Anomaly Detection

Yanbing Xue

Agenda

- Introduction
- Classification-based
- Nearest Neighbor-based
- Cluster-based
- Statistical
- Information Theoretic and Spectral
- Contextual and Collective
- Conclusion

Agenda

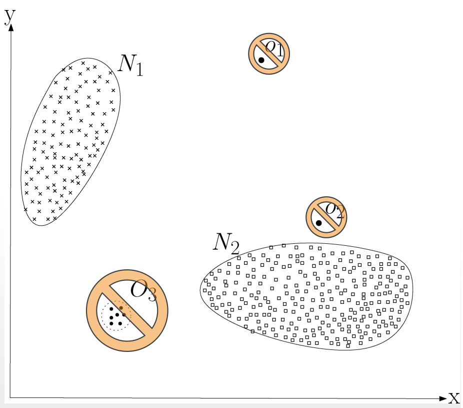
- ***Introduction***
- Classification-based
- Nearest Neighbor-based
- Cluster-based
- Statistical
- Information Theoretic and Spectral
- Contextual and Collective
- Conclusion

Definition

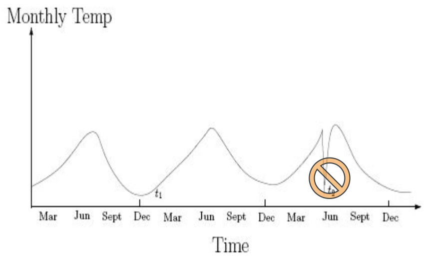
- Not conform to expected patterns or rest of data sets.
- vs Noise?

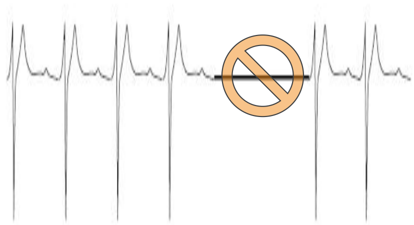
  Does it always produce Anomalous outputs?

  Do we care about them?



Types

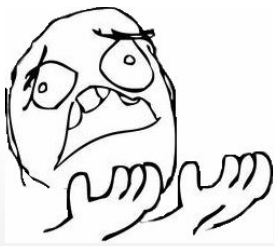




- Point Anomalies
- Contextual Anomalies
- Collective Anomalies

Challenges

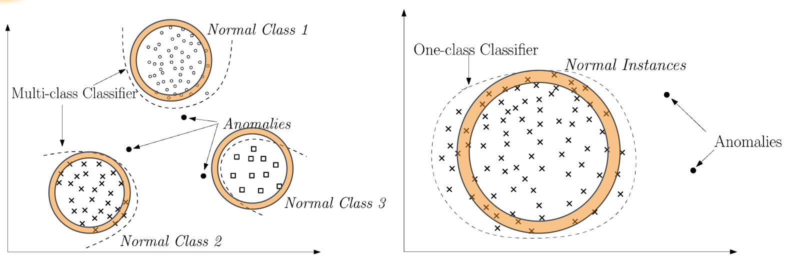
- Labels usually unavailable
  Semi-supervised: only labels of normal instances available
  Unsupervised: No labels, assuming anomalies are very rare
- How to distinguish normal entries from anomalies
  Criterion covering all normal situations;
  Definition of "normal" changes over time
- Not remarkable
- Hard for exact notion
- Noise contamination



Agenda

- Introduction
- ***Classification-based***
- Nearest Neighbor-based
- Cluster-based
- Statistical
- Information Theoretic and Spectral
- Contextual and Collective
- Conclusion

Classification-based



(a) Multi-class Anomaly Detection (b) One-class Anomaly Detection

- Assumption

  A classifier that can distinguish between normal and anomalous classes can be learned in given feature space

Classification-based (Cont'd)

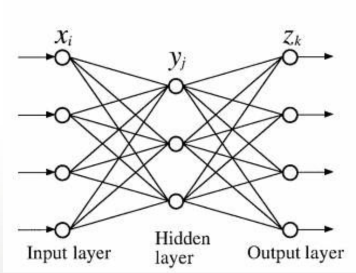
- **Multi-class classification**
  - Anomalies are not classified by any of the classifiers
- **One-class classification**
  - A discriminative boundary around normal entries and anomalies
- **Supervised**
  - Require knowledge of both normal and anomaly classes
  - Build classifier to distinguish between normal and known anomalies
  - Not interesting, similar with traditional classifications
- **Semi-supervised**
  - Require knowledge of normal classes only
  - Use modified classification model to learn normal behaviors and then detect any deviations from normal behaviors as anomalous

Neural Network-based

- **Multi-class Classification**
  - Train a neural network on normal instances for normal classes;
  - Normal instances have labels of normal classes in training set;
  - *Normal*: if accepted by the neural network as any of the normal classes;
  - *Anomalous*: if rejected by the neural network;
- **One-class Classification**
- **Replicator Neural Network**
  - Semi-supervised
  - A multi-layer feed-forward neural network
  - Assumption: Lower dimensional space captures patterns of normal instances w/ little loss

Neural Network-based (Cont'd)

- $N_{input} = N_{output}$
- $N_{hidden} < N_{input}$
  $N_{hidden} < N_{output}$
- Input $\mathbf{x}_i$, output $\mathbf{o}_i$
- Reconstruction Error
  Also as anomaly score



$$\delta_i = \frac{1}{n} \sum_{j=1}^{n} (x_{ij} - o_{ij})^2$$

- RNN vs SVD?

Bayesian Network-based

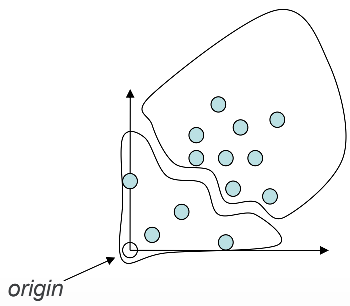
- For multi-class anomaly detection
- Semi-supervised
- Uni-variate settings
  Class label w/ highest posterior chosen as predicted class
  Likelihood and prior learned from training set
- Multi-variate settings
  Aggregation of posteriors of each attribute
  Complex Bayesian networks for conditional dependencies

Support Vector Machine-based

- For one-class anomaly detection
  - One-class support vector machine (OC-SVM)
  - Assuming all training instances have only one normal class label
- Use kernels for complex regions
  - Usually radial basis function (RBF)
- Normal: if falls within the learned region
- Anomalous: if falls outside the learned region

Support Vector Machine-based (Cont'd)

- Separate training data from origin
  - Find a small region where most instances lies and label these instances as one class
  - Separate regions containing instances from regions containing none
  - Push boundary away from origin as much as possible
- Schölkopf's implementation



$$\min_{\mathbf{w},\xi_i,\rho} \frac{1}{2}\mathbf{w}^T\mathbf{w} + \frac{1}{\nu n}\sum_{i=1}^{n}\xi_i - \rho$$

subject to :

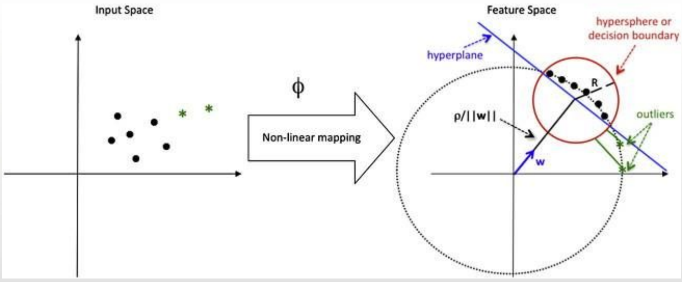
$$\mathbf{w}\phi(\mathbf{x}_i) \geq \rho - \xi_i$$

$$\xi_i \geq 0$$

$$f(\mathbf{x}) = sign(\sum_{i=1}^{n}\alpha_i\langle\mathbf{x}_i,\mathbf{x}\rangle - \rho)$$

Support Vector Machine-based (Cont'd)

- Two implementations in kernel space
- Hyperplane between normal and anomalous
- Smallest hypersphere containing all normal



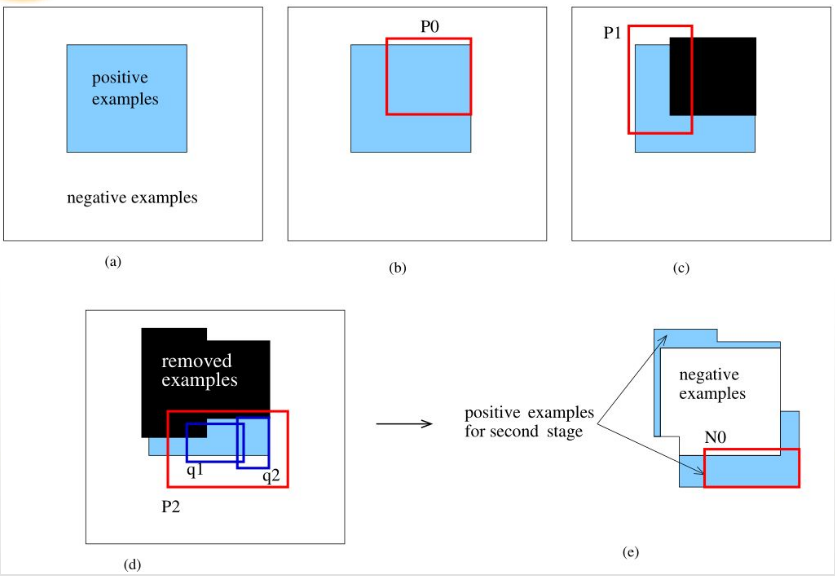
Rule-based

- For multi-class anomaly detection
- Rule learning algorithm (RIPPER, decision tree, concept learning)
  - Confidence ∝ precision rate
- Find rule best capturing the data entry
  - Anomaly score = inverse of confidence
- For one-class anomaly detection
- Association rule mining
  - Support threshold for pruning

PN Rule Learning

- For multi-class anomaly detection
- P-phase
  - Cover most of the positive examples w/ high support
  - Seek good recall
- N-phase:
  - Remove false positive instances covered in P-phase
  - N-rules give high accuracy and significant support

PN Rule Learning (Cont'd)



Association Rule Mining

- For one-class anomaly detection
- $I = \{i_1, i_2, \ldots, i_n\}$

  Items: a set of *n* binary attributes
- $D = \{t_1, t_2, \ldots, t_m\}$

  Database: a set of *m* transactions containing a subset of *I*
- Rules: $X \Rightarrow Y$

  X, Y are subsets of I and $X \cap Y = \Phi$

Apriori Algorithm

- Steps

  Set threshold *p*, subsets w/ frequency no less than *p* are frequent

  Scan for frequent 1-size subset

  $k = 1$

  Repeat
  - *k*++
  - Scan frequent *k*-size subsets based on frequent *k*-1-size subsets

  Until
  - there is no frequent *k*-size subset

Apriori Algorithm (Cont'd)

- Database

  $t_1 = \{i_1, i_3, i_4\}$

  $t_2 = \{i_2, i_3, i_5\}$

  $t_3 = \{i_1, i_2, i_3, i_5\}$

  $t_4 = \{i_2, i_5\}$

- $k = 1$, $p = 2$

  $\{i_1\} = 2$

  $\{i_2\} = 3$

  $\{i_3\} = 3$

  $\{i_4\} = 1$

  $\{i_5\} = 3$

- $k = 2$, $p = 2$

  $\{i_1, i_2\} = 1$

  $\{i_1, i_3\} = 2$

  $\{i_1, i_5\} = 1$

  $\{i_2, i_3\} = 2$

  $\{i_2, i_5\} = 3$

  $\{i_3, i_5\} = 2$

- $k = 3$, $p = 2$

  $\{i_2, i_3, i_5\} = 2$

- Apriori stops

Classification-based (Summary)

- Training Complexity

  It depends

  Decision tree is usually fast $O(n\log n)$

  Support vector machine is usually expensive $O(n^3)$

- Testing Complexity

  Usually very fast

- Cons

  × Rely on accurate labels for normal classes

  × Assign a label to each test instance

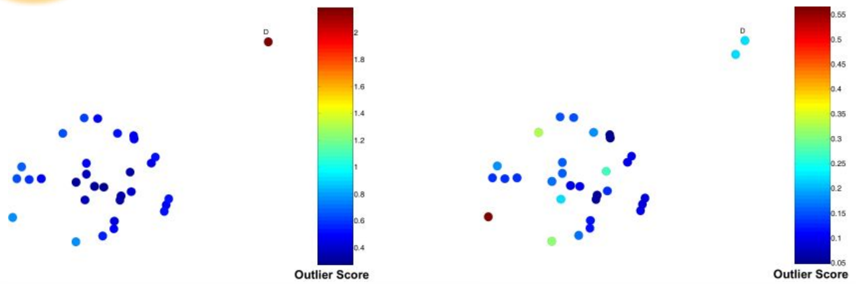
Agenda

- Introduction
- Classification-based
- ***Nearest Neighbor-based***
- Cluster-based
- Statistical
- Information Theoretic and Spectral
- Contextual and Collective
- Conclusion

Nearest Neighbor-based

- Assumptions
  - Normal ∽ dense neighborhoods
  - Anomalous ∽ far from closest neighbors
- Basic distance measurement
  - Continuous ∽ Euclidean
  - Categorical ∽ Matching coefficient
  - Multivariate ∽ Attribute combination
- Complex distance measurement
  - Positive-definite
  - Symmetric

K^{th} Nearest Neighbor Distance



- Basic Idea
- Anomaly score = k^{th} nearest neighbor distance

  Is k = 1 a good idea? Why?

K^{th} Nearest Neighbor Distance (Cont'd)

- Alternative implementations
- Different criteria

  Set a threshold between normal entries and anomalies

  Select a certain number of anomalies w/ highest anomaly scores
- Different measurements

  Sum of k nearest neighbor distance

  Number of neighbors less than a given distance

  Hypergraph connectivity

  Combination of matching coefficient and covariance matrix

K^{th} Nearest Neighbor Distance (Cont'd)

- Complexity

  Expensive $O(n^2)$

- Different complexity improvements

  Set threshold as anomaly score of weakest anomaly to a given entry;

  Drop clusters not possibly containing top *k* anomalies after computing upper and lower bounds of k^{th} nearest neighbor in each cluster;

  Only compute anomaly score of a given entry w/ samples;

  Number of instances in local hypercube and adjoining hypercubes;

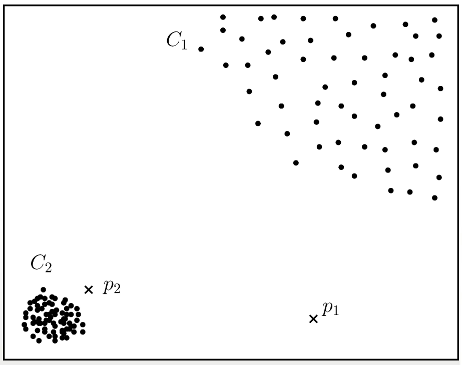
  Combinations of *k* nearest neighbor and Hilbert space filling curve

Relative Density

- Basic implementation

  Inverse of k^{th} nearest neighbor distance

- Low performance when densities vary



Relative Density (Cont'd)

- Local outlier factor (LOF)

  Ratio between average local density of *k* nearest neighbors and self local density

- Basic Ideas

  Find smallest hypersphere containing *k'* nearest neighbors

  Local density = $k' / V_{hypersphere}$

  Normal: self local density ≈ average local density of *k* nearest neighbors

  Anomalous: self local density << average local density of *k* nearest neighbors

Local Outlier Factor (LOF)

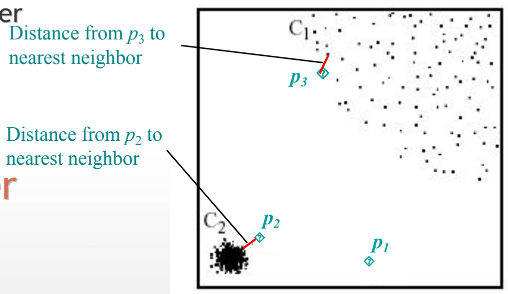
- For each instance *A* compute the distance to the k^{th} nearest neighbor *kd*(*A*)
- Get reachability distance for each instance *A* with respect to instance *B*

  $rd(A, B) = \max\{kd(B), d(A, B)\}$

- Get local reachability density of *A* based on its k' nearest neighbors $lrd(A) = \frac{k'}{\sum_{B \in NN(k',A)} rd(A,B)}$
- Compute LOF of A as $lof(A) = \frac{\sum_{B \in NN(k',A)} lrd(B)}{k' \cdot lrd(A)}$

Local Outlier Factor (Cont'd)

- LOF finds both p_1 and p_2 as outliers
  - NN may not consider p_2 as outlier
- LOF does not consider p_3 as outlier
  - NN may consider p_3 as outlier,

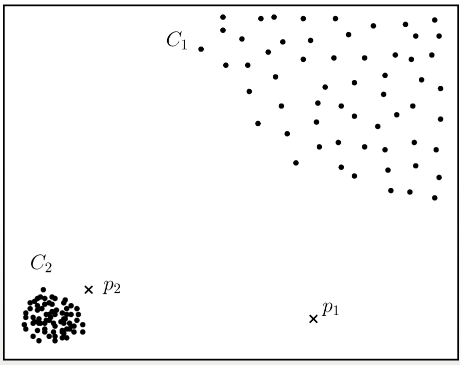


Relative Density (Cont'd)

- Connectivity-based outlier factor (COF)
  - *k* nearest neighbors for averaging determined incrementally
  - Special patterns of normal instances can be captured
- Outlier detection using in-degree number (ODIN)
  - Anomaly score = $1 \ / \ N_{\text{mutual } k \text{ nearst neighbor}}$
- Multi-Granularity Deviation Factor (MDEF)
  - Anomaly score = $1 \ / \ \sigma_{\text{nearest neighbors and self local density}}$

Relative Density (Cont'd)

- **For categorical attributes**
  Similarity measurements
- **Complexity improvements**
  Only top n anomalies after finding upper and lower bounds of LOF in each cluster



Nearest Neighbor-based (Summary)

- **Complexity**
  Expensive $O(n^2)$
- **Limitations of improvements**
  k-d trees, *R*-trees, hypergrids ∽ exponential in number of attributes
  Only keep top few anomolies ∽ what if each anomaly score is expected?
  Sampling ∽ inaccurate anomaly scores under small sample sizes

Nearest Neighbor-based (Summary)

- Pros
  - ✓ Unsupervised
  - ✓ Semi-supervised ~ higher performance in missed anomalies
  - ✓ Easy adaption to different data types
- Cons
  - × Unsupervised ~ missed anomalies
  - × Semi-supervised ~ high false positive rate
  - × High testing complexity
  - × Rely on distance measurements

Agenda

- Introduction
- Classification-based
- Nearest Neighbor-based
- ***Cluster-based***
- Statistical
- Information Theoretic and Spectral
- Contextual and Collective
- Conclusion

Cluster-based

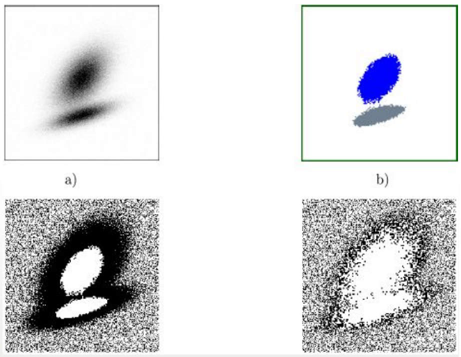
- Normal instances belong to a cluster, while anomalies do not
- Implementations
  - DBSCAN: not all instances must belong to a cluster
  - FindOut: remaining treated as anomalies after cluster removal
- Cons
  - They are essentially still clustering algorithm

FindOut

- By-product of WaveCluster
- Main idea
  - Remove clusters from original data and then identify outliers.
- Transform data into multi-dimensional signals via wavelet transformation
  - High frequency of signals are regions of rapid change of distribution, usually boundaries of clusters;
  - Low frequency parts are regions of concentrated data, usually clusters

FindOut (Cont'd)

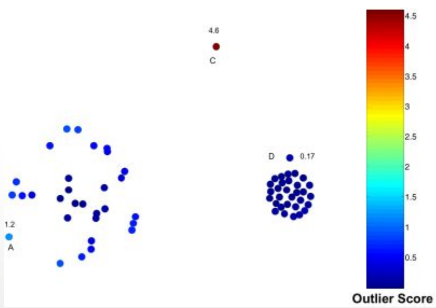
- Remove these high and low frequency parts
- All remaining instances are treated as outliers



Cluster-based (Cont'd)

- Normal instances close to nearest cluster centroids, anomalies far
- Implementations

  Anomaly score = distance to nearest cluster centroid after clustering



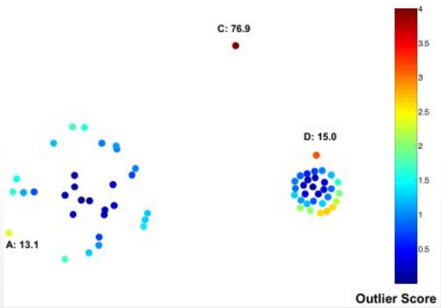
Cluster-based (Cont'd)

- Alternative implementations

  Item-set mining before clustering;

  Using relative distance compared with *k* nearest neighbors of centroid;

  Semi-supervised: semantic anomaly factor, high if different from cluster majority



Cluster-based (Cont'd)

- Normal instances belong to large and dense cluster, while anomalies belong to small or sparse ones
- Implementations

  Anomalies belong to clusters whose size or density is below threshold

- Cluster-based Local Outlier Factor (CBLOF)

  The product of size of cluster where it is and:

  Distance to centroid of nearest large cluster (if in a small cluster)

  Distance to centroid of cluster where it is (if in a large cluster)

Cluster-based Local Outlier Factor

- Determine CBLOF for each instance by size of cluster and distance to cluster centroid
- When instance in a ***small*** cluster, CBLOF is product of size of cluster where instance belongs and distance to centroid of closest larger cluster
- When instance in a ***large*** cluster, CBLOF is product of size of cluster where instance belongs and distance to centroid of cluster

Cluster-based (Summary)

- Training Complexity
  Differ from linear to quadratic
- Testing Complexity
  Fast, the number of clusters is usually small
- Pros
  ✓ Share same pros with nearest neighbor-based detection
- Cons
  × Rely on clustering algorithms (Must each instance be assigned to a cluster? Do anomalies also form clusters themselves?)
  × Essentially, many algorithms are still clustering algorithms

Agenda

- Introduction
- Classification-based
- Nearest Neighbor-based
- Cluster-based
- ***Statistical***
- Information Theoretic and Spectral
- Contextual and Collective
- Conclusion

Statistical

- Normal instances in high probability regions of stochastic models, while anomalies in low probability ones
- Fit a statical model for normal instances

  Anomalous: instances w/ low probabilities of being generated from trained models
- Parametric

  Normal instances are generated from a parametric distribution
- Non-parametric

  Models determined by given instances

Parametric Techniques

- Normal instances are generated from an underlying parametric distribution $f(\mathbf{x}|\theta)$
  - Anomaly score = inverse of its density
- Gaussian model-based
  - Box plot rule
  - Grubb's test
- Regression model-based
- Mixture of parametric distribution-based

Gaussian Model-based

- Entries are generated from normal distribution
  - Parameters obtained via MLE
  - Anomaly score: distance to estimated mean
  - Anomalous: if anomaly score is greater than threshold
- Implementation
  - Threshold = 3σ
- Alternative implementation
  - $Q_3 - Q_1 = IQR \approx 1.349\sigma$
  - $[Q_1 - 1.5IQR, Q_3 + 1.5IQR] \approx [\mu - 2.698\sigma, \mu + 2.698\sigma]$

Grubb's Test

- z-score

  Uni-variate: $z(x) = \frac{1}{\sigma} | x - \bar{x} |$

  Multi-variate: $z(\mathbf{x}) = \sqrt{(\mathbf{x} - \bar{\mathbf{x}})^T \boldsymbol{\Sigma}^{-\frac{1}{2}} (\mathbf{x} - \bar{\mathbf{x}})}$

- Hypothesis

  H_0: there is no outlier

- Reject H_0 if: $z > \frac{N-1}{\sqrt{N}} \sqrt{\frac{t^2_{\frac{\alpha}{2N}, N-2}}{N - 2 + t^2_{\frac{\alpha}{2N}, N-2}}}$

  t: threshold taken by t-distribution at significance level of α / 2N

Mixture of Parametric Distributions

- Assumptions

  Normal instances and anomalies are of separate parametric distributions

- Implementation

  Normal ∽ N(0, σ^2)

  Anomalous ∽ N(0, $k^2\sigma^2$) where $k > 1$

  Use Grubb's test on both distributions

- Alternative implementation

  D = (1 - λ)**M** + λ**A**

  Assumption: # of normal instances in data set is significantly larger than # of anomalies

Mixture of Parametric Distributions (Cont'd)

- Alternative implementation

  Expectation maximization

  D: Actual probability distribution of data set

- **D** = $(1 - \lambda)$**M** + λ**A**

  M: majority distribution, **A**: anomalous distribution

- L_t: likelihood of **D** at t^{th} iteration

  M_t: normal instance set, A_t: anomaly set

  Initial state: all instances in M_0, $A_0 = \Phi$

  Iterations: calculate $(L_t - L_{t-1})$ when $M_t = M_{t-1} - \{\mathbf{x}_t\}$, $A_t = A_{t-1} \cup \{\mathbf{x}_t\}$

  Anomalous: if $(L_t - L_{t-1})$ is high

Nonparametric Techniques

- Histogram-based

  Anomalous: if fall into empty or rare bins

  Multi-variate: attribute-wise histograms

- Kernel Function-based

  Estimate *pdf* of normal instances via kernel functions

  Anomalous: if fall into low probability areas

Statistical (Summary)

- Complexity
  - Linear per iteration (iterative techniques on exponential family)
  - Quadratic (Kernel-based techniques)
- Pros
  - ✓ Also provide confidence intervals
  - ✓ Unsupervised when using robust models
- Cons
  - × Rely on assumption that instances are generated from a given distribution
  - × Choices of anomaly criteria are not straightforward
  - × Attribute-wise techniques cannot detect attribute correlations

Agenda

- Introduction
- Classification-based
- Nearest Neighbor-based
- Cluster-based
- Statistical
- ***Information Theoretic and Spectral***
- Contextual and Collective
- Conclusion

Information Theoretic (Intro)

- **Assumption**

  Anomalies significantly alter information contents in data sets

- **Implementations**

  Detect data instances altering information contents significantly

  Kolmogorov complexity-based

  Entropy-based: find *k* instances whose removal minimize entropy

- **Pros**

  ✓ Unsupervised

  ✓ No underlying statistical distributions needed

- **Cons**

  × Rely on size of substructures and information theoretic measurements

Spectral (Intro)

- **Assumption**

  Instances can be projected onto lower dimensional spaces

  Lower dimensional spaces express normal instances well

  Lower dimensional spaces express anomalies significantly different

- **Implementations**

  Principal Component Analysis (PCA)

  Top few principal components capture variability in normal instances

  Smallest components capture variability in anomalies

- **Pros**

  ✓ Compatible with unsupervised modes

Robust Principal Component Analysis

- $z_1, z_2, \ldots, z_p$: projection of feature vector **x** on principle components
- $\lambda_1, \lambda_2, \ldots, \lambda_p$: eigen-values
- Anomalous: if $\sum_{i=1}^{q} \frac{z_i^2}{\lambda_i} > \chi_q^2(\alpha)$

q: number of principle components to be kept, $q \le p$

α: significance level

Agenda

- Introduction
- Classification-based
- Nearest Neighbor-based
- Cluster-based
- Statistical
- Information Theoretic and Spectral
- ***Contextual and Collective***
- Conclusion

Contextual (Intro)

- **Assumption**

  Normal instances within a context will be similar in behavior and attributes, while anomalies will be different

- **Basic Ideas**

  Identify a context around an instance via *contextual attributes*

  Finding anomalies w.r.t. context via *behavioral attributes*

- **Pros**

  ✓ Detect anomalies hard for detection when using instance anomaly detection techniques

- **Cons**

  × Rely on good contextual attributes

Contextual (Cont'd)

- **Contextual Attributes**

  Define a neighborhood (context) for each instance

  Spatial Context (Latitude, Longitude)

  Graph Context (Edges, Weights)

  Sequential Context (Position, Time)

  Profile Context (User demographics)

- **Reduction to instance anomaly detection**

  Segment data via contextual attributes

  Instance anomaly detection within segments via behavioral attributes

- **Utilizing structure in data**

  Build models from data using contextual attributes (e.g. time series)

Conditional Anomaly Detection

- Each instance is represented as [*x*, *y*]

  x: environmental (contextual) attributes

  y: indicator (behavioral) attributes

  Mixture of N_U Gaussian models, *U* is learnt from the contextual data

  Mixture of N_V Gaussian models, *V* is learn from the behavioral data

  p(V_j|U_i) indicates conditional probability of behavioral part to be generated by V_j when contextual part is generated by U_i

- For an instance [*x*, *y*]

  Anomaly score = $\sum_{i=1}^{|N_U|} p(x \in U_i) \sum_{j=1}^{|N_V|} p(y \in V_j) p(V_j \mid U_i)$

Collective (Intro)

- Exploit relationship among instances
- Sequential anomaly detection

  Detect anomalous sequences
- Spatial anomaly detection

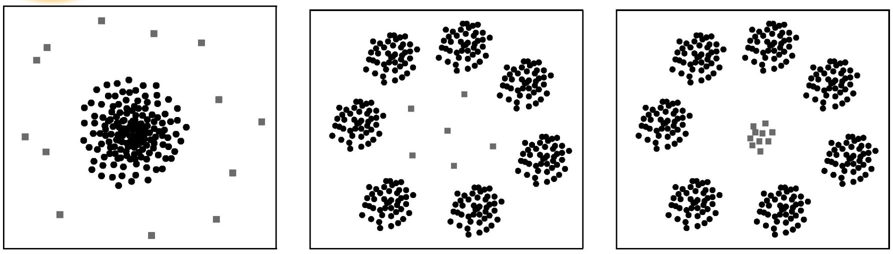
  Detect anomalous sub-regions within a spatial data set
- Graph anomaly detection

  Detect anomalous sub-graphs in graph data

Agenda

- Introduction
- Classification-based
- Nearest Neighbor-based
- Cluster-based
- Statistical
- Information Theoretic and Spectral
- Contextual and Collective
- ***Conclusion***

Conclusion



(a) Data Set 1 (b) Data Set 2 (c) Data Set 3

- Different methods work in different scenarios

  Most work on one class w/ few and far-away anomalies;

  Multi-class works on multi dense classes w/ few and sparse anomalies;

  Clustering-based and nearest neighbor-based cannot work when anomalies also cluster tightly

Conclusion (Cont'd)

- More complex scenarios …

  Nearest neighbor-based and clustering-based suffer from high dimensions;

  Spectral relies on distinguishability between normal instances and anomalies in lower dimensional spaces;

  Classification-based needs labels of both normal and anomalous instances;

  Classification-based also suffers when numbers of labels are biased;

  Statistical only works in low dimensional spaces;

  Information theoretic also requires measurements distinguishing normal instances from anomalies;

  Slow Training and Fast Testing vs Slow Testing

  What if anomalies are frequent, while normal instances are rare?

Thank you!