Mathematical Modeling of Anomalies in Structured Large-Scale Data

Martin Flusser

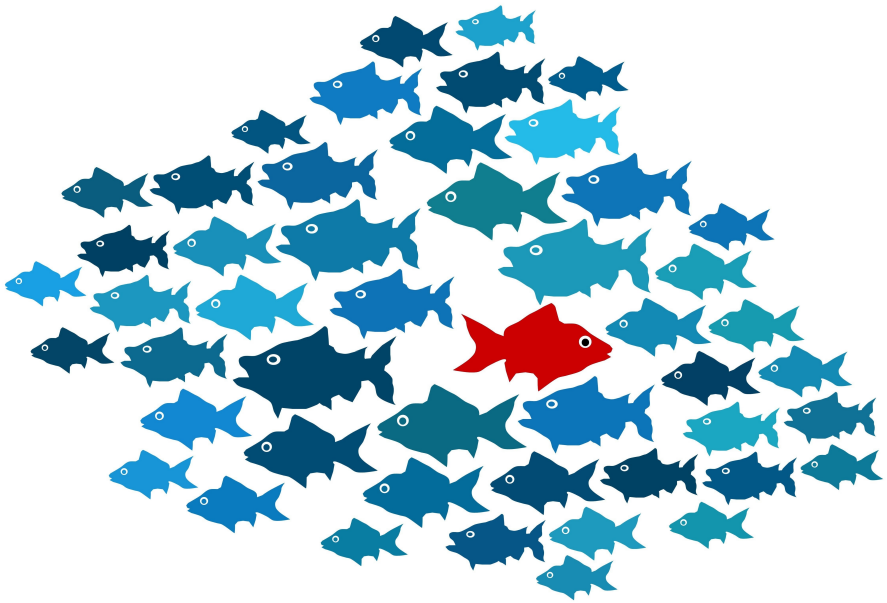
Faculty of Nuclear Sciences and Physical Engineering
Czech Technical University in Prague

May 9, 2019

Outline

1. Anomaly detection
2. Prior art
3. DANNMAD
4. Evaluation and benchmarks
5. Experimental evaluation
6. Outlook and conclusion

Anomaly detection

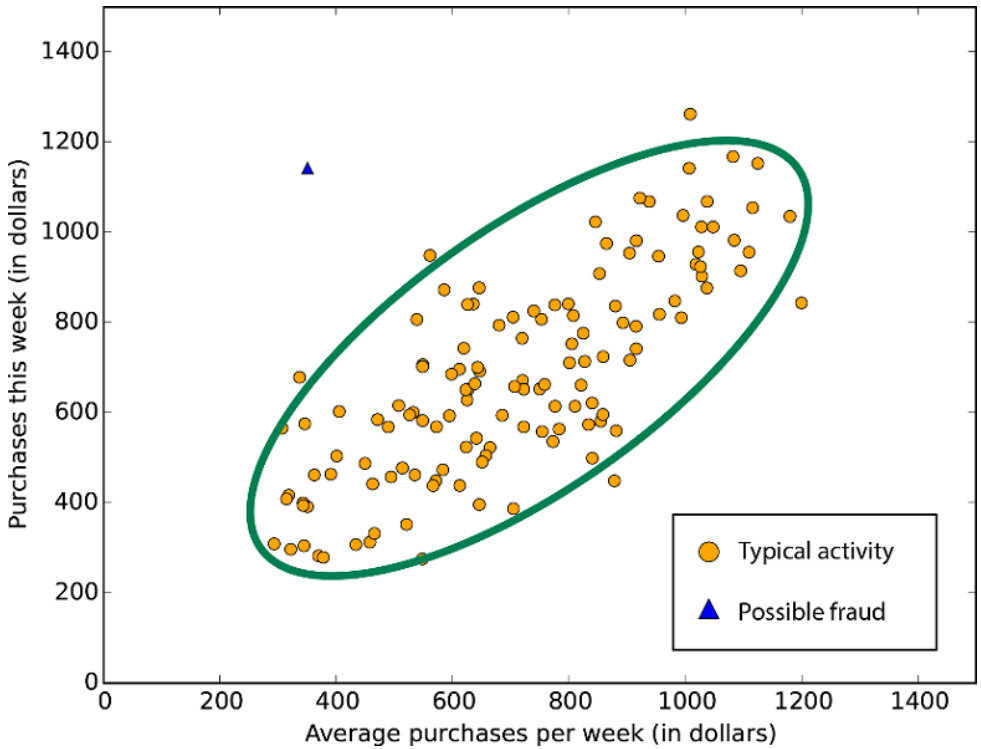


Credit: *http://www.tatvic.com/blog/wp-content/uploads/2017/01/fetured.jpg*

techopedia.com

Anomaly detection is the identification of data points, items, observations or events that do not conform to the expected pattern of a given group. These anomalies occur very infrequently but may signify a large and significant threat such as cyber intrusions or fraud.

Toy example - Credit card fraud detection



Credit: *https://docs.microsoft.com/en-us/azure/machine-learning/media/machine-learning-algorithm-choice/image8.png*

Figure: Example of data representation in a feature space for a credit card fraud problem

Motivation

In network security, the anomaly detection operates on data that are:

- Large
- High dimensional
- Unevenly distributed
- Noisy and corrupted

Prior art

1 Anomaly detection

2 Prior art

3 DANNMAD

4 Evaluation and benchmarks

5 Experimental evaluation

6 Outlook and conclusion

Prior art

- Autoencoders (neural networks)
- Density-based techniques (k-nearest neighbor, local outlier factor,...)
- One class support vector machines
- Subspace and correlation-based outlier detection for high-dimensional data.
- Isolation Forest
- Ensemble Gaussian Mixture Model

One class kNN

- Set of regular observations $\mathbf{X} = \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, ..., \mathbf{x}^{(n)}\}$
- $\mathbf{t}$ – tested sample, k – parameter of the method

1. Find k nearest neighbors of $\mathbf{t}$ from $\mathbf{X}$
2. Obtain the anomaly score as a mean distance to these neighbors

Autoencoder

'The more regularities there are in the data the more it can be compressed. The more we are able to compress the data the more we have learned about the data.' (Peter Grünwald, 1998)

Autoencoder

'The more regularities there are in the data the more it can be compressed. The more we are able to compress the data the more we have learned about the data.' (Peter Grünwald, 1998)

1. the input vector $\mathbf{x} \in \mathbb{R}^d$ is encoded to $\mathbf{y} \in \mathbb{R}^{d'}$
2. $\mathbf{y}$ is decoded to $\mathbf{x}' \in \mathbb{R}^d$.

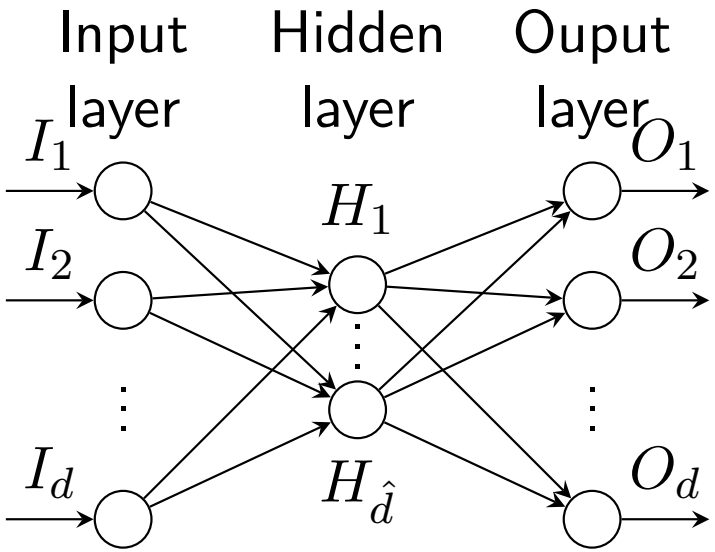


Figure: Structure of the autoencoder

Autoencoder

1. the input vector $\mathbf{x} \in \mathbb{R}^d$ is encoded to $\mathbf{y} \in \mathbb{R}^{d'}$

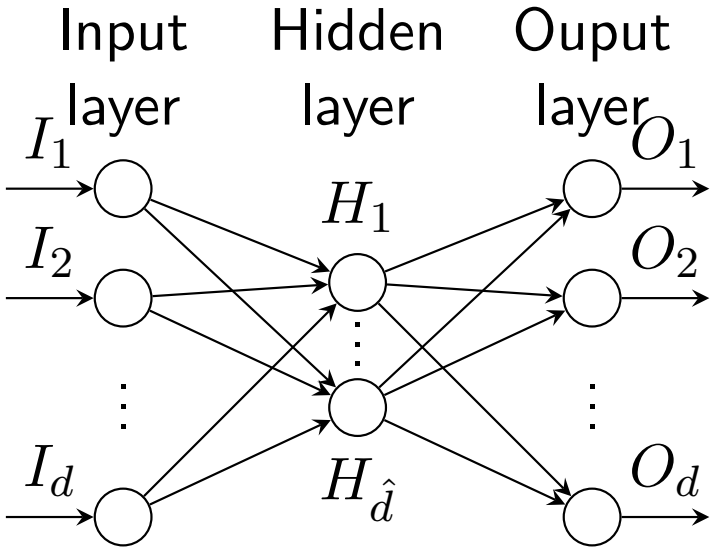


Figure: Structure of the autoencoder

The encoding is performed as:

$$\mathbf{y} = f_\theta(\mathbf{x}) = a(\mathbf{W}\mathbf{x} + \mathbf{b})$$

where f is parameterized by $\theta = \{\mathbf{W}, \mathbf{b}\}$, a is an activation function, $\mathbf{W}$ is a $d' \times d$ weight matrix and $\mathbf{b}$ is a bias vector.

Autoencoder - training

Training set $\mathbf{X} = \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, ..., \mathbf{x}^{(n)}\}$, projections $\mathbf{x}'^{(i)}$
Reconstruction error is minimized:

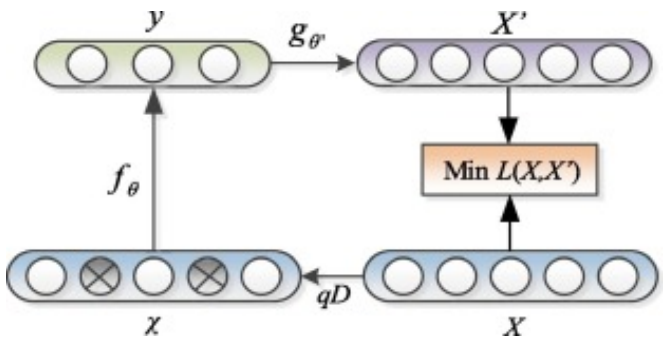
$$\theta^*, \theta'^* = \arg\min_{\theta', \theta} \frac{1}{n} \sum_{i=1}^{n} L\left(\mathbf{x}^{(i)}, \mathbf{x}'^{(i)}\right) =$$

$$\arg\min_{\theta', \theta} \frac{1}{n} \sum_{i=1}^{n} L\left(\mathbf{x}^{(i)}, g_{\theta'}\left(f_{\theta}(\mathbf{x}^{(i)})\right)\right)$$

where L represents a loss function. For example:
$L(x, x') = ||x - x'||^2$.

Denoising autoencoder

- Noise robust
- Samples are noised for each training iteration

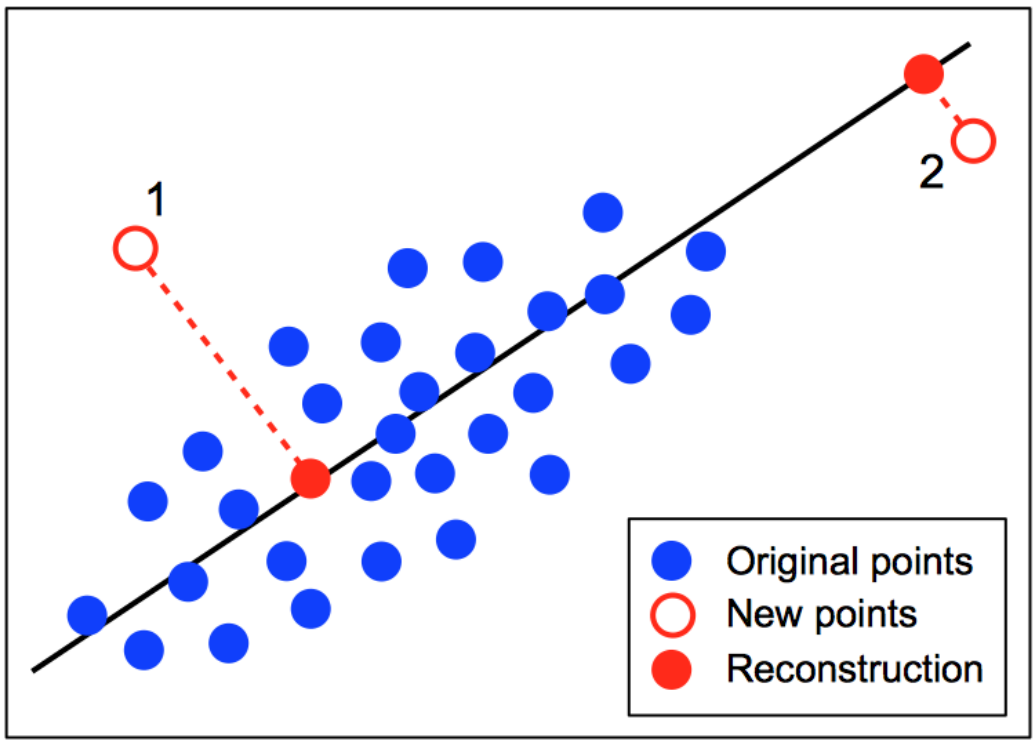


Credit: *sciencedirect.com/science/article/pii/S0263224116300641*

Figure: Salt and pepper noise

Principal component analysis

- Dimensionality reduction
- Frequently compared with autoencoders
- Similar principle



Credit: *https://i.stack.imgur.com/1j5X1.png*

Figure: PCA

Nearest Neighbor Based Models in Anomaly Detection

Campos et al. (2016): No classical anomaly detection algorithm provides a comprehensive improvement over kNN. [1]

[1] Campos, Guilherme O., et al. "On the evaluation of unsupervised outlier detection: measures, datasets, and an empirical study." Data Mining and Knowledge Discovery 30.4 (2016): 891-927.

Nearest Neighbor Based Models in Anomaly Detection

Campos et al. (2016): No classical anomaly detection algorithm provides a comprehensive improvement over kNN. [1]

- **Nearest neighbor based models**
  - + Accuracy
  - – Computational complexity

[1] Campos, Guilherme O., et al. "On the evaluation of unsupervised outlier detection: measures, datasets, and an empirical study." Data Mining and Knowledge Discovery 30.4 (2016): 891-927.

Nearest Neighbor Based Models in Anomaly Detection

Campos et al. (2016): No classical anomaly detection algorithm provides a comprehensive improvement over kNN. [1]

- **Nearest neighbor based models**
  - + Accuracy
  - – Computational complexity
- **Neural based models (auto-encoders)**
  - + Computational complexity
  - ? Accuracy

[1] Campos, Guilherme O., et al. "On the evaluation of unsupervised outlier detection: measures, datasets, and an empirical study." Data Mining and Knowledge Discovery 30.4 (2016): 891-927.

Nearest Neighbor Based Models in Anomaly Detection

Campos et al. (2016): No classical anomaly detection algorithm provides a comprehensive improvement over kNN. [1]

- **Nearest neighbor based models**
  - + Accuracy
  - – Computational complexity
- **Neural based models (auto-encoders)**
  - + Computational complexity
  - ? Accuracy
- **Density-approximating neural network models for anomaly detection**
  - + Computational complexity
  - + Accuracy

[1] Campos, Guilherme O., et al. "On the evaluation of unsupervised outlier detection: measures, datasets, and an empirical study." Data Mining and Knowledge Discovery 30.4 (2016): 891-927.

Density-Approximating Neural Network Models for Anomaly Detection (DANNMAD) [2]

- Neural network anomaly detector with anomaly score as output
  - Accurate as kNN (density imposed by kNN)
  - Fast (NN)
- Trained in two logical steps
  1. Auxiliary set is constructed and its corresponding scores are computed.
  2. The neural network is trained

[2] M. Flusser, T. Pevný, and P. Somol. Density-approximating neural network models for anomaly detection. ACM SIGKDD workshop on outlier detection de-constructed (8 2018). London, United Kingdom.

Computing the auxiliary data set

The auxiliary set $\mathbf{A} = \{\mathbf{a}_1, ..., \mathbf{a}_m\}$ is computed from the training set $\mathbf{X} = \{\mathbf{x}_1, ..., \mathbf{x}_n\}$:

1. Bounding hyper-block of $\mathbf{X}$ is observed
2. The hyper-block is filled with randomly generated and uniformly distributed samples $\{\mathbf{a}_1, ..., \mathbf{a}_m\}$
3. For each $\mathbf{a}_i$, the score $\mathbf{y}_i$ is computed as k-Nearest Neighbor mean distance $G(\cdot)$

$$y_i = G(\mathbf{a}_i) = \frac{1}{k} \sum_{j=1}^{k} D_j(\mathbf{a}_i)$$

where $D_j(\mathbf{a}_i)$ represents the j-th smallest distance of $\mathbf{a}_i$ to samples from $\mathbf{X}$.

Structure of the NN

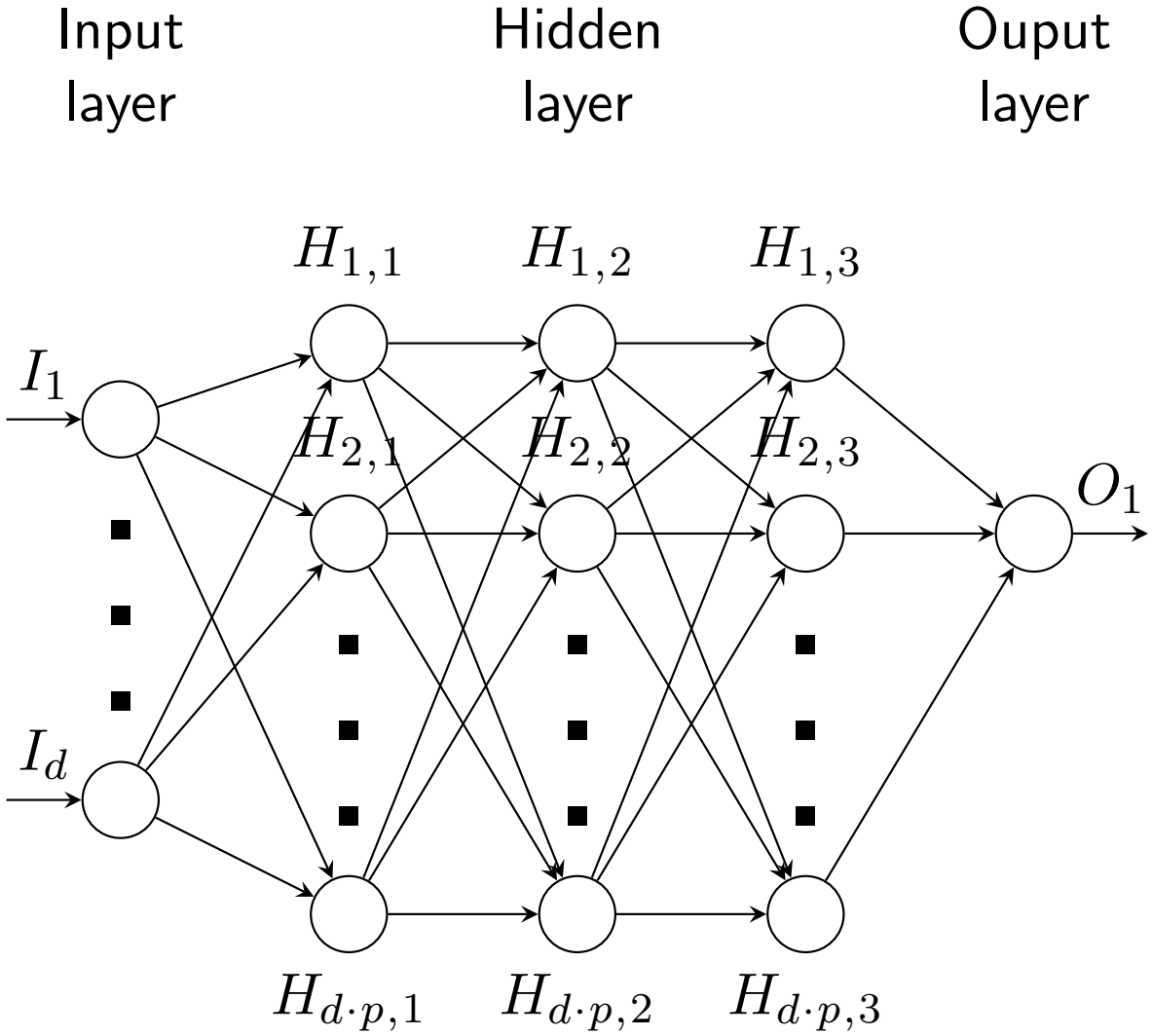


Figure: Structure of density-approximating neural network

Definition

Input vector $\mathbf{a}_i \in \mathbb{R}^d$ is projected to $y'_i \in \mathbb{R}$ as:

$$y'_i = f_\theta(\mathbf{a}_i) = f^{(4)}_{\theta^{(4)}}(f^{(3)}_{\theta^{(3)}}(f^{(2)}_{\theta^{(2)}}(f^{(1)}_{\theta^{(1)}}(\mathbf{a}_i))))$$

where $f^{(j)}_{\theta^{(j)}}$ represents the j-th layer:

$$f^{(j)}_{\theta^{(j)}}(\mathbf{a}_i) = c(\mathbf{W}^{(j)}\mathbf{a_i} + \mathbf{b}^{(j)})$$

thus $f^{(j)}$ is parameterized by $\theta^{(j)} = \{\mathbf{W}^{(j)}, \mathbf{b}^{(j)}\}$, c is an activation function, $\mathbf{W^{(j)}}$ is a weight matrix and $\mathbf{b^{(j)}}$ is a bias vector of the j-th layer. The parameters of the model are optimized as follows:

$$\theta^* = \arg\min_\theta \frac{1}{m}\sum_{i=1}^{m} L\Big(y_i, y'_i\Big)$$

Comparison of anomaly score heatmaps

Left: obtained by kNN. Right: obtained by DANNMAD

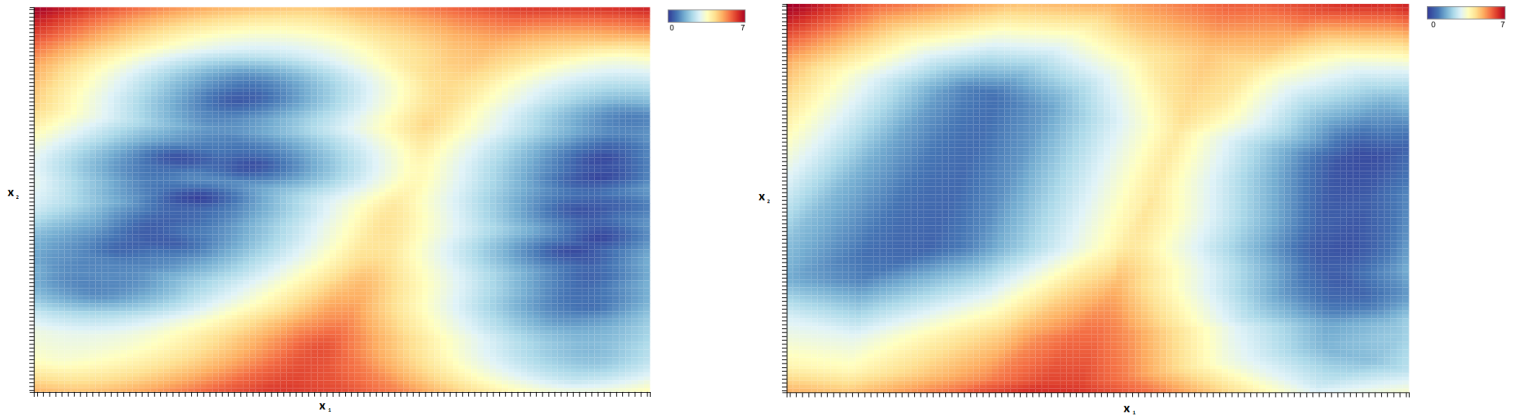


Figure: *Iris* data set

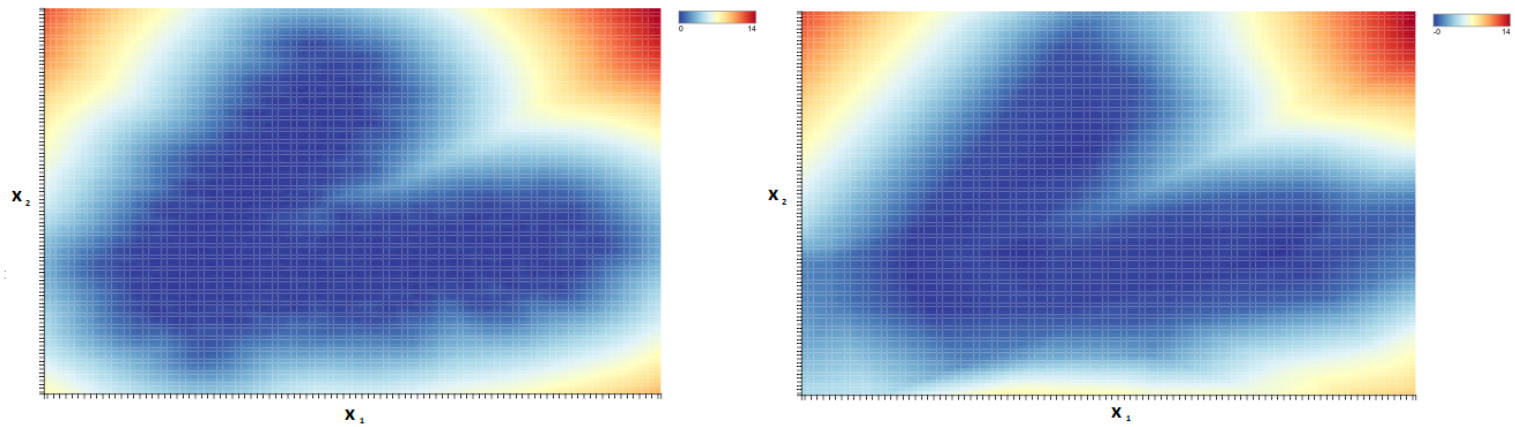


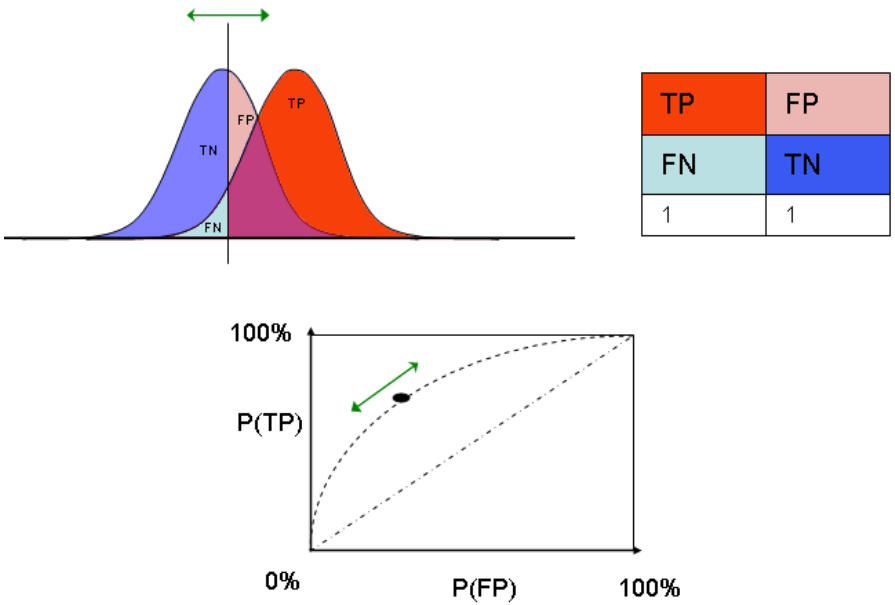
Figure: *Waveform* data set

Evaluation and benchmarks

1. Anomaly detection
2. Prior art
3. DANNMAD
4. Evaluation and benchmarks
5. Experimental evaluation
6. Outlook and conclusion

Thresholding

- Various applications require different thresholds
- ROC evaluates performance for all thresholds



Credit: *https://en.wikipedia.org/wiki/Receiver_operating_characteristic*

Figure: Receiver operating characteristic

Evaluation criteria

- Receiver operating characteristics (area under curve)
- Precision-Recall (area under curve)
- F-score
- Others...

Benchmark

- Number of different sets makes comparison hard
- Frequently used sets are obsolete:
  - KDD-99
  - MNIST
  - 99 DARPA IDEVAL
- 2013 Emmott: Systematic Construction of Anomaly Detection Benchmarks from Real Data
  - Data set consisting of many various sets
  - Tested performance of 6 popular methods
  - Neural networks were not included

Evaluation over multiple data sets

Existing methods:

- Averages over the data sets
- Pairwise Wilcoxon signed-ranks test (or t-test)
  - one vs. others
  - each vs. each
- Counts of wins/ties/losses
- Counts of significant wins/ties/losses
- Friedman score (and test)

Issues connected with usage multiple data sets

There is no general method to find out optimal setting of

- Structure of the neural network (size of the bottleneck for AE, nr. of layers and neurons,...)
- Type and intensity of noise for AE
- Nr. of neurons for DANNMAD

The prior works on auto-encoders usually evaluate on few data sets and tune parameters empirically

Experimental evaluation

1. Anomaly detection
2. Prior art
3. DANNMAD
4. Evaluation and benchmarks
5. Experimental evaluation
6. Outlook and conclusion

Experiment

The aim is to compare the performance of the anomaly detection methods below with the most advanced methodology.

- Accuracy:
  - DANNMAD vs kNN
  - DANNMAD vs Autoencoder
- Computational complexity of DANNMAD vs kNN with respect to:
  - Number of samples
  - Dimension
- Metric: AUC ROC

Experiment - Data

Most advanced Emmott's methodology (2013)

- 18 various real-data sets
- 4 levels of anomalies
- 64 data sets used

Experiment - Parameters

- kNN
  - Supporting structures: kd-tree, ball-tree
  - k=5 (empirically)
- AE
  - Denoising (4 levels of Gaussian noise)
  - Three layers
  - 6 setups for bottleneck
  - 24 various models trained for each set
- DANNMAD
  - kNN k=5
  - Number of hidden layers: 3 and 2
  - Hidden layer size: 3d and 5d
  - 4 models trained for each set

Accuracy results: DANNMAD vs kNN

Table: Counts of wins of *DANNMAD* versus k*NN*, grouped by problem difficulty. Wilcoxon signed rank test at 0.05 level is used to verify statistical significance of wins for each level

	Easy	Medium	Hard	V. Hard	Sum
DANNMAD	10	8	9	9	36
kNN	8	10	7	3	28
Significance	no	no	no	no	–

Accuracy results: DANNMAD vs Auto-encoder

Table: Counts of wins of the *DANNMAD* versus *auto-encoder*, grouped by problem difficulty. Wilcoxon signed rank test at 0.05 level is used to verify statistical significance of wins for each level

	Easy	Medium	Hard	V. Hard	Sum
DANNMAD	12	13	9	9	43
Auto-encoder	6	5	7	3	21
Significance	no	yes	no	yes	–

Time complexity - dimensionality

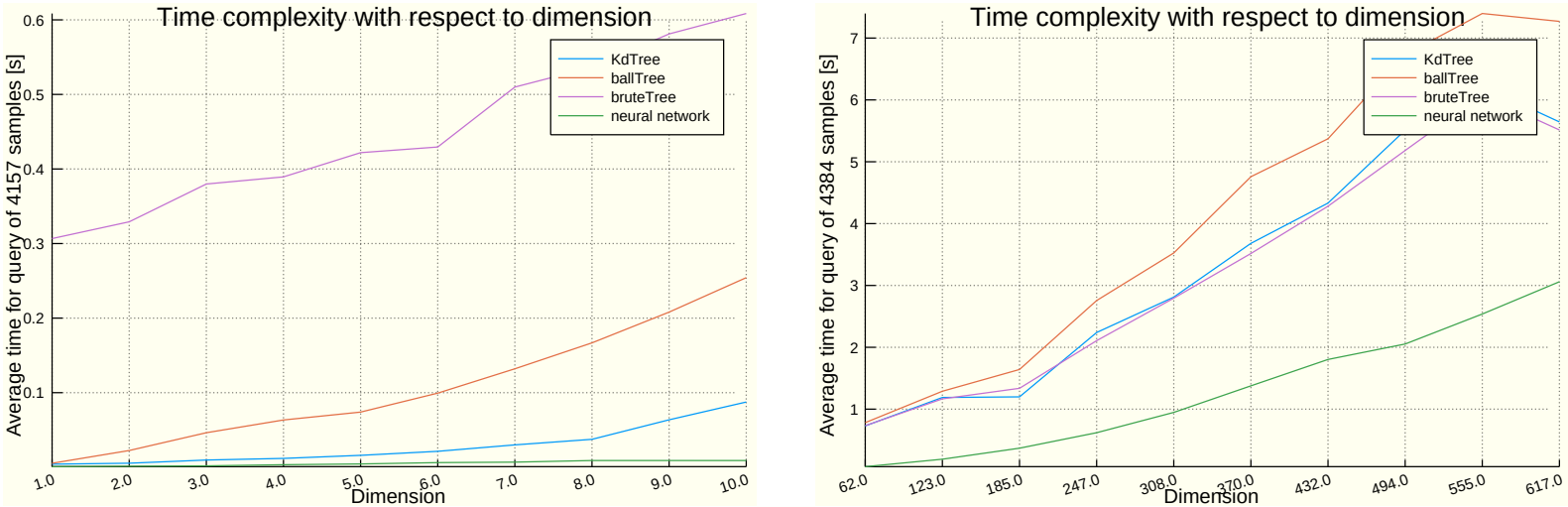


Figure: Anomaly detectors' prediction time dependence on dimensionality in application phase. Tested on *magic telescope* and *Isolet* data sets

Time complexity - nr. of samples

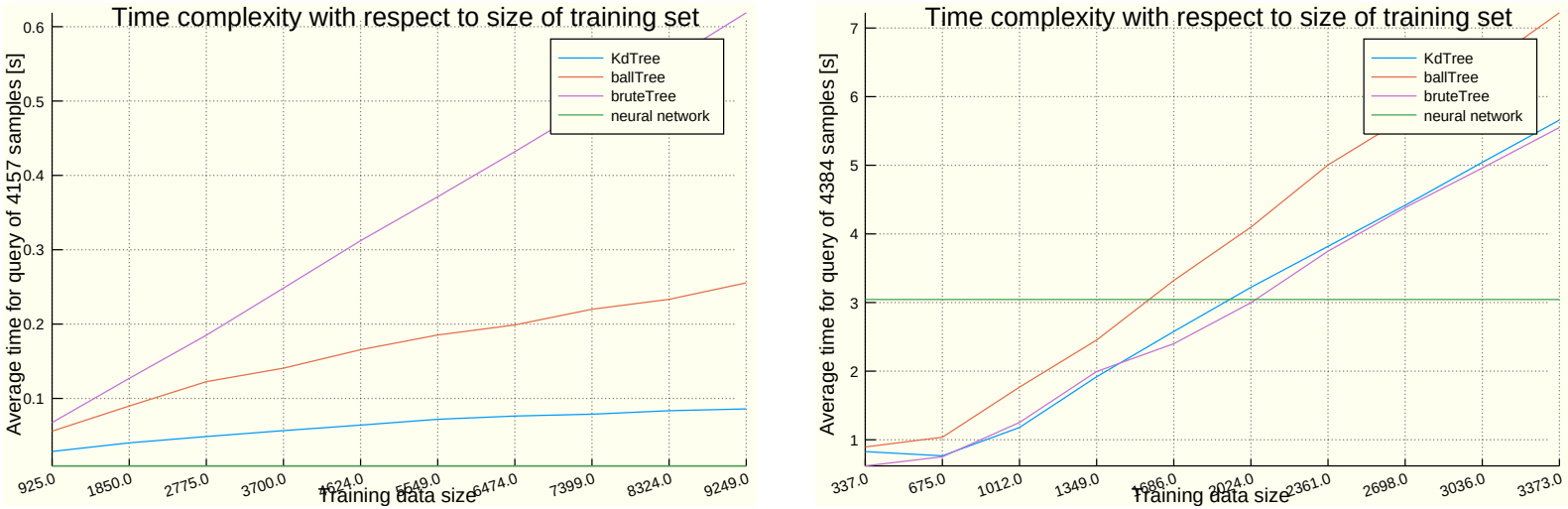


Figure: Anomaly detectors' prediction time dependence on training data size in application phase. Tested on *magic telescope* and *Isolet* data sets. Neural model prediction speed does not depend on training data size (note the close-to-zero time in *magic telescope* case)

Discussion

- Accuracy comparable to kNN based models
- Computational complexity lower by orders of magnitude in comparison to kNN
- Outperforming auto-encoders often
- Validated with the most advanced methodology for anomaly detection

- Beneficial especially for
  - Industry (large-scale data)
  - Embedded systems (low memory and computational demands)

Outlook and conclusion

1 Anomaly detection

2 Prior art

3 DANNMAD

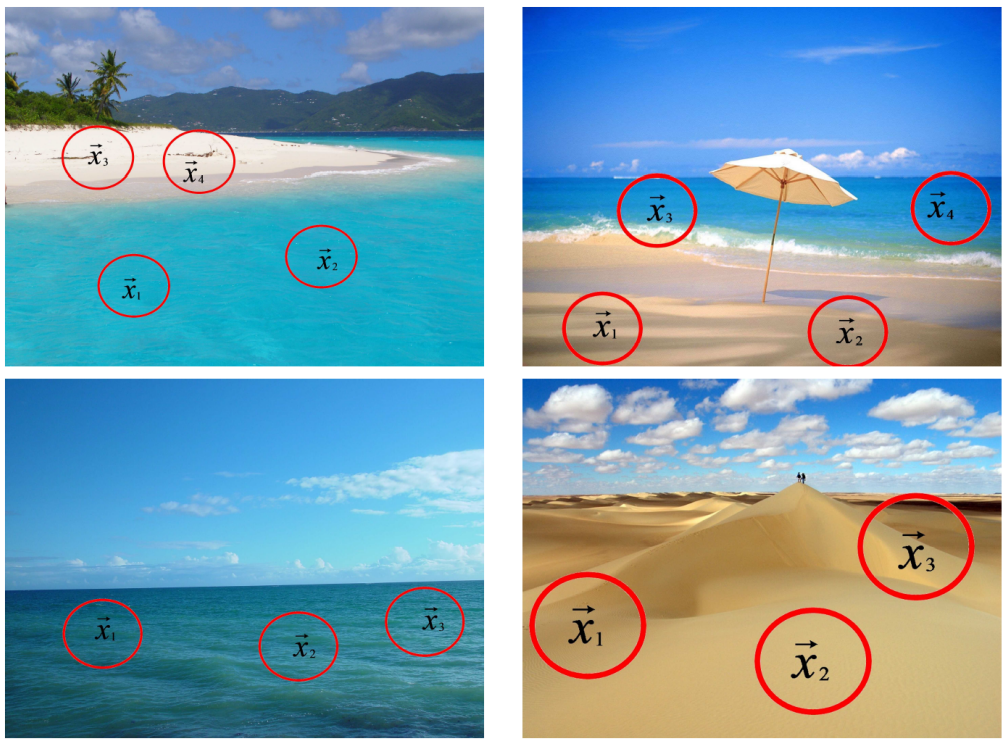
4 Evaluation and benchmarks

5 Experimental evaluation

6 Outlook and conclusion

Outlook - Multiple Instance Learning

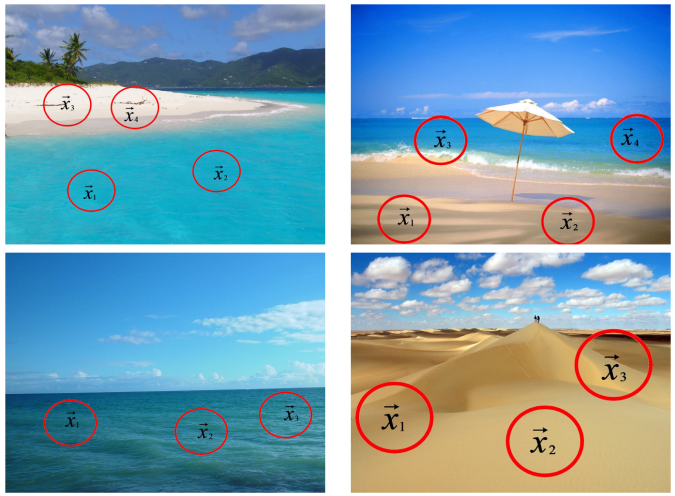
- Many real-world problems have structured representation
- Operates on (labeled) bags
- Bag $\mathbf{X}$ consists of vectors $\{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N\}$ where $\mathbf{x}_i \in \mathbb{R}^d$



Credit: *http://158.109.8.37/files/Amo2013.pdf*

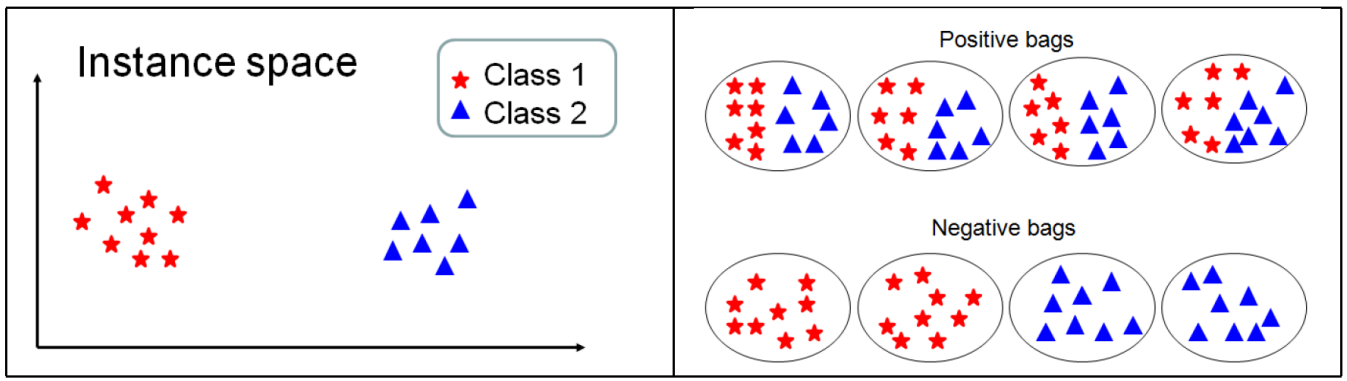
Figure: Example of MIL

Outlook - Multiple Instance Learning



Credit: *http://158.109.8.37/files/Amo2013.pdf*

Figure: Example of MIL



Credit: *http://158.109.8.37/files/Amo2013.pdf*

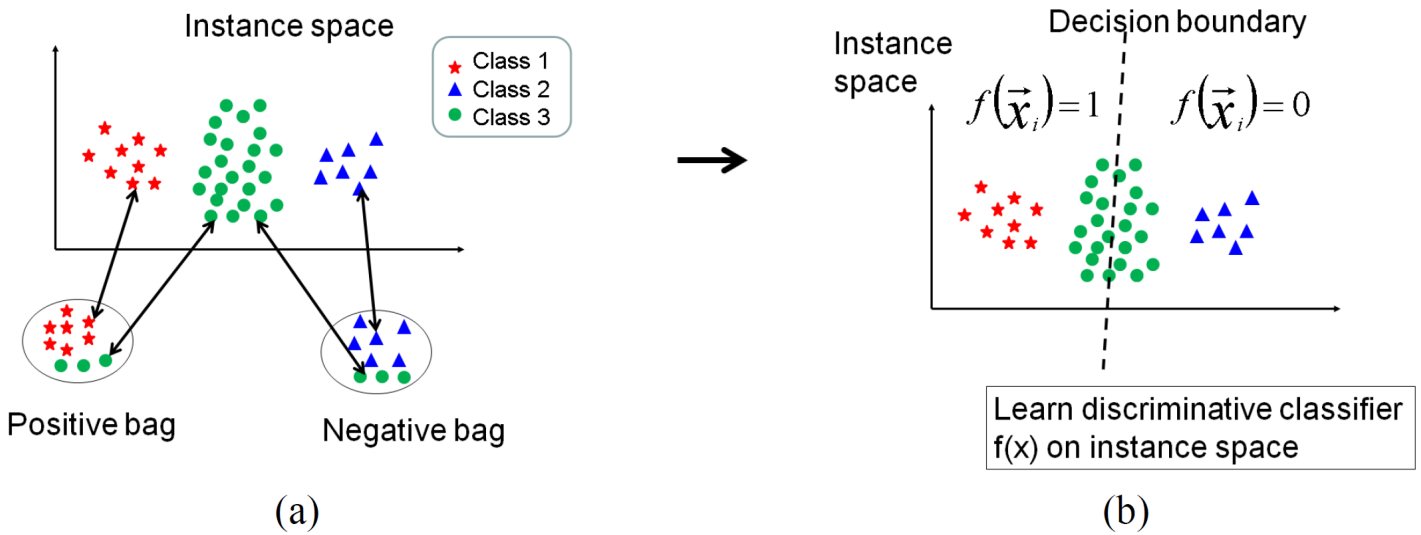
Figure: Instance space vs bag space

MIL - paradigms

1. Instance space paradigm
2. Embedded space paradigm
3. Bag space paradigm

Instance space paradigm

- $F(\mathbf{X}) = \frac{1}{|X|} \sum_{\mathbf{x} \in \mathbf{X}} f(\mathbf{x})$
- $F(\mathbf{X}) = \max_{\mathbf{x} \in \mathbf{X}} f(\mathbf{x})$

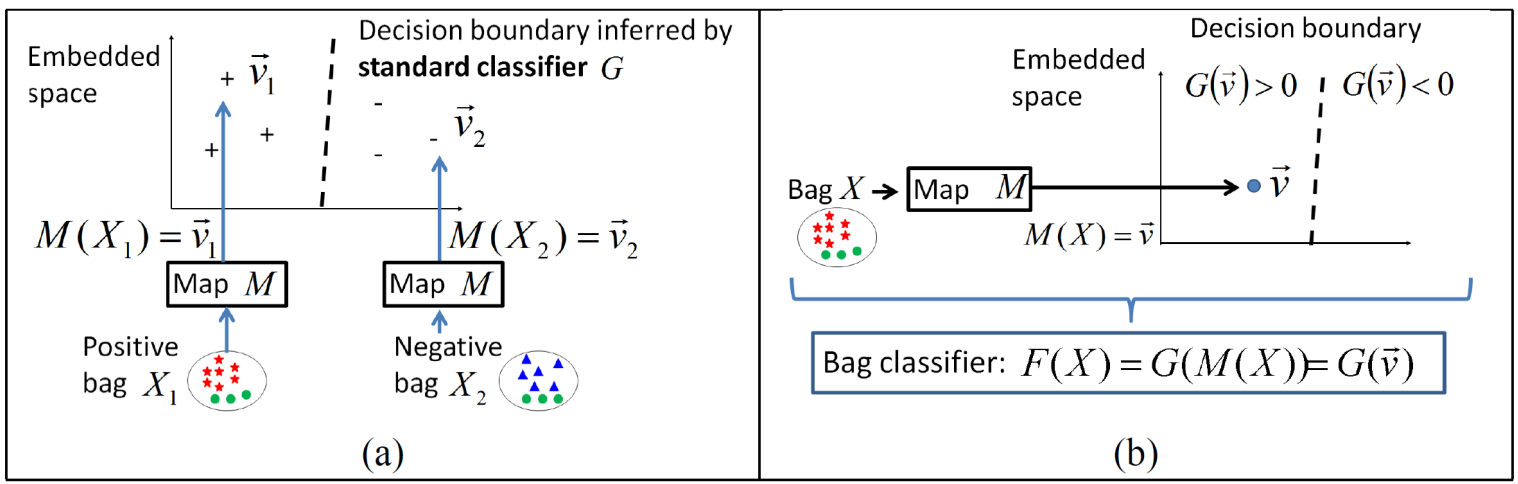


Credit: *http://158.109.8.37/files/Amo2013.pdf*

Figure: Instance space paradigm

Embedded space paradigm

- "Simple MI" $\mathcal{M}(\mathbf{X}) = \frac{1}{|\mathbf{X}|} \sum_{\mathbf{x} \in \mathbf{X}} \mathbf{x}$
- Min-max vector $\mathcal{M}(\mathbf{X}) = (a_1, a_2, ..., a_d, b_1, b_2, ..., b_d)$ where $a_j = \min_{\mathbf{x} \in \mathbf{X}} x_j$, for $j = 1, ..., d$
- Vocabulary methods

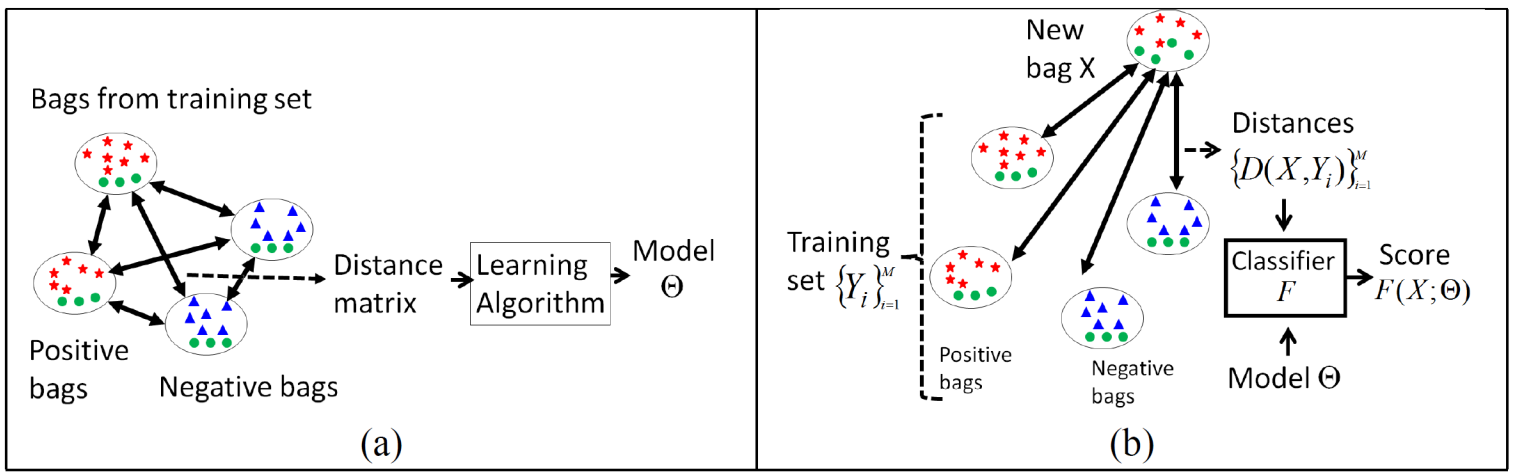


Credit: *http://158.109.8.37/files/Amo2013.pdf*

Figure: ES paradigm, training (a), test(b)

Bag space paradigm

- Defines pairwise comparison on bags
- Minimal Hausdorff distance: $D(\mathbf{X}, \mathbf{Y}) = \min_{\mathbf{x}\in\mathbf{X},\mathbf{y}\in\mathbf{Y}} ||\mathbf{x} - \mathbf{y}||$



Credit: *http://158.109.8.37/files/Amo2013.pdf*

Figure: BS paradigm, training (a), test(b)

Outlook

- kNN is a direct way of utilizing BS paradigm
  - For AD only a few metrics has been utilized yet
  - Too high complexity to be utilized in practice
- We plan to utilize DANNMAD base method for MIL AD
  - DANNMAD is defined for vector representation only
  - How to construct auxiliary set?
- Alternatively observe conditions for utilizing ES paradigm

Outlook

- Only about 10 relevant papers of AD for MIL
- Evaluation not standardized yet

Conclusion

- DANNMAD
  - Reducing complexity of k-NN
  - Often outperforming auto-encoders
- Multiple instance learning
  - Direct approach is k-NN

Thank you

Questions?

Credit: *http://www.incimages.com/uploaded_files/image/1940x900/rotten-apple-1725x810_12112.jpg*