Big Data Analytics in VLSI Design Automation and Test –

Principles, Challenges, and Promises

Li-C Wang
University of California, Santa Barbara

Introduction
Basic Concepts

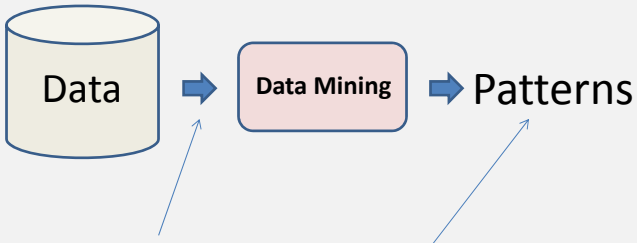
Recent Tutorial Papers

- **"Data Mining in EDA" – Design Automation Conference (DAC) 2014**
  - http://mtv.ece.ucsb.edu/licwang/PDF/2014-DAC.pdf
- **"Data Mining in Functional Debug" – ICCAD 2014**
  - http://mtv.ece.ucsb.edu/licwang/PDF/2014-ICCAD.pdf
- **"Data Mining in Functional Test Content Optimization" – ASPDAC 2015**
  - http://mtv.ece.ucsb.edu/licwang/PDF/2015-ASPDAC.pdf
- **"Machine learning in simulation based analysis" – ISPD 2015**
  - http://mtv.ece.ucsb.edu/licwang/PDF/2015-ISPD.pdf
- **For general reference: http://mtv.ece.ucsb.edu/licwang/**

Data Mining

- **Data mining is the process of extracting (statistically significant) "*patterns*" from the data**
- **"Pattern" – Something that does not appear just once**

Two Fundamental Questions



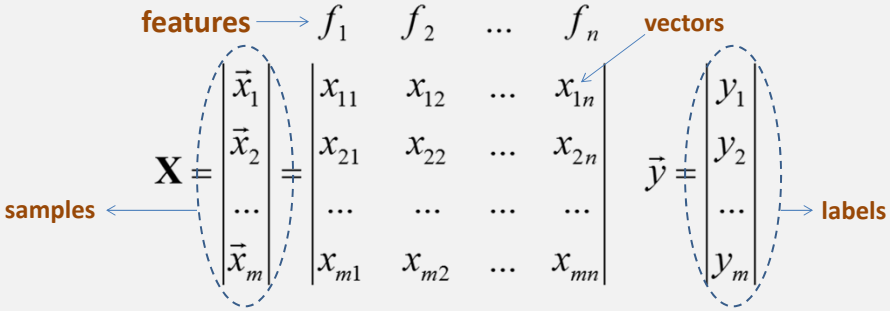
- How to feed the data to a data mining tool box?
- How to represent the "Patterns?"

Li-C. Wang Tutorial - April 24th, 2015 5

For Example: http://scikit-learn.org/

Li-C. Wang Tutorial - April 24th, 2015 6

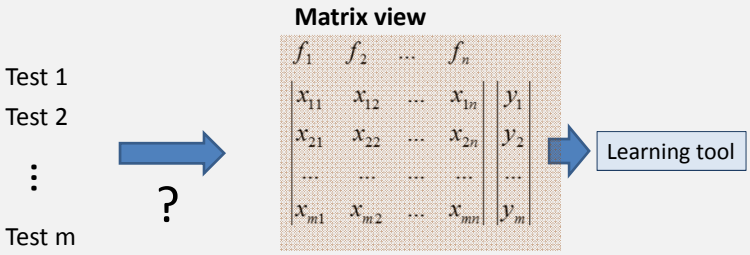
Expected Matrix View of Data



- **A learning algorithm usually sees the dataset as above**
  - Samples: examples to be reasoned on
  - Features: aspects to describe a sample
  - Vectors: resulting vector representing a sample
  - Labels: care behavior to be learned from (optional)

Li-C. Wang Tutorial - April 24th, 2015 7

Fundamental Question 1

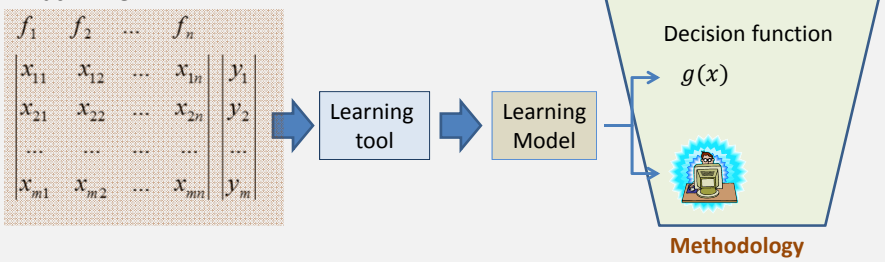


- **A learning tool takes data as a matrix**
- **Suppose we want to analyze *m* tests (assembly programs)**
  - Each test comprises 100-150 instructions
- **How do I apply learning on objects which are not represented as a vector?**

Li-C. Wang Tutorial - April 24th, 2015 8

Fundamental Question 2

Matrix view

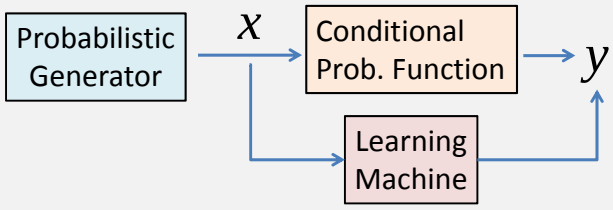


- **How the learning model is represented?**
  - Tree, rule, neural network, SVM, etc.
- **Depend on what the learning model is used for**
  - As a decision function used by another tool (**prediction**)
  - As interpreted by a person (**interpretation**)
- **One needs to develop the methodology to enable the utilization of the learning model**

Theoretical Hardness

- **Many Design Automation and Test problems have complexity NP-hard or beyond**
- **Data mining does not make an NP-hard problem easier**
- **Theoretical hardness**
  - Learning a 3-term DNF formulae is NP-hard
- **If not for solving a difficult problem, what is it good for?**

Learning Setup – Why It Is Hard



Function estimation model (Supervised)

- **(Probable) - Guaranteed probability the learning machine produces the "desired" output**
- **(Approximately Correct) – The "desired" output has a guaranteed error bound with respect to the correct output**
- **If you demand both (PAC Model), a learning problem can be hard**
- **You can make a learning task easier by incorporating more domain knowledge**

Li-C. Wang Tutorial - April 24th, 2015 11

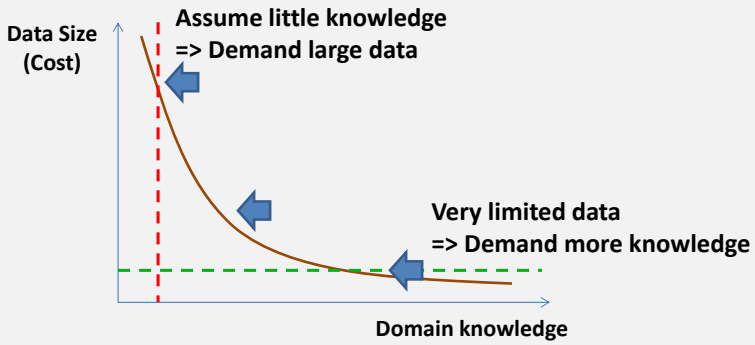
Can't Learn Without Knowledge

Learning = Data + Knowledge

- **In theory, learning from data is not possible without some domain knowledge**
  - e.g. In its basic form, one assumes i.i.d. of the data
- **Some domain knowledge is always required**
- **In our domain of applications, since usually we don't have "large" datasets, domain knowledge becomes important**

Li-C. Wang Tutorial - April 24th, 2015 12

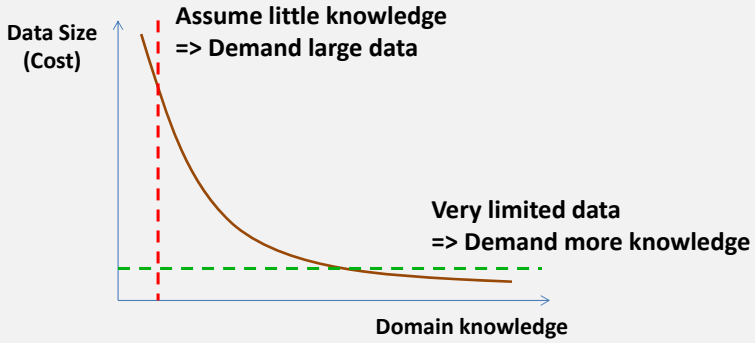
Learn = Data + Knowledge



- **Big data applications**
  - Social network mining, marketing data mining
- **Medium data applications**
  - Production test data mining (Terabytes of data)
- **Small data applications**
  - Verification and validation (simulation data)

Li-C. Wang Tutorial - April 24th, 2015 13

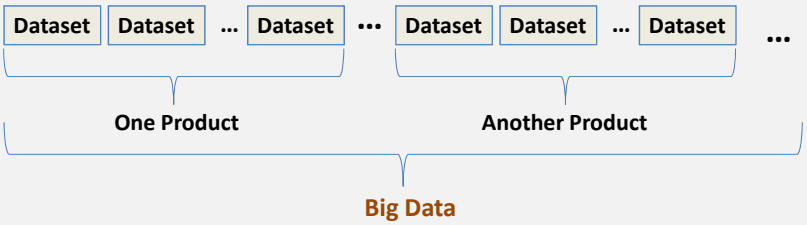
Learn = Data + Knowledge



- **Applying data mining in an application:**
  - Developing a methodology enabling an optimal tradeoff between knowledge requirement and data requirement

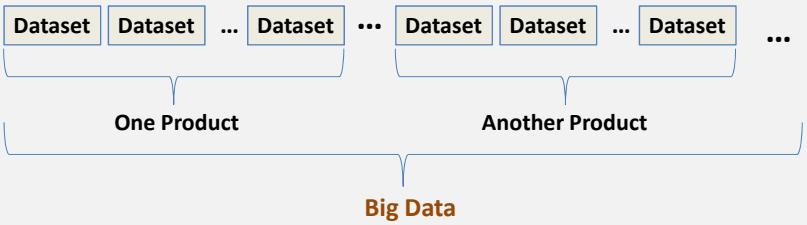
Li-C. Wang Tutorial - April 24th, 2015 14

Big Data?



- **Individual Task level**
  - e.g. Simulation traces from a specific testbench, one lot of test data
- **Individual Product level**
  - e.g. Simulation traces from multiple testbenches, one year of test data
- **Cross-product level**

Big Data Question?



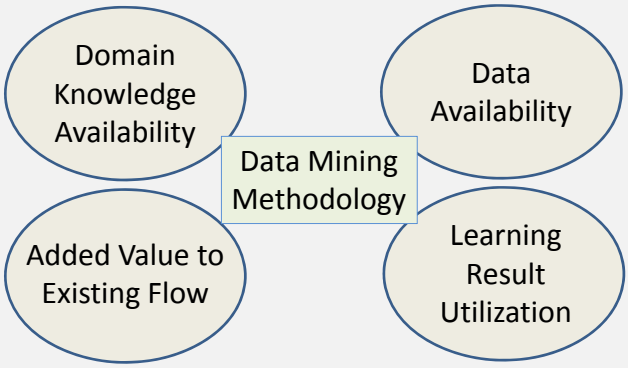
- **Task level question**
  - e.g. How to hit this bug? How to screen this die?
- **Product level question**
  - e.g. What should be my regression test? How many analog tests to apply?
- **Cross-product level question**
  - e.g. Where can I cut simulation cost? Where can I cut test cost?

Time Sensitivity & Accuracy Constraints

- **Time sensitivity**
  - Time for learning depends on the application scenario
  - Can be from a few seconds to a few weeks
  - Keep in mind there is an existing flow to solve the same problem – can't be more costly than that
- **Accuracy**
  - What's the requirement for learning accuracy?
  - The higher the requirement, the harder the learning – usually this means more data (or knowledge) is required
  - If some guarantee is demanded, learning can become impractical

Li-C. Wang Tutorial - April 24th, 2015 17

Four Key Considerations



- **In this picture, I do not mention a "learning algorithm"**
- **Because to implement a successful practical data mining methodology, it involves integration of a collection of "simple" algorithms**
  - Individually simple, but collectively very complex

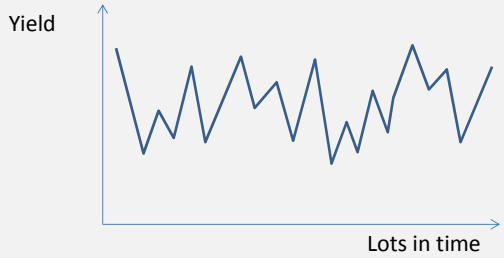
Li-C. Wang Tutorial - April 24th, 2015 18

General Principles

- **Infrastructure support - Make sure the data is available, or the cost to collect the data is acceptable**
- **Solve the right problem - Develop a methodology to incorporate as much domain knowledge as possible**
- **Involve human decision – Human involvement in the flow relaxes the requirement of guaranteed result**
- **Complementary - The approach has to provide added value to the existing flow**

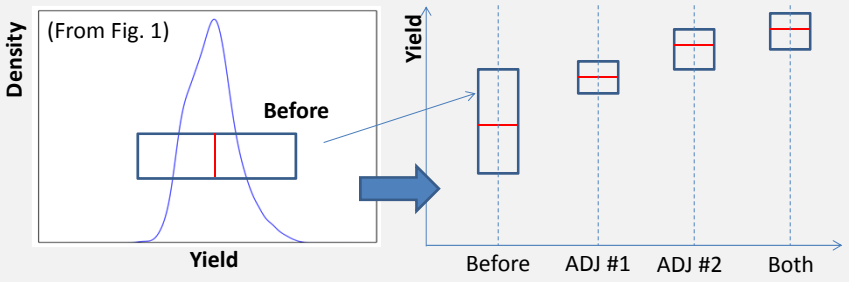
A Yield Improvement Example

Problem: (for illustration)



- **An automotive SoC product**
- **Yield fluctuated over time**
- **Product engineer had studied the problem for months but could not find a solution to fix it**
  - The design had gone through one revision of fix but did not solve the problem
- **Data: all the test data and e-test measurements**
- **Question: can you do better?**

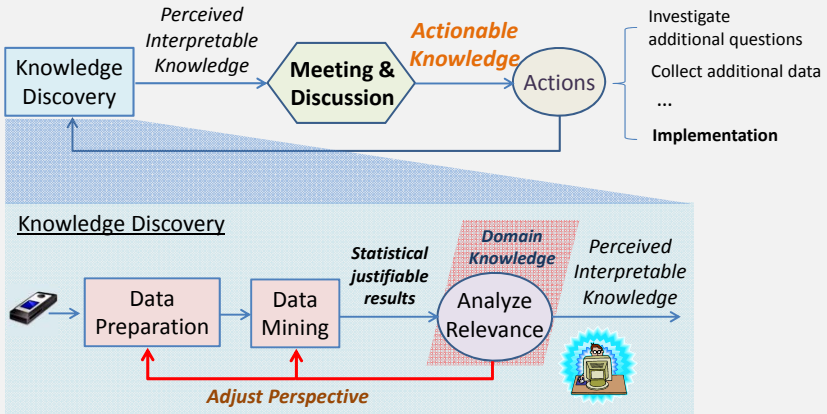
Yield Improvement – Silicon Result



- **ITC 2014: "Yield Optimization Using Advanced Statistical Correlation Methods"**
  - **http://mtv.ece.ucsb.edu/licwang/PDF/2014-ITC.pdf**

Li-C. Wang Tutorial - April 24th, 2015 21

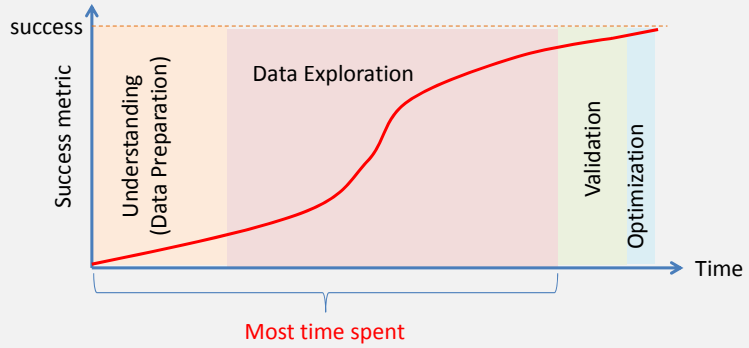
Yield Application: Data Mining = "Knowledge Discovery"



- **In practice, data mining is an iterative Knowledge Discovery process**
  - A search process for finding answers in a huge hypothesis space
  - Finding interpretable and actionable knowledge
- **Results summarized into a PPT used in meetings**
- **In this case, learning models need to be "simple"**
  - Tree, rules, or projectable to a 2/3-dimensional space for visualization

Li-C. Wang Tutorial - April 24th, 2015 22

Four Stages In A Knowledge Discovery Process



- **1. Understand the constraints and preparing the data**
- **2. Data exploration – search for the "right" perspective**
- **3. Validation with either data or domain knowledge**
- **4. Result optimization for applicability**
  - This step is where a sophisticated learning algorithm might make a difference

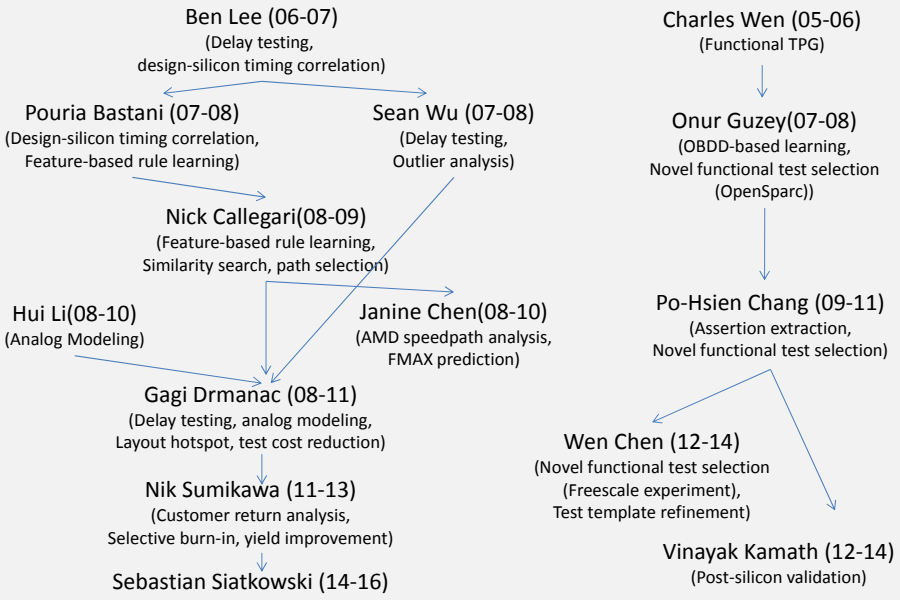
Li-C. Wang Tutorial - April 24th, 2015 23

Disclaimer and Students

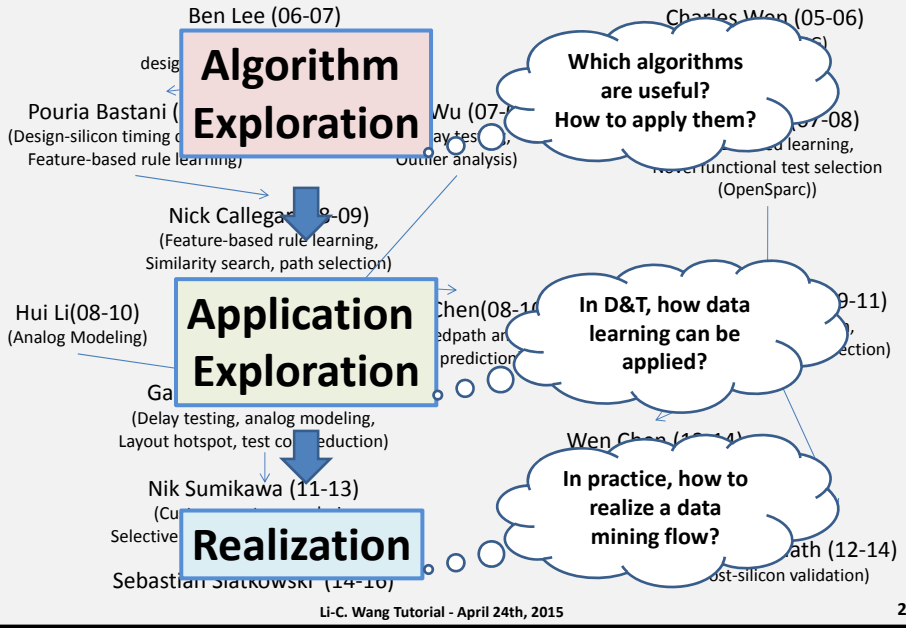
- **Disclaimer**
  - This tutorial is largely based on research works done by my students since 2006
  - It is not intended to be a survey of the field
- **Students (2006 – current)**
  - Ben Lee (Startup) - 2006
  - Charles Wen (NCTU, Taiwan)
  - Pouria Bastani (Intel)
  - Onur Guzey (Intel -> MIT -> Wall Street -> Istanbul Sehir University, Turkey)
  - Sean Wu (TPK, Taiwan)
  - Nick Callegari (nVidia)
  - Hui Lee (Intel)
  - Janine Chen (AMD)
  - Po-Hsien Chang (Oracle)
  - Gagi Drmanac (Intel)
  - Nik Sumikawa (Freescale) - 2013
  - Wen Chen (Freescale) - 2014
  - Vinayak Kamath (AMD) - 2014

Li-C. Wang Tutorial - April 24th, 2015 24

Students and Lines Of Works



Three Phases Of R&D

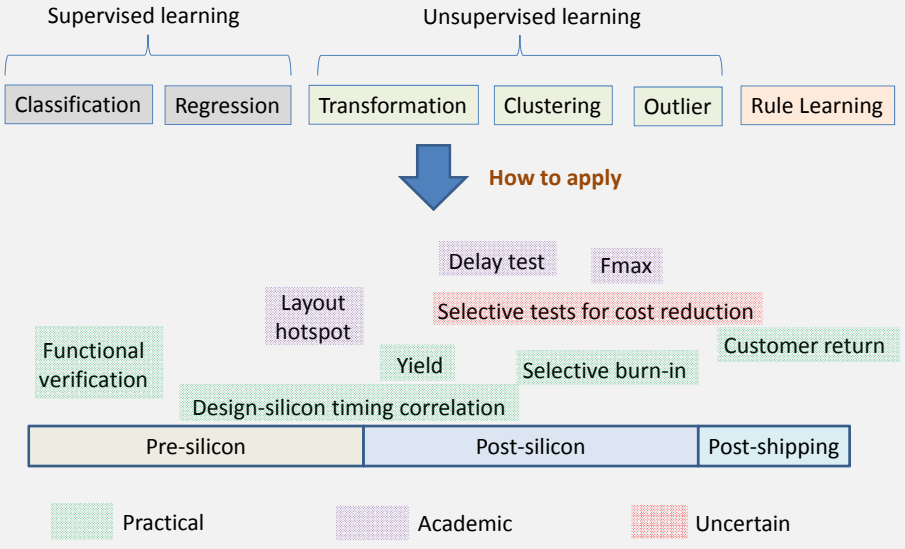


Plan For The Tutorial (6 hours = 360 minutes)

- **Opening (40 minutes)**
- **An introduction to data mining in design and test (120 minutes)**
  - Basic learning concepts and intuitions to algorithms
  - Example problem formulations and application contexts
- **Learning theory, SVM and Kernel Method (50 minutes)**
- **Application examples, working principals and findings (50 minutes)**
- **Application examples in pre-silicon simulation (50 minutes)**
- **Application examples in post-silicon testing (40 minutes)**
- **Final Remark and Questions (10 minutes)**

Li-C. Wang Tutorial - April 24th, 2015 27

Quick Overview

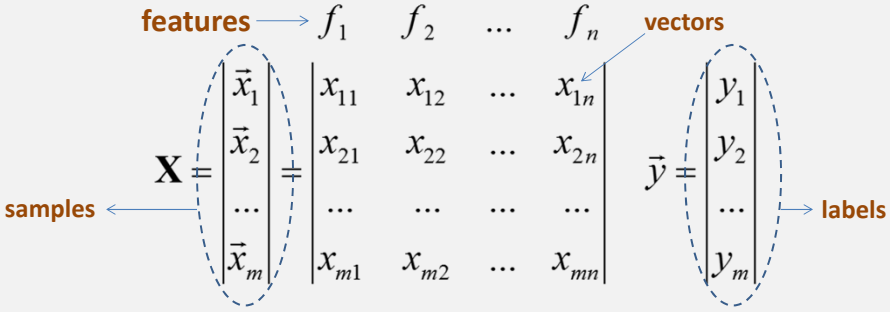


Li-C. Wang Tutorial - April 24th, 2015 28

An introduction to data mining and some applications in design & test

(120 minutes)

Data Mining 101



- **A learning algorithm usually sees the dataset as above**
  - Samples: examples to be reasoned on
  - Features: aspects to describe a sample
  - Vectors: resulting vector representing a sample
  - Labels: care behavior to be learned from (optional)

Data Mining Approaches

- Classification
- Regression
- Clustering
- Transformation
- Outlier Detection
- Density Estimation
- Rule Learning

Data Mining Approaches

- Classification
- Regression
- Clustering
- Transformation
- Outlier Detection
- Density Estimation
- Rule Learning

Data Mining 101 – Supervised Learning - Classification

$$\text{(features)}\quad f_1 \quad f_2 \quad \dots \quad f_n$$

$$\mathbf{X} = \begin{vmatrix} \vec{x}_1 \\ \vec{x}_2 \\ \dots \\ \vec{x}_m \end{vmatrix} = \begin{vmatrix} x_{11} & x_{12} & \dots & x_{1n} \\ x_{21} & x_{22} & \dots & x_{2n} \\ \dots & \dots & \dots & \dots \\ x_{m1} & x_{m2} & \dots & x_{mn} \end{vmatrix} \qquad \vec{y} = \begin{vmatrix} y_1 \\ y_2 \\ \dots \\ y_m \end{vmatrix}$$

Class labels

- **Classification**
  - There are labels y's
  - Each y's represents a class
- **For example, in binary classification, y= -1 or y = +1**

Example Learning Algorithms For Classification

- **Nearest Neighbors**
- **Linear Discriminant Analysis (LDA)**
  - Quadratic Discriminant Analysis (QDA)
- **Naïve Bayes**
- **Decision Tree**
  - Random Forest
- **Support Vector Machine**
  - Linear
  - Radius Based Function (RBF)

Example Learning Algorithms For Classification

- **Nearest Neighbors**
- **Linear Discriminant Analysis (LDA)**
  - Quadratic Discriminant Analysis (QDA)
- **Naïve Bayes**
- **Decision Tree**
  - Random Forest
- **Support Vector Machine (discussed later)**
  - Linear
  - Radius Based Function (RBF)

Li-C. Wang Tutorial - April 24th, 2015 35

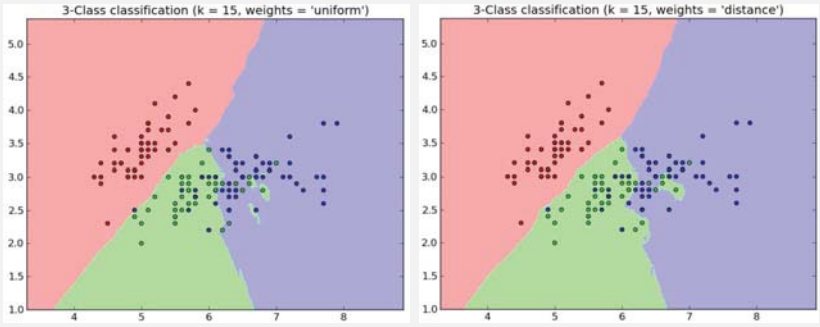
Nearest Neighbors

$y = f(x) =$ average of the k nearest neighbors to x

Uniform average or weighted by inverse of distance

User choose

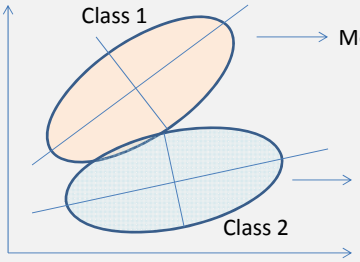
a given distance function In a given space



Source: http://scikit-learn.org/stable/auto_examples/neighbors/plot_classification.html#example-neighbors-plot-classification-py

Li-C. Wang Tutorial - April 24th, 2015 36

Linear Discriminant Analysis (LDA)



Model it as a Gaussian Distribution $N(\mu_1, COV_1)$

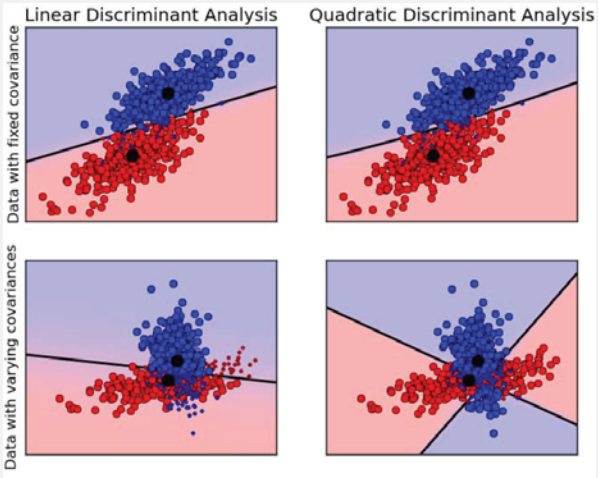
Model it as another Gaussian Distribution $N(\mu_2, COV_2)$

Decision function: $f(x) = log\frac{Prob(x \text{ in class } 1 \mid \text{given } x)}{Prob(x \text{ in class } 2 \mid \text{given } x)}$

- **For each class, the mean and covariance are estimated based on the data**
- **In LDA, the two covariances are assumed to be the same**
  - Otherwise, it is called Quadratic Discriminant Analysis (QDA)
- **In many cases, the difference between LDA and QDA is small**

Li-C. Wang Tutorial - April 24th, 2015 37

LDA vs. QDA



Source: http://scikit-learn.org/stable/auto_examples/plot_lda_qda.html

Li-C. Wang Tutorial - April 24th, 2015 38

Bayesian Inference – Naïve Bayes Classifier

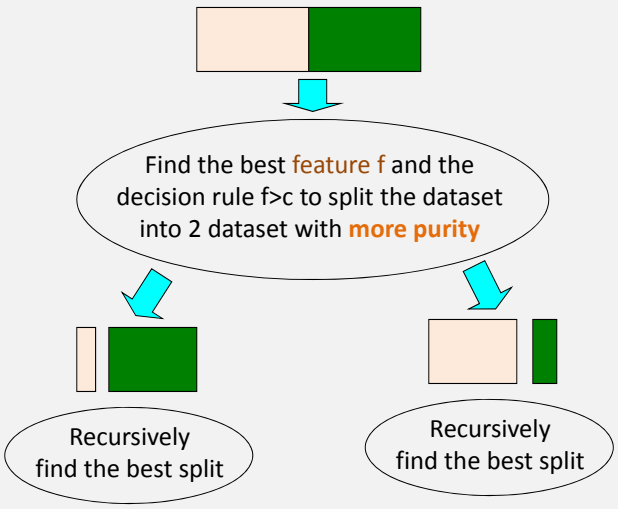
$$p(class \mid x_1,...,x_n) = \frac{p(class)\,p(x_1,...,x_n \mid class)}{p(x_1,...x_n)} = \frac{prior \times likelihood}{evidence}$$

$$p(class \mid x_1,...,x_n) \propto p(class)\,p(x_1,...,x_n \mid class) \propto p(class)\,p(x_1 \mid class)\cdots p(x_n \mid class)$$

Independent assumptions

- **The naïve Bayes classifier uses the assumption that features are mutually independent**
  - This is not usually not true as we have seen in the test data
- **Also, if each xi is a continuous variable, we either need to estimate the probability density, or we need to discretize the value into ranges**

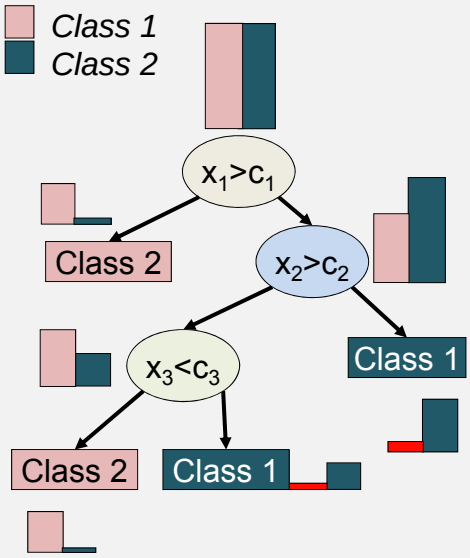
Decision Tree Classifier



- **An easy and popular learning algorithm CART (1984 Breiman et al.)**
- **Of course, the key question is how to measure "purity"**

CART Approach

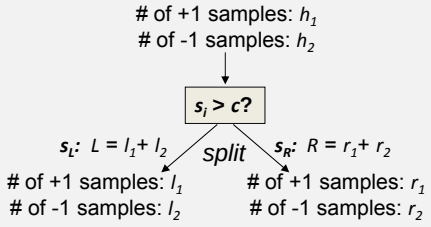
- **Randomly select $m^{1/2}$ variable to be tried at each split node**
- **Find the variable that split the data the best (purity meas.)**
- **Stop Criterion**
  1. The split has fully separated the subset
  2. None of the variable can further separated the subset anymore.



Li-C. Wang Tutorial - April 24th, 2015 41

Gini Index – impurity measure

- **Gini index - a measure of impurity of a dataset**
  - It is calculated before and after a node split
- **From Gini index the Gini importance impot(s_i) can be calculated**



$$Gini(s_i) = 1 - \left(\frac{h_1}{h_1+h_2}\right)^2 - \left(\frac{h_2}{h_1+h_2}\right)^2$$

$$Gini(s_L) = 1 - \left(\frac{l_1}{l_1+l_2}\right)^2 - \left(\frac{l_2}{l_1+l_2}\right)^2$$

$$Gini(s_R) = 1 - \left(\frac{r_1}{r_1+r_2}\right)^2 - \left(\frac{r_2}{r_1+r_2}\right)^2$$

$$impot(s_i) = Gini(s_i) - Gini(s_L) * \frac{L}{L+R} - Gini(s_R) * \frac{R}{L+R}$$

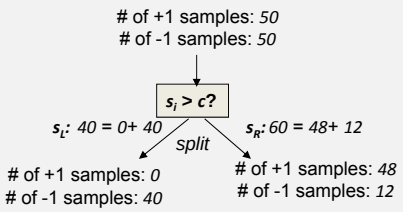
Li-C. Wang Tutorial - April 24th, 2015 42

An Example of Gini Index

- **Before split: 50-50**
  - equal number of +1 and -1's

$$Gini(s_i) = 1 - \left(\frac{50}{50+50}\right)^2 - \left(\frac{50}{50+50}\right)^2 = 0.5$$

- **After split:**
  - L=40, R=60
  - left tree contains only -1 class,
  - right tree has 48 of +1's, 12 of -1's



$$Gini(s_L) = 1 - \left(\frac{0}{0+40}\right)^2 - \left(\frac{40}{0+40}\right)^2 = 0$$

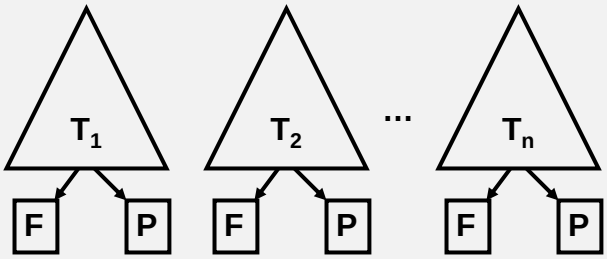
$$Gini(s_R) = 1 - \left(\frac{48}{48+12}\right)^2 - \left(\frac{12}{48+12}\right)^2 = 0.308$$

$$impot(s_i) = 0.5 - 0 * \frac{40}{40+60} - 0.32 * \frac{60}{40+60} = 0.308$$

Li-C. Wang Tutorial - April 24th, 2015 43

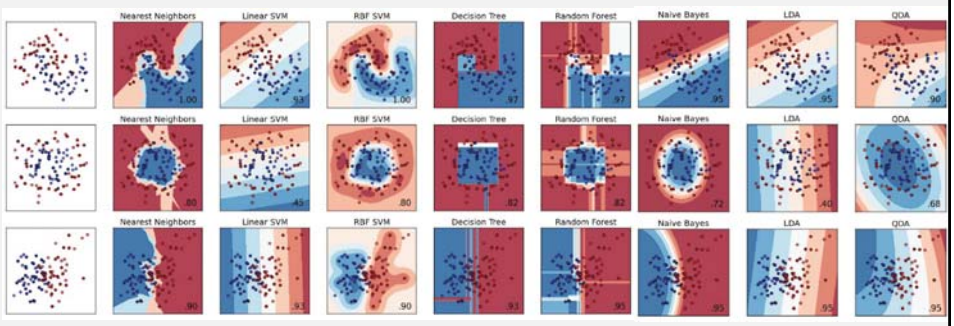
Random Forests

- **Ensemble learning: If you have n weak learners, together they can be strong – each tree is a weak learner (over-fitting the data)**
  - Build a collection of trees
- **Select a random set of (training) samples (2/3 subset)**
- **Grow a tree based on only the selected samples (in-bag data)**
- **Use the unselected samples (out-of-bag data) to validate the tree performance, i.e. prediction accuracy**
- **Grow many trees until the average accuracy saturates**
- **The prediction is based on votes from all trees (votes = confidence)**



Li-C. Wang Tutorial - April 24th, 2015 44

A Comparison of Classifiers

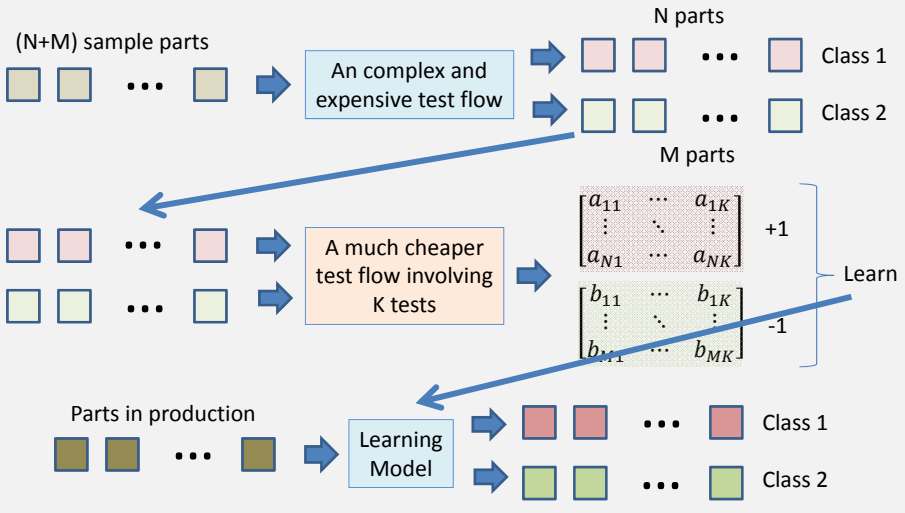


Source: http://scikit-learn.org/stable/

- **Algorithms are comparable on the 1st and 3rd examples**
- **Performance on the 2nd example varies**
- **In practical application, a more complex algorithm is not necessarily better**
- **Results also largely depend on the "space" the data is projected onto**

Li-C. Wang Tutorial - April 24th, 2015 45

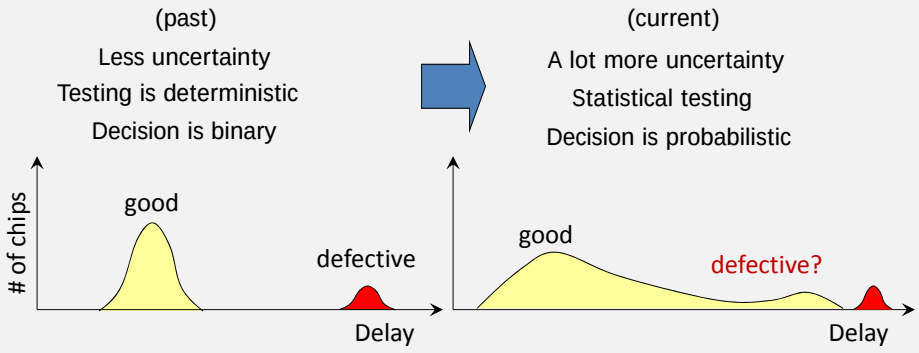
An Application Example



- **Learning model tries to replace the expensive flow with the cheaper one**

Li-C. Wang Tutorial - April 24th, 2015 46

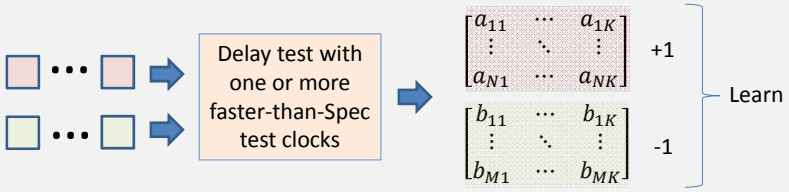
Specific Example – Parametric Delay Test



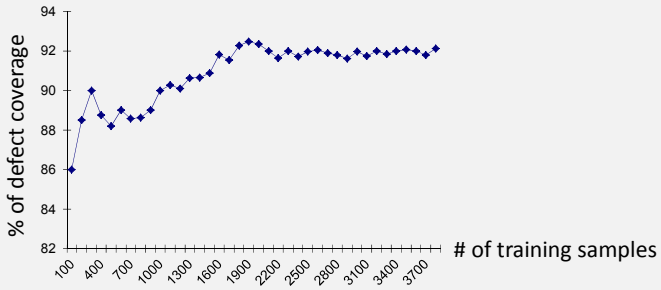
- **In the past, good and bad behavior can be clearly separated in a one-dimensional view**
  - Consider what being shown is the delay based on a pattern
- **Variations blur the boundary between good and bad**
  - Making it hard to separate in a one-dimensional view

Li-C. Wang Tutorial - April 24th, 2015 47

Turning Delay Test Into Parametric Measurement



- **Each measured value is an integer depending on the # of clocks applied**



Ben Lee et al. (ITC 2006)
"Issues on Test Optimization with Known Good Dies and Known Defective Dies —A Statistical Perspective"
http://mtv.ece.ucsb.edu/licwang/PDF/2006-ITC.pdf

Li-C. Wang Tutorial - April 24th, 2015 48

RF vs. SVM In Parametric Delay Test

	Support Vector Machine (SVM)	Random Forests (RF)
Application effectiveness	Comparable	Comparable
Test selection	Treat it as a separate problem	Naturally fit with the algorithm
Human interpretation of the model	Not intuitive	Intuitive
Implementation	More complex	Less complex
Tools used	NTU LIBSVM	UCSB librf

- **Findings:**
  - Both SVM and RF are effective in learning the pass/fail behavior
  - A RF model is more intuitive to interpret than a SVM model
- **See:**
  - SVM based: Ben Lee et al. (ITC 2006)
    - "Issues on Test Optimization with Known Good Dies and Known Defective Dies —A Statistical Perspective"
    - **http://mtv.ece.ucsb.edu/licwang/PDF/2006-ITC.pdf**
  - RF based: Sean Wu et al. (ITC 2007)
    - "Statistical Analysis and Optimization of Parametric Delay Test"
    - **http://mtv.ece.ucsb.edu/licwang/PDF/2007-ITCb.pdf**

Data Mining Approaches

- Classification
- **Regression**
- Clustering
- Transformation
- Outlier Detection
- Density Estimation
- Rule Learning

Data Mining 101 – Supervised Learning - Regression

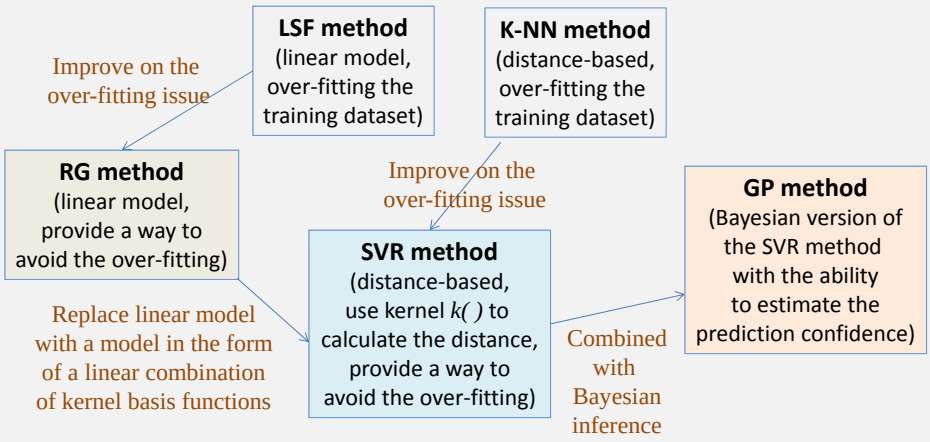
$$\text{(features)}\quad f_1 \quad f_2 \quad \dots \quad f_n$$

$$\mathbf{X} = \begin{vmatrix} \vec{x}_1 \\ \vec{x}_2 \\ \dots \\ \vec{x}_m \end{vmatrix} = \begin{vmatrix} x_{11} & x_{12} & \dots & x_{1n} \\ x_{21} & x_{22} & \dots & x_{2n} \\ \dots & \dots & \dots & \dots \\ x_{m1} & x_{m2} & \dots & x_{mn} \end{vmatrix} \quad \vec{y} = \begin{vmatrix} y_1 \\ y_2 \\ \dots \\ y_m \end{vmatrix}$$

Numerical output values

- **Regression**
  - There are outputs y's
  - Each y's is a numerical output value of some sort
- **For example, y is a frequency**

Example Learning Algorithms For Regression



- **See Janine Chen et al. (ITC 2009)**
  - **"Data Learning Techniques and Methodology for Fmax Prediction"**
  - **http://mtv.ece.ucsb.edu/licwang/PDF/2009-ITCb.pdf**

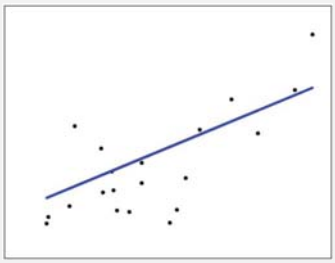
Least Square Fit

$$\mathbf{X} = \begin{vmatrix} \vec{x}_1 \\ \vec{x}_2 \\ \cdots \\ \vec{x}_m \end{vmatrix} = \begin{vmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{vmatrix} \quad \vec{y} = \begin{vmatrix} y_1 \\ y_2 \\ \cdots \\ y_m \end{vmatrix}$$

Assume model:

$f(x) = w_1x_1 + w_2x_2 + \cdots + w_nx_n + b$

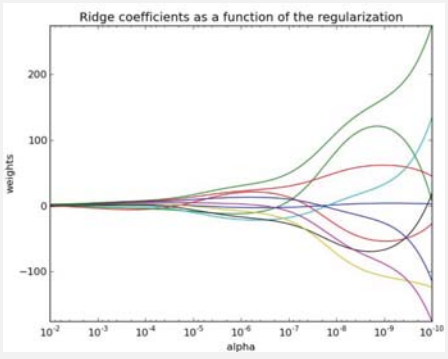
$\min \text{SE} = \sum_{i=1}^{m} (f(\vec{x_i}) - y_i)^2$



- **Assume a model**
  - Minimize the sum of squares to find values for the coefficients

Ridge Regression

$\min \text{SE} = \sum_{i=1}^{m} (f(\vec{x_i}) - y_i)^2 + \alpha \sum_{i=1}^{m} (w_i)^2$ ⟶ Regularization term

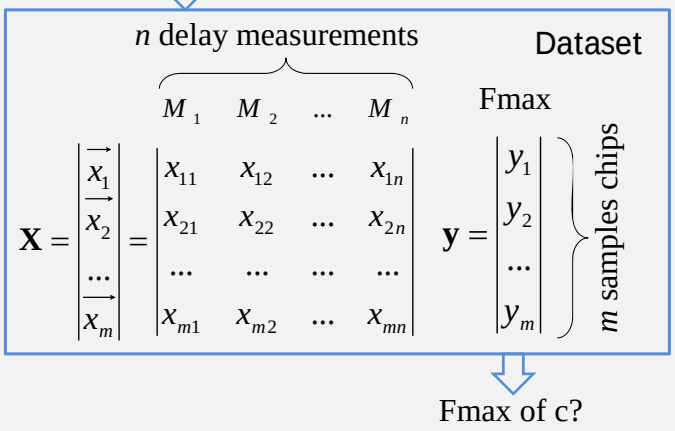


Source: http://scikit-learn.org/stable/modules/linear_model.html

- **Adding a regularization term makes the model more robust**
  - Avoid over-fitting the data

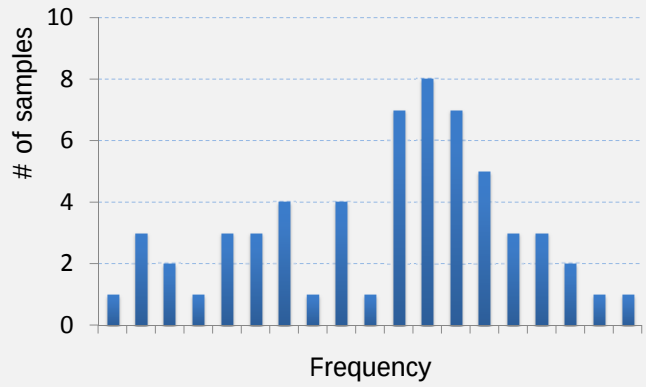
An Application Example – Fmax Prediction

$\vec{x} = x_1 \quad x_2 \quad \cdots \quad x_n$ (a new chip c)



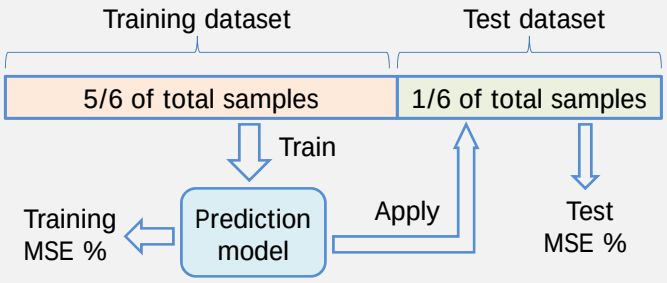
- **Delay measurements can be**
  - FF based, pattern based, path based, or RO based

Example Fmax Data



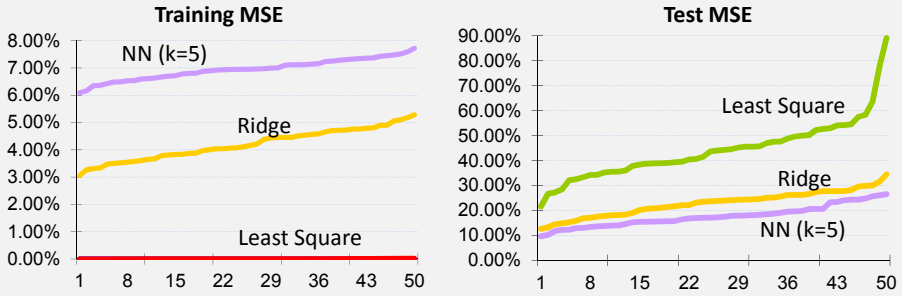
- **See Janine Chen et al. (ITC 2009)**
  - "**Data Learning Techniques and Methodology for Fmax Prediction**"
  - Consider FF based, pattern based and path based data

Experiment Setup



- **Training MSE% - Show how well the model fits the data**
- **Test MSE% - Show how well the model generalizes**
  - Test MSE% is what an application cares about

Some Result



- **Least Square**
  - With a small dataset and a high dimensional space, the model tends to over-fit the data
- **Ridge**
  - Regularization helps to alleviate the over-fitting situation
- **Nearest Neighbors**
  - Although simple, show best result
  - More complex algorithm is not always better!

Data Mining Approaches

- Classification
- Regression
- **Clustering**
- Transformation
- Outlier Detection
- Density Estimation
- Rule Learning

Data Mining 101 – Unsupervised Learning

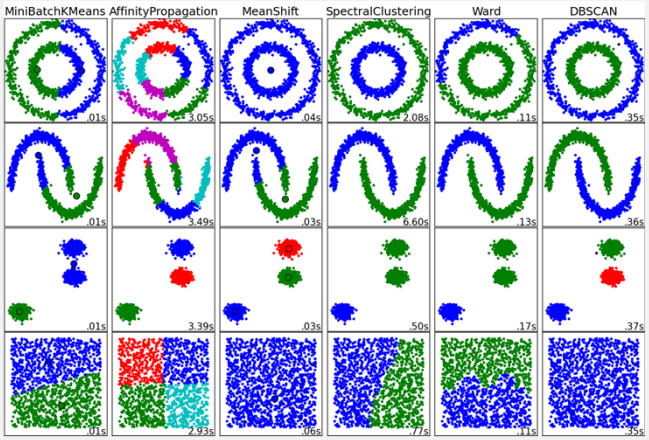
(features) $f_1 \quad f_2 \quad \dots \quad f_n$

$$\mathbf{X} = \begin{vmatrix} \vec{x}_1 \\ \vec{x}_2 \\ \dots \\ \vec{x}_m \end{vmatrix} = \begin{vmatrix} x_{11} & x_{12} & \dots & x_{1n} \\ x_{21} & x_{22} & \dots & x_{2n} \\ \dots & \dots & \dots & \dots \\ x_{m1} & x_{m2} & \dots & x_{mn} \end{vmatrix} \qquad \vec{y} = \begin{vmatrix} y_1 \\ y_2 \\ \dots \\ y_m \end{vmatrix}$$

No y's

- **Popular approaches**
  - Clustering
  - Transformation (dimension reduction)
  - Novelty Detection (Outlier analysis)
  - Density Estimation

Clustering Algorithms



Source: http://scikit-learn.org/stable/auto_examples/cluster/plot_cluster_comparison.html

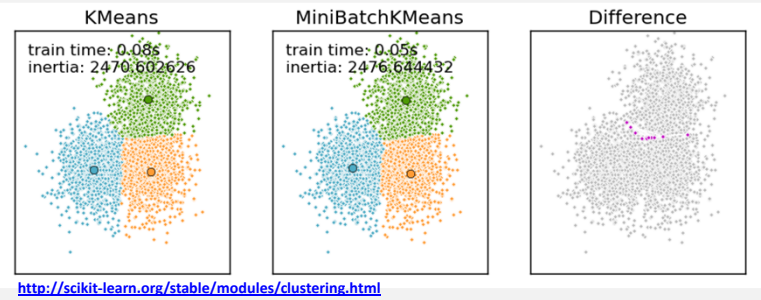
- **Clustering largely depends on**
  - The space the samples are projected onto
  - The definition of the concept "similarity"

Li-C. Wang Tutorial - April 24th, 2015 61

Clustering: K-Mean

- **K-Means**
  - User gives the number of clusters k
  - The algorithm follows simple 3 steps
    - 1. Randomly start with k samples as cluster centers
    - Loop until centroids coverage
      - A. **Assign the rest of points to its nearest center**
      - B. **For each cluster, create a new centroid by taking the mean of all points in the cluster**

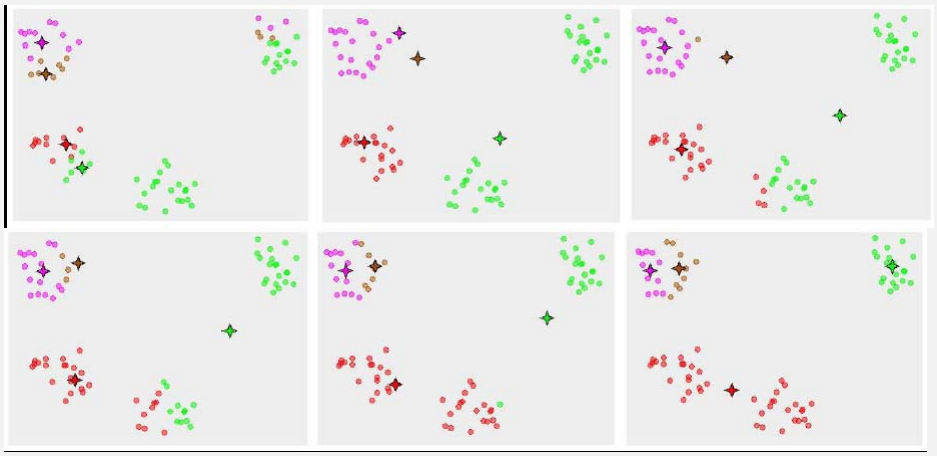
- **Mini Batch K-Means (for speed reason)**
  - In each iteration, randomly sample b points and assign them to centroids
  - Centroids are updated based on all points currently and previously assigned to them



http://scikit-learn.org/stable/modules/clustering.html

Li-C. Wang Tutorial - April 24th, 2015 62

K-Means Is Not Robust



http://en.wikipedia.org/wiki/K-means_clustering

- **The result depends on the initial points selected**
- **Final solution may converge to a local minimum**

Li-C. Wang Tutorial - April 24th, 2015 63

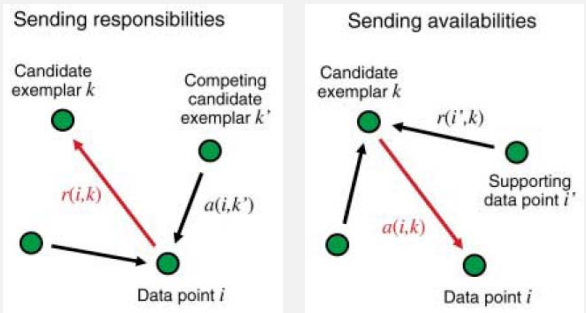
Clustering – Affinity Propagation

$r(i,k)$: how strong k should be the exemplar for i

$$r(i,k) \leftarrow s(i,k) - \max_{k' \text{ s.t. } k' \neq k} \{a(i,k') + s(i,k')\}$$

$a(i,k)$: how strong i should use k as the examplar

$$a(i,k) \leftarrow \min\Big\{0, r(k,k) + \sum_{i' \text{ s.t. } i' \notin \{i,k\}} \max\{0, r(i',k)\}\Big\}$$

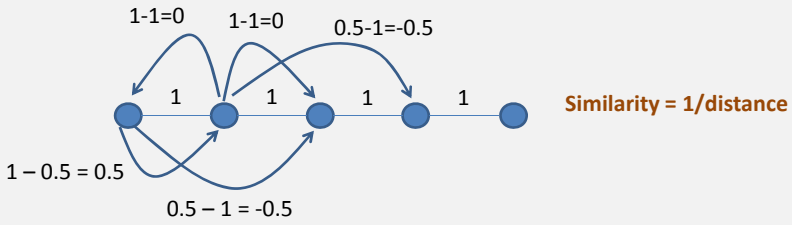


- **Initially, a(i, k) = 0**
  - s(i, k) = similarity measure between i and k
  - Iterate to find "exemplar"
  - When r(k,k) becomes negative, it is no longer a candidate for exemplar

Li-C. Wang Tutorial - April 24th, 2015 64

Clustering – Affinity Propagation

$r(i,k)$: how strong k should be the exemplar for i $r(i,k) \leftarrow s(i,k) - \max_{k' \text{ s.t. } k' \neq k} \{a(i,k') + s(i,k')\}$

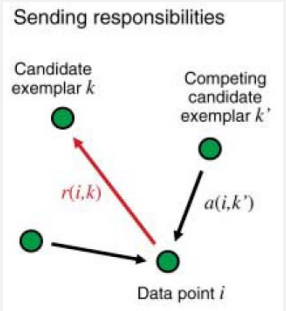


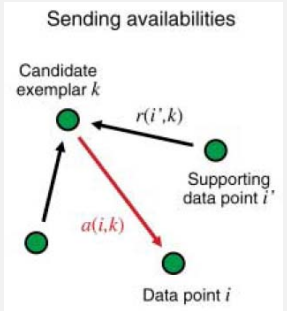
Li-C. Wang Tutorial - April 24th, 2015 65

Clustering – Affinity Propagation

$r(i,k)$: how strong k should be the exemplar for i $r(i,k) \leftarrow s(i,k) - \max_{k' \text{ s.t. } k' \neq k} \{a(i,k') + s(i,k')\}$

$a(i,k)$: how strong i should use k as the examplar $a(i,k) \leftarrow \min\left\{0, r(k,k) + \sum_{i' \text{ s.t. } i' \notin \{i,k\}} \max\{0, r(i',k)\}\right\}$



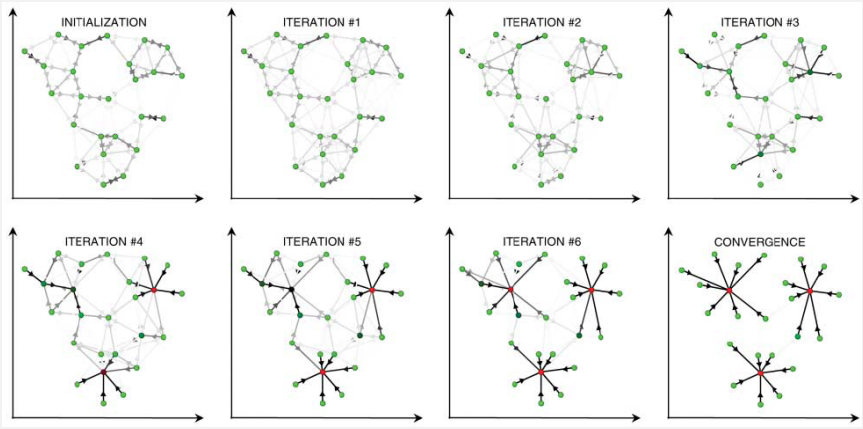


- **Initially, a(i, k) = 0**
  - s(i, k) = similarity measure between i and k – e.g. set to negative square error
- **Iterate to find "exemplar"**
  - When r(k,k) becomes negative, it is no longer a candidate for exemplar

Li-C. Wang Tutorial - April 24th, 2015 66

Clustering – Affinity Propagation

$r(i,k)$: how strong k should be the exemplar for i

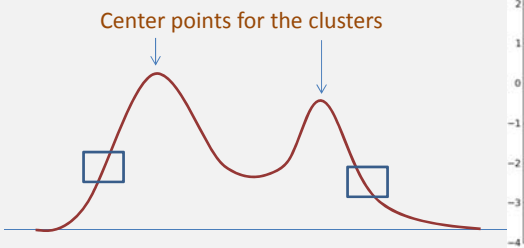


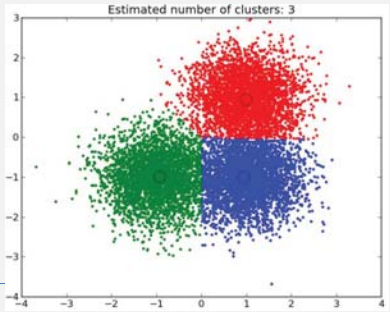
- **See** Brendan J. Frey and Delbert Dueck
  - "Clustering by Passing Messages Between Data Points"
  - SCIENCE www.sciencemag.org, Vol 315, Feb 16, 2007

Li-C. Wang Tutorial - April 24th, 2015 67

Clustering – Mean Shift

- **Mean shifts intends to find the "modes" in a distribution**
- **The algorithm follows simple 3 steps**
  - 1. Fix a "window" around each point
  - Loop until coverage
    - A. Compute the mean of the data within a window
    - B. Shift the window to the mean





http://scikit-learn.org/stable/auto_examples/cluster/plot_mean_shift.html

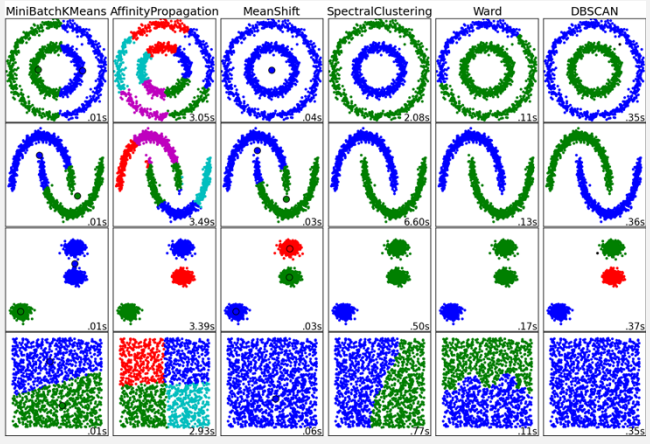
Li-C. Wang Tutorial - April 24th, 2015 68

Clustering – Other Algorithms

- **Spectral clustering**
  - Perform a low-dimensional data projection first
  - Operate the K-Means in the reduced dimensional space
- **Hierarchical clustering (Ward)**
  - Following a tree-like structure
    - Leaves are individual samples
  - Work bottom-up to the root of the tree
  - Merge similar samples into the same parent when moving up
  - Decide a level to output (# of nodes at the level = # of clusters)
- **DBSCAN**
  - User defines two parameters: ***min_samples*** and ***eps***
  - A core sample
    - There are at least ***min_samples*** points within ***eps*** distance
  - A cluster = defined by a set of core samples close to each other
  - The algorithm tries to identify "dense" region in the space

Li-C. Wang Tutorial - April 24th, 2015 69

Recall: Clustering Algorithms



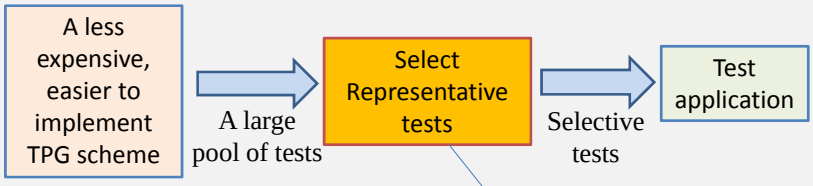
Source: http://scikit-learn.org/stable/auto_examples/cluster/plot_cluster_comparison.html

- **Clustering largely depends on**
  - Input parameter(s) chosen
  - The space the samples are projected onto
  - The definition of the concept "similarity"

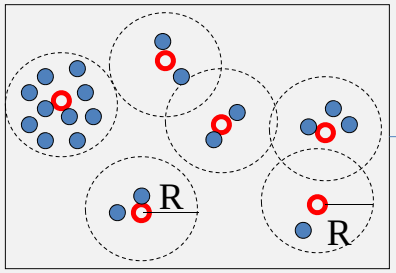
Li-C. Wang Tutorial - April 24th, 2015 70

An Example Application – Functional Test Selection

- **Simulation for functional verification is time and resources consuming**
- **However, many tests do not seem to capture anything**
- **What if we can select "representative tests" before simulation?**



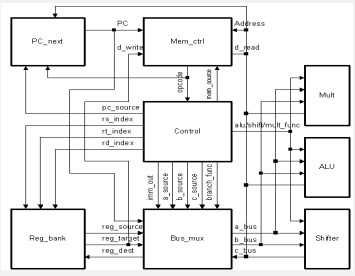
Clustering is a natural fit



The real challenge is
How to define a metric space that make sense?

A metric space is where the similarity (or distance) of two tests can be calculated

Some Result



Plasma (MIPS) core

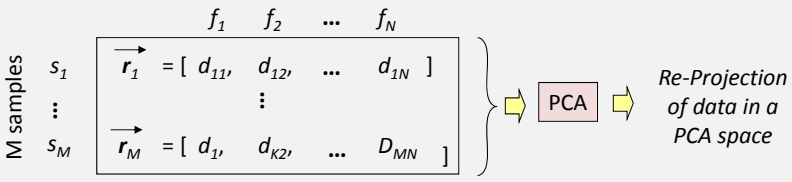
Coverage metric	Boot	1000 tests	K-Means (168)
Statements	57.5	**85.8**	85.2
Branches	60.6	**84.4**	84.1
Expressions	76.9	**92.3**	92.3
Conditions	63.0	**78.3**	78.2
Toggle	52.4	**76.6**	76.6

- **See Po-Hsien Chang et al. (ITC 2010)**
  - **"A Kernel-Based Approach for Functional Test Program Generation"**
  - http://mtv.ece.ucsb.edu/licwang/PDF/2010-ITCb.pdf
- **Findings**
  - The real challenge is not the learning algorithm, but to define a "kernel" function that measures the similarity between two assembly programs
  - Even though clustering seems to be a natural fit, a better way is to employ the "novelty detection" approach (discussed later)

Data Mining Approaches

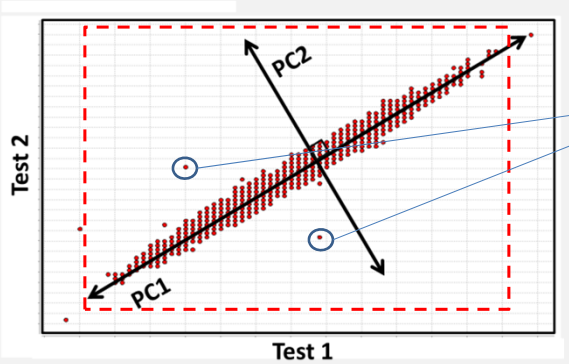
- Classification
- Regression
- Clustering
- **Transformation**
- Outlier Detection
- Density Estimation
- Rule Learning

Transformation – Principal Component Analysis



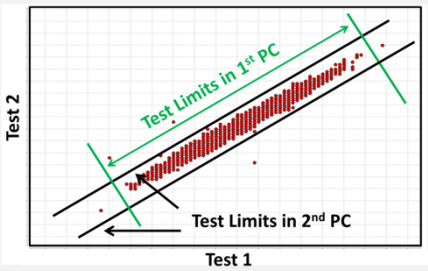
- **Principal Component Analysis (PCA) – find directions where the data spread out with large variance**
  - 1st PC – data spread out with the most variance
  - 2nd PC – data spread out with the 2nd most variance
  - …
- **PCA is good for**
  - Dimension reduction – feature selection
  - Visualization of high-dimensional data
  - Outlier analysis

PCA for Outlier Analysis in Test

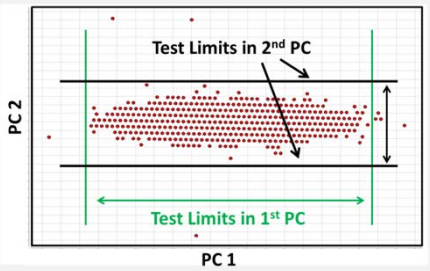


- **Each test is used to screen with a test limit**
  - Two tests essentially define a bounding box
- **Multivariate outliers are not screened by applying tests individually**

Multivariate Outlier Analysis



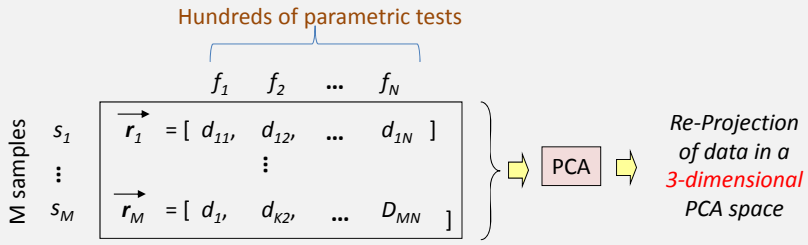
This is what we desire



PCA helps achieve that

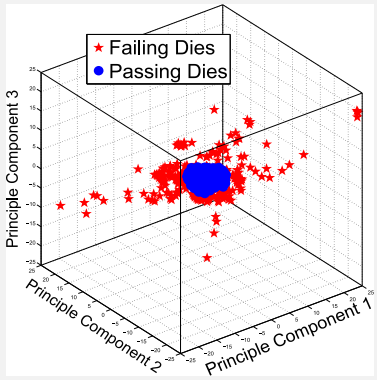
- **Use PCA to re-project the data into a PCA space**
  - then define the test limits in the PCA space
  - Each PC becomes just another test individually
- **Also see Nik Sumikawa et al. (ITC 2012)**
  - "Screening Customer Returns With Multivariate Test Analysis"
  - http://mtv.ece.ucsb.edu/licwang/PDF/2012-ITCa.pdf

PCA for Visualization



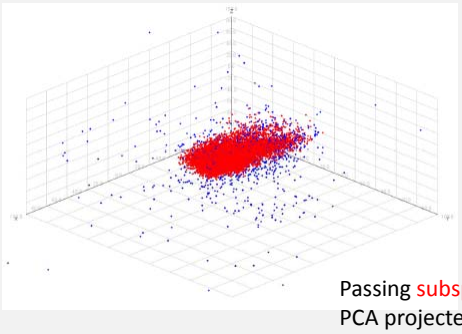
- **In wafer probe, hundreds of parametric tests are measured**
  - It is difficult to visualize data in such a high dimension
- **Idea: Use PCA to re-project the data into a 3-dimensional PC space**

PCA for Test Data Visualization



- **Example – 700+ wafer probe tests for a RF/A device**
- **PCA analysis shows there is a low-dimension "good" space that may be learned – suggesting the potential of many redundant tests**
- **See Gagi Drmanac et al. (ITC 2011)**
  - "Wafer Probe Test Cost Reduction of an RF/A Device by Automatic Testset Minimization: A Case Study"
  - **http://mtv.ece.ucsb.edu/licwang/PDF/2011-ITCb.pdf**

PCA for Test Data Visualization – Another Product

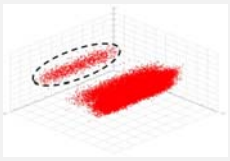


Passing subspace captured in the PCA projected space (red)

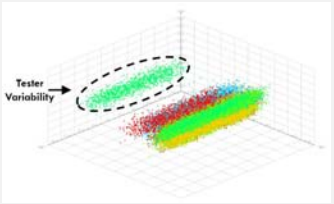
- **Again, PCA analysis shows there is a low-dimension "good" space that may be learned – suggesting the potential of redudant tests**
- **Note that this only shows that there is a low-dimension space to capture the majority of the passing dies (e.g. 90%)**
- **If your problem is to differentiate 50 failing parts out of 1M (0.005%)**
  - Results shown here tell you nothing about that possibility

Li-C. Wang Tutorial - April 24th, 2015 79

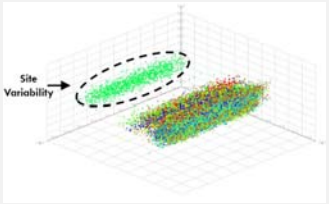
Visualization Reveals Variability



All passing dies



Colored by tester #



Colored by site #

- **PCA allows visualization of test data**
  - To reveal a clear tester variability
  - To reveal a clear site variability

Li-C. Wang Tutorial - April 24th, 2015 80

Data Mining Approaches

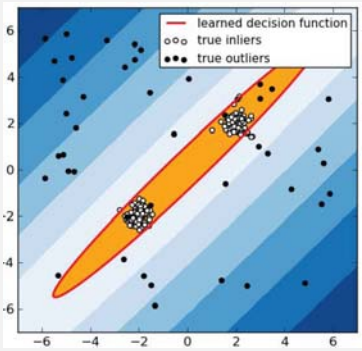
- Classification
- Regression
- Clustering
- Transformation
- **Outlier Detection**
- Density Estimation
- Rule Learning

Novelty Detection – Outlier Analysis

- **Principal Component Analysis**
- **Covariance based**
  - Mahalanobis distances
- **Density based**
  - Support Vector Machine one class
- **Tree based**
  - Random Forest
- **Not the same as clustering**
  - We only care about finding outliers

Covariance Based Outlier Detection

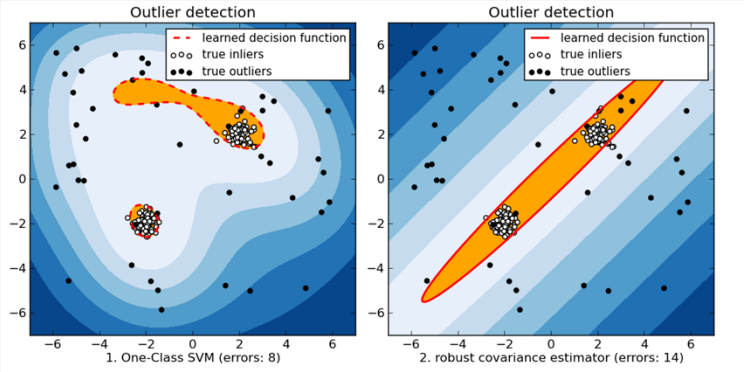
Mahalanobis distance $\mathrm{MD}(\vec{x}) = (\vec{x} - \vec{\mu})\ \sum^{-1}(\vec{x} - \vec{\mu})$



- **Assume data follows a multivariate Gaussian distribution**
- **Essentially, find one oval shaped model to fit most of the data**

Li-C. Wang Tutorial - April 24th, 2015 83

Covariance Based vs. Density Based



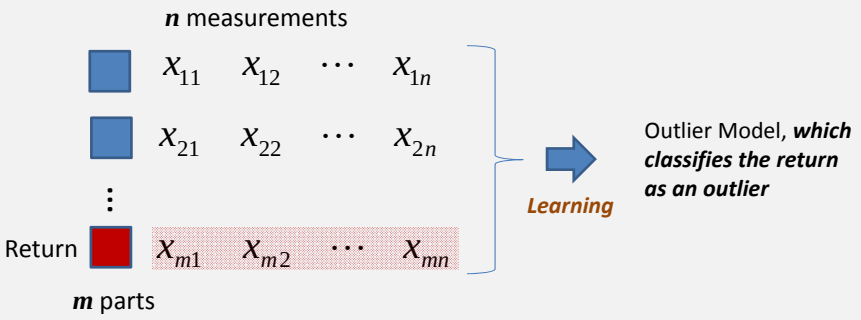
Source: http://scikit-learn.org/stable/auto_examples/covariance/plot_outlier_detection.html

- **If the data does not follow the Gaussian distribution assumption, then a density based approach would be better**
  - 1-class SVM is a density based approach (discussed later)
- **Otherwise, variance based approach would probably be sufficient**

Li-C. Wang Tutorial - April 24th, 2015 84

Customer Return Analysis

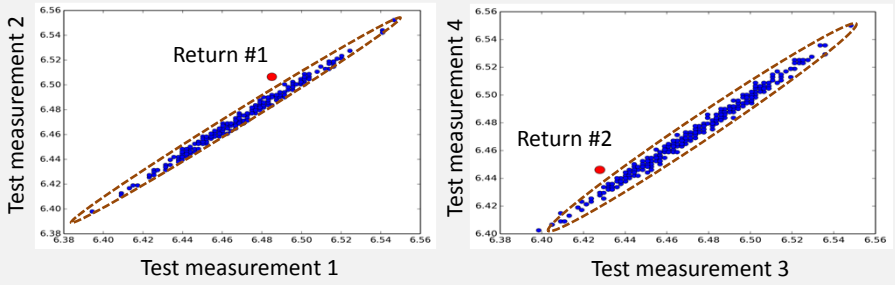
- **Few parts known as customer returns**
  - Majority of other parts (even though they pass all tests) in this analysis, are treated without a label (they could be still bad)



Li-C. Wang Tutorial - April 24th, 2015 85

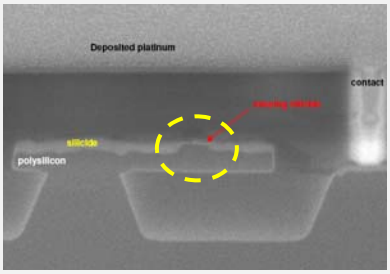
Customer Return Analysis - Example

- **Returns from a power management chip product**
- **Outlier models were searched over 900+ test measurements**
- **Good 2-dimensional outlier models were found**
  - Models can be used to prevent future similar returns
  - Or can be used to facilitate debug



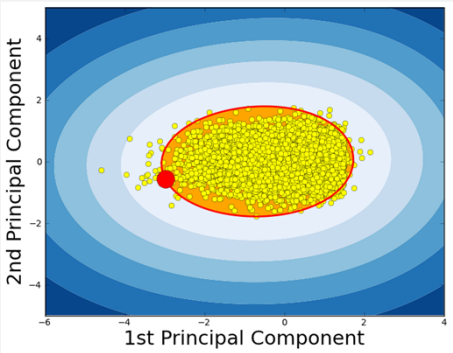
Li-C. Wang Tutorial - April 24th, 2015 86

Another Example



- **A customer return passes all tests**
  - But fail at customer site
  - It is mostly due to latent defect
- **In this particular example**
  - SOC controller for automotive
  - Start to fail after driving 15000 miles
  - Show failure only under -40°C
    - Failure is also frequency dependent
  - Determined to be a latent defect

Outlier Model For Customer Return



- **In this case, we start with 3 tests**
  - Apply PCA first – use the first two PCs
  - Apply variance based outlier model
- **The return is the 33rd outlier in the entire lot**
- **See** Jeff Tikkanen et al. (IRPS 2013)
  - "Statistical Outlier Screening For Latent Defects"
  - http://mtv.ece.ucsb.edu/licwang/PDF/2013-IRPS.pdf

Data Mining Approaches

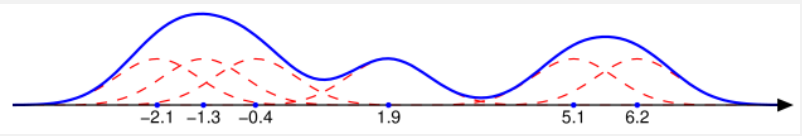
- Classification
- Regression
- Clustering
- Transformation
- Outlier Detection
- **Density Estimation**
- Rule Learning

Density Estimation

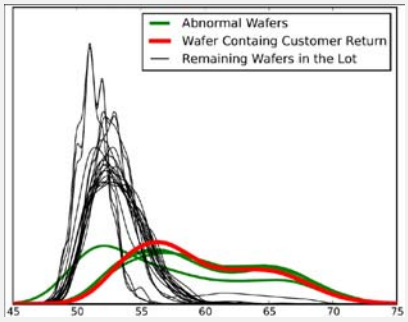
- **For density estimation, several non-parametric methods were proposed in 1960s**
  - Non-parametric because no fixed functional is given
- **One famous example is the Parzen's window**
  - Requires the definition of a *kernel* function that is a symmetric unimodal density function

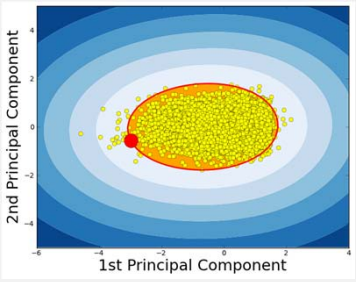
$$k(x, x_i, \gamma) = \frac{1}{\gamma^n} k(\frac{x - x_i}{\gamma}), x \in R^n \qquad P(x) = \frac{1}{k} \sum_{i=1}^{k} k(x, x_i, \gamma)$$

$$k(x, x_i, 1) = \frac{1}{\sigma\sqrt{2\pi}} \exp\{-\frac{(x - x_i)^2}{2\sigma^2}\}$$ Gaussian kernel



Density Estimation for Visualization





- **In the previous example, the customer return is located on a wafer whose distribution of the test is different from majority of the wafers**
- **One can use Kolmogorov-Smirnov test to identify similar wafers**
  - Hence, the outlier model is applied only to the abnormal wafers
  - This dramatically reduce the overkill rate
- **See** Jeff Tikkanen et al. (IRPS 2013)
  - "Statistical Outlier Screening For Latent Defects"
  - **http://mtv.ece.ucsb.edu/licwang/PDF/2013-IRPS.pdf**

Data Mining Approaches

- Classification
- Regression
- Clustering
- Transformation
- Outlier Detection
- Density Estimation
- **Rule Learning**

Data Mining 101 – Rule Learning

$$\text{(features)} \quad f_1 \quad f_2 \quad \dots \quad f_n$$

$$\mathbf{X} = \begin{vmatrix} \vec{x}_1 \\ \vec{x}_2 \\ \dots \\ \vec{x}_m \end{vmatrix} = \begin{vmatrix} x_{11} & x_{12} & \dots & x_{1n} \\ x_{21} & x_{22} & \dots & x_{2n} \\ \dots & \dots & \dots & \dots \\ x_{m1} & x_{m2} & \dots & x_{mn} \end{vmatrix} \qquad \vec{y} = \begin{vmatrix} y_1 \\ y_2 \\ \dots \\ y_m \end{vmatrix}$$

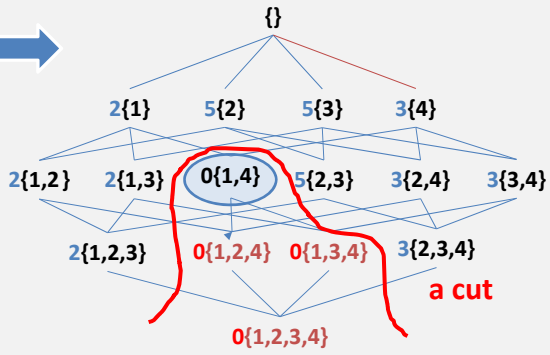
Binary label

- With y's label (binary class)
  - Classification rule learning
- Without y's label (unsupervised)
  - Association rule mining

Classification Rule Learning – An Application Example

p1 is a special path

Paths	Features				
	c1	*c2*	*c3*	*c4*	*c5*
p1	1	1	1	1	**0**
p2	1	1	1	0	**1**
p3	0	1	1	1	**0**
p4	0	1	1	1	**1**
p5	1	1	1	0	**1**
p6	0	1	1	1	**0**



Output hypothesis: {1,4}
Confidence: 75%

- See Nick Callegari et al. (DAC 2009)
  - "Speed path analysis based on hypothesis pruning and ranking"
  - http://mtv.ece.ucsb.edu/licwang/PDF/2009-DAC.pdf

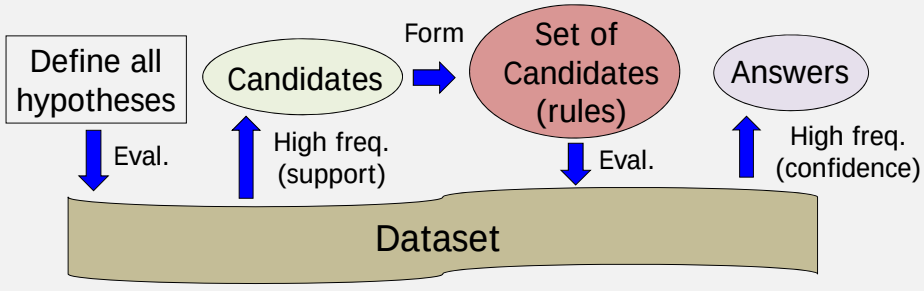
Note and Issue

- **This is a problem studied in the area of association rule mining**
  - Apriori – a breadth first search strategy to find the cut
  - Mafia – a depth first search strategy to find the cut
- **The hypothesis lattice does not work for features with numerical values**
  - We need to convert numerical features into the binary-valued features
  - The most intuitive way is to bin a range
- **Subgroup discovery problem (multiple causes)**
  - If the target set has n items, how do we know what is the right partitioning (each partitioning is due to one cause)?

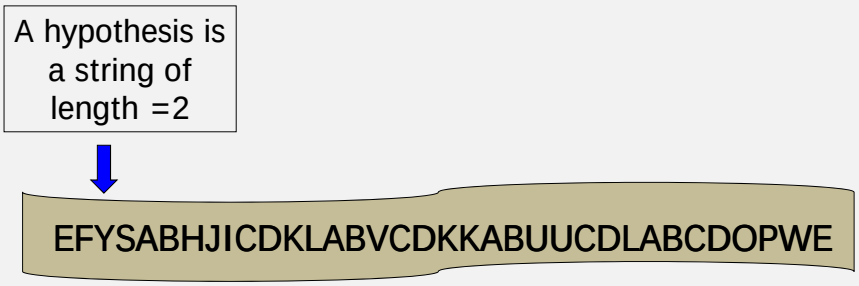
If we group like this, we will not find a good hypothesis to explain all three

Associate Rule Mining – An Application Example

- **Rule mining follows a Support-Confidence Framework**
- **The basic principle is simple and intuitive**
  - From data, form a hypothesis space of candidates
  - If a candidate appears "frequently" in a dataset, the candidate must have some meaning
- **The evaluation of this frequency is a 2-step process – Support and then Confidence**



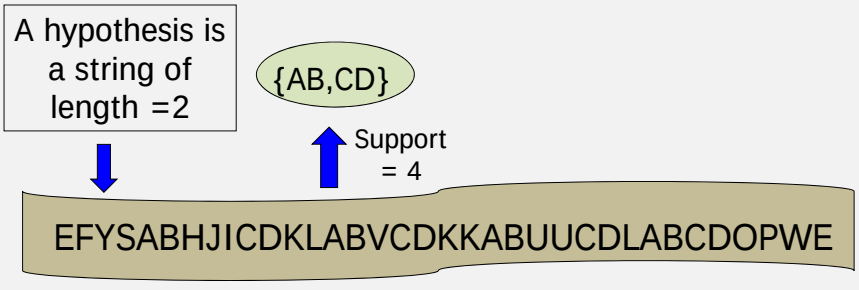
Example – Sequential Episode Mining



Li-C. Wang Tutorial - April 24th, 2015 97

Example – Sequential Episode Mining

EFYSABHJICDKLABVCDKKABUUCDLABCDOPWE



Li-C. Wang Tutorial - April 24th, 2015 98

Example – Sequential Episode Mining

EFYSABHJICDKLABVCDKKABUUCDLABCDOPWE

AB ⇘ CD AB ⇘ CD AB ⇘ CD AB ⇘ CD

A hypothesis is a string of length =2

{AB,CD} → Form → AB ⇒ CD → AB ⇒ CD

Support = 4

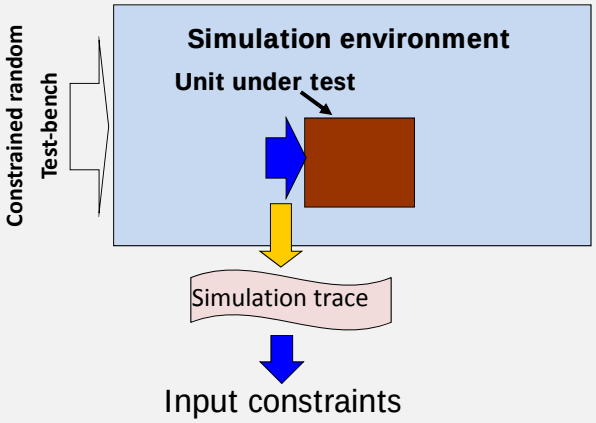
Eval.

Confidence = 4/4 = 100%

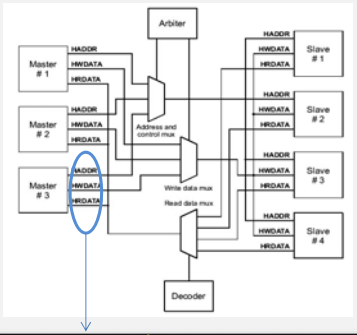
EFYSABHJICDKLABVCDKKABUUCDLABCDOPWE

An Application – Simulation Trace Abstraction

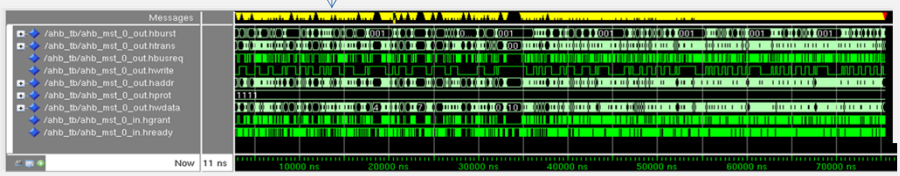
- **Analyze a simulation trace symbolically for better understanding**
- **Possibly extract sequential patterns in the trace**



A Simple Example

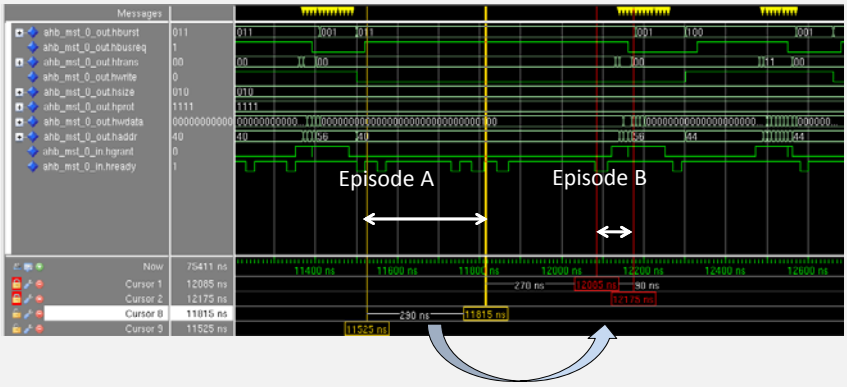


AMBA 2.0 (AHB)



- **See Po-Hsien Chang et al. (ASP-DAC 2010)**
  - "Automatic assertion extraction via sequential data mining of simulation traces"
  - http://mtv.ece.ucsb.edu/licwang/PDF/2010-ASPDACa.pdf

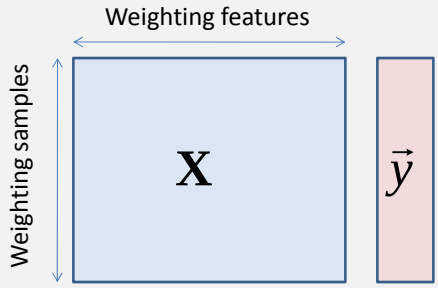
An Example Rule



A (request/wait) ⇒ B (transfer)

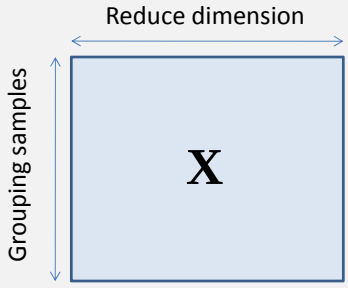
- **The separation in between A and B can be with an arbitrary number of cycles**

Summary – Supervised Learning



- **Supervised learning learns in 2 directions:**
  - Weighting the features
    - Tree learning, feature selection algorithms, Gaussian Process
  - Weighting the samples
    - SVM, Gaussian Process (discussed later)
- **Supervised learning includes**
  - Classification – y are class labels
  - Regression – y are numerical values
  - Classification rule learning

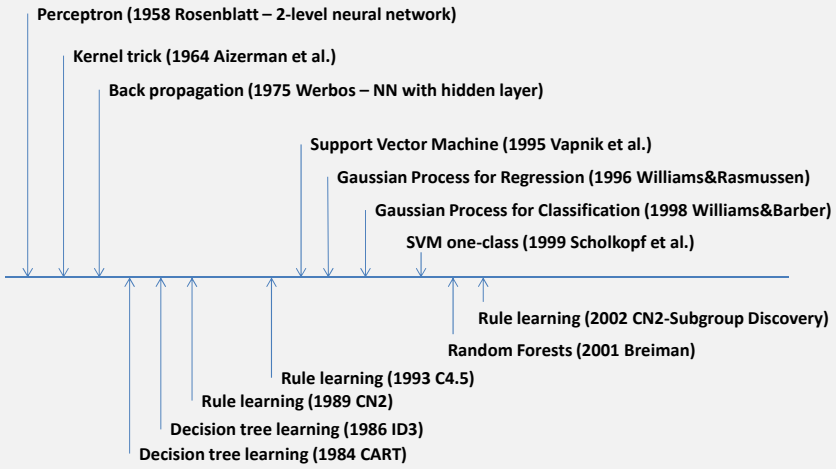
Unsupervised Learning



- **Unsupervised learning also learns in 2 directions:**
  - Reduce feature dimension
    - Principal Component Analysis (PCA), Association Rule Mining
  - Grouping samples or finding outliers
    - Clustering algorithms, Outlier detection algorithms
- **Unsupervised learning includes**
  - Clustering
  - Transformation (PCA, multi-dimensional scaling)
  - Novelty detection (outlier analysis)
  - Density estimation
  - Association rule mining (explore feature relationship)

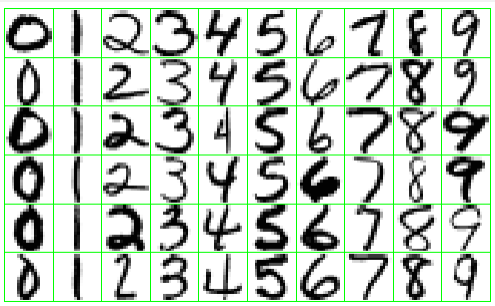
Learning Theory, SVM and Kernel Method (50 minutes)

Classification, Machine Learning, Pattern Recognition



- In machine learning, Perceptron is widely considered as one of the earliest examples to show that a machine can actually "learn"
- SVM is based on statistical learning theory that provides the necessary and sufficient conditions where a machine is guaranteed to "learn"

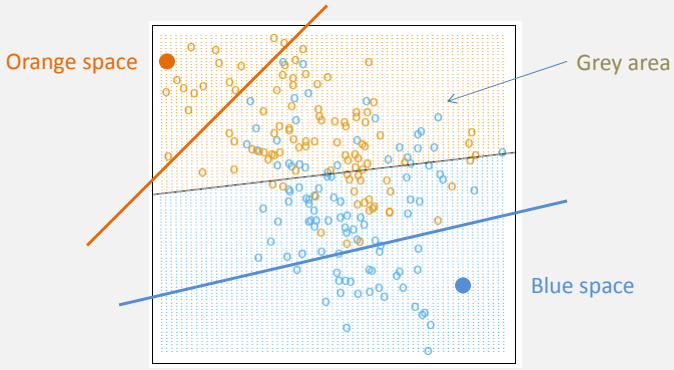
A Popular Dataset For Machine Learning Research



Source: Hastie, et al. "The Elements of Statistical Learning" 2nd edition 2008 (very good introduction book)

- **One of the most popular datasets used in ML research was the USPS dataset for hand-written postal code recognition**
  - e.g. When SVM was introduced, it substantially outperformed others based on this dataset
- **Question: What is the difference between this problem and yours?**

Binary Classification

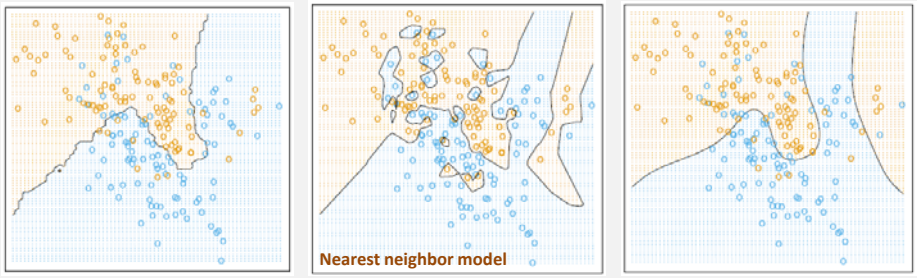


Source: Hastie, et al. "The Elements of Statistical Learning" 2nd edit 2008 (very good introduction book)

- **There are subspaces that are easy to classify (all algorithms agree)**
- **One algorithm differs from another on how each partitions the subspace in the "grey area"**
  - What's the "best way" to define the "orange-blue" boundary?

Model Complexity

Source: Hastie, et al. "The Elements of Statistical Learning" 2nd edition 2008 (very good introduction book)

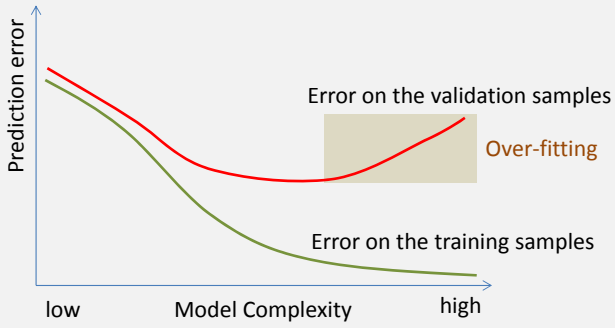


Complex – rough edge | Complex – fragmented | Smooth

- **You can always find a model that perfectly classifies the two classes of training samples (middle picture – based on nearest neighbor strategy)**
  - The model is usually complex
- **However, this may not be what you want**
  - Because your model is highly biased by the training data

Li-C. Wang Tutorial - April 24th, 2015 109

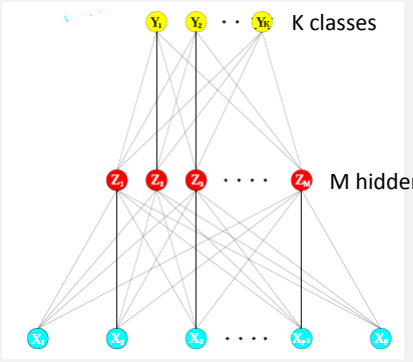
Model Complexity Vs. Prediction Error



- **In learning, an algorithm tries to explore this tradeoff to avoid over-fitting**
- **There are two fundamental approaches**
  - Fixing a model complexity
    - Find the best fit model to the train data
    - e.g. Neural Network, equation based models
  - Fixing a training error (say almost 0)
    - Find the low-complexity model (given ALL possible functional choices in a space)
    - e.g. SVM

Li-C. Wang Tutorial - April 24th, 2015 110

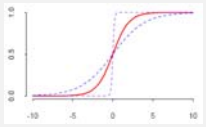
Neural Network (Fixed Complexity)



$$Y_1 = b_{11}Z_1 + b_{21}Z_2 + \ldots + b_{M1}Z_M + b_{01}$$

$$Z_1 = \partial(a_{11}X_1 + a_{21}X_2 + \ldots + a_{P1}X_P + a_{01})$$

$$\partial(x) = \frac{1}{1+\exp(-x)}$$



Source: Hastie, et al. "The Elements of Statistical Learning" 2nd edition

- **A neural network model complexity is fixed by fixing the number of Z variables**
- **Learning is by finding the best-fit values for the parameters**
  - (M+1)K parameters
  - (P+1)M parameters
- **e.g. Use the back propagation algorithm (1975 Werbos)**

Support Vector Machine

- **Fix the training error (say 0), minimize the model complexity**
  - Find the "simplest model" to fit the data
  - Occam's razor (William of Ockham 1287-1347)
    - The simplest is the best
    - The razor states that one should proceed to simpler theories until simplicity can be traded for greater explanatory power.
- **What is the complexity of a learning model?**
  - What is the model like?

What Is The Model Like?

- **Suppose we have a similarity function that measures the similarity between two sample vectors**

  $k(\vec{x}, \vec{x}_i)$ measures the similarity between two vectors

- **A simple classifier is to compare the average similarity to class +1 samples to the average similarity to class -1 samples**

$$y = f(x) = sign\left(\frac{1}{m_1}\sum_{\{i:y_i=+1\}} k(\vec{x}, \vec{x_i}) - \frac{1}{m_2}\sum_{\{j:y_j=-1\}} \frac{1}{m_2} k(\vec{x}, \vec{x_j}) + b\right)$$

$$f(\vec{x}) = b + \Sigma \alpha_i k(\vec{x}, \vec{x}_i)$$

Li-C. Wang Tutorial - April 24th, 2015 113

Model Complexity

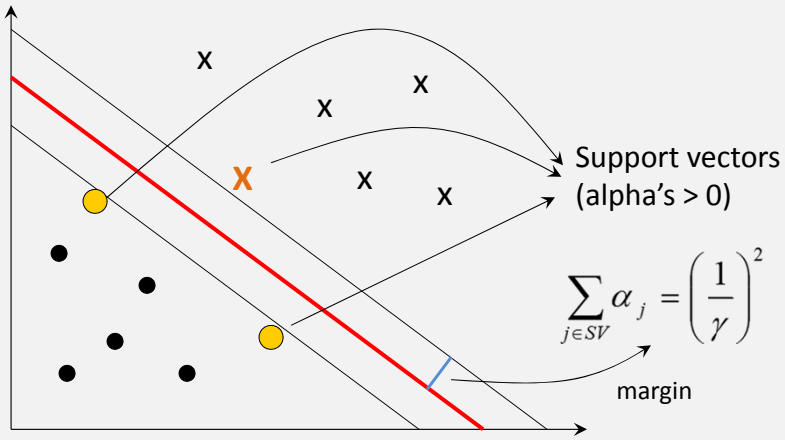
$$\mathbf{X} = \begin{vmatrix} \vec{x}_1 \\ \vec{x}_2 \\ \dots \\ \vec{x}_m \end{vmatrix} = \begin{vmatrix} x_{11} & x_{12} & \dots & x_{1n} \\ x_{21} & x_{22} & \dots & x_{2n} \\ \dots & \dots & \dots & \dots \\ x_{m1} & x_{m2} & \dots & x_{mn} \end{vmatrix} \quad \vec{\alpha} = \begin{vmatrix} \alpha_1 \\ \alpha_2 \\ \dots \\ \alpha_m \end{vmatrix}$$

Model complexity $\propto (\alpha_1 + \ldots + \alpha_m)$

- **In SVM theory, model complexity is measured by the sum of alpha's**

Li-C. Wang Tutorial - April 24th, 2015 114

Lowest Complexity = Maximum Margin



- **The square root of the sum of Lagrangian multipliers (alpha's), is the same as the inverse of the margin γ achieved by the hyperplane**
- **Minimizing the model complexity -> finding the maximum margin**

Summary of SVM

$$\mathbf{X} = \begin{vmatrix} \vec{x}_1 \\ \vec{x}_2 \\ \dots \\ \vec{x}_m \end{vmatrix} = \begin{vmatrix} x_{11} & x_{12} & \dots & x_{1n} \\ x_{21} & x_{22} & \dots & x_{2n} \\ \dots & \dots & \dots & \dots \\ x_{m1} & x_{m2} & \dots & x_{mn} \end{vmatrix} \quad \vec{\alpha} = \begin{vmatrix} \alpha_1 \\ \alpha_2 \\ \dots \\ \alpha_m \end{vmatrix}$$

Find the least complex model that guarantees a given training error

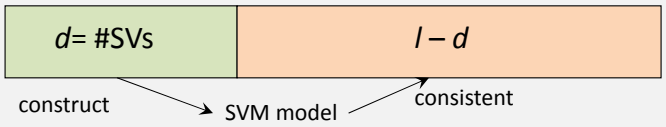
Model: $M(\vec{x}) = b + \Sigma \alpha_i k(\vec{x}, \vec{x}_i)$

Model complexity $\propto (\alpha_1 + \ldots + \alpha_m)$

where $k(\vec{x}, \vec{x}_i)$ measures the similarity between two vectors

- **k( ) is called a kernel function – measuring the similarity between 2 samples**
- **The strategy is to minimize the model complexity, under the constraint of a given training error (quadratic optimization problem)**
- **In the model, some alphas are zero, or close to zero**
  - Those samples are NOT support vectors
- **Others become support vectors**

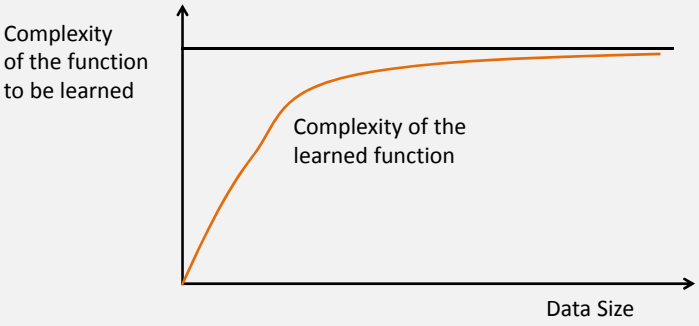
Generalization – Why Simplest Is The Best



- **SVM theorem says that the fewer the support vectors, the better generalization can be expected (less complex model)**
  - The number of support vectors is also a measure of our confidence on the learned model
- **An easy way is therefore to look at the quantity (#SVs) / *l* which tells us the confidence on the model derived with SVM**
- **See Schokopf and Smola "Learning with Kernels" 2002 MIT Press**
  - We found it the best book for learning SVM and related topics

Li-C. Wang Tutorial - April 24th, 2015 117

Robustness and Efficiency



- **Modern learning algorithms such as SVM improve the consistency, robustness and efficiency for converging to the "truth"**
  - Consistency: As data size approaches infinity, it guarantees to learn the truth
  - Robustness: The more data, the better learning model is
  - Efficiency: The best algorithm has the highest rate of convergence
- **In contrast, a traditional fixed-model-complexity approach does not guarantee this consistency and robustness**

Li-C. Wang Tutorial - April 24th, 2015 118

Fundamental Statistical Learning Theory

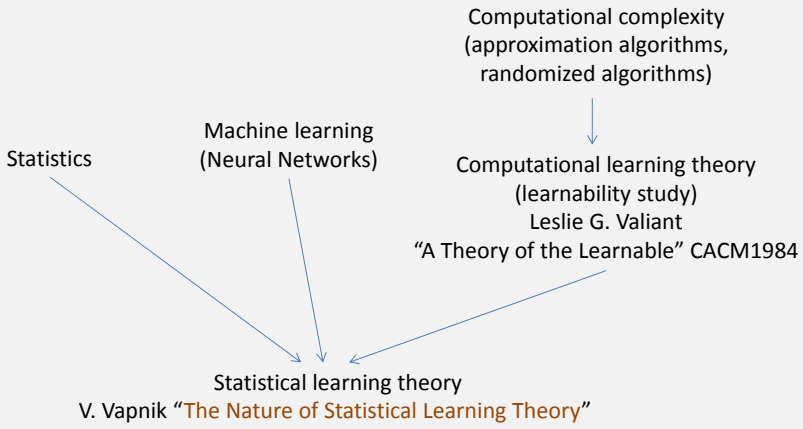
- **The necessary and sufficient condition for the learning to be consistent**

$$\lim_{m\to\infty} \frac{\text{VC Entropy}(m)}{m} = 0$$

See V. Vapnik "The Nature of Statistical Learning Theory" 2nd edition, 2000

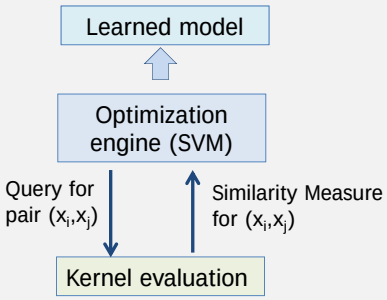
- **VCEntropy(m) is a measure of complexity based on *m* samples**
  - Measured as the size of some dimensionality
- **The complexity seen on *m* samples asymptotically grows slower than *m***
- **If this is not true, then the data cannot be learned**
  - Because additional samples keep introduce new behavior
  - There is no end of the behavioral space to be modeled
- **VCEntropy(m) cannot be measured directly – it is a theoretical concept**
  - In practice, you can use the SVM alpha's to estimate its grow

Statistical Learning Theory – A Unified View To Learning



- **Statistical learning theory is a theory about what is learnable in the continuous domain**
  - Is a theory to measure the complexity of continuous function

Kernel-Based Learning

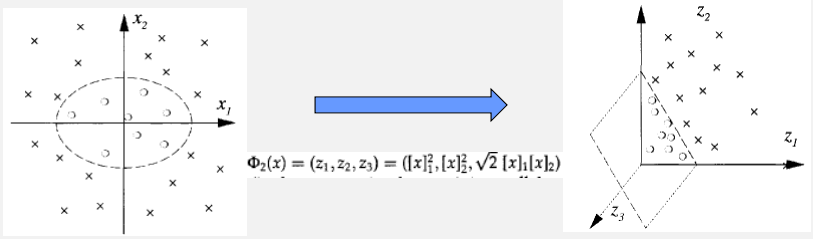


- **SVM engine and kernel are separated entities**
- **SVM always builds a "linear" model in the space defined by the kernel**
  - to build a non-linear model, we just use a non-learning kernel
- **A well-defined kernel k(x,y) = <ϕ(x), ϕ(y)> for some mapping ϕ( )**

$$\text{dot product} : < \mathbf{x}, \mathbf{y} >= x_1 y_1 + \ldots + x_n y_n$$

Li-C. Wang Tutorial - April 24th, 2015 121

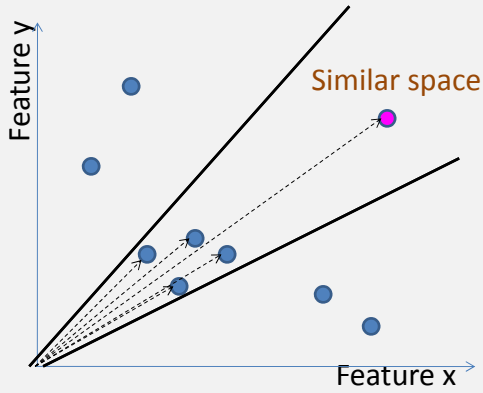
Kernel Function – Turn Non-Linear Into Linear

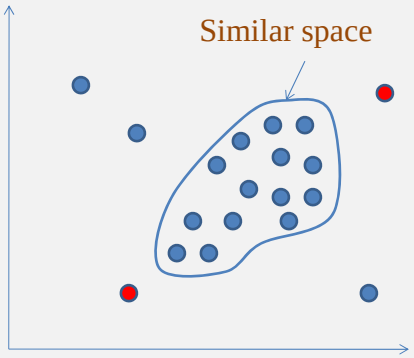


- **The points are not linearly separable in the input space**
- **After mapping, they are linearly separable in the mapped feature space**
- **With a complex enough feature mapping, the two classes of data points are always linearly separable**

Li-C. Wang Tutorial - April 24th, 2015 122

Angle Based Kernel Vs. Distance Based Kernel



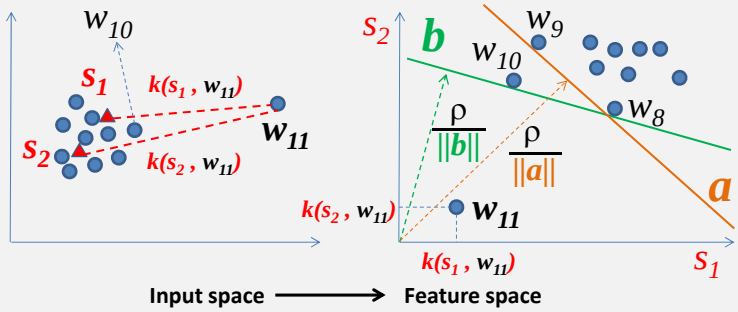


$$k(x,z) = \left(\frac{< x, z >}{\|x\| \|z\|} \right) = \cos(x,z)$$

$$k(x,z) = \exp(-g\|x-z\|^2)$$

- **Different kernels let the learning algorithm see the space differently**

SVM One-Class – Separate From The Origin In The Kernel Space



Mapping is based on $(k(s2,w), k(s1,w))$ for a kernel function $k()$

- **Finding a model to capture most of the sample points in the input space =**
  - Finding a maximum margin hyperplace (under a given threshold on the % of outliers) to separate most of the sample points from the origin **0**
- **This idea leads to the popular ν-SVM one-class algorithm**
- **See Schokopf and Smola "Learning with Kernels" 2002 MIT Press**

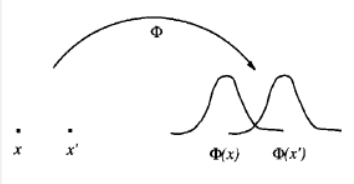
Use A Gaussian Kernel To Work In An Infinite Dimensional Space

$$Gaussian\ k(x,x') = \exp(-g\|x-x'\|^2)$$

For all possible points $\phi(x) = (k(x,x_1), k(x,x_2), k(x,x_3), \ldots, k(x,x_\infty))$

… together forms a **Gaussian distribution**

Or they represents a point in an infinite dimensional space



- **Then, we apply the strategy to find a maximum-margin hyperplace to separate most of the sample points from the origin in the Gaussian space**
- **User specifies the parameter ν**
  - Upper bound on the # of outliers; lower bound on the # of SVs
- **The model is**

$$f(x) = sign\{(\Sigma_i \alpha_i k(x,x_i)) - \rho\}$$

Bayesian SVM

- **In SVM, a given kernel is a prior**
  - That gives our belief on how the data points are distributed in the kernel-induced learning space
  - This prior may not be optimal
- **If we have a perfect kernel, separation of two classes will become extremely easy**
- **Bayesian inference can be combined to estimate a best kernel**
  - Learning includes finding the best kernel for the prediction
- **The overall framework is called Gaussian Process (or GP, see the book, Gaussian Process for Machine Learning, http://www.gaussianprocess.org/)**
  - Very successful in regression
  - Not yet applicable in unsupervised learning

Recall: Bayesian Inference – Naïve Bayes Classifier

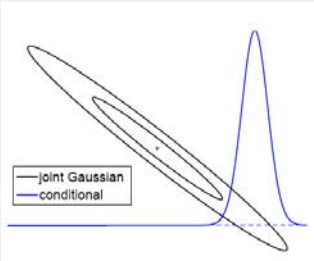
$$p(class \mid x_1,...,x_n) = \frac{p(class)\,p(x_1,...,x_n \mid class)}{p(x_1,...x_n)} = \frac{prior \times likelihood}{evidence}$$

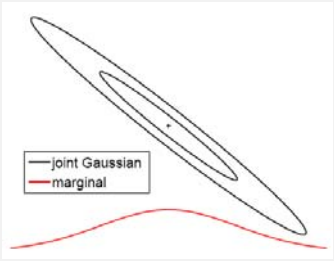
$$p(class \mid x_1,...,x_n) \propto p(class)\,p(x_1,...,x_n \mid class) \propto p(class)\,p(x_1 \mid class)\cdots p(x_n \mid class)$$

Independent assumptions

- **The naïve Bayes classifier uses the assumption that features are mutually independent**
  - This is not usually not true as we have seen in the test data
- **Also, if each xi is a continuous variable, we either need to estimate the probability density, or we need to discretize the value into ranges**

What Gaussian? – Easy To Deal With Joint Distribution





Source: Carl E. Rasmussen, Advances in GP, NIPS 2006

- **Joint of Gaussian distributions is Gaussian**
- **Marginal Gaussian distribution is Gaussian**
- **These two properties make it a nice choice if we do not want to take the independent assumption any more**
  - Then, we need to consider joint distributions

What is a GP

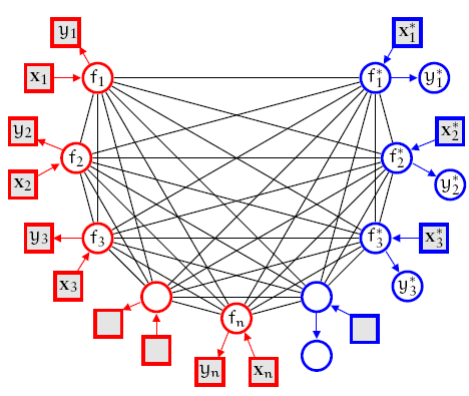
- **A GP is a collection of random variables, any finite subset of which have a joint Gaussian distribution**
- **A GP is fully specified by two things**
  - A mean function $m(x)$
  - A co-variance function $k(x,x')$
    - This co-variance function corresponds to the kernel before
- **We wrie**

$$f(x) \sim GP(m(x), k(x, x'))$$

Li-C. Wang Tutorial - April 24th, 2015 129

Gaussian Field Model

- **Each y_i is the output of a function $f_i(\,)$ on the input x_i**
- **Each $f_i(\,)$ is a random variable with Gaussian distribution G(0,σ^2)**
  - σ^2 is the **noise**
- **Any pair of functions f_1,f_2 have a joint Gaussian distribution G(0, $k(f_1,f_2)$) where $k(\,)$ is the co-variance function (kernel)**
- **n functions have a joint Gaussian distribution G(0, K) where K is the kernel matrix**
- **Prediction based on a new point x_* is based on predicting the function f_* that is correlated to all other functions following the joint Gaussian distribution on the n+1 functions**
- **The Gaussian process is fixed by the co-variance function and zero mean**
  - Adding more functions does not change the distribution of the data



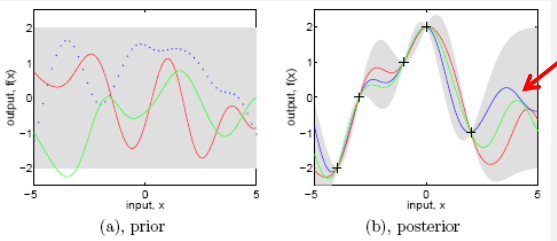
Source: Carl E. Rasmussen, Advances in GP, NIPS 2006

Li-C. Wang Tutorial - April 24th, 2015 130

Prediction Distribution

GP prior: $f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}'))$.

GP posterior: $\mathcal{GP}(m_{\text{post}}(x) = k(x, \mathbf{x})[K(\mathbf{x}, \mathbf{x}) + \sigma^2_{\text{noise}} I]^{-1}\mathbf{y},$
$k_{\text{post}}(x, x') = k(x, x') - k(x, \mathbf{x})[K(\mathbf{x}, \mathbf{x}) + \sigma^2_{\text{noise}} I]^{-1} k(\mathbf{x}, x'))$



Mean prediction
but with a large variance

- **GP inference provides a natural confidence estimation on each point**

Application examples, working principals and findings (50 minutes)

1. Fmax Prediction
2. Layout hotspot detection
3. Design-silicon timing correlation
4. Outlier delay test
5. Novel functional test program selection
6. Selective test for parametric test cost reduction

Practical Academic Uncertain

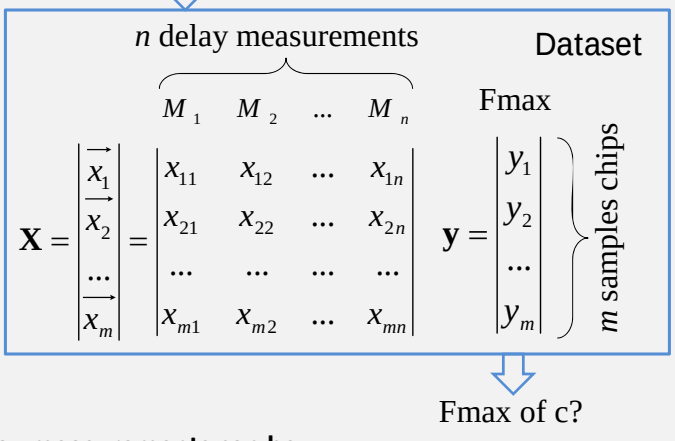
Application Examples

1. **Fmax Prediction**
2. Layout hotspot detection
3. Design-silicon timing correlation
4. Outlier delay test
5. Novel functional test program selection
6. Selective test for parametric test cost reduction

Practical Academic Uncertain

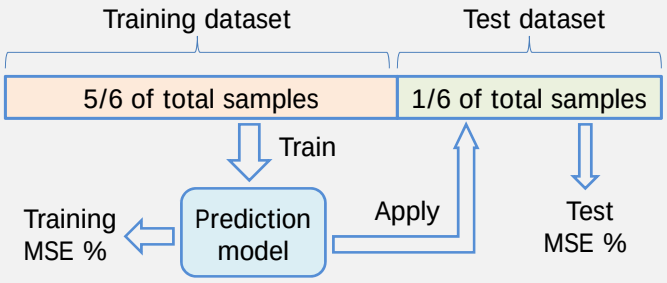
Recall: An Application Example – Fmax Prediction

$\vec{x} = x_1 \quad x_2 \quad \cdots \quad x_n$ (a new chip c)



- **Delay measurements can be**
  - FF based, pattern based, path based, or RO based

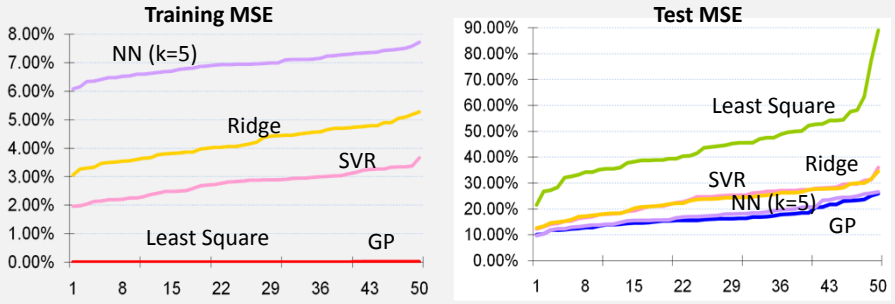
Recall: Experiment Setup



- **Training MSE% - Show how well the model fits the data**
- **Test MSE% - Show how well the model generalizes**
  - Test MSE% is what an application cares about

Li-C. Wang Tutorial - April 24th, 2015 135

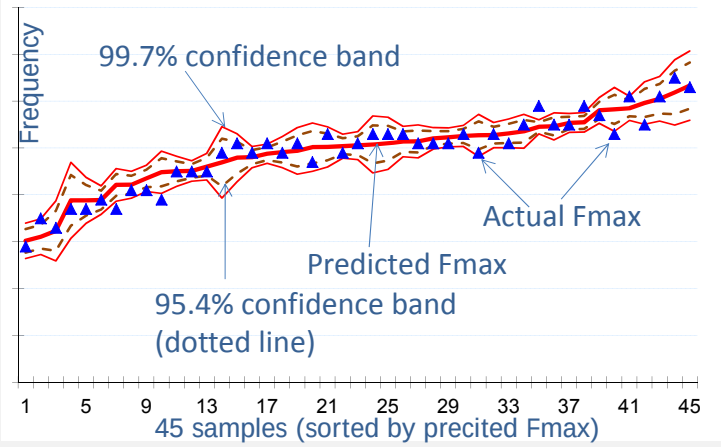
Adding SVR and GP To The Result



- **Support Vector Regression (SVR)**
  - Perform similarly to Ridge
  - SVM is good for classification and novelty detection
- **Gaussian Process (GP)**
  - Gives the best result

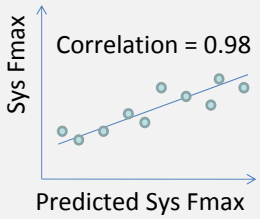
Li-C. Wang Tutorial - April 24th, 2015 136

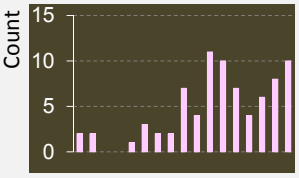
How Each Fmax Is Predicted By Others?



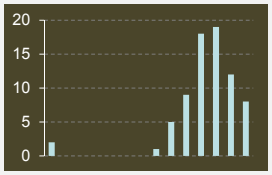
- **Each of the Fmax is predicted by a model built on the other 44 cores**
- **GP provides a confidence interval**

System Fmax Prediction





Fmax Freq. (lot1 – 79 cores)

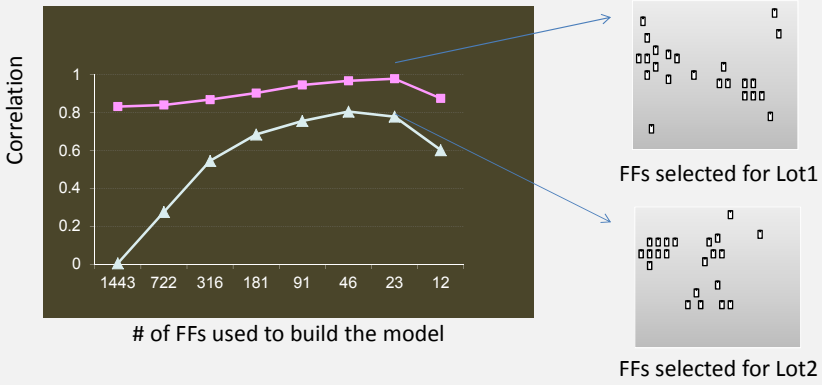


Fmax Freq. (lot2 – 74 cores)

Correlation	Lot1	Lot2
Predictive model	0.98	0.87
Best single FF	0.83	0.55

- **Structural Fmax on 1443 FFs were measured**
- **5/6 samples were used to build a predictive model for system Fmax**
- **Model was applied on remaining 1/6 samples**
- **See Janine Chen et al. (ASP-DAC 2010)**
  - "Correlating System Test Fmax with Sturctural Test Fmax and Process Monitoring Measurements"
  - **http://mtv.ece.ucsb.edu/licwang/PDF/2010-ASPDACb.pdf**

A More Interesting Question – Selective FF Model



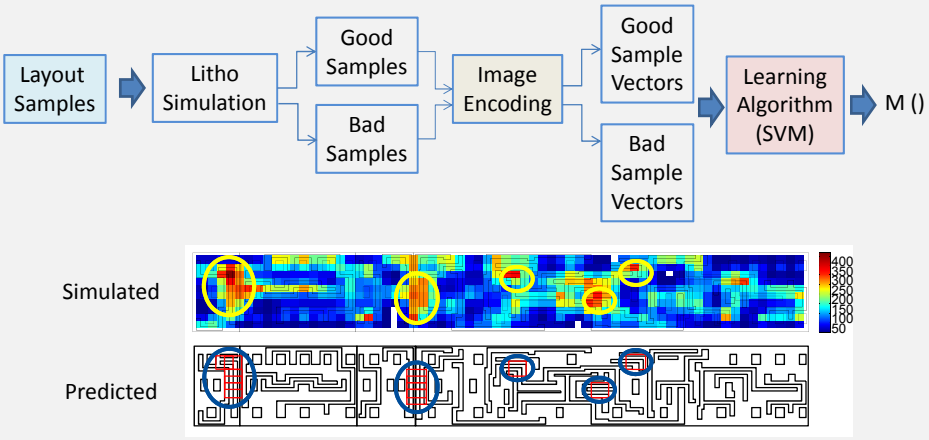
- **GP provides a weight to rank features (in this case, FFs)**
  - We use it to search for the small set of FFs to apply
- **Investigate across multiple lots**
  - Result is inconclusive

Application Examples

1. Fmax Prediction
2. **Layout hotspot detection**
3. Design-silicon timing correlation
4. Outlier delay test
5. Novel functional test program selection
6. Selective test for parametric test cost reduction

Practical Academic Uncertain

Binary Classification – Layout Hotspot Detection

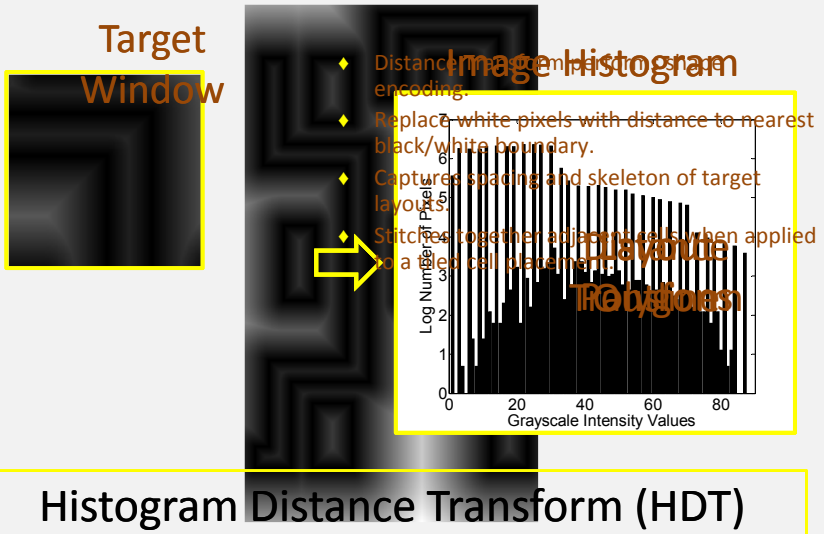


- **After learning, the SVM model becomes a surrogate for the Litho simulator**
  - But it is much faster
- **See Gagi Drmanac et al. (DAC 2009)**
  - "Predicting Variability in Nanoscale Lithography Processes"
  - http://mtv.ece.ucsb.edu/licwang/PDF/2010-DACb.pdf

Two Fundamental Issues

- **How layout is represented**
  - So that similarity between two layout samples can be captured?
- **How big is a layout sample?**
- **The choices have non-trivial influence to the result**

Layout Representation

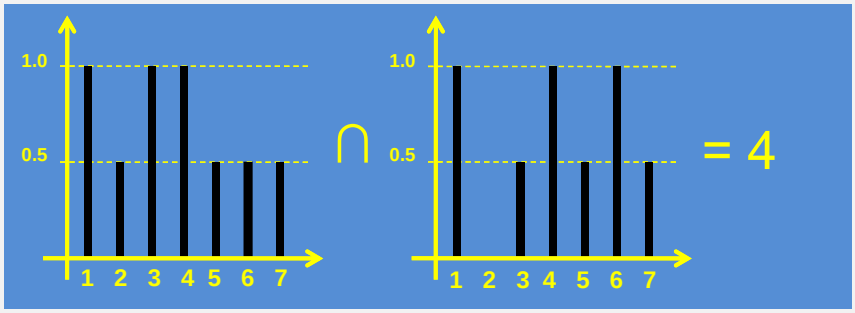


- Distance transform image encoding.
- Replace white pixels with distance to nearest black/white boundary.
- Captures spacing and skeleton of target layout.
- Stitches together adjacent cells when applied to a tiled cell layout.

Histogram Distance Transform (HDT)

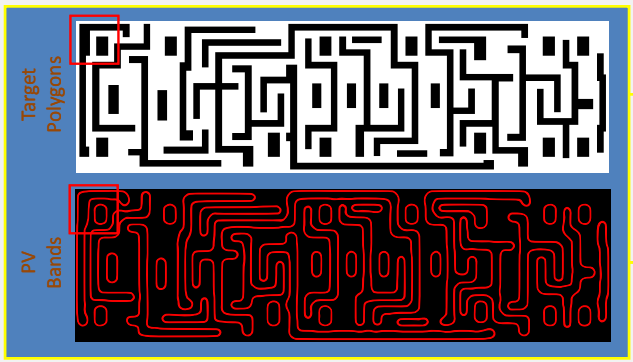
Similarity Measure

- **Histogram Intersection Kernel**
  - KHI(x, y) = ∑min(xi, yi)
  - xi yi correspond to the contents of histogram bins.
- **The larger the intersection the more similar the histograms are**
- **This kernel is proved positive semi-definite**



Extracting Layout Samples

- Start with a 100 x 100 pixel window
- Scan image with 50 pixels step for 50% overlap
- 1 image pixel = 32 nm target area



Generate SVM Dataset

```
- 1 1:3.4 2:3.4 3:5.2
+1 1:1.1 2:6.8 3:0.3
- 1 1:3.2 2:1.7 3:0.9
+1 1:4.1 2:1.4 3:1.0
- 1 1:1.8 2:2.2 3:2.3
+1 1:5.9 2:3.7 3:4.3
+1 1:2.7 2:0.9 3:7.2
- 1 1:1.6 2:3.7 3:9.1
- 1 1:1.7 2:5.3 3:4.0
+1 1:3.7 2:4.6 3:0.3
- 1 1:1.3 2:2.2 3:2.2
+1 1:2.1 2:1.7 3:0.1
```


Challenges

- **The work was discontinued because**
  - Not sure if it provide either **accuracy** and/or **speed** benefit to the rule-based approach
  - Or, learning should be used to extract rules, not just a prediction model
  - It should be applicable to the next technology node – difficult to obtain data

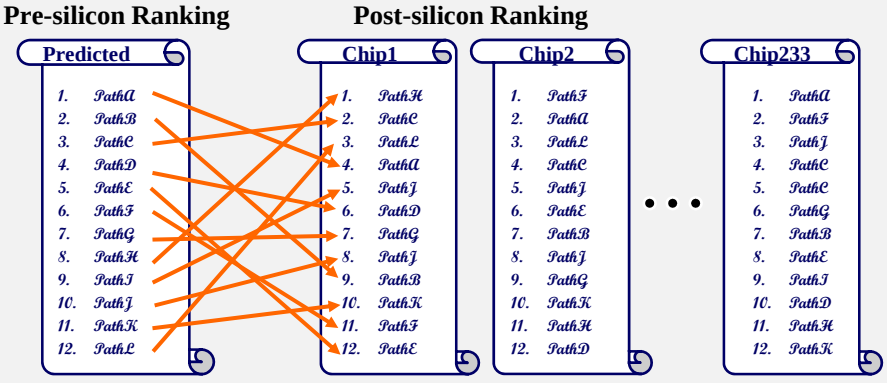
Application Examples

1. Fmax Prediction
2. Layout hotspot detection
3. **Design-silicon timing correlation**
4. Outlier delay test
5. Novel functional test program selection
6. Selective test for parametric test cost reduction

Practical Academic Uncertain

Li-C. Wang Tutorial - April 24th, 2015 147

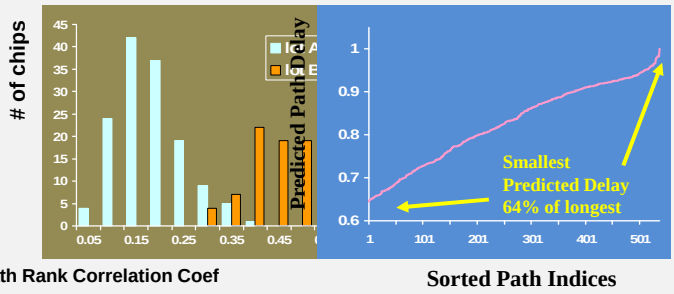
Design-Silicon Timing Correlation



- **233 Freescale microprocessors chips**
- **495 simple latch-to-latch**
- **Calculate Spearman's correlation coefficient for each chip**
- **See Pouria Bastani et al. (ITC 2008)**
  - "Diagnosis of design-silicon timing mismatch with feature encoding and importance ranking – the methodology explained"

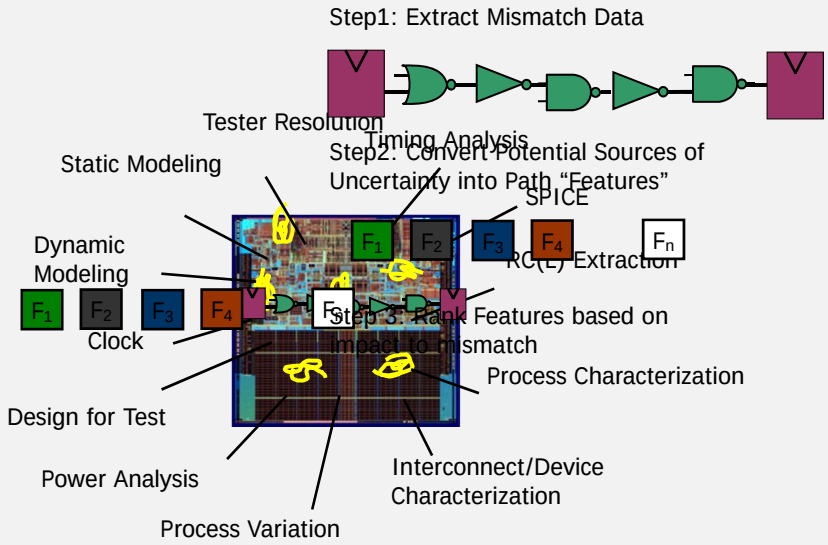
Li-C. Wang Tutorial - April 24th, 2015 148

Example of timing mismatch

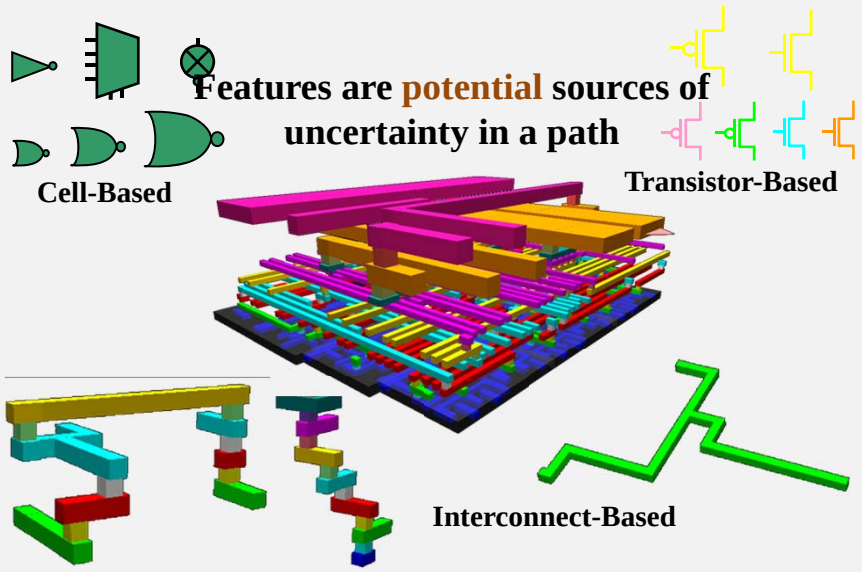


- **Lot A 141 and lot B 92 Freescale microprocessors chips**
- **Ranking correlation between predicted and observed**
- **Similar results shown for complex paths**

Initial Idea – Feature Ranking

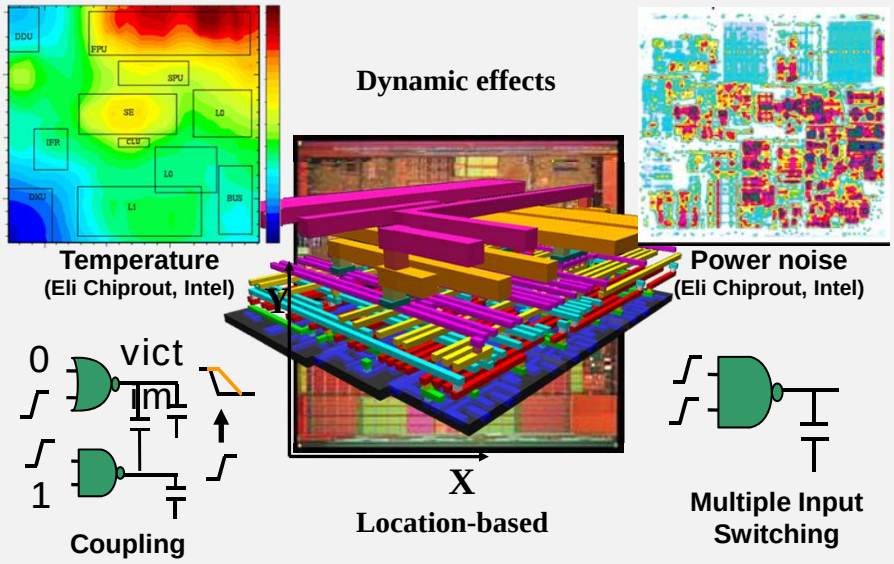


Main Issue – Feature Generation



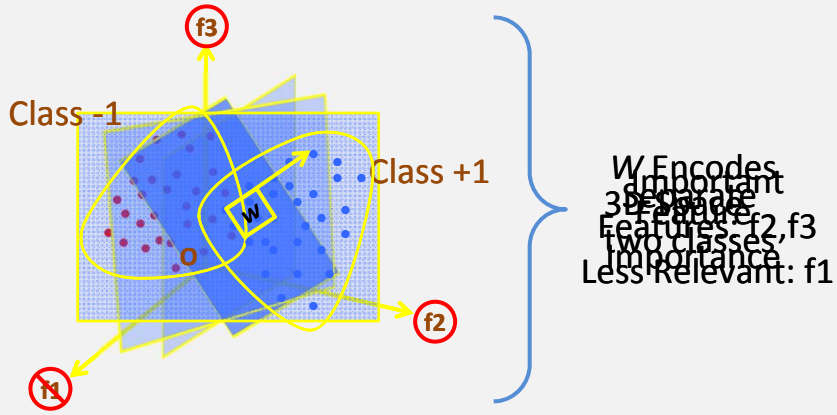
Li-C. Wang Tutorial - April 24th, 2015 151

Feature Generation



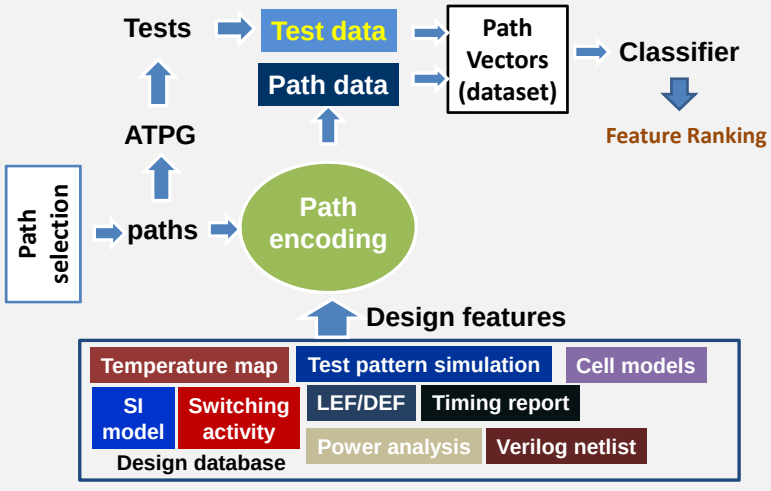
Li-C. Wang Tutorial - April 24th, 2015 152

SVM Classifier Based Feature Ranking



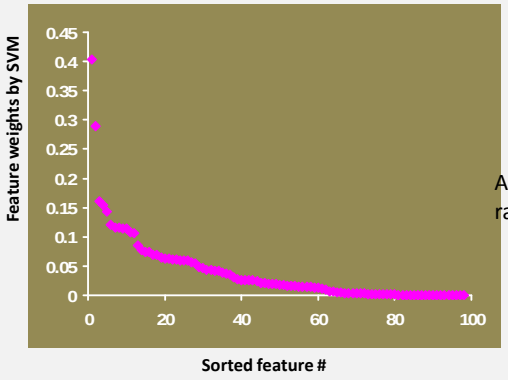
- Feature importance is characterized by the direction of the hyperplane

Summary Of The Methodology



- The learning does not have to be for feature ranking
- Once setup, we can also apply classification rule learning to extract rules

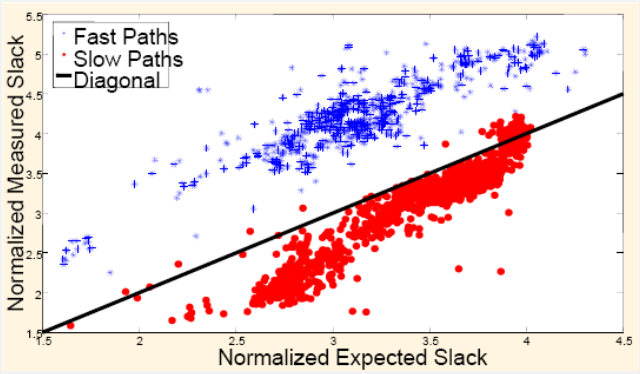
Binary Classification – Feature Ranking



An example of feature ranking output

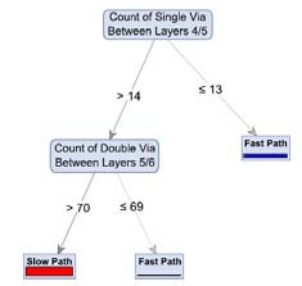
- **Typical action – investigate the top features**
- **See Pouria Bastani et al. (DAC 2008)**
  - "Statistical Diagnosis of Unmodeled Systematic Timing Effects"
  - http://mtv.ece.ucsb.edu/licwang/PDF/2008-DAC.pdf

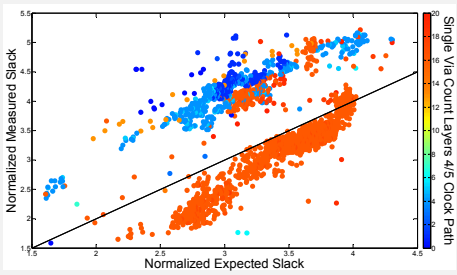
A Practical Application



- **Application to explain this example of timing abnormality**

Binary Classification – Tree Learning

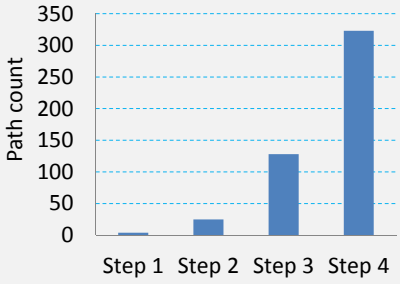


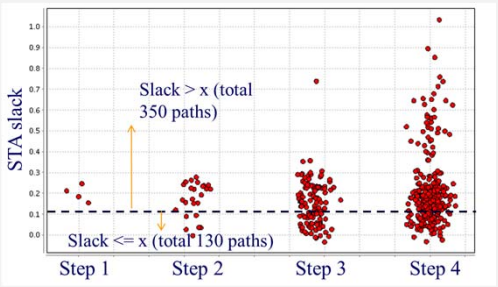


Validation of the tree model

- **Design features extracted from timing reports and GDSII**
- **Tree model: There are > 14 single vias between layers 4/5 and > 70 double vias between layers 5/6**
- **One can validate a tree model by visualizing the colored scatter plot**

Another Practical Application





- **Fifteen 4-core processor parts**
- **480 critical paths from AC delay tests (shown on top 4 freq. steps)**
- **Many are STA non-critical (Slack > x)**
- **See Janine Chen et al. (ITC 2010)**
  - "Mining AC Delay Measurements for Understanding Speed-limiting Paths"
  - http://mtv.ece.ucsb.edu/licwang/PDF/2010-ITCa.pdf

Binary Classification – Rule Learning (M >> k)

12,248 silicon non-critical paths vs. 158 silicon critical paths

- **There are STA-critical 12,248 paths activated by patterns**
  - with slacks <= x
  - Do not show up as critical paths in top 4 frequencies
- **For 480 failing FFs from top 4 frequency steps**
  - 169 failing FFs with unique path activation
  - 169 sure silicon critical paths
  - Only 11 paths have slacks <= x
  - 158 silicon critical but STA non-critical paths
- **Question: What are unique about the 158 paths?**
  - Use 12,248 silicon non-critical paths as the basis for comparison

Binary Classification – Rule Learning (M >> k) – Feature Creation

- **92 path features**
  - Basic information: for example, whether the path is a half-cycle path, transition directions on N/PMOS, etc.
  - Timing statistics: delay information from STA timing report.
  - Usage of Vt devices: counts of various Vt devices used on the path, i.e. high, low, regular, etc.
  - Cell type: features of the usage of various important cell types.
  - RC related: features capturing the load information on the path.
  - MUX related: special features focusing on the usage of some MUX cells.
  - Location: describing location of the path.
- **16 features correlate to others**

Binary Classification – Rule Learning (M >> k) – Example Result

Rule#	Rule	$S_{critical}$	$S_{non-critical}$
#1	CMAC $\in [8,14] \wedge$ CCT $\in [6,7]$	68	1
#2	CMAC $\in [8,14] \wedge$ PS $\in [247,\infty]$	47	0
#3	VRC $\in [0.565,0.571] \wedge$ PS $\in [247,\infty]$	47	0
#4	CMAC $\in [8,14] \wedge$ VRC $\in [0.565,0.571]$	46	0
#5	CMAC $\in [8,14] \wedge$ VRD $\in [0.520,0.545]$	46	0

- **A rule may not be the root cause**
- **Multiple rules may be reflecting the same cause**
- **For example,**
  - Rules #1,2,4,5 all involves CMAC feature
  - They actually reflect the same cause (later confirmed)
- **Finding: The methodology is practical – finding something missed by designer**
  - If one is willing to pay the price for setting up the infrastructure (features)

Li-C. Wang Tutorial - April 24th, 2015 161

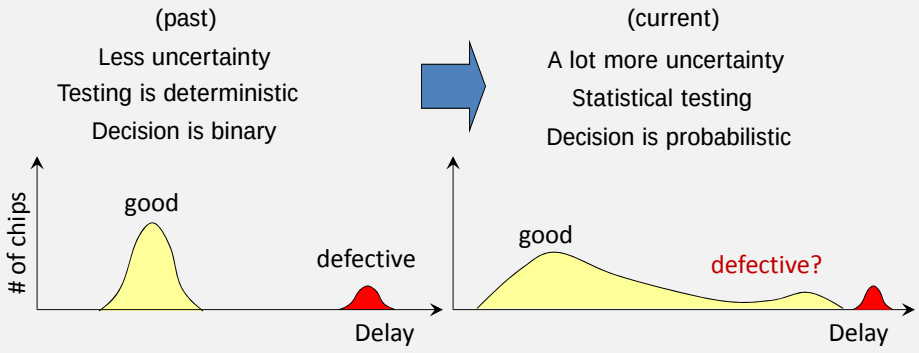
Application Examples

1. Fmax Prediction
2. Layout hotspot detection
3. Design-silicon timing correlation
4. **Outlier delay test**
5. Novel functional test program selection
6. Selective test for parametric test cost reduction

Practical Academic Uncertain

Li-C. Wang Tutorial - April 24th, 2015 162

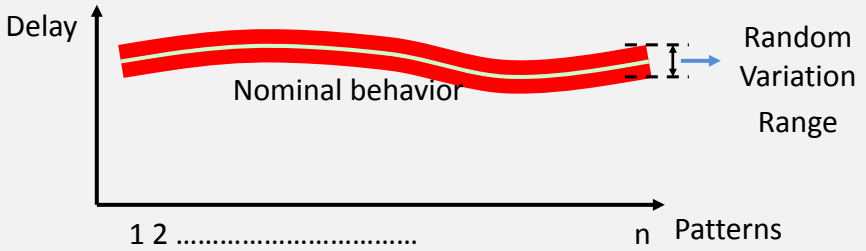
Recall: Parametric Delay Test



- **In the past, good and bad behavior can be clearly separated in a one-dimensional view**
  - Consider what being shown is the delay based on a pattern
- **Variations blur the boundary between good and bad**
  - Making it hard to separate in a one-dimensional view

Li-C. Wang Tutorial - April 24th, 2015 163

Delay Testing => Outlier Analysis View

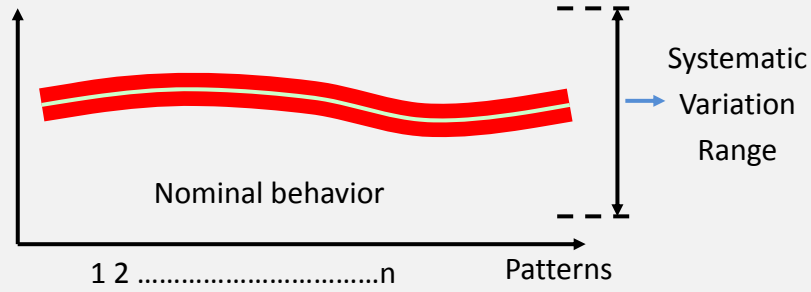


- **Traditional view is to set a threshold clock**
- **In the outlier view, one or more clocks are used to collect information on "normal delay behavior" of a part**
- **Then, we screen out outliers**

- **Effect of random variation:**
  - Collective timing behavior across multiple test patterns follows the trend, but individual pattern delay has fluctuation within a range

Li-C. Wang Tutorial - April 24th, 2015 164

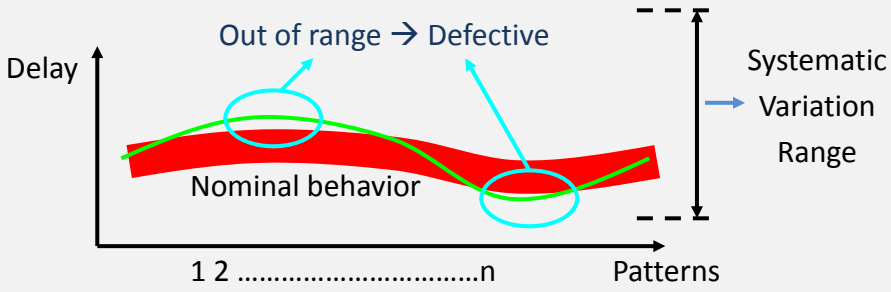
Outlier Delay Testing



- **Effect of systematic variation:**
  - Collective timing behavior is either fast or slow (shift up and shift down)
  - But the trend does not alter

Li-C. Wang Tutorial - April 24th, 2015 165

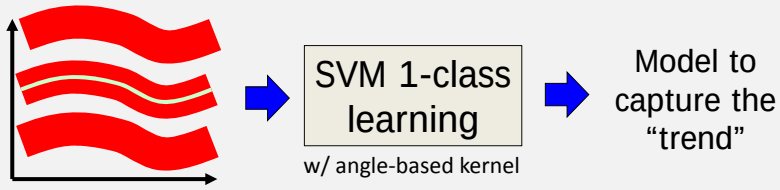
What Is An Outlier In Multivariate View



- **What is an outlier?**
  - An outlier has a collective timing behavior that does not follow the trend
  - It is a multivariate outlier – the outlier aspect is defined based on the behavior on the collection of tests
- **Screen out chips behave as an outlier**

Li-C. Wang Tutorial - April 24th, 2015 166

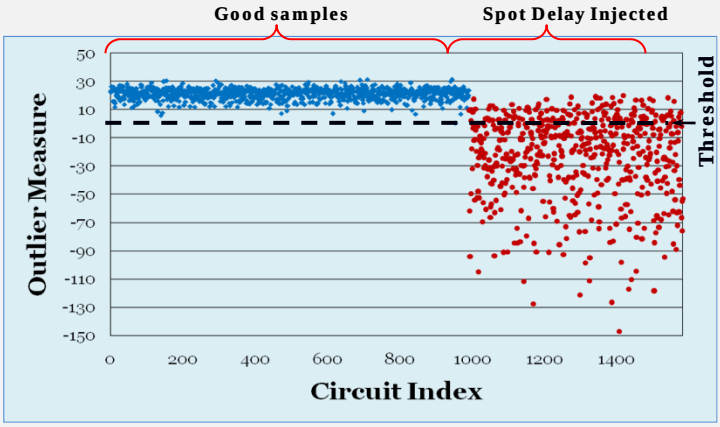
Apply 1-Class SVM



- **Formulate it as a one-class learning problem**
  - Learning the normal class
  - As a result, identify the outliers
- **This approach is variation-invariant and identifies multivariate outliers**
- **See Sean Wu et al. (ITC2008)**
  - "A Study of Outlier Analysis Techniques for Delay Testing"
  - http://mtv.ece.ucsb.edu/licwang/PDF/2008-ITCb.pdf

Li-C. Wang Tutorial - April 24th, 2015 167

Finding



- **Effective to capture subtle "small" delay defects under variations**
- **See Wu et al. (ITC2008)**
  - "A Study of Outlier Analysis Techniques for Delay Testing"
  - http://mtv.ece.ucsb.edu/licwang/PDF/2008-ITCb.pdf

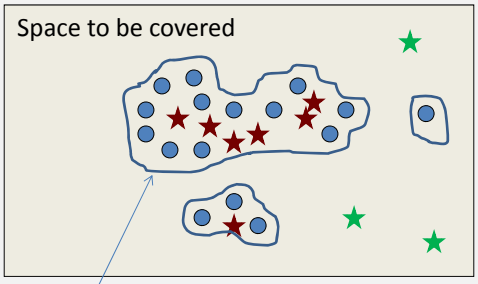
Li-C. Wang Tutorial - April 24th, 2015 168

Application Examples

1. Fmax Prediction
2. Layout hotspot detection
3. Design-silicon timing correlation
4. Outlier delay test
5. **Novel functional test program selection**
6. Selective test for parametric test cost reduction

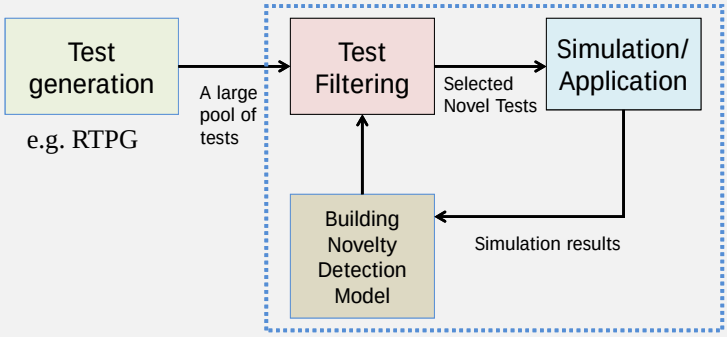
Practical Academic Uncertain

Novel Functional Test Program Selection



- **In SoC/Processor verification, tremendous amounts of test programs (e.g. assembly programs, instruction sequences) are simulated**
- **We applied novelty detection to identify "novel test programs" before simulation – to avoid simulation of ineffective sequences**
- **See ITC 2010 "A Kernel-Based Approach for Functional Test Program Generation"**
  - http://mtv.ece.ucsb.edu/licwang/PDF/2010-ITCb.pdf

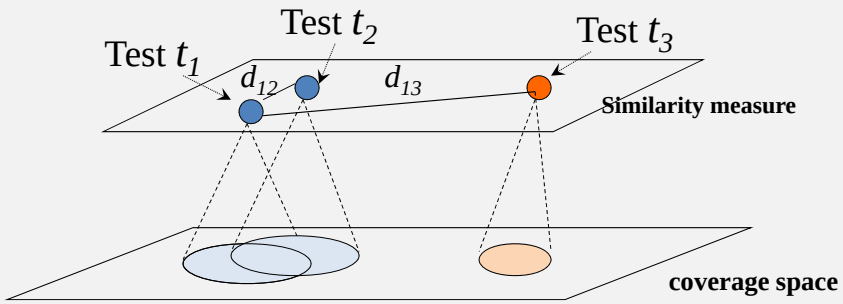
The Methodology for Reducing Simulation Cost



- **Novelty detection is used to identify novel tests for simulation/application**
  - Avoid applying ineffective tests
- **The key question: How to measure similarity between two tests?**

Li-C. Wang Tutorial - April 24th, 2015 171

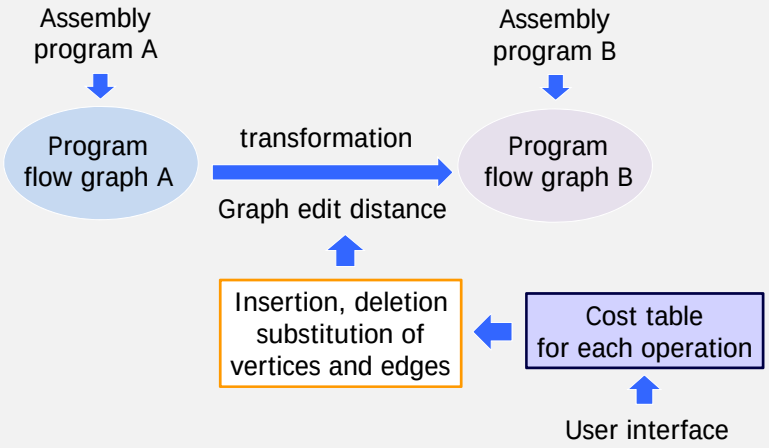
Challenge



- **Ideally, two tests are more similar if their covered spaces are more similar**
  - How to define such a kernel function?
- **See ITC 2010 "A Kernel-Based Approach for Functional Test Program Generation"**
  - **http://mtv.ece.ucsb.edu/licwang/PDF/2010-ITCb.pdf**

Li-C. Wang Tutorial - April 24th, 2015 172

Similarity Between Two Assembly Programs



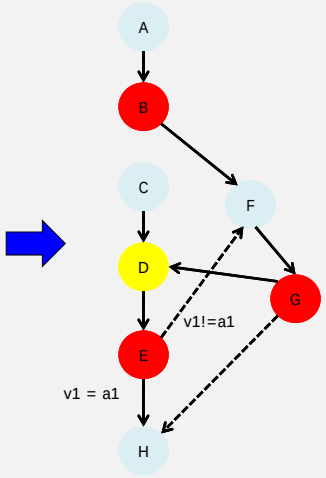
- **k(A,B) = similarity between two program graphs**
  - = shortest Graph Edit Distance (GED)
  - = minimum cost to transform A into B

Li-C. Wang Tutorial - April 24th, 2015 173

Program Flow Graph

- **A graph representation of an assembly program**
  - Vertex : instruction
  - Edge : actual execution flow

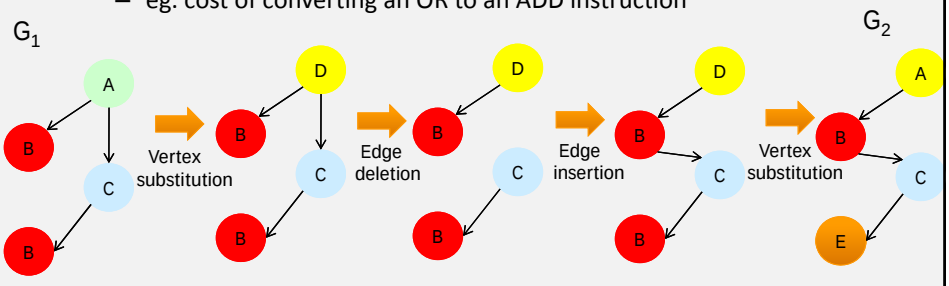
Address	Instruction		Vertex
4	add	a1,v0,2	A
8	j	0x22	B
12	addu	v1,v1,24	C
16	sb	v0,0(a1)	D
20	beq	v1,a1,0x28	E
22	sub	a0,a1,4	F
24	beqz	v0,0x16	G
28	addiu	v0,a2,10	H



Li-C. Wang Tutorial - April 24th, 2015 174

Graph Edit Distance

- A sequence of edit operations that transform a graph G_1 to a graph G_2
- Each operation is associated with a cost defined in a cost table
  - eg. cost of converting an OR to an ADD instruction



Li-C. Wang Tutorial - April 24th, 2015 175

MIPS Core

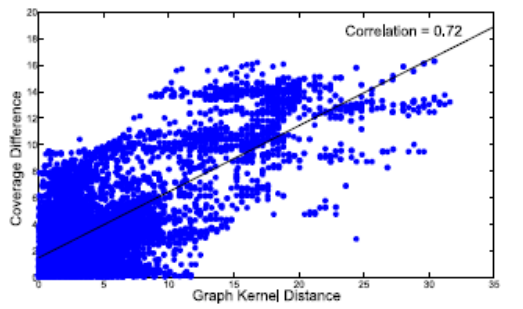
- A public domain 32-bit RISC processor design
- The RTL design of the Plasma CPU is described in VHDL with about 4800 lines of code
- The Plasma core executes all MIPS I(TM) user mode instructions

Li-C. Wang Tutorial - April 24th, 2015 176

Experimental Setup

- **Developed an RTPG for this core**
  - Ensure validity of randomly generated programs
- **RTPG produced 2000 programs**
  - Based on more than 50 types of instructions such as arithmetic, logic, branch, load, store and jump instructions
  - Various number of instructions (80-100) including a boot sequence (50 instruction)
- **Coverage metrics : statements, branches, expressions, conditions and toggle coverage**

Similarity vs Coverage



- **Coverage difference**
  - The difference of toggle coverage based on the two assembly programs
- **Graph kernel distance**
  - The distance computed between the two assembly programs
- **Correlation = 0.72**
  - The similarity measure captures some of the trend

Result

- **Each iteration**
  - Select 50 novel tests to apply
  - 5 iterations (250 tests) to achieve the maximal coverage (2000 tests)
- **Save at least 80% of simulation time**

Coverage metrics	Initial	1st	2nd	3rd	4th
Statements	81	87.9	92.8	94.3	**97.2**
Branches	80.6	87.7	91	93.4	**95.4**
Expressions	75	84.8	89.1	89.1	**89.1**
Conditions	88.5	92.3	92.3	92.3	**96.2**
Toggle	76.4	79.8	81.5	83.4	**84.8**

Li-C. Wang Tutorial - April 24th, 2015 179

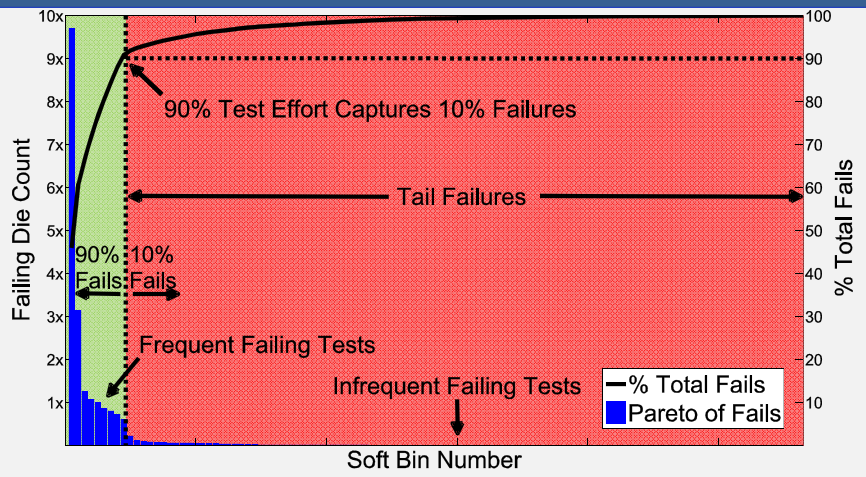
Application Examples

1. Fmax Prediction
2. Layout hotspot detection
3. Design-silicon timing correlation
4. Outlier delay test
5. Novel functional test program selection
6. **Selective test for parametric test cost reduction**

Practical Academic Uncertain

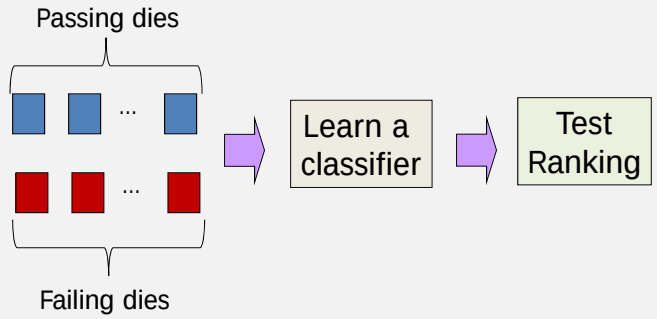
Li-C. Wang Tutorial - April 24th, 2015 180

Parametric Test Set Reduction



- **90% of the failing dies are captured by 10% of the tests**
- **See Gagi Drmanac et al. (ITC 2011)**
  - "Wafer Probe Test Cost Reduction of an RF/A Device by Automatic Testset Minimization"
  - **http://mtv.ece.ucsb.edu/licwang/PDF/2011-ITCb.pdf**

General Idea – Test Importance Selection

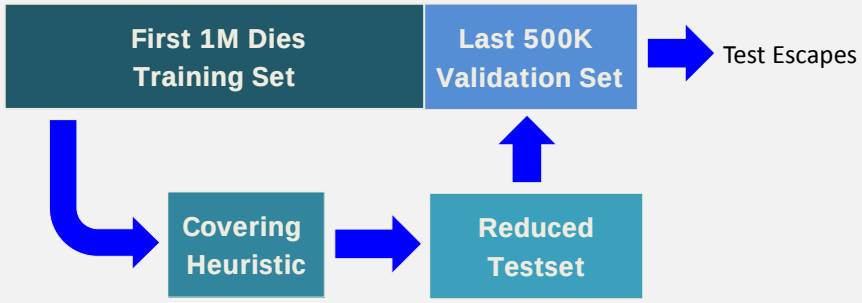


- **Given the test data (say based on 100K dies that are already tested to decide pass and fail), learn a classifier to separate the pass and fail**
- **From the classifier, extract test importance measures**
  - Rank tests for potential test removal

Some Result

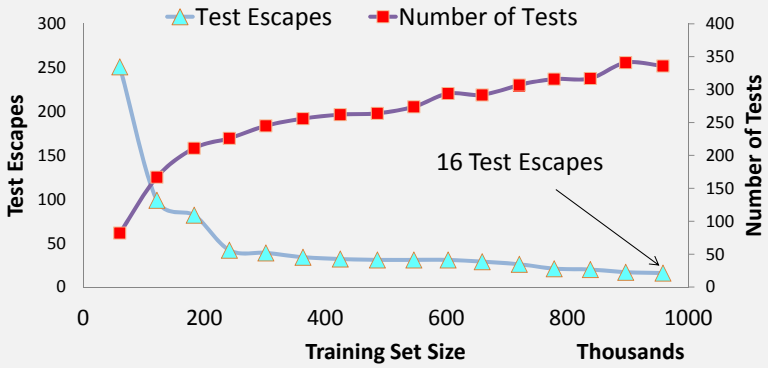
- **Based on a data set**
  - 700+ parametric wafer probe tests
  - RF/A device (Qualcomm)
  - 1.5M samples
- **Result**
  - **Learn from 10K-100K samples**
  - **Drop 30% of the tests**
  - **0.4% escape (capture in final test stage)**
  - **0.28% overkill**
- **Test team demands less than 50 DPPM impact – result not acceptable**
- **See Gagi Drmanac et al. (ITC 2011)**
  - "Wafer Probe Test Cost Reduction of an RF/A Device by Automatic Testset Minimization"
  - **http://mtv.ece.ucsb.edu/licwang/PDF/2011-ITCb.pdf**

What If We Have Large Enough Data?



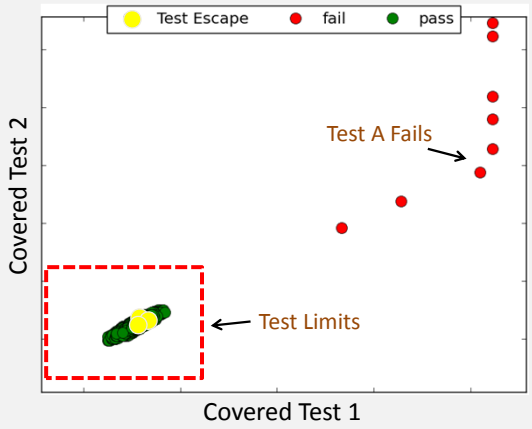
- **Don't apply any learning – simply solve a covering problem to get a baseline result**
- **Solve a covering Problem – Find the minimum test set that cover ALL defective dies saw in the 1M sample set**

Result Of Covering Based Approach



- **Using 1M dies,**
  - 16 test escapes and 336 tests kept
  - 77 Derived Tests
  - 259 Measuring Tests
- **Roughly 55% reduction in the number of tests with only 32DPPM impact**

Question: Can Statistical Learning Improve Result Further

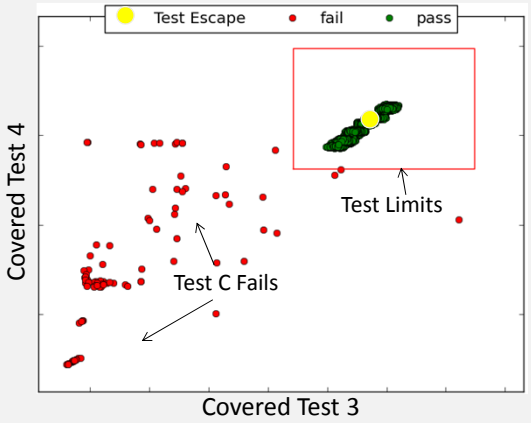


	Test A	Test 1	Test 2
Test A	1	0.97	0.96
Test 1	0.97	1	0.92
Test 2	0.96	0.92	1

Correlation Matrix

- **Results based on 1M dies seem perfect**
- **3 test escapes occur in the remaining 0.5M dies**
  - How to statistically predict these?
  - The idea of building a "better" outlier model won't help

Question: Can Statistical Learning Improve Result Further?



	Test L	Test 3	Test 4
Test C	1	0.98	0.99
Test 3	0.98	1	0.98
Test 4	0.99	0.98	1

Correlation Matrix

- **Similarly, tests 3 and 4 are highly correlated to test C**
  - Based on the passing dies
- **1M dies show perfect screening**
  - 1 test escape in the remaining 0.5M dies

Li-C. Wang Tutorial - April 24th, 2015 187

Lessons Learned

- **A statistical learning approach tries to generalize beyond what it sees in a given dataset**
  - That should be why a statistical approach is better than a simple covering approach that only tries to fit the given data
- **However, even though a statistical approach gives good result, the approach may not make sense**
  - Need to be better than the simple approach
  - Need to make the comparison with large dataset
- **We are intrigued by a complex algorithm with beautiful math**
  - In practice, with enough data, perhaps a naïve simple approach will work just fine
- **In data mining, data is more important than algorithm**

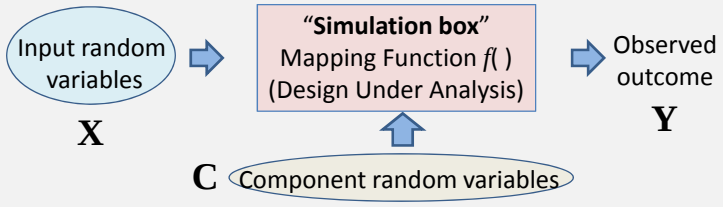
Li-C. Wang Tutorial - April 24th, 2015 188

Applications in Pre-Silicon Simulation (50 minutes)

Two Sets of Papers

- **Improve Simulation Efficiency**
  - "Machine learning in simulation based analysis" – ISPD 2015
    - **http://mtv.ece.ucsb.edu/licwang/PDF/2015-ISPD.pdf**
  - "Novel Test Detection to Improve Simulation Efficiency – A Commercial Experiment" – ICCAD 2012
    - **http://mtv.ece.ucsb.edu/licwang/PDF/2012-ICCAD.pdf**
- **Improve Testbench Quality**
  - "Simulation Knowledge Extraction and Reuse in Constrained Random Processor Verification" – DAC 2013
    - **http://mtv.ece.ucsb.edu/licwang/PDF/2013-DAC.pdf**
  - "Data Mining in Functional Debug" – ICCAD 2014
    - **http://mtv.ece.ucsb.edu/licwang/PDF/2014-ICCAD.pdf**
  - "Data Mining in Functional Test Content Optimization" – ASPDAC 2015
    - **http://mtv.ece.ucsb.edu/licwang/PDF/2015-ASPDAC.pdf**

Generic Problem Setting

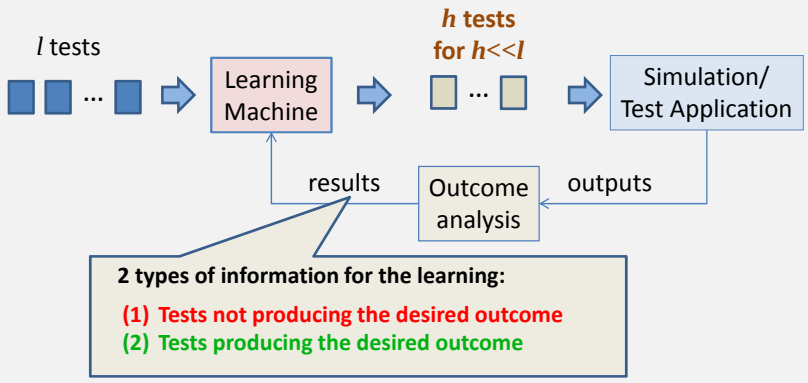


- **Inputs to the "simulation box"**
  - X (tests): e.g. regression tests, waveforms, assembly programs, etc.
  - Optional C: e.g. device parameters to model statistical variations
- **Output from the "simulation box":**
  - Y: e.g. simulation traces, waveforms, coverage reports

Two Questions To Ask: 1st Question

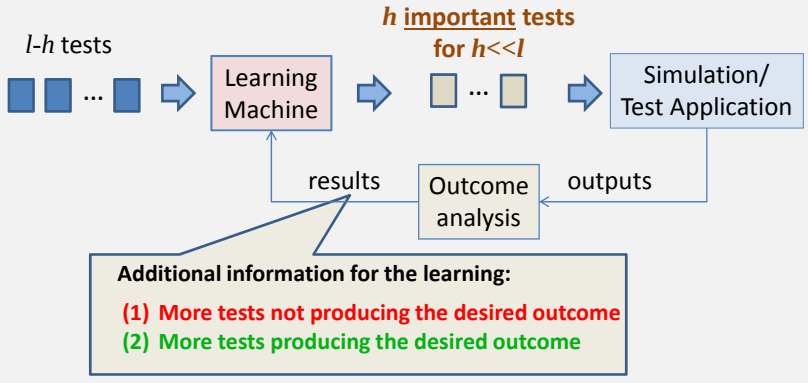
- **Concern the simulation efficiency (or cost)**
- **I have too many tests to simulate/apply**
- **Before simulation/application, how can I predict those important ones to apply to reduce cost?**

Our Solution: Iterative Learning for Test Ranking



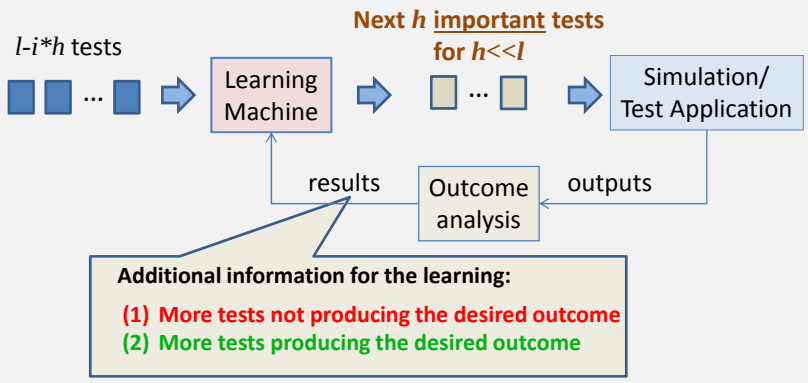
- **Iteration 0: select a small *h* tests to apply first**

Our Solution: Iterative Learning for Test Ranking



- **Iteration 1: learn and select a small *h* important tests**

Our Solution: Iterative Learning for Test Ranking

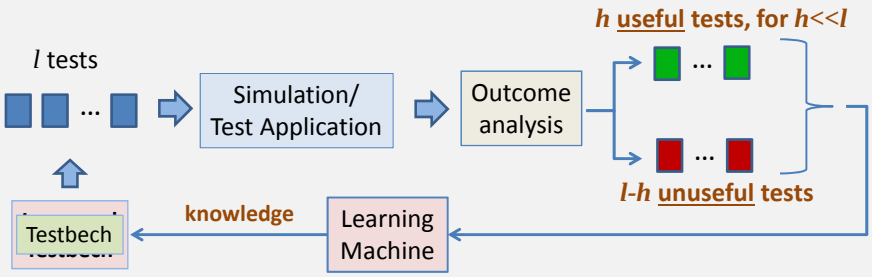


- Iteration *i*: learn and select a small *h* important tests

Two Questions To Ask: 2nd Question

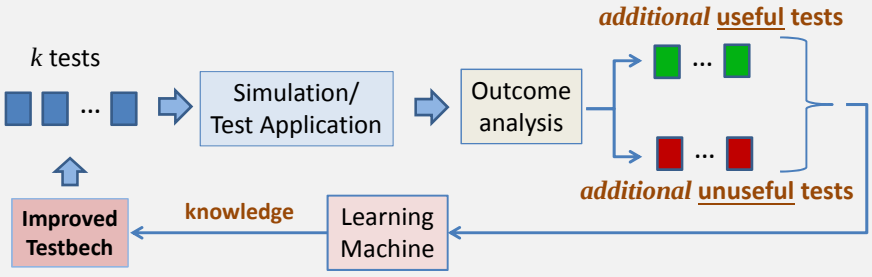
- Concern testbench quality (test content optimization)
- I apply many tests but only a few are useful
- How can I learn from the useful tests and produce additional useful tests?

Our Solution: Iterative Learning for Knowledge Extraction



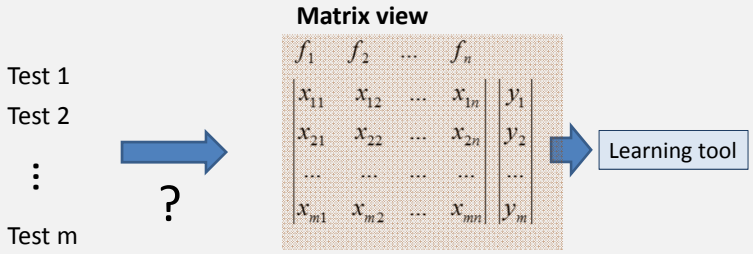
- Iteration *0*: learn from useful/unuseful tests

Our Solution: Iterative Learning for Knowledge Extraction



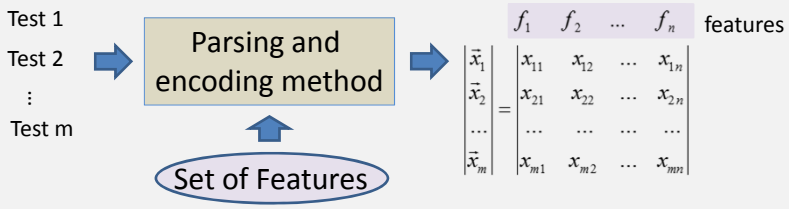
- Iteration *i*: learn from more useful/unuseful tests

Recall: Fundamental Question 1



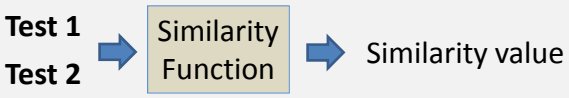
- **A learning tool takes data as a matrix**
- **Suppose we want to analyze *m* functional tests**
  - Each test comprises 100-150 instructions
- **How do I apply learning on objects which are not represented as a vector?**

Explicit Approach – Feature Encoding



- **Need to develop two things:**
  - 1. Define a set of features (e.g. Arch states variables)
  - 2. Develop a parsing and encoding method based on the set of features (e.g. using Arch. Simulation)
- **Does learning result depend on the features and encoding method? (Yes!)**
  - That's why learning focuses on "learning the features"

Implicit Approach – Kernel Based Learning



- **Do not explicitly define a set of features**
- **Instead, define a similarity function (kernel function)**
  - It is a computer program that computes a similarity value between any two tests
- **Most of the learning algorithms can work with such a similarity function directly**
  - Without using a matrix data input
- **Does learning result depend on the kernel? (Yes!)**
  - That's why learning is about "learning good kernels"

Li-C. Wang Tutorial - April 24th, 2015 201

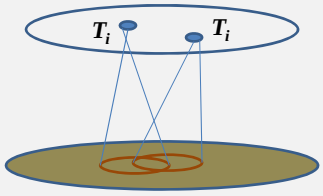
RTL Simulation Context

Li-C. Wang Tutorial - April 24th, 2015 202

Example 1: Test Filtering, Test Ranking

- **Approach – implicit: use kernel function**
- **Papers:**
  - (Adaptive kernel) ISPD 2015: "Machine learning in simulation based analysis"
  - (Simulation based kernel) ICCAD 2012: "Novel Test Detection to Improve Simulation Efficiency - A Commercial Experiment"
  - (Assembly program graph based kernel) ITC 2010 "A kernel-based approach for function test program generation"

High-level space (Tests)
(visible to the learning machine)



Low-level space (Coverage)
(Invisible to the learning machine)

Enabling learning

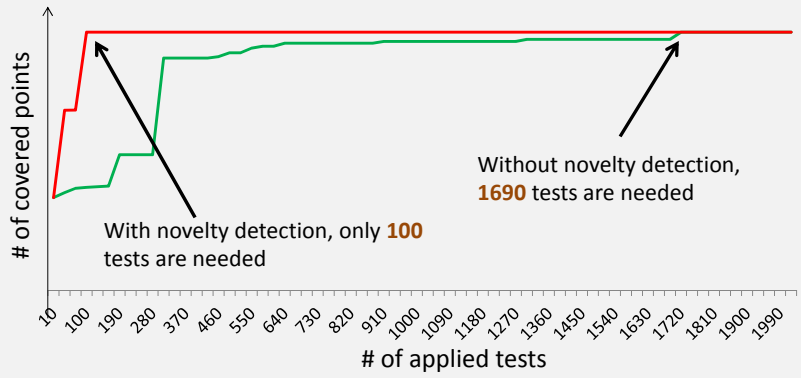
$k(Ti, Tj)$ Similarity between two tests

Fundamental question:
Does k( ) reflect the actual similarity of the two tests in the low-level (coverage) space?

Learning is about learning a mapping from a high-level space to a low-level space

Li-C. Wang Tutorial - April 24th, 2015 203

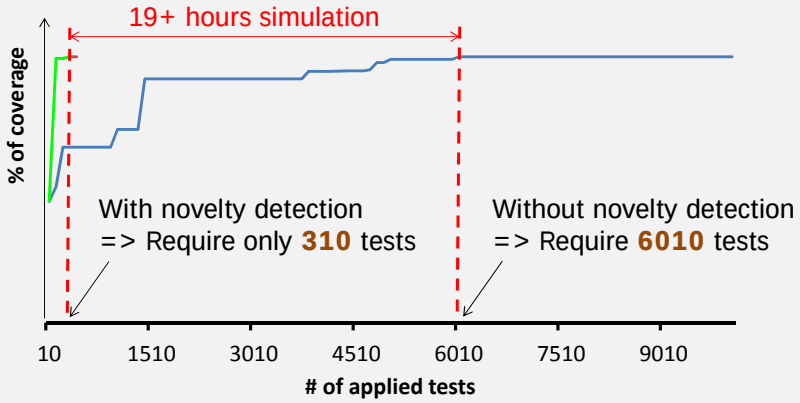
One Example



- **Design: 64-bit Dual-thread low-power processor (Power Architecture)**
- **Each test is with 50 generated instructions**
- **Roughly saving: 94%**

Li-C. Wang Tutorial - April 24th, 2015 204

Another Example

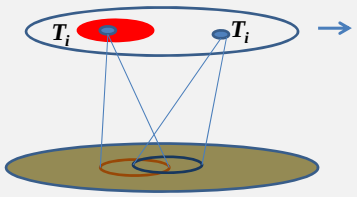


- **Each test is a 50-instruction assembly program**
- **Tests target on Complex FPU (33 instruction types)**
- **Roughly saving: 95%**
  - Simulation is carried out in parallel in a server farm

Example 2: Test Knowledge Extraction

- **Approach – explicit: use a set of features**
- **Papers:**
  - ICCAD 2014: "On Application of Data Mining in Functional Debug"
  - DAC 2013: "Simulation Knowledge Extraction and Reuse in Constrained Random Processor Verification"
  - ITC 2010 "Functional Test Content Optimization for Peak Power Validation - An Experimental Study"

High-level space (Tests)
(visible to the learning machine)



Low-level space (Signal activities)
(Invisible to the learning machine)

f_1 f_2 … f_n (features)

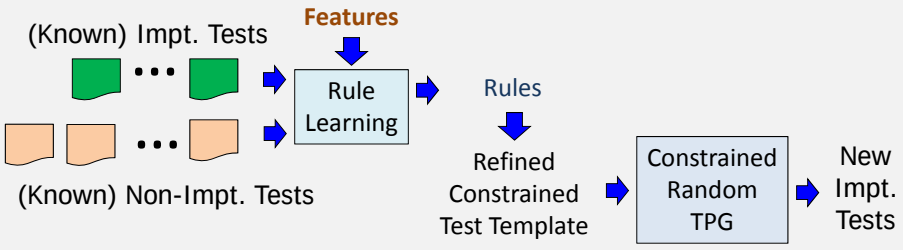
$Ti: (xi_1, \ldots, xi_n)$
$Tj: (xj_1, \ldots, xj_n)$

Any features good for differentiating them?

Fundamental question:
Do we include all relevant features?

Learning is about learning a mapping from a high-level space to a low-level space

Knowledge Extraction Flow (DAC 2013)

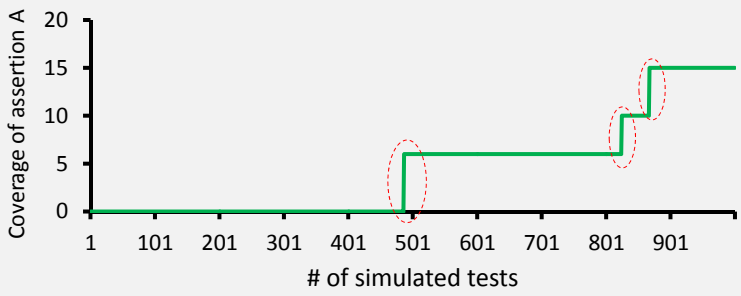


- **Encode each test with a set of features**
- **Discover rules (combination of features) to describe the special properties of important tests**
- **Use them to produce more important tests**
- **The learning can be applied iteratively on newly-generated important tests**

Feature Development

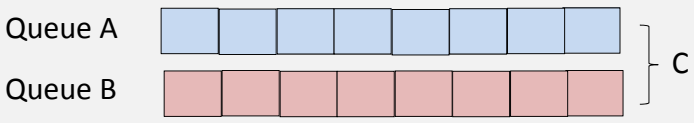
- **Feature development is mostly one-time cost**
  - Reused for multiple generations of design
- **Three categories**
  - Architecture state variables
  - Microarchitecture state variables
    - Special registers, eg. MSR, XER, L1CSR, and etc
  - Instruction-based features
    - Inst types, data patterns
    - Virtual/real addresses, associated page attributes
    - Data dependency
    - Address collision

Example #1



- **Assertion A depends on filling up a queue A**
  - Assertion A is hit by 3 tests out of 1000
- **Learn a rule based on these 3 novel tests**

Assertions B,C



- **Concern cache miss caused by a load instruction**
- **When there is a miss, there are two (possibly overlapping) cases to record**
  - Queue A stores the instruction for case A
  - Queue B stores the instruction for case B
- **Assertion A: queue A is full**
- **Assertion B: queue B is full**
- **Assertion C: both queues are full**

Result of Example #1

	# of tests hitting the assertions	
	Original 1000	New 200
Assertion A	3	33
Assertion B	0	9
Assertion C	0	5

- One example rule can be illustrated as the following
  - There is a lmw (load-multiple-word) instructions
  - The page associated with the lmw address is not guarded
  - The destination register of lmw is prior to G20
- Improve coverage on assertion A
  - Also cover related assertions B and C

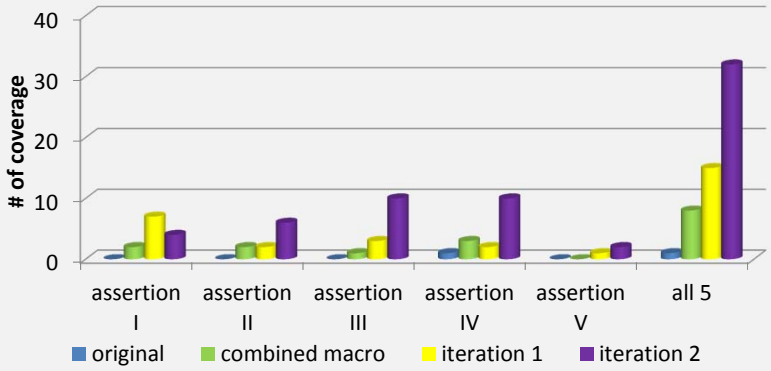
Example #2

- Five assertions of interest-I, II, III, IV, V
- Only assertion IV was hit by one test out of 2000
- Conceptually, each assertion i comprises
  - Satisfying a local state condition CondW[i]
  - Satisfying another global condition CF
  - CondW[i] and CF happen in the same clock cycle

Illustration of example rules:

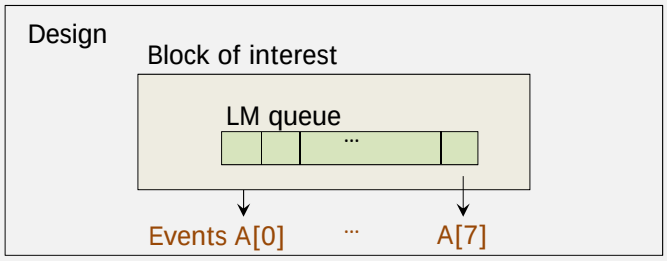
Rule for CondW[i]	There is a lfd instruction and the instructions prior to the lfd are not memory instructions whose addresses collide with the lfd
Initial Rule for CF	There is a mulld instruction and the two multiplicands are larger than 2^{32}

Result of Example #2



- **Combining the two rules:**
  - With 100 tests, cover 4 assertions
- **Refine the rule for condition CF**
  - Iteration 1: With 100 tests, cover all 5 assertions
  - Iteration 2: Increase coverage

Example #3



- **Interested in activating events A[0]-A[7]**
- **We know how to constrain the PRTP to produce tests likely to cause activities in the block**
- **Initially, we observe some coverage on A[0] and A[1], but not other events in the family**

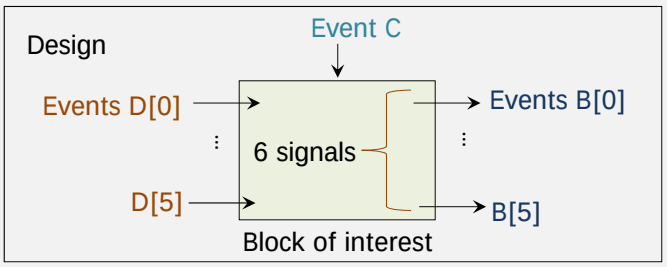
Result of Example #3

Test set	Initial	Iteration 1	Iteration 2
# of tests	400	100	50
A[0]	10	3	**72**
A[1]	17	11	**59**
A[2]	0	10	**71**
A[3]	0	10	**83**
A[4]	0	4	**79**
A[5]	0	2	**97**
A[6]	0	1	**96**
A[7]	0	1	**87**

- **Iteration 1:**
  - Learning rules based on the tests activating A[0] and A[1]
  - Applying rules to generate 100 new tests
- **Iteration 2:**
  - Learning rules based on good tests found in iteration 1
  - Applying refined rules to generate 50 new tests

Li-C. Wang Tutorial - April 24th, 2015 215

Example #4



- **Interested in activating a family of events B[0]-B[5]**
- **Initially, no tests activate B[0]-B[5]**

- **Identify relevant events C and D[0]-D[5] to be observed and learned on**
- **C is an architecture feature, so just need to learn about how to activate the predecessor events D[0]-D[5]**

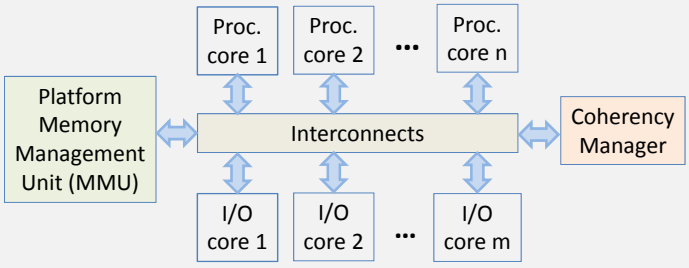
Li-C. Wang Tutorial - April 24th, 2015 216

Result of Example #4

Test set	Initial	Iteration 1	Iteration 2
# of tests	>30k	1200	100
B[0]	0	1	**2**
B[1]	0	0	**1**
B[2]	0	0	**1**
B[3]	0	16	**56**
B[4]	0	25	**61**
B[5]	0	26	**77**

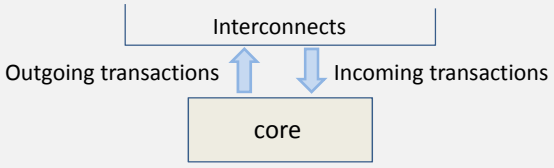
- **Similarly, Iteration 1:**
  - Learning rules based on the tests activating D[0] to D[5]
  - Applying rules to generate 1200 new tests to target on D[0] to D[5]
  - Some tests now activate B[0], B[3] to B[5]
- **Iteration 2:**
  - Learning rules based on these good tests found in iteration 1
  - Applying refined rules to generate 100 new tests

Example #5



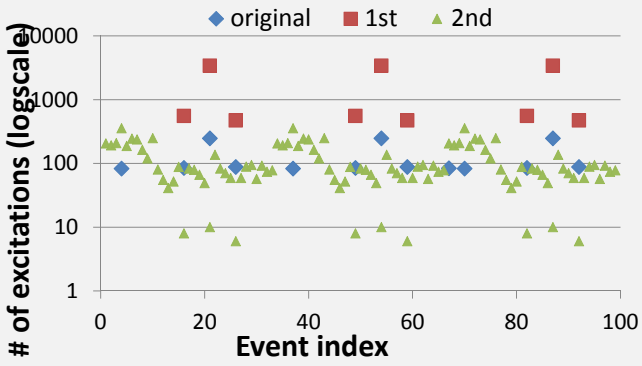
- **Example #5 was based on an experiment run on a platform design**

Example #5 (Platform)



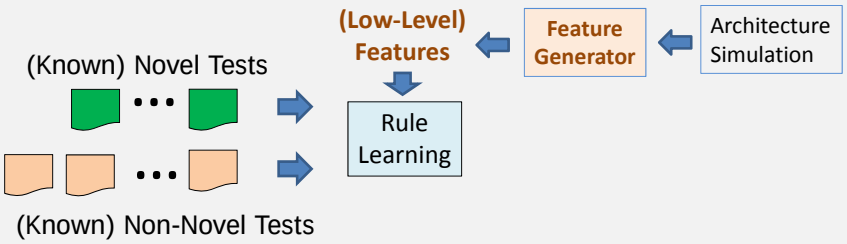
- **I/O ports monitoring activities on outgoing and incoming transactions**
- **Outgoing transactions**
  - Engineer has a good grasp of how to control the outputs of the core
- **Incoming transactions**
  - Result depends on the state and configuration of the system – Hard to control and generate tests for

Result of Example #5



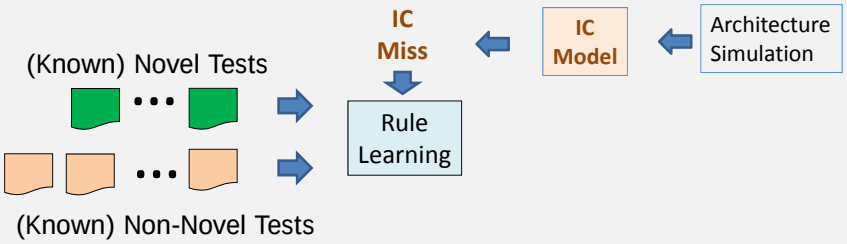
- **Interested in 99 events on incoming signals**
- **Initial 2K tests -> only 13 were excited (original)**
- **Learning two rules (subgroup discovery) – 1st and 2nd**

Feature Generator



- **In some application, only information available is from the architecture simulation**
  - Lack of low-level features
- **Solution: Develop a feature generator to produce low-level features to facilitate learning**
  - Comprises a library of manual created models

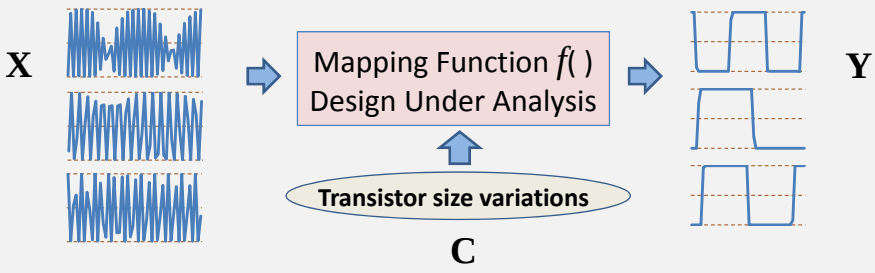
Feature Generator Example



- **Suppose the novel tests depend on the IC miss**
  - If architecture does not have that information, learning is not possible
- **Create a simpler IC model to approximate IC miss state**
  - This model might not be 100% accurate

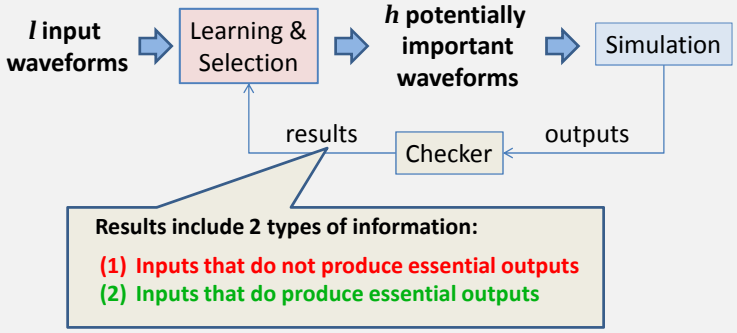
SPICE Simulation Context

SPICE Simulation Context



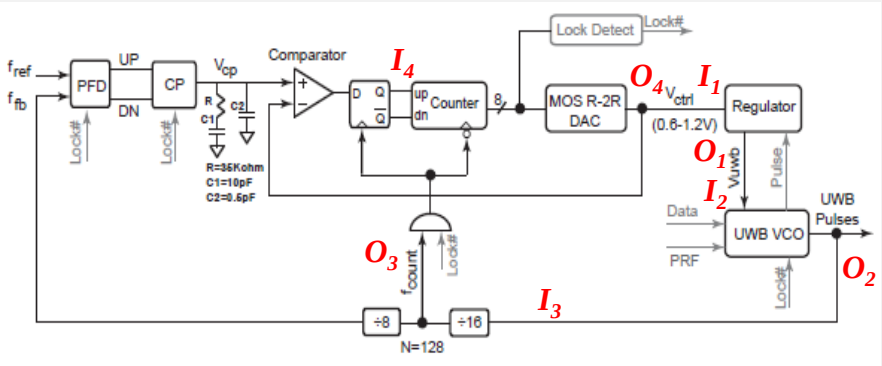
- **Mapping function f()**
  - SPICE simulation of a transistor netlist
- **Inputs to the simulation**
  - X: Input waveforms over a fixed period
  - C: Transistor size variations
- **Output from the function: Y – output waveforms**

Recall: Iterative Learning



- In each iteration, we will learn a model to predict the inputs likely to generate additional essential output waveforms

Example Result – UWB-PLL



- We will perform 4 sets of the experiments – each set is for each input-output pair

Initial Result

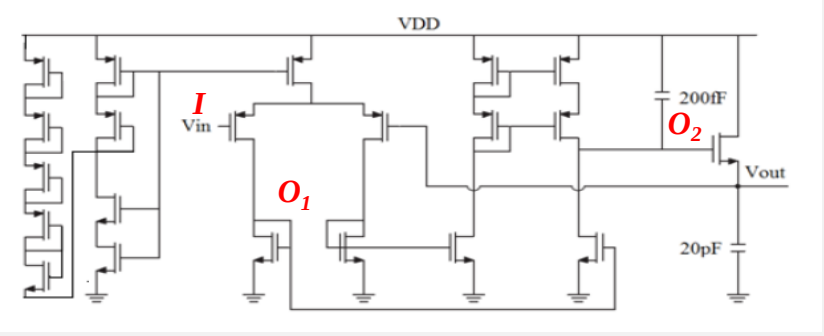
"IS's" – Number (#) of input waveforms selected and simulated

of "Essential Outputs (EO's) covered" Those deemed important for verification

In	Out	Apply learning			Random	
		# Iters	# IS's	# EO's	# IS's	# EO's
I_1	O_1	4	128	31	700	31
I_2	O_2	4	222	64	2200	64
I_3	O_3	5	270	56	750	56
I_4	O_4	30	1979	50	2800	50

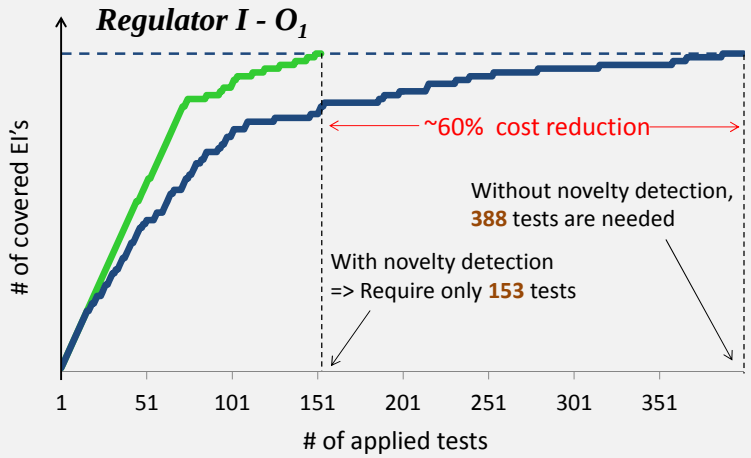
- Comparing to random input selection
- For each case, the # of essential outputs is shown
- Learning enables simulation of less # of inputs to obtain the same coverage of the essential outputs

Additional Result – Regulator



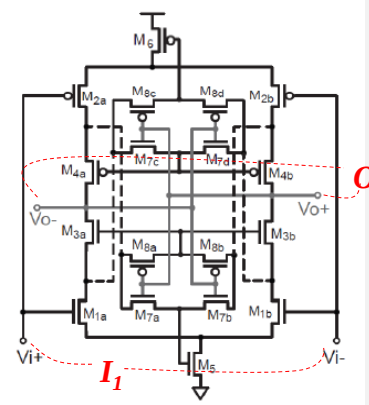
In	Out	Apply Learning		Random	
		# IS's	#EO's	# IS's	#EO's
I	O_1	153	84	388	84
I	O_2	107	49	355	49

Coverage Progress



Li-C. Wang Tutorial - April 24th, 2015 229

Additional Result – Low Power, Low Noise Amp.



In	Out	Apply Learning		Random	
		# IS's	#EO's	# IS's	#EO's
I_1	O_1	96	75	615	75

Li-C. Wang Tutorial - April 24th, 2015 230

Summary

- **Machine learning provides viable approaches for improving simulation-based verification**
- **In our approaches, learning is all about learning**
  - The features, or
  - The kernel function
- **Our learning framework is generic and can be applied to diverse verification contexts**
- **We continue to develop the learning framework and expand its applications**
  - RTL simulation (Low-power features, security features)
  - SPICE simulation (SoC verification, mixed-signal)
  - Post-silicon application (Debug)

Li-C. Wang Tutorial - April 24th, 2015 231

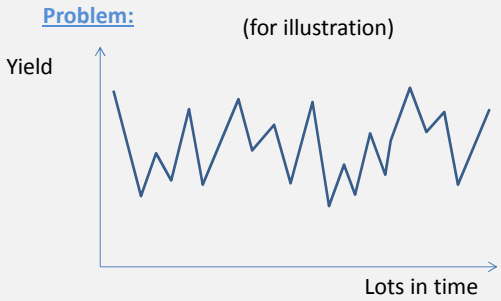
Applications in Post-Silicon Testing (40 minutes)

Li-C. Wang Tutorial - April 24th, 2015 232

The Beginning Of A Knowledge Discovery Task

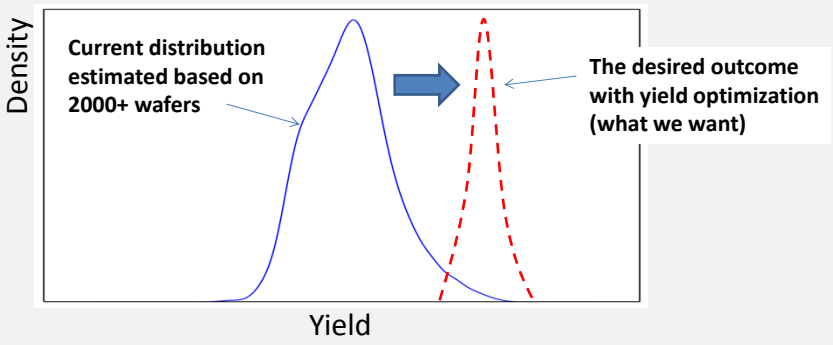
- **Yield scenario**
  - There is a yield fluctuation that sometime the yield drops significantly.
  - **Can you find the relevant process parameters that I can adjust to reduce this yield loss?**
- **Burn-In scenario**
  - Here are 30 chips that fail at the burn-in step.
  - **Can you find out if we can screen these fails with wafer probe tests?**
- **Customer return scenario**
  - Here are the 15 customer returns this year.
  - **Can you find test rules to screen any of them?**

Recall: Yield Improvement Application



- **ITC 2014: "Yield Optimization Using Advanced Statistical Correlation Methods"**
  - http://mtv.ece.ucsb.edu/licwang/PDF/2014-ITC.pdf
- **An automotive SoC product**
- **Yield fluctuated over time**
- **Went through one design revision, multiple test revisions, and process adjustment experiments**
  - Our result would provide "added value" to these efforts

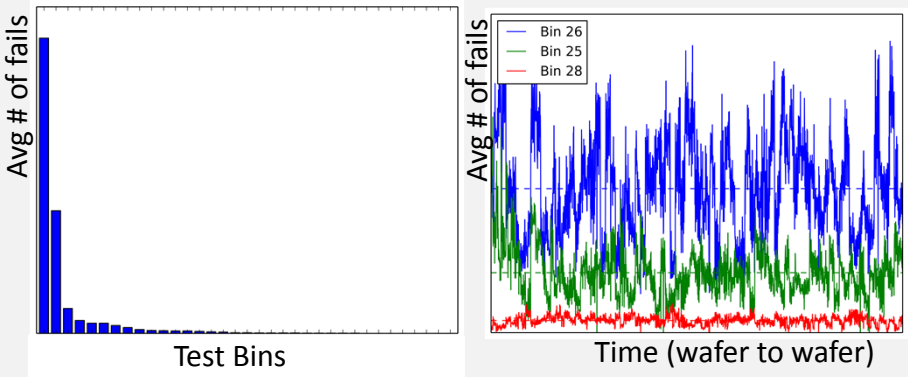
Objective: Discover Process Adjustments to Improve Yield



- **Mining goal: Discover strong correlations between a test fallout and a process parameter**
- **Hope for: Adjustment of the process parameter leads to improved yield and reduced yield fluctuation**

Li-C. Wang Tutorial - April 24th, 2015 235

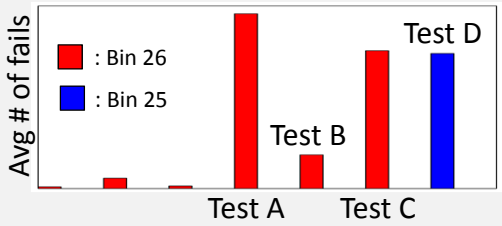
Identify The Test Bins To Analyze



- **Some bins had more fallouts than others**
- **It is natural to focus on the high-fallout test bins**

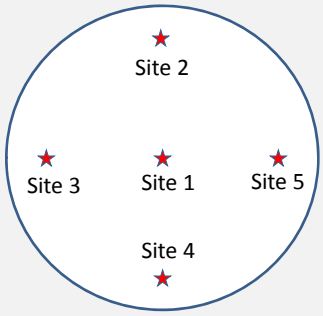
Li-C. Wang Tutorial - April 24th, 2015 236

Identify Specific Tests To Focus

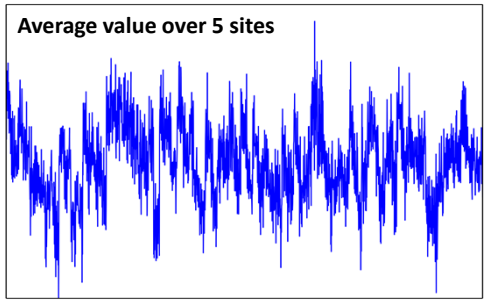


- **Some tests had more fallouts than others**
- **It is natural to focus on high-fallout tests first**

Correlation Targets



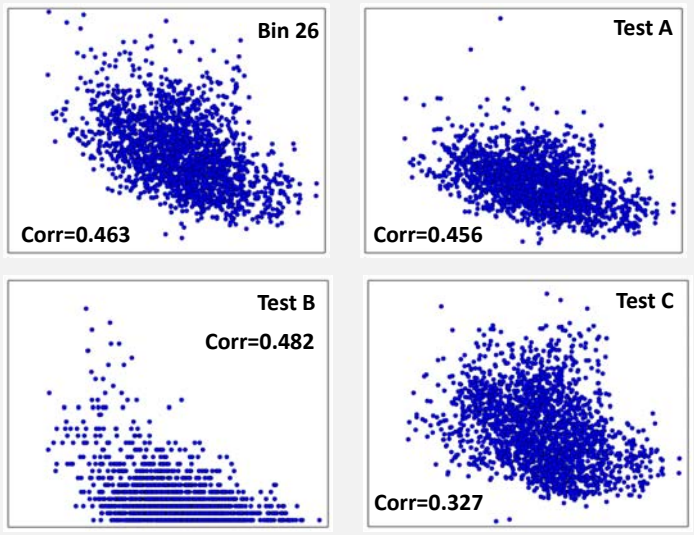
Site Locations



Example of fluctuation of measured values over time

- **More than 130 process parameters were measured**
- **Each was measured at 5 sites**
- **Naïve correlation method: Take the average of the five values**

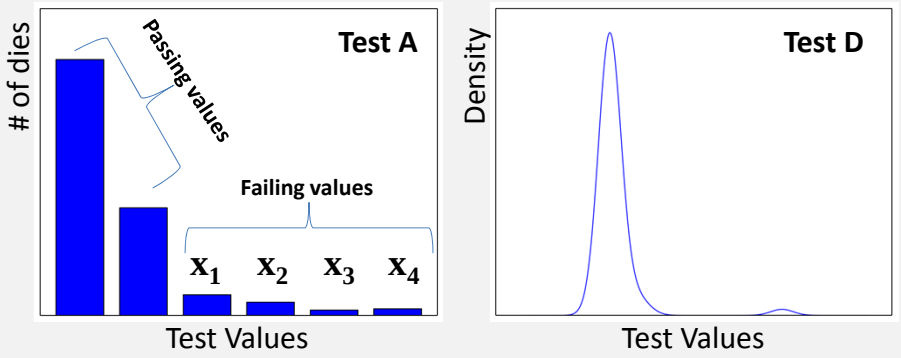
Starting Point: Result of Univariate Correlation



- **Process engineer had tried – no good result was found**

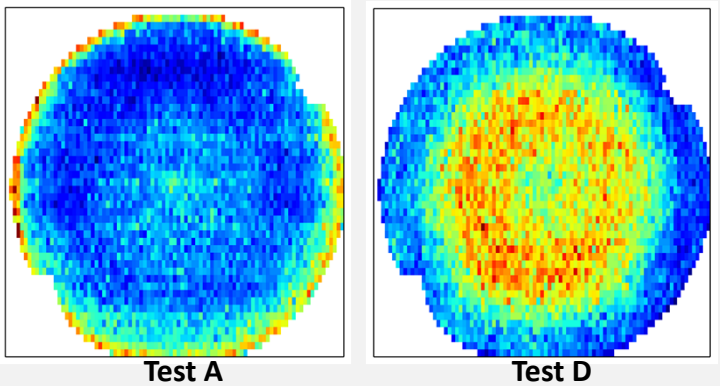
Need for Multivariate Correlation

- **Correlating to the number of fallouts with each test is not enough**
  - How do we know there is no correlation to other characteristics?



Discrete case **Continuous case**

Need for Multivariate Pattern Based Correlation



Test A Test D

- **Correlation may exist to some wafer pattern characteristics, and not to the number of fallouts from the entire patterns**

Our Correlation Methodology

- **Multivariate linear correlation analysis**
  - Uncover correlations not found by univariate correlation methods
- **Overcome the process shift issues**
  - Correlations themselves can fluctuate over time due to process shift
  - Formulate it as a subset correlation discovery problem
- **Once a process change is found – Perform risk evaluation**
  - Determine statistical independence between the process change and ***all other*** test fallouts
- **Detail: See SRC publication P069684 (also attached to this presentation)**

Multivariate Correlation

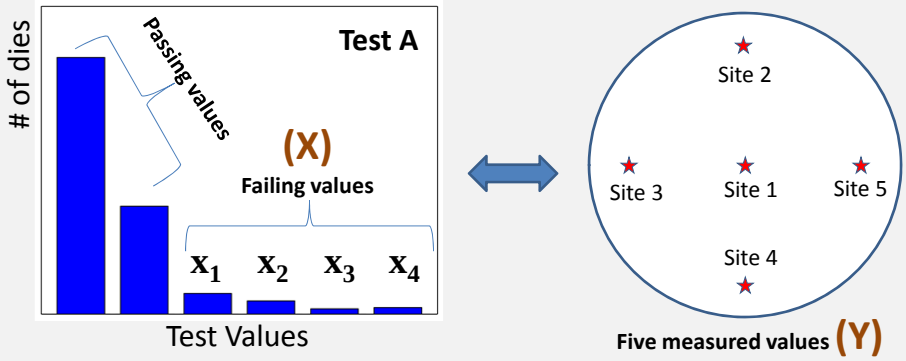
$$\begin{array}{c} \\ \vec{u}_1 \\ \vec{u}_2 \\ \vdots \\ \vec{u}_N \end{array}\!\!\begin{array}{c} \begin{array}{cccc} \vec{x}_1 & \vec{x}_2 & \cdots & \vec{x}_n \end{array} \\ \begin{vmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{N1} & x_{N2} & \cdots & x_{Nn} \end{vmatrix} \end{array} \times \begin{vmatrix} w_1^x \\ w_2^x \\ \vdots \\ w_n^x \end{vmatrix} \qquad \begin{array}{c} \\ \vec{v}_1 \\ \vec{v}_2 \\ \vdots \\ \vec{v}_N \end{array}\!\!\begin{array}{c} \begin{array}{cccc} \vec{y}_1 & \vec{y}_2 & \cdots & \vec{y}_m \end{array} \\ \begin{vmatrix} y_{11} & y_{12} & \cdots & y_{1m} \\ y_{21} & y_{22} & \cdots & y_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ y_{N1} & y_{N2} & \cdots & y_{Nm} \end{vmatrix} \end{array} \times \begin{vmatrix} w_1^y \\ w_2^y \\ \vdots \\ w_n^y \end{vmatrix}$$

$X \qquad W_x \qquad\qquad Y \qquad W_y$

$$CC(X,Y) = \max_{W_x, W_y} Corr(\langle W_x, X\rangle, \langle W_y, Y\rangle)$$

- **Linear case: Canonical Correlation Analysis**
- **Finding linear transform (*Wx*, *Wy*) to maximize the correlation**

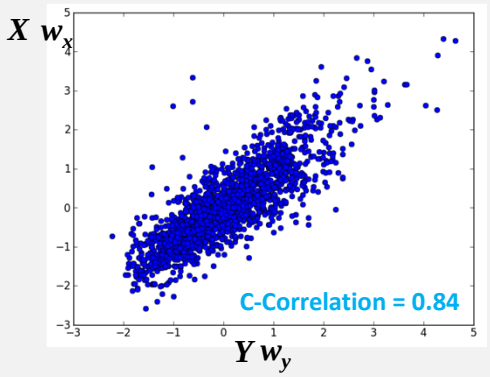
Applied To Test A Analysis

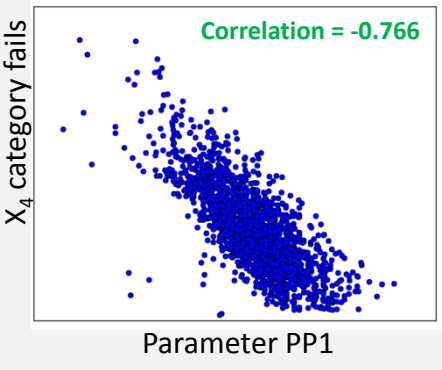


Discrete case

- **Applied multivariate linear correlation**
  - X: 4 dimensions
  - Y: 5 dimensions

Found 1 Good Result

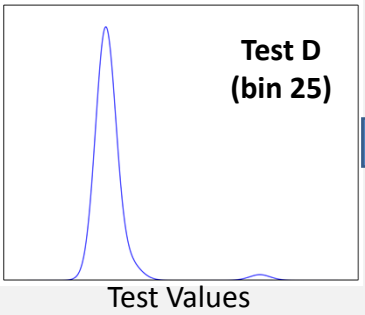




Analyze the left and lead to the result on the right

- **Process parameter PP1 is negatively correlated to category X4 fallout**

Applied to Test D Analysis



Encode distribution characteristics

$D_0 = \#\text{ of fails}$

$D_1 = \text{mean} = \mu$

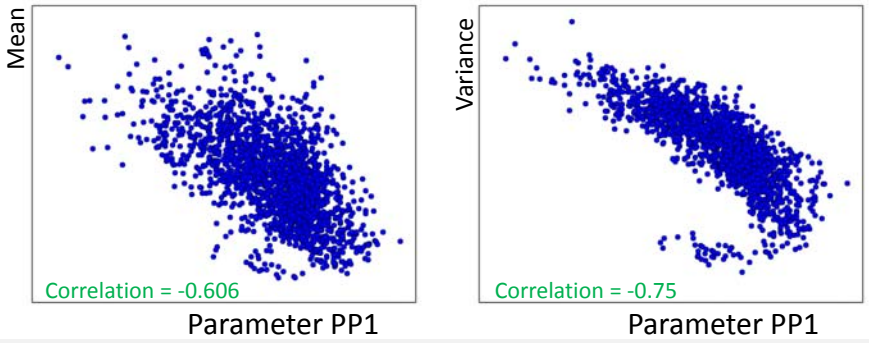
$D_2 = \text{variance} = E[(x-\mu)^2]$

$D_3 = \text{skew} = E[(x-\mu)^3]$

$D_4 = \text{kurtosis} = E[(x-\mu)^4] - 3[E[(x-\mu)^2]]^2$

- **Applied multivariate linear correlation**
  - X: 5 dimensions as shown above
  - Y: 5 dimensions as before (5 sites)

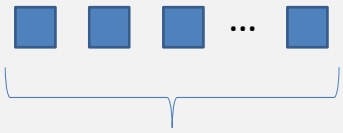
Additional Results



- **Process parameter PP1 is negatively correlated to the mean and variance of the Bin 25 fallouts**

Dealing With Process Shift

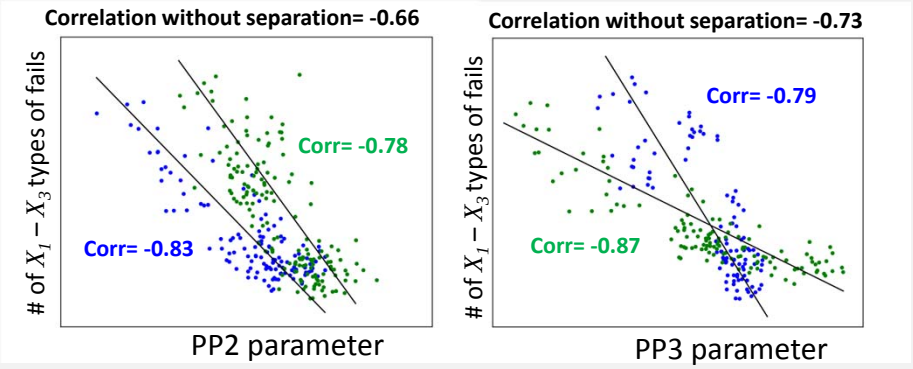
Test data from multiple lots manufactured at different time



Does a strong correlation exist on the subset of the lots but not on all?

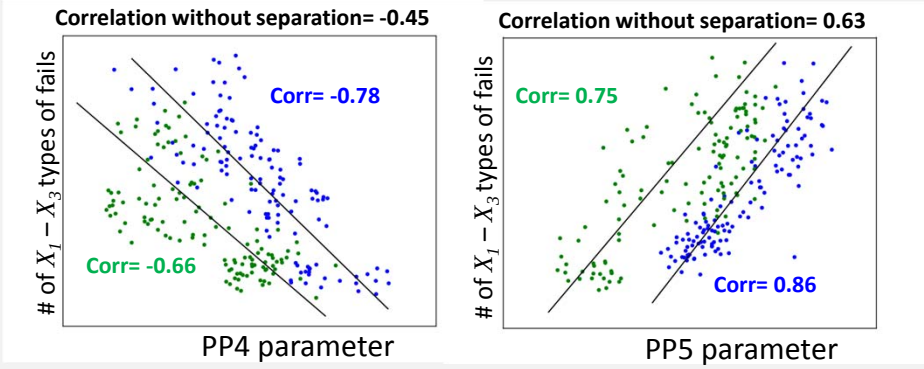
- **Formulate as a subset correlation discovery problem**
- **See publication P069684 for detail**

Two More Results



- **Fallouts are negatively correlated to parameters PP2 and PP3**

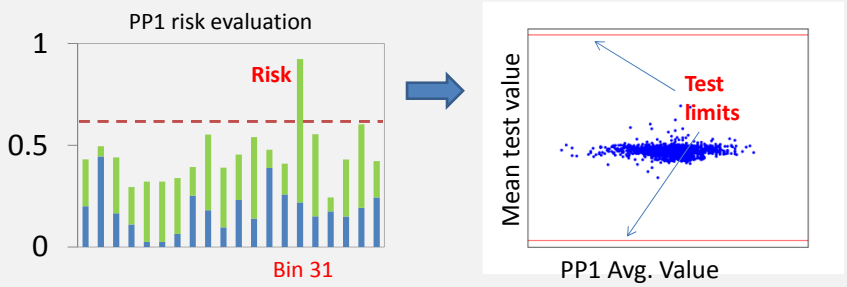
Two More Results



- **Fallouts are negatively correlated to parameter PP4, and**
- **Fallouts are positively correlated to parameter PP5**

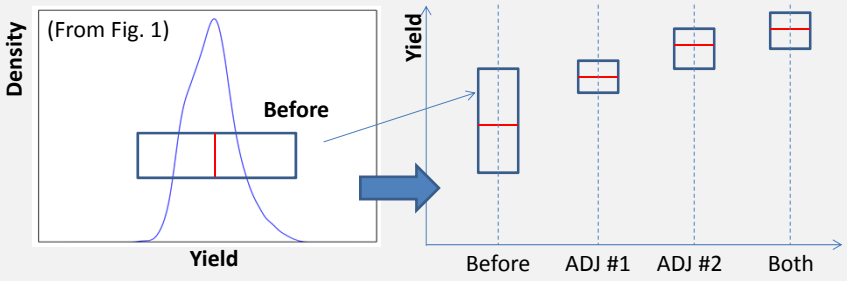
Risk Evaluation

- **Before silicon experiment, we had to ensure a recommended process change does not affect other test fallouts**
  - We don't want to solve one issue and cause another
- **Our method: ensuring statistical independence between a process change and each of the other test fallout**
  - Implementation is complicated – see report for detail



Risk inspection and containment

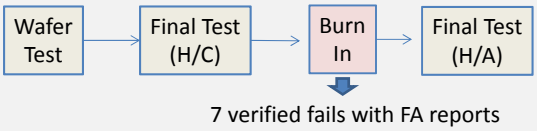
Silicon Result



- **Multiple lots with recommended process changes**
  - 3 process changes based on 5 parameters
  - 1 change applied to all experimental lots
- **Result: Significant yield improvement and reduction of fluctuation**

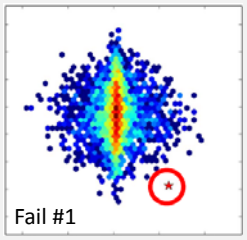
Another Example – Selective Burn-in Study

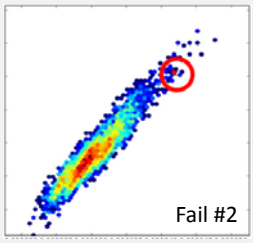
Case Study 1

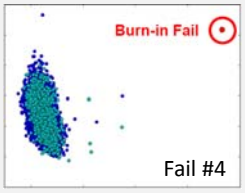


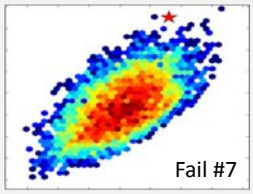
- **Product**
  - Analog and sensor product
- **Fail characteristics**
  - Fails were verified – we knew the root causes
- **Objective**
  - Identify **pre-burn-in** outlier models to capture them

Case Study 1 – 2-test models









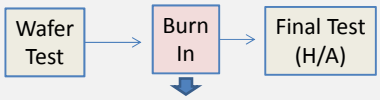
- **All 7 models were verified with test and design knowledge**
  - The tests used in the model were testing the failing site
- **The study confirms that there are pre-burn-in test signatures for burn-in fails**

Case Study 1 - Summary

Fail #	Fail Site	Outlier Model	Kill %
1	Gate Oxide	2-test model	1%
2	Polysilicon	2-test model	0.44%
3	Gate Oxide	Univeriate	1.28%
4	Polysilicon	2-test model	0.01%
5	Gate Oxide	Univariate	0.23%
6	Gate Oxide	Univariate	0.48%
7	Gate Oxide	2-test model	1.24%
Accumulated Kill %			4.54%

- **To capture all 7 fails, only < 5% of the population require burn-in**

Case Study 2



Category	Type	# of fails
A	1	8
	2	3
	3	9
	4	2
B	1	15
C	1	11
Total		48

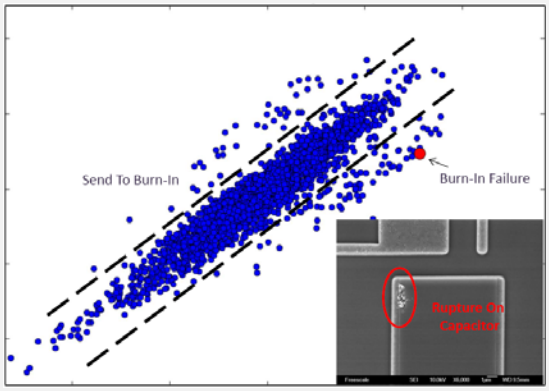
- **Product**
  - Automotive microcontroller
- **Fail characteristics**
  - Some fails had FA reports, but most did not
- **Objective**
  - Identify outlier models to capture them before burn-in

Case Study 2

Category	Type	# of fails	Model	# of fails screened by the model	Kill %	Kill % to include the next fail
A	1	8	2-test model X	6	7.3%	24.9%
	2	3	2-test model X	2	6.7%	16.8%
	3	9	2-test model X	5	7.8%	11.4%
	4	2	2-test model X	2	6.9%	-
B	1	15	Univariate	8	4.1%	9%
C	1	11	Univariate	7	7.5%	27.9%
Total		48	3 models	30 fails		

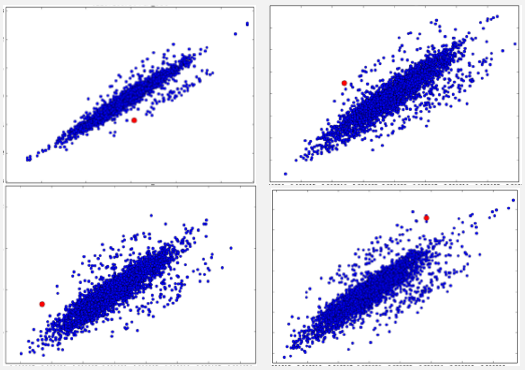
- **Models are not selected based on only outlying properties**
  - The tests in use are testing the block containing the fails
  - The model is shared by most of the fails
- **Difficulties**
  - Even the fails are given with the same type – not necessarily the case
  - Not sure if all 48 fails were true burn-in fails
  - If we couldn't find a good screen, what did we tell the product team – need supporting evidence to show that the part was **not screenable** with the tests

Case Study 2

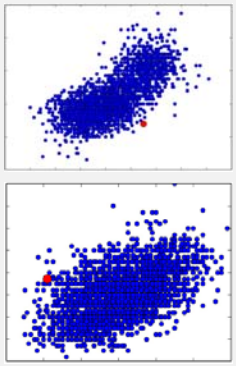


- **New tests became available later**
  - Allow better models to be built on

Case Study 2 – Additional Models



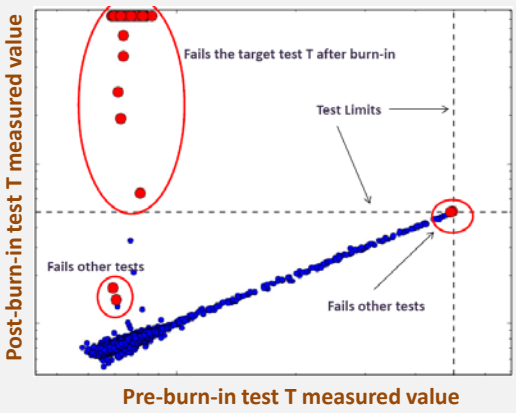
Same model from the previous slide project many other fails as outliers



Another model projects multiple fails as outliers

- **As new tests became available, more models could be found**
  - This was not possible with the previous test data

Case Study 2 - Challenge



- **Consider a target test T with many post-burn-in fails**
  - Observe significant change of T measured value pre- and post- burn-in
- **Only 2 fails were marginal in both pre- and post- burn-in stage**
- **Without the stress, how can one screen those fails?**
  - We need alternative stress

Summary

- **Carried out studies on multiple products over multiple technology nodes**
- **Positive result**
  - Many burn-in fails did expose signatures in pre-burn-in tests
- **Main challenge**
  - Some fails require development of additional stress tests
- **Test data mining alone (without adding new tests) cannot solve the problem entirely**

Another Example – Customer Return Analysis

Field Failure Analysis Problem

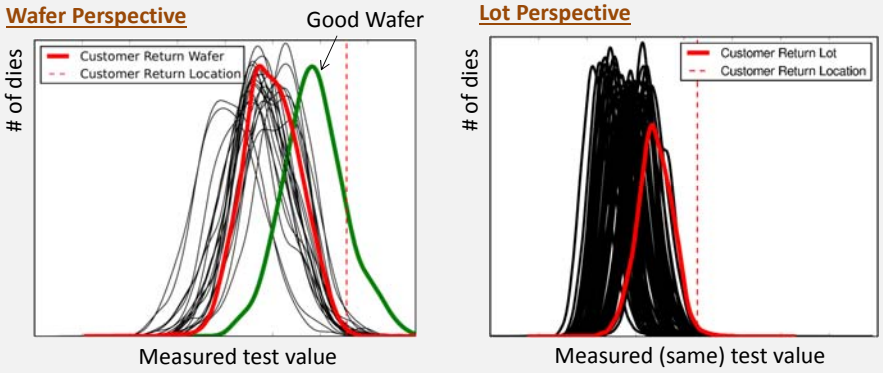
- **Spatial aspect for the search**
  - Does an abnormality appear on the die, the wafer, or the lot?
- **Test aspect for the search**
  - Is the abnormality exposed based on one test or more tests?
- **Process/design aspect for the search**
  - Is the abnormality process dependent or design dependent?

Two References

- Nik Sumikawa, et al. "Screening Customer Returns With Multivariate Test Analysis" International Test Conference 2012
  - http://mtv.ece.ucsb.edu/licwang/PDF/2012-ITCa.pdf
- Nik Sumikawa, et al. "A Pattern Mining Framework for Inter-Wafer Abnormality Analysis" International Test Conference 2013
  - http://mtv.ece.ucsb.edu/licwang/PDF/2013-ITC.pdf

Li-C. Wang Tutorial - April 24th, 2015 265

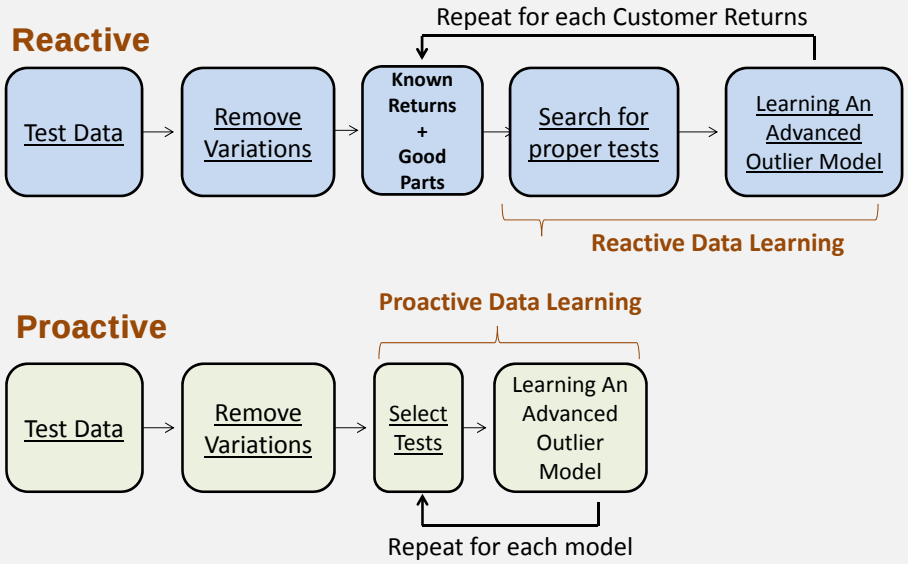
Abnormality is a "Relative" Term



- Abnormality is relative to the population of normality
- Search for abnormality to explain the fail
  - Search for a proper population of normality
  - Search for a space to model the normality
  - Difficult to exhaust the search manually – demand a data mining tool

Li-C. Wang Tutorial - April 24th, 2015 266

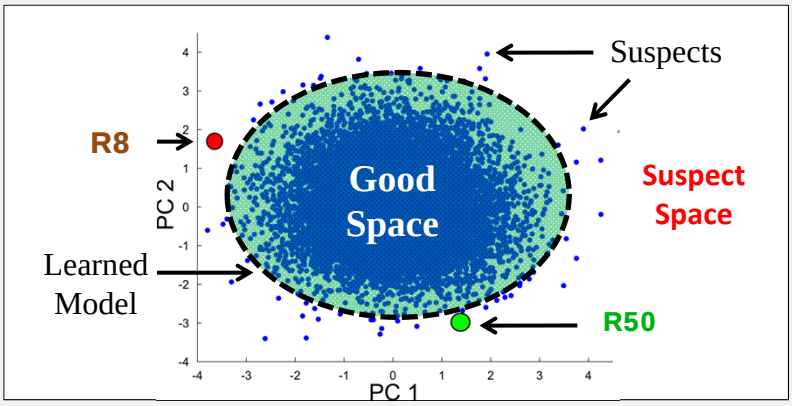
Analysis Methodologies



Case Study: Automotive SOC

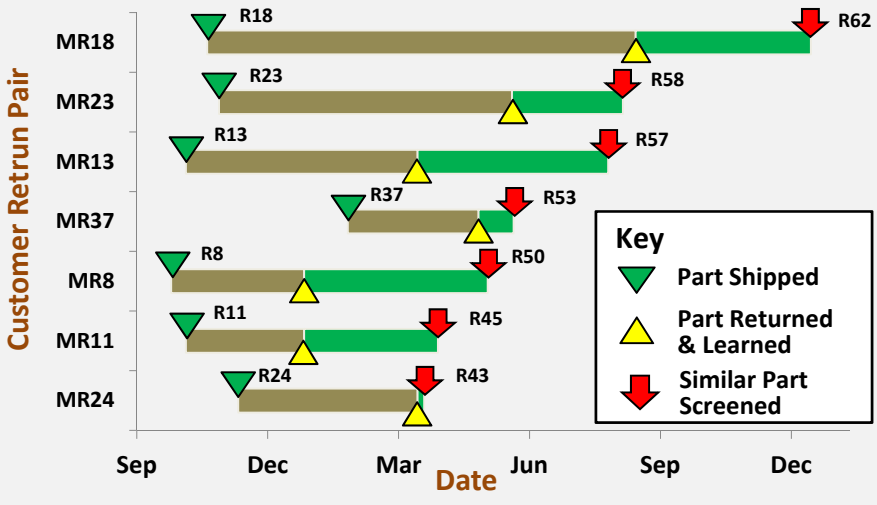
- **High Quality SoC for automotive market**
  - Containing flash and analog components
- **Dataset**
  - 62 customer returns belonging to 50 lots
  - >12K good parts per lot
  - Data collected over an 18-month period
- **1000+ parametric wafer level tests**
  - Belonging to 3 wafer level insertions

Reactive Example: Learn R8 -> Screen R50



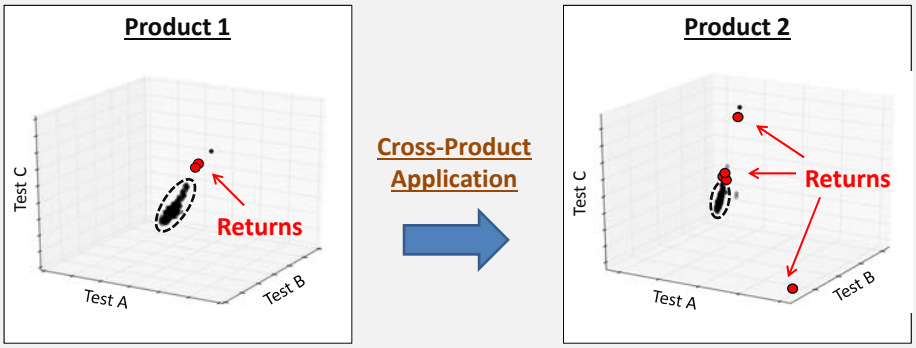
- R8 was an earlier return
- This example shows we could learn from R8 and screen out a later return R50

Results – Timeline View



- Timeline shows that it is possible to learn and prevent future customer returns in a production setting

Cross Product Application



- Products 1 and 2 belong to the same family & technology
- An outlier model was learned for two returns from Product 1
- The same outlier model could prevent five returns when applied to Product 2

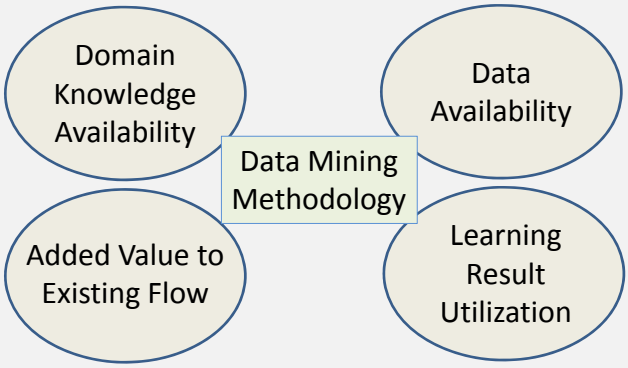
Li-C. Wang Tutorial - April 24th, 2015 271

Final Remark and Questions

(10 Minutes)

Li-C. Wang Tutorial - April 24th, 2015 272

Recall: Four Key Considerations



- **In this picture, I do not mention a "learning algorithm"**
- **Because to implement a successful practical data mining methodology, it involves integration of a collection of "simple" algorithms**
  - Individually simple, but collectively very complex

Recall: General Principles

- **Infrastructure support - Make sure the data is available, or the cost to collect the data is acceptable**
- **Solve the right problem - Develop a methodology to incorporate as much domain knowledge as possible**
- **Involve human decision – Human involvement in the flow relaxes the requirement of guaranteed result**
- **Complementary - The approach has to provide added value to the existing flow**

Conclusion

- Data mining can be used for two purposes
  - As a decision function used by another tool
  - As interpreted by a person (Knowledge Discovery, PPT generation)
- Learn = Data + Knowledge
  - In our applications, domain knowledge is important
  - Learning is about learning the knowledge into (1) Kernel (2) Features
- Successful data mining implementation
  - (1) Data availability (2) Knowledge availability (3) Tackle the right problem (4) Provide added value to existing flow
- Data mining application in EDA and test
  - Promises in many areas – many to be explored
  - Many encouraging results shown in industrial settings

The End

THANK YOU!