Outlier Detection

Ninghao Liu

University of Georgia

April 10, 2024

Overview

1. Background and Problem Definition
2. Outlier Detection (OD) Strategies
   - Supervised, Unsupervised, Semi-supervised
3. Outlier Detection Methods
   - {Statistical, Proximity, Clustering, Classification} based
   - k-means for Outlier Detection
   - Isolation Forest
4. Dealing with High-Dimensional Data in OD
   - Autoencoders for OD

1. A Real-World Application

Imagine you are a transaction auditor in a credit card company:

- Protect customers from credit card fraud.
- Pay special attention to card usages that are different from typical cases.
- Not-that-typical cases: the purchase occurs *far* from the owner's resident city, the purchase amount is much *bigger* than usual, very *frequent* purchases in a short time, ...
- Detect such transactions as soon as they occur and contact the card owner for verification.

1. Problem Definition

An **outlier** is a data object that deviates significantly from the rest of the objects, as if it were generated by a different mechanism.

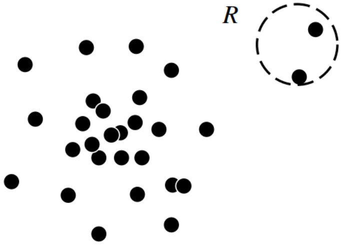


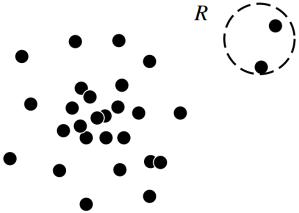
Figure 1: The objects in region R are outliers.

1. Problem Definition

Outlier detection (or **anomaly detection**) is the process of finding data objects with behaviors that are very different from expectation.

Please note that:

- Outliers are different from *noisy* data.
- It may be important to justify *why* the outliers are detected.
- Outlier detection is also related to *novelty detection* (but they are not the same).



2. Outlier Detection Strategies

Outlier detector training strategies:

- Supervised Learning
- Unsupervised Learning
- Semi-supervised Learning
- ...

2. Outlier Detection Strategies

2.1 Supervised Learning for OD

Given a dataset $\{(\boldsymbol{x}_i, y_i)\}$, $1 \leq i \leq N$, train a classifier $f : \mathcal{X} \rightarrow \{0, 1\}$. For a new instance $\boldsymbol{x}$, it is classified as an outlier if $f(\boldsymbol{x}) = 1$, or a normal instance if $f(\boldsymbol{x}) = 0$.

2. Outlier Detection Strategies

2.1 Supervised Learning for OD

Given a dataset $\{(\boldsymbol{x}_i, y_i)\}$, $1 \leq i \leq N$, train a classifier $f : \mathcal{X} \rightarrow \{0, 1\}$. For a new instance $\boldsymbol{x}$, it is classified as an outlier if $f(\boldsymbol{x}) = 1$, or a normal instance if $f(\boldsymbol{x}) = 0$.

Challenges:

- Data imbalance: Methods are needed for handling data imbalance.
- Evaluation gap: Classification maximizes prediction accuracy, which may not be of interest in applications.
- Labeling difficulty: The sampled data for labeling may not sufficiently represent the outlier distribution.

2. Outlier Detection Strategies

2.2 Unsupervised Learning for OD

Given a dataset $\{\boldsymbol{x}_i\}$, $1 \leq i \leq N$, train a detector $f : \mathcal{X} \rightarrow \{0, 1\}$. For a new instance $\boldsymbol{x}$, it is regarded as an outlier if $f(\boldsymbol{x}) = 1$, or a normal instance if $f(\boldsymbol{x}) = 0$.

2. Outlier Detection Strategies

2.2 Unsupervised Learning for OD

Given a dataset $\{\boldsymbol{x}_i\}$, $1 \leq i \leq N$, train a detector $f : \mathcal{X} \to \{0, 1\}$. For a new instance $\boldsymbol{x}$, it is regarded as an outlier if $f(\boldsymbol{x}) = 1$, or a normal instance if $f(\boldsymbol{x}) = 0$.

- Implicit assumption: The normal objects are "clustered".
- This assumption may not be true in some cases.
  - Outliers may also be "clustered".
  - Normal objects are diversely distributed.
- Computational challenge: We have to process a large population of non-target data entries (i.e., the normal objects) before one can touch the real meat (i.e., the outliers).

2. Outlier Detection Strategies

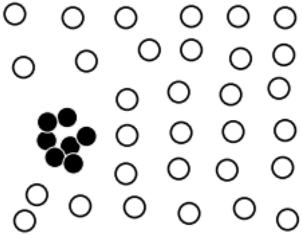


Figure 2: The black objects form outliers collectively.

2. Outlier Detection Strategies

2.3 Semi-supervised Learning for OD

Given a dataset $\{\boldsymbol{x}_i\}$ and $\{y_j\}$, $1 \leq i \leq N$, $j \in \mathcal{J}$, $|\mathcal{J}| \ll N$, train a detector $f : \mathcal{X} \to \{0, 1\}$.

- Most of the instances are unlabeled, so effective unsupervised methods serve as the important basis.
- Both labeled normal/outlier instances are possible.

3. Outlier Detection Methods

- Statistical Methods
- Clustering-Based Methods
- Proximity-Based Methods
- ...

3.1 Statistical Methods

The general idea: Learn a probability density function $p(\boldsymbol{x})$, denoting the probability that $\boldsymbol{x}$ is generated by the normal mechanism.

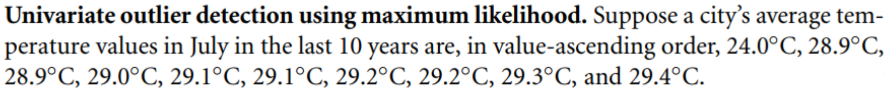
- Parametric methods: Assume p follows certain priori distribution (e.g., gaussian) with parameters θ to be estimated.
- Non-parametric methods: Do not assume a priori distribution.

3.1 Statistical Methods

Steps of parametric methods:

- Find a certain distribution to work with.
- Estimate the parameters θ of the distribution.
- Calculate $p(\boldsymbol{x};\theta)$, and decide if the probability is low enough.

Example 1:



Univariate outlier detection using maximum likelihood. Suppose a city's average temperature values in July in the last 10 years are, in value-ascending order, 24.0°C, 28.9°C, 28.9°C, 29.0°C, 29.1°C, 29.1°C, 29.2°C, 29.2°C, 29.3°C, and 29.4°C.

$$p(\boldsymbol{x}) = \frac{1}{\sigma\sqrt{2\pi}} \cdot \exp\left(-\frac{1}{2}\frac{(x-\mu)^2}{\sigma^2}\right)$$

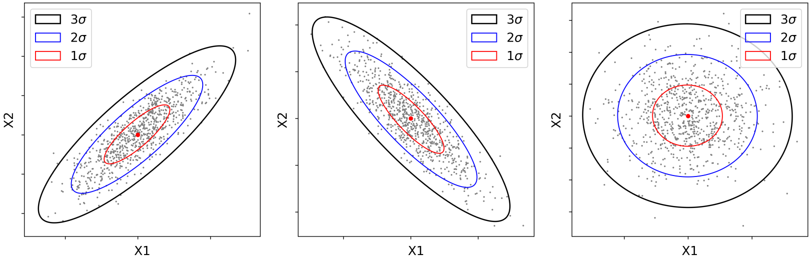
3.1 Statistical Methods

Steps of parametric methods:

- Find a certain distribution to work with.
- Estimate the parameters θ of the distribution.
- Calculate $p(\boldsymbol{x};\theta)$, and decide if the probability is low enough.

Example 2:

$$p(\boldsymbol{x}) = (2\pi)^{-\frac{k}{2}} \cdot \det(\boldsymbol{\Sigma})^{-\frac{1}{2}} \cdot \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^\intercal \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right)$$



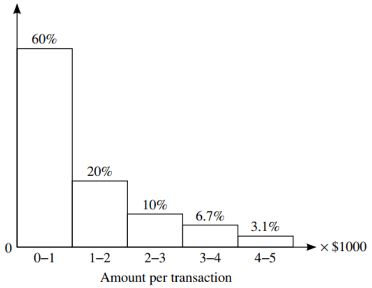
3.1 Statistical Methods

Commonly used non-parametric techniques:

- Histogram
- Kernel based density estimation

Example:

Outlier detection using a histogram. *AllElectronics* records the purchase amount for every customer transaction. Figure 12.5 uses a histogram (refer to Chapters 2 and 3) to graph these amounts as percentages, given all transactions.



Q: Should a transaction amounted $6500 be regarded as an outlier?

3.1 Statistical Methods

Steps of histogram methods:

- Build the histogram from data.
- Given an instance $\boldsymbol{x}$, check it against the histogram, and assign it with a outlier score of $\frac{1}{hist(\boldsymbol{x})}$.

Challenges

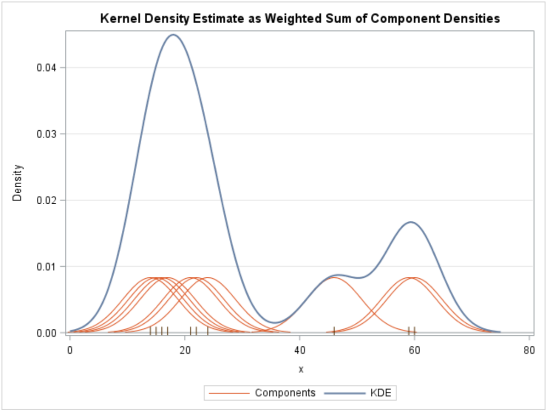
- How to choose the bin size?

3.1 Statistical Methods

Another Method: Treat an observed object as an indicator of high probability density in the surrounding region.

3.1 Statistical Methods

Another Method[1]: Treat an observed object as an indicator of high probability density in the surrounding region.



[1]`https://blogs.sas.com/content/iml/2016/07/27/visualize-kernel-density-estimate.html`

3.1 Statistical Methods

Kernel density estimation

Given samples $\{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_N\}$ and a test instance $\boldsymbol{x}$,

$$p(\boldsymbol{x}) = \frac{1}{Nh} \sum_{i=1}^{N} K(\frac{\boldsymbol{x} - \boldsymbol{x}_i}{h}) \tag{1}$$

where $K()$ is the kernel function, h is the bandwidth parameter.

A kernel $K()$ is a non-negative real-valued function that satisfies:

- $\int_{-\infty}^{+\infty} K(u)du = 1$,
- $K(u) = K(-u)$.

A frequently used kernel is the RBF kernel: $K(\boldsymbol{x}) = \exp(-\gamma \cdot \|\boldsymbol{x}\|_2^2)$.

3.1 Statistical Methods

Properties of statistical methods:

- Statistically justifiable
- Handling **high-dimensional data** is challenging
- Computational cost: For simple parametric models, it is linear time cost. For kernel density estimation, the model learning cost can be up to quadratic.

3.3 Clustering-Based Methods

Possible scenarios:

- Does the instance belong to any cluster?
    - If no, then an outlier.
- Is there a large distance between the instance and the cluster to which it is closest?
    - If yes, then an outlier.
- Is the instance part of a small or sparse cluster?
    - If yes, then the whole cluster is outlying.

Usually **two steps**: (1) clustering, (2) outlier score measuring.

- Use existing outlier detection methods for the second step.

3.3 Clustering-Based Methods

A review of k-mean clustering:

- Given a set of observations $(x_1, x_2, ..., x_N)$, k-means clustering aims to partition the N observations into $k(\leq n)$ sets $S = \{S_1, S_2, ..., S_k\}$.
- Formally, the objective is to find:

$$\underset{S}{\operatorname{argmin}} \sum_{i=1}^{k} \sum_{\boldsymbol{x} \in S_i} \|\boldsymbol{x} - \boldsymbol{\mu_i}\|^2, \tag{2}$$

  where $\boldsymbol{\mu_i}$ is the mean of S_i.
- This is equivalent to:

$$\underset{S}{\operatorname{argmin}} \sum_{i=1}^{k} \frac{1}{2|S_i|} \sum_{\boldsymbol{x},\boldsymbol{y} \in S_i} \|\boldsymbol{x} - \boldsymbol{y}\|^2, \tag{3}$$

  i.e., minimizing the pairwise squared deviations of points in the same cluster.

3.3 Clustering-Based Methods

Equivalence between the two objectives:

$$\underset{S}{\operatorname{argmin}} \sum_{i=1}^{k} \sum_{\boldsymbol{x} \in S_i} \|\boldsymbol{x} - \boldsymbol{\mu_i}\|^2$$

$$\underset{S}{\operatorname{argmin}} \sum_{i=1}^{k} \frac{1}{2|S_i|} \sum_{\boldsymbol{x},\boldsymbol{y} \in S_i} \|\boldsymbol{x} - \boldsymbol{y}\|^2$$

3.3 Clustering-Based Methods

A review of k-means clustering:

- Initiate the k mean vectors $\{\boldsymbol{\mu_1}, \boldsymbol{\mu_2}, ..., \boldsymbol{\mu_k}\}$.
- Alternate between the two steps as below.
  1. **Assignment:** Assign each instance to the cluster with the nearest mean vector:

$$S_i^{(t)} = \{\boldsymbol{x}_p : \|\boldsymbol{x}_p - \boldsymbol{\mu}_i\| \leq \|\boldsymbol{x}_p - \boldsymbol{\mu}_j\|, \forall j, 1 \leq j \leq k\} \quad (4)$$

  2. **Update:** Recalculate mean vectors

$$\boldsymbol{u}_i = \frac{1}{|S_i^{(t)}|} \sum_{\boldsymbol{x}_p \in S_i^{(t)}} \boldsymbol{x}_p \quad (5)$$

- The algorithm converges when the assignments no longer change.

3.3 Clustering-Based Methods

Some additional details in k-means:

- How to initialize?
  - The Forge method.
  - The Random Partition method.
- How to find the best k?
  - Through validation.
  - We could use the Gap Statistic [2].
- Limitations of k-mean?

[2]"Estimating the number of clusters in a data set via the gap statistic." Journal of the Royal Statistical Society. 2001.

3.3 Clustering-Based Methods

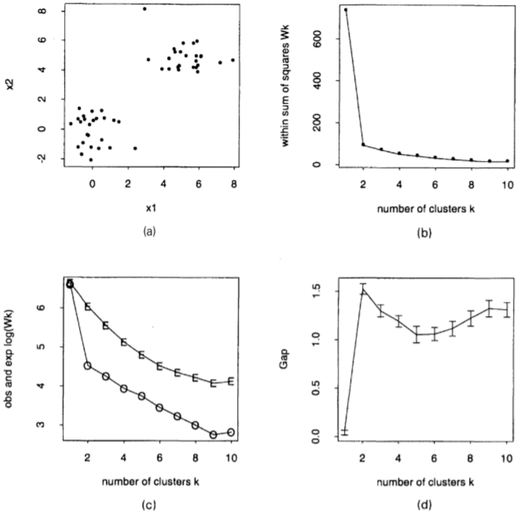


Fig. 1. Results for the two-cluster example: (a) data; (b) within sum of squares function W_k; (c) functions $\log(W_k)$ (O) and $\hat{E}_n^*\{\log(W_k)\}$ (E); (d) gap curve

3.3 Clustering-Based Methods

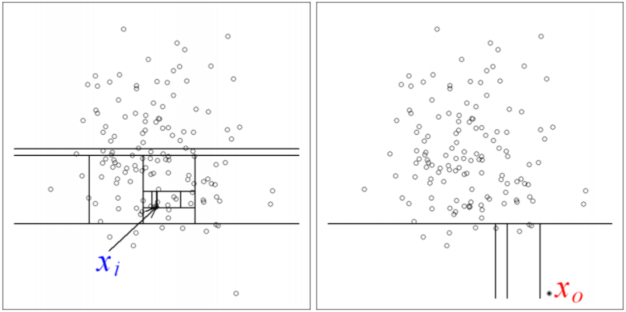
Properties of clustering-based methods:

- Obtain a summary of data
- Good explainability
- The first step could be time consuming
- The detection performance largely depends on clustering quality
    - Gaussian Mixture Model
    - Hierarchical Clustering
    - Spectral Clustering
    - Deep Model Based Clustering
    - ...

3.5 iForest

Isolation Forest (iForest) [3]

- **Isolation** means "separating an instance from the rest of the instances".
- Easily being separated → More likely to be an outlier.



(a) Isolating x_i (b) Isolating x_o

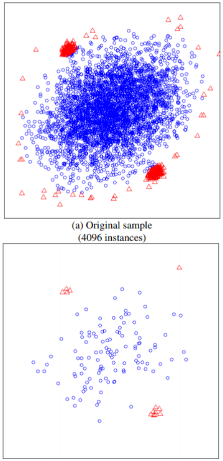
[3] "Isolation forest." ICDM. 2008.

3.5 iForest

Algorithm 1 : $iForest(X, t, \psi)$

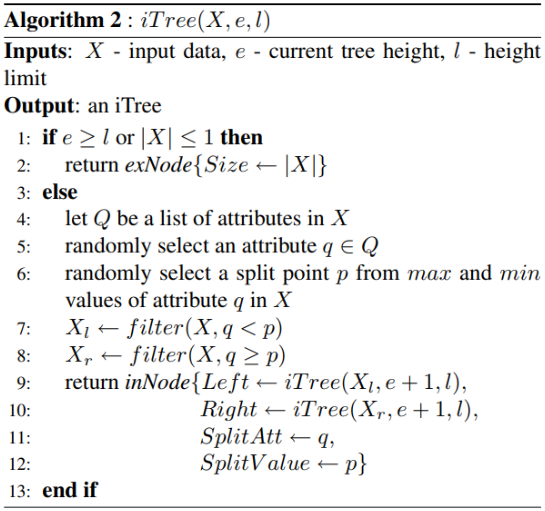
Inputs: X - input data, t - number of trees, ψ - su
sampling size
Output: a set of t *iTrees*

```
1: Initialize $Forest$
2: set height limit $l = ceiling(\log_2 \psi)$
3: for $i = 1$ to $t$ do
4:     $X' \leftarrow sample(X, \psi)$
5:     $Forest \leftarrow Forest \cup iTree(X', 0, l)$
6: end for
7: return $Forest$
```


(a) Original sample
(4096 instances)

(b) Sub-sample
(128 instances)

3.5 iForest



Algorithm 2 : $iTree(X, e, l)$

Inputs: X - input data, e - current tree height, l - height limit

Output: an iTree

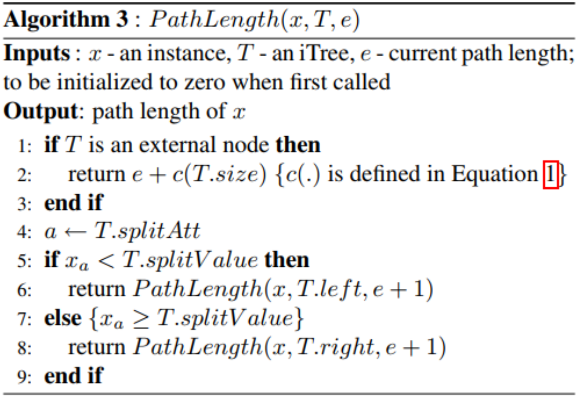
```
1:  if $e \geq l$ or $|X| \leq 1$ then
2:      return exNode$\{Size \leftarrow |X|\}$
3:  else
4:      let $Q$ be a list of attributes in $X$
5:      randomly select an attribute $q \in Q$
6:      randomly select a split point $p$ from $max$ and $min$
        values of attribute $q$ in $X$
7:      $X_l \leftarrow filter(X, q < p)$
8:      $X_r \leftarrow filter(X, q \geq p)$
9:      return inNode$\{Left \leftarrow iTree(X_l, e+1, l),$
10:                    $Right \leftarrow iTree(X_r, e+1, l),$
11:                    $SplitAtt \leftarrow q,$
12:                    $SplitValue \leftarrow p\}$
13: end if
```

There are two types of nodes: inNode and exNode.

Not always separate each point.

3.5 iForest

Complexity $O(nt\log\psi)$



Algorithm 3 : $PathLength(x, T, e)$

Inputs : x - an instance, T - an iTree, e - current path length; to be initialized to zero when first called

Output: path length of x

```
1: if $T$ is an external node then
2:     return $e + c(T.size)$ $\{c(.)$ is defined in Equation 1$\}$
3: end if
4: $a \leftarrow T.splitAtt$
5: if $x_a < T.splitValue$ then
6:     return $PathLength(x, T.left, e + 1)$
7: else $\{x_a \geq T.splitValue\}$
8:     return $PathLength(x, T.right, e + 1)$
9: end if
```

3.5 iForest

The outlier score is an instance $\boldsymbol{x}$ is:

$$s(\boldsymbol{x}, \psi) = 2^{-\frac{E(h(\boldsymbol{x}))}{c(\psi)}} \tag{6}$$

- $h(\boldsymbol{x})$: a single path length from root to $\boldsymbol{x}$.
- $c(\psi)$: normalization
- iTrees have an equivalent structure to Binary Search Tree (BST).
- The estimation of $E(h(x))$ for external node terminations is the same as the unsuccessful search in BST.
- $c(\psi) =$ the average path length of unsuccessful search in BST.

3.5 iForest

The outlier score is an instance $\boldsymbol{x}$ is:

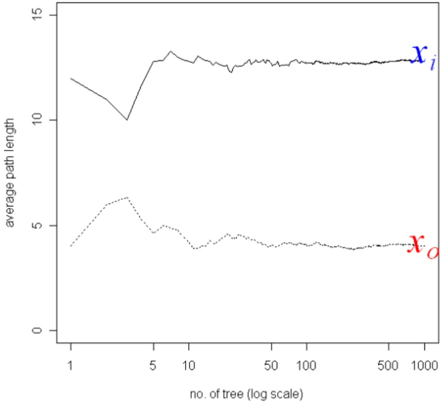
$$s(\boldsymbol{x}, \psi) = 2^{-\frac{E(h(\boldsymbol{x}))}{c(\psi)}} \tag{7}$$

- If instances return $s \approx 1$, then they are outliers.
- If instances have $s \ll 0.5$, then they are quite safe to be seen as normal instances.
- If all the instances return $s \approx 0.5$, then the entire sample does not really have any distinct outliers.

3.5 iForest

Two variables to choose in iForest:

- number of trees to build t (default: 100)
- sub-sampling size ψ (default: 256)



4 Outlier Detection in High-Dimensional Data

Dimensionality Curse: Various phenomena that arise when analyzing data in high-dimensional spaces that do not occur in low-dimensional ones [4].

- The *pairwise distances* between different samples in the space converging to the same value as the dimensionality of the data increases.
- As the dimensionality increases, the distance between objects may be heavily dominated by noise.
- The *proximity* or *similarity* between samples may not be qualitatively relevant in higher dimensions.

[4] `https://en.wikipedia.org/wiki/Curse_of_dimensionality`

4 Outlier Detection in High-Dimensional Data

Dimensionality Reduction: compress data

- Classical approaches
  - E.g., Principle Component Analysis (PCA)
- Deep learning approaches.
  - E.g., Robust Deep Autoencoder (RDA) [5]

[5] Zhou, Chong, and Randy C. Paffenroth. "Anomaly detection with robust deep autoencoders." KDD. 2017.

4.1 PCA

- A feature/dimensionality reduction technique.
- A relevant (but different) technique is feature selection.
- The goal of PCA is to transform the high-dimensional features to a lower-dimensional space.
- PCA is unsupervised.

What is a good data transformation?

4.1 PCA

What is a good data compression (transformation)?

- High compression rate (smaller data size).
- Key characteristics are maintained (keep more information).

In the deep learning era, the transformed/compressed data is also called as:

- the “representation” of data
- “latent representation”
- “embeddings”

4.1 PCA

- Let us start from the simple case where PCA projects the features to a **1-dimensional space**.
- Suppose we have n p-dimensional ($p > 1$) instances, $\boldsymbol{x}_1$, $\boldsymbol{x}_2$, ..., $\boldsymbol{x}_n \in \mathbb{R}^p$.
- We want to learn a mapper $\boldsymbol{a} \in \mathbb{R}^p$ that $\boldsymbol{a}^\intercal \boldsymbol{x}_i = z_i$. Here z_i is where $\boldsymbol{x}_i$ locates in the 1-dimensional space.
- PCA tries to solve:

$$\boldsymbol{a}^* = argmax_{\|\boldsymbol{a}\|=1} \underbrace{\frac{1}{n}(z_i - \overline{z})^2}_{\text{variance}}, \tag{8}$$

where PCA tries to find the projection with the maximum variance in reduced data.

4.1 PCA

- PCA tries to solve:

$$\boldsymbol{a}^* = argmax_{\|\boldsymbol{a}\|=1} \; \underbrace{\frac{1}{n}(z_i - \overline{z})^2}_{\text{variance}} . \tag{9}$$

- Q1: Why $\|\boldsymbol{a}\| = 1$?
- Q2: What does the objective imply?
  - Trying to make the data points distinguishable in the new space.
  - Why?

4.1 PCA

Since

$$\bar{z} = \frac{1}{n}\sum_{i=1}^{n} z_i = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{a}^T \boldsymbol{x}_i = \boldsymbol{a}^T\Big(\frac{1}{n}\sum_{i=1}^{n} \boldsymbol{x}_i\Big) = \boldsymbol{a}^T \bar{\boldsymbol{x}},$$

the problem can be written as

$$\begin{aligned}
\boldsymbol{a}^* &= \underset{||\boldsymbol{a}||=1}{\arg\max}\, \frac{1}{n}\sum_{i=1}^{n}(z_i - \bar{z})^2 \\
&= \underset{||\boldsymbol{a}||=1}{\arg\max}\, \frac{1}{n}\sum_{i=1}^{n}(\boldsymbol{a}^T\boldsymbol{x}_i - \bar{z})^2 \\
&= \underset{||\boldsymbol{a}||=1}{\arg\max}\, \frac{1}{n}\sum_{i=1}^{n}(\boldsymbol{a}^T\boldsymbol{x}_i - \boldsymbol{a}^T\bar{\boldsymbol{x}})^2 \\
&= \underset{||\boldsymbol{a}||=1}{\arg\max}\, \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{a}^T(\boldsymbol{x}_i - \bar{\boldsymbol{x}})(\boldsymbol{x}_i - \bar{\boldsymbol{x}})^T\boldsymbol{a} \\
&= \underset{||\boldsymbol{a}||=1}{\arg\max}\, \boldsymbol{a}^T \underbrace{\left(\frac{1}{n}\sum_{i=1}^{n}(\boldsymbol{x}_i - \bar{\boldsymbol{x}})(\boldsymbol{x}_i - \bar{\boldsymbol{x}})^T\right)}_{p\times p \text{ covariance matrix}} \boldsymbol{a} \\
&= \underset{||\boldsymbol{a}||=1}{\arg\max}\, \boldsymbol{a}^T\boldsymbol{C}\boldsymbol{a},
\end{aligned}$$

where $\boldsymbol{C} = \frac{1}{n}\sum_{i=1}^{n}(\boldsymbol{x}_i - \bar{\boldsymbol{x}})(\boldsymbol{x}_i - \bar{\boldsymbol{x}})^T$, denotes the covariance matrix.

4.1 PCA

The problem is

$$\begin{aligned} &\max_{\boldsymbol{a}} \ \boldsymbol{a}^\intercal \boldsymbol{C}\boldsymbol{a} \\ &s.t., \ \|\boldsymbol{a}\|^2 = 1. \end{aligned} \tag{10}$$

How to solve this?

4.1 PCA

Eigen-Decomposition

A square $n \times n$ matrix $\boldsymbol{C}$ with n linearly independent eigenvectors can be factorized as:

$$\boldsymbol{C} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^{-1} \tag{11}$$

- $\boldsymbol{U}$ is a $n \times n$ matrix whose columns are eigenvectors of $\boldsymbol{C}$,
- $\boldsymbol{\Lambda}$ is a diagonal matrix whose diagonal elements are the corresponding eigenvalues.

If $\boldsymbol{C}$ is symmetric, then

$$\boldsymbol{C} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^{\intercal} = \sum_{i=1}^{n} \lambda_i \boldsymbol{u}_i \boldsymbol{u}_i^{\intercal}. \tag{12}$$

4.1 PCA

Q: What about mapping data into a k-dimensional space?
A: The PCA problem is then defined as:

$$\boldsymbol{A}^* = \underset{\boldsymbol{A}\in\mathbb{R}^{p\times k}:\boldsymbol{A}^\intercal\boldsymbol{A}=\boldsymbol{I}_k}{\operatorname{argmax}} \operatorname{trace}(\boldsymbol{A}^\intercal\boldsymbol{C}\boldsymbol{A}) \tag{13}$$

where $\boldsymbol{A} = [\boldsymbol{a}_1, \boldsymbol{a}_2, ..., \boldsymbol{a}_k] \in \mathbb{R}^{p\times k}$. PCA requires different projection vectors to be orthogonal.

$$\operatorname{trace}(\boldsymbol{A}^\intercal\boldsymbol{C}\boldsymbol{A}) = \sum_{i=1}^{k}(\boldsymbol{a}_i^\intercal\boldsymbol{C}\boldsymbol{a}_i), \tag{14}$$

which is a natural extension of 1-D PCA.

Q: How to solve this?

4.1 PCA

When $k = 2$, we want to solve

$$\begin{aligned} \max_{\boldsymbol{a}_2} \;\; & \boldsymbol{a}_2^\intercal \boldsymbol{C} \boldsymbol{a}_2 \\ s.t., \; & \|\boldsymbol{a}_i\|^2 = 1 \\ & \boldsymbol{a}_2^\intercal \boldsymbol{a}_1 = 0. \end{aligned} \tag{15}$$

How to solve this?

4.1 PCA

We then have the following theorem.

Theorem. (*Ky Fan*) Let $\boldsymbol{H} \in \mathbb{R}^{n\times n}$ be a symmetric matrix with eigenvalues

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n,$$

and the corresponding eigenvectors $\boldsymbol{U} = [\boldsymbol{u}_1, \ldots, \boldsymbol{u}_n]$. Then

$$\lambda_1 + \cdots \lambda_k = \max_{\boldsymbol{A}\in\mathbb{R}^{n\times k}:\boldsymbol{A}^T\boldsymbol{A}=\boldsymbol{I}_k} \text{trace}\left(\boldsymbol{A}^T\boldsymbol{H}\boldsymbol{A}\right).$$

And the optimal $\boldsymbol{A}^*$ is given by $\boldsymbol{A}^* = [\boldsymbol{u}_1, \ldots, \boldsymbol{u}_k]\boldsymbol{Q}$ with $\boldsymbol{Q}$ an arbitrary orthogonal matrix. ■

4.1 PCA

1. Note that in Eqn. (2), the covariance matrix $\boldsymbol{C}$ is a symmetric matrix. Given the above theorem, we directly obtain

$$\begin{aligned} \lambda_1 + \cdots \lambda_k &= \underset{\boldsymbol{A} \in \mathbb{R}^{n \times k} : \boldsymbol{A}^T \boldsymbol{A} = \boldsymbol{I}_k}{\arg\max} \operatorname{trace}\left(\boldsymbol{A}^T \boldsymbol{C} \boldsymbol{A}\right), \\ \boldsymbol{A}^* &= [\boldsymbol{u}_1, \ldots, \boldsymbol{u}_k]\boldsymbol{Q}, \end{aligned}$$

   where $\lambda_1, \ldots, \lambda_k$ are the k largest eigenvalues of the covariance matrix $\boldsymbol{C}$, and the solution $\boldsymbol{A}^*$ is the matrix whose columns are corresponding eigenvectors.

2. It also follows from the above theorem that solutions to PCA are not unique, and they differ by an orthogonal matrix. We used the special case where $\boldsymbol{Q} = \boldsymbol{I}$, *i.e.*, $\boldsymbol{A}^* = [\boldsymbol{u}_1, \ldots, \boldsymbol{u}_k]$.

4.1 PCA for Outlier Detection

Using PCA, we obtain the mappers $\boldsymbol{A} = \{\boldsymbol{u}_i\}$, $1 \leq i \leq k$. An original data instance $\boldsymbol{x}$ will be mapped to $\boldsymbol{z} = [z_1, z_2, ..., z_k]$. After mapping all the data, outlier detection could be done in the $\boldsymbol{z}$-space.

- Apply the detection methods we already discussed.
- Choose the mappers corresponding to higher or lower eigenvalues?
- How many k's to choose?

4.1 PCA for Outlier Detection

There is another way to understand PCA from the **data reconstruction** perspective.

- Assume $\boldsymbol{X}$ is centered.
- Data mapping: $\boldsymbol{Z} = \boldsymbol{A}^\intercal \boldsymbol{X}$.
- Reconstructed data: $\tilde{\boldsymbol{X}} = \boldsymbol{A}\boldsymbol{Z} = \boldsymbol{A}\boldsymbol{A}^\intercal \boldsymbol{X}$.
- Reconstruction error: $\|\boldsymbol{X} - \tilde{\boldsymbol{X}}\|_F = \|\boldsymbol{X} - \boldsymbol{A}\boldsymbol{A}^\intercal \boldsymbol{X}\|_F$.
- It has also been shown that:

$$\boldsymbol{A}\boldsymbol{A}^\intercal \boldsymbol{X} = \boldsymbol{B}^* = \underset{\boldsymbol{B}:\text{rank}(\boldsymbol{B})\leq k}{\text{argmin}} \|\boldsymbol{X} - \boldsymbol{B}\|_F. \tag{16}$$

  PCA projects the high-dimensional features to a lower-dimensional space with minimum reconstruction error.
  - Keep most of the information = Be able to recover.
- Instances with large reconstruction errors $\|\boldsymbol{x} - \boldsymbol{A}\boldsymbol{A}^\intercal \boldsymbol{x}\|_2$ are outliers.

4.1 PCA for Outlier Detection

In previous parts, we need to compute $\boldsymbol{C}$ first before doing PCA.
This step could be costly if p or n is large.
Can this step be avoided?

Singular Value Decomposition:
The singular value decomposition (SVD) of an $m \times n$ real matrix $\boldsymbol{R}$ (we assume $m \geq n$) can be written as:

$$\boldsymbol{R} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\intercal, \tag{17}$$

where $\boldsymbol{U} \in \mathbb{R}^{m\times m}$, $\boldsymbol{\Sigma} \in \mathbb{R}^{m\times n}$, and $\boldsymbol{V} \in \mathbb{R}^{n\times n}$, $\boldsymbol{U}\boldsymbol{U}^\intercal = \boldsymbol{U}^\intercal\boldsymbol{U} = \boldsymbol{I}_{m\times m}$, and $\boldsymbol{V}\boldsymbol{V}^\intercal = \boldsymbol{V}^\intercal\boldsymbol{V} = \boldsymbol{I}_{n\times n}$.

4.1 PCA for Outlier Detection

1. Pre-process data.

$$\begin{aligned} \boldsymbol{X} &= [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n] \in \mathbb{R}^{p\times n}, \\ \bar{\boldsymbol{X}} &= \frac{1}{n}\boldsymbol{X}\mathbf{1}_n \in \mathbb{R}^{p\times 1}, \\ \tilde{\boldsymbol{X}} &= (\boldsymbol{X} - \bar{\boldsymbol{X}}\mathbf{1}_n^T) \in \mathbb{R}^{p\times n}, \end{aligned}$$

2. We know $\boldsymbol{C} = \frac{1}{n}\sum_{i=1}^{n}(\boldsymbol{x}_i - \overline{\boldsymbol{x}})(\boldsymbol{x}_i - \overline{\boldsymbol{x}})^\intercal = \frac{1}{n}\tilde{\boldsymbol{X}}\tilde{\boldsymbol{X}}^\intercal$.
3. If the SVD of $\tilde{\boldsymbol{X}}$ is $\tilde{\boldsymbol{X}} = \boldsymbol{U}\tilde{\Sigma}\boldsymbol{V}$, the columns of $\boldsymbol{U}$ (left-singular vectors) are eigenvectors of $\tilde{\boldsymbol{X}}\tilde{\boldsymbol{X}}$.
4. We do not perform eigen-value decomposition of $\boldsymbol{C}$, but do singular-value decomposition of $\tilde{\boldsymbol{X}}$, to obtain the mappers $\boldsymbol{U}$.

4.1 PCA for Outlier Detection

A more efficient way of computing PCA can be described as:

$$\underbrace{\boldsymbol{X}}_{p\times n} \rightarrow \underbrace{\tilde{\boldsymbol{X}}}_{p\times n} \rightarrow \tilde{\boldsymbol{X}} = \boldsymbol{U}\tilde{\boldsymbol{\Sigma}}\boldsymbol{V}^T \rightarrow \boldsymbol{G} = \boldsymbol{U}_k \in \mathbb{R}^{p\times k}$$

$$\boldsymbol{G}^T\tilde{\boldsymbol{X}} = \boldsymbol{Z} \in \mathbb{R}^{k\times n}$$

4.1 Connection between PCA and Autoencoders

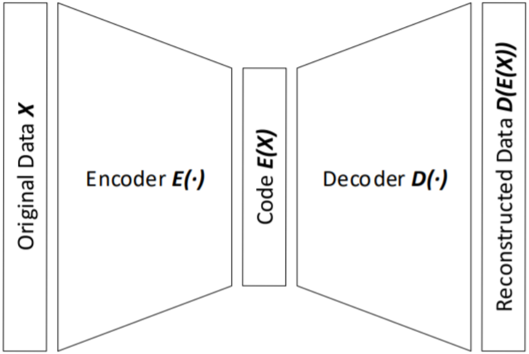


Figure 3: An autoendoer.

- What does each module in PCA correspond to?
- How to go deeper?

4.2 Outlier Detection with Deep Autoencoders

Deep Autoencoders

- A feed-forward multi-layer neural network.
- The desired output is the input itself.
  - Looks trivial?
  - Making hidden representation to be low-dimensional.
- The loss function of training an autoencoder:

$$\min_{D,E} \|\boldsymbol{X} - D(E(\boldsymbol{X}))\| \tag{18}$$

4.2 Outlier Detection with Deep Autoencoders

Robust PCA (RPCA)

- Try to reduce the sensitivity of PCA to outliers. Intuition?
- Split the data matrix into two parts

$$\boldsymbol{X} = \boldsymbol{L} + \boldsymbol{S}, \tag{19}$$

where $\boldsymbol{L}$ is a low-rank matrix, and $\boldsymbol{S}$ is a sparse matrix.

4.2 Outlier Detection with Deep Autoencoders

Robust PCA (RPCA)

- The problem can be formulated as:

$$\begin{aligned} \min_{\boldsymbol{L},\boldsymbol{S}} \; & \|\boldsymbol{L}\|_* + \lambda\|\boldsymbol{S}\|_1 \\ s.t.\; & \|\boldsymbol{X} - \boldsymbol{L} - \boldsymbol{S}\|_F^2 = 0, \end{aligned} \tag{20}$$

  where $\|\cdot\|_*$ is the nuclear norm, $\|\cdot\|_1$ is the L1 norm.
- If we let D and E to be linear mappers, then PCA is the optimal solution for recovering $\boldsymbol{L}$.

4.2 Outlier Detection with Deep Autoencoders

Robust Deep Autoencoders (RDA)

- The problem can be formulated as:

$$\begin{aligned} &\min_{\boldsymbol{L},\boldsymbol{S}} \|\boldsymbol{L} - D_\theta(E_\theta(\boldsymbol{L}))\|_2 + \lambda\|\boldsymbol{S}\|_1 \\ &s.t.\, \boldsymbol{X} - \boldsymbol{L} - \boldsymbol{S} = 0, \end{aligned} \tag{21}$$

  where both the encoder and the decoder are parameterized by θ.
- The first term encourages $\boldsymbol{L}$ to contain the "main features" in data.
- λ plays an essential role:
  - a smaller $\lambda \rightarrow$ encourage more data into $\boldsymbol{S}$
  - a larger $\lambda \rightarrow$ include less data in $\boldsymbol{S}$

4.2 Outlier Detection with Deep Autoencoders

Robust Deep Autoencoders (RDA)

- How to choose D_θ and E_θ?
  - Any commonly used deep autoencoder architectures could be used.
  - The paper choose a simple architecture:

$$\begin{aligned} E_\theta(\boldsymbol{x}) &= g(\boldsymbol{W}\boldsymbol{x} + b_E) \\ D_\theta(\boldsymbol{x}) &= g(\boldsymbol{W}^\intercal E_\theta(\boldsymbol{x}) + b_D) \end{aligned} \tag{22}$$

  - $g(x) = \frac{1}{1+\exp(-x)}$, a nonlinear transformation.

4.2 Outlier Detection with Deep Autoencoders

Robust Deep Autoencoders (RDA)

- Is $\|S\|_1$ always a good choice?
- What if each anomaly point is not a matrix entry, but the whole row or column?

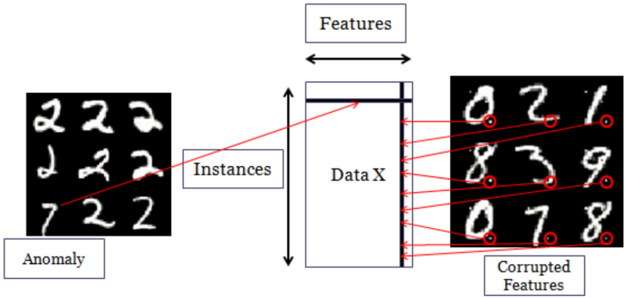


Figure 4: Two examples of group anomalies.

4.2 Outlier Detection with Deep Autoencoders

Regularization

- $\min \|\boldsymbol{x}\|_1$ vs $\min \|\boldsymbol{x}\|_2$
- Which one encourages a sparse $\boldsymbol{x}$?
- An example of taxes.
- Can we combine them?

4.2 Outlier Detection with Deep Autoencoders

Robust Deep Autoencoders (RDA)

- $l_{2,1}$ normalization encourages structured/group sparsity.
- $\|\boldsymbol{X}\|_{2,1} = \sum_{j=1}^{n} \|\boldsymbol{x}_j\|_2 = \sum_{j=1}^{n} (\sum_{i=1}^{m} |x_{i,j}|^2)^{1/2}$.
- $\|\boldsymbol{X}^\top\|_{2,1}$?

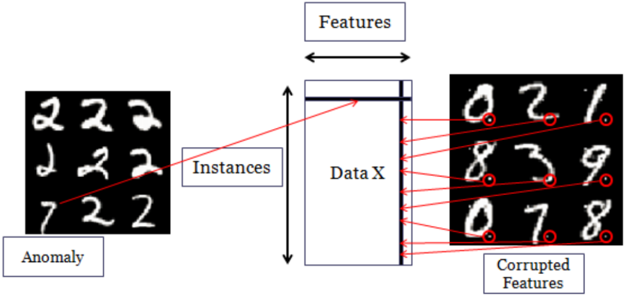


Figure 5: Two examples of group anomalies.

4.2 Outlier Detection with Deep Autoencoders

Robust Deep Autoencoders (RDA)

- The problem can be formulated as:

$$\min_{\boldsymbol{L},\boldsymbol{S}} \|\boldsymbol{L} - D_\theta(E_\theta(\boldsymbol{L}))\|_2 + \lambda(\|\boldsymbol{S}\|_1 \text{ or } \|\boldsymbol{S}\|_{2,1} \text{ or } \|\boldsymbol{S}^\intercal\|_{2,1})$$

$$s.t.\ \boldsymbol{X} - \boldsymbol{L} - \boldsymbol{S} = 0,$$

- Just denoising: choose $\|\boldsymbol{S}\|_1$.
- To find abnormal data instances: choose $\|\boldsymbol{S}^\intercal\|_{2,1}$.

4.2 Outlier Detection with Deep Autoencoders

Robust Deep Autoencoders (RDA)

- How to solve the problem?

$$\min_{\boldsymbol{L},\boldsymbol{S}} \|\boldsymbol{L} - D_\theta(E_\theta(\boldsymbol{L}))\|_2 + \lambda(\|\boldsymbol{S}\|_1 \text{ or } \|\boldsymbol{S}\|_{2,1} \text{ or } \|\boldsymbol{S}^\intercal\|_{2,1})$$
$$s.t. \, \boldsymbol{X} - \boldsymbol{L} - \boldsymbol{S} = 0,$$

- Alternating Direction Method of Multipliers (ADMM):
    - Divide the objective into several pieces.
    - Optimizes one of the pieces while keeping the others fixed.
- How to update $\boldsymbol{L}$ with respect to $\min_{\boldsymbol{L}} \|\boldsymbol{L} - D_\theta(E_\theta(\boldsymbol{L}))\|_2$?
- How to update $\boldsymbol{S}$ with respect to $\min_{\boldsymbol{S}} \|\boldsymbol{S}\|_1$?

4.2 Outlier Detection with Deep Autoencoders

Robust Deep Autoencoders (RDA)

The algorithm of solving

$$\min_{\boldsymbol{L},\boldsymbol{S}} \|\boldsymbol{L} - D_\theta(E_\theta(\boldsymbol{L}))\|_2 + \lambda(\|\boldsymbol{S}\|_1 \text{ or } \|\boldsymbol{S}\|_{2,1} \text{ or } \|\boldsymbol{S}^\intercal\|_{2,1})$$

$$s.t.\ \boldsymbol{X} - \boldsymbol{L} - \boldsymbol{S} = 0.$$

```
1  Initialize $\theta$, $\boldsymbol{S}$
2  while not converge:
     1  $\boldsymbol{L} = \boldsymbol{X} - \boldsymbol{S}$
     2  Minimize $\|\boldsymbol{L} - D_\theta(E_\theta(\boldsymbol{L}))\|_2$ with gradient descent
     3  $\boldsymbol{L} \leftarrow D_\theta(E_\theta(\boldsymbol{L}))$
     4  $\boldsymbol{S} = \boldsymbol{X} - \boldsymbol{L}$
     5  Optimize $\boldsymbol{S}$ using a proximal operator
     6  Check convergence.
```

4.2 Outlier Detection with Deep Autoencoders

Experiment Design

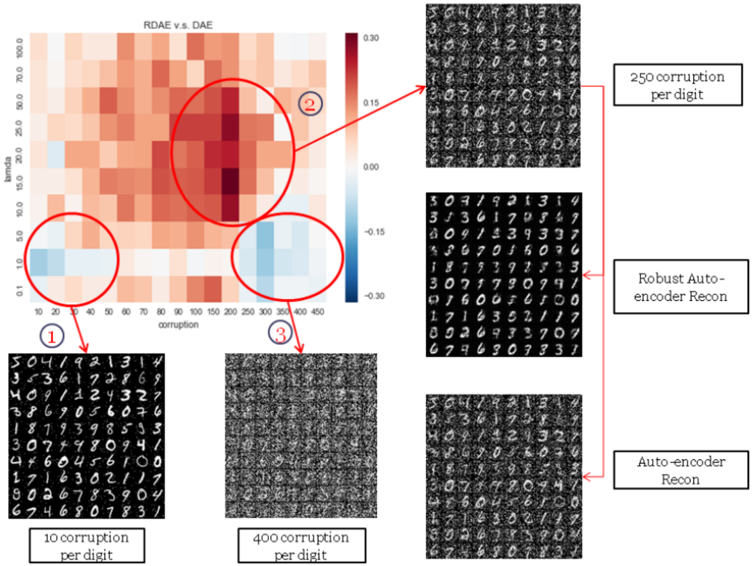
- Datasets
- Research Problems
  - Baseline methods
  - Metrics
  - Visualization
- Effect of hyperparameters or modules

4.2 Outlier Detection with Deep Autoencoders

Experiment Design in RDA

- Datasets: MNIST
- Research Problem 1: Whether excluding from data an independent sparse matrix using $\|\boldsymbol{S}\|_1$ could help the autoencoder in capturing the essential patterns in data.
  - Baseline methods: Autoencoder
  - Metrics: Classification error
- Experiment design:
  - Extract hidden features from $E_\theta(\boldsymbol{L})$.
  - Use the hidden features to build a classifier to classify digits.
  - Lower error $\rightarrow$ Better model.

4.2 Outlier Detection with Deep Autoencoders

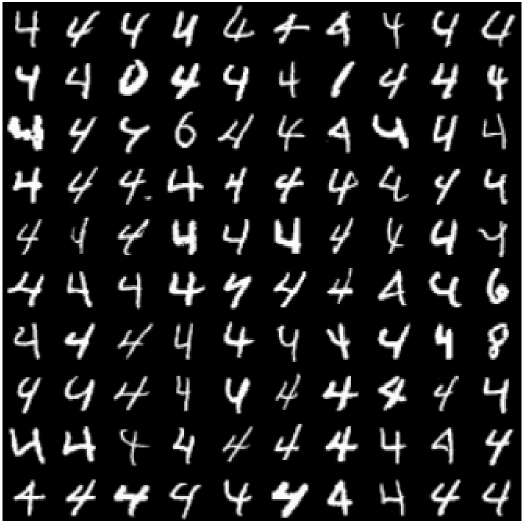


4.2 Outlier Detection with Deep Autoencoders

Experiment Design in RDA

- Datasets: MNIST
- Research Problem 2: Is $\|\boldsymbol{S}\|_{2,1}$ regularized RDA effective in detecting outliers?
  - Baseline method: Isolation Forest (IS)
  - Evaluation metric: F1-score
- Experiment design:
  - Compare detected outliers with the ground truths.
  - 0.64 (RDA) vs 0.37 (IS)

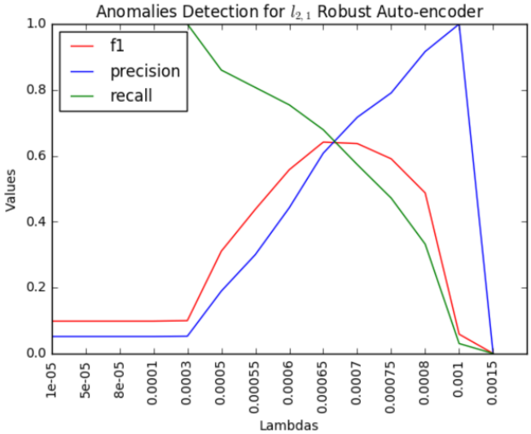
4.2 Outlier Detection with Deep Autoencoders



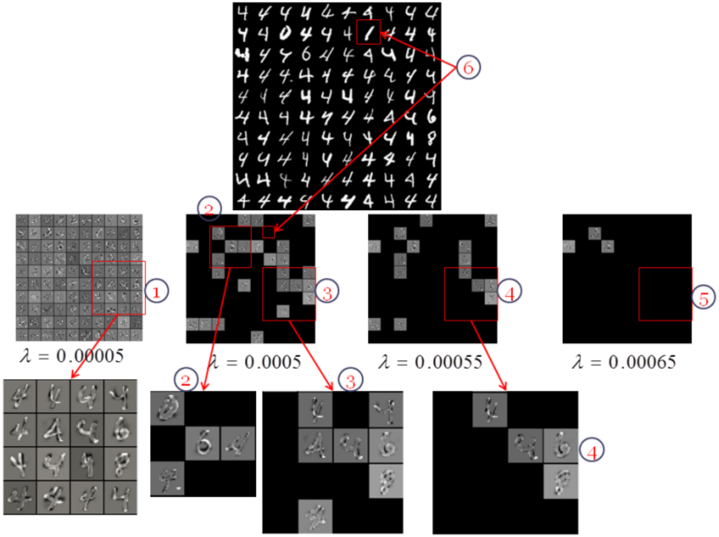
4.2 Outlier Detection with Deep Autoencoders

Experiment Design in RDA

- Datasets: MNIST
- Research Problem 3: Effect of λ?



4.2 Outlier Detection with Deep Autoencoders



4.2 Outlier Detection with Deep Autoencoders