ECS 607/766 Data Mining
Lecture 8 – Anomaly Detection & Association Rules

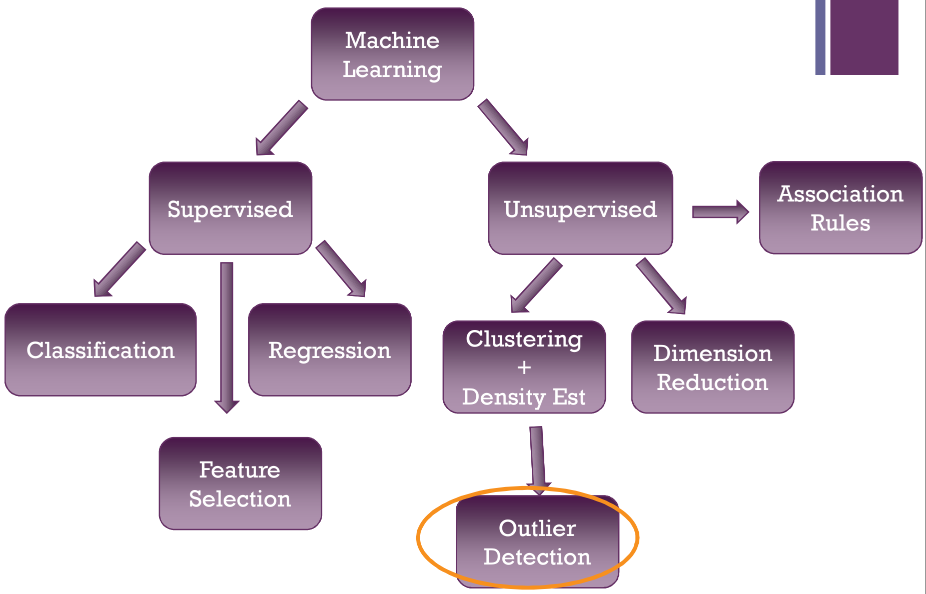
Dr. Ioannis Patras
EECS, Queen Mary University of London

Slide thanks: Tim Hospedales

Overview

- Anomaly Detection
  - Univariate Gaussian
  - Multivariate Gaussian
  - Complex Data
  - Evaluation
  - Contrast to Supervised Learning
- Association Rule Mining
  - Apriori

+ A Taxonomy



+ Anomaly Detection: Motivation

- Previously we looked for:
  - Related columns (dimensionality reduction)
  - Related rows (clustering)
- Sometimes you are interested in finding unusual items
  - "Anomalies", "Outliers"
- …As a pre-processing step
  - E.g., algorithms that use Sum-Squared objectives are not robust to outliers. Use outlier detection first to find and discard such rows
- …As an end goal.
  - The aim of the data mining exercise is anomaly detection

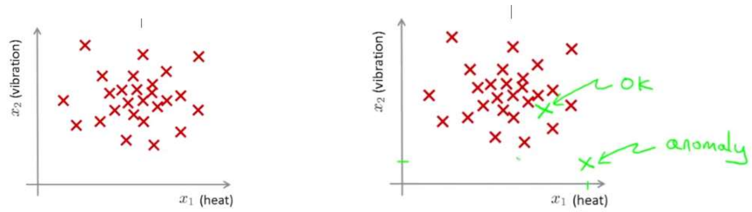
Anomaly Detection: Motivation

Applications

- Fraud detection
- Cyber-security
- Machine / Factory maintenance
- Data-center maintenance
- Process-control
- Security and surveillance
- Anti-terrorism

Anomaly Detection: Example

- Example: Manufacturing Quality Control: Aircraft Engines
  - x_1 = heat, x_2 = vibration.



- Can you imagine an algorithm?
  - Input: x_1,x_2
  - Output: Anomaly or Not Anomaly?

Anomaly Detection: Probability Recap I

Two common probability distributions

- Categorical variables: Multinomial
  - x is 1-of-N encoding. u from 0 to 1. u's sum to 1.

$$p(\mathbf{x};\mathbf{u}) = \prod u_i^{x_i} \quad \Rightarrow \quad u_k = \frac{\sum_i x_{ik}}{\sum_i \sum_k x_{ik}} = \frac{N_k}{N}$$

- Continuous variables: Gaussian
  - x is real vector. u is a real vector. S is a matrix.

$$p(\mathbf{x};\mathbf{u},S) = \frac{1}{Z}\exp\left(-\frac{1}{2}(\mathbf{x}-\mathbf{u})^T S^{-1}(\mathbf{x}-\mathbf{u})\right) \quad \Rightarrow \quad \mathbf{u} = \frac{1}{N}\sum_i \mathbf{x}_i \quad S = \frac{1}{N}\sum_i (\mathbf{x}-\mathbf{u})(\mathbf{x}-\mathbf{u})^T$$

Fitting Example: Multinomial

$$u_k = \frac{\sum_i x_{ik}}{\sum_i \sum_k x_{ik}} = \frac{N_k}{N}$$

- Fitting multinomial parameters
  - Suppose throw coin: H, T, T, H, H
  - Suppose dice role: 2, 1, 3, 5, 5, 4, 6, 1

Throw	Head	Tail
x_1	1	0
x_2	0	1
x_3	0	1
x_4	1	0
x_5	1	0
u_k	3/(3+2) =3/5	2/(3+2) =2/5

Roll	1	2	3	4	5	6
x_1	0	1	0	0	0	0
x_2	1	0	0	0	0	0
x_3	0	0	1	0	0	0
x_4	0	0	0	0	1	0
x_5	0	0	0	0	1	0
x_6	0	0	0	1	0	0
x_7	0	0	0	0	0	1
x_8	1	0	0	0	0	0
u_k	2/8	1/8	1/8	1/8	2/8	1/8

Fitting Example: Gaussian

- Fitting Gaussian parameters
  - Suppose fish length = [1, 3, 5, 3, 2]
- Mean, u = (1+3+6+3+2)/5 = 3
- Variance, v = $((1-3)^2+(3-3)^2+(6-3)^2+(3-3)^2+(2-3)^2)/5=14/5$

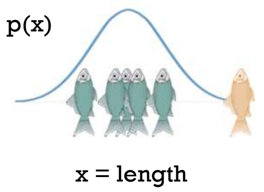
$$\mathbf{u} = \frac{1}{N}\sum_i \mathbf{x}_i \quad S = \frac{1}{N}\sum_i (\mathbf{x}-\mathbf{u})(\mathbf{x}-\mathbf{u})^T$$

Anomaly Detection: Probability Recap II

- Continuous variables: Gaussian
  - x is real vector. u is a real vector. S is a matrix.

$$p(\mathbf{x};\mathbf{u},S) = \frac{1}{Z}\exp\left(-\frac{1}{2}(\mathbf{x}-\mathbf{u})^T S^{-1}(\mathbf{x}-\mathbf{u})\right) \Rightarrow \mathbf{u} = \frac{1}{N}\sum_i \mathbf{x}_i \quad S = \frac{1}{N}\sum_i (\mathbf{x}-\mathbf{u})(\mathbf{x}-\mathbf{u})^T$$

- Tells us:
  - For a specified Gaussian:
    - What data do we expect?
    - How likely is any particular piece of data?
  - For a specified Dataset:
    - What Gaussian best explains it?
- Algorithm:
  - Anomaly if p(x)<T



Anomaly Detection Algorithm

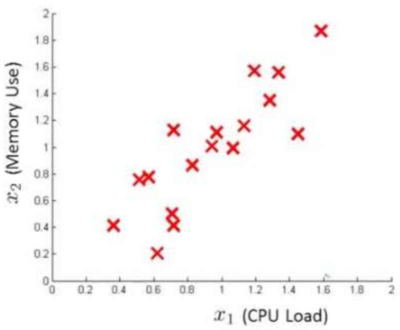
- Algorithm:
  - Read in normal training data, {x}
  - Compute the Gaussian (u,S) that best explains the data {x}

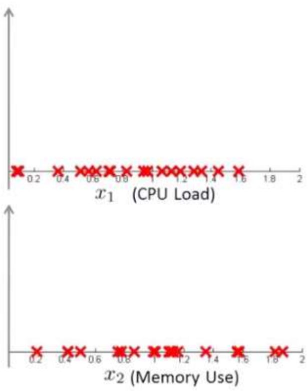
$$\mathbf{u} = \frac{1}{N}\sum_i \mathbf{x}_i \quad S = \frac{1}{N}\sum_i (\mathbf{x}-\mathbf{u})(\mathbf{x}-\mathbf{u})^T$$

  - Given a new example x and estimated u,S, compute p(x)
  - If p(x)<T
    - Then Anomaly
  - Else
    - Ok

$$p(\mathbf{x};\mathbf{u},S) = \frac{1}{Z}\exp\left(-\frac{1}{2}(\mathbf{x}-\mathbf{u})^T S^{-1}(\mathbf{x}-\mathbf{u})\right)$$

Anomaly Detection Example: Data Center Monitoring

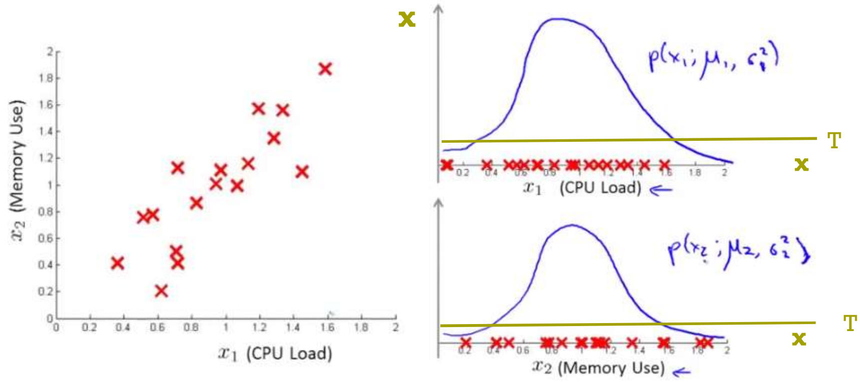




+
Anomaly Detection Example: Data Center: Univariate Gaussians

Algorithm:

Anomaly if p(x)<T

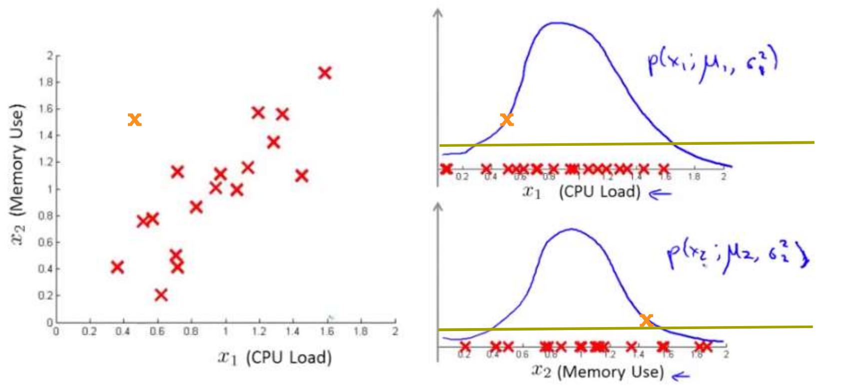


+
Anomaly Detection Example: Data Center: Univariate Gaussians

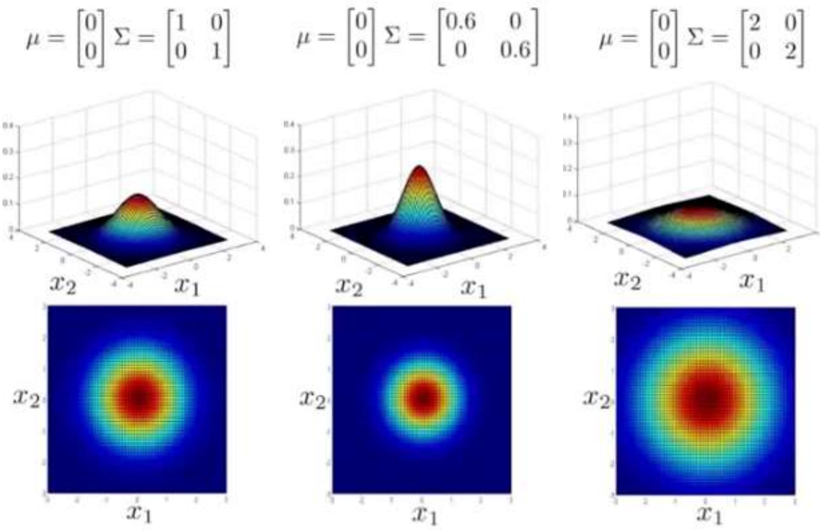
Algorithm:

Anomaly if p(x)<T

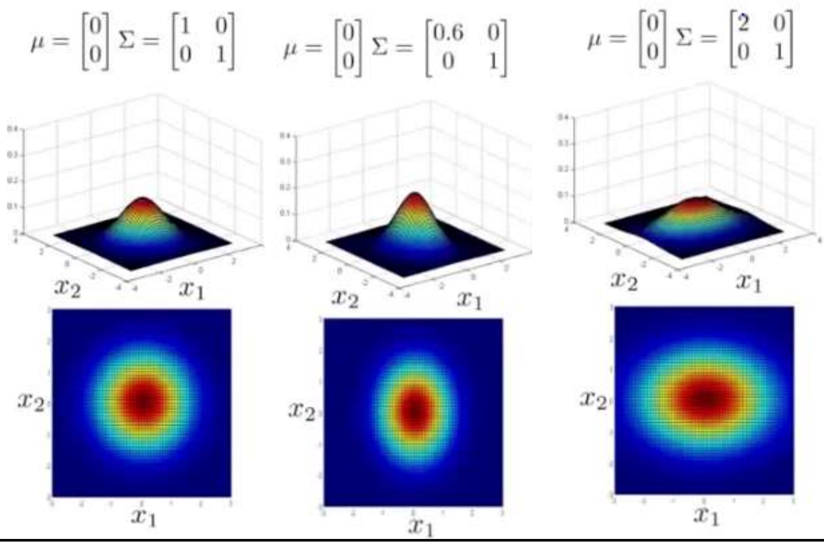
No anomaly!
False negative!



+ Multivariate Gaussians: Intuition

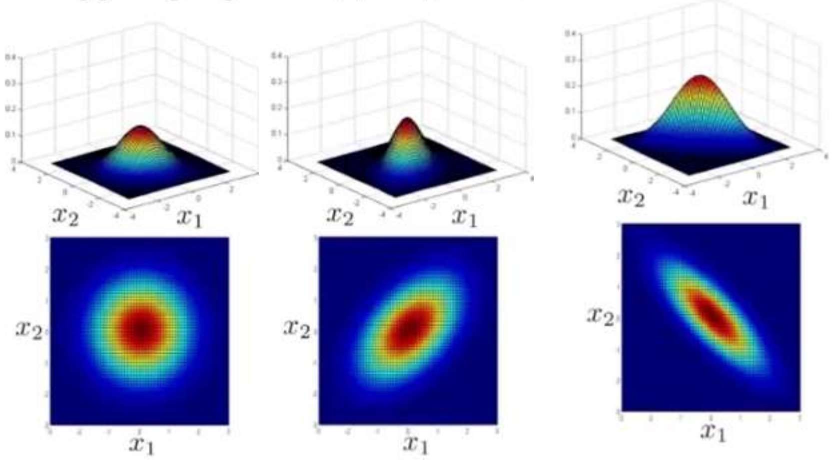


+ Multivariate Gaussians: Intuition



Multivariate Gaussians: Intuition

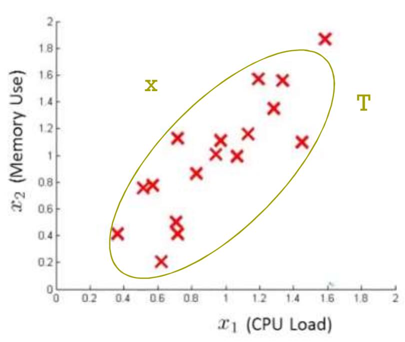
$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \; \Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ $\quad$ $\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \; \Sigma = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}$ $\quad$ $\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \; \Sigma = \begin{bmatrix} 1 & -0.8 \\ -0.8 & 1 \end{bmatrix}$

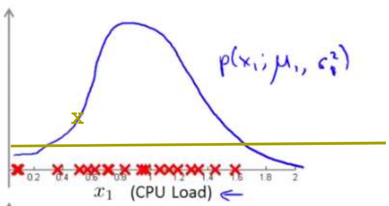


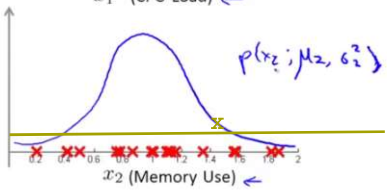
Anomaly Detection Example: Data Center: Multivariate

Algorithm:

Anomaly if p(x)<T

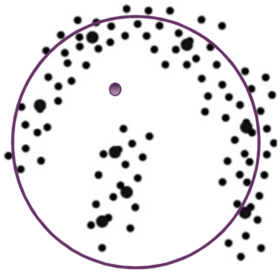


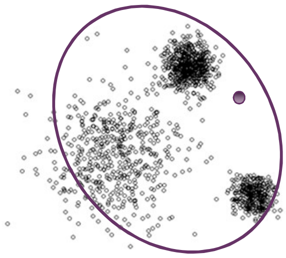




Anomaly Detection:
What if the data is more complicated than a multivariate Gaussian?

- Some data cannot be well modeled by a multivariate Gaussian.
  - How to detect anomalies here?
  - …Ideas?

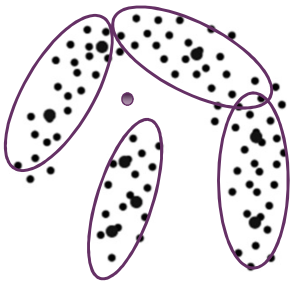


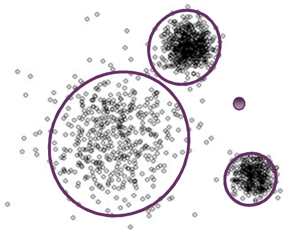


Anomaly Detection:
What if the data is more complicated than a multivariate Gaussian?

- Some data cannot be well modeled by a multivariate Gaussian.
  - => Model with Gaussian Mixture model

$$p(\mathbf{x}) \propto \exp\left(-\frac{1}{2}(\mathbf{x}-\mathbf{u})^T S^{-1}(\mathbf{x}-\mathbf{u})\right) \Rightarrow p(\mathbf{x}) \propto \sum_k \pi_k \exp\left(-\frac{1}{2}(\mathbf{x}-\mathbf{u}_k)^T S_k^{-1}(\mathbf{x}-\mathbf{u}_k)\right)$$





+ How to Evaluate?

- Input:
  - 10,000 normal aircraft engines (Features, x)
  - 20 broken/flawed ones (Label, y)
- Split into half train and test
- Compute {u,S} for model p(x) on train $\{x_1,..,x_{5000}\}$
- On a test set, predict anomaly if p(x)<T
- Evaluation metric
  - TP/FP/FN/TN rate
  - ROC Curve
  - Precision/Recall
- How to choose T?
  - Cross-validation!

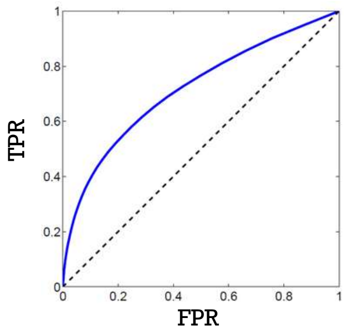
+ How to Evaluate?

Predict anomaly if p(x)<T

Evaluation metrics:

- TP/FP/FN/TN rate
- ROC Curve
  - Considers a variety of thresholds

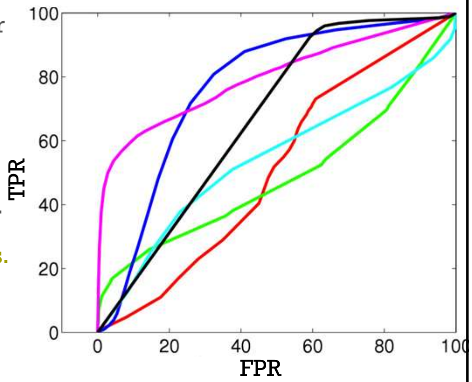
prediction outcome	actual value: *p*	actual value: *n*	total
p'	True Positive	False Positive	P'
n'	False Negative	True Negative	N'
total	P	N	



Understanding ROC curves

- Suppose you have a set of models to choose from (colored lines)…
  - Scenario 1: Early-stage financial fraud filtering. Potentially anomalous transactions flagged for an analyst to examine.
  - Scenario 2: Late stage flagging of potential low-quality items on a production line. Flagging an anomaly means stopping the factory production line for an hour.

S1: Expensive to miss an anomaly.
False alarms not that costly.
⇒ Prioritize high TPR
⇒ Pick Blue/Black systems

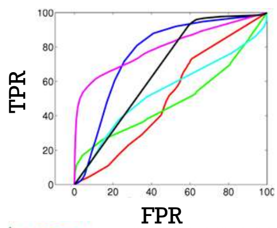


S2: Not critical to catch all anomalies.
False alarms very costly.
⇒ Prioritize low FPR
⇒ Pick Magenta system

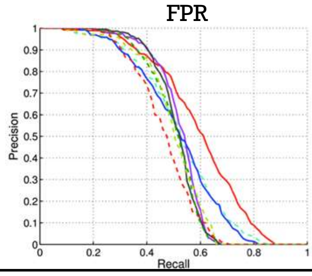
How to Evaluate?

		actual value p	actual value n	total
prediction outcome	p'	True Positive	False Positive	P'
prediction outcome	n'	False Negative	True Negative	N'
total		P	N	

- ROC Curve
  - TPR=Recall=TP/P=TP/(TP+FN)
    - "What fraction of true anomalies did we successfully detect?"
  - FPR=FP/N=FP/(FP+TN)
    - "What fraction of normal data did we erroneously call anomalous?"



- Precision/Recall
  - Precision = TP/(TP+FP)
    - "How many of our predicted anomalies are true anomalies?"
  - Recall = TP/P = TP/(TP+FN)
    - "What fraction of true anomalies did we successfully detect?"



Anomaly Detection Versus Supervised Learning

Anomaly Detection	Supervised Learning
Small number of positive examples (e.g., 0-10)	Large number of positive and negative examples (E.g., 100s)
Many negative examples	
Many different types of anomalies. Hard to model what they look like.	Enough positive examples to model their typical characteristics.
Future anomalies may look nothing like previous ones	Future positive examples likely to be similar to train set ones.

Anomaly Detection Versus Supervised Learning: Quiz

Task	Method to Use?
Fraud Detection	Anomaly
Spam classification	Supervised
Weather prediction	Supervised
Manufacturing defects	Anomaly
Data center crash prediction	Anomaly
Surveillance crime detection	Anomaly
Disease classification	Supervised

+ # Example Application: EECS Work ☺

After unsupervised learning, we can..
infer behaviour class and
screen for unusual events.

Hospedales et al, IEEE ICCV 2009, IJCV 2011

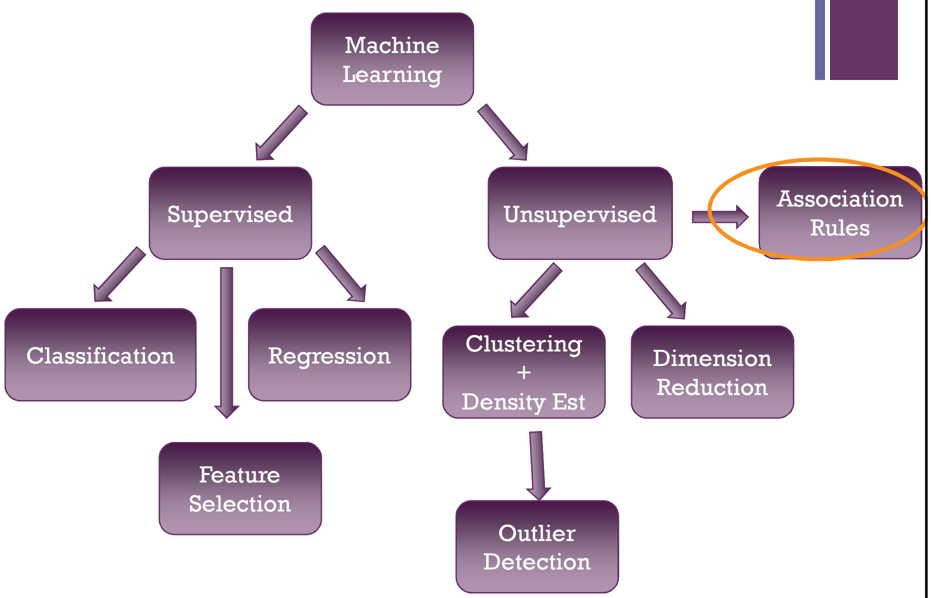
+ # Alternative Application of Anomaly Detection: Scientific Data Exploration

- Apply anomaly detection to large scale data streams: flags new items of interest to focus limited manual inspection time.
  - E.g., telescope images => new astronomical phenomena.
  - E.g., radio telescope images => focus SETI search.
  - E.g., particle accelerator readings => focus new particle search

Overview

- Anomaly / Outlier Detection
- Association Rule Mining
  - Apriori

A Taxonomy



+
Association Mining

- What is association mining?
  - Find frequent patterns, associations, correlations or causations among items or objects in transaction or relational databases.
- Example: Shopping baskets and shopping rules

TID	Items
1	Bread, Peanuts, Milk, Fruit, Jam
2	Bread, Jam, Soda, Chips, Milk, Fruit
3	Steak, Jam, Soda, Chips, Bread
4	Jam, Soda, Peanuts, Milk, Fruit
5	Jam, Soda, Chips, Milk, Bread
6	Fruit, Soda, Chips, Milk
7	Fruit, Soda, Peanuts, Milk
8	Fruit, Peanuts, Cheese, Yogurt

Examples
$\{bread\} \Rightarrow \{milk\}$
$\{soda\} \Rightarrow \{chips\}$
$\{bread\} \Rightarrow \{jam\}$

+
Association Mining

TID	Items
1	Bread, Peanuts, Milk, Fruit, Jam
2	Bread, Jam, Soda, Chips, Milk, Fruit
3	Steak, Jam, Soda, Chips, Bread
4	Jam, Soda, Peanuts, Milk, Fruit
5	Jam, Soda, Chips, Milk, Bread
6	Fruit, Soda, Chips, Milk
7	Fruit, Soda, Peanuts, Milk
8	Fruit, Peanuts, Cheese, Yogurt

Examples
$\{bread\} \Rightarrow \{milk\}$
$\{soda\} \Rightarrow \{chips\}$
$\{bread\} \Rightarrow \{jam\}$

- Given a set of transactions, find rules that will predict the occurrence of an item based on occurrences of other items in the transaction

Association Mining

Why do it?

- Shopping Baskets
  - Marketing
  - Inform store layout
  - Catalog design
  - Recommendations
- Search: Which keywords occur together?
  - => Query Expansion.
- Medicine: Find typical disease symptoms
- Understand your database

Original Paper (Agrawal '93) is one of top most cited papers in all of computer science

TID	Items
1	Bread, Peanuts, Milk, Fruit, Jam
2	Bread, Jam, Soda, Chips, Milk, Fruit
3	Steak, Jam, Soda, Chips, Bread
4	Jam, Soda, Peanuts, Milk, Fruit
5	Jam, Soda, Chips, Milk, Bread
6	Fruit, Soda, Chips, Milk
7	Fruit, Soda, Peanuts, Milk
8	Fruit, Peanuts, Cheese, Yogurt

Examples
$\{bread\} \Rightarrow \{milk\}$
$\{soda\} \Rightarrow \{chips\}$
$\{bread\} \Rightarrow \{jam\}$

su.edu/stats/articles

Most Cited Computer Science Articles

This list is generated from documents in the CiteSeer^x database as of May 20, 2012. This list is automatically generated and may contain errors. The list is generated in batch mode and citation counts may differ from those currently in the CiteSeer^x database, since the database is continuously updated.

All Years | 1990 | 1991 | 1992 | 1993 | 1994 | 1995 | 1996 | 1997 | 1998 | 1999 | 2000 | 2001 | 2002 | 2003 | 2004 | 2005 | 2006 | 2007 | 2008 | 2009 | 2010 | 2011 | 2012

1. A P Dempster, N M Laird, D B Rubin.
   Maximum likelihood from incomplete data via the em algorithm. Journal of Royal Statistical Society, Series B, 1977
   5972
2. C A R Hoare.
   Communicating sequential processes. ISSN 0001-0782. URL http://doi.acm.org/10.1145/359576, 1978
   3056
3. L Rabiner.
   A tutorial on hidden Markov models and selected applications inspeech recognition. Proceedings of the IEEE, 1989
   3014
4. I Stoica, R Morris, D Liben-Nowell, D Karger, M F Kaashoek, F Dabek, H Balakrishnan.
   Chord: A scalable peer-to-peer lookup service for Internet applications. In SIGCOMM, 2001
   2971
5. J R QUINLAN.
   Induction of Decision Trees. 0
   2845
6. D G Lowe.
   Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vision, 0
   2804
7. S Kirkpatrick, C D Gelatt, M P Vecchi.
   Optimization by simulated annealing. Science, 1983
   2741
8. R S Sutton, A G Barto.
   Reinforcement Learning: An Introduction. 1998
   2674
9. Randal E Bryant.
   Graph-based algorithms for boolean function manipulation. IEEE Transactions on Computers, 1986
   2577
10. C L Liu, J W Layland.
    Scheduling algorithms for multiprogramming in a hard real-time environment. Journal of the Association for Computing Machinery, 1973
    2515
11. S Brin, L Page.
    The anatomy of a large-scale hypertextual web search engine. in: Proceedings of the 7th World Wide Web Conference, 1998
    2453
12. R L Rivest, A Shamir, L Adleman.
    A method for obtaining digital signatures and public-key cryptosystems. Commun. ACM, 1978
    2440
13. M Kass, A Witkin, D Terzopoulos.
    Snakes: Active contour models. International Journal of Computer Vision, 1988
    2391
14. S Ratnasamy, P Francis, M Handley, R Karp, S Shenker.
    A scalable content-addressable network. SIGCOMM Computer Communication Review, 0
    2304
15. W Diffie, M Hellman.
    New directions in cryptography. IEEE Trans. Inform. Theory, 1976
    2230
16. M Turk, A Pentland.
    Eigenfaces for recognition. Journal of Cognitive Neuroscience, 1991
    2187
17. J M Kleinberg.
    Authoritative sources in a hyperlinked environment. Journal of the ACM, 1999
    2152
18. R Agrawal, R Srikant.
    Fast Algorithm for Mining Association Rules. Proc. 20th Int'l Conf. Very Large Data Bases (VLDB '94, 1994
    2096
19. S Deerwester, S Dumais, T Landauer, G Furnas, R Harshman.
    Indexing by latent semantic analysis. Journal of the American Society of Information Science, 1990

+ Association Mining: Definitions

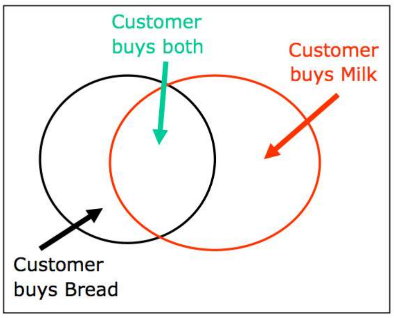
- Itemset:
  - A collection of one or more items
  - E.g., {milk, bread jam}
- Association Rule
  - X=>Y implication
- Count: N(X). Occurrences of itemset X
- Support of Itemset S(X)
  - % transactions contain itemset: N(X)/N
  - S({milk,bread})=3/8, ({soda, chips})=4/8
- Support of Rule S(X=>Y)
  - % transactions with X+Y: N(X+Y)/N
- Confidence of Rule: C(X=>Y)
  - N (X+Y)/N (X)

- Challenge:
  - Find all rules X=>Y where:
    - S(X=>Y) > MinSup
    - C(X=>Y) > MinConf
  - I.e., both (frequent and confident)

TID	Items
1	Bread, Peanuts, Milk, Fruit, Jam
2	Bread, Jam, Soda, Chips, Milk, Fruit
3	Steak, Jam, Soda, Chips, Bread
4	Jam, Soda, Peanuts, Milk, Fruit
5	Jam, Soda, Chips, Milk, Bread
6	Fruit, Soda, Chips, Milk
7	Fruit, Soda, Peanuts, Milk
8	Fruit, Peanuts, Cheese, Yogurt

+ Support and Confidence

- Support of Rule S(X=>Y)
  - % transactions with X+Y: N(X+Y)/N
  - Intersection over total area.
- Confidence of Rule: C(X=>Y)
  - N (X+Y)/N (X)
  - Intersection over X



Association Mining: Example

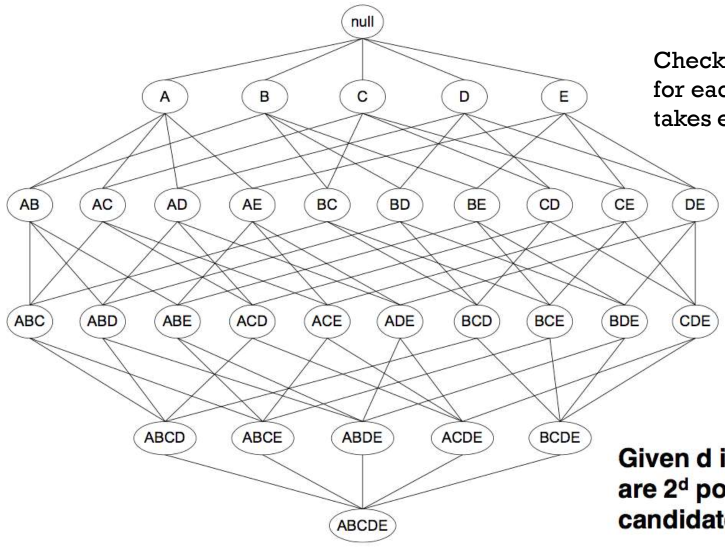
- Example:
  1. {shoes, socks, tshirt}
  2. {socks, sweater, pants}
  3. {tshirt, pants, socks}
  4. {shoes, socks}
- MinSup = 2/4, Minconf = 2/3
  - {shoes=>socks}
    - Sup = ½ , conf=2/2
  - {socks=>shoes}
    - Sup = ½ , conf= 2/4
  - {sweater}
    - Sup = 1/4

- Challenge
  - Find all rules X=>Y where:
    - S(X=>Y) > MinSup
    - C(X=>Y) > MinConf
  - (frequent and confident)
- Support of Itemset s(X)
  - % transactions contain itemset: N(X)/N
- Support of Rule S(X=>Y)
  - % transactions with X+Y: N(X+Y)/N
- Confidence of Rule: C(X=>Y)
  - Given you saw X, how confident about Y?
  - N (X+Y)/N (X)

Association Mining: Algorithm?

- Algorithm Sketch
  - List all possible rules
  - Check the support and confidence for each
  - Keep the ones above MinSup and MinConf
- Algorithm Sketch
  - List all possible itemsets
  - Count their support
  - Count their confidence
  - Generate rules above MinSup & MinConf
- => Cost $O(2^N)$ ☹

Frequent Itemset Generation is Costly



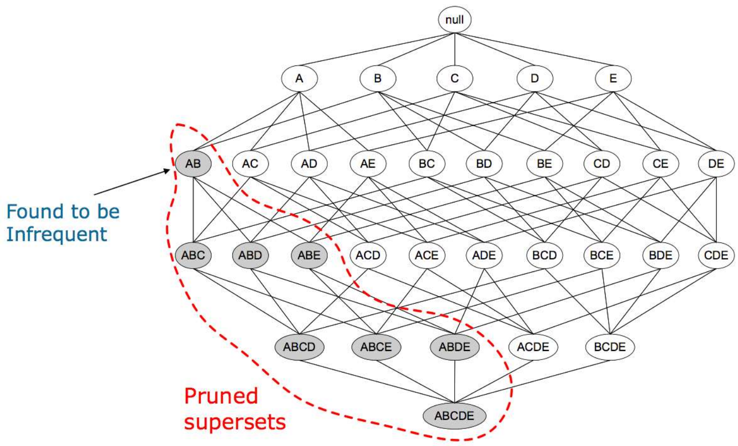
Checking support for each candidate takes even more

Given d items, there are 2^d possible candidate itemsets

Classic Solution: "Apriori Algorithm"

- Apriori Observation
  - Any subset of a frequent itemset must also be frequent.
  - Any superset of an infrequent itemset must also be infrequent
- Apriori Algorithm
  - Generate frequent itemsets bottom up. Don't bother to consider those which have a subset known to be infrequent.

+ Illustration of Apriori Principle



+ Apriori Algorithm

- K=1.
- Generate frequent itemsets of length k=1.
- Repeat until no new frequent itemsets
  - Generate K+1 candidate itemsets from K sized frequent itemsets
  - Count the support of each candidate by scanning the DB
  - Eliminate infrequent candidates
- Check confidence for frequent itemsets

Illustration

TID	Items
1	Bread, Peanuts, Milk, Fruit, Jam
2	Bread, Jam, Soda, Chips, Milk, Fruit
3	Steak, Jam, Soda, Chips, Bread
4	Jam, Soda, Peanuts, Milk, Fruit
5	Jam, Soda, Chips, Milk, Bread
6	Fruit, Soda, Chips, Milk
7	Fruit, Soda, Peanuts, Milk
8	Fruit, Peanuts, Cheese, Yogurt

Items (1-itemsets)

Item	Count
Bread	4
Peanuts	4
Milk	6
Fruit	6
Jam	5
Soda	6
Chips	4
Steak	1
Cheese	1
Yogurt	1

Minimum Support = 4

2-itemsets

2-Itemset	Count
Bread, Jam	4
Peanuts, Fruit	4
Milk, Fruit	5
Milk, Jam	4
Milk, Soda	5
Fruit, Soda	4
Jam, Soda	4
Soda, Chips	4

3-itemsets

3-Itemset	Count
Milk, Fruit, Soda	4

Aprori Details

- If MinSup is high, we could miss itemsets involving interesting rare items (E.g., expensive).
- If MinSup is low, computation is prohibitive
- Cost:
  - Depends on MinSup
  - Depends on Data

Summary: You Should Know

- Anomalies
  - What is the anomaly detection problem?
  - Sketch an anomaly detection algorithm
  - Applications of anomaly detection
  - When anomaly detection versus supervised learning is appropriate
  - Connection to clustering with Gaussian Mixtures
- Itemset Mining
  - What is the itemset mining problem?
  - Sketch the Apriori algorithm
  - Applications of itemset mining