Honeywell

Data Mining for Diagnosis

Gautam Biswas
Daniel L.C. Mack
Vanderbilt University, Nashville, TN
Dinkar Mylaraswamy
Honeywell Aerospace, Minneapolis, MN

Tutorial Presented at PHM 2011, Montreal, Canada

Acknowledge: NASA NRA NASA NNL09AD44T; Aviation Safety Program

Who are we?

- Gautam Biswas
  - Professor of Computer Science and Engineering, Vanderbilt University
  - Research Interests: Modeling and Analysis of Complex Systems, Fault Detection and Isolation, Intelligent Systems, Data Mining
- Daniel L.C. Mack
  - Graduate Research Assistant, Computer Science, Vanderbilt University
  - Research Interests: Generative Machine Learning, Anomaly Detection, Network Theory, Modeling and Analysis of Complex Systems
- Dinkar Mylaraswamy
  - Technology Fellow, Honeywell
  - Research Interests: Condition-based management, Analysis of Complex Systems

Outline of Talk

- What is Data Mining?
  - Motivation
  - Definition and the Data Mining process
  - History of primary Data Mining society
- Data Curation and Preparation
- Techniques for Data Mining
  - Predictive Modeling – Supervised Learning
    - Classification
  - Segmenting Data – Unsupervised Learning
    - Clustering Methods
  - Anomaly Detection
- Case Studies: Applications to Diagnosis
  - Enhancing online diagnosers on aircraft
  - Subsystem-level diagnosis
  - System-level diagnosis
  - Demonstrations
- Discussion and Conclusions

Demonstrations

- Interspersed in Presentation
- We will be using the Weka Workbench
  - Waikato Environment for Knowledge Analysis developed at Univ. of Waikato, NZ
  - Written in Java, distributed under a Gnu General Public license under Linux, Windows, Mac OS
  - Collection of state of the art machine learning programs and data preprocessing tools

Access: *www.cs.waikato.ac.nz/ml/**weka**/*

Why Data Mining?

- Systems and processes that we work with in this world are complex
  - Manufacturing systems & plants, Power generation and distribution, Transportation systems
  - Business and retail systems, Economics
  - Social Systems
  - System of systems
- Very important that they are safe and secure, cost-effective, & efficient in operations

Have to envision and analyze how these systems work in different scenarios & environments; solve problems
State of the art first principles approaches not sufficient

- In contrast, we are overwhelmed with data about systems and processes
  - Advanced sensors
  - Extensive processing power and memory
  - Ability to bring together diverse sources of data

General agreement that there is useful, but hidden information in the data

Drowning in Data, Starving for Knowledge

What is data mining?

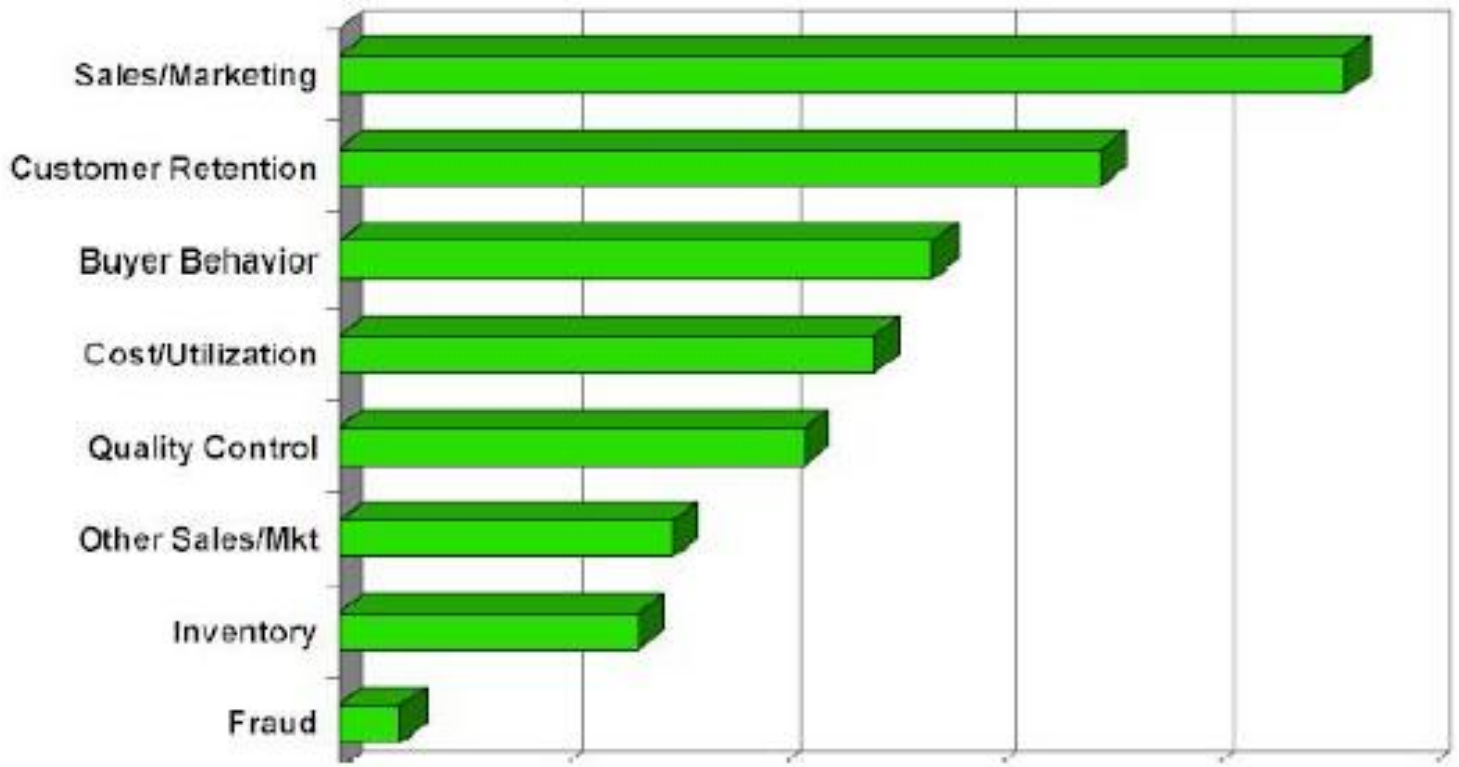
- Non-trivial extraction of novel, implicit, and actionable (useful, applicable) knowledge from large data bases
  - Challenges
    - Large data sets
    - Real world data is noisy, incomplete, sometimes erroneous; needs interpretation
    - Hypothesis formation may be part of the search or discovery process
    - Make sure knowledge and information extracted is non-obvious
    - Can lead to measurable improvements and gains

Cannot be done manually

- Technologies that enable data exploration, data analysis, and data visualization from very large (sometimes heterogeneous) data bases at a level of abstraction, such that meaningful and useful patterns may be extracted from the data

Data Mining Applications

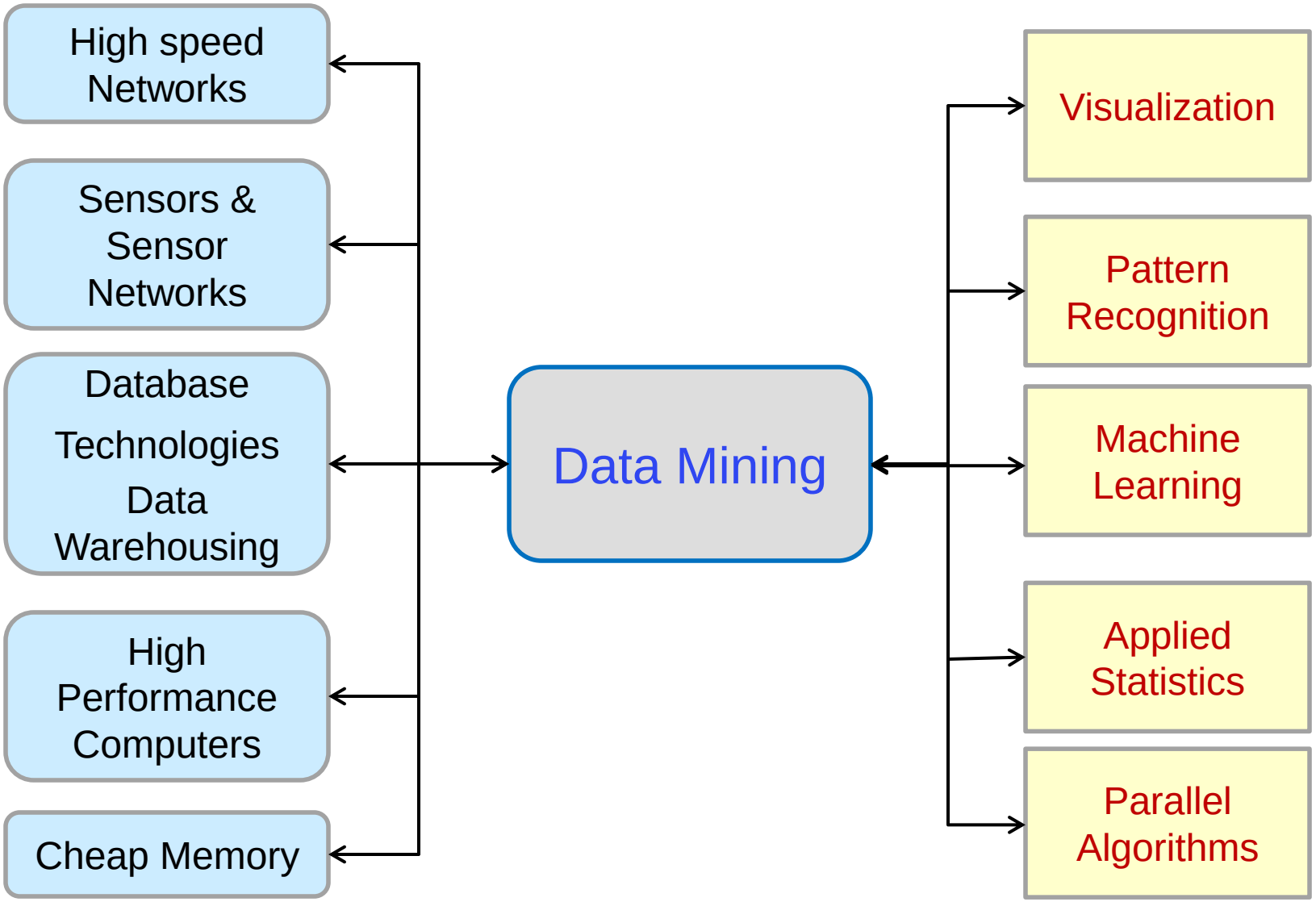
- Business-related Applications



Data Mining Applications

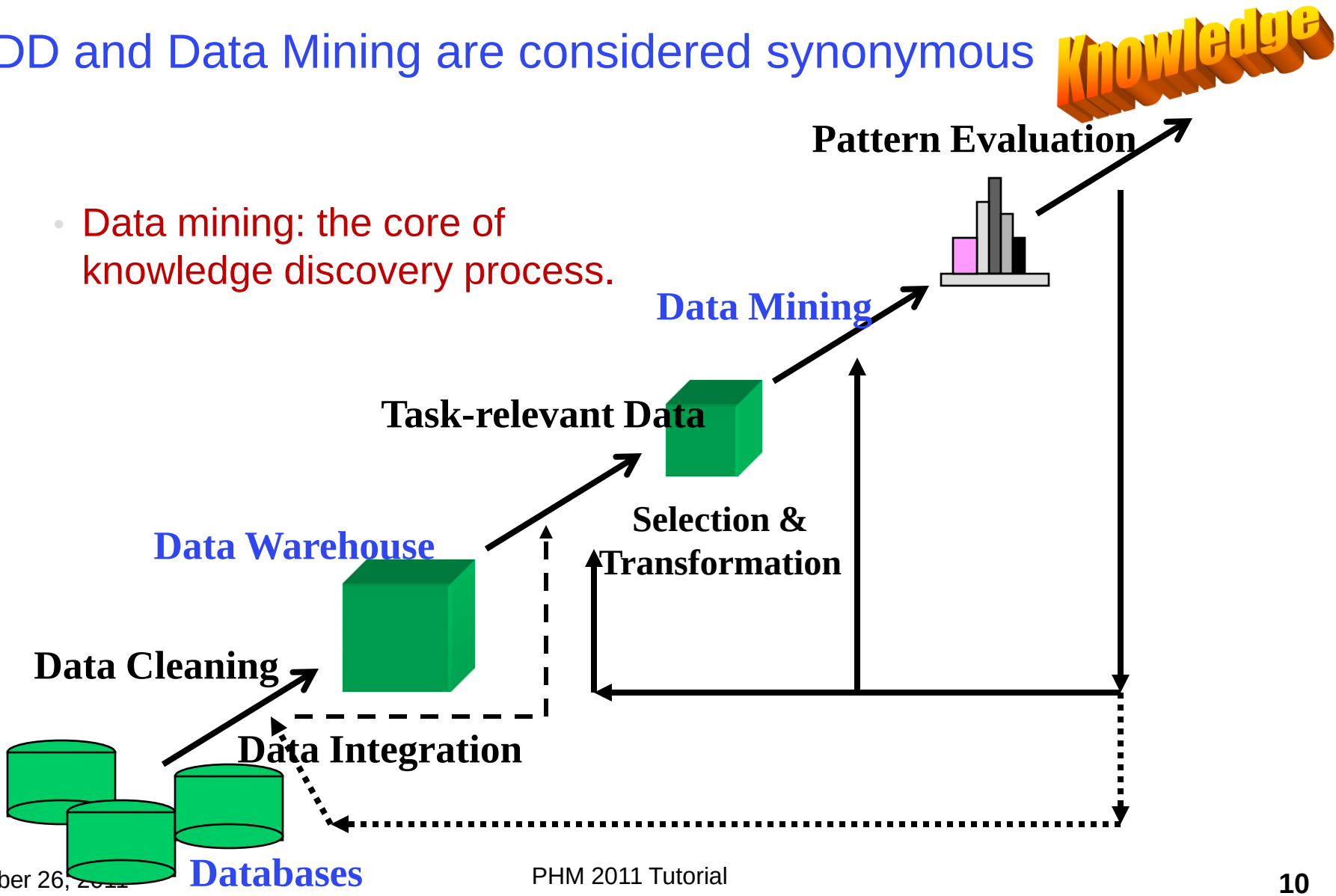
- Other Notable Applications
  - Web Mining
    - Page Ranking
    - Learning Page and Query relevance
    - Using search queries and pages read to derive user profiles
  - Image Analysis
    - Training classifiers for detecting weather conditions, oil spills, etc. in satellite images
    - Hazard detection
  - Load forecasting for Utilities
    - Prediction models to enhance static load models
  - Diagnosis
    - Mine vibration signatures in electromagnetic devices
    - Diagnostic rules to extend human expert judgment for complex systems

Data Mining Enablers



Knowledge Discovery in Databases (KDD)

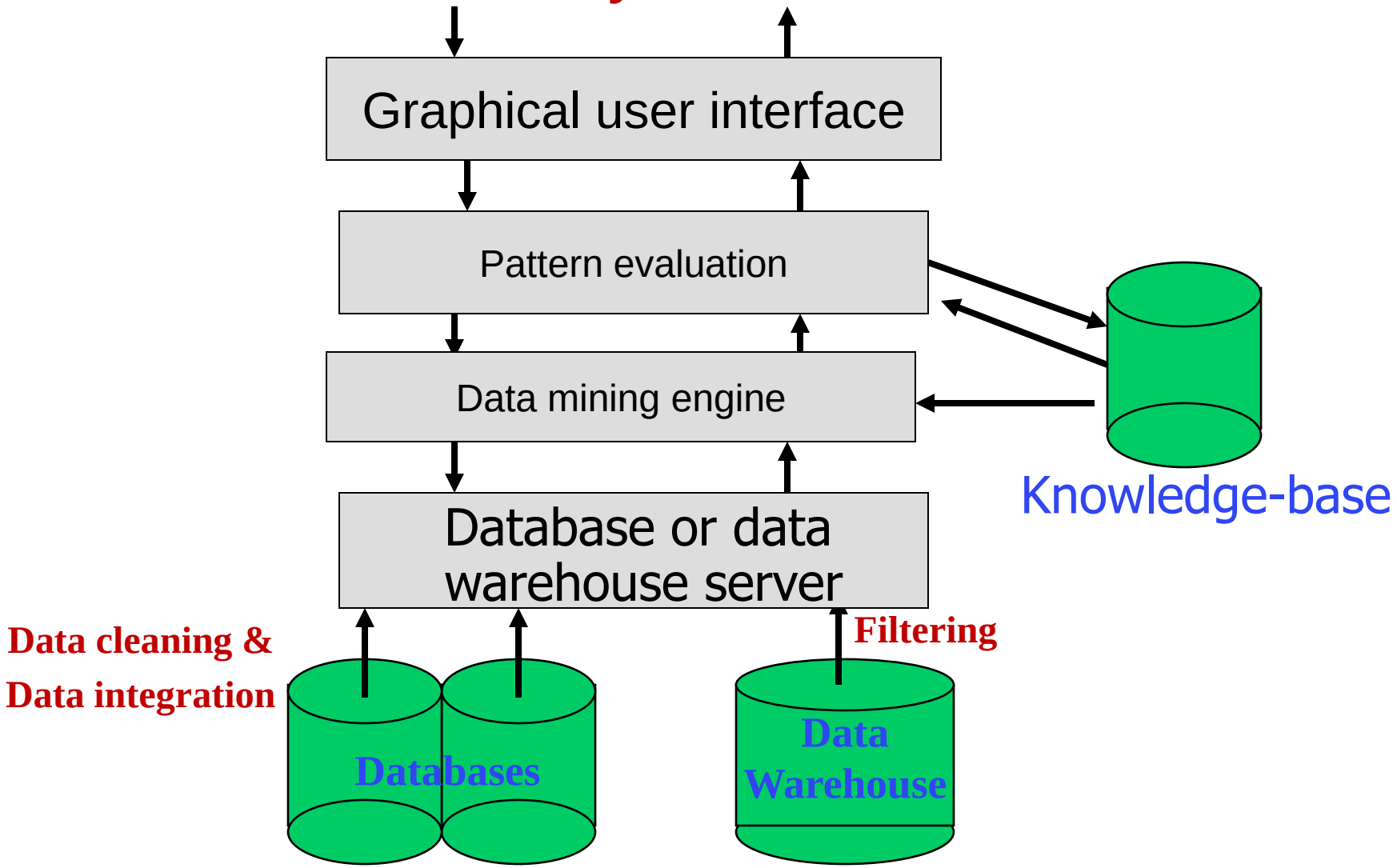
- KDD and Data Mining are considered synonymous
  - Data mining: the core of knowledge discovery process.



KDD (Data Mining) Important Steps

- Learning the application domain:
  - Relevant prior knowledge and goals of application
- Data cleaning and preprocessing: (may take 60% of effort!)
- Creating a target data set: data selection
- Data reduction and transformation:
  - Find useful features, dimensionality/variable reduction, invariant representation.
- Choosing data mining approach (task-driven)
  - summarization, classification, regression, association, clustering.
- Choosing the mining algorithm(s)
- Data mining: search for patterns of interest
- Pattern evaluation and knowledge presentation
  - visualization, transformation, removing redundant patterns, etc.
- Use of discovered knowledge

Architecture of a Typical Data Mining System



What kind of data?

- Relational databases
- Transactional databases
- Advanced DB and information repositories
  - Object-oriented and object-relational databases
  - Spatial databases
  - Time-series data and temporal data
  - Text databases and multimedia databases
  - WWW (Web)

Data Mining Functions

- Concept description: Characterization and discrimination
  - Generalize, summarize, and contrast data characteristics, e.g., dry vs. wet regions
- Association (correlation and causality)
  - Multi-dimensional vs. single-dimensional association
  - age(X, “20..29”) ^ income(X, “20..29K”) → buys(X, “PC”) [support = 2%, confidence = 60%]
  - Buys (diaper, brand A) → Buys (beer, brand B)
- Classification and Prediction
  - Discrimination: Finding models that describe and distinguish classes or concepts; use models to classify new data
    - e.g., classify countries based on climate, or classify cars by gas mileage
    - Presentation: decision-tree, classification rule, neural network, SVMs,
  - Regression: Model continuous valued functions; predict some unknown or missing numerical values
- Cluster analysis
  - Class label is unknown: Group data to form new classes, e.g., cluster houses to find distribution patterns
  - Clustering based on the principle: maximizing the intra-class similarity and minimizing the interclass similarity

Data Mining Additional Functions

- Outlier analysis
  - Outlier: a data object that does not comply with the general behavior of the data
  - It can be considered as noise or exception but is quite useful in fraud detection, rare events analysis
- Trend and evolution analysis
  - Trend and deviation: regression analysis
  - Sequential pattern mining, periodicity analysis
  - Similarity-based analysis

Relevant for Fault Analysis and Prognostics

Brief History of Data Mining Society

- 1989 IJCAI Workshop on Knowledge Discovery in Databases (Piatetsky-Shapiro)
  - Knowledge Discovery in Databases (G. Piatetsky-Shapiro and W. Frawley, 1991)
- 1991-1994 Workshops on Knowledge Discovery in Databases
  - Advances in Knowledge Discovery and Data Mining (U. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, 1996)
- 1995-1998 International Conferences on Knowledge Discovery in Databases and Data Mining (KDD'95-98)
  - Journal of Data Mining and Knowledge Discovery (1997)
- 1998 ACM SIGKDD, SIGKDD' conferences, and SIGKDD Explorations Newsletter, KDD Cup
  - SIGKDD conferences held yearly: see http://www.kdd.org/
- More conferences on data mining
  - PAKDD, PKDD, SIAM-Data Mining, (IEEE) ICDM, etc.

Data Curation and Preparation

Why preprocess data?

- Data cleaning
- Data integration and transformation
- Data reduction

Why do we need to preprocess data ?

- Data in the real world is dirty
  - incomplete: lack attribute values, lack attributes of interest, hard to annotate, or contain only aggregate data
  - noisy: contain errors or outliers
  - inconsistent: contain discrepancies in codes or names
- Lack of quality data $\Rightarrow$ results generated: inconsistent, lack robustness, do not contribute to knowledge gain!
  - Quality decisions must be based on quality data
  - Data warehouse needs consistent integration of relevant, quality data

Measures of data quality

- A well-accepted multidimensional view:
  - Accuracy
  - Completeness
  - Consistency
  - Timeliness
  - Believability
  - Value added
  - Interpretability
  - Accessibility

(source: *Han and Kamber, Data Mining: Concepts and Techniques, Morgan Kaufmann, 2006*)

Major data preprocessing tasks

- Data cleaning
  - Fill in missing values, smoothing noisy data, identify or remove outliers, and resolve inconsistencies
- Data integration
  - Integration of multiple databases, data cubes, or files
- Data transformation
  - Normalization and aggregation
- Data reduction
  - Obtains reduced representation in volume but produces the same or similar analytical results

Still an active area of research

Practical Example

- Regional Airline Data
  - Used in Case Studies
  - Raw Sensors
  - Several Aircraft over Several Years
- Trail of Data
  - CDs
    - Raw Binary Files
    - Capture Noise
  - Build Database

Curating the Flight Data

- Complex Data vs. Complex Representation
  - Simplifying of Representation While Maintaining Data
  - Simplify Workflow
  - Improving Readability
- Design Requirements
- Understanding Raw Data
  - Size
  - Data Types
- Initial Design
  - Good vs. Bad
  - Lessons
- Improved Design

Requirements of Database

- Fast to Access
  - Scale with Flights
  - Joining Multiple Tables
  - Easy to Update
- Balance with Size
  - Indexes on Multiple Tables
  - Can Grow with New Data
  - Doesn’t Need a High Grade Server
- Easy to Navigate
  - Essential over Raw Data
  - Lower overhead to pick up schema

Understanding the Raw Data

- The Single Element - A Flight
  - Binary File(Most Compact Rep)
  - 182 Sensors at Different Sampling Rates
  - Varying Flight Durations(Minutes to Hours)
  - Clean vs. Corrupt
  - Up to 12MB
- Full Set(to Date)
  - 12 Tail Numbers
  - Time for Each
  - >6000 Flights
  - Multiple Fault Annotations
- Reconcile with Requirements

Current Design

- Modified Library Model
  - Like Dewey Decimal system – two-step process
  - Still Hold Commonly Used Information
- Controller Table
  - Metadata on All Flights
  - Access Binary Filename
  - Built to work with Scripts and Java
- Monitor Tables
  - Summarize Entire Flights
  - Tie into the Controller Tables
  - Quick Access to Commonly Used Info
  - Easily Updated(or entire tables added)
  - 31.6GB $\rightarrow$ ~12GB + Raw Values

Supervised Learning

Classification Algorithms

- Discriminative: Decision Trees, SVM
- Generative: Bayesian classifiers

Classification

- What is classification?
  - Predicts categorical class labels
  - Classifies data by constructing a model (using some other attributes or features) based on a training set and class labels
  - Uses model to classify new data with unknown labels
- Example: US Congressional Voting Record 1984(from UCI Machine Learning Repository: http://archive.ics.uci.edu/ml/datasets/Congressional+Voting+Records)
  - 435 data points – each data point defined by 16 features (votes)
    - Example features: adoption of budget, physician fee freeze, aid to Nicaraguan contras, mx-missile, education spending, crime, …..
  - Labels: Democrat, Republican
  - Reference: Schlimmer, J. C. (1987). Concept acquisition through representational adjustment. Doctoral dissertation, Department of Information and Computer Science, University of California, Irvine, CA.

Supervised ← Class labels are known

Classification: Two-Step Process

- Model construction: describing a set of predetermined classes
  - Each data object is assumed to belong to a predefined class, as determined by the class label attribute
  - The set of tuples used for model construction: training set
  - The model is represented as classification rules, decision trees, or mathematical formulae
- Model use: for classifying future or unknown objects
  - Estimate accuracy of the model
    - The known label of test sample is compared with the classified result from the model
    - Accuracy rate is the percentage of test set samples that are correctly classified by the model (also, false positives, false negatives)
      - Cross validation studies
    - Test set is independent of training set, otherwise over-fitting will occur

Classification Methods

- Decision Trees
  - Top-down divide and conquer
- Covering algorithms for constructing rules
  - Start with rule that covers some of the instances; expand to include other instances and exclude instances that are not of the same type
- Linear and Nonlinear Regression – for numeric data
- Instance-based classifiers
  - *k*-nearest neighbor algorithms
- Bayesian classifiers
  - Probabilistic, predict multiple hypotheses based on probabilities
- Neural Networks
  - Backpropagation algorithms; robust, accurate, but hard to interpret and reconcile with domain knowledge
- Support Vector Machines (SVM)
  - Discriminant functions – hyperplanes, radial basis functions, kernel methods

Decision Trees: Illustrated

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no

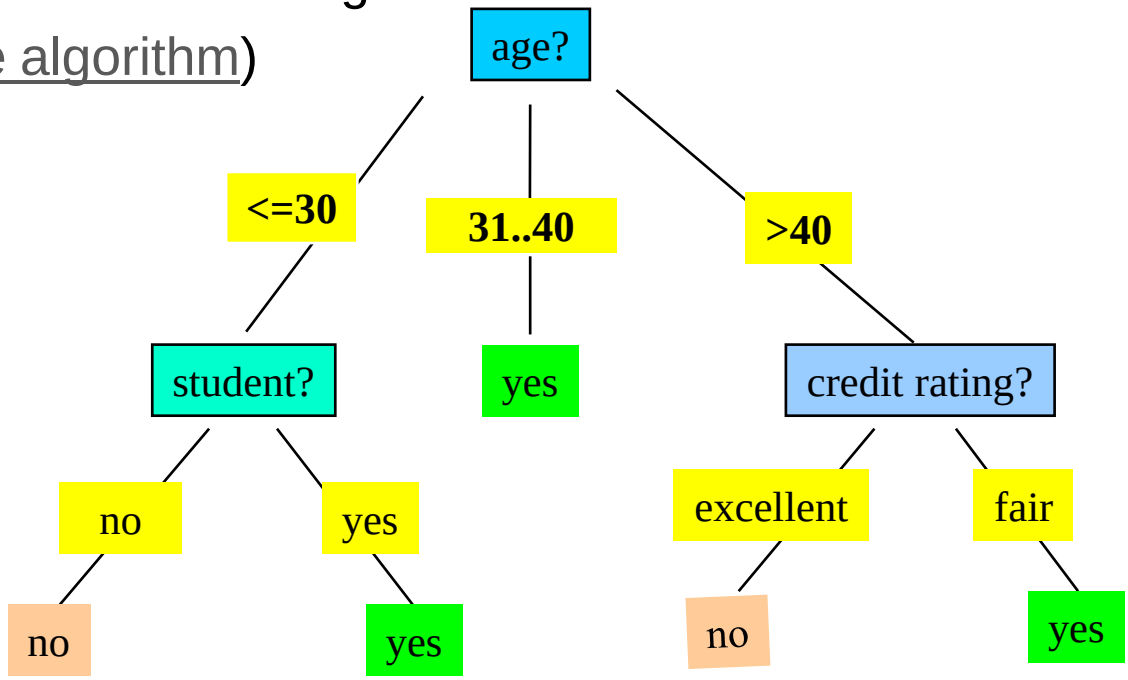
Simpler dataset

- Given user profiles (four features) + class label: (buys computer)
- Build model of computer buyer & non buyer

Decision Tree Classifier: Discriminative

- Decision Tree represents a tree-structured plan of a set of features to test in order to predict the output
- Algorithm:
  - Choose feature with highest information gain
  - Then, repeat … (recursive algorithm)

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no



Information Gain (ID3/C4.5)

J. Ross Quinlan: **C4.5: Programs for Machine Learning**. Morgan Kaufmann Publishers, Inc., 1993.

- To satisfy Inductive Bias: shorter trees preferred to deeper ones (also satisfies Occam's Razor principle)
- Select features with the highest information gain first
- Assume there are two classes, *P* and *N*
  - Let the set of examples *S* contain *p* elements of class *P* and *n* elements of class *N*
  - The amount of information, needed to decide if an arbitrary example in *S* belongs to *P* or *N* is defined as

$$I(p,n) = -\frac{p}{p+n}\log_2\frac{p}{p+n} - \frac{n}{p+n}\log_2\frac{n}{p+n}$$

Information Gain in Decision Tree Induction

- Assume that using feature *F*, a set *S* will be partitioned into sets $\{S_1, S_2, \ldots, S_v\}$
  - If S_i contains p_i examples of *P* and n_i examples of *N*, the entropy, or the expected information needed to classify objects in all subtrees S_i is

$$E(F) = \sum_{i=1}^{v} \frac{p_i + n_i}{p + n} I(p_i, n_i)$$

- The encoding information that would be gained by branching on *F*

$$Gain(F) = I(p, n) - E(F)$$

Attribute Selection by Information Gain Computation

- Class P: buys_computer = "yes"
- Class N: buys_computer = "no"
- I(p, n) = I(9, 5) =0.940
- Compute the entropy for age:

age	p_i	p_i	$I(p_i, n_i)$
≤ 30	2	3	0.971
30..40	4	0	0
> 40	3	2	0.971

$$E(age) = \frac{5}{14} I(2,3) + \frac{4}{14} I(4,0) + \frac{5}{14} I(3,2) = 0.69$$

- Hence

$$Gain(age) = I(p,n) - E(age)$$

- Similarly.

$$Gain(income) = 0.029$$
$$Gain(student) = 0.151$$
$$Gain(credit\_rating) = 0.048$$

Continuous-valued features

- Gini index (IBM IntelligentMiner)
  - All attributes are assumed continuous-valued
  - Assume there exist several possible split values for each attribute
  - May need other tools, such as clustering, to get the possible split values
- If a data set *T* contains examples from *n* classes, gini index, *gini*(*T*) is defined as $gini(T) = 1 - \sum_{j=1}^{n} p_j^2$

  where p_j is the relative frequency of class *j* in *T*.
- If a data set *T* is split into two subsets T_1 and T_2 with sizes N_1 and N_2 respectively, the *gini* index of the split data contains examples from *n* classes, the *gini* index *gini*(*T*) is defined as

$$gini_{split}(T) = \frac{N_1}{N} gini(T_1) + \frac{N_2}{N} gini(T_2)$$

- The attribute provides the smallest $gini_{split}(T)$ is chosen to split the node (*need to enumerate all possible splitting points for each attribute*).

Extracting Classification Rules from Trees

- Represent the knowledge in the form of IF-THEN rules
  - One rule is created for each path from the root to a leaf
  - Each attribute-value pair along a path forms a conjunction
  - The leaf node holds the class prediction
  - Rules are easier for humans to understand
- Example

  IF *age* = "<=30" AND *student* = "*no*" THEN *buys_computer* = "*no*"

  IF *age* = "<=30" AND *student* = "*yes*" THEN *buys_computer* = "*yes*"

  IF *age* = "31…40" THEN *buys_computer* = "*yes*"

  IF *age* = ">40" AND *credit_rating* = "*excellent*" THEN *buys_computer* = "*yes*"

  IF *age* = ">40" AND *credit_rating* = "*fair*" THEN *buys_com puter* = "*no*"

Validation of Decision Tree structure

- Separate training (e.g., 67%, 75%, 90%) and testing (33%, 25%, 10%) sets
- Leave one out classifier
- Use cross validation, e.g., 10-fold cross validation
  - Divide data into 10 equal parts – use 9 parts for training, 1 for test; repeat 10 times with each part playing the role of test …

Avoid Overfitting in Decision Trees

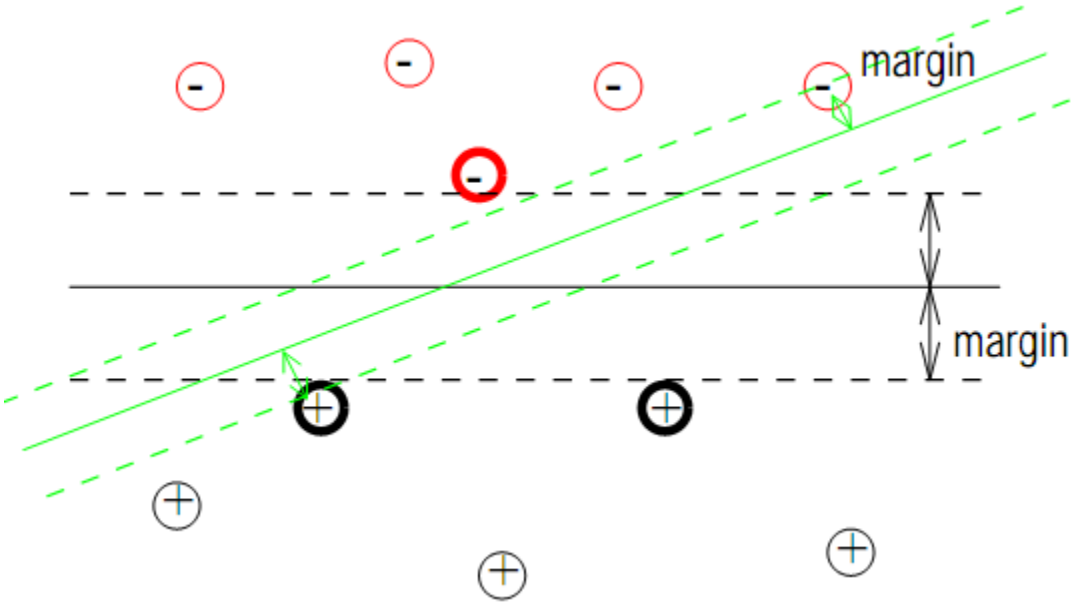
- The generated tree may overfit the training data
  - Too many branches, some may reflect anomalies due to noise or outliers
  - Result is in poor accuracy for unseen samples
- Two approaches to avoid overfitting
  - Pre-pruning: Halt tree construction early—do not split a node if this would result in the goodness measure falling below a threshold
    - Difficult to choose an appropriate threshold
  - Post-pruning: Remove branches from a "fully grown" tree – get a sequence of progressively pruned trees
    - Use a set of data different from the training data to decide which is the "best pruned tree"
- Use minimum description length (MDL) principle:
  - halting growth of the tree when the encoding is minimized

Decision Tree Demo

- Understanding the Data
- Congressional Voting Record
  - 1984 – Reagan Era
  - 16 Votes
    - Test-ban/religious groups in schools
  - Missing Values
- Utilizing a Decision Tree Algorithm
  - Examining the Metrics and Statistics
- Examining the Structure

Support Vector Machines: Discriminative

- Based on Statistical Learning Theory from the 60s (Vapnik, Chervonenkis)
- Method for Building Regression Lines and Classifiers



- Images and Information from slides by Jason Weston at NEC Labs America

Support Vector Machines

- Vapnik and Chervonenkis found Upper Bound on True Risk

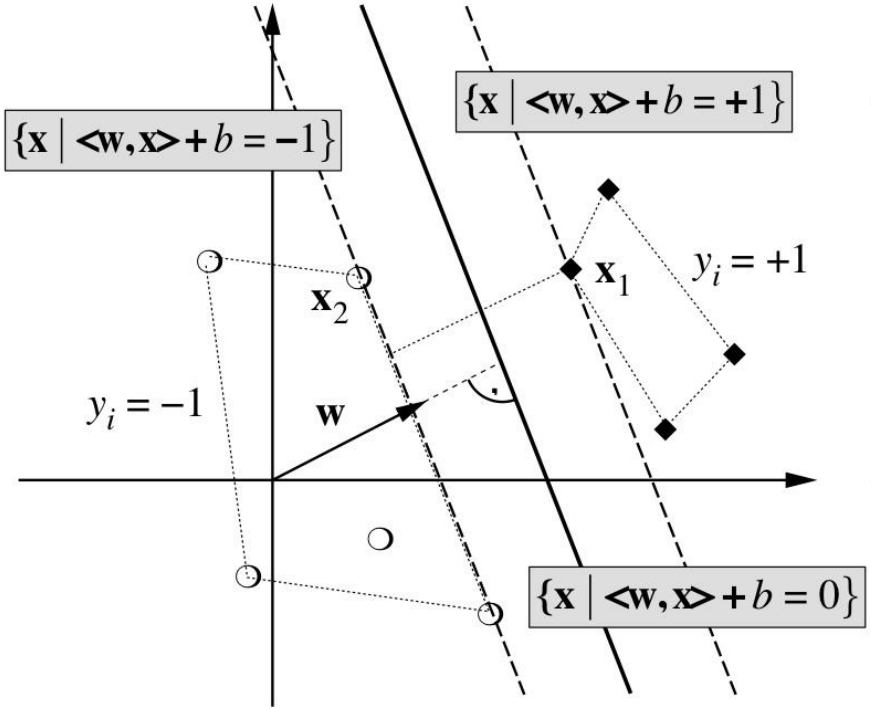
$$R(\alpha) \leq R_{emp}(\alpha) + \sqrt{\frac{h(log(\frac{2m}{h}+1) - log(\frac{\eta}{4})}{m}}$$

- h is the VC Dimension of the type of classifier
  - As a hyperplane, this is n+1 where the plane is in n dimensions

- Minimizing this function produces a bounded classifier with optimally low risk.

Support Vector Machines

- Sticking with Hyper planes, we want to separate the data as cleanly as possible.



Note:

$$\langle w, x_1\rangle + b = +1$$
$$\langle w, x_2\rangle + b = -1$$
$$\Rightarrow \quad \langle w, (x_1 - x_2)\rangle = 2$$
$$\Rightarrow \left\langle \frac{w}{\|w\|}, (x_1 - x_2)\right\rangle = \frac{2}{\|w\|}$$

Support Vector Machines

- This ends up as a minimization of the weights w, according to the constraints.

Minimize $||w||^2$, subject to:

$$(w \cdot x_i + b) \geq 1, \quad \text{if} \quad y_i = 1$$

$$(w \cdot x_i + b) \leq -1, \quad \text{if} \quad y_i = -1$$

The last two constraints can be compacted to:

$$y_i(w \cdot x_i + b) \geq 1$$

- This is a quadratic program

SVM Extensions and Demo

- Nonseperable Data
  - C-Term
- Complex Boundaries on Data
  - Kernel Trick
- One-Class Learning
- Demo
  - Same Data
  - Understanding the Metrics
  - Different Kernels

Bayesian Classifiers: Generative

- Why Bayesian classifiers?
  - Probabilistic learning: Calculate explicit probabilities for hypothesis, among the most practical approaches to certain types of learning problems; Combine prior knowledge with existing data
  - Incremental: Each training example can incrementally increase/decrease the probability that a hypothesis is correct. Prior knowledge can be combined with observed data.
  - Probabilistic prediction: Predict multiple hypotheses, weighted by their probabilities
  - Standard: Even when Bayesian methods are computationally intractable, they can provide a standard of optimal decision making against which other methods can be measured

Tom M. Mitchell: **Machine Learning**. McGraw Hill, International Editions, 1997.

Bayes Theorem

- Given training data *D*, *posteriori probability of a hypothesis h, P(h|D)* follows the Bayes theorem

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)}$$

- MAP (maximum posteriori) hypothesis

$$h_{MAP} = \arg\max_{h \in H} P(h|D) = \arg\max_{h \in H} P(D|h) \cdot P(h)$$

- Maximum Likelihood Estimate

$$h_{ML} = \arg\max_{h \in H} P(D|h)$$

Practical difficulty: require initial knowledge of many probabilities
incur significant computational cost

Bayes Optimal Classifier

- Most probable classification of new instance – combine predictions of all hypotheses weighed by posterior probabilities

$$P(v_j \mid D) = \sum_{h_i \in H} P(v_j \mid h_i) \cdot P(h_i \mid D)$$

v_j new instance from set of values V

- Bayes optimal classification

$$\arg\max_{v_j \in V} \sum_{h_i \in H} P(v_j \mid h_i) \cdot P(h_i \mid D)$$

  - Maximizes probability that new instance is classified correctly

Computationally expensive: compute posterior probability for every hypothesis in H, then combine the predictions of each hypothesis

Naïve Bayes Classifier

- Practical solution based on simplified assumption: attributes are conditionally independent given the target value, *v*
  - Each instance (object) to be classified defined by set of features

$$\langle f_1, f_2, \cdots f_n \rangle$$

$$v_{MAP} = \arg\max_{v_j \in V} P(v_j \mid f_1, f_2, \cdots, f_n)$$

$$= \arg\max_{v_j \in V} \frac{P(f_1, f_2, \cdots, f_n \mid v_j) \cdot P(v_j)}{P(f_1, f_2, \cdots, f_n)}$$

$$= \arg\max_{v_j \in V} P(v_j) \cdot \prod_i P(f_i \mid v_j) \quad : \text{Naive Bayes Classifier}$$

  - $P(v_j)$ and $P(f_i|v_j)$ estimated from frequencies in training data
  - If f_i continuous use Gaussian density functions to compute $P(f_i|v_j)$

Greatly reduces the computation cost, only count the class distribution

Naïve Bayes Classifier: Play Tennis Data set

Outlook	Temperature	Humidity	Windy	Class
sunny	hot	high	false	N
sunny	hot	high	true	N
overcast	hot	high	false	P
rain	mild	high	false	P
rain	cool	normal	false	P
rain	cool	normal	true	N
overcast	cool	normal	true	P
sunny	mild	high	false	N
sunny	cool	normal	false	P
rain	mild	normal	false	P
sunny	mild	normal	true	P
overcast	mild	high	true	P
overcast	hot	normal	false	P
rain	mild	high	true	N

Naive Bayesian Classifier: example

- Given a training set, we can compute the probabilities
- Training set: Given outside conditions, should one play tennis?
  - 14 data points
  - Four features: (1) outlook; (2) Temperature; (3) Humidity; (4) Wind

Outlook	Play	No	Humidity	Play	No
sunny	2/9	3/5	high	3/9	4/5
overcast	4/9	0	normal	6/9	1/5
rain	3/9	2/5			
Temperature			**Wind**		
hot	2/9	2/5	strong	3/9	3/5
mild	4/9	2/5	weak	6/9	2/5
cool	3/9	1/5			

P(*Play Tennis* = yes) = 9/14 = 0.64;
P(*Play Tennis* = no) = 5/14 = 0.36

$$P(PlayTennis = yes \mid cond) = P(PlayTennis)\prod_i P(f_i \mid PlayTennis)$$

$$= \frac{9}{14} \times \frac{2}{9} \times \frac{3}{9} \times \frac{3}{9} \times \frac{3}{9} = 0.0053$$

Given: cond = ⟨ sunny, cool, high, strong⟩
will Play Tennis = yes?

$$P(PlayTennis = no \mid cond) = 0.0206$$

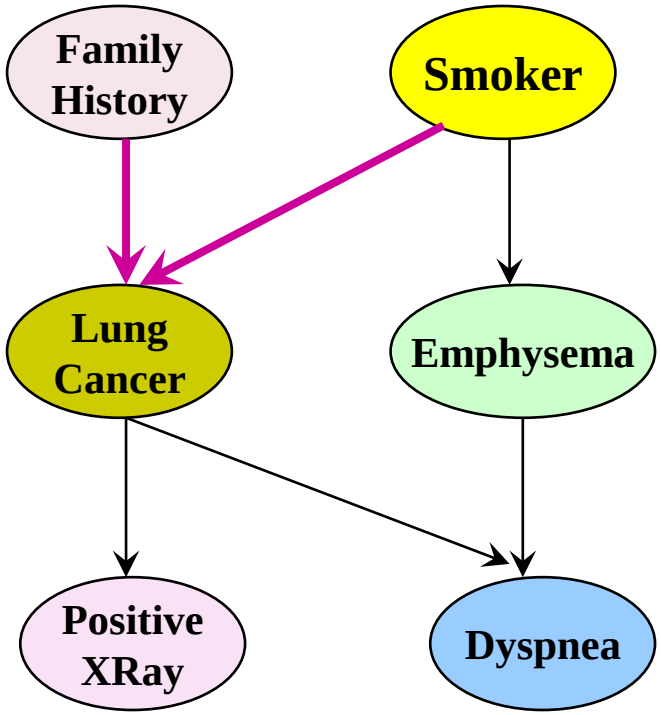
Naïve Bayes classifier

- Problems
  - How well do sample frequencies truly represent the true distributions in the domain?
    - Example: consider building a classifier to analyze faults or diseases
    - How does one establish prior probabilities?
      - May have to rely on expert judgment
    - What to do when the number of occurrences of a particular feature are small?
      - In the extreme, what if number = 0 in data set: Since probabilities are multiplied this will strongly bias result
  - In real life, features are not truly independent
    - Example: Outlook and Temperature

Bayesian Belief Networks

- Conditional independence assumption on the features overly restrictive
  - Use Bayesian belief networks that extend conditional independence to subsets of variables
  - Bayes net structure derived using known causal relations between features – Graphical model

	(FH, S)	(FH, ~S)	(~FH, S)	(~FH, ~S)
LC	0.8	0.5	0.7	0.1
~LC	0.2	0.5	0.3	0.9



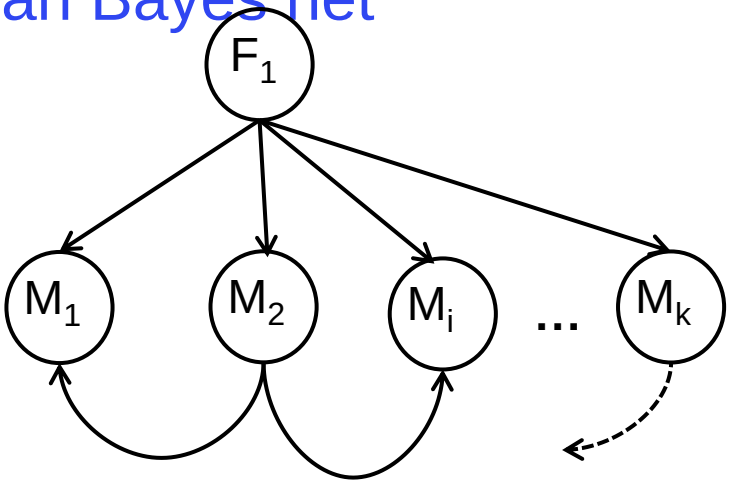
- Learning classifier
  - Learn structure of network
  - Learn parameters of network structure: conditional probabilities (tables, distributions)

Tree Augmented Naïve Bayesian Networks (TANs)

- Extends Naïve Bayes structure to include relations among nodes, with the following restriction
  - Each node in the network can have at most two parents: (i) root node – class variable; and (ii) one other node in the network

  Therefore, not as general as a Bayesian network, but a tree-structured network that is a Markov tree

- Computationally faster to learn than Bayes net
- Structure helps readability



Algorithms for learning TAN structures

- Algorithms to Learn TANs
  - Focus on the tree structure to help improve process
  - Equivalent to selecting the best *k* features that uniquely classify a particular hypothesis – known as the feature selection problem (equivalent to inducing a Markov tree)
  - Computational Complexity: $O(kn^2N)$, where *n* is the number of available monitors, and *N* is the number of different flight segments available for analysis
  - Two search methods for TAN structure
    - Greedy search (Cohen, et al., 2004)
    - Mutual information function in combination with a minimum spanning tree algorithm (Chen, 2006)

Algorithm for Learning TAN structure

- Dataset *D* with *N* features and a class label *C*
- Observational Root Node F_{Root},
- *CorrelationFunction* (*Bayesian Values or BIC*)
- Corr : likelihood matrix for each pair of nodes (*i*,*j*)
- AdjMat: adjacency between nodes: can access parents of node
- Steps
  1. Build a Minimum Weighted Spanning Tree (MWST) using the Correlation Matrix and the Root chosen
  2. Connect every feature to the Class Node to build the TAN
  3. Estimate the parameters, starting with the class
  4. Return (AdjMat, ProbVec); ProbVec: marginal distributions for links

Notes:
- This algorithm uses MWST with BIC to generate TAN structure; Could have used "greedy" search as alternative
- Choice of observational root node: an important feature

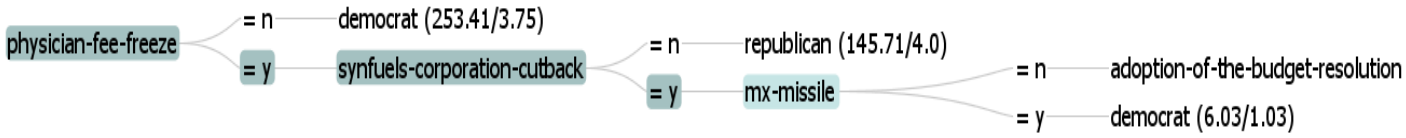
Bayesian Learning Demo

- Naïve Bayes Classifier
  - Examine Metrics
- Extend to TANs
  - Examine Metrics

- Adding Complexity
  - Examine Metrics
  - Examine Structure

Compare Supervised Methods

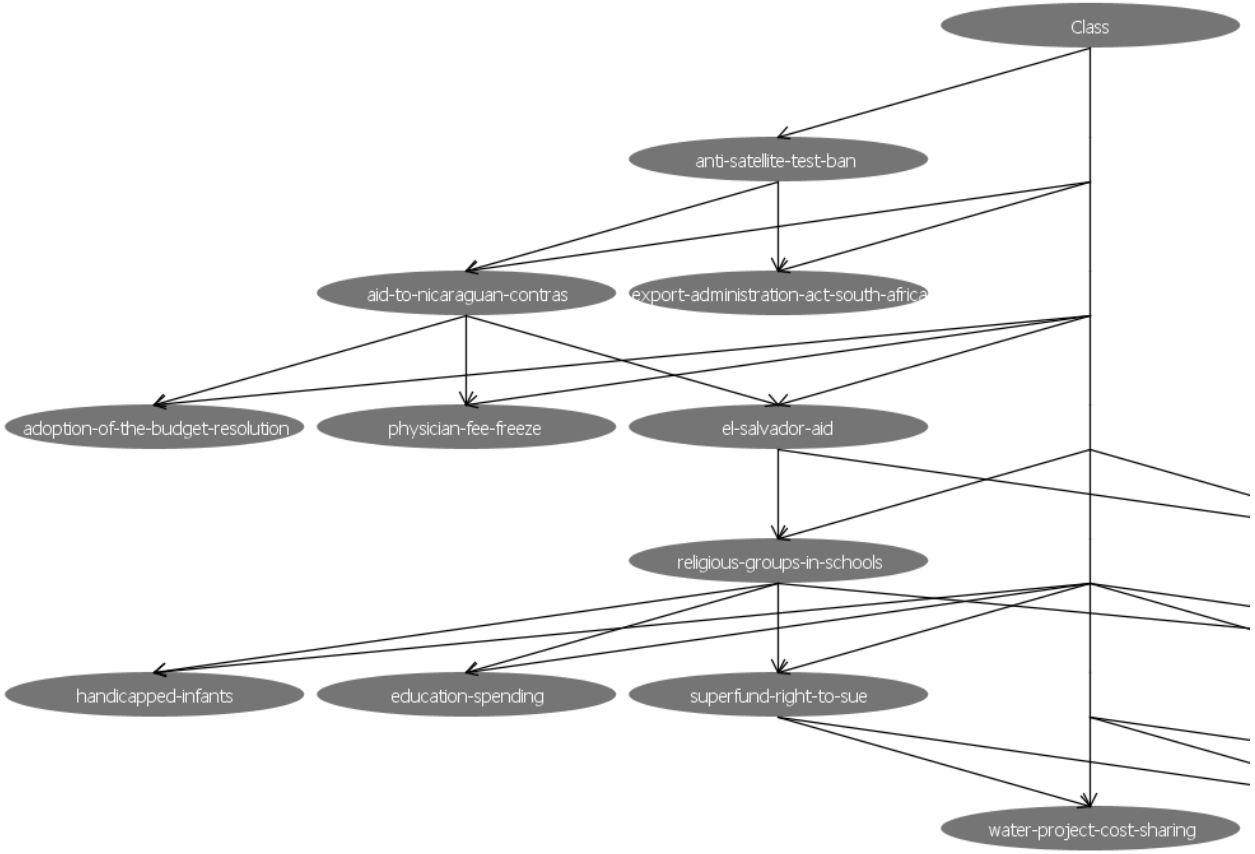
- Decision Tree
  - Readability
  - Rule Based



- Support Vector Machines
  - Readability
    - Weights
    - Support Vectors
  - Hyperplane Dividers

- Bayesian Method
  - Readability
    - Size issues
  - Probabilistic

Compare Supervised Methods: Bayesian Network



Unsupervised Learning

Clustering Methods

- Hierarchical Clustering: Single- & Complete-Link
- Partitional Clustering: k-Means
- Expectation-Maximization (EM) and a mixture of Gaussians

What defines a good clustering result?

- A good clustering method will produce high quality clusters defined by
  - high intra-class similarity (low intra-class dissimilarity)
  - low inter-class similarity (high inter-class dissimilarity)
- The quality of a clustering result depends on both the similarity (dissimilarity) measure used by the method and its implementation (control structure)
- From data mining viewpoint: quality of a clustering method is measured by its ability to discover hidden patterns in data.

Cluster Analysis Fundamentals

- Data Object definition
  - Each data object defined by a set of *m* features
    - *n* data objects: $n \times m$ matrix
  - Data objects defined by similarity or dissimilarity between all pairs of objects
    - *n* data objects: $\frac{1}{2}\, n.(n-1)$ numbers
  - Feature values
    - Binary: 0,1
    - Nominal: discrete, but multi-valued: color: red, yellow, blue
    - Ordinal: discrete or continuous, but order is important (1, 17, 54, 60)
    - Interval: separation between numbers; scores: (45, 50) versus (20, 80) on a scale of (0,100)
    - Ratio: numbers have absolute meaning, e.g., integers, reals
    - Mixed:
- Feature values define type of similarity or dissimilarity measure employed
  - Example: ratio-valued (real-valued) features distance measures used are metrics -- satisfy the triangle inequality
    - Generalized form: Minkowski, special forms: Manhattan, Euclidean, Mahalanobis

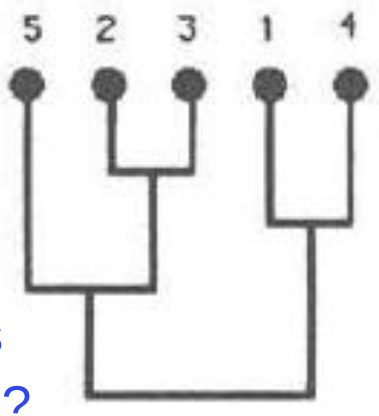
Major Clustering Approaches

- Hierarchical algorithms: Create a hierarchical decomposition of the set of data (or objects) using some criterion
- Partitioning algorithms: Construct various partitions and then evaluate them by some criterion
- Density-based: based on connectivity and density functions
- Grid-based: based on a multiple-level granularity structure
- Model-based: A model is hypothesized for each of the clusters and the idea is to find the best fit of that model to each other

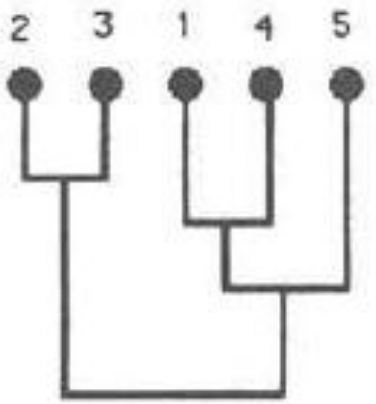
Hierarchical Clustering

- Graph Theoretic Methods:
  - Single-link
    - Bring together groups that have smallest dissimilarity between any pair of objects
  - Complete-Link clustering algorithms
    - Bring together groups that form minimal complete subgraphs

$$\mathcal{D}_1 = \begin{array}{c} \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{array} \begin{array}{c} \begin{array}{ccccc} x_1 & x_2 & x_3 & x_4 & x_5 \end{array} \\ \begin{bmatrix} 0 & 6 & 8 & 2 & 7 \\ 6 & 0 & 1 & 5 & 3 \\ 8 & 1 & 0 & 10 & 9 \\ 2 & 5 & 10 & 0 & 4 \\ 7 & 3 & 9 & 4 & 0 \end{bmatrix} \end{array}$$



Single Link



Complete Link

- Clusters displayed as dendrograms
- How do you determine # of clusters?
- Algorithms do not scale easily

Partitional Clustering

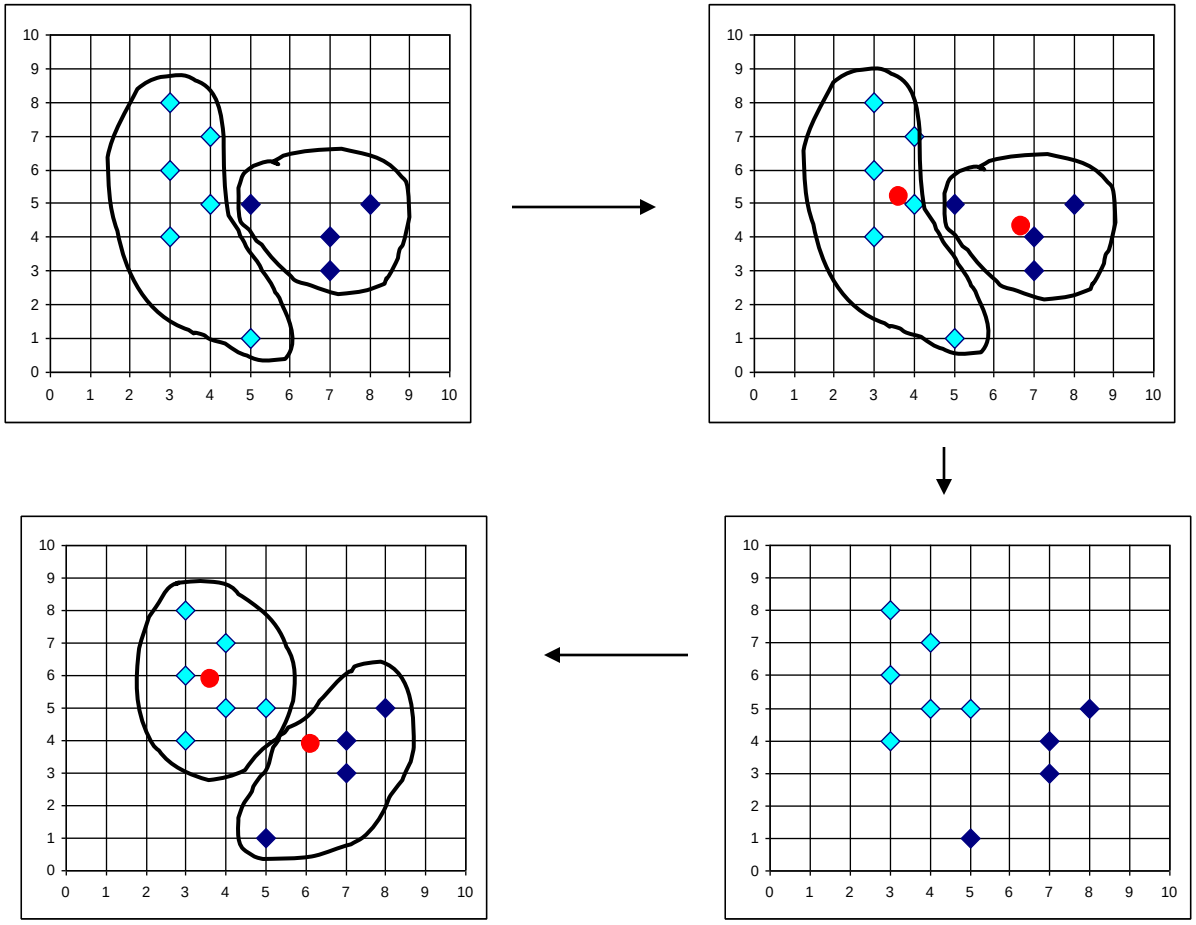
- Partitioning method: Construct a partition of a database ***D*** of ***n*** objects into a set of ***k*** clusters
- Given a *k*, find a partition of *k clusters* that optimizes the chosen partitioning criterion (square-error clustering methods)
  - Global optimal: exhaustively enumerate all partitions
  - Heuristic methods: *k-means* and *k-medoids* algorithms
    - *k-means* (MacQueen'67): Each cluster is represented by the center of the cluster
    - *k-medoids* or PAM (Partition around medoids) (Kaufman & Rousseeuw'87): Each cluster is represented by one of the objects in the cluster

The *K-Means* Clustering Method

- Given *k*, the *k-means* algorithm is implemented in 4 steps:
  - Partition objects into *k* nonempty subsets
  - Compute seed points as the centroids of the clusters of the current partition. The centroid is the center (mean point) of the cluster.
  - Assign each object to the cluster with the nearest seed point.
  - Go back to Step 2, stop when no more new assignments.

The *K-Means* Clustering Method

- Example



Comments on the *K-Means* Method

- Strength
  - *Relatively efficient*: *O*(*tkn*), where *n* is # objects, *k* is # clusters, and *t* is # iterations. Normally, *k*, *t* << *n*.
  - Often terminates at a *local optimum*. The *global optimum* may be found using techniques such as: *deterministic annealing* and *genetic algorithms*

- Weakness
  - Applicable only when *mean* is defined, then what about categorical data?
  - Need to specify *k,* the *number* of clusters, in advance
  - Unable to handle noisy data and *outliers*
  - Not suitable to discover clusters with *non-convex shapes*

Variations of the *K-Means* Method

- A few variants of the *k-means* which differ in
  - Selection of the initial *k* means
  - Dissimilarity calculations
  - Strategies to calculate cluster means
- Handling categorical data: *k-modes* (Huang'98)
  - Replacing means of clusters with modes
  - Using new dissimilarity measures to deal with categorical objects
  - Using a frequency-based method to update modes of clusters
  - A mixture of categorical and numerical data: *k-prototype* method

The *K-Medoids* Clustering Method

- Find *representative* objects, called medoids, in clusters
- *PAM* (Partitioning Around Medoids, 1987)
  - starts from an initial set of medoids and iteratively replaces one of the medoids by one of the non-medoids if it improves the total distance of the resulting clustering
  - *PAM* works effectively for small data sets, but does not scale well for large data sets
- *CLARA* (Kaufmann & Rousseeuw, 1990)
- *CLARANS* (Ng & Han, 1994): Randomized sampling
- Focusing + spatial data structure (Ester et al., 1995)

K-Means Demo

- Set Number of Clusters
- Examine Output

Density-Based Clustering Methods

- Clustering based on density (local cluster criterion), such as density-connected points
- Major features:
  - Discover clusters of arbitrary shape
  - Handle noise
  - One scan
  - Need density parameters as termination condition
- Several interesting studies:
  - DBSCAN: Ester, et al. (KDD'96)
  - OPTICS: Ankerst, et al (SIGMOD'99).
  - DENCLUE: Hinneburg & D. Keim (KDD'98)
  - CLIQUE: Agrawal, et al. (SIGMOD'98)

Probabilistic Clustering: Mixture of Gaussians

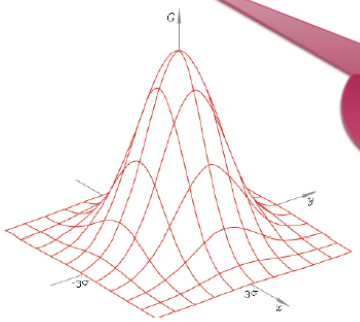
WHAT'S A GAUSSIAN?

$$p(X) = N(X \mid \mu, \Sigma)$$

Mean

Variance

Gaussian == "Normal" distribution



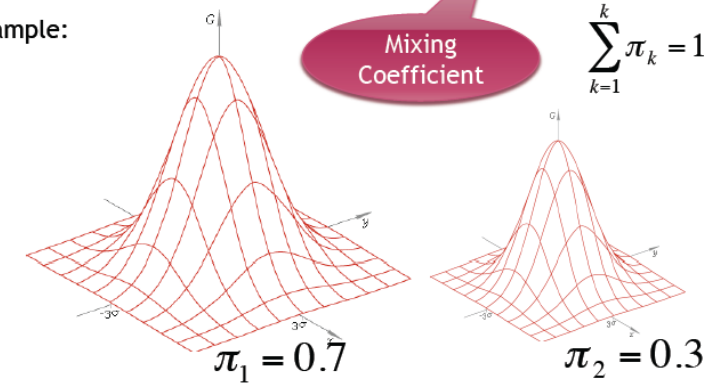
MIXTURE OF GAUSSIANS

$$p(X) = \pi_1 N(X \mid \mu_1, \Sigma_1) + \pi_2 N(X \mid \mu_2, \Sigma_2)$$

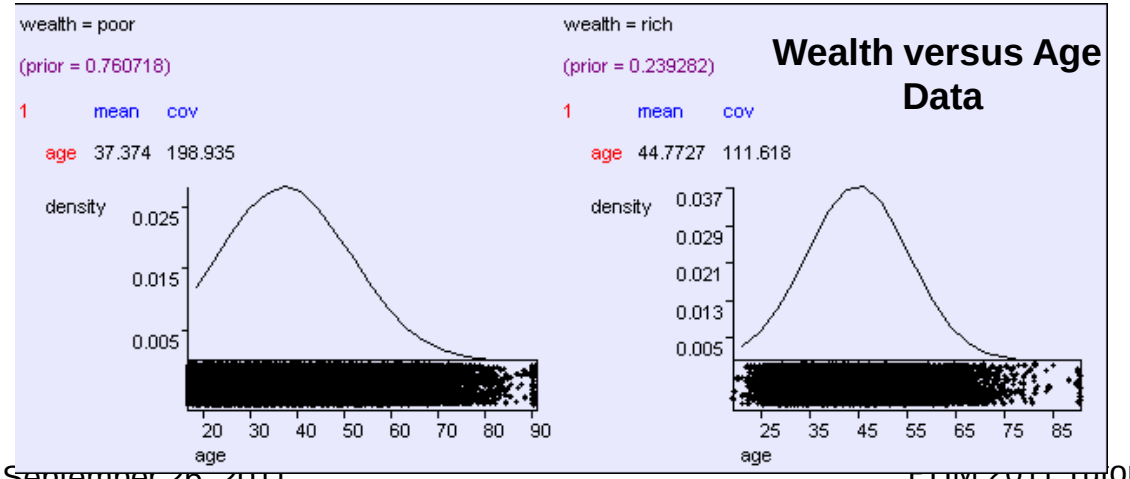
Example:

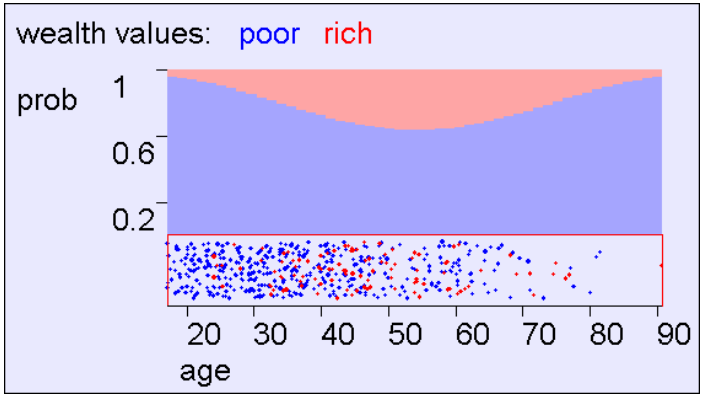
Mixing Coefficient

$$\sum_{k=1}^{k} \pi_k = 1$$



- WHY DO WE NEED A MIXTURE OF GAUSSIANS?





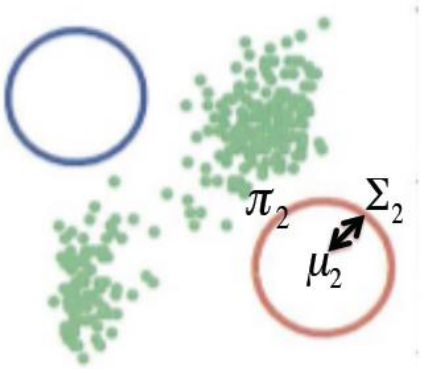
Clustering using a Mixture of Gaussians

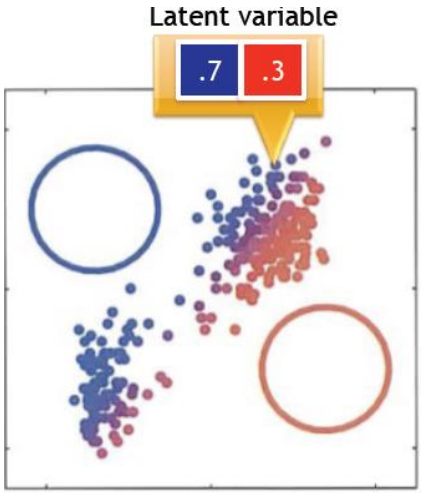
- K-Means (Hard clustering)
  - Based on mean
- Expectation Maximization (EM) Algorithm using a mixture of Gaussians (Soft clustering)
  - Based on mean, variance, and mixing factor
- EM-Based Clustering Algorithm
  - Choose k : number of clusters
  - Initialize μ_i, Σ_i, π_i , $1 \leq i \leq k$ (use k-Means)
  - ***E-step****:* For each data object X_j , $1 \leq j \leq n$ *determine* assignment score, $\gamma(z_{ji})$ to each Gaussian (how much is each Gaussian responsible for X_j)
  - ***M-step***: Update parameters for each Gaussian using new $\gamma(z_{ji})$
  - Evaluate likelihood. If likelihood or parameter converge, stop, else repeat *E*- and *M*-steps.

EM algorithm: E & M steps

- Initialize using k-means

$$\mu_k \leftarrow \mu_k$$
$$\Sigma_k \leftarrow \text{cov}(cluster(K))$$
$$\pi_k \leftarrow \frac{\text{Number of points in k}}{\text{Total number of points}}$$



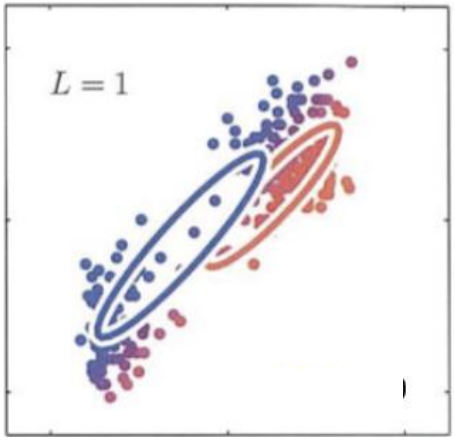


- E-step: computing assignment score for each data point

$$\gamma(z_{ji}) = \frac{\pi_i \cdot N(x_j \mid \mu_i, \Sigma_i)}{\sum_{i=1}^{k} \pi_i \cdot N(x_j \mid \mu_i, \Sigma_i)}$$

- M-step: update Gaussian parameters

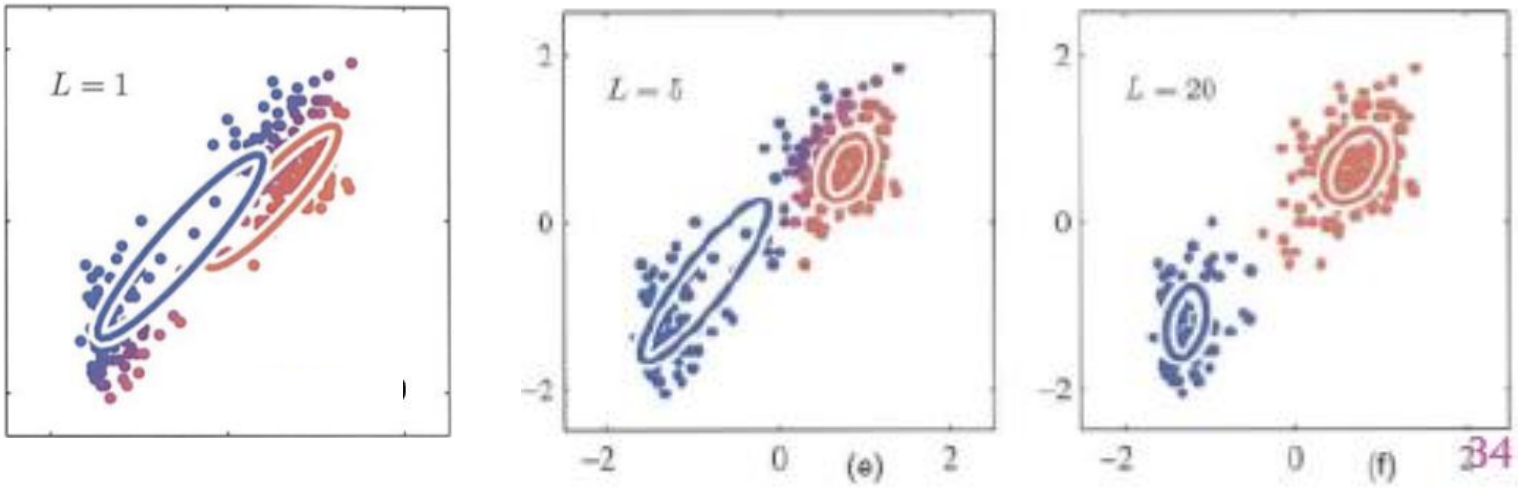
$$\mu_i^{new} = \frac{1}{N_i} \sum_{j=1}^{n} \gamma(z_{ji}) x_j; \quad N_i = \sum_{j=1}^{n} \gamma(z_{ji}); \quad \pi_i^{new} \frac{N_i}{n}$$



EM Algorithm: Stopping criterion

- Evaluate log likelihood. If likelihood parameters converge, stop

$$\ln p(\mathbf{X}|\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi}) = \sum_{n=1}^{N} \ln \left\{ \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right\}$$



Mixture of Gaussians Demo

- Search for Optimal Cluster Assignments

- Examine Output
  - With Class
  - Without Class
  - Compare Visualization

Anomaly Detection

Anomaly (Deviation Detection)

Chandola, Banerjee, & Kumar (2009). “Anomaly Detection: A Survey,” ACM Computing Surveys, 41(3): 15-58.

Anomaly Detection

- Finding patterns in data that do not correspond to expected (normal) behavior
  - Also called outliers
  - Anomaly detection related to Novelty detecion
- Challenges
  - How do we define an anomaly more precisely? Is it a point behavior, a trend, a distinct pattern, and so on.
    - They are a function of the domain of interest
  - Hard to completely characterize nominal behavior; boundary between nominal and fuzzy is often not precise (context-dependent)
  - Most natural (and many artificial) systems designed to compensate for anomalies; anomalous situations are compensated for by internal feedback or control actions
  - Labeled data to train classifiers for anomaly detection often hard to find
    - Have to resort to semi-supervised methods
  - How do we deal with situations, such as noise?
- Anomaly detection: multi-disciplinary field
  - Statistics, machine learning, data mining, pattern recognition, image processing, engineering (systems dynamics, information theory, spectral analysis)

Anomaly Detection Applications

- Cyber-Intrusion detection
- Fraud detection
- Medical anomaly detection
- Industrial Damage Detection & Condition-based Maintenance
- Image Analysis
- Text anomaly detection
- Sensor Networks

Types of input data

- Data types
  - Binary
  - Categorical
  - Continuous
- Relationship among data elements
  - Point data – data instances not related
  - Sequence data – data linearly ordered
    - Time-series data
    - Genome and protein data
  - Spatial data – concept of physical proximity, neighborhood (data can be spatio-temporal)
    - Vehicular traffic
    - Weather patterns

Types of Anomalies

- Point anomalies
  - Credit card fraud detection
- Contextual anomalies
  - Patterns extracted from a spatial region or a time sequence
    - Need contextual + behavioral attributes
  - Used a lot in time series applications
    - Fault detection
- Collective anomalies
  - Collection of data points represents an anomaly with respect to the entire data set
    - Example, a decreasing trend in time series data – each point is within bounds but the data points over time should be steady or increasing gradually

Anomaly Detection Methods

- Classification-based – supervised
  - Labeled instances for both nominal and anomaly classes
  - Approach – decision trees, Bayesian classifiers
  - Problems
    - Anomalous instances may be sparse – Imbalanced class distributions (SIGKDD Explorer 2004 – special issue on learning from imbalanced date)
    - Accurate instances of nominal versus anomalous behavior may be hard to find (Steinwart, et al. 2005, J. Machine Learning Research)
- Semi-supervised
  - Only nominal data labeled – often generated from a model of nominal operations for a system (Dasgupta and Majumdar, 2002)
  - Problems
    - Hard to find training data that covers range of nominal behaviors

Anomaly Detection Methods

- Clustering-based
  - No training data, widely applicable
  - Assumption in these algorithms – nominal data much more frequent than anomalous data
  - Nominal instances will have high likelihood of belonging (or be close in distance) to nominal cluster(s); anomalies will low likelihood (or be further away)
  - Algorithms – two step approach
    - Cluster
    - For new instances, check distances/likelihood
  - Smith et al. (2002) – comparison of 3 methods
  - Budalakoti, et al. (2006) – applied to time sequence data
  - Sometimes anomalous data may form clusters by themselves (He, et al, 2003)
    - Anomalous data – sparse clusters
    - May require expert interpretation

Anomaly Detection Methods

- Nearest neighbor algorithms
  - Like clustering algorithms, but base their analysis on a local neighborhood
  - k-nearest neighbor techniques
- Statistical
  - Nominal data from high density regions of space; anomalous from low density
  - Parametric
    - Gaussian
    - Regression Analysis
  - Non parametric methods
    - Histogram & kernel functions
- Spectral
  - Embed data in low dimensional space; use transformations that highlight differences in data
  - Principal Component Analysis (PCA)
- Information-Theory
  - Irregularities in data – detected by measures, such as Kolmogorov Complexity

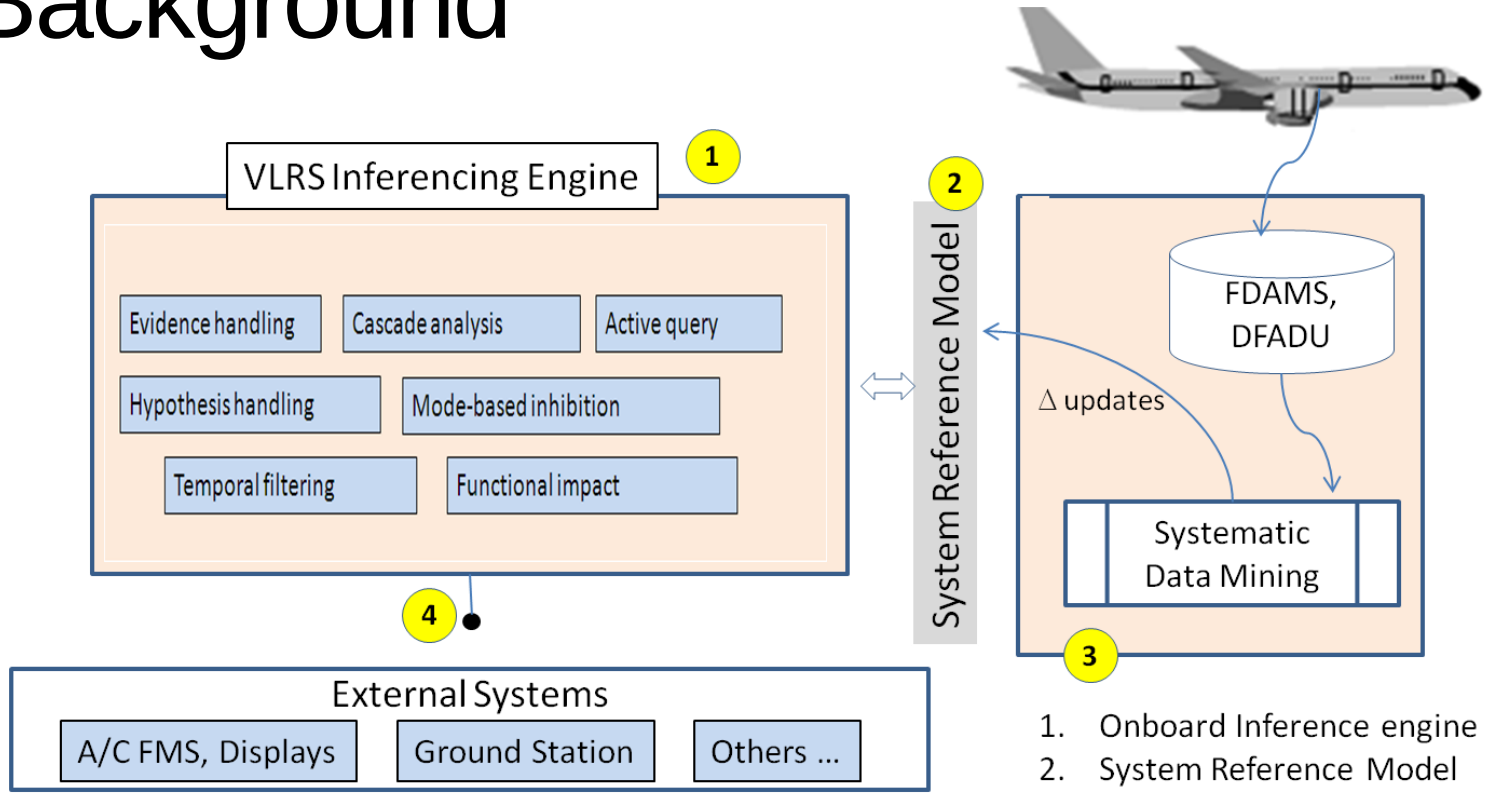
Case Studies

Improving Aircraft Diagnosis Reference Models and Reasoner Performance

Case Studies – Aircraft Diagnosis

- Vehicle Reasoning
  - Expert Models
  - Balance Simplicity and Completeness
- Data Collection
  - Raw Values
  - Large Data
- Improve Expert Models
  - Work with Expert
  - Combine Data with Model
  - Design Methodologies

Background

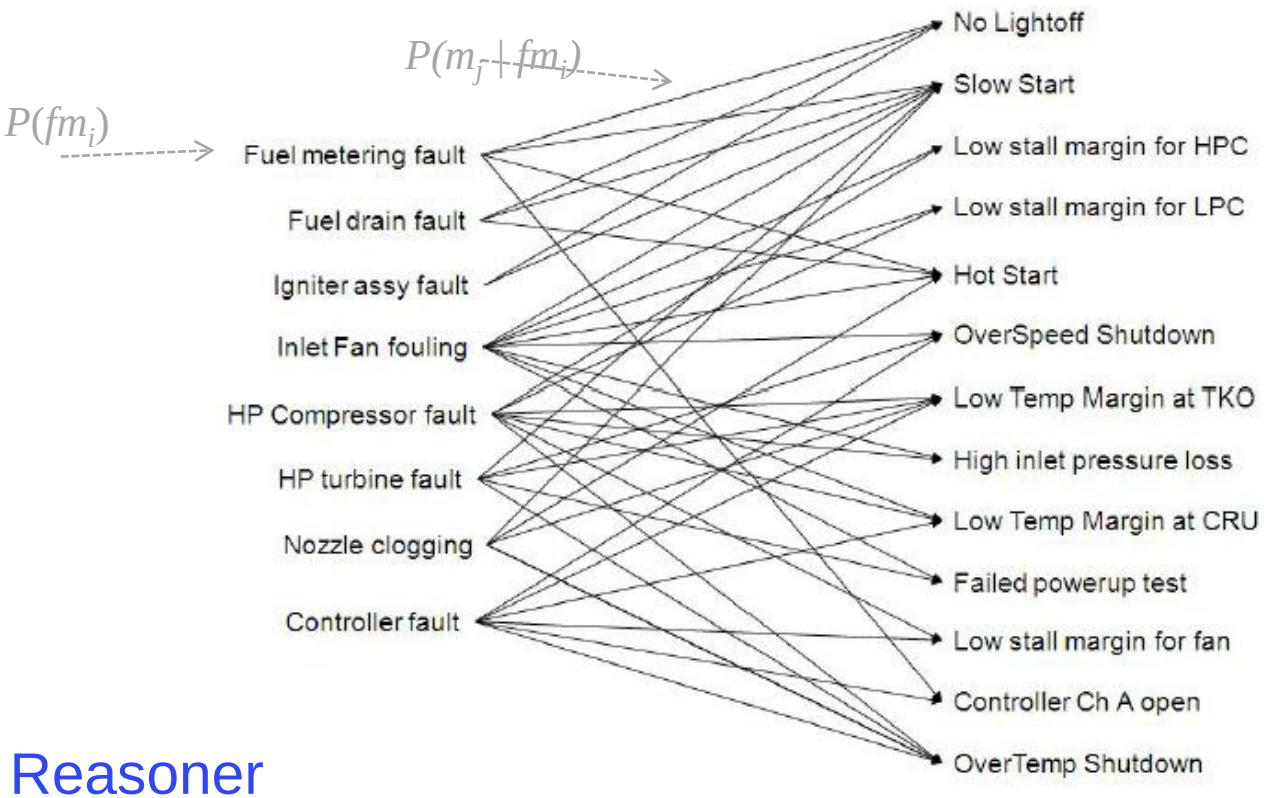


Broadly consists of (a) aircraft condition monitoring function – ACMF, (b) central maintenance computer function - CMCF. Both configured through a reference model

- ACMF generates on/off evidence based on pre-defined trigger rule
- CMCF operates on binary evidence to calculate the most probable cause

VLRS: Reference Model + Reasoner

- Example Reference Model



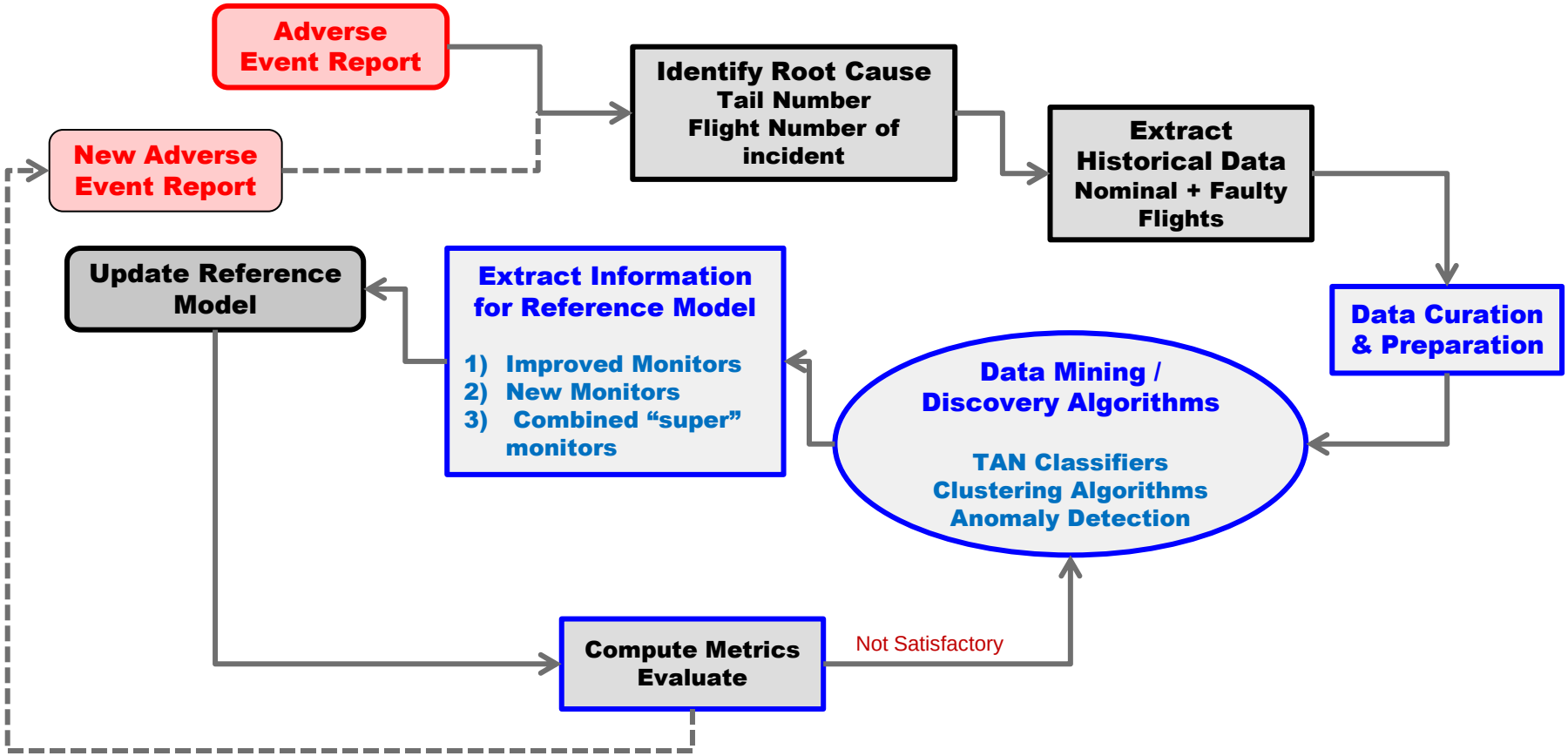
- Reasoner
  - For computing likelihood of fault, assume Naïve Bayes model:

$$P(fm_i \mid m_i, m_j, \ldots, m_k) = \alpha \times P(m_i \mid fm_i) \times P(m_j \mid fm_i) \times \ldots\ldots \times P(m_k \mid fm_i)$$

and $fm_1, fm_2, \ldots\ldots, fm_n$ do not interact.

Data Mining / Learning Loop 1

- Achieve continual improvement in performance of VIPR reference model



Supervised Learning Loop

Case Study Overview

- Start with a Subsystem
  - Develop a Method
    - Using Knowledge Discovery Loop
- Explore Other Subsystems
  - Verify effectiveness of Method
  - Gain Perspective on Structures
  - Gain Perspective on Data Mining Tools
- Abstract to Higher Levels of Diagnosis
  - Examine Viability of Methodology
  - Understand how it impacts Subsystem Models

Case Study 1 and 2 – Subsystem Exploration

- Engine Failures
  - Improve General Engine Model
  - Explore the Nature of an Engine Failure
- What is needed?
  - Plenty of Data (nominal and faulty) from Regional Airline Database
  - Clear Timeline
  - Domain Knowledge about fault and related CI's and monitors
- Goal
  - Understand the Methodology
    - The General Steps
    - Human in the Loop
- Twist: One fault at a time to Multiple Faults
  - Improve Understanding with More(Diverse) Data

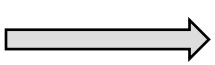
Case Study 1: Fuel HMA

- Isolate Event from Data and Annotation
  - Over speed and over temperature engine #3 shutdown
- Understand the Nature of Event
  - Engine Shutdown
  - Isolate Component
    - Examine Raw Data
      - Graphically
    - Establish Root cause: Fuel HMA (Hydromechanical unit) sluggish
- Domain Knowledge obtained from human expert
  - Likely Manifestation Time: ~50 Flights
- Extract Monitors and CIs related to Fault
  - Sample Reference Model
  - Three Phases of Monitors
- Extract Data From Database
  - Simple Script

Airline Data: Feature Transformation

- Which Features to Use?
  - Over 180 sensors in the raw data
  - How well will features correlate with the Reference Model and be sensitive to the fault ?
- Option 1: Diagnostic Monitors
  - Binary valued features from reference model
  - May suffer from loss of information because of the abstraction
- Option 2: Condition Indicators(CI)
  - Extend classifier variables to include CIs
  - Functions on raw sensor values
  - Computed for Phases of Operation
  - Apply threshold to produce diagnostic monitors

Data Transformation Path



Conditional Indicators

Diagnostic Monitors

Raw Parameters
Engine 1 Speed
Engine 2 Speed
Engine 3 Speed
Engine 4 Speed
Core Speed Engine 1
Core Speed Engine 2
Core Speed Engine 3
Core Speed Engine 4
Air Temperature
Engine 1 Exaust Gas Temperature
Engine 2 Exaust Gas Temperature
Engine 3 Exaust Gas Temperature
Engine 4 Exaust Gas Temperature
Flight Phase
Altitude

Startup Indicators
StartTime
IdleSpeed
peak Engine Temperature
Core Speed at Peak
StartSlope
StrtCutOff
LiteOff
prelit Engine Temperature
phaseTWO
timeToPeak

TakeOff Indicators
peak Core Speed
peak Engine Speed
peak Engine Temperature
takeoff Core Speed
takeoff Engine Speed
takeoff Air Temperature
takeoff Altitude
takeoff Engine Temperature
takeoff Margin

Rolldown Indicators
Rolltime
resdTemperature
dip Engine Temperature
Corespeed at Dip
Corespeed Slope
Corespeed Cutoff

- no Start
- slow Start
- Hung Start
- Hight Temp
- multStart
- phOneDwell
- hotStart
- medTempMargin
- lowTempMargin
- overSpeed
- overTemp
- abruptRoll
- highRollEGT
- rollBearing

Case Study 1: Fuel HMA

- Data Information
  - 50 Flights
  - 3rd Engine In Set Labeled Faulty
  - Other Labeled Nominal
  - 200 Samples in 50 Flights
  - Other Flights After Labeled Nominal
- Utilize Bayesian Learning – Derive TAN structure
- Results from Demo

Case Study 1: Fuel HMA

- Generality of Classifier
  - Engine 1 vs. Engine 3
  - Fault vs. Nominal
- Examining the Proximity of Data
  - Earlier Detection
- Split Data into Bins
  - Further away from the Fault
  - Test on Holdout Set(remainder of bins + more nominal)
- Examine the Bin
  - Accuracy
  - FP
  - Structure

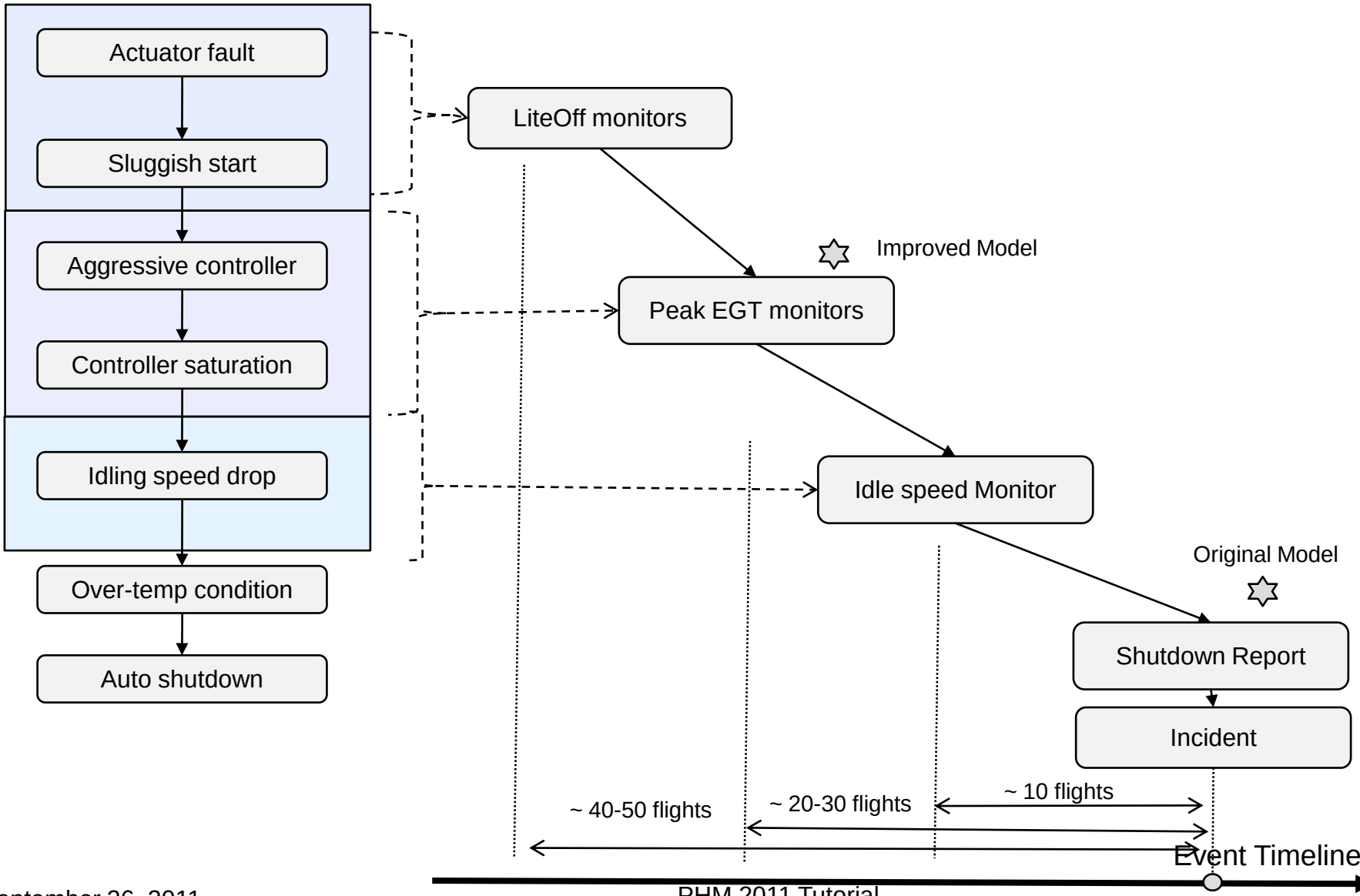
Case Study 1: Fuel HMA

- Results from Binning

Bin	Training Flights	Acc.on Holdout Set	FP%	Obs. Root Node	Children of ORN	Notes
1	1 to 10	97.65%	2.30%	IdleSpeed	StartTime	Thresholds Chosen from this Bin due to low FP
2	11 to 20	93.90%	5.70%	peakEGTC	liteOff,dipEGTC	peakEGTC Important Node
3	21 to 30	94.65%	5.30%	peakEGTC	liteOff,dipEGTC	peakEGTC Important Node
4	31 to 40	96.62%	3.50%	startTime	peakEGTC	Links startTime and PeakEGTC
5	41 to 50	96.06%	4.10%	liteOff	phaseTwo,RollTime	Links Startup and Rolldown CI

- Select Bin 1
  - For Monitor Updates
- Notice startTime and peakEGTC
  - Form Super Monitor

Anatomy of the FuelHMA incident & impact on Reasoner



Case Study 1 to Case Study 2

- For consideration
  - Context
  - “Nominal” Data

- Results for FuelHMA fault

Accuracy	False Positives	False Negatives
99.6%	0.5%	0%

- How well will this method work for other faults?
- What if more data is available?
  - Understand of Robustness(or Lack thereof)

- Look at Case Study 2

Case Study 2: Power Turbine Blade

- Adverse Event
  - NUMBER FOUR ENGINE DEVELOPED A VIBRATION
  - CREW SHUT DOWN THE ENGINE, DECLARED AN EMERGENCY.
  - POWER TURBINE BUCKET HAD A MISSING BLADE.
- Domain Knowledge
  - Fault: Power Turbine Missing Blade: Result: Excessive Vibration
  - Manifestation Time: ~50 Flights
- Follow Methodology
  - Features: Utilize Same Monitors
    - Initial
  - Classifier
  - Bins
- Extract Data and Derive Classifier
  - Classifier Results

Case Study 2: Power Turbine Blade

- Results of Binning 50 flights into 5 bins

Bin	Flights	Acc	FP	ORN	Ch of ORN
1	1 to 10	90.625	4.2	Starttime(Slow)	Everything but Rolltime and DipEGTC
2	11 to 20	92.5	2.5	Starttime(Slow)	Everything but N2atPeak,peakN2,tkoN2, dipeEGTC
3	21 to 30	87.5	5	Starttime(Slow)	Everything but N2atPeak
4	31 to 40	88.125	12.5	Starttime(Slow)	Everything but startslope and rolltime
5	41 to 50	85.625	11.7	Starttime(Slow)	Everything but startslope,peakEGTC,rolltime,resdtemp,N2atPeak and dipEGTC

- Next Steps

Case Study 1 and 2 - Combined

- Added to Methodology
  - Context
- New Information with the Fuel HMA TAN

Learned Fault	Fuel HMA ACC	Fuel HMA FP	Fuel HMA FN	Turbine Blade Acc	Turbine Blade FP	Turbine Blade FN
Fuel HMA	99.60%	0.50%	0%	95.93%	4.10%	0%

- Turbine Vibration TAN

Learned Fault	Fuel HMA ACC	Fuel HMA FP	Fuel HMA FN	Turbine Blade Acc	Turbine Blade FP	Turbine Blade FN
Turbine Blade	85%	15%	0%	92.18%	2.10%	25%

- Lessons Learned
  - Features

Case Study 3: System-level fault

- Event Information
  - A NUMBER ONE ENGINE FIRE WARNING ILLUMINATED AFTER
  - FUEL MANIFOLD WAS LEAKING FUEL,
- What can we know?
  - Manifold Fault
  - Engine One
  - Manifestation time?
- Methodology
  - Using Existing Features
    - Label Engine 1 Faulty

Case Study 3: System-level fault

- Context with other Data

- Build TAN and Examine on Other Data Sets?

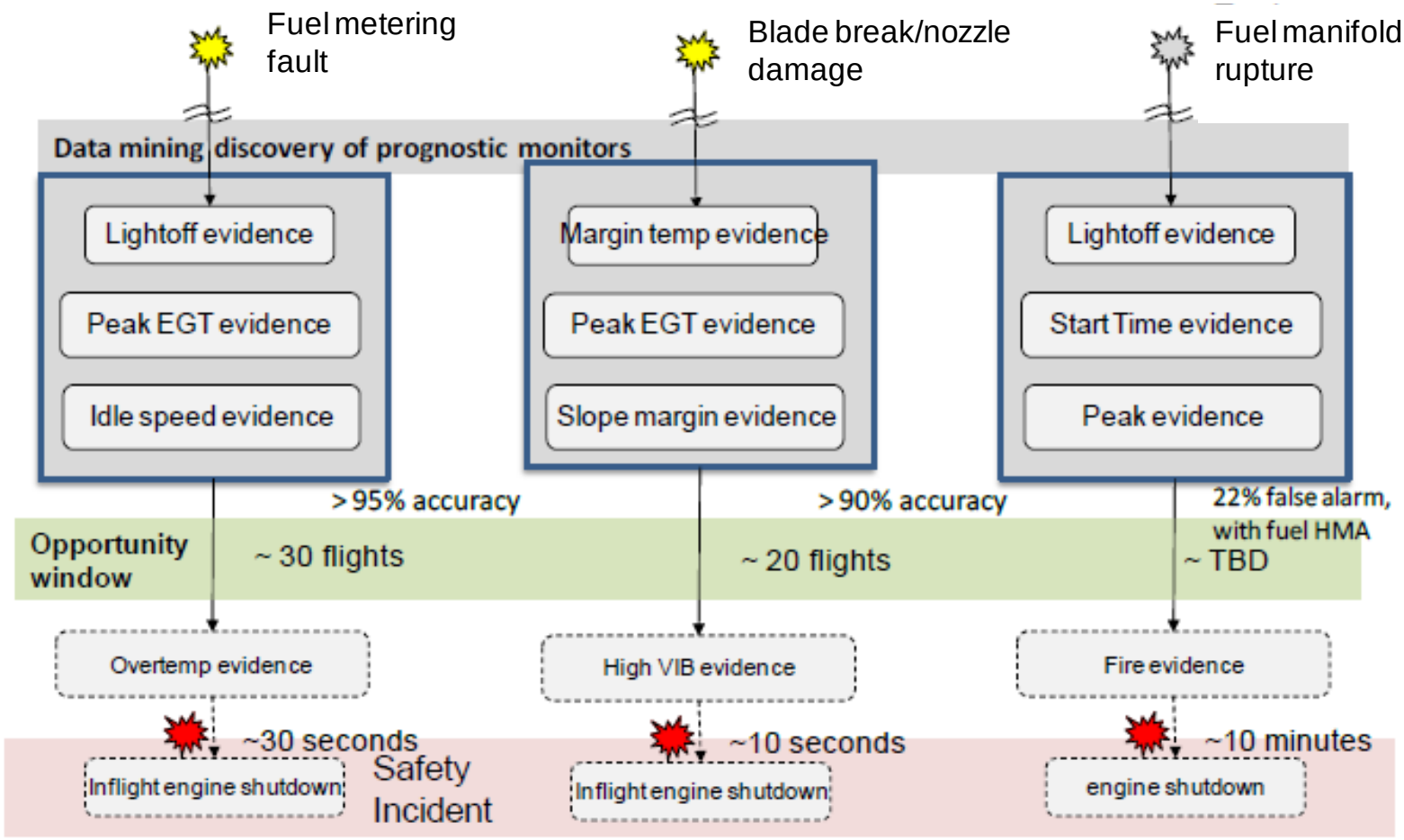
LearnedFault	Fuel HMA ACC	Fuel HMA FP	Fuel HMA FN	Fuel Manifold ACC	Fuel Manifold FP	Fuel Manifold FN	Vibration Acc	Vibration FP	Vibration FN
Fuel HMA	99.60%	0.50%	0%	97.18%	2.80%	0%	95.93%	4.10%	0%
Fuel Manifold	77.50%	22.50%	0%	90.31%	5.40%	22.50%	44.38%	55.60%	0%
Turbine Blade	85%	15%	0%	91.88%	8.10%	0%	92.18%	2.10%	25%

- Consult with Domain Expert
  - Manifold Issues – system level
  - Multi fault analysis

Case Study 3: System-level fault

- Multiple Engines
  - Initial Thoughts
- Unsupervised Methods
  - Clustering for grouping anomalies
- Using Discovered Knowledge

Summary: Three case studies

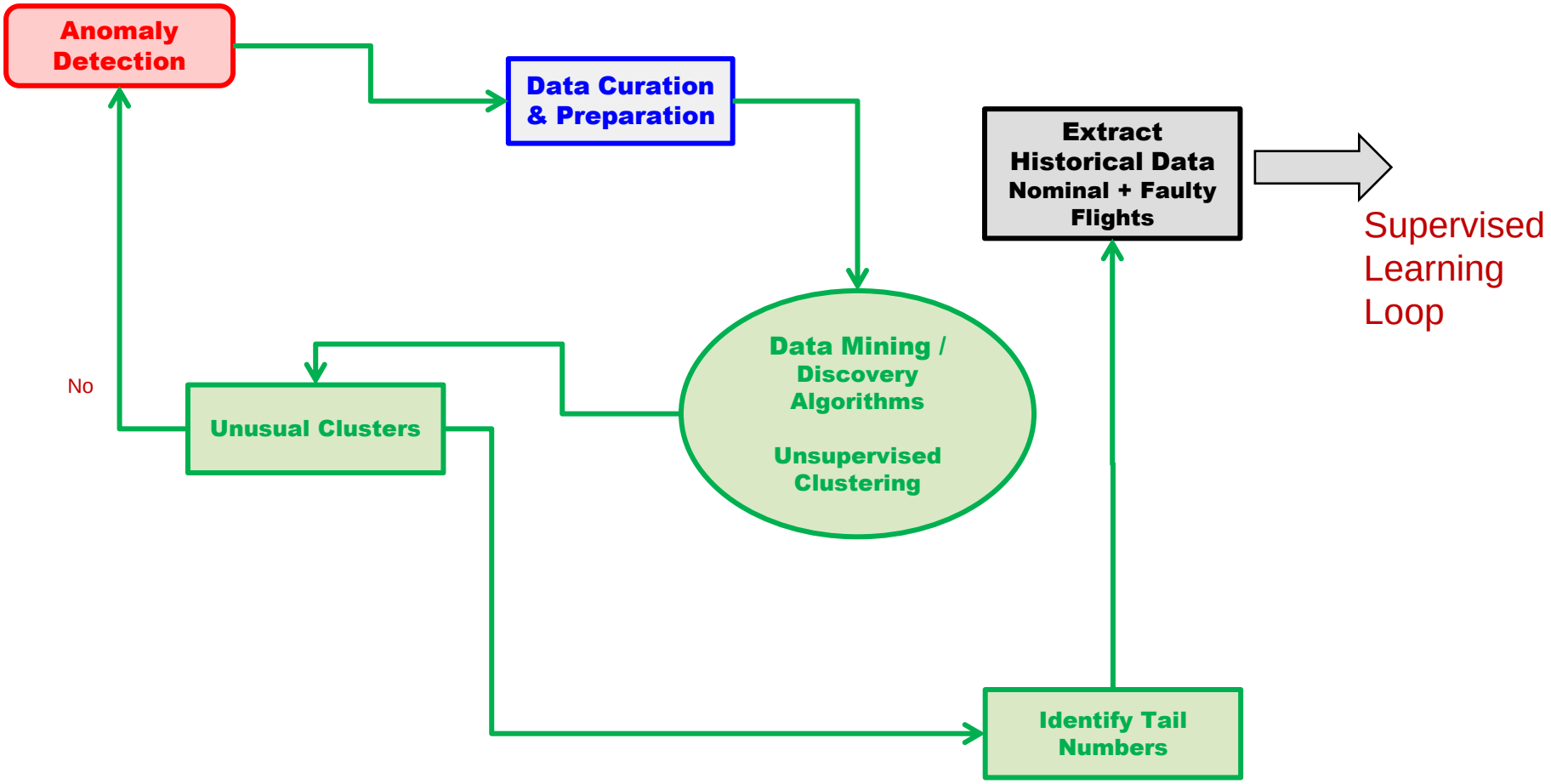


Studies 1 through 3

- Single subsystem (engine) model vs. more global system model
  - At system level, fault may affect multiple subsystems … have to reconcile monitors from multiple subsystems
- Understanding what is missing
- Using a Naïve approach to the model
- Looking at Other Subsystems?

Data Mining / Learning Loop 2

- Unsupervised Learning methods for 'Novel' Anomaly detection



Case Study 4: Introduce Navigation data

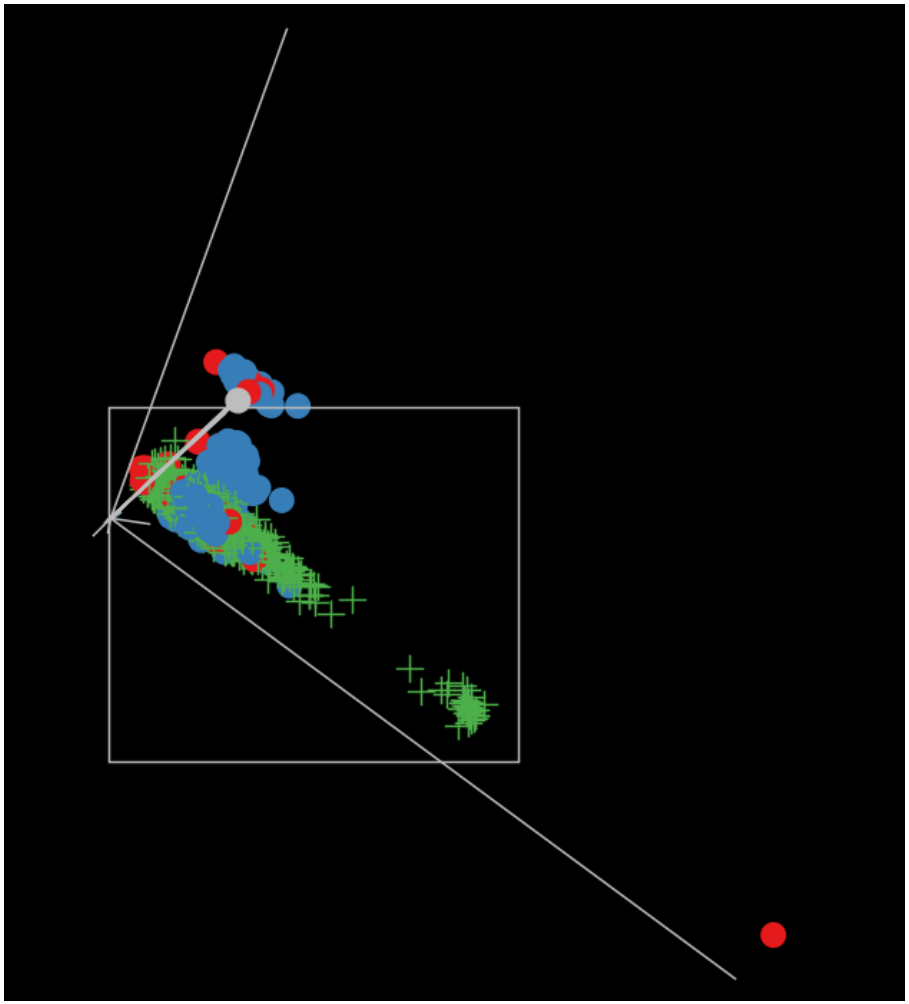
- Multiple Subsystems
  - Engine + Nav data
    - Can the combined data tell us more?
    - Will the combined data improve detection metrics and help resolve faults more accurately?
- Good Navigation from Bad Navigation
  - Have to discover fault from no fault situation
    - Extend analysis to anomaly detection schemes
- Understand Flight Profiles
- Examine Implicit Structure
  - Can we extract relations between subsystems?

Case Study 4 Conclusions

- Normal Operation
- Several “Abnormal” Modes

- Visualization
  - Projection Plots
  - X-Y Coordinate Analysis

- Initial Results
  - Understand “Abnormal” Flights



Next Steps and Future Work

- Single fault to Multi-fault analysis
  - Extend from Naïve Bayes calculations to more complete Bayes net calculations – should produce more accurate likelihood of multiple fault hypotheses
- Work with human experts to analyze conditional probabilities produced by TANs into conditional probabilities in reference models
  - Tricky issue because classifier conditional probabilities are a function of initial probability distributions
- Extend single subsystem analysis to system level analysis
- Deal with cascading faults
- Extend supervised learning methods to unsupervised learning to address anomaly detection problem

References (1)

- I. H. Whitten, E. Frank, M.A. Hall (2011). **Data Mining: Practical Machine Learning Tools and Techniques**, Elsevier (Morgan Kaufmann), Boston, MA.
- Han and Kamber (2006). **Data Mining: Concepts and Techniques**, Morgan Kaufmann, San Mateo, CA.
- Mitchell, T.M. (1997) **Machine Learning**. McGraw Hill, International Editions,
- Pearl, J. (1988). **Probabilistic reasoning in intelligent systems: networks of plausible inference**. Morgan Kaufmann Publishers, San Mateo, CA.
- Jain, A. & Dubes, R.C. (1988) **Algorithms for Clustering Data,** Prentice Hall, Englewood Cliffs, NJ.
- Schlimmer, J. C. (1987). Concept acquisition through representational adjustment. *Doctoral dissertation*, Department of Information and Computer Science, University of California, Irvine, CA.
- UCI Repository: http://archive.ics.uci.edu/ml/datasets/
- Hall, M., Eibe, F., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009). The WEKA Data Mining Software: An Update. *SIGKDD Explorations*, 11(1): 10-18.

- R. Agrawal; T. Imielinski; A. Swami (1993) Mining Association Rules Between Sets of Items in Large Databases", *SIGMOD Conference* 207-216.

References (2)

- C. Strobl, A.L. Boulesteix, T. Augustin (2006) Unbiased split selection for classification trees based on the Gini Index, *Computational Statistics and Data Analysis*.
- Chickering, D. M., Heckerman, D., & Meek, C. (1997). A Bayesian approach to learning Bayesian networks with local structure. In *Proceedings of Thirteenth Conference on Uncertainty in Artificial Intelligence*. Morgan Kaufmann
- Cohen, I., Goldszmidt, M., Kelly, T., Symons, J., & Chase, J. S. (2004). Correlating instrumentation data to system states: a building block for automated diagnosis and control. In *Proceedings of the 6th conference on Symposium on Operarting Systems Design & Implementation* , Berkeley, CA, USA: USENIX Association, 6:16–30.
- Cheng, J., Greiner, R., Kelly, J., Bell, D., & Liu, W. (2002). Learning Bayesian networks from data: An information-theory based approach. *Artificial Intelligence*, 137(1-2): 43 – 90
- Kruskal, J. & Joseph B. (1956). On the Shortest Spanning Subtree of a Graph and the Traveling Salesman Problem. Proceedings of the American Mathematical Society, 7(1): 48-50.
- Vapnik V., **Statistical learning theory**, John Wiley, New-York, 1998.
- Spitzer, C. (2007). Honeywell Primus Epic Aircraft Diagnostic and Maintenance System. Digital Avionics Handbook(2), pp. 22-23.
- Kaufmann, L. & Rousseeuw, P.J. (1990) **Finding Groups in Data – An Introduction to Cluster Analysis**, Wiley,
- MacQueen, J.B. (1967) Some methods for classification and analysis of multivariate observations. Proc. 5th Berkeley Symposium on Mathematical Statistics and Probability, 281-297,
- Ng & Han (1994): Efficient and Effective methods for spatial data mining. Proc. 20th VLDB conference, Santiago, Chile, 144-155.
- Ester et al. (1996) A density-based algorithm for discovering clusters in lage spatial databases with noise. Proc. Intl. Conf. on Knowledge Discovery and Data Mining, Portland, OR, AAAI press, 226-231.