k(B)$ as the "core" points in DBScan for point B*

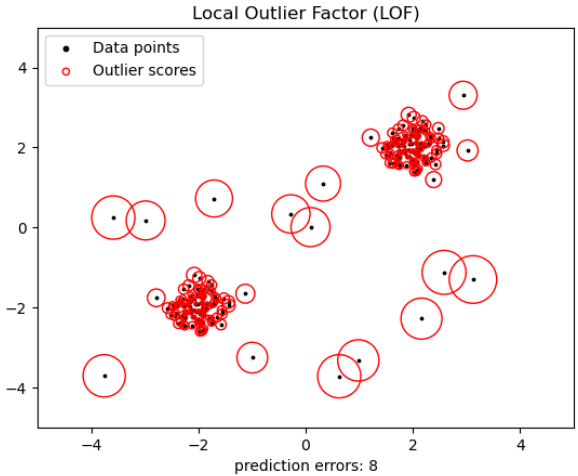
$lrd_k(A)$: local reachability density of A

- $\frac{|N_k(A)|}{(\text{Sum of } rd(A,B) \text{ for all core neighbours of} A)}$
- can be see as the average distance it takes for all of A's "core" neighbours (closer than kth neighbour) to reach A.
- Very general and robust notion of density.

How LOF Works

- For some query point A,
- LOF compares the $lrd_k(A)$ for different points to $lrd_k(B)$ of its neighbours.
- This is then normalized by $|N_k(A)|$ itself.
- The result? A useful scoring function:
  - $LOF(k) = 1$: A has similar density to its neighbors
  - $LOF(k) < 1$: A has higher density than its neighbors
  - $LOF(k) > 1$: A has lower density than neighbors
- We could call this an outlier or anomaly. *(if we believe the assumptions of this model)*

What LOF Looks Like



Online (Incremental) LOF

If the data is arriving continously then we can use **incremental LOF**[PLL07].

- updates the local density and other algorithm variables for the new data points as they arrive
- updates the density for the existing points which are k-nearest neighbors of the new datapoints

Oversampling Principal Component Analysis (osPCA)

- This is an online anomaly detection method which oversamples a point and calculates the principal direction both with and without the oversampled point.
- If the principal direction deviates significantly, the point is considered to be anomalous.
- There are two versions of osPCA:
  - osPCA1 with power method [YLL09]
  - osPCA2 with least squares approximation [LYW13].

Lecture Outline

1 What is Anomaly Detection?

2 Anomaly Detection By Density

3 Anomaly Detection By Isolation : Isolation Forests

4 Anomaly Detection By Isolation : iMondrian Forests

5 Anomaly Detection By Classification : One-Class SVM

6 Comparing Algorithms

7 References and Further Reading

Isolation Forests

[LTZ08][LTZ12]

- Used solely for Anomaly Detection
- Algorithm: Grows an extremely randomized tree, without training labels, until every data point is isolated into a leaf of size 1.
- The depth of the final leaf is used as a proxy for how anomolous the datapoint is
- Anomaly score is computed by using depth and normalizing it by the expected depth of a balanced binary search tree on the same datapoints.

Isolation Forests

[LTZ08][LTZ12]

- Used solely for Anomaly Detection
- Algorithm: Grows an extremely randomized tree, without training labels, until every data point is isolated into a leaf of size 1.
- The depth of the final leaf is used as a proxy for how anomolous the datapoint is
- Anomaly score is computed by using depth and normalizing it by the expected depth of a balanced binary search tree on the same datapoints.

DURING LECTURE: draw picture of isolation tree (one balanced, one unbalanced)

A Bit About Binary Trees

The structure of an isolation tree is similar to a binary search tree. So, we can estimate the *expected path length* in isolation trees as:

$$c(n) := 2\,h(n-1) - \big(2\,(n-1)/n\big),$$

where $h(i)$ is the i^{th} harmonic number, defined as:

$$h(i) := \ln(i) + e,$$

A Bit About Binary Trees

The structure of an isolation tree is similar to a binary search tree. if the data is random enough
So, we can estimate the *expected path length* in isolation trees as:

$$c(n) := 2\,h(n-1) - \big(2\,(n-1)/n\big),$$

where $h(i)$ is the i^{th} harmonic number, defined as:

$$h(i) := \ln(i) + e,$$

where the added constant is the Euler's constant

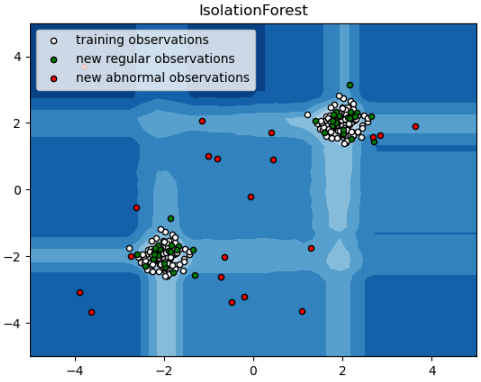


Figure: From scikit-learn

Lecture Outline

1 What is Anomaly Detection?

2 Anomaly Detection By Density

3 Anomaly Detection By Isolation : Isolation Forests

4 Anomaly Detection By Isolation : iMondrian Forests
- iMondrian Forests Concept
- Experiments
- Comparison Amongst All Algorithms

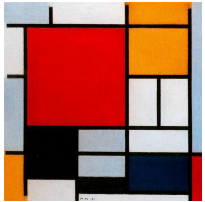
5 Anomaly Detection By Classification : One-Class SVM

Mondrian Processes and Mondrian Trees

- *Mondrian processes* are families of random, hierarchical, binary partitions and probability distributions over tree data structures (Roy, Daniel and Teh, 2008).
- In a *Mondrian Tree* every node r has a *split time* τ_r
- τ_r increases with the *depth* of the node
  but the increase is sampled *stochastically*

$$\text{root}=0 \quad \rightarrow \quad \text{leaves}=\infty$$

- A *Mondrian forest* (Lakshminarayanan, Roy and Teh, 2014) is an ensemble of *Mondrian trees*



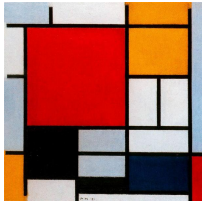
"Composition with Large Red Plane, Yellow, Black, Gray and Blue"

Piet Mondrian, 1921

Mondrian Processes and Mondrian Trees

- *Mondrian processes* are families of random, hierarchical, binary partitions and probability distributions over tree data structures (Roy, Daniel and Teh, 2008).
- In a *Mondrian Tree* every node r has a *split time* τ_r
- τ_r increases with the *depth* of the node
  but the increase is sampled *stochastically* sampling is from an exponential distribution proportional to the size of the current block

  root=0 $\rightarrow$ leaves=∞
- A *Mondrian forest* (Lakshminarayanan, Roy and Teh, 2014) is an ensemble of *Mondrian trees*



"Composition with Large Red Plane, Yellow, Black, Gray and Blue"

Piet Mondrian, 1921

Visualization of Mondrian Forests

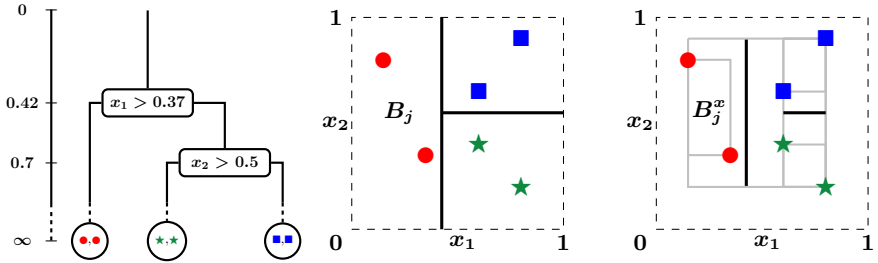


Figure 1: (left) A Mondrian tree over six *data points* in $[0,1]^2$. Every node in the tree represents a split and is embedded in time (vertical axis). **(middle)** An ordinary decision tree partitions the whole space. **(right)** In a Mondrian tree, splits are committed only within the range of the data in each block (denoted by gray rectangles). Let $j = \mathsf{left}(\epsilon)$ be the left child of the root: then $B_j = (0, 0.37] \times (0, 1]$ is the block containing the red circles and $B_j^x \subseteq B_j$ is the smallest rectangle enclosing the two data points. (Adapted from [13], with permission.)

Uses

- Note that the Mondrian forest is designed for *classification* and *regression* so its authors propose an analysis for smooth posterior updates in the blocks.
- However, the Mondrian forest can be used for *unsupervised* purposes where the posterior analysis and the pausing process can be bypassed.
- This is useful **anomaly detection** which is primarily an unsupervised task.

Motivation

- It is an attractive idea to build an **Anomaly Detection** algorithm that can handle *streaming data*.
- As we've seen, **classification** algorithms can be used to *supervised anomaly detection* where anomaly detection is seen as a classification task with two classes of normal and anomalous.
- So, it's natural that the streaming classficiation/regression ensemble method **Mondrian Forests** has been used ([**?**]).

Isolation Mondrian Forests

As easy as iMF = iF + MF [MGS+20]

iMondrian Forests simply combines:

- the online, training ability of Mondrian Forests
- with the simple, but powerful, isolation concept from iForests

But the *result* is an algorithm which...

- performs almost as well in batch mode as iForests
- outperforms other streaming methods which use more complex transformations

How the iMondrian Forest Algorithm Works

The details take a little bit of working out. *see the paper [MGS+20] for all the details*

Splitting a block $\mathcal{B}_r$ (same as MF)

- sample e from an exp distribution with rate $\lambda = \sum_{j=1}^{d} (\boldsymbol{u}_{\mathcal{X}_b}(j) - \boldsymbol{\ell}_{\mathcal{X}_b}(j))$
- split time of node = split time of parent + e
- sample value p from continuous uniform distribution $U(\boldsymbol{\ell}_{\mathcal{X}_b}(q), \boldsymbol{u}_{\mathcal{X}_b}(q))$

Stopping (inspired by iForest)

- The tree is grown until every node contains a single data point $|\mathcal{X}| = 1$

Building an Ensemble of iMondrian Trees

- The iMondrian forest is an ensemble of iMondrian trees

How the iMondrian Forest Algorithm Works

The details take a little bit of working out. *see the paper [MGS+20] for all the details*

Splitting a block $\mathcal{B}_r$ (same as MF)

- sample e from an exp distribution with rate $\lambda = \sum_{j=1}^{d} (\boldsymbol{u}_{\mathcal{X}_b}(j) - \boldsymbol{\ell}_{\mathcal{X}_b}(j))$
- split time of node = split time of parent + e
- sample value p from continuous uniform distribution $U(\boldsymbol{\ell}_{\mathcal{X}_b}(q), \boldsymbol{u}_{\mathcal{X}_b}(q))$

Stopping (inspired by iForest)

- The tree is grown until every node contains a single data point $|\mathcal{X}| = 1$

Building an Ensemble of iMondrian Trees

- The iMondrian forest is an ensemble of iMondrian trees

just as with random forests

Evaluating a Datapoint

- given trained trees, calculate the path length of every tree for a data point $\boldsymbol{x}$
- The path length for the t-th tree, $l_t(\boldsymbol{x})$, is the number of edges traversed by the point from the root to the node containing point $\boldsymbol{x}$.
- The expected path length

$$\mathbb{E}(l(\boldsymbol{x})) := \frac{1}{|\mathcal{F}|} \sum_{t=1}^{|\mathcal{F}|} l_t(\boldsymbol{x})$$

- anomaly score

$$s(\boldsymbol{x}) := 2^{-\mathbb{E}(l(\boldsymbol{x}))/c(n)}$$

where $|\mathcal{F}|$ is the number of trees in the forest.

Anomaly Evaluation

- compare anomaly score for a data point against a threshold to determine if it is anomolous
- setting the threshold is problem specific
- In [LTZ08] a `threshold = 0.5` is used

- we compare this value to one determined dynamically via k-means clustering

Anomaly Evaluation

- compare anomaly score for a data point against a threshold to determine if it is anomolous
- setting the threshold is problem specific
- In [LTZ08] a `threshold = 0.5` is used The theoretical reason for threshold 0.5 is that the expected path length for the data point is $c(n)$.
- we compare this value to one determined dynamically via k-means clustering

in practice this seems to come out similar to the 0.5 threshold approach
In the K-means approach, we assign the scores of training data into two clusters and take the points in the cluster with greater mean as the anomaly points.

Datasets

- Used a number of standard batch anomaly detection datasets

	WBC	Pima	Thyroid	Satellite	Optdigits	Ionosphere	Wine	SMTP	CICIDS
#Instances	278	768	3772	6435	5216	351	129	95156	691406
#Features	30	8	6	36	64	33	13	3	77
% anomalies	37%	35%	2.5%	32%	3%	36%	7.7%	0.03%	36%

- Also created 4 synthetic datasets (a-d)
- Online AD problems were simulated using stratified sampling in fix steps

Datasets

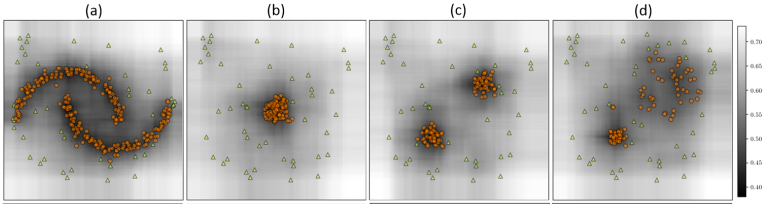
- Used a number of standard batch anomaly detection datasets

	WBC	Pima	Thyroid	Satellite	Optdigits	Ionosphere	Wine	SMTP	CICIDS
#Instances	278	768	3772	6435	5216	351	129	95156	691406
#Features	30	8	6	36	64	33	13	3	77
% anomalies	37%	35%	2.5%	32%	3%	36%	7.7%	0.03%	36%

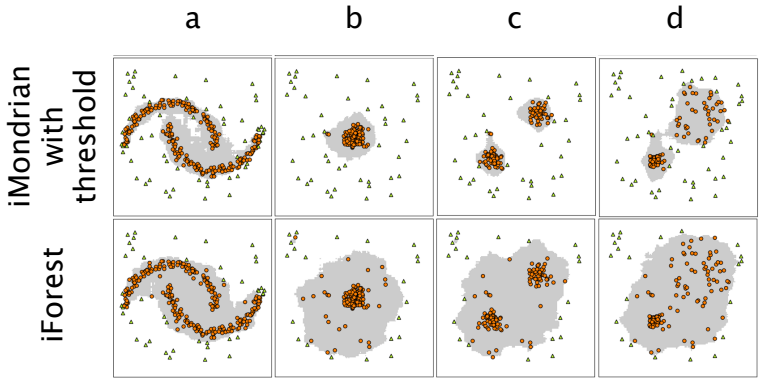
- Also created 4 synthetic datasets (a-d)
- Online AD problems were simulated using stratified sampling in fix steps we kept equal proporiton of points from each class

iMondrian Scores

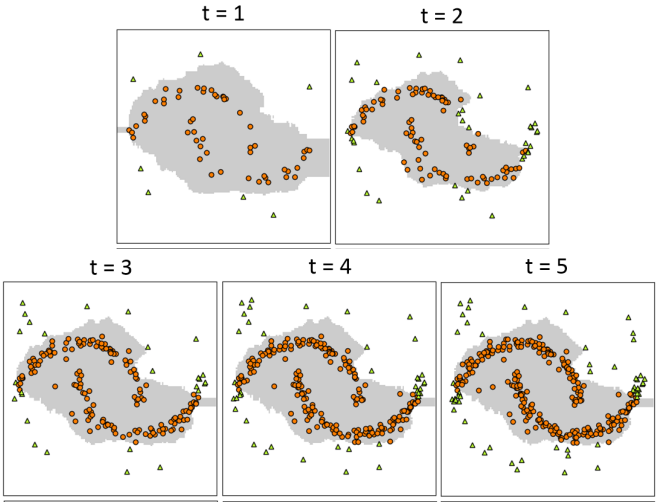
Raw iMondrian scores on the four synthetic datasets:



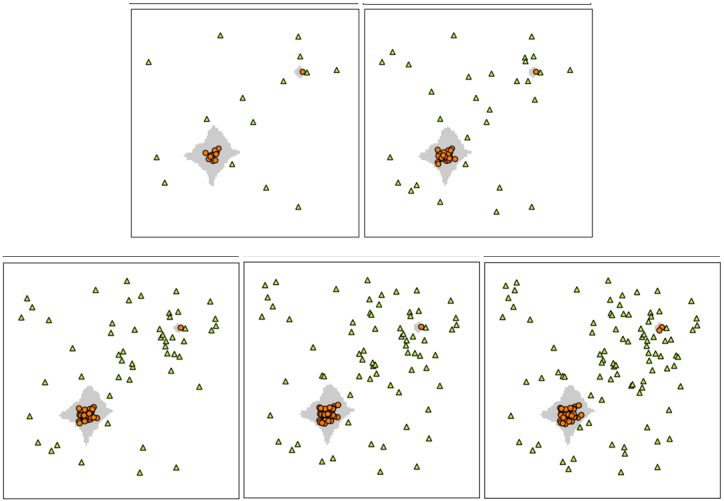
Synthetic Data - Static



Synthetic Data (a) - Streaming



Synthetic Data (d) - Streaming



Batch Comparative Results

			WBC	Pima	Optdigits	Wine	SMTP
iMForest	**Train:**	**Time:**	2.40 ± 0.01	2.54 ± 0.04	5.28 ± 0.03	0.48 ± 0.00	68.64 ± 1.11
		AUC:	86.35 ± 1.31	63.63 ± 1.02	72.90 ± 3.49	99.01 ± 0.16	86.76 ± 1.47
	Test:	**Time:**	0.04 ± 0.00	0.07 ± 0.01	0.35 ± 0.00	0.02 ± 0.00	7.42 ± 0.11
		AUC:	86.25 ± 5.02	63.74 ± 9.39	73.00 ± 7.64	99.71 ± 0.28	85.12 ± 14.25
iForest	**Train:**	**Time:**	0.14 ± 0.00	0.14 ± 0.00	0.81 ± 0.01	0.08 ± 0.00	4.41 ± 0.05
		AUC:	78.75 ± 1.62 ↑	67.49 ± 1.36	68.38 ± 4.64 ↑	79.56 ± 10.59 ↑	90.74 ± 1.37
	Test:	**Time:**	0.01 ± 0.00	0.01 ± 0.00	0.03 ± 0.00	0.01 ± 0.00	0.19 ± 0.00
		AUC:	78.81 ± 6.20 ↑	68.00 ± 5.14	68.36 ± 8.11 ↑	76.09 ± 10.11 ↑	89.44 ± 10.02
LOF	**Train:**	**Time:**	0.01 ± 0.00	0.01 ± 0.00	1.84 ± 0.05	0.01 ± 0.00	1.05 ± 0.06
		AUC:	61.12 ± 1.56 ↑	49.91 ± 1.41 ↑	60.84 ± 1.67 ↑	98.70 ± 1.29 ↑	53.51 ± 7.25 ↑
	Test:	**Time:**	0.01 ± 0.00	0.01 ± 0.00	2.13 ± 0.08	0.01 ± 0.00	1.14 ± 0.04
		AUC:	61.94 ± 7.56 ↑	51.36 ± 7.50 ↑	61.12 ± 11.65 ↑	92.58 ± 5.69 ↑	56.23 ± 25.90 ↑
SVM	**Train:**	**Time:**	0.03 ± 0.00	0.04 ± 0.00	3.10 ± 0.01	0.01 ± 0.00	240.93 ± 3.81
		AUC:	49.40 ± 3.16 ↑	51.93 ± 0.02 ↑	50.52 ± 3.81 ↑	68.59 ± 4.25 ↑	84.14 ± 1.75 ↑
	Test:	**Time:**	0.01 ± 0.00	0.01 ± 0.00	0.15 ± 0.00	0.01 ± 0.00	4.89 ± 0.09
		AUC:	94.21 ± 1.35	60.01 ± 8.37 ↑	37.49 ± 7.41 ↑	91.13 ± 3.83 ↑	83.06 ± 17.75 ↑

Online Comparative Results

	Stages	Pima	Thyroid	SMTP	CICIDS
iMForest	**Time:**	1.25	6.59	185.33	2.6E3
	AUC:	70.19	95.41	95.48	71.02
	Time:	1.48	8.40	364.03	1.0E4
	AUC:	68.07	94.27	95.01	70.95
	Time:	1.59	9.20	319.45	5.9E3
	AUC:	65.45	94.65	96.58	70.76
	Time:	1.72	10.32	349.33	7.3E3
	AUC:	64.51	94.58	93.84	70.80
	Time:	1.88	11.29	393.93	8.6E3
	AUC:	65.50	94.64	92.87	70.83
Incremental LOF	**Time:**	0.001	0.006	0.71	4.1E2
	AUC:	58.81 ↑	85.61 ↑	94.90 ↑	46.58 ↑
	Time:	0.001	0.02	1.87	1.5E3
	AUC:	55.13 ↑	70.62 ↑	95.44	46.06 ↑
	Time:	0.001	0.04	3.93	3.3E3
	AUC:	53.31 ↑	72.34 ↑	58.59 ↑	45.99 ↑
	Time:	0.003	0.07	5.39	4.3E3
	AUC:	49.10 ↑	69.61 ↑	52.79 ↑	46.00 ↑

Comparative Results

- In most cases iMondrian outperforms all the baseline methods.
- In terms of time it is faster than osPCA1 and osPCA2 but slower than other methods.
- While there are methods that can do well on *batch anomaly detection* or *online anomaly detection* there are no other algorithms which
  - do equally well on *both*
  - do so without a complex algorithmic approach

Lecture Outline

1 What is Anomaly Detection?

2 Anomaly Detection By Density

3 Anomaly Detection By Isolation : Isolation Forests

4 Anomaly Detection By Isolation : iMondrian Forests

5 Anomaly Detection By Classification : One-Class SVM

6 Comparing Algorithms

7 References and Further Reading

One-Class SVM

- One-class SVM (OC-SVM) [SWS$^+$00] estimates a function to be either 1 and -1 in the regions with *high and low density of data*, respectively.
- This uses the **One-vs-All** formulation of Anomaly Detection.
- Kernels can be used to map the data to a higher dimensional feature space for a possible better performance.
- A new one-class margin parameter ν is also defined which represents
  - an upper bound on the number of outliers
  - a lower bound on the number of Support Vectors (ie. where $\alpha_i \neq 0$)
  - ν is sometimes also called a "frontier" variable

How One-class SVM Works

Original *SVM* Formulation:

$$\min_{w,b,\zeta} \frac{1}{2} w^T w + C \sum_{i=1}^{n} \zeta_i$$

$$\text{subject to } y_i(w^T \phi(x_i) + b) \geq 1 - \zeta_i,$$
$$\zeta_i \geq 0, i = 1, \ldots, n$$

$\nu - SVM$ Formulation:

$$\min_{w \in F, \boldsymbol{\xi} \in \mathbb{R}^\ell, \rho \in \mathbb{R}} \quad \tfrac{1}{2}\|w\|^2 + \tfrac{1}{\nu\ell} \textstyle\sum_i \xi_i - \rho$$

$$\text{subject to} \quad (w \cdot \Phi(\mathbf{x}_i)) \geq \rho - \xi_i, \quad \xi_i \geq 0$$

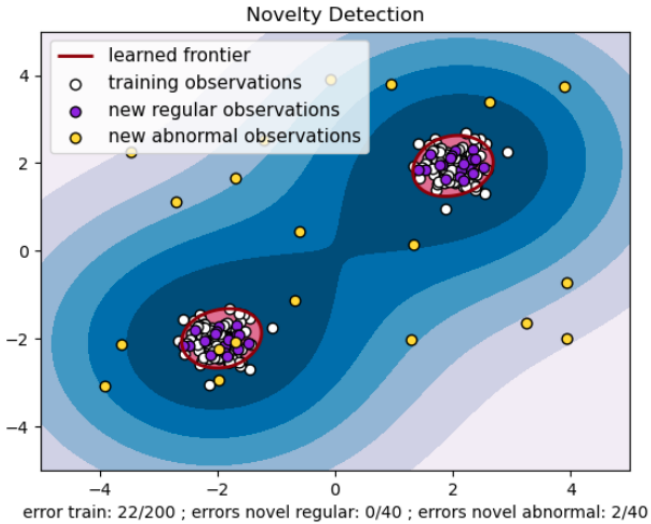


Figure: From scikit-learn

Lecture Outline

1 What is Anomaly Detection?

2 Anomaly Detection By Density

3 Anomaly Detection By Isolation : Isolation Forests

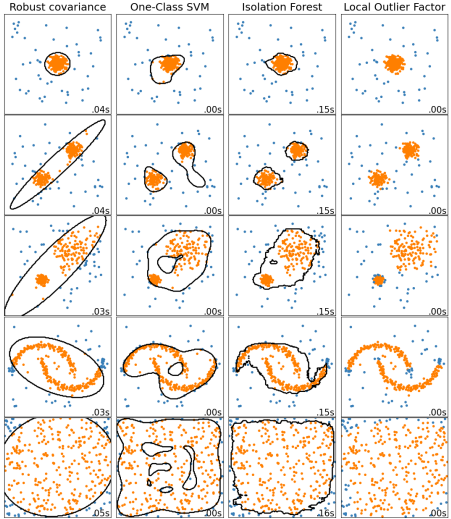
4 Anomaly Detection By Isolation : iMondrian Forests

5 Anomaly Detection By Classification : One-Class SVM

6 Comparing Algorithms

7 References and Further Reading

Comparison of Algorithms



A Review of Tree-Based Approaches for Anomaly Detection

Table 2 ROC AUC score of a selection of methods from literature.

	Code available	HTTP	SMTP	Forest C.	Shuttle	Mammogr.	Satellite	Pima	Breastw	Arrhytmia	Ionosph.
IF	✓	**1.00**	0.88	0.88	**1.00**	0.86	0.71	0.67	**0.99**	0.80	0.85
SCIForest	✓	**1.00**	-	0.74 [98]	**1.00** [98]	0.59 [11]	0.71 [98]	0.65 [98]	0.98 [98]	0.72 [98]	0.91 [93]
EIF	✓	*0.99*	0.85 [37]	0.92	0.99 [37]	0.86	0.78	0.70	***0.99***	0.80 [37]	0.91
RRCF	✓	0.99 [29]	0.89 [29]	*0.91*	0.91 [18]	0.83 [18]	0.68 [18]	0.59 [18]	0.64 [18]	0.74 [18]	0.90 [18]
IMF	✓	***1.00***	0.87	*0.90*	*0.99*	*0.74*	0.74	0.64	*0.97*	*0.80*	0.86
MPF		**1.00**	0.84	0.77	0.51	0.87	0.70	0.66	0.97	0.81	0.88
PIDForest	✓	0.99	**0.92**	0.84	**0.99**	0.84	0.70	0.70 [18]	0.99	-	0.84 [18]
OPHiForest		-	-	-	0.99	-	0.77	0.72	0.96	0.78	0.93
LSHiForest	✓	-	-	0.94	0.97	-	0.77	0.71	0.98	0.78	0.91
HIF	✓	-	0.90	-	**1.00**	**0.88**	0.74	0.70	*0.98*	0.80	*0.86*
HEIF		-	0.90	-	0.99	0.83	0.73	0.72	-	0.80	-
OneClassRF	✓	0.98	**0.92**	0.85	0.95	-	-	0.71	-	0.70	0.90
T-Forest		0.99	-	-	0.99	-	0.68	0.71	-	**0.84**	0.94
EGiTree		-	-	**0.97**	0.94	-	0.73	-	-	-	0.94
GIF	✓	-	-	0.94	-	0.87	**0.86**	**0.84**	-	-	-
dForest		**1.00**	-	-	**1.00**	-	-	0.75	**0.99**	-	**0.97**
ReMass IF		**1.00**	0.88	0.96	**1.00**	0.86	0.71	-	**0.99**	0.80	0.89
HSF	✓	**1.00**	0.90	0.89	**1.00**	0.86	-	0.69	**0.99**	**0.84**	0.80

Selected algorithms are: Isolation Forest (IF) [53], Split-Criteria Isolation Forest (SCIForest) [54], Extended Isolation Forest (EIF) [33], Robust Random Cut Forest (RRCF) [30], Isolation Mondrian Forest (IMF) [60], Mondrian Polya Forest (MPF) [18], Partial Identification Forest (PIDForest) [29], Order Preserving Hashing Based Isolation Forest (OPHIF) [93], Locality Sensitive Hashing Isolation Forest (LSHiForest) [98], Hybrid Isolation Forest (HIF) [64], Hybrid Extended Isolation Forest (HEIF) [37], One-class Random Forest (OneClassRF) [27], Trident Forest (T-Forest) [97], Entropy-based Greedy Isolation Tree (EGiTree) [50], Generalized Isolation Forest (GIF) [11], Distribution Forest (dForest) [95], Re-Mass Isolation Forest (ReMass IF) [6], Half-Spaces Forest (HSF) [84].

Figure: From `https://link.springer.com/chapter/10.1007/978-3-030-83819-5_7`

Lecture Outline

1 What is Anomaly Detection?

2 Anomaly Detection By Density

3 Anomaly Detection By Isolation : Isolation Forests

4 Anomaly Detection By Isolation : iMondrian Forests

5 Anomaly Detection By Classification : One-Class SVM

6 Comparing Algorithms

7 References and Further Reading

Deep Anomaly Detection

A more recent review of use of deep learning and generative models for anomaly detection can be found in [RKV+21].

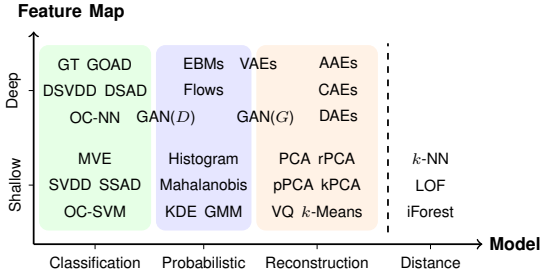


Figure: Anomaly detection approaches arranged *Model* and *Feature Map* to obtain four main groups of algorithms. Besides Model and Feature Map, we identify Loss, Regularization, and Inference Mode as other important modeling components of the anomaly detection problem. From [RKV+21].

References


Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander, *LOF: identifying density-based local outliers*, ACM Sigmod Record **29** (2000), no. 2, 93–104.


Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou, *Isolation forest*, 2008 Eighth IEEE International Conference on Data Mining, IEEE, 2008, pp. 413–422.


_____, *Isolation-based anomaly detection*, ACM Transactions on Knowledge Discovery from Data (TKDD) **6** (2012), no. 1, 3.


Yuh-Jye Lee, Yi-Ren Yeh, and Yu-Chiang Frank Wang, *Anomaly detection via online oversampling principal component analysis*, IEEE transactions on knowledge and data engineering **25** (2013), no. 7, 1460–1470.


Haoran Ma, Benyamin Ghojogh, Maria N Samad, Dongyu Zheng, and Mark Crowley, *Isolation mondrian forest for batch and online anomaly detection*, IEEE International Conference on Systems, Man, and Cybernetics (IEEE-SMC-2020) (Toronto, Canada (virtual)), IEEE, IEEE SMC, October 2020, p. 7.


Dragoljub Pokrajac, Aleksandar Lazarevic, and Longin Jan Latecki, *Incremental local outlier detection for data streams*, 2007 IEEE symposium on CIDM, IEEE, 2007, pp. 504–515.


Lukas Ruff, Jacob R. Kauffmann, Robert A. Vandermeulen, Grégoire Montavon, Wojciech Samek, Marius Kloft, Thomas G. Dietterich, and Klaus-Robert Müller, *A Unifying Review of Deep and Shallow Anomaly Detection*, Proceedings of the IEEE (2021), 1–40, arXiv: 2009.11732.


Bernhard Schölkopf, Robert C Williamson, Alex J Smola, John Shawe-Taylor, and John C Platt, *Support vector method for novelty detection*, NeurIPS conference, 2000, pp. 582–588.


Yi-Ren Yeh, Zheng-Yi Lee, and Yuh-Jye Lee, *Anomaly detection via over-sampling principal component analysis*, New Advances in Intelligent Decision Technologies, Springer, 2009, pp. 449–458.