Anomaly Detection: A Tutorial

Theory and Applications

Sanjay Chawla[1] Varun Chandola[2]

[1]School of Information Technologies
University of Sydney
NSW, Australia
chawla@it.usyd.edu.au

[2]Computational Sciences and Engineering Division
Oak Ridge National Laboratory
Oak Ridge, TN, USA
chandolav@ornl.gov

December 14, 2011

Tutorial Outline

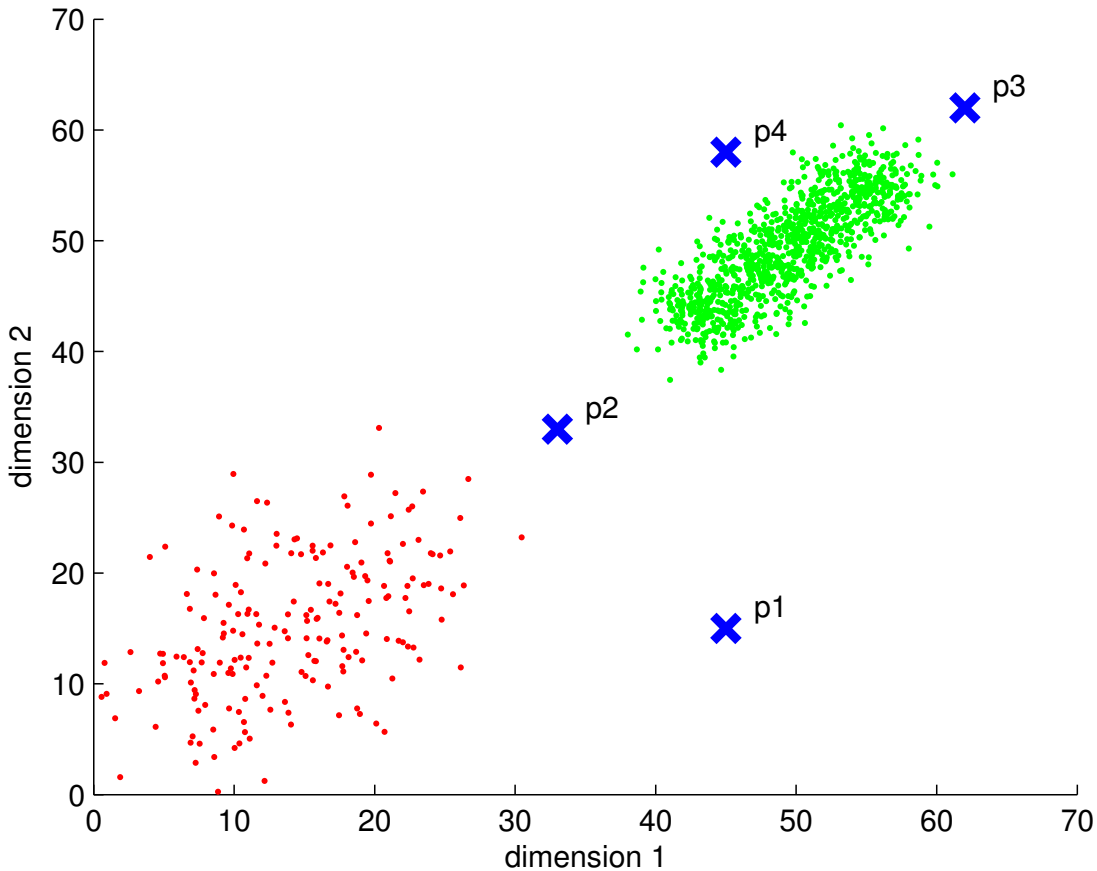
- Introduction and Overview
- Theory
  - Statistical Methods
  - Distance and Density Based Methods
  - Addressing Scalability
  - Anomalies in Complex Data
  - Evaluation Methods
- Applications
  - Network Intrusion Detection
  - Fraud Detection
  - Epidemiological Studies
  - Climate and Weather Data Analysis

Anomaly Detection - Overview

- In Data Mining, anomaly or outlier detection is one of the four tasks.
  - Classification
  - Clustering
  - Pattern Mining
  - Anomaly Detection
- Historically, detection of anomalies has led to the discovery of new theories. Famous examples include
  - El Nino and Southern Oscillation Index (SOI).
  - The discovery of the planet Neptune.
  - The use of fluoride in toothpaste!
- Anomalies often lead to "surprise" - a form of inference known as abduction (different from induction and deduction).

Definition

- Hawkins: "an outlier is an observation, which deviates so much from other observations as to arouse suspicions that it was generated by a different mechanism." [15]



Statistical Methods

- Lets begin with the univariate Normal distribution

$$f(x) = \frac{1}{(2\pi\sigma^2)^{\frac{1}{2}}} e^{-[(x-\mu)/\sigma]^2/2}$$

- Notice exponent measures square of deviation from mean and normalized by standard deviation

$$\left(\frac{x-\mu}{\sigma}\right)^2 = (x-\mu)(\sigma^2)^{-1}(x-\mu)$$

- For d dimension, the exponent is called (square of) **Mahalanobis distance**

$$(x-\mu)'\Sigma^{-1}(x-\mu)$$

where Σ is the $d \times d$ variance-covariance matrix.

Anomaly Detection with Mahalanobis Distance

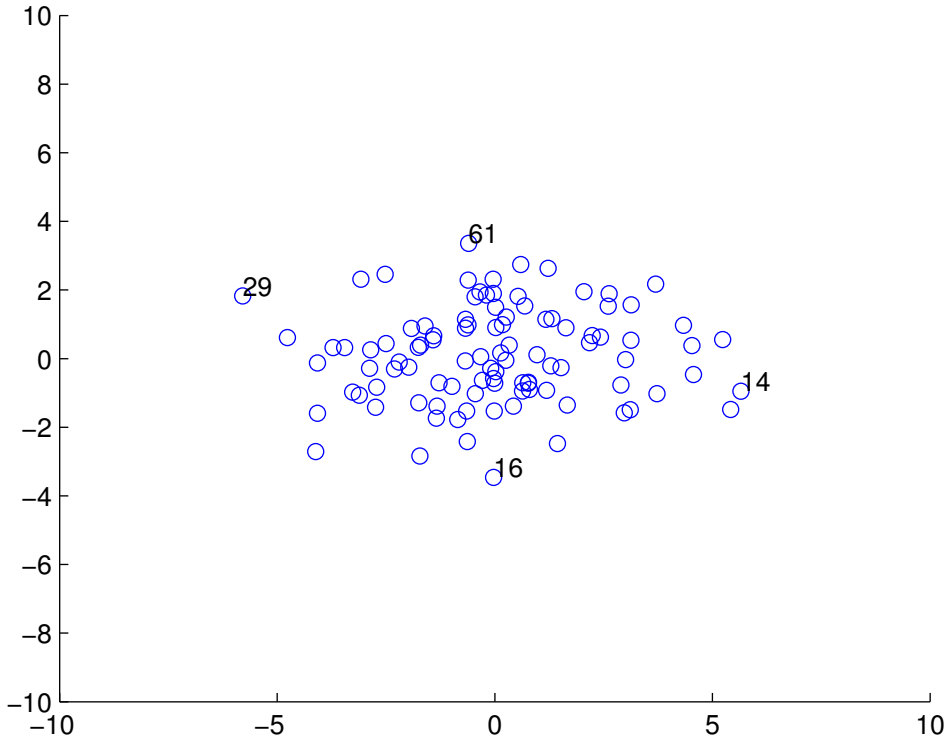
- The key observation is that if data x follows a d dimensional Gaussian distribution then:

$$(x-\mu)'\Sigma^{-1}(x-\mu) \approx \chi^2_d$$

- Anomalies can be found in the tail of the distribution.
- There are three major weaknesses of the above approach.
  - Data may not follow a Normal distribution or be a mixture of distributions.
  - Both mean and variance of χ^2 is d. For high-dimensional data this is a problem.
  - Mean and thus variance are extremely sensitive to outliers -and we are using them to find anomalies - often leads to false negatives.

Mahalanobis vs. Euclidean Distance

- Mahalanobis normalizes for variance



Point Pairs	Mahalanobis	Euclidean
(14,29)	5.07	11.78
(16,61)	4.83	6.84

Distance-based anomalies

- Intuition: A data point which is far away from its nearest neighbors is a candidate anomaly.
- Several definitions which capture the above intuition.

- $DB(p, D)$ anomaly [20]: an object o in a data set T is a $DB(p, D)$ anomaly if at least a fraction p of objects in T have distances greater than D from o.

- Generalizes the notion of "three standard deviation from the mean."
- This definition had a huge influence on subsequent development in outlier detection.

$DB(p, D)$ outlier

- To build some intuition, consider data generated from the Normal distribution $N(0, 1)$. Then if O is a $DB(p, D)$ outlier:

$$\frac{1}{(2\pi)^{\frac{1}{2}}} \int_{O-D}^{O+D} e^{-\frac{x^2}{2}} \, dx \quad \leq 1 - p$$

- Example: If O is 3 (3 standard deviations away from the mean) then it is a $DB(0.1, 0.999)$ outlier.
- Thus for particular settings of D and p, $DB(p, D)$ captures standard outliers.
- But much more general (e.g., any distance metric).

Distance-based methods ($DB(k, N)$

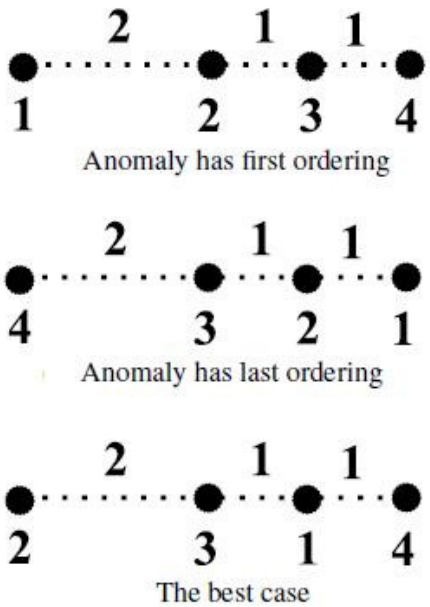
- $DB(k, N)$ anomaly [28]: top N data instances whose distances to its k-th nearest neighbor are largest.
- Several advantages. Ranking for anomalies is more intuitive. Setting of parameters generally easier.
- A Simple Nested Loop (SNL) algorithm can be used to select the top N, $DB(k, N)$ outliers. Time complexity is $O(n^2 d)$ where n is the database size and d is the dimensionality.

Pruning rule

- $DB(k, N)$ anomaly [3]: a data instance is not an anomaly if its distance to its k-th current nearest neighbor is less than the score of the weakest anomaly among top N anomalies found so far.
- A large number of non-anomalies can be pruned without carrying out a full data search.
- Complexity: nearly $O(n)$

Examples of pruning technique

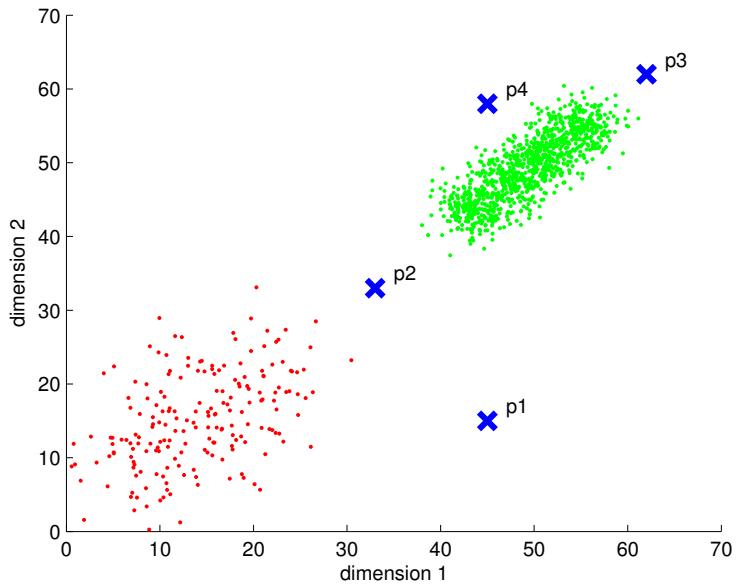
- Non-anomalies are pruned earlier.



Index	Distance to knn	Comparisons	Weakest anomaly	Weakest score
Anomaly has first ordering (No. of comparisons = 10)				
1	2	3	1	2
2	1	2	1	2
3	1	2	1	2
4	1	3	1	2
Anomaly has last ordering (No. of comparisons = 12)				
1	1	3	1	1
2	1	3	1	1
3	1	3	1	1
4	2	3	4	2
The best case (No. of comparisons = 8)				
1	1	3	1	1
2	2	3	2	2
3	1	1	2	2
4	1	1	2	2

Strengths and weaknesses - Distance-based techniques

- Do not make any assumption about the distribution of the data
- Scalable for large dataset $(O(n))$
- Capable of finding only global anomalies
- Can lead to non-intuitive results in Top-k situations



Density-based anomaly

- Calculate the density of an object based on the density of its k nearest neighbours.

$$density(p) = 1/\left(\frac{\sum_{q \in N_k(p)} dist_k(p,q)}{|N_k(p)|}\right)$$

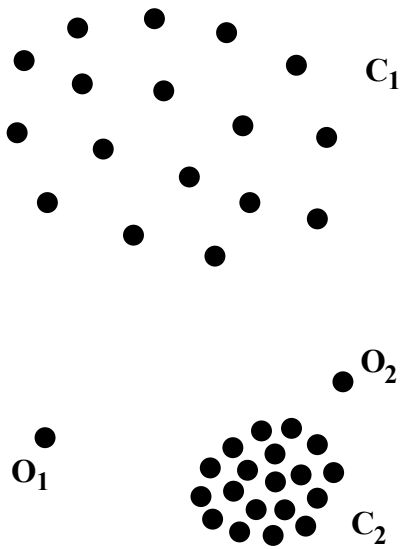
$$relative\text{-}density(p) = \frac{density(p)}{\frac{1}{|N_k(p)|}\sum_{q \in N_k(p)} density(q)}$$

$$anomaly\text{-}score(p) = \frac{1}{relative\text{-}density(p)}$$

- LOF: indicates a degree of local outlier-ness [6]

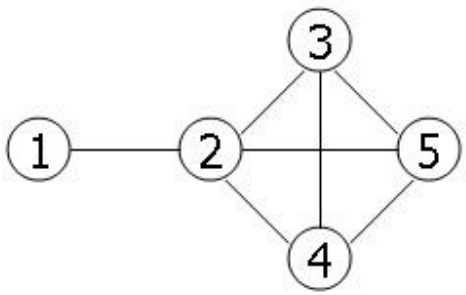
Strengths and Weaknesses

- Can detect global and local anomalies
- Cannot use pruning technique and has a complexity of $O(n^2)$
- Require a method combining the strengths of distance and density based approaches? A distance based approach which can capture density?



Commute time

- Commute time between i and j is the expected number of steps that a random walk starting at i will take to reach j once and go back to i for the first time.
- Commute time can capture both the distance between points and the data densities.



Euclidian Distance						Commute Distance				
Index	1	2	3	4	5	1	2	3	4	5
1	0	**1.00**	1.85	1.85	2.41	0	**12.83**	19.79	19.79	20.34
2	1.00	0	1.00	1.00	1.41	12.83	0	6.96	6.96	7.51
3	1.85	1.00	0	1.41	1.00	19.79	6.96	0	7.51	6.96
4	1.85	1.00	1.41	0	1.00	19.79	6.96	7.51	0	6.96
5	2.41	1.41	1.00	1.00	0	20.34	7.51	6.96	6.96	0

Computation of commute time

- Commute time can be computed using graph Laplacian matrix L

$$c_{ij} = V_G(e_i - e_j)^T L^+ (e_i - e_j)$$

L^+: pseudo-inverse of L
V_G: graph volume
e_i: i-th column of the identity matrix

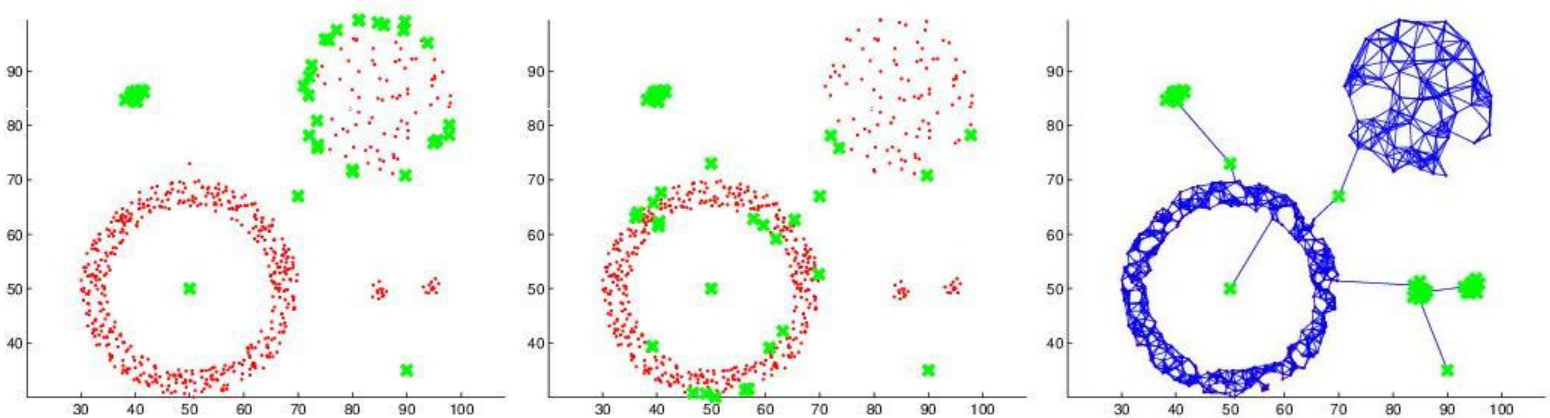
- Commute time is Euclidean distance in the space spanned by eigenvectors of L.

$$c_{ij} = V_G[(S^{-1/2}V^T)(e_i - e_j)]^T[(S^{-1/2}V^T)(e_i - e_j)]$$

V, S: eigenvectors and eigenvalues of L

Anomaly detection using commute time (CDOF)

- Construct the mutual k nearest neighbor graph G from the dataset
- Compute the Laplacian matrix L of G and its eigensystems
- Find top N anomalies using the distance-based technique in commute time with pruning rule
- Complexity: $O(n^3)$
- Commute time method can detect global, local, and group anomalies.



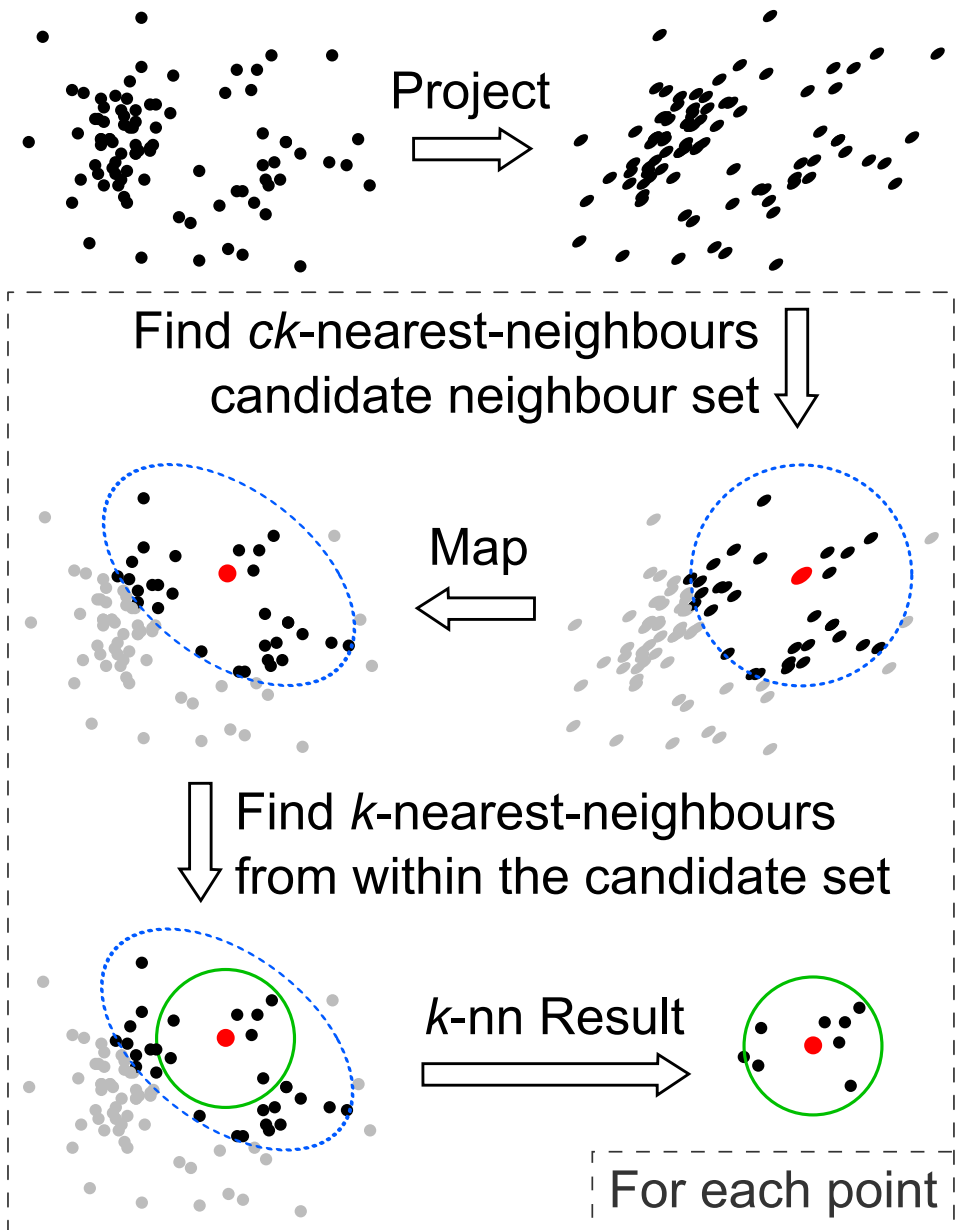
Fast estimation of commute time

- Speilman and Srivastava [31] combined random projection and a linear time solver to build a structure where we can compute the compute time between two nodes in $\widetilde{O}(\log n)$ time.
- Complexity of CDOF: $O(n^3) \rightarrow O(n \log n)$
- Uses a near linear time solver for a linear system of equation $Ax = b$
- Spielman and Teng solvers. Also see work by Iannis Koutis from CMU.

Scalability for Density-based method

- The pruning rule for Distance-based methods does not apply to Density-based approaches.
- We can go from $O(n^2)$ to nearly $O(n \log n)$ by using an index.
- One solution for the curse of high dimensionality is to use of random projections.

PINN Algorithm (ICDM 2010)



PINN Guarantee

- The PINN Algorithm provides probabilistic guarantees.
- Under certain assumptions about intrinsic dimensionality (c) with high probability

$$\frac{1-\epsilon}{1+\epsilon} \cdot LOF(p) \leq \overline{LOF}(p) \leq \frac{1+\epsilon}{1-\epsilon} \cdot LOF(p).$$

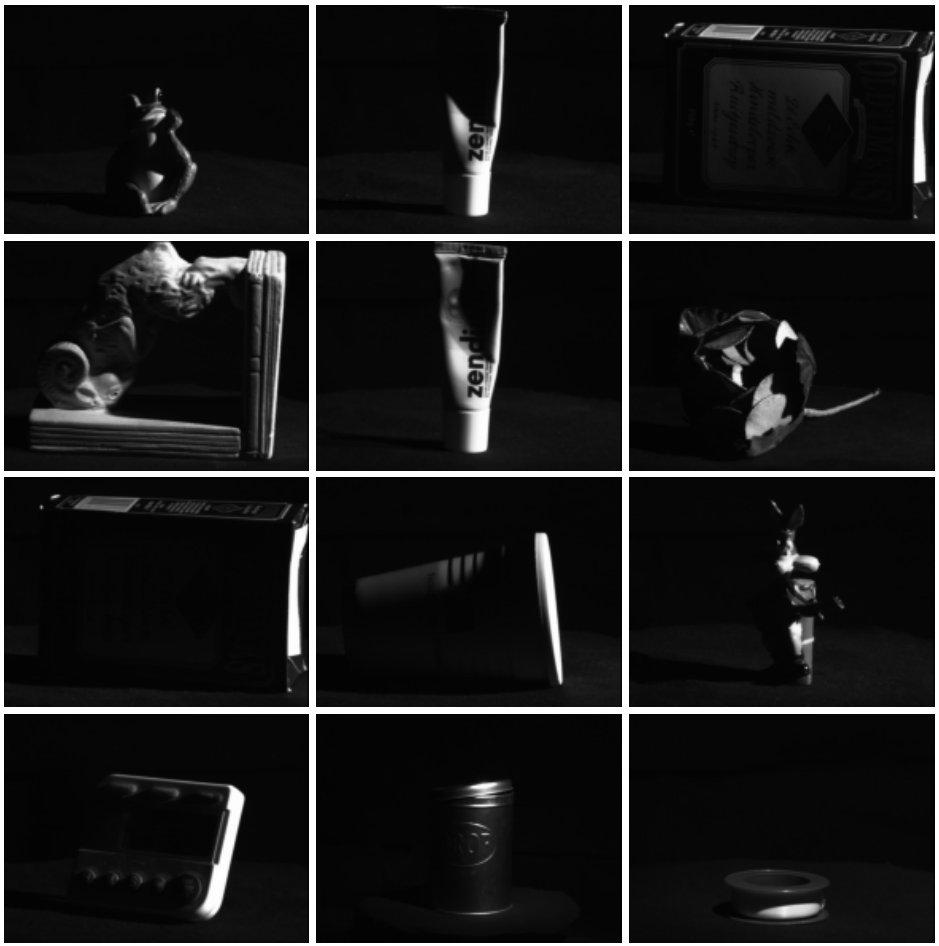
- In practice we do not know the intrinsic dimensionality of data. However random projections are quite robust.

Examples: high-dim distance-based outliers

- On a large database of images, the bright images show up as distance based outliers

Examples: high-dim density-based outliers

- On a large database of images, occluded images show up as density based outliers



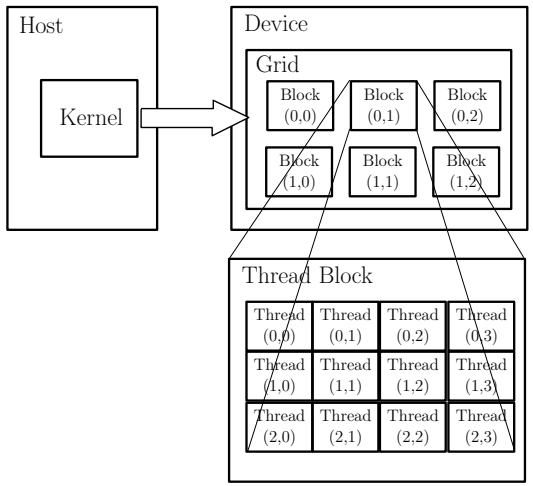
Examples: local density-based outliers

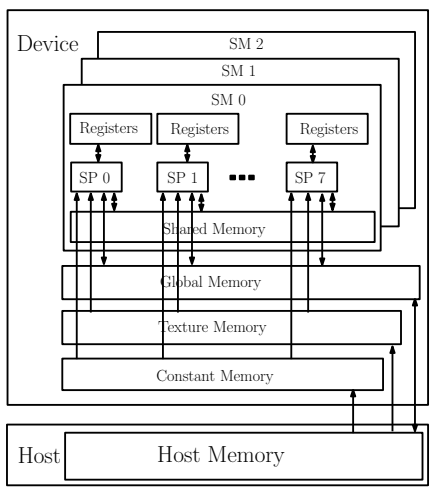
- Examples of images ranked by LOF

Addressing Scalability

Using GPUs for Anomaly Detection

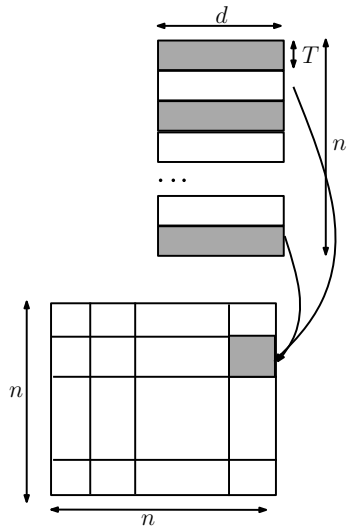
- Well suited for **data parallel** algorithms
  - Using CUDA - *Compute Unified Device Architecture* (Nvidia)
- Need to re-engineer existing algorithms
  - Utilization of device memory
  - Minimize CPU⇔GPU transfer of data
  - Keep threads homogeneous
- Most model based algorithms are naturally setup for the testing phase
- Model building needs careful redesign
- What about unsupervised algorithms?





Implementing $DB(k, N)$ on GPUs

- Return top N data instances whose distances to k-th nearest neighbor are largest (Serial $DB(k, N)$ is $O(n^2)$)
- Involves computing pairwise distances
- Load block i and block j to shared memory
  - Data layout in memory should be optimized
- Each thread computes distance between a pair of instances
  - Can utilize this time to load next chunk of data from host to device memory
- Writes results to corresponding output block
- Sorting can be done efficiently in CUDA [29]



Moving Beyond Multi-dimensional Record Data

Categorical (Mixed)

- Fraud Detection
- Cyber Networks

Discrete Sequences

- Genomic
- System Calls

Spatio-temporal

- Remote sensing
- Climate

Time Series

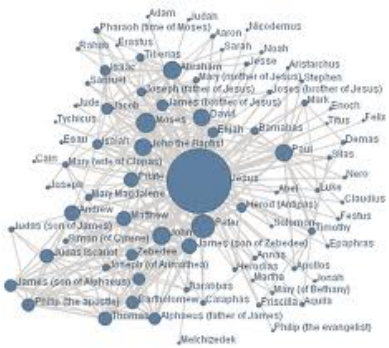
- Sensor Networks
- Healthcare

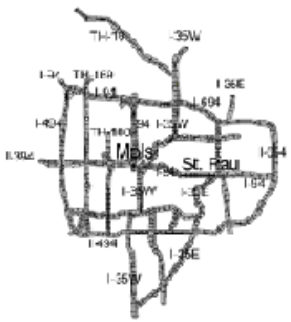
Spatial

- GIS
- Image analysis

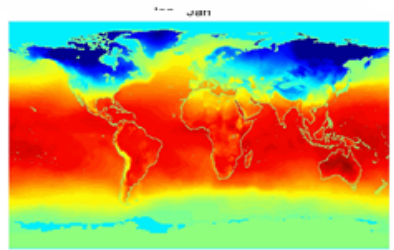
Graphs

- Social networks
- Epidemiology





```
GGTTCCGCCTTCAGCCCCGCGCC
CGCAGGGCCCGCCCCGCGCCGTC
GAGAAGGGCCCGCCTGGCGGGCG
GGGGGAGGCGGGGCCGCCCGAGC
CCAACCGAGTCCGACCAGGTGCC
CCCTCTGCTCGGCCTAGACCTGA
GCTCATTAGGCGGCAGCGGACAG
GCCAAGTAGAACACGCGAAGCGC
TGGGCTGCCTGCTGCGACCAGGG
```



Handling Categorical Data

- Each attribute can belong to one of many categories.
- No ordering between categories
- Mixed data (categorical and continuous attributes)?

cap-shape	cap-surface	···	habitat	type
convex	smooth		urban	poisonous
convex	smooth		grasses	edible
bell	smooth		meadows	edible
convex	scaly		urban	poisonous
convex	smooth		grasses	edible
...				

Table: Mushroom Data Set [2].

Approaches to Identify Categorical Anomalies

Using Association Analysis [24]

- *Binarize* data
- Learn **rules** ($X \Rightarrow Y$)
  - Choose high confidence rules ($P(Y|X)$)
- For test record $Z = \langle X, Y \rangle$ find rules of the form $P(!Y|X)$
  - Y is not observed when X is observed

Using Bayesian Networks [33]

- Learn Bayesian network structure and parameters
- Compute $P(Z)$ for test data record Z
- Flag anomaly if $P(Z) < \delta$

Using Similarity Metrics [10]

- Use a similarity measure ($S(X_1, X_2)$)
- Apply distance/density/clustering based method (e.g. *lof*)

Conditional Probability Test [18]

- Identify unusual combinations of attribute values

$$r(a_t, b_t) = \frac{P(a_t, b_t)}{P(a_t)P(b_t)}$$

- $\mathbf{A} \cap \mathbf{B} = \phi$
- *Assumption*: If $r(a_t, b_t)$ is low and is observed in test record $\mathbf{t}$, then $\mathbf{t}$ is anomalous
- For a test record $\mathbf{t}$:
  - For each *mutually exclusive* pair of attribute sets $\{\mathbf{A}, \mathbf{B}\}$ compute $r(a_t, b_t)$
  - Score $\mathbf{t}$ based on all r-values:
    - Assign minimum r-value as score
    - Take product of all r-values
- Need to compare exponential pairs of subsets!!!
  - Only consider subsets upto size k
  - Ignore subsets with frequency less than a threshold α
  - Avoid comparing *independent* subsets of attributes

$$\mu(A, B) \geq \beta_\mu$$

Estimating Probabilities for CPT [18]

- *Maximum Likelihood Estimation*

$$\frac{P(a_t, b_t)}{P(a_t)P(b_t)} = \frac{C(a_t, b_t)}{N} \times \frac{N}{C(a_t)} \times \frac{N}{C(b_t)}$$

  - $C(a_t)$: Number of training instances with $\mathbf{A} = a_t$
  - N: Total number of training instances

- *Laplace Smoothing*:

$$E(p) = \frac{C(p) + 1}{N + 2}$$

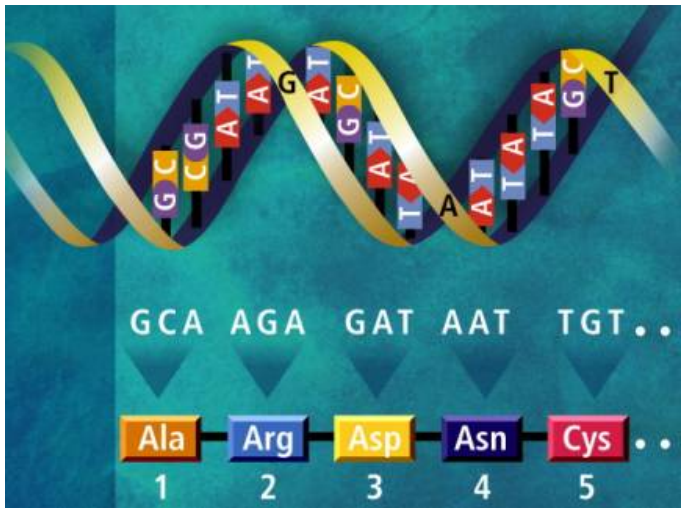
$$\boxed{r(a_t, b_t) = \frac{E(a_t, b_t)}{E(a_t) \times E(b_t)}}$$

Speedup Tricks

- Replace rare attribute values with generic attribute (reduce *arity*)
- Use efficient data structure to querying for counts(AD Trees [26])
  - ADTrees work faster for low arity attributes

Finding Anomalies in Discrete Sequences

- Many problem formulations:
  1. **Anomalous symbols** in a sequence
  2. **Anomalous subsequence** in a sequence
  3. **Anomalous sequence** in a database of sequences
- See [11] for a comparative evaluation, [9] for a survey



$login, pwd, mail, ssh, \ldots, mail, web, logo$
$login, pwd, mail, web, \ldots, web, web, web,$
$login, pwd, mail, ssh, \ldots, mail, web, web,$
$login, pwd, web, mail, ssh, \ldots, web, mail,$
$\mathbf{login}, \mathbf{pwd}, \mathbf{login}, \mathbf{pwd}, \mathbf{login}, \mathbf{pwd}, \ldots,$

Treating Sequences as Points

- Utilize a distance/similarity measure
  - Plug into a distance/density/clustering based method
- Simplest: *Hamming Distance*

$$\begin{aligned} h(A_i, B_i) &= 1, A_i \neq B_i \\ &= 0, A_i = B_i \\ H(A, B) &= \sum_{i=1}^{n} h(A_i, B_i) \end{aligned}$$

- Issues: Unequal lengths, misalignment
- Normalized Length of Longest Common Subsequence

$$D(A, B) = 1 - \frac{|LCS(A, B)|}{\sqrt{|A||B|}}$$

  - Standard Dynamic Programming method is slow
  - Faster versions available (*Hunt Szymnaski* method [7])
- Weaknesses:
  - Cannot *localize anomalies* within a sequence
  - Weak anomaly signals might get lost

Using Sliding Windows

- Slide a window of size k
- Extract all windows from a sequence $(n - k + 1)$
- Training (Creating a normal dictionary): Store all unique windows in all normal sequences and their counts
- Testing:
  - For each window find the frequency in *normal dictionary*
  - Anomaly score is *inverse* of the aggregate frequencies for all windows (normalized by length)
- Many variants exist:
  - For each window find the hamming distance to the closest window in the normal dictionary [16]
- Issues:
  - Penalizes low frequency windows in the normal dictionary
  - Rewards high frequency windows that might not be *relevant*
  - Can construct anomalous sequences that will escape detection

Using Probabilistic Models

- Probability of occurrence of sequence S

$$P(S) = \prod_{i=1}^{n} P(S_i|S_1, \ldots, S_{i-1})$$

- *Short memory* property of sequences:

$$P(S_i|S_1, \ldots, S_{i-1}) = P(S_i|S_{i-k}, \ldots, S_{i-1})$$

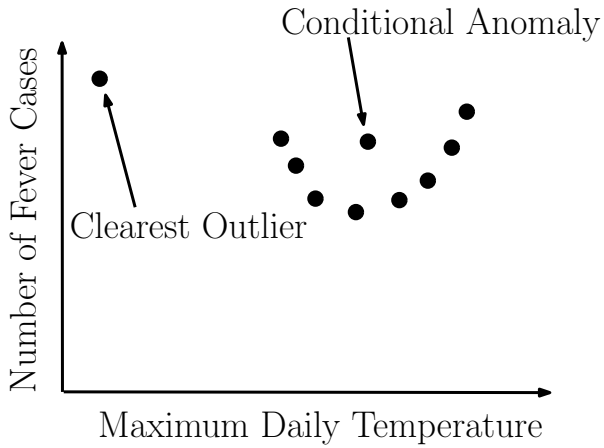
- Conditional probability estimates for a symbol S_i:

$$P(S_i|S_{i-k}, \ldots, S_{i-1}) = \frac{f(S_i|S_{i-k}, \ldots, S_i)}{f(S_i|S_{i-k}, \ldots, S_{i-1})}$$

  - f is estimated from the *normal dictionary*
- Anomaly score for a test sequence is inverse of the normalized probability of occurrence
- Issues: What if the suffix occurs very infrequently in the normal data (or not at all)?
  - Replace with the longest suffix that occurs sufficient number of times [32] - Probabilistic Suffix Trees
  - Significantly reduces the size of the model

Finding Contextual Anomalies

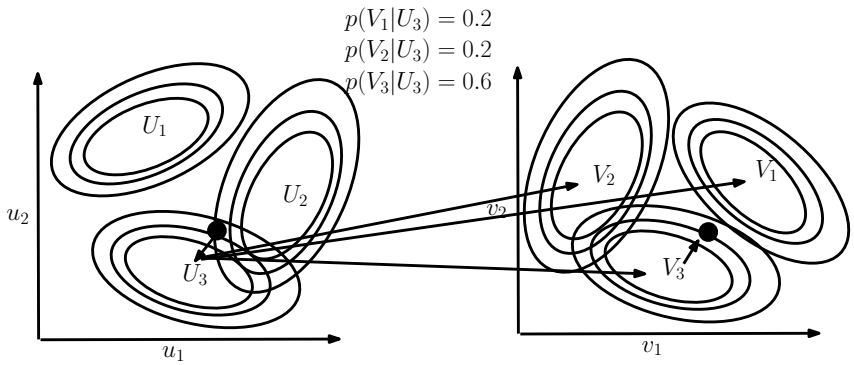
- Sometimes *contextual information* is available about data
  - Not used directly as a feature
  - Are well understood, no anomalies in the context
  - Can reduce false positives and yield interesting anomalies



- Example adapted from [30]
  - Contextual anomalies - Anomalous with respect to a *context*
  - Context is defined using *environmental* variables
    - Spatial (*Latitude, Longitude*)
    - Graph context (*Edges, Weights*)
    - Temporal location
    - Domain specific (*Demographic, other*)
  - How to incorporate context?
    - Reduce to traditional anomaly detection (subset on context)
    - Explicitly model contextual information (time series, spatial)

Conditional Anomaly Detection

- Data instance $\mathbf{d} \Rightarrow$
  $u_1, u_2, \ldots, u_{d_U}, v_1, v_2, \ldots, v_{d_V}$
  - d_U environmental attributes
  - d_V indicator attributes

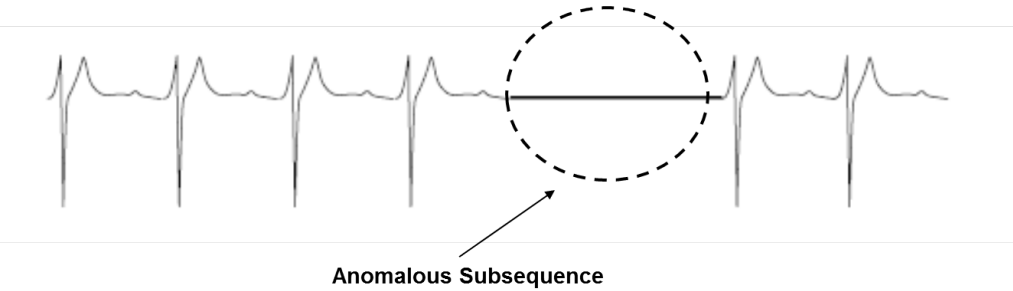


- **Algorithm** [30]:
  1. Learn a Gaussian Mixture Model (GMM) $U = U_1, U_2, \ldots, U_{n_U}$, each with dimensionality d_U
  2. Learn a set of Gaussians $U = V_1, V_2, \ldots, V_{n_V}$, each with dimensionality d_V
  3. Learn a probabilistic mapping function $p(V_j|U_i)$
  4. Score a test instance $\mathbf{d} = [\mathbf{u}, \mathbf{v}]$:

$$S = \sum_{i=1}^{n_U} p(\mathbf{u}|U_i) \sum_{j=1}^{n_V} p(\mathbf{v}|V_j) p(V_j|U_i)$$

Finding Collective Anomalies

- Find a collection of data points
- Each point by itself is normal
- The collection *as a whole* is anomalous
- Relevant when data has inherent structure, and
- When domain definition of anomalies cannot be described as point anomalies



A Simple Solution

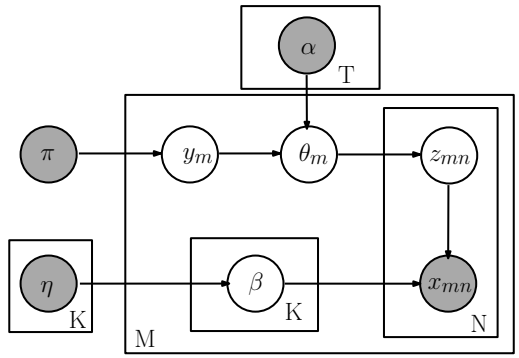
1. Break data into groups
2. Compute features for each group
3. Apply traditional anomaly detection

Examples

1. Time series
2. Image
3. Spatial clusters of galaxies

Using Latent Dirichlet Allocation for Group Anomalies

- Find **anomalous groups** in data [34]
- Example: Spatial clusters of galaxies
  - **topics**: *red*, *green*, *emissive*
  - **words**: continuous features



Flexible Genre Model (FGM)

- For each **group**:
  1. Draw a genre $1, 2, \ldots, T \ni y_m \sim \mathcal{M}(\pi)$
  2. Draw topic distribution for $y_m : S^K \ni \theta_m \sim Dir(\alpha_{y_m})$
  3. Draw K topics $\{\beta_{mk} \sim P(\beta_{mk}|\eta_k)\}_{k=1,2,\ldots,K}$
  4. For each **point** in group:
     1. Draw topic membership: $z_{mn} \sim \mathcal{M}(\theta_n)$
     2. Generate point $x_{m,n} \in P(x_{m,n}|\beta_{m,z_{m,n}})$

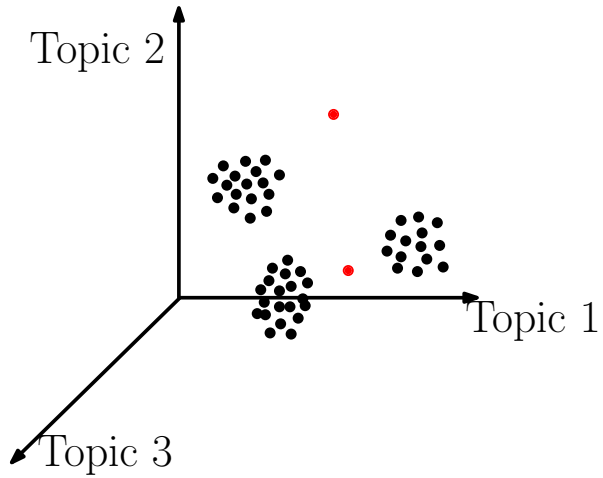
Model Parameters

- $\mathcal{M}(\pi)$ - Multinomial
- *Genre* - $Dir(\alpha_t)$
- Topic generators $P(.|\eta_k)$- *GIW*
- Point generators $P(x_{mn}|\beta_{nk})$ - Multivariate *Gaussian*

Inference and Testing for FGM

Inference and Learning Parameters

- Approximate inference of latent variables (*Gibbs Sampling*)
- Use samples to learn parameters (*Single step Monte Carlo EM*)



Anomaly Detection

- Infer the topic distribution θ_m
- Compute negative log likelihood w.r.t. α_t
- Rationale: An anomalous group will be unlikely to be generated from any genre
- Geometric interpretation: Mapping each group into a T dimensional space and finding anomalies

Evaluating Anomaly Detection Methods - Labels

- Labeled *validation data set* exists
  - Confusion matrix
  - Traditional evaluation metrics
    - *Class imbalance?*
  - ROC Curve
- *Validation* set does not exist
  - Use *domain expertise* to find TP and FP
  - FN is harder to estimate
    - Pseudo false negative estimation techniques [25]

		Predicted	
		a	**n**
Actual	**a**	TP	FN
	n	FP	TN

$$Acc = \frac{TP + TN}{\sum}$$

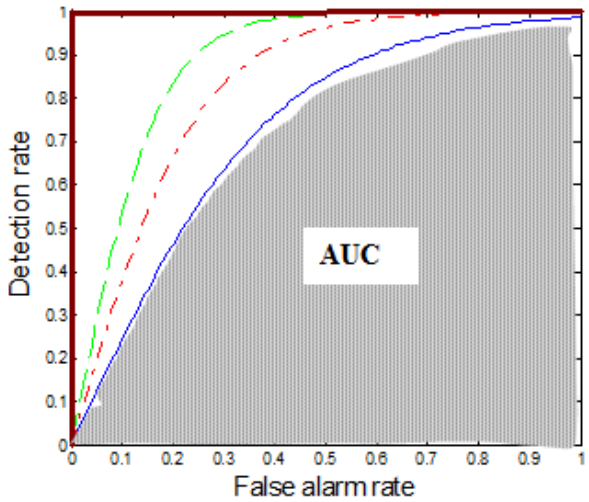
$$Rec\ (R) = \frac{TP}{TP + FN}$$

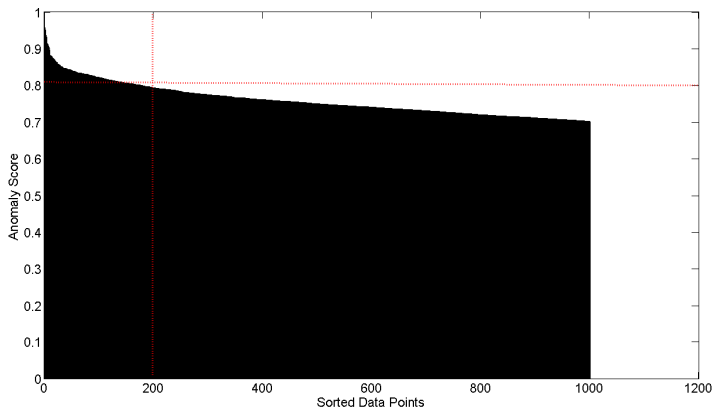
$$Prec\ (P) = \frac{TP}{TP + FP}$$

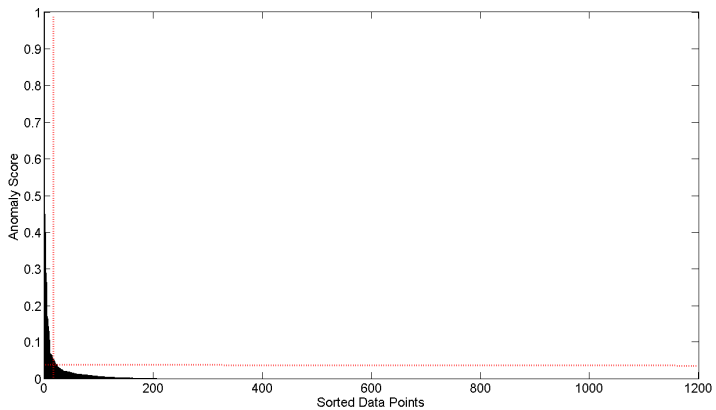
$$F = \frac{2 * R * P}{R + P}$$

Evaluating Anomaly Detection Methods - Scores

- Convert to binary output
  - Use threshold δ on score (Scale issues? [21])
  - Take top x% as anomalies
- ROC curve by varying x or δ
- Quality of output
  - Does the output "suggest" x or δ?
  - Which output is better?

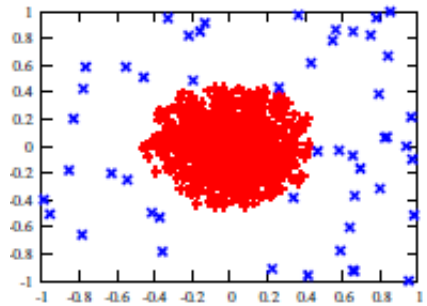






Unifying Scores

- Different methods assign scores in different ranges
  - *kNN*-based scores $[0, 1]$
  - *Lof* scores $[1, 6]$
  - *ABOD* scores $[0, 80000]$!!
    - Anomalies have lower scores
  - Direct scaling to $[0, 1]$ might lose distinction between normal and anomalies
  - Desired scaling: Stretch *interesting* ranges and shrink irrelevant ones



- Generalized Procedure for Normalizing Outlier Scores [21]
  - *Regularity:* $\Rightarrow S(o) \geq 0, \forall o$, $S(o) \approx 0$ if o is normal and $S(o) \gg 0$ if o is anomalous
  - *Normality:* S is regular and $S(o) \in [0, 1], \forall o$

Regularization and Normalization of Scores

Regularization

1. $R(o) := max\{0, S(o) - base_S\}$
2. $R(o) := S_{max} - S(o)$
3. $R(o) := -\log \frac{S_{max}}{S(o)}$

Normalization

1. $N(o) := \frac{S(o)}{S_{max}}$
2. $N(o) := max\left(0, erf\left(\frac{S(o)-\mu_S}{\sigma_S.\sqrt{2}}\right)\right)$ (*Gaussian Scaling*)
   - Suited for high dimensional data
3. $N(o) := max\left(0, \frac{cdf_S^{\gamma}(o)-\mu_\gamma}{1-\mu_\gamma}\right)$ (*Gamma Scaling*)
   - Where, $cdf_S^{\gamma}(o) := P(k, S(o), \theta)$
   - P is the regularized Gamma function
   - Suited for low dimensional data

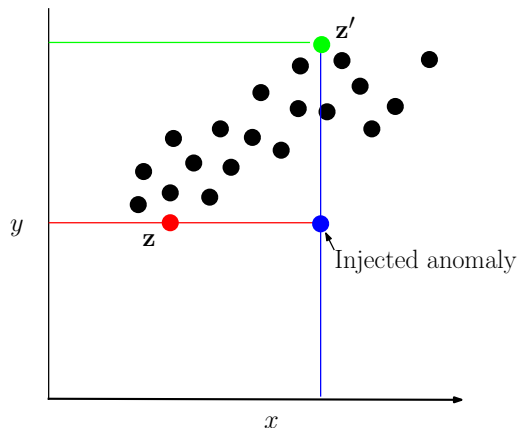
Generating Labeled Data for Validation

Generating Both Normal and Anomalous Data

- Use generative models for normal and anomalous behavior
- Several generators available
  - Multivariate continuous data [27]
  - Multivariate categorical data [5]
  - Discrete sequences using HMM [12]
- Drawbacks: Might not capture the domain characteristics

Injecting Anomalies - Random Perturbation [30]

- Given data point $\mathbf{z} = \{\mathbf{x}, \mathbf{y}\}$, $\mathbf{x}$ and $\mathbf{y}$ are *partitions* of feature space
- Take a random sample D of the entire data set
- Let $\mathbf{z}' = \{\mathbf{x}', \mathbf{y}'\} \in D$, such that distance between $\mathbf{y}$ and $\mathbf{y}'$ is maximum
- Replace $\mathbf{x}$ with $\mathbf{x}'$ and add $\mathbf{z}$ back to data set



Applications: Overview

- How to set up an anomaly detection solution for a given application domain?
  - **Available data?**
  - **Define anomalies**, define normal behavior
  - Identify requirements and constraints (online, real-time, limited resources)
  - What domain knowledge available
    - Feature identification
    - Defining normal and anomalous behavior
    - Tuning parameters
  - Available ground truth (training, **validation**)

What is Network Anomaly Detection?

Anomaly Detection or Intrusion Detection?

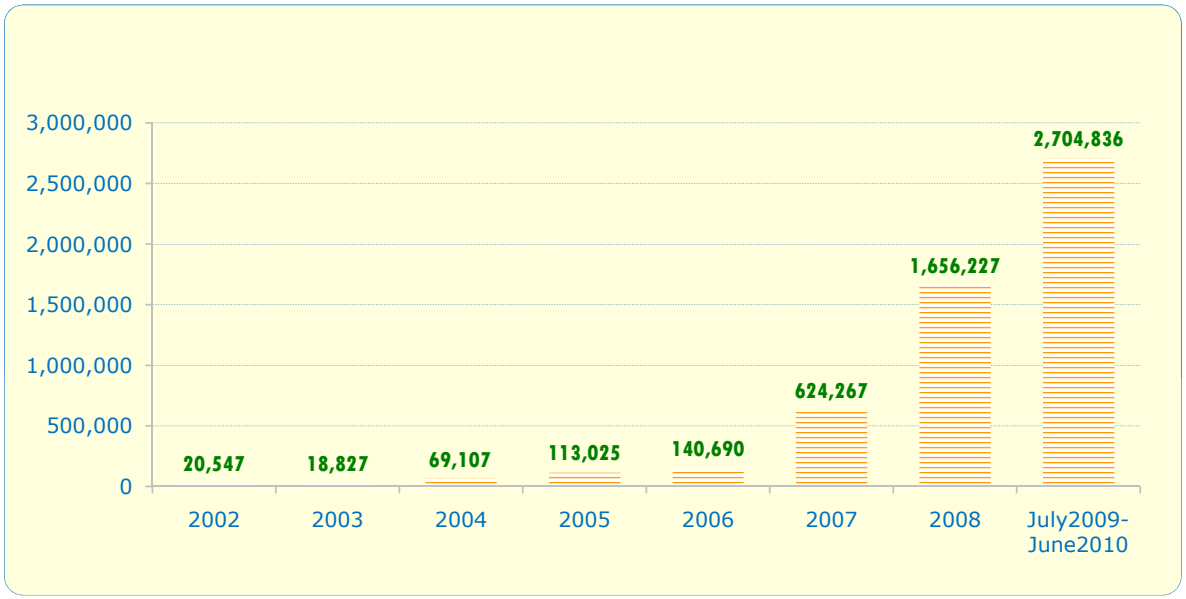
Traditional Intrusion Detection Systems (IDS): finding attacks corresponding to predefined pattern data set known as signatures, therefore system is absolutely vulnerable against zero-day attacks

Network Anomaly Detection Systems (NADS): to detect zero-day attacks without any pre-identified signature besides profile normal behavior of the network and address suspected incidents

- **Network Anomaly Detection:** finding unusual and large changes in the network traffic.
- **Examples:** intentional attacks (e.g Distributed Denial of Service - DDoS) or unusual network traffic (e.g flash crowds).

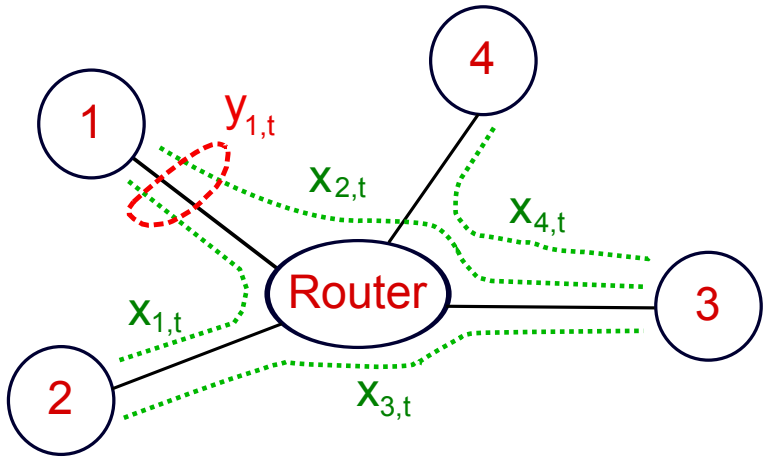
Motivation: How Much Serious?

- According to Symantec report, released in early 2011, more than 286 million new threats have been detected in 2010 which is a huge number.



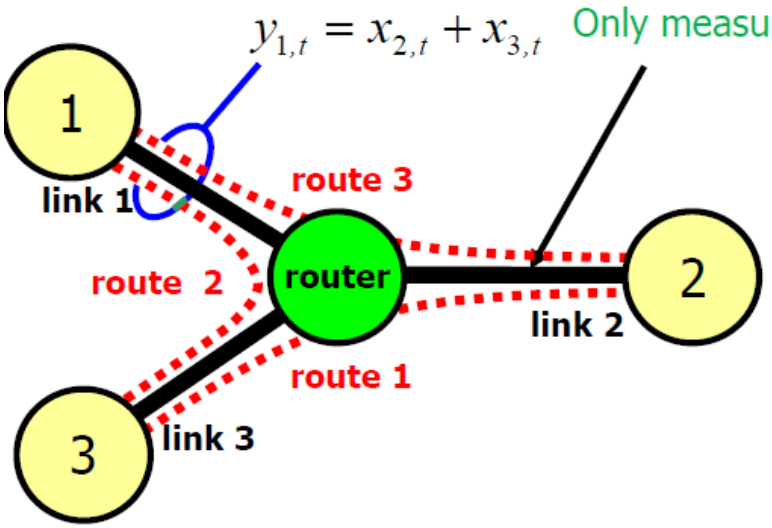
Network Topology

- A Typical network



- Origin-Destination (OD) flow is the traffic that enters at an origin node and exits at a destination node of a backbone network: $x_{1,t}$, $x_{1,t}$.
- Link measurement is the traffic enters at an node during an interval: $y_{1,t}$
- Relationship between link traffic and OD flow traffic is captured by the routing matrix A.

Network Anomalies Detection: Problem



$$\begin{pmatrix} y_{1,t} \\ y_{2,t} \\ y_{3,t} \end{pmatrix} = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix} \begin{pmatrix} x_{1,t} \\ x_{2,t} \\ x_{3,t} \end{pmatrix}$$

$$\mathbf{Y_t = A_t x_t} \quad (t=1,\ldots,T)$$

A has size (No of links) x (no of OD flows), A_{ij} = 1 if OD flow j traverses through link i

$$\mathbf{Y = AX}$$

Time-*invariant* A_t (= A), $Y=[y_1 \ldots y_T]$, $X=[x_1 \ldots x_T]$

Typically massively under-constrained!

Network Anomalies Detection: Problem

Every sudden change in an OD flow X is formally considered to be a volume anomaly...

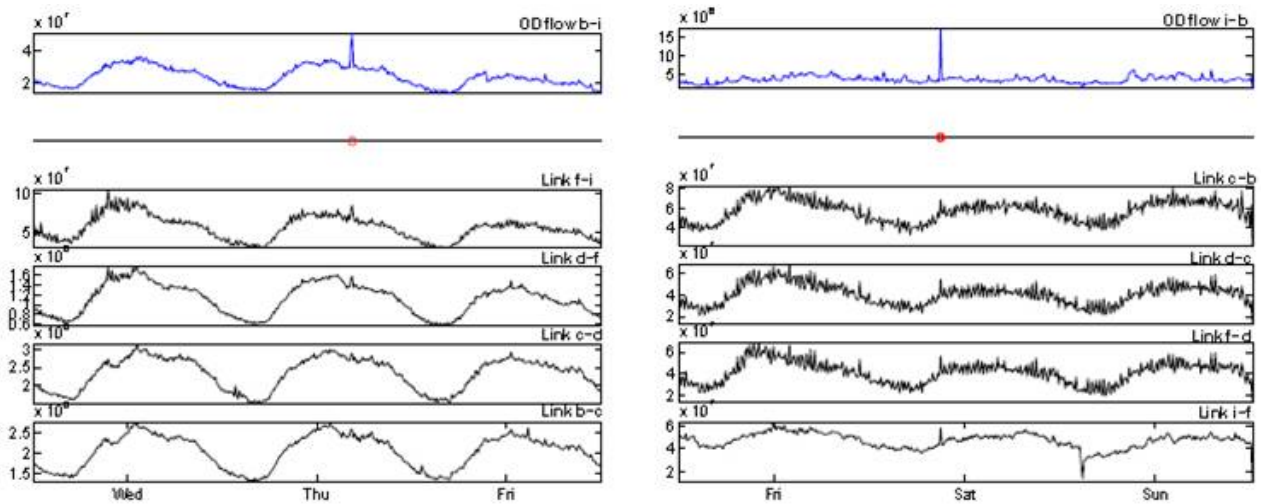
$$\begin{array}{c} \\ time\cdot bin\cdot 1 \\ \vdots \\ \vdots \\ time\cdot bin\cdot t \\ \vdots \\ \vdots \\ time\cdot bin\cdot n \end{array} \begin{array}{c} \begin{array}{cccccc} OD_1 & \dots & \dots & OD_j & \dots & \dots & OD_m \end{array} \\ \begin{pmatrix} x_{1,1} & \dots & \dots & x_{j,1} & & & x_{m,1} \\ & \ddots & & & \iddots & & \\ x_{1,t} & & & x_{j,t} & & & x_{m,t} \\ & \iddots & & & \ddots & & \\ x_{1,n} & & & x_{j,n} & & & x_{m,n} \end{pmatrix} \end{array}$$

Anomalous?

- **A network with *m* node will have *m^2* OD flows.**
  - » Thus OD flows are high dimensional data.
    - 20 node will result in 400 dimensions.
- **However, quite intuitively OD flows are correlated.**
  - » Hence they can be represented with far fewer dimensions.

Why care about OD Flows

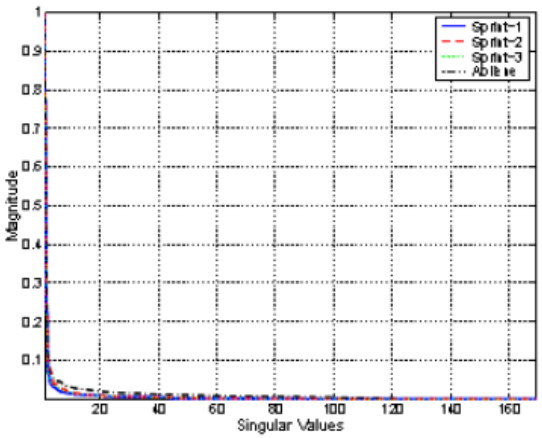
- Volume anomaly typically arises on an OD flow (traffic arriving at one node and destined for another node)
- If we only monitor traffic on network links, volume arising from an OD flow may not be noticeable. Thus, naive approach wont work if OD flow info is not available



- Figure source [22]

PCA and Subspace Method

- If data along the p dimensions are correlated (high positive or negative covariance), then it can be represented with fewer dimensions (k)
- Only 5-10 dimensions are sufficient to capture 95+$\%$ of the traffic, Lakhina et al. (SIGMETRICS'04)



Data mapped onto the k dimensions are usually called the normal component, remaining data is called the residual component

$$Z = \hat{Z} + \tilde{Z}$$

Traffic vector of all links at a particular point in time

Normal traffic vector

Residual traffic vector

Subspace Method Algorithm

- **Step1**- Determine the PCs based on eigenvalue decomposition of the covariance matrix of the dataset
- **Step2**- Choose first top k principle components with the highest eigenvalues as matrix P
- **Step3**-normal traffic subspace called $\hat{Z}$:

$$\hat{Z} = PP^T Z = CZ$$

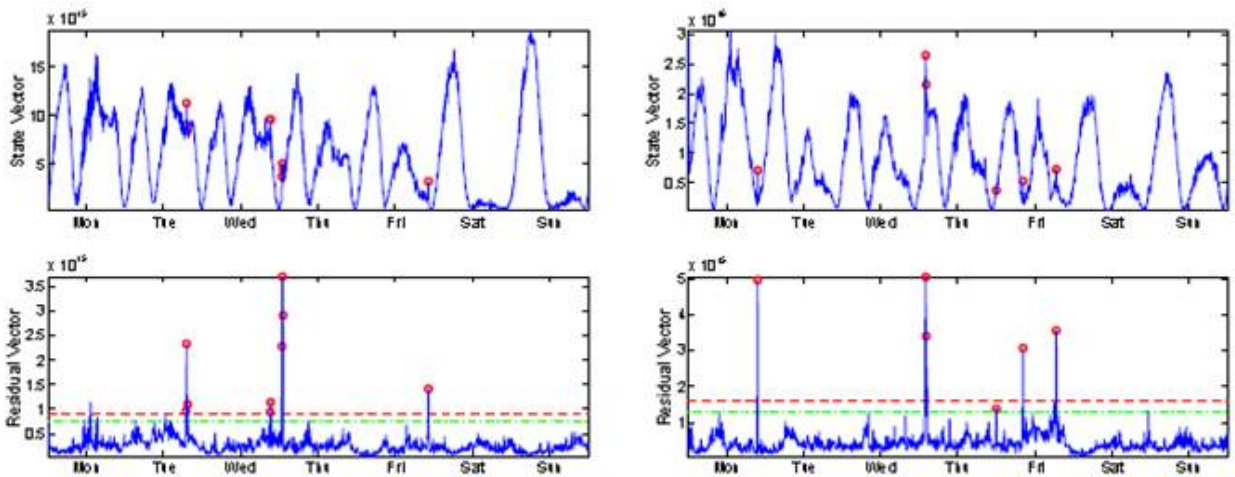
- **Step4**-abnormal traffic subspace called $\tilde{Z}$:

$$\tilde{Z} = (I - PP^T)Z = \tilde{C}Z$$

- **Step5**-If the norm of a vector is large then it is an "anomaly".

Subspace Analysis Results

- Note that during anomaly, normal component does not change that much while residual component changes quite a lot
- Thus, anomalies can be detected by setting some threshold



- Figure source [22]

Discussion: Typical Characteristics of Anomaly

- **Most Anomalies induce a change in distributional aspects of packet header fields** (**called features**).
- **Most important features include 5-tuple: Source & destination IP addresses, Source and destination port numbers, and IP protocol.**
  - » DOS attack – multiple source IP address concentrated on a single destination IP address
  - » Network scan – dispersed distribution of destination addresses
  - » Most worms/viruses also induce some change in distribution of certain features
  - » However these changes can be very subtle and mining them is like searching for needles in a haystack
- **Unlike many previous approach, this paper aims to detect events which disturb the distribution of traffic** features **rather than traffic** volume

Limitation of Volume

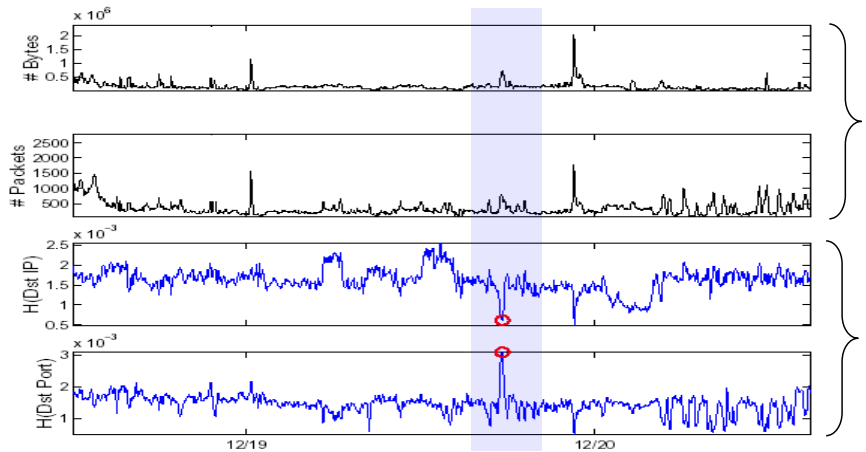
- Figure source [22]

  - Port scan anomaly (traffic feature changes, however traffic volume remains more or less the same)

We can use entropy to capture the variations in the traffic feature

$$H(X) = -\sum_{i=1}^{N} \left(\frac{n_i}{S}\right) \log_2 \left(\frac{n_i}{S}\right),$$

- Takes value 0 when distribution is maximally concentrated.
- Takes value $\log_2 N$ when distribution is maximally dispersed.



Port scan dwarfed in volume metrics…

But stands out in feature entropy, which also reveals its structure

Entropy Based versus Volume Based

- **DoS/DDoS Attacks**- a spike in traffic data toward a dominant destination IP.
- **Scan anomaly**-a spike in traffic data from a dominant source IP .
- **Flash Crowd anomaly**- again a spike in traffic data to a dominant destination IP.
- **Worm anomaly**-a Spike in traffic with a dominant port.

Anomaly Label	# Found in Volume	# Additional in Entropy
Alpha Flows	84	137
DOS	16	11
Flash Crowd	6	3
Port Scan	0	30
Network Scan	0	28
Outage Events	4	11
Point to Multipoint	0	7
Unknown	19	45
False Alarm	23	20
Total	152	292

Fraud Detection

- Domain Question: Identify fraudulent activities or *players* from observed *transaction* data
- Data
  - Transactions between different players in the system
  - Meta information about the individuals
  - An underlying graph structure
- Challenges:
  - Track and model human behavior
  - Anomalies caused by *adaptive* human adversaries
  - Massive data sizes

- Insurance (auto, health[a])
  - **Claimant**, **Provider**, Payer
- Telecommunications
  - **Customer**, Provider
- Credit Cards
  - **Customer**, **Supplier**, Bank
- Web Advertising
  - **User**, **Advertiser**, Publisher

[a] http://www.cheatingculture.com/past-cases-of-medicare-frau

A Generic Fraud Detection Method

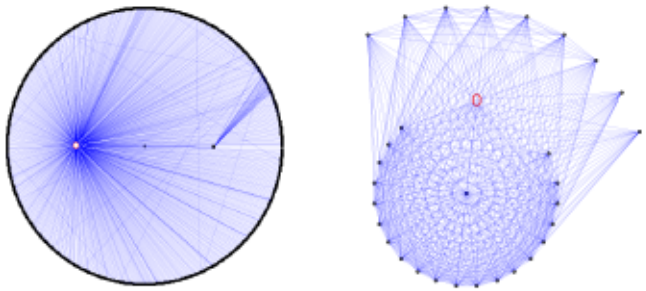
- *Activity Monitoring* [13]
  1. Build profiles for individuals (customers, users, etc.) based on historic data
     - User X makes n calls on an average in January
  2. Compare current behavior with historical profile for *significant deviations*
- *Clustering based* [4]
  1. Cluster historical profiles of customers
  2. Identify small clusters or outlying profiles as anomalies
- Strengths
  - Anomaly detection is fast (good for real time)
  - Results are easy to explain
- Weaknesses
  - Need to create and maintain a large number of profiles
  - Not *dynamic*
  - Adequate historical data might not be available
  - Too many false positives

Exploiting Graph Structure - Weighted Graphs

- Represent data as a weighted graph
  - Communication networks (phone, email, SMS)
  - Provider referral networks
- Objective: *Identify anomalous nodes*
- For each node, extract several features based on the properties of the induced sub-graph (*egonet*) of **neighboring** nodes [1]
- Choose features that can highlight anomalous nodes
  1. N_i: degree of node
  2. E_i: number of edges in egonet
  3. W_i: total weight of egonet
  4. λ_i: principal eigen vector of weighted adjacency matrix of egonet
- Data is transformed into a point in a multi-dimensional space

Identifying Anomalies

- Traditional anomaly detection (*lof*)
  - Can be slow but can identify *any* type of anomalous structure
- Faster method to identify specific types of anomalous structures:
  - Identify relevant feature pairs and power law relationship



  - E.g., **Egonet Density Power Law**: N_i vs. E_i - detect near cliques and stars

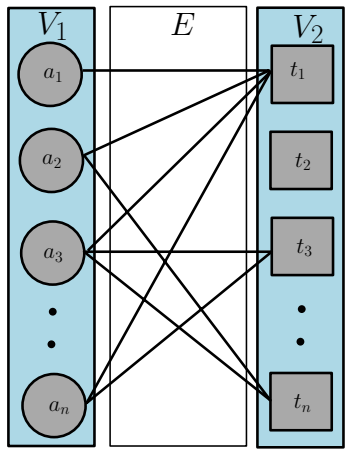
$$E_i \propto N_i^{\alpha}, 1 \leq \alpha \leq 2$$

  - Anomaly score for node i w.r.t. a pair of features ($y = Cx^{\theta}$)

$$S_i = \frac{max(y_i, Cx_i^{\theta})}{min(y_i, Cx_i^{\theta})} * \log\left(|y_i - Cx_i^{\theta}| + 1\right)$$

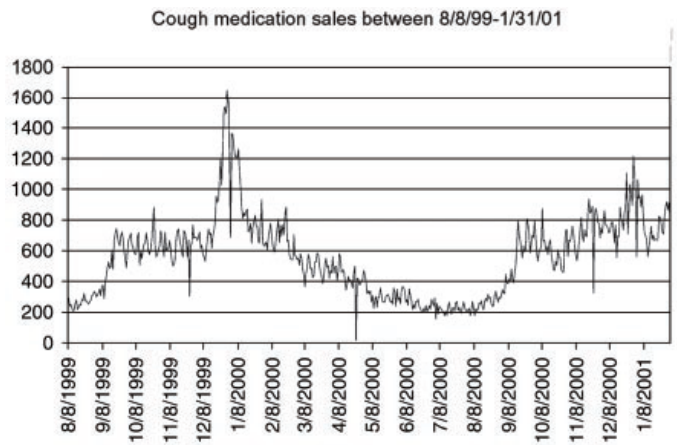
Exploiting Graph Structure - Bipartite Graphs

- Represent data as a bipartite graph
  - Healthcare Data (Beneficiaries vs. Providers)
  - Insider trading (Traders vs. Stocks)
- Objective: *Identify anomalous links*
- Given a query node $a \in V_1$ find the "relevance" of all other nodes in V_1 to a
  - $RelevanceScore(a, b) \propto$ Number of times a "random walk" from a reaches b
- Use the relevance scores to compute the *normality scores* for a node $t \in V_2$
  - Find set $S_t = \{a | \langle a, t \rangle \in E\}$
  - Compute $|S_t| \times |S_t|$ similarity matrix using relevance vectors for $a \in S_t$
  - Normality Score = mean of non-diagonal entries of similarity matrix



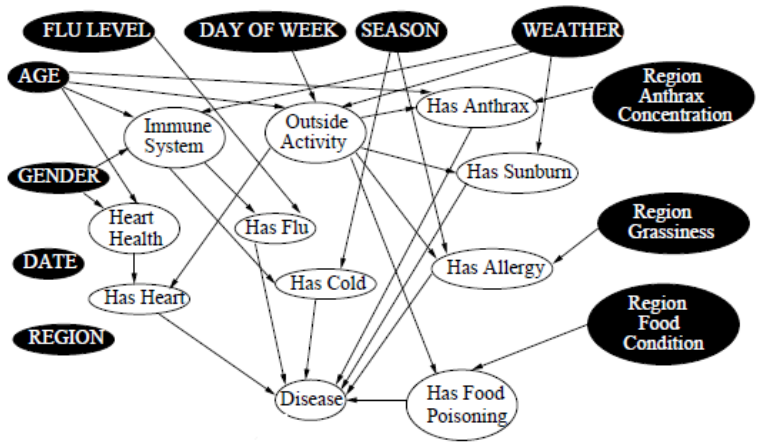
Detecting Disease Outbreaks

- Domain Question:
  - Early detection of disease outbreaks
    - Anthrax attack?
- Data:
  - Emergency department visits
  - Grocery data (*Example* [14])
  - Clinical visits
  - Weather/climate data
- Challenges:
  - *Weak* signals in the data
    - ED cases involving cough $\Rightarrow$ *Flu* or *SARS*
  - Integration of multiple signals (*lag analysis*)
  - Account for spatial and temporal correlations



What's Significant About Recent Events (WSARE)? [33]

1. Learn Bayesian network from historical data
   - *Environmental* and *response* variables



2. Sample from the BN ($DB_{baseline}$|Current Environment)
3. Compute contingency table for *rules* for $DB_{baseline}$ and $DB_{current}$
   - Rules are single assignment rules ($X_i = Y_i^j$) or conjunctions
4. Find p-value for rules using χ^2-test
   - *Null Hypothesis*: Rows and columns of tables for $DB_{baseline}$ and $DB_{current}$ are independent
5. Find rule with largest p-value. Repeat Step 2.

Incorporating Spatial and Temporal Relationships

- WSARE does not explicitly model the spatial and temporal relationships
  - What happened yesterday?
  - What happened in the adjoining neighborhood (*yesterday*)?
- **Bayesian Network Spatio-Temporal** (BNST) modeling framework [17]
- Add nodes for temporal and spatial dependencies
  - Need more data to train!!

Anomaly Detection in Climate and Weather

- Science Questions:
  - Identify natural and anthropogenic disasters.
  - Identify long time scale events - droughts, atmospheric rivers, cold fronts, etc.
- Data:
  - Ground observations, Remote sensing data (satellites, air-borne), Climate model simulation outputs
  - Multiple variables, spatio-temporal (often has height dimension as well)
- Challenges:
  - Model spatio-temporal relationships across multiple variables
  - Explain the cause of anomalies
  - Massive data sizes

Climate and weather extreme events are well defined

Key challenge is to find significant events and explain the cause.

Anomalies are Widely Used in Climate!

- Most analysis done on "anomaly" time series
- Difference from a "base period" (Too simplistic?)
- Brings spatial smoothness (e.g., a mountain top and nearby valley can have very different temperatures), Removes seasonality
- Understand climate and weather phenomenon
- *Southern Oscillation Index (SOI)*
  - Difference between Sea Level Pressure (SLP) anomalies for Tahiti and Darwin, Australia.

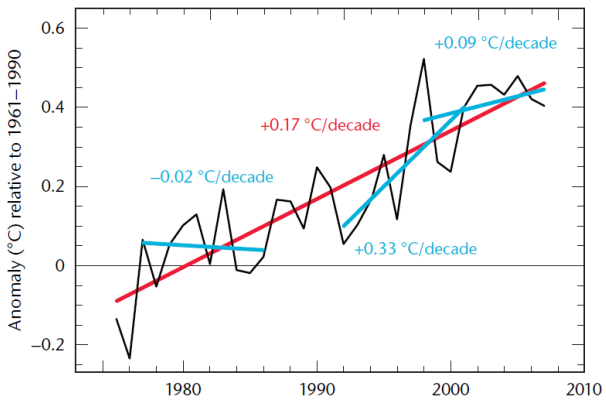


Figure: Global Average Temperature Anomaly (1975 - 2007) Src: www.metoffice.gov.uk

Constructing Anomalies from Raw Data

- Anomaly time series for a given location, i:

$$v_i' = v_i - b_i$$

  where b_i is the base (reference).
- How to choose b_i?
  - Mean of all data for location i
  - Monthly mean values (account for seasonality)
  - Monthly z-score values
  - Median (more robust)
  - Using a shorter "reference period"
    - 30 year moving window
- Different methods show statistically significant differences[19]
  - *What is the right strategy?*
  - Weighted mean of different strategies (Pick weights using Monte Carlo sampling)

Anomaly Detection for Identifying Droughts

- Science question: Identify significant drought patterns using historical observation data or future simulation data or both
- Find persistent spatiotemporal anomalies in precipitation data

- A two step approach:
  1. Find precipitation anomalies using thresholds
  2. Find large **connected components** across space and time
     - Matlab - **bwlabel**, **bwlabeln**
- Followup Science question: Explain cause?

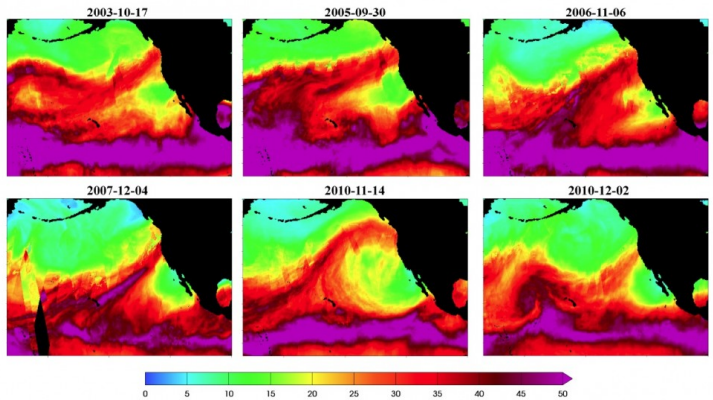
(drought190120061.avi)

Figure: Video courtsey Dr. Arindam Banerjee

More Climate Extreme Events using Anomaly Detection

Atmospheric Rivers [8]

- Water Vapor Content
- Anomalies using a threshold
- Connected components
- *Example: Src - http://newscenter.lbl.gov*



Cold Fronts [23]

- Surface winds and Potential temperature fields
- Methodology:
  1. Compute features for every grid
  2. Cluster grids into K clusters
  3. Label clusters as anomalous or not using thresholds
  4. Filter out false positives using domain knowledge

Validation is Key!!

- How useful are the anomalies from the domain perspective?
- Common pitfalls:
  - Anomalies are algorithmically *correct* but are not relevant (bad data, noise, simplistic)
  - Anomalies are not *actionable*
    - Not identified in timely fashion
    - Resolution is not fine enough
    - Cause not explained
  - Anomalies *lost* among *false positives*
- Solution?
  - **Good validation data** during design
  - Clear definition of a *domain anomaly* and distinction from other potential *competitors*


L. Akoglu, M. McGlohon, and C. Faloutsos.
OddBall: Spotting Anomalies in Weighted Graphs.
In *In Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD)*, June 2010.


A. Asuncion and D. J. Newman.
UCI machine learning repository.
[http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, 2007.


S. D. Bay and M. Schwabacher.
Mining distance-based outliers in near linear time with randomization and a simple pruning rule.
In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 29–38. ACM Press, 2003.


R. Bolton and D. Hand.
Unsupervised profiling methods for fraud detection.
In *Credit Scoring and Credit Control VII*, 1999.


S. Boriah, V. Chandola, and V. Kumar.
Similarity measures for categorical data: A comparative evaluation.
In *SDM*, pages 243–254, 2008.


M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander.
Lof: identifying density-based local outliers.
In *Proceedings of 2000 ACM SIGMOD International Conference on Management of Data*, pages 93–104. ACM Press, 2000.


S. Budalakoti, A. Srivastava, and M. Otey.
Anomaly detection and diagnosis algorithms for discrete symbol sequences with applications to airline safety.
Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, 37(6), 2007.


S. Byna, Prabhat, M. Wehner, and K. Wu.
Detecting atmospheric rivers in large climate datasets.

In *2nd International Workshop on Petascale Data Analytics: Challenges, and Opportunities (PDAC-11)*, 2011.

V. Chandola, A. Banerjee, and V. Kumar.

Anomaly detection for discrete sequences: A survey.
IEEE Transactions on Knowledge and Data Engineering, 99(PrePrints), 2010.

V. Chandola, S. Boriah, and V. Kumar.

A framework for exploring categorical data.
In *Proceedings of the ninth SIAM International Conference on Data Mining*, 2009.

V. Chandola, V. Mithal, and V. Kumar.

A comparative evaluation of anomaly detection techniques for sequence data.
In *Proceedings of International Conference on Data Mining*, 2008.

V. Chandola, V. Mithal, and V. Kumar.

Understanding anomaly detection techniques for sequence data.
Technical Report 09-001, University of Minnesota, Computer Science Department, January 2009.

T. Fawcett and F. Provost.

Activity monitoring: noticing interesting changes in behavior.
In *Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 53–62. ACM Press, 1999.

A. Goldenberg, G. Shmueli, R. A. Caruana, and S. E. Fienberg.

Early statistical detection of anthrax outbreaks by tracking over-the-counter medication sales.
Proceedings of the National Academy of Sciences of the United States of America, 99(8):pp. 5237–5240, 2002.

D. Hawkins.

Identification of outliers.
Monographs on Applied Probability and Statistics, May 1980.

S. A. Hofmeyr, S. Forrest, and A. Somayaji.

Intrusion detection using sequences of system calls.
Journal of Computer Security, 6(3):151–180, 1998.

X. Jiang and G. F. Cooper.

A bayesian spatio-temporal method for disease outbreak detection.
JAMIA, pages 462–471, 2010.

J. S. Kaustav Das.

Detecting anomalous records in categorical datasets.
In *Proc. of the thirteenth ACM SIGKDD international conference on Knowledge discovery and data mining*, Aug 2007.

J. Kawale, S. Chatterjee, A. Kumar, S. Liess, M. Steinbach, and V. Kumar.

Anomaly construction in climate data: Issues and challenges.
In *Proceedings of NASA Conference on Intelligent Data Understanding*, 2011.

E. M. Knorr and R. T. Ng.

Algorithms for mining distance-based outliers in large datasets.
In *Proceedings of the 24rd International Conference on Very Large Data Bases*, pages 392–403. Morgan Kaufmann Publishers Inc., 1998.

H.-P. Kriegel, P. Kröger, E. Schubert, and A. Zimek.

Interpreting and unifying outlier scores.
In *SDM*, pages 13–24, 2011.

A. Lakhina, M. Crovella, and C. Diot.

Diagnosing network-wide traffic anomalies.
In *Proceedings of ACM SIGCOMM*, pages 219–230, 2004.

X. Li, R. Ramachandran, S. Graves, S. Movva, B. Akkiraju, D. Emmitt, S. Greco, R. Atlas, J. Terry, and J.-C. Jusem.

Automated detection of frontal systems from numerical model-generated data.
In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, KDD '05, pages 782–787, New York, NY, USA, 2005. ACM.


M. V. Mahoney and P. K. Chan.
Learning rules for anomaly detection of hostile network traffic.
In *ICDM*, pages 601–604, 2003.


S. V. Mane.
False negative estimation: theory, techniques and applications.
ProQuest, UMI Dissertation Publishing, 2011.


A. Moore and M. S. Lee.
Cached sufficient statistics for efficient machine learning with large datasets.
J. Artif. Int. Res., 8:67–91, March 1998.


Y. Pei and O. Zaane.
A synthetic data generator for clustering and outlier analysis.
Technical report, University of Alberta, 2006.


S. Ramaswamy, R. Rastogi, and K. Shim.
Efficient algorithms for mining outliers from large data sets.
In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pages 427–438. ACM Press, 2000.


N. Satish, M. Harris, and M. Garland.
Designing efficient sorting algorithms for manycore gpus.
In *IPDPS*, pages 1–10, 2009.


X. Song, M. Wu, C. Jermaine, and S. Ranka.
Conditional anomaly detection.
IEEE Trans. on Knowl. and Data Eng., 19:631–645, May 2007.


D. A. Spielman and N. Srivastava.
Graph sparsification by effective resistances.
In *Proceedings of the 40th annual ACM symposium on Theory of computing*, STOC '08, pages 563–568, New York, NY, USA, 2008. ACM.

P. Sun, S. Chawla, and B. Arunasalam.

Mining for outliers in sequential databases.

In *In SIAM International Conference on Data Mining*, 2006.

W.-K. Wong, A. Moore, G. Cooper, and M. Wagner.

Bayesian network anomaly pattern detection for disease outbreaks.

In T. Fawcett and N. Mishra, editors, *Proceedings of the Twentieth International Conference on Machine Learning*, pages 808–815, Menlo Park, California, August 2003. AAAI Press.

L. Xiong, B. Poczos, and J. Schneider.

Group anomaly detection using flexible genre models.

In *NIPS*, 2011.

Acknowledgements

- Linsey Pang, Tara Babie and Khoa Nguyen (University of Sydney)
- Arindam Banerjee (University of Minnesota)