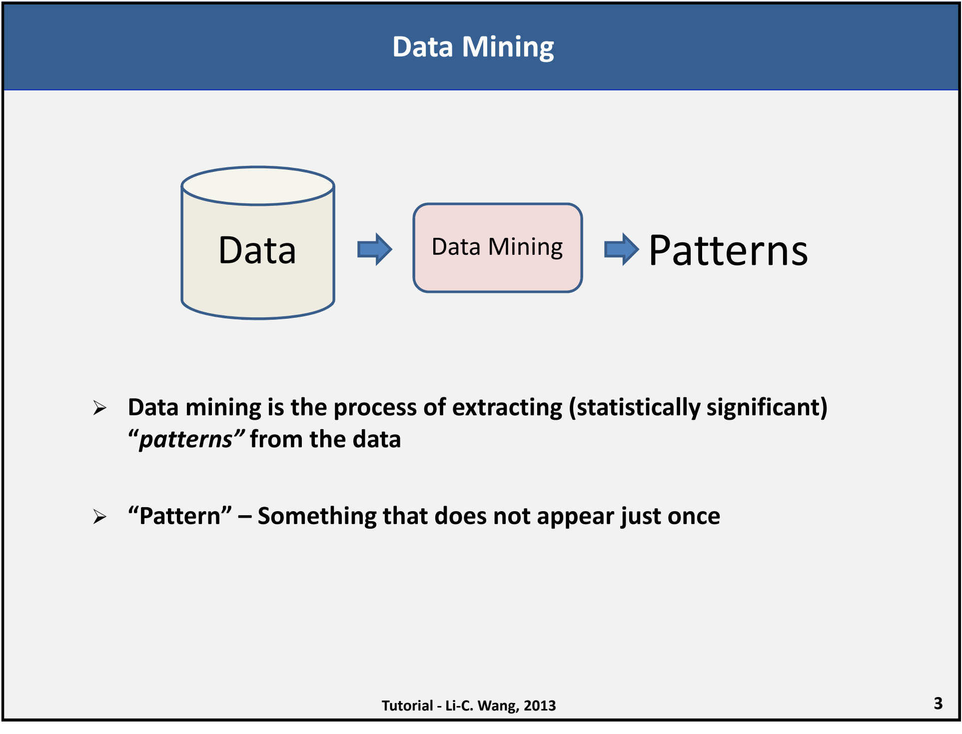
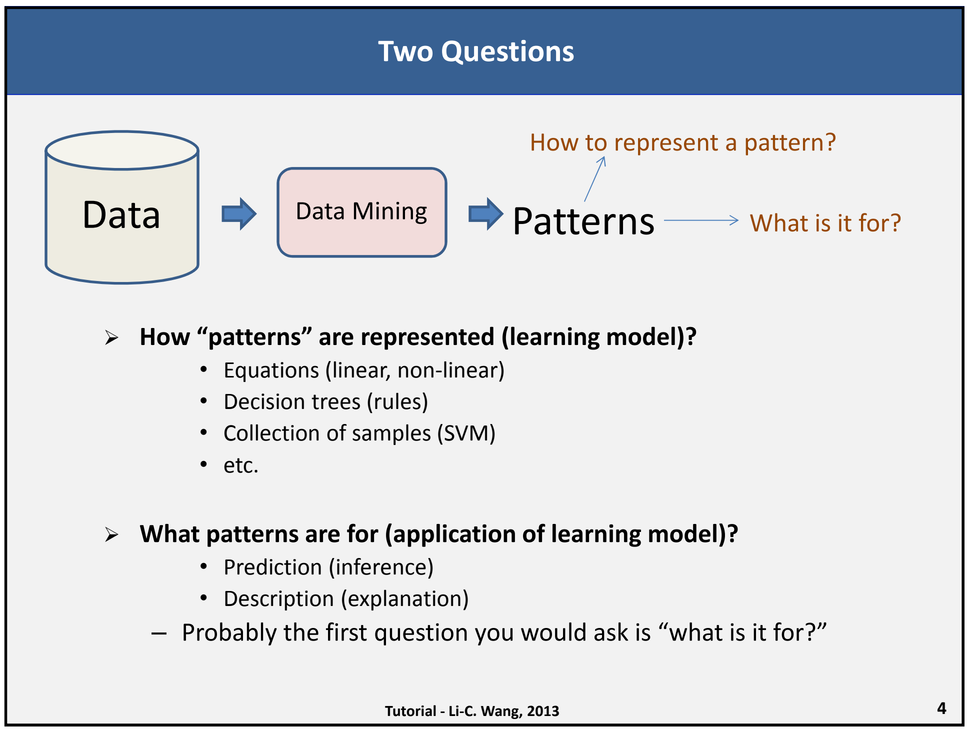
Data Mining In Design & Test
- Principles and Practices

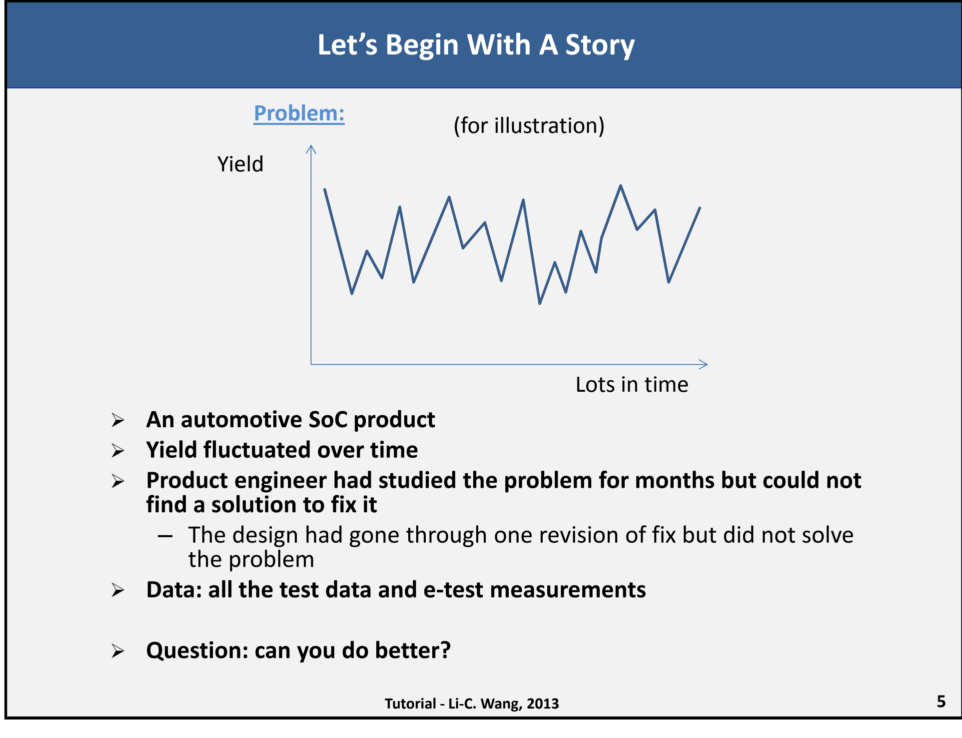
Li-C. Wang, UC Santa Barbara

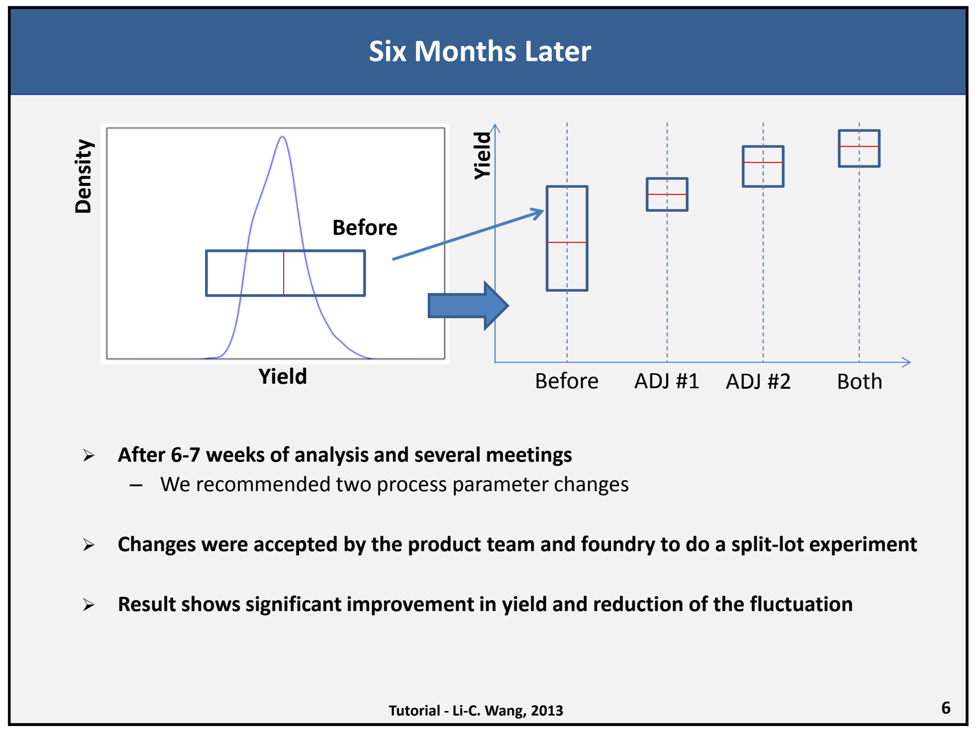
Preface
(10 minutes)

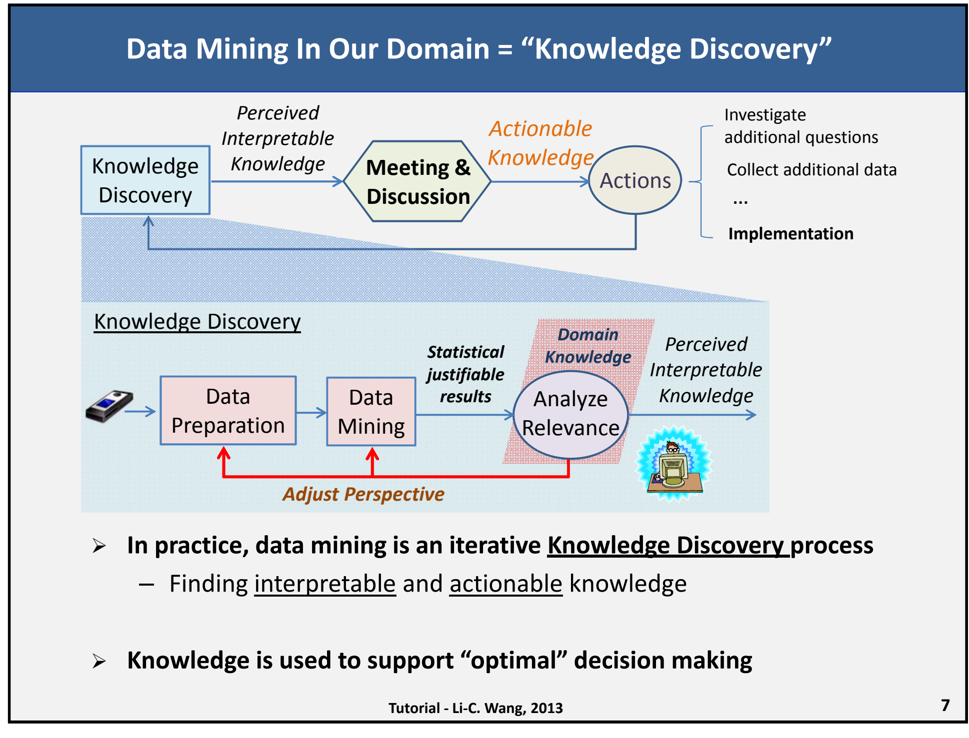
The "Data Mining" discussed in this tutorial
Historical view of the works included
What to be expected

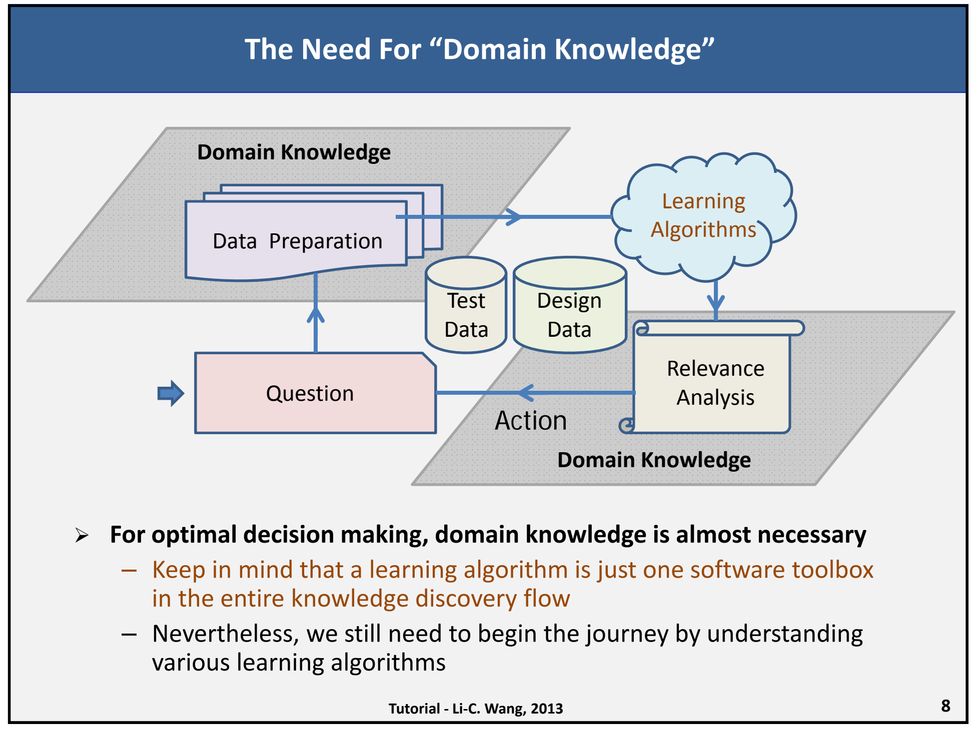

Data Mining
Data
Data Mining
Patterns
Data mining is the process of extracting (statistically significant) "patterns" from the data
"Pattern" – Something that does not appear just once
Tutorial - Li-C. Wang, 2013
3

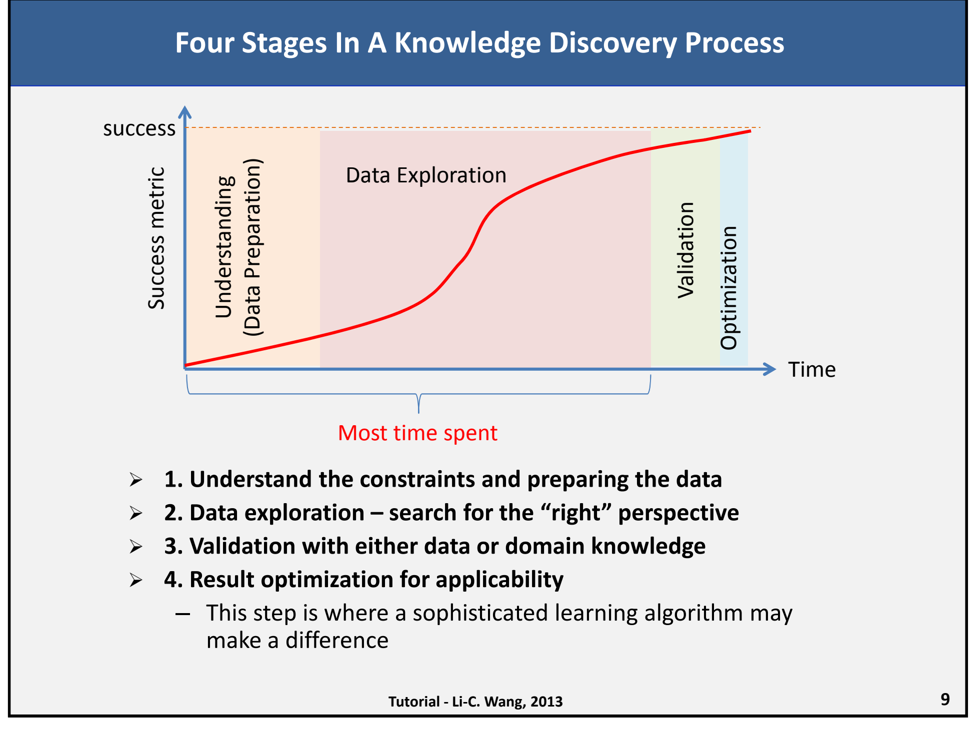

Two Questions
Data
Data Mining
Patterns
How to represent a pattern?
What is it for?
How "patterns" are represented (learning model)?
Equations (linear, non-linear)
Decision trees (rules)
Collection of samples (SVM)
etc.
What patterns are for (application of learning model)?
Prediction (inference)
Description (explanation)
Probably the first question you would ask is "what is it for?"
Tutorial - Li-C. Wang, 2013
4


Let's Begin With A Story
Problem:
(for illustration)
Yield
Lots in time
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?
Tutorial - Li-C. Wang, 2013
5


Six Months Later
Density
Before
Yield
Yield
Before
ADJ #1
ADJ #2
Both
After 6-7 weeks of analysis and several meetings
– We recommended two process parameter changes
Changes were accepted by the product team and foundry to do a split-lot experiment
Result shows significant improvement in yield and reduction of the fluctuation
Tutorial - Li-C. Wang, 2013
6


Data Mining In Our Domain = "Knowledge Discovery"
Knowledge Discovery
Perceived Interpretable Knowledge
Meeting & Discussion
Actionable Knowledge
Actions
Investigate additional questions
Collect additional data
...
Implementation
Knowledge Discovery
Data Preparation
Data Mining
Statistical justifiable results
Domain Knowledge
Analyze Relevance
Perceived Interpretable Knowledge
Adjust Perspective
In practice, data mining is an iterative Knowledge Discovery process
– Finding interpretable and actionable knowledge
Knowledge is used to support "optimal" decision making
Tutorial - Li-C. Wang, 2013
7

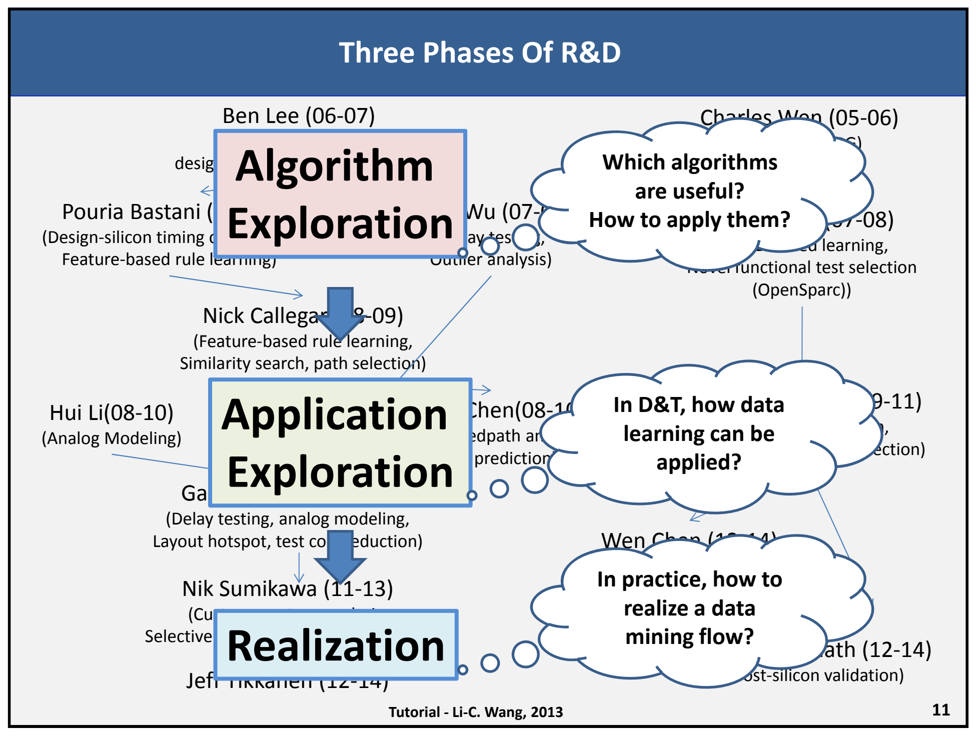

The Need For "Domain Knowledge"
Domain Knowledge
Data Preparation
Learning Algorithms
Test Data
Design Data
Question
Relevance Analysis
Action
Domain Knowledge
For optimal decision making, domain knowledge is almost necessary
– Keep in mind that a learning algorithm is just one software toolbox in the entire knowledge discovery flow
– Nevertheless, we still need to begin the journey by understanding various learning algorithms
Tutorial - Li-C. Wang, 2013
8



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
- **PhD Students (2006 – current)**
  - Ben Lee (Startup) - 2006
  - Charles Wen (NCTU, Taiwan)
  - Pouria Bastani (Intel)
  - Onur Guzey (Intel -> MIT)
  - Sean Wu (TPK, Taiwan)
  - Nick Callegari (nVidia)
  - Hui Lee (Intel)
  - Janine Chen (AMD)
  - Po-Hsien Chang (Oracle)
  - Gagi Drmanac (Intel)
  - Nik Sumikawa (Freescale) - 2013

  - Jeff Tikkanen (TBD)
  - Wen Chen (TBD)
  - Vinayak Kamath (TBD)



Plan For The Tutorial (6 hours = 360 minutes)

- **Opening (20 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 (60 minutes)**
- **Application examples, working principals and findings (60 minutes)**
- **Knowledge discovery – Application examples in Tests (60 minutes)**
- **Knowledge discovery – Application examples in Verification (30 minutes)**
- **Final Remark and Questions (10 minutes)**

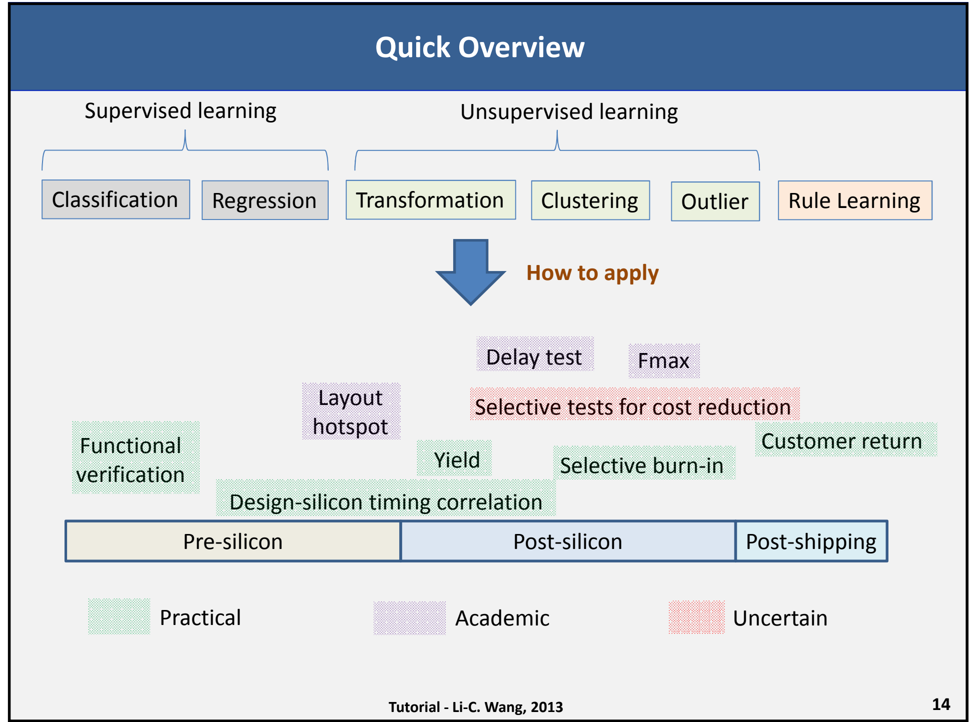
Tutorial - Li-C. Wang, 2013 12

Plan For The Short-Version Tutorial (2.5 hours = 150 minutes)

- **Opening (10 minutes)**
- **An introduction to data mining in design and test (120->60 minutes)**
  - Basic learning concepts and intuitions to algorithms
  - Example problem formulations and application contexts
- **Learning theory, SVM and Kernel Method (60->15 minutes)**
- **Application examples, working principals and findings (60->40 minutes)**
- **Knowledge discovery – Application examples in Tests (60->30 minutes)**
- **Knowledge discovery – Application examples in Verification (if have time)**
- **Final Remark and Questions (10->5 minutes)**

Tutorial - Li-C. Wang, 2013 13

Quick Overview



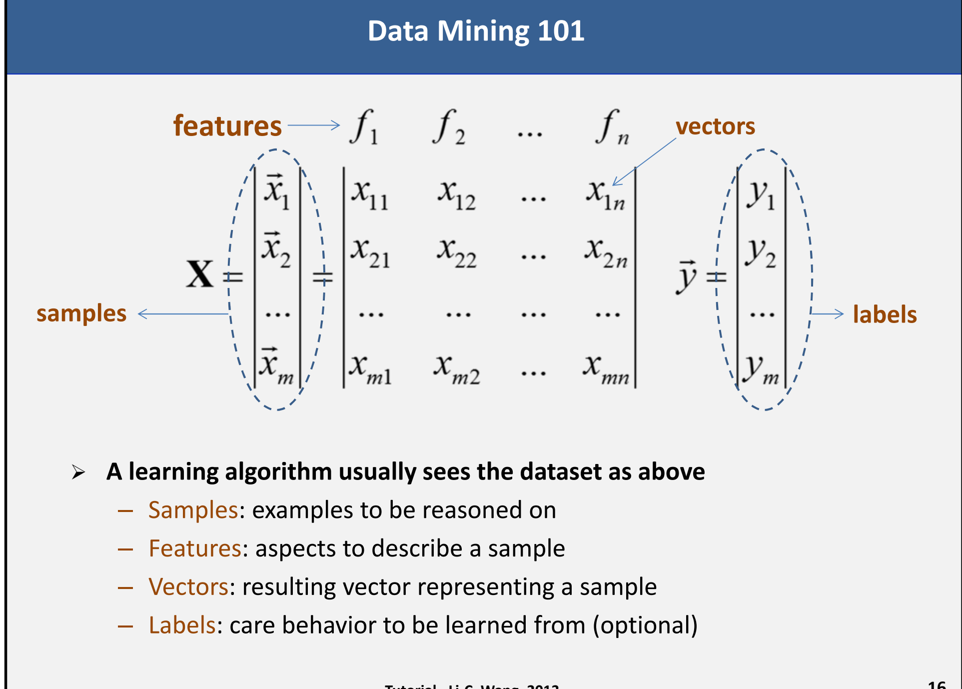
Tutorial - Li-C. Wang, 2013 14

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

(120->60 minutes)

Tutorial - Li-C. Wang, 2013

15



Data Mining Approaches

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

Tutorial - Li-C. Wang, 2013 17

Data Mining Approaches

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

Tutorial - Li-C. Wang, 2013 18

Data Mining 101 – Supervised Learning - Classification

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

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

Class labels

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

Tutorial - Li-C. Wang, 2013 19

Example Learning Algorithms For Classification

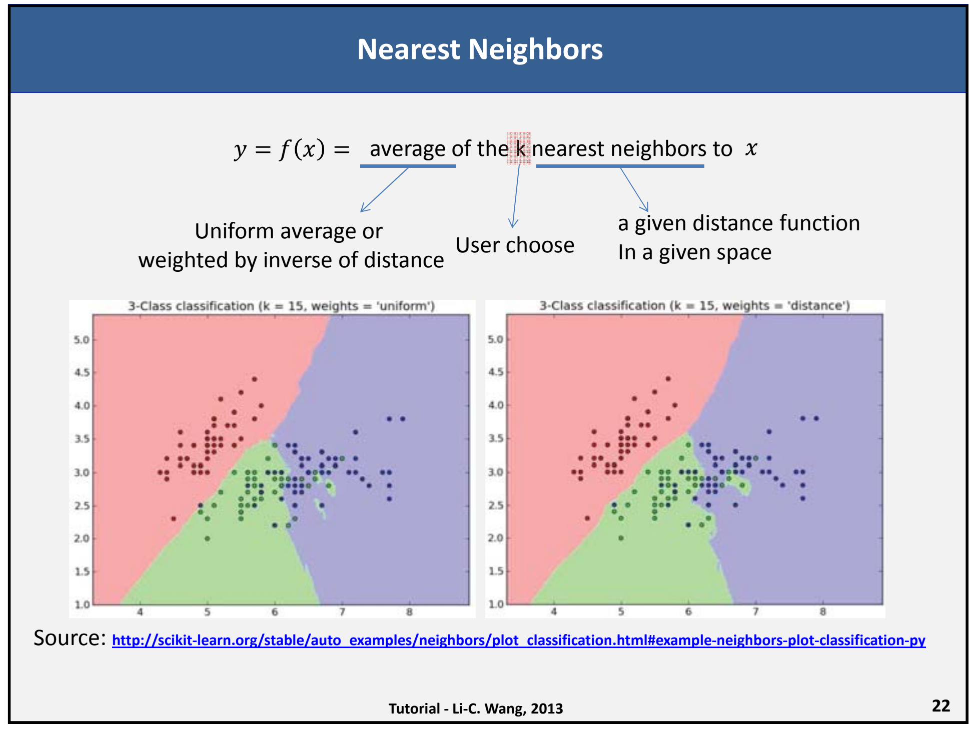
- **Nearest Neighbors**
- **Linear Discriminant Analysis (LDA)**
  - Quadratic Discriminant Analysis (QDA)
- **Naïve Bayes**
- **Decision Tree**
  - Random Forest
- **Support Vector Machine**
  - Linear
  - Radius Based Function (RBF)

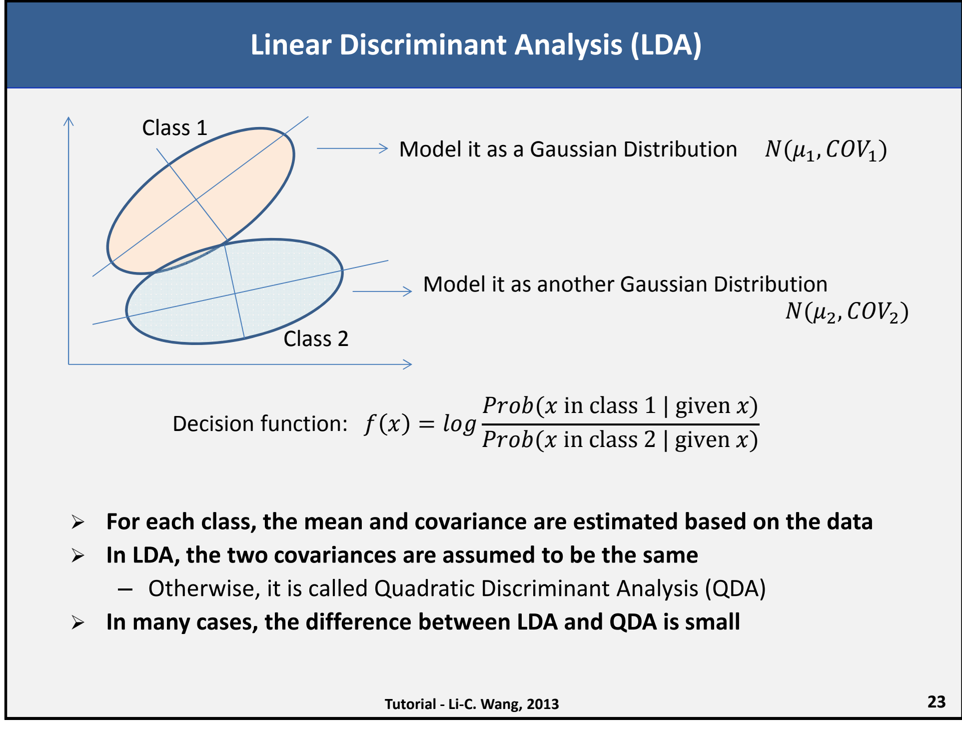
Tutorial - Li-C. Wang, 2013 20

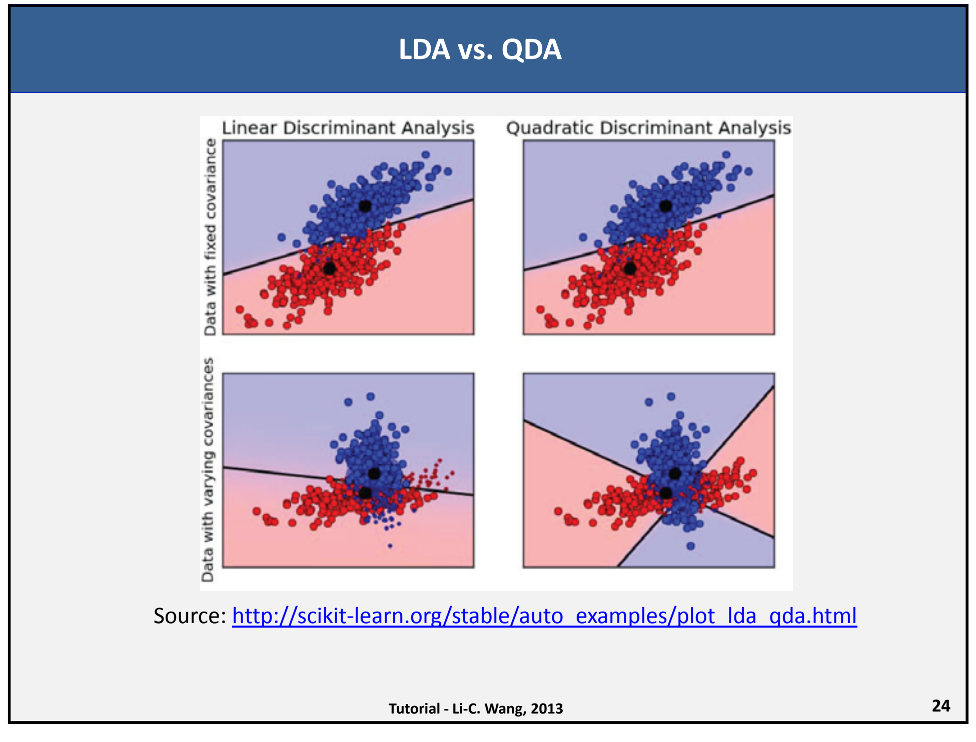
Example Learning Algorithms For Classification

- **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)

Tutorial - Li-C. Wang, 2013 21




Linear Discriminant Analysis (LDA)
Class 1
Model it as a Gaussian Distribution $N(\mu_1, COV_1)$
Model it as another Gaussian Distribution $N(\mu_2, COV_2)$
Class 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
Tutorial - Li-C. Wang, 2013
23


LDA vs. QDA
Linear Discriminant Analysis
Quadratic Discriminant Analysis
Data with fixed covariance
Data with varying covariances
Source: http://scikit-learn.org/stable/auto_examples/plot_lda_qda.html
Tutorial - Li-C. Wang, 2013
24

Bayesian Inference – Naïve Bayes Classifier

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

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

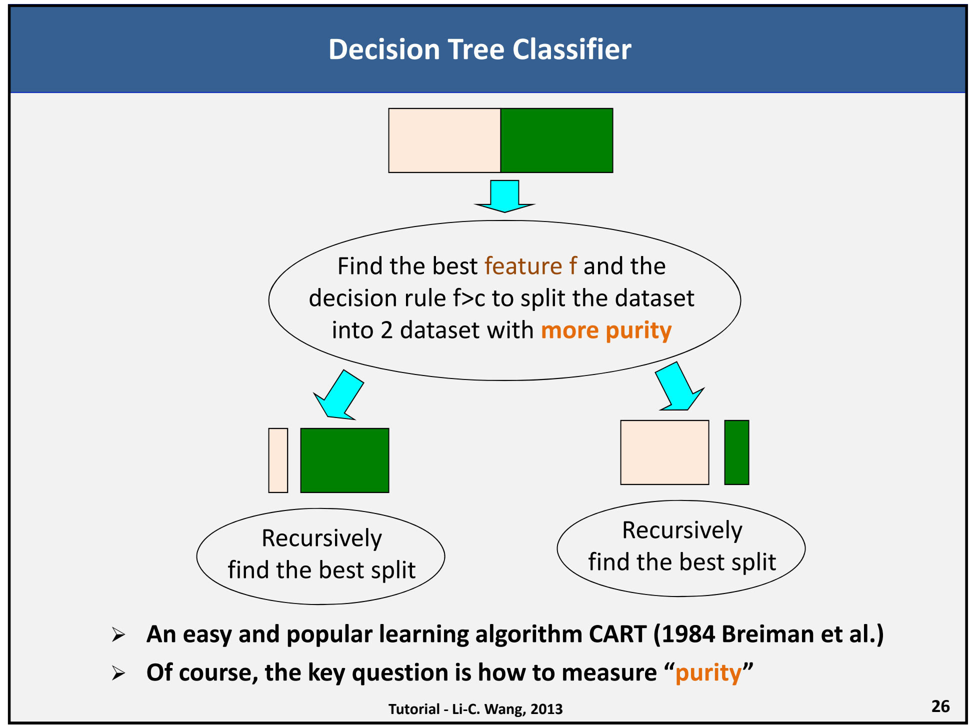
Independent assumptions

- **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**

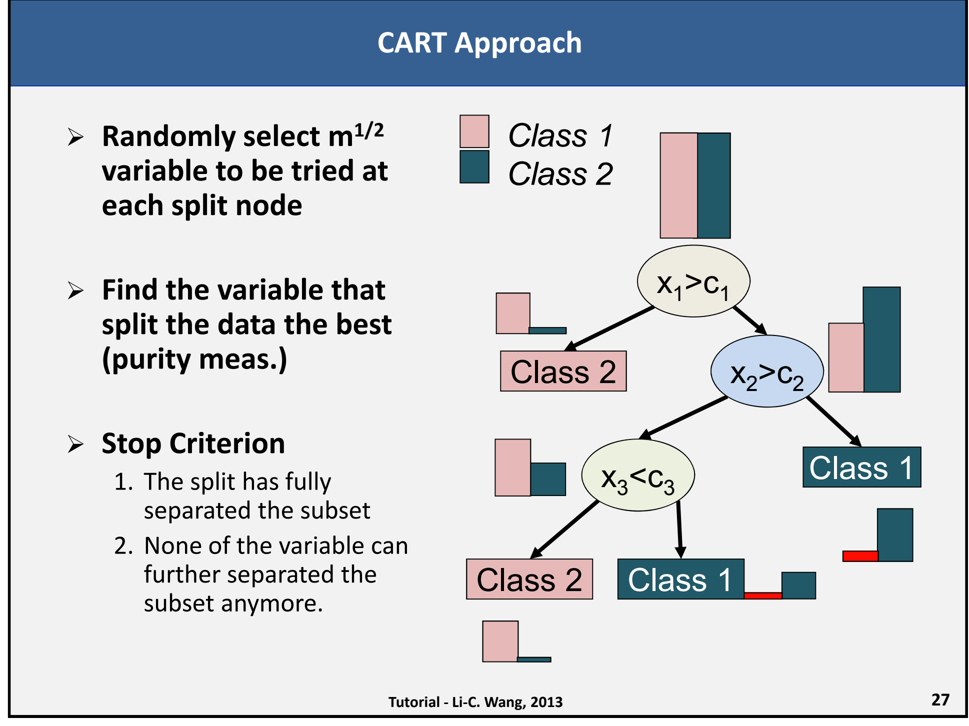
Tutorial - Li-C. Wang, 2013 25

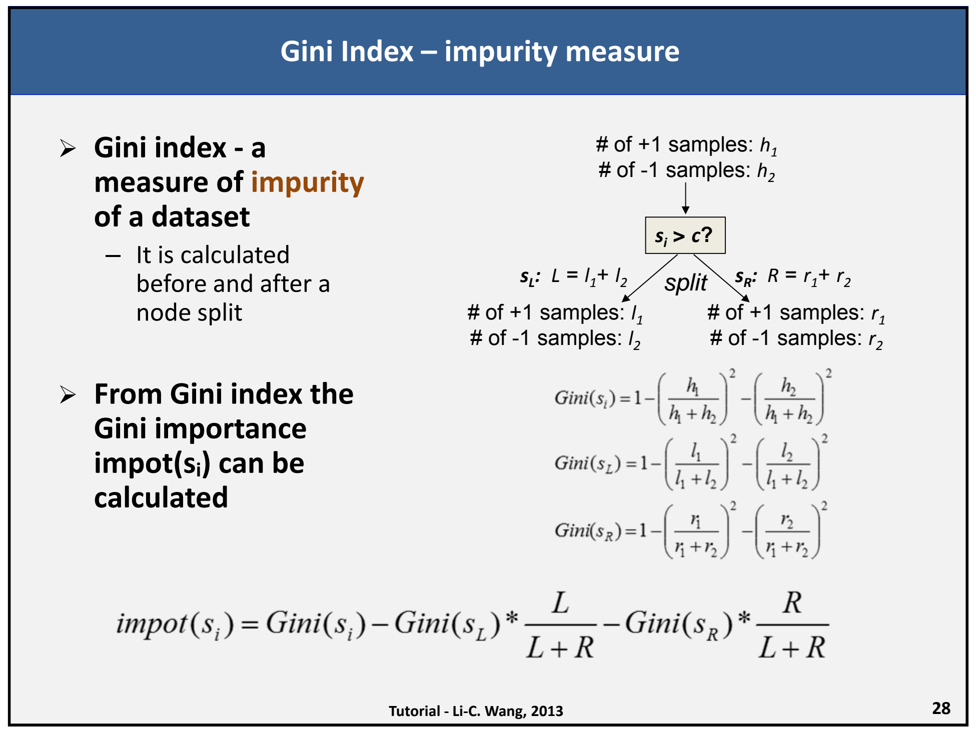
Decision Tree Classifier



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

Tutorial - Li-C. Wang, 2013 26

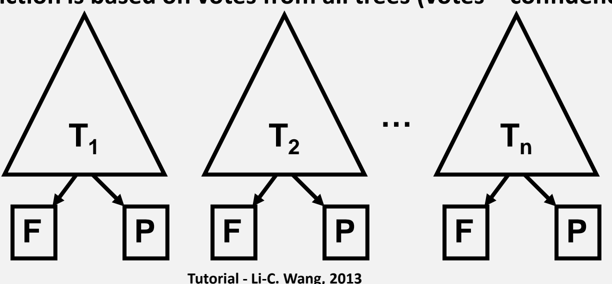

CART Approach
Randomly select m^1/2 variable to be tried at each split node
Find the variable that split the data the best (purity meas.)
Stop Criterion
1. The split has fully separated the subset
2. None of the variable can further separated the subset anymore.
Class 1
Class 2
x1>c1
Class 2
x2>c2
x3<c3
Class 1
Class 2
Class 1
Tutorial - Li-C. Wang, 2013
27


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(si) can be calculated
of +1 samples: h1
of -1 samples: h2
si > c?
sL: L = l1+ l2
split
sR: R = r1+ r2
of +1 samples: l1
of -1 samples: l2
of +1 samples: r1
of -1 samples: r2
$Gini(s_i) = 1 - \left(\frac{h_1}{h_1+h_2}\right)^2 - \left(\frac{h_2}{h_1+h_2}\right)^2$
$Gini(s_L) = 1 - \left(\frac{l_1}{l_1+l_2}\right)^2 - \left(\frac{l_2}{l_1+l_2}\right)^2$
$Gini(s_R) = 1 - \left(\frac{r_1}{r_1+r_2}\right)^2 - \left(\frac{r_2}{r_1+r_2}\right)^2$
$impot(s_i) = Gini(s_i) - Gini(s_L) * \frac{L}{L+R} - Gini(s_R) * \frac{R}{L+R}$
Tutorial - Li-C. Wang, 2013
28

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

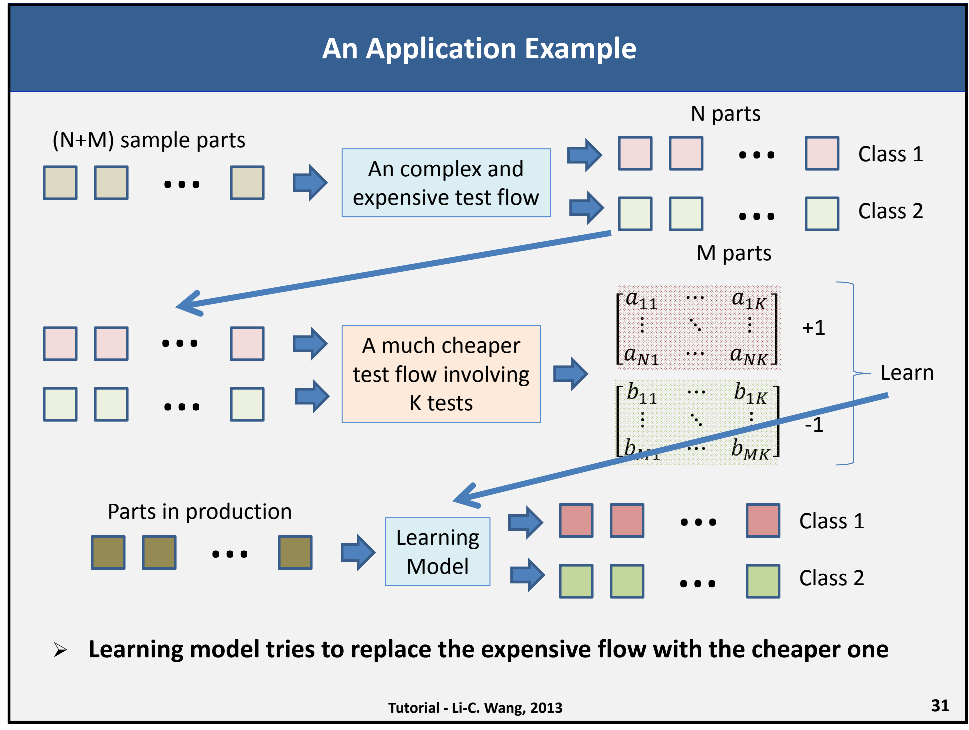
- **Select a random set of (training) samples (2/3 subset)**
- **Grow a tree based on only the selected samples (in-bag data)**
- **Use the unselected samples (out-of-bag data) to validate the tree performance, i.e. prediction accuracy**
- **Grow many trees until the average accuracy saturates**
- **The prediction is based on votes from all trees (votes = confidence)**

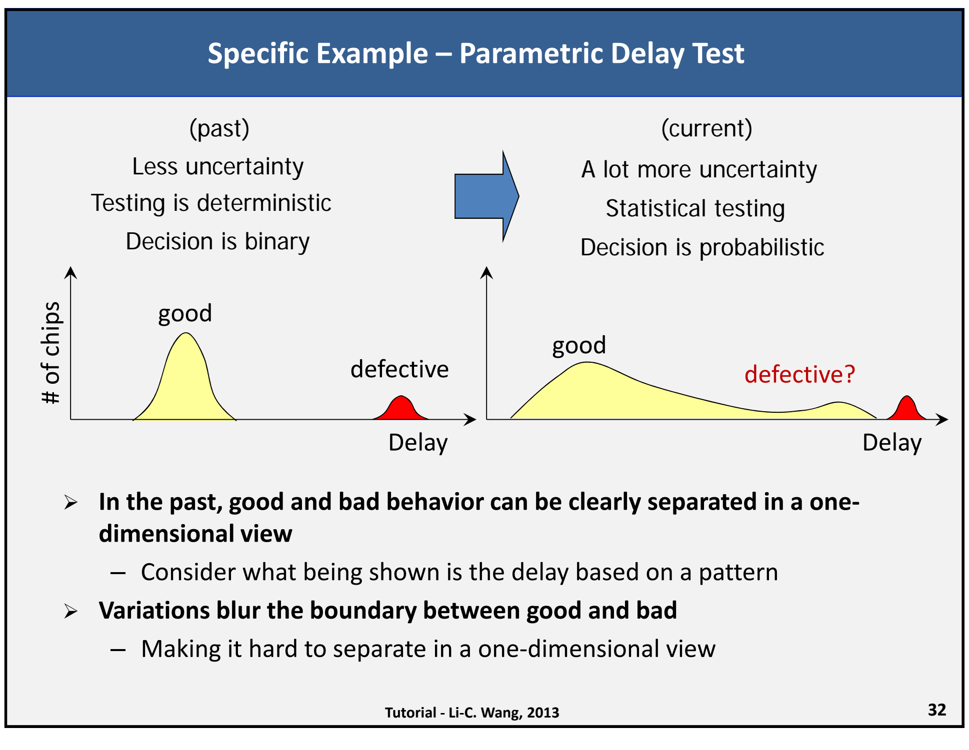


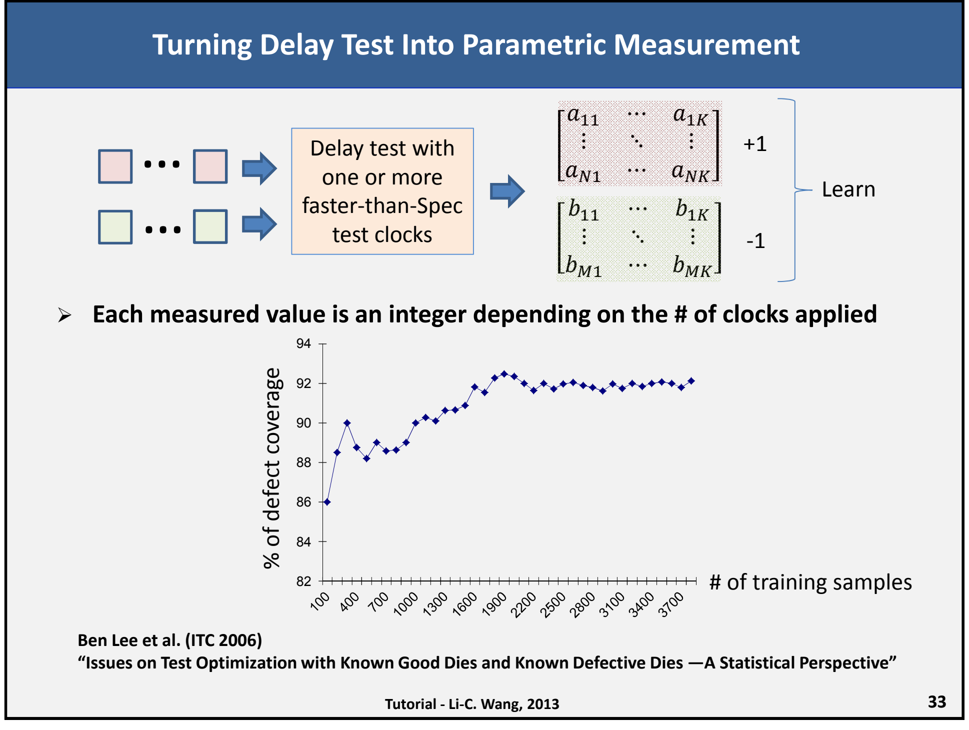
A Comparison of Classifiers

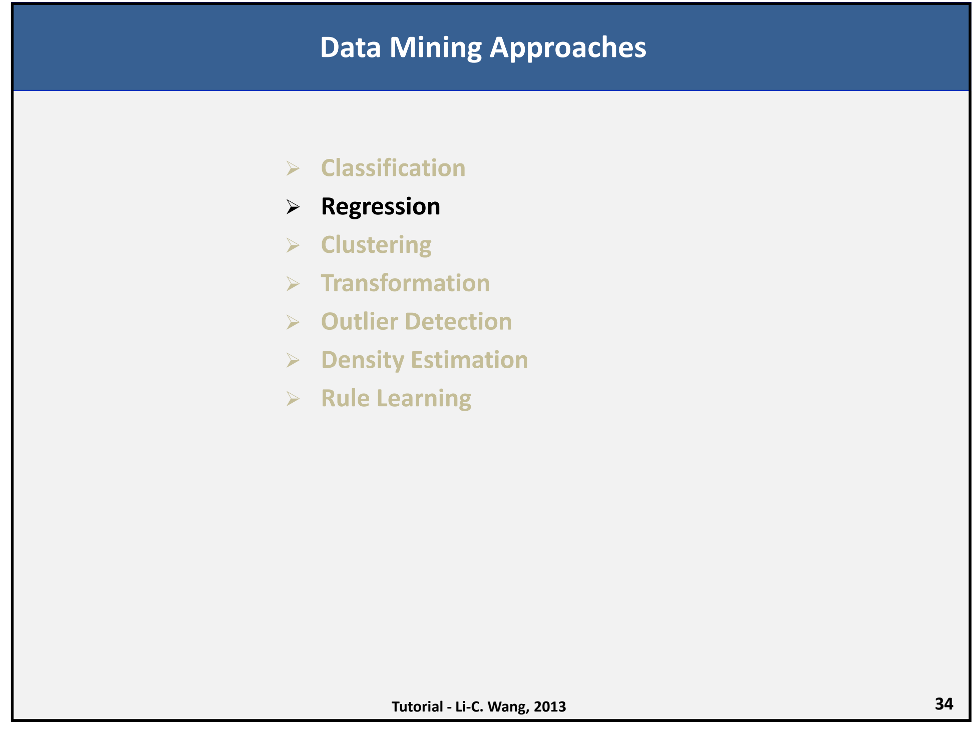
Source: http://scikit-learn.org/stable/

- **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**


An Application Example
(N+M) sample parts
An complex and expensive test flow
N parts
Class 1
Class 2
M parts
A much cheaper test flow involving K tests
+1
-1
Learn
Parts in production
Learning Model
Class 1
Class 2
Learning model tries to replace the expensive flow with the cheaper one
Tutorial - Li-C. Wang, 2013
31


Specific Example – Parametric Delay Test
(past)
Less uncertainty
Testing is deterministic
Decision is binary
(current)
A lot more uncertainty
Statistical testing
Decision is probabilistic
# of chips
good
defective
Delay
good
defective?
Delay
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
Tutorial - Li-C. Wang, 2013
32


Turning Delay Test Into Parametric Measurement
Delay test with one or more faster-than-Spec test clocks
+1
-1
Learn
Each measured value is an integer depending on the # of clocks applied
% of defect coverage
# of training samples
Ben Lee et al. (ITC 2006)
"Issues on Test Optimization with Known Good Dies and Known Defective Dies —A Statistical Perspective"
Tutorial - Li-C. Wang, 2013
33


Data Mining Approaches
Classification
Regression
Clustering
Transformation
Outlier Detection
Density Estimation
Rule Learning
Tutorial - Li-C. Wang, 2013
34

Data Mining 101 – Supervised Learning - Regression

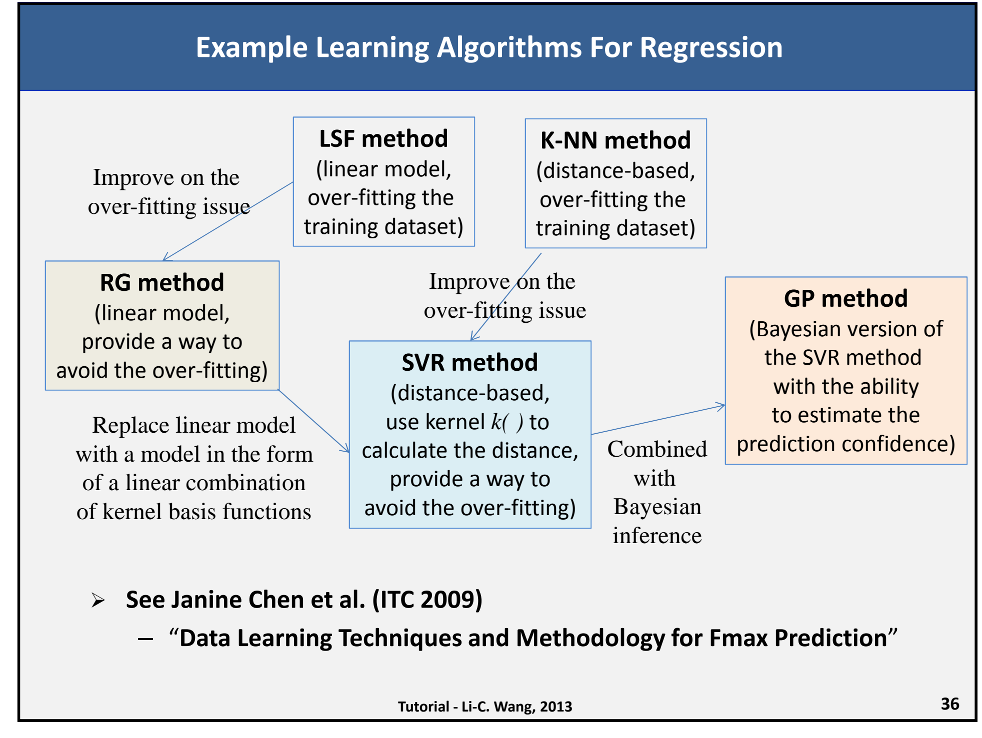
(features) $f_1 \quad f_2 \quad \dots \quad f_n$

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

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**

Tutorial - Li-C. Wang, 2013 35



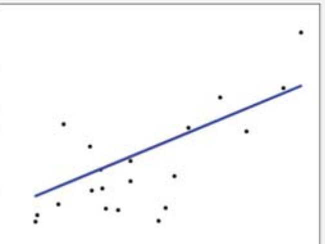
Least Square Fit

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

Assume model:

$$f(x) = w_1 x_1 + w_2 x_2 + \cdots + w_n x_n + b$$

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

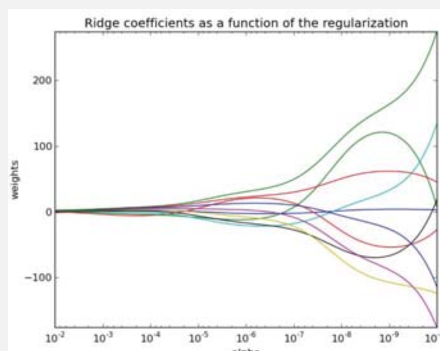


- **Assume a model**
  - Minimize the sum of squares to find values for the coefficients

Tutorial - Li-C. Wang, 2013 37

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

Tutorial - Li-C. Wang, 2013 38

An Application Example – Fmax Prediction

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

Dataset

n delay measurements

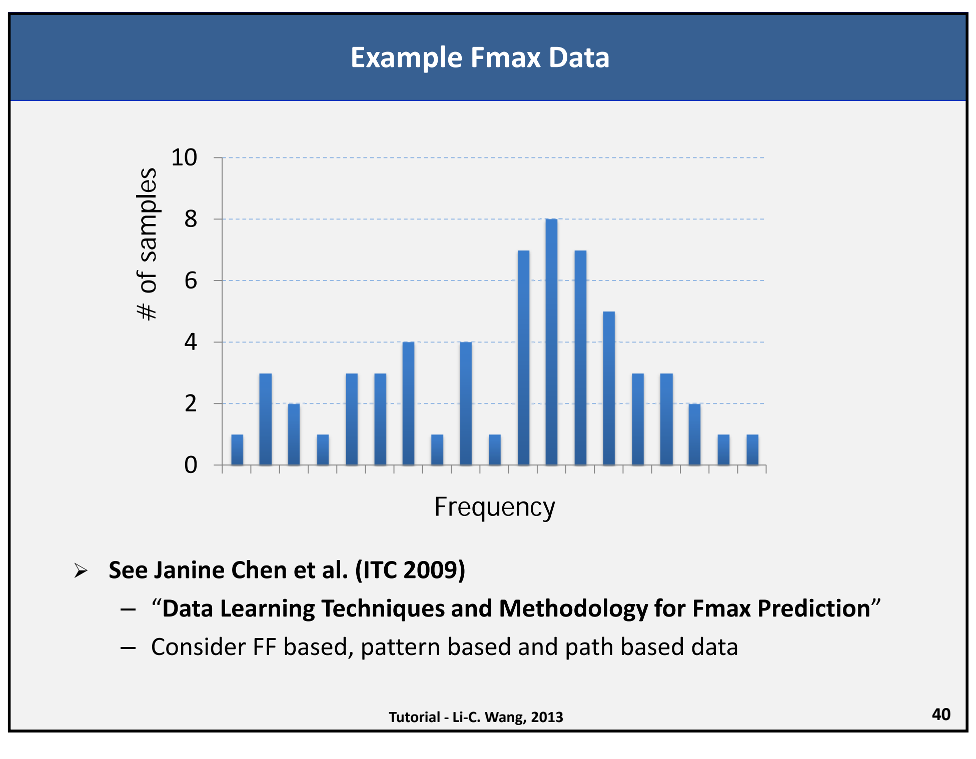
$$\mathbf{X} = \begin{vmatrix} \vec{x}_1 \\ \vec{x}_2 \\ \cdots \\ \vec{x}_m \end{vmatrix} = \begin{vmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{vmatrix} \quad \mathbf{y} = \begin{vmatrix}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

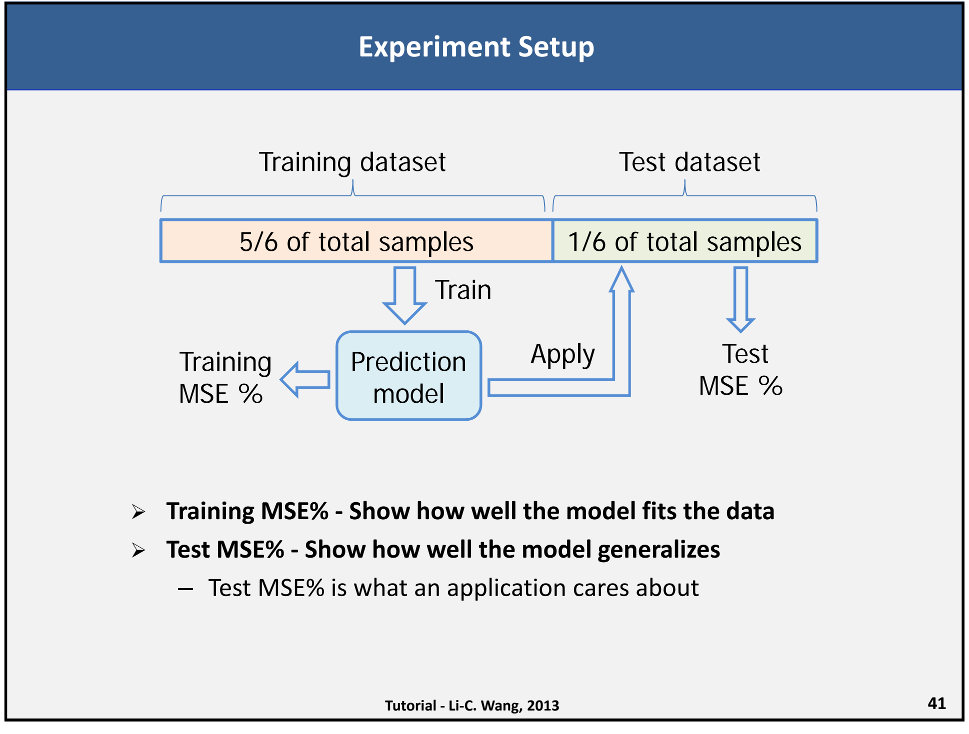
Columns: M_1, M_2, …, M_n; Fmax; m samples chips

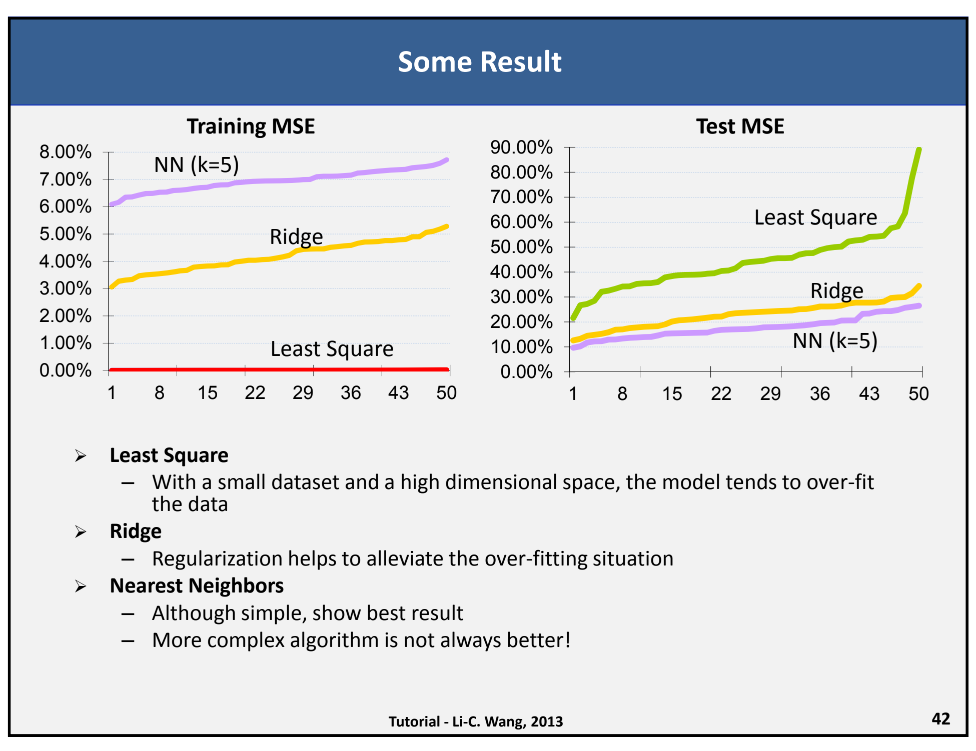
Fmax of c?

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

Tutorial - Li-C. Wang, 2013 39




Experiment Setup
Training dataset
Test dataset
5/6 of total samples
1/6 of total samples
Train
Training MSE %
Prediction model
Apply
Test MSE %
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
Tutorial - Li-C. Wang, 2013
41


Some Result
Training MSE
NN (k=5)
Ridge
Least Square
Test MSE
Least Square
Ridge
NN (k=5)
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!
Tutorial - Li-C. Wang, 2013
42

Data Mining Approaches

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

Tutorial - Li-C. Wang, 2013 43

Data Mining 101 – Unsupervised Learning

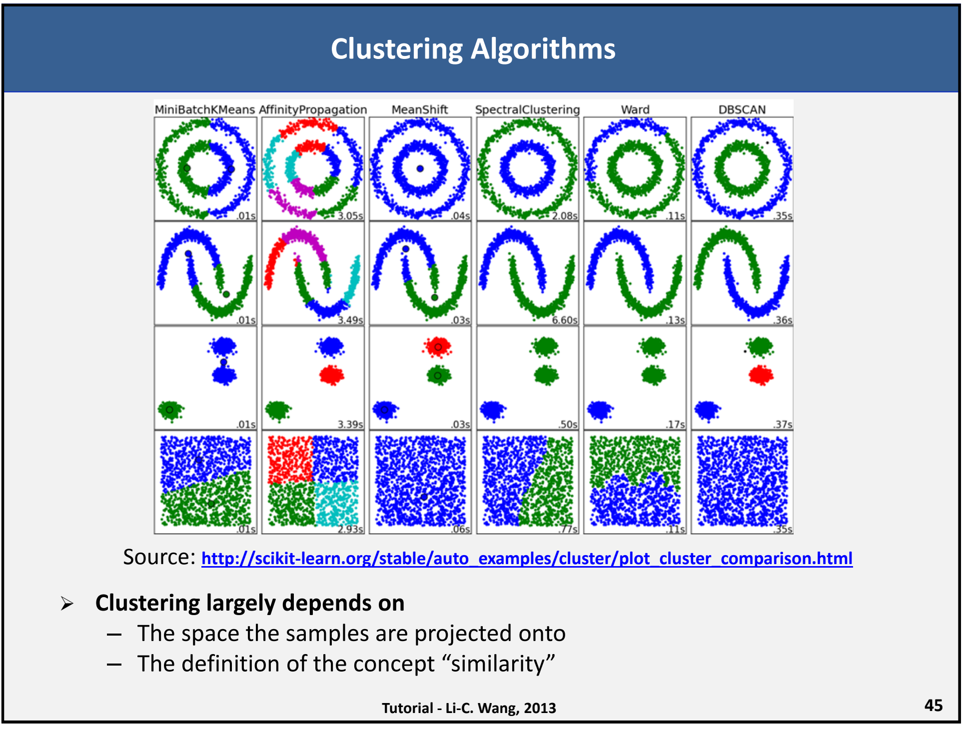
(features) $f_1 \quad f_2 \quad \dots \quad f_n$

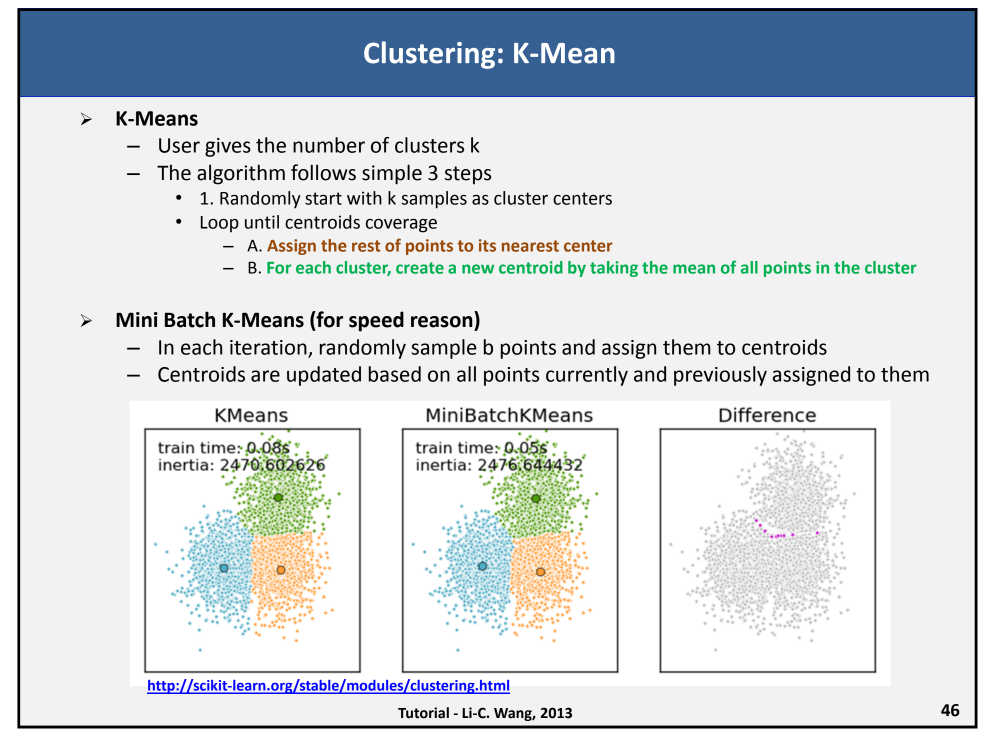
$$\mathbf{X} = \begin{vmatrix} \vec{x}_1 \\ \vec{x}_2 \\ \dots \\ \vec{x}_m \end{vmatrix} = \begin{vmatrix} x_{11} & x_{12} & \dots & x_{1n} \\ x_{21} & x_{22} & \dots & x_{2n} \\ \dots & \dots & \dots & \dots \\ x_{m1} & x_{m2} & \dots & x_{mn} \end{vmatrix} \qquad \vec{y} = \begin{vmatrix} y_1 \\ y_2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

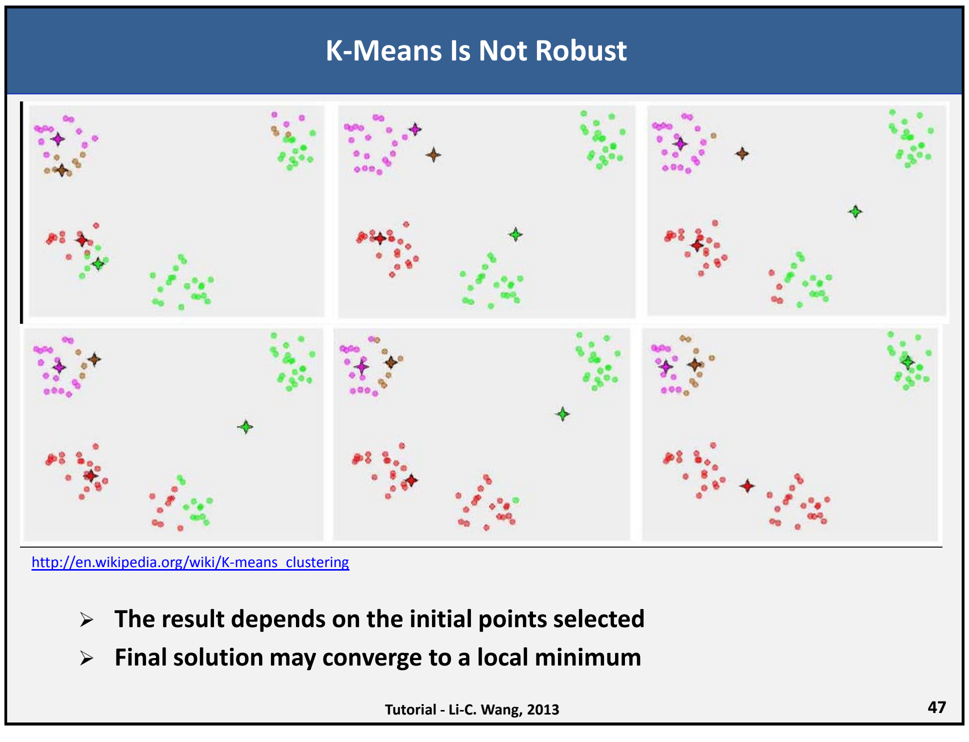
No y's

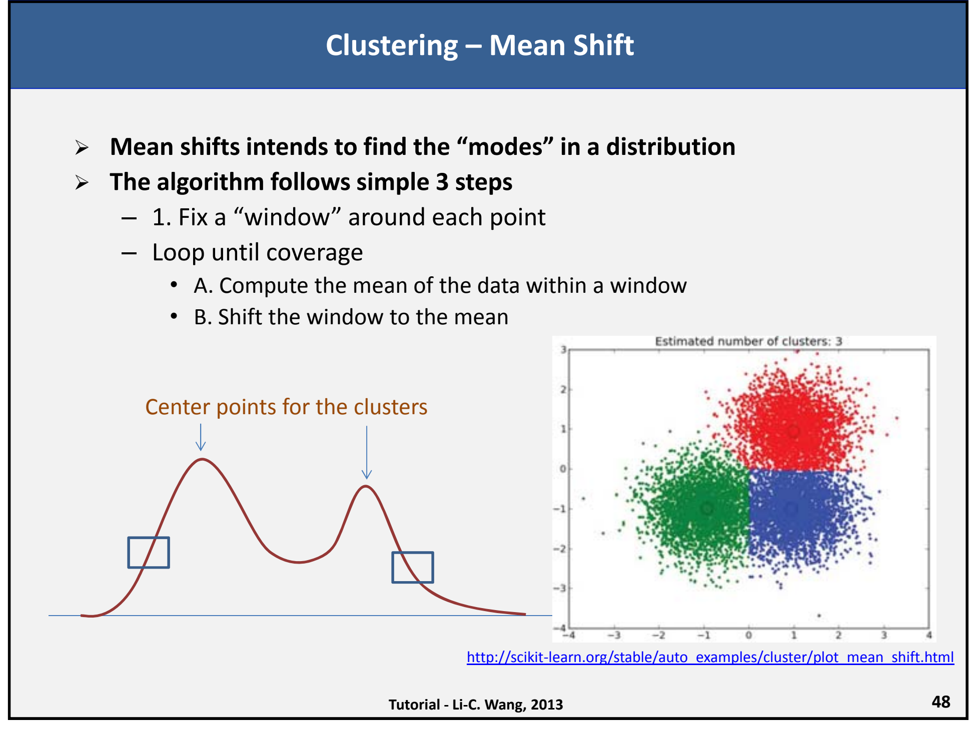
- **Popular approaches**
  - Clustering
  - Transformation (dimension reduction)
  - Novelty Detection (Outlier analysis)
  - Density Estimation

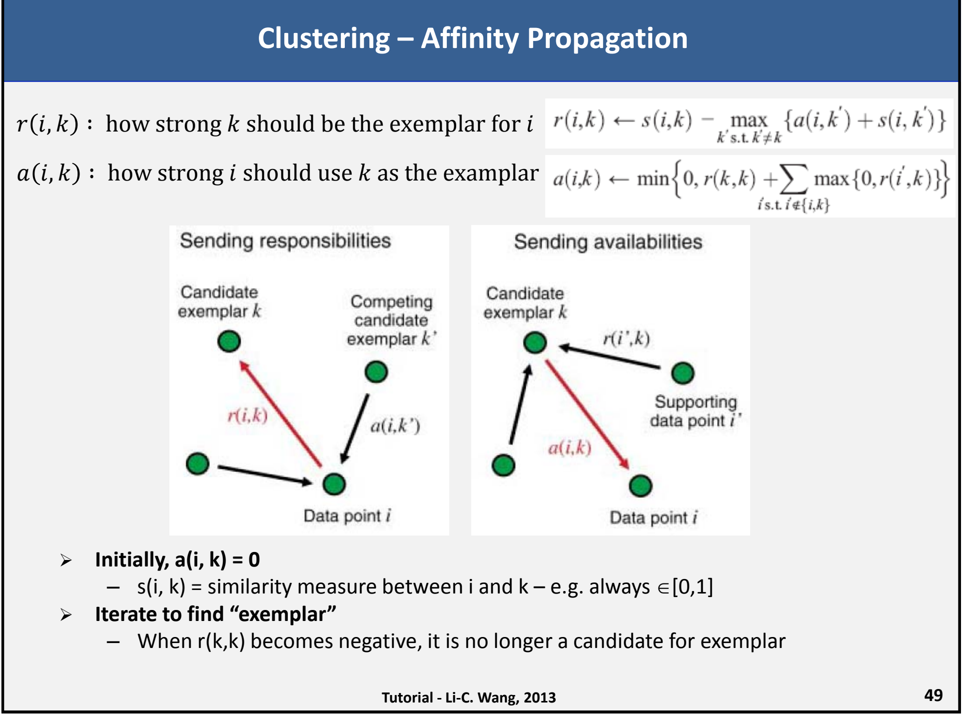
Tutorial - Li-C. Wang, 2013 44

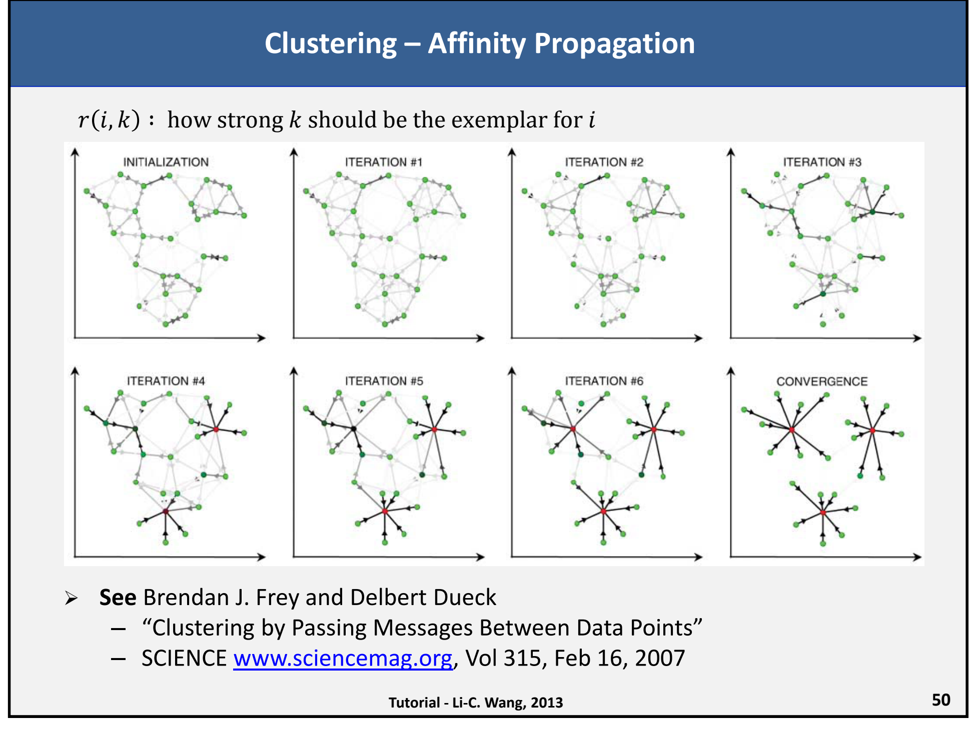

Clustering Algorithms
MiniBatchKMeans
AffinityPropagation
MeanShift
SpectralClustering
Ward
DBSCAN
Source: http://scikit-learn.org/stable/auto_examples/cluster/plot_cluster_comparison.html
Clustering largely depends on
The space the samples are projected onto
The definition of the concept "similarity"
Tutorial - Li-C. Wang, 2013
45


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
KMeans
train time: 0.08s
inertia: 2470.602626
MiniBatchKMeans
train time: 0.05s
inertia: 2476.644432
Difference
http://scikit-learn.org/stable/modules/clustering.html
Tutorial - Li-C. Wang, 2013
46


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
Tutorial - Li-C. Wang, 2013
47


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
Center points for the clusters
Estimated number of clusters: 3
http://scikit-learn.org/stable/auto_examples/cluster/plot_mean_shift.html
Tutorial - Li-C. Wang, 2013
48


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
$a(i,k) \leftarrow \min\{0, r(k,k) + \sum_{i' \text{s.t.} i' \notin \{i,k\}} \max\{0, r(i',k)\}\}$
Sending responsibilities
Candidate exemplar k
Competing candidate exemplar k'
$r(i,k)$
$a(i,k')$
Data point i
Sending availabilities
Candidate exemplar k
$r(i',k)$
Supporting data point i'
$a(i,k)$
Data point i
Initially, a(i, k) = 0
– s(i, k) = similarity measure between i and k – e.g. always ∈[0,1]
Iterate to find "exemplar"
– When r(k,k) becomes negative, it is no longer a candidate for exemplar
Tutorial - Li-C. Wang, 2013
49

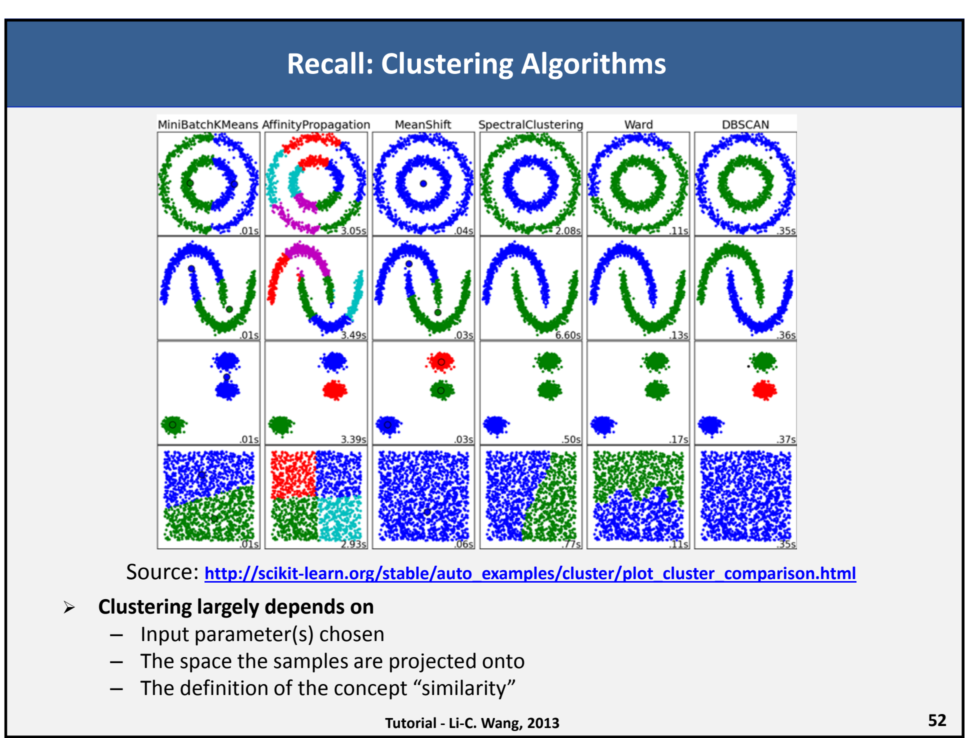

Clustering – Affinity Propagation
$r(i,k)$: how strong k should be the exemplar for i
INITIALIZATION
ITERATION #1
ITERATION #2
ITERATION #3
ITERATION #4
ITERATION #5
ITERATION #6
CONVERGENCE
See Brendan J. Frey and Delbert Dueck
– "Clustering by Passing Messages Between Data Points"
– SCIENCE www.sciencemag.org, Vol 315, Feb 16, 2007
Tutorial - Li-C. Wang, 2013
50

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

Tutorial - Li-C. Wang, 2013 51

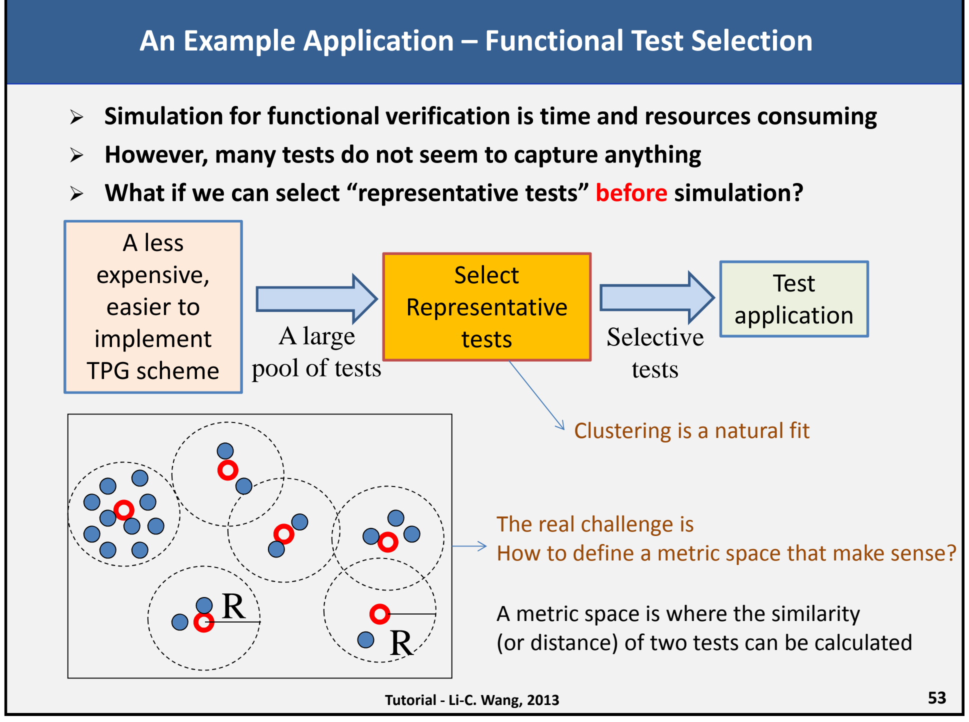
Recall: Clustering Algorithms

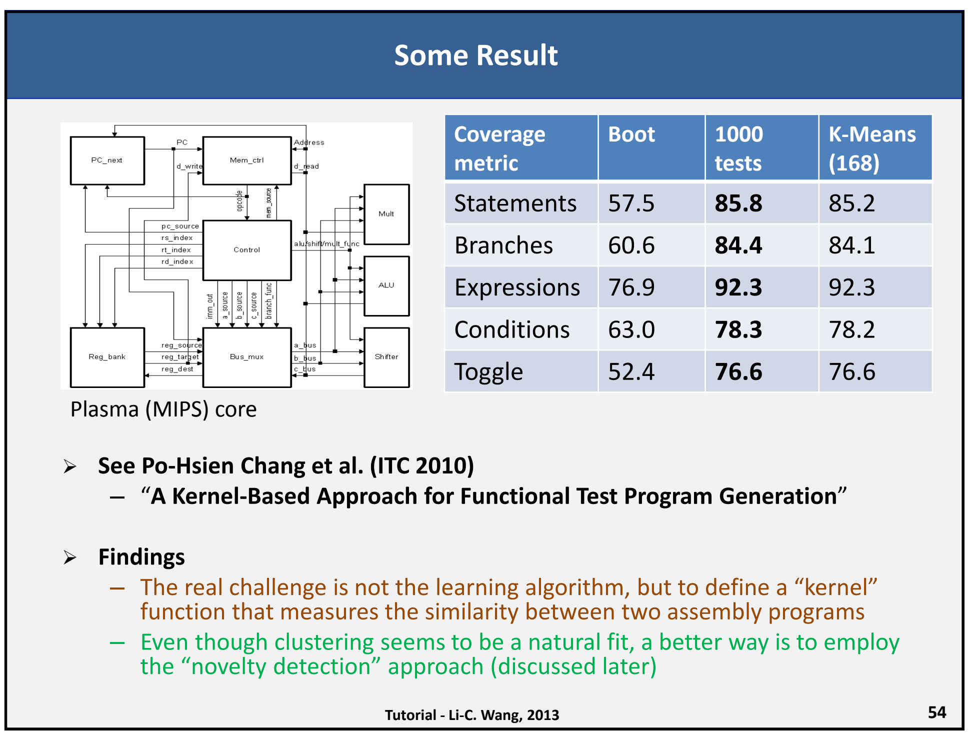


Source: http://scikit-learn.org/stable/auto_examples/cluster/plot_cluster_comparison.html

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

Tutorial - Li-C. Wang, 2013 52





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

Data Mining Approaches

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

Tutorial - Li-C. Wang, 2013 55

Transformation – Principal Component Analysis

f_1 f_2 ... f_N

M samples

s_1 $\vec{r}_1$ = [d_{11}, d_{12}, ... d_{1N}]

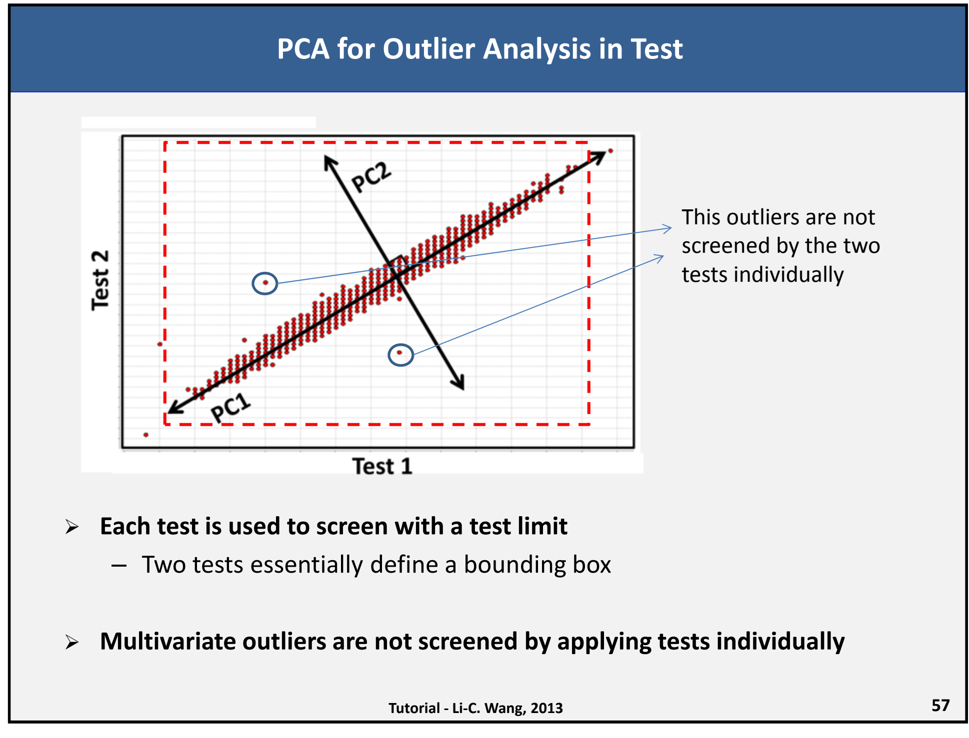
⋮ ⋮

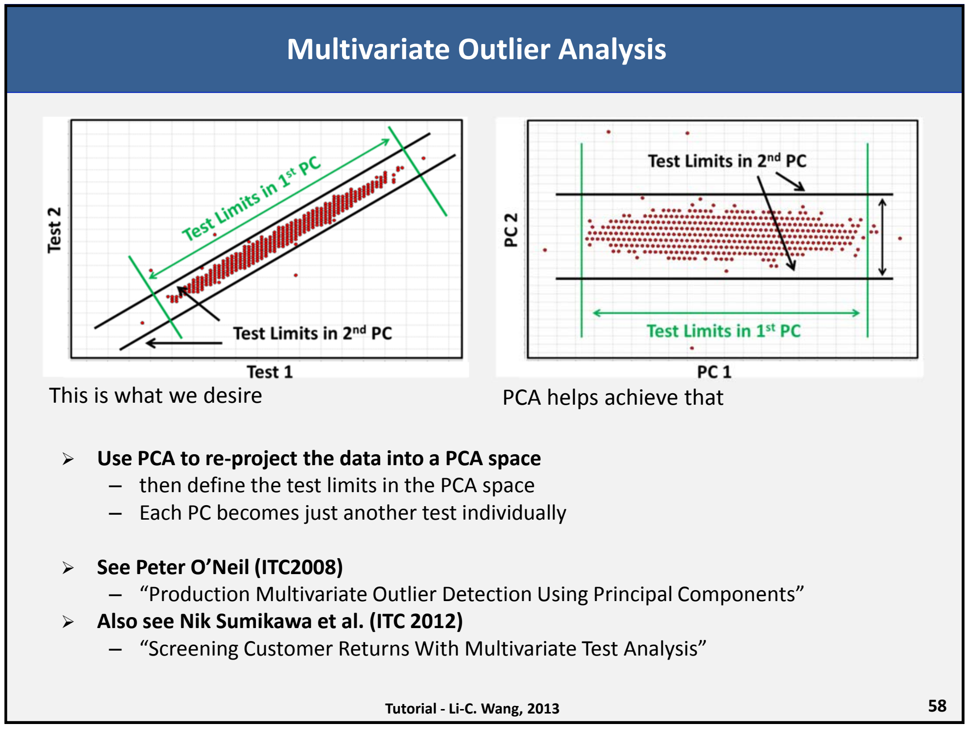
s_M $\vec{r}_M$ = [d_1, d_{K2}, ... D_{MN}]

→ PCA → *Re-Projection of data in a PCA space*

- **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

Tutorial - Li-C. Wang, 2013 56


PCA for Outlier Analysis in Test
PC2
PC1
Test 2
Test 1
This outliers are not screened by the two tests individually
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
Tutorial - Li-C. Wang, 2013
57


Multivariate Outlier Analysis
Test Limits in 1st PC
Test Limits in 2nd PC
Test 2
Test 1
This is what we desire
Test Limits in 2nd PC
Test Limits in 1st PC
PC 2
PC 1
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
See Peter O'Neil (ITC2008)
– "Production Multivariate Outlier Detection Using Principal Components"
Also see Nik Sumikawa et al. (ITC 2012)
– "Screening Customer Returns With Multivariate Test Analysis"
Tutorial - Li-C. Wang, 2013
58

Data Mining Approaches

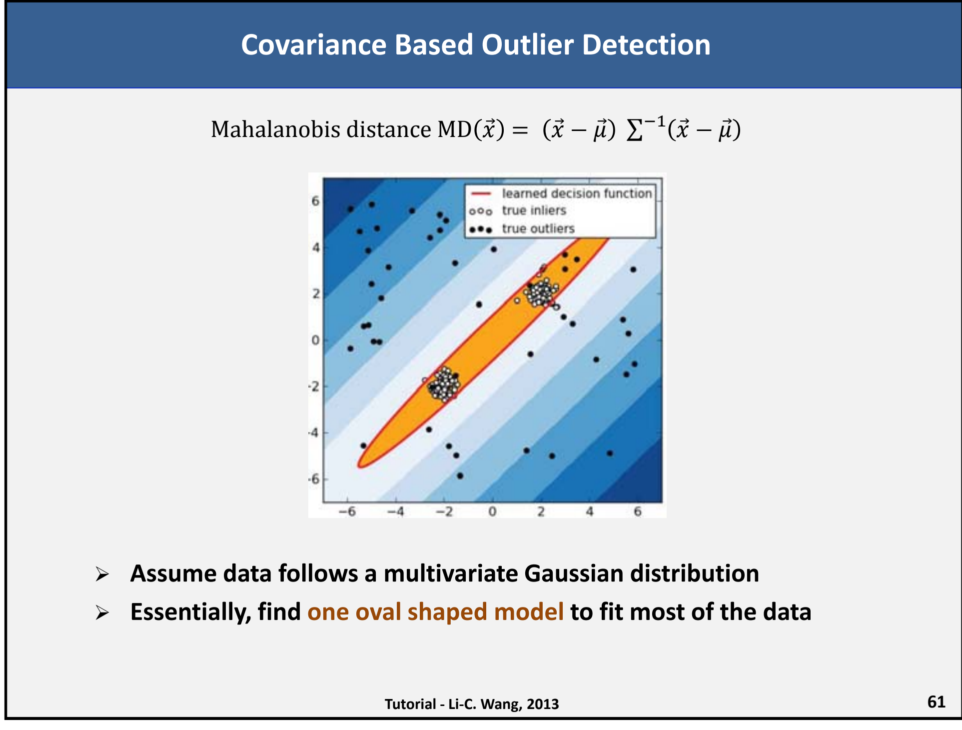
- Classification
- Regression
- Clustering
- Transformation
- **Outlier Detection**
- Density Estimation
- Rule Learning

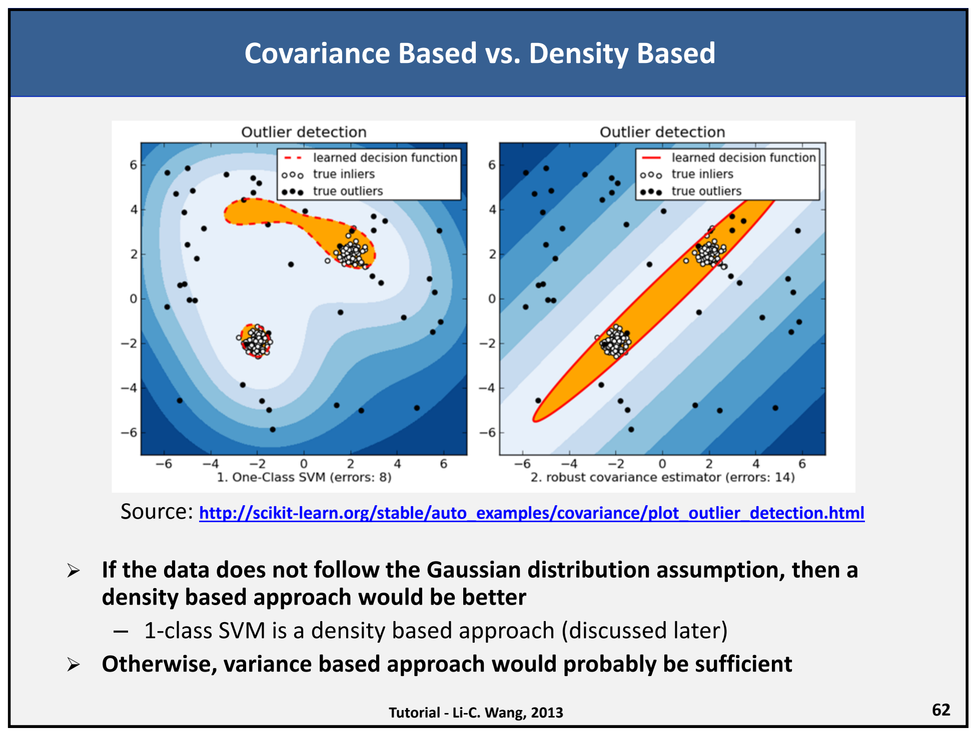
Tutorial - Li-C. Wang, 2013 59

Novelty Detection – Outlier Analysis

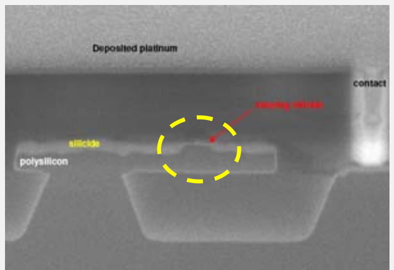
- **Principal Component Analysis**
- **Covariance based**
  - Mahalanobis distances
- **Density based**
  - Support Vector Machine one class
- **Tree based**
  - Random Forest
- **Not the same as clustering**
  - We only care about finding outliers

Tutorial - Li-C. Wang, 2013 60


Covariance Based Outlier Detection
Mahalanobis distance $\mathrm{MD}(\vec{x}) = (\vec{x}-\vec{\mu}) \sum^{-1}(\vec{x}-\vec{\mu})$
learned decision function
true inliers
true outliers
Assume data follows a multivariate Gaussian distribution
Essentially, find one oval shaped model to fit most of the data
Tutorial - Li-C. Wang, 2013
61


Covariance Based vs. Density Based
Outlier detection
learned decision function
true inliers
true outliers
1. One-Class SVM (errors: 8)
Outlier detection
2. robust covariance estimator (errors: 14)
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
Tutorial - Li-C. Wang, 2013
62

An Application – Customer Return Analysis



- **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

Tutorial - Li-C. Wang, 2013 63

Outlier Model For Customer Return

2nd Principal Component

1st Principal Component

- **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"

Tutorial - Li-C. Wang, 2013 64

Data Mining Approaches

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

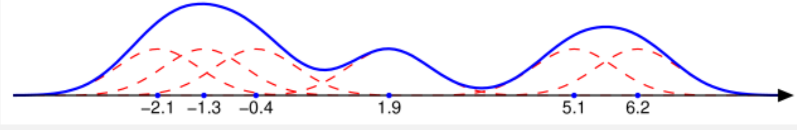
Density Estimation

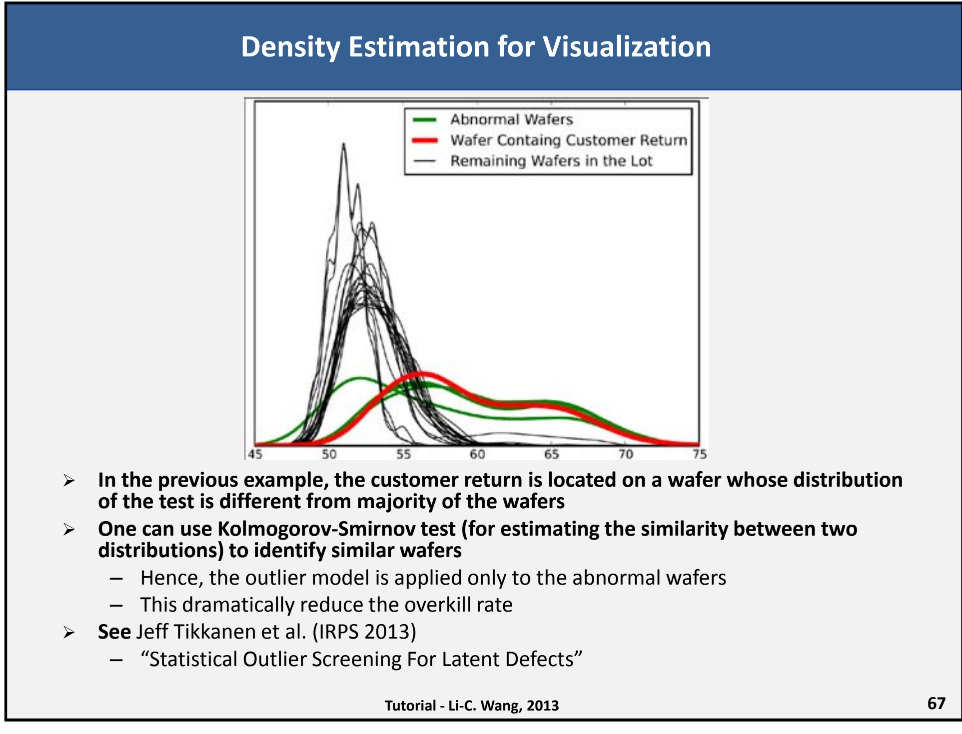
- **For density estimation, several non-parametric methods were proposed in 1960s**
  - Non-parametric because no fixed functional is given
- **One famous example is the Parzen's window**
  - Requires the definition of a *kernel* function that is a symmetric unimodal density function

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

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

Gaussian kernel





Data Mining Approaches

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

Tutorial - Li-C. Wang, 2013 68

Data Mining 101 – Rule Learning

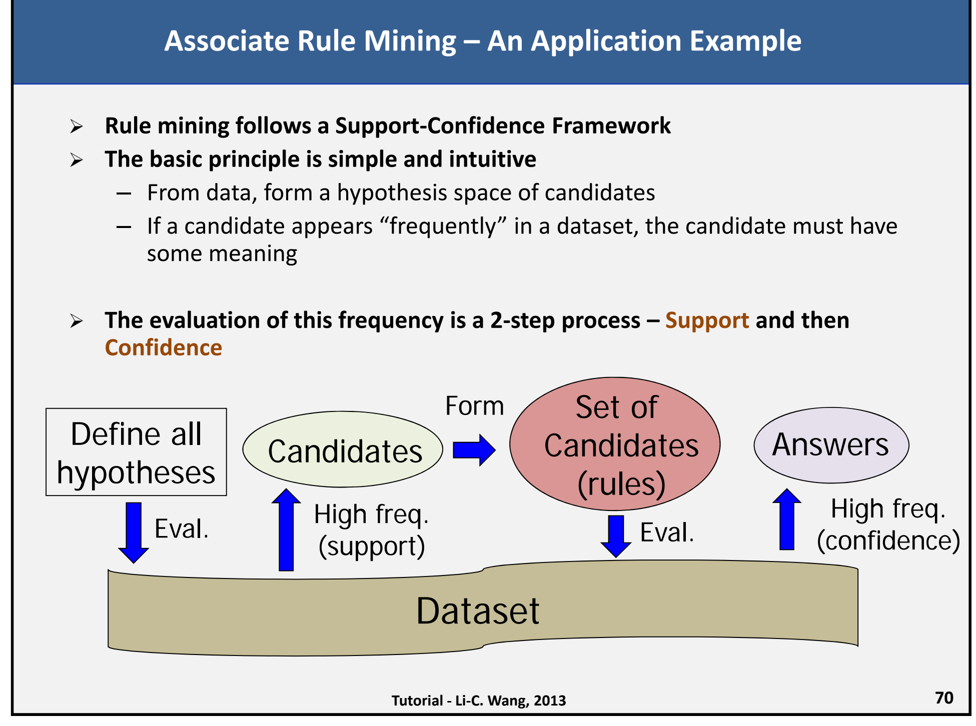
(features) $f_1 \quad f_2 \quad \dots \quad f_n$

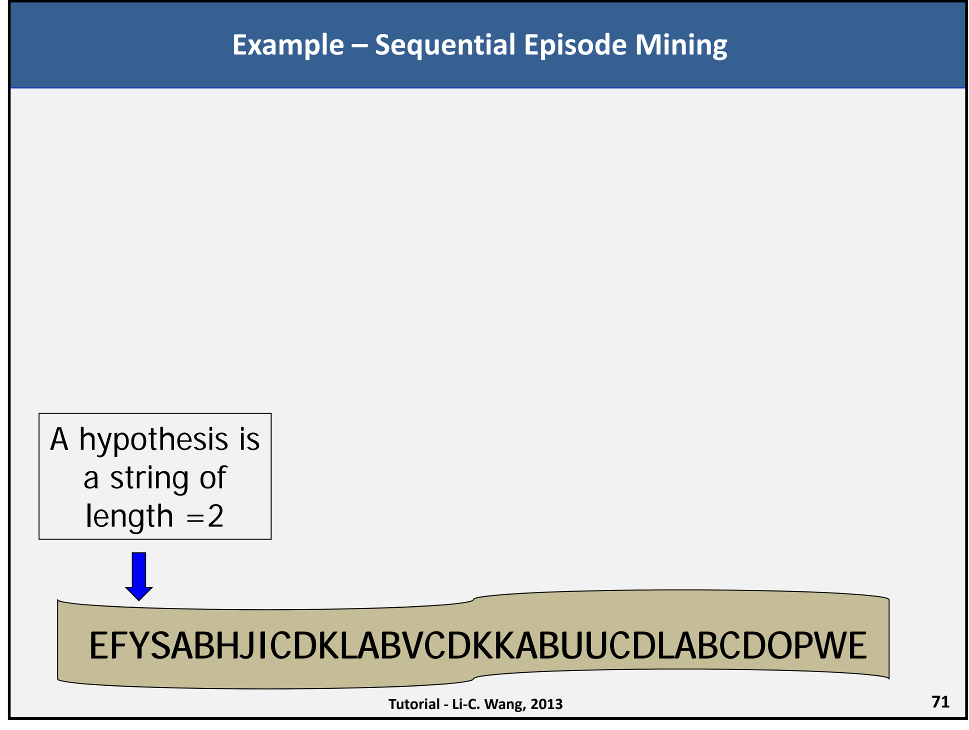
$$\mathbf{X} = \begin{vmatrix} \vec{x}_1 \\ \vec{x}_2 \\ \dots \\ \vec{x}_m \end{vmatrix} = \begin{vmatrix} x_{11} & x_{12} & \dots & x_{1n} \\ x_{21} & x_{22} & \dots & x_{2n} \\ \dots & \dots & \dots & \dots \\ x_{m1} & x_{m2} & \dots & x_{mn} \end{vmatrix} \qquad \vec{y} = \begin{vmatrix} y_1 \\ y_2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

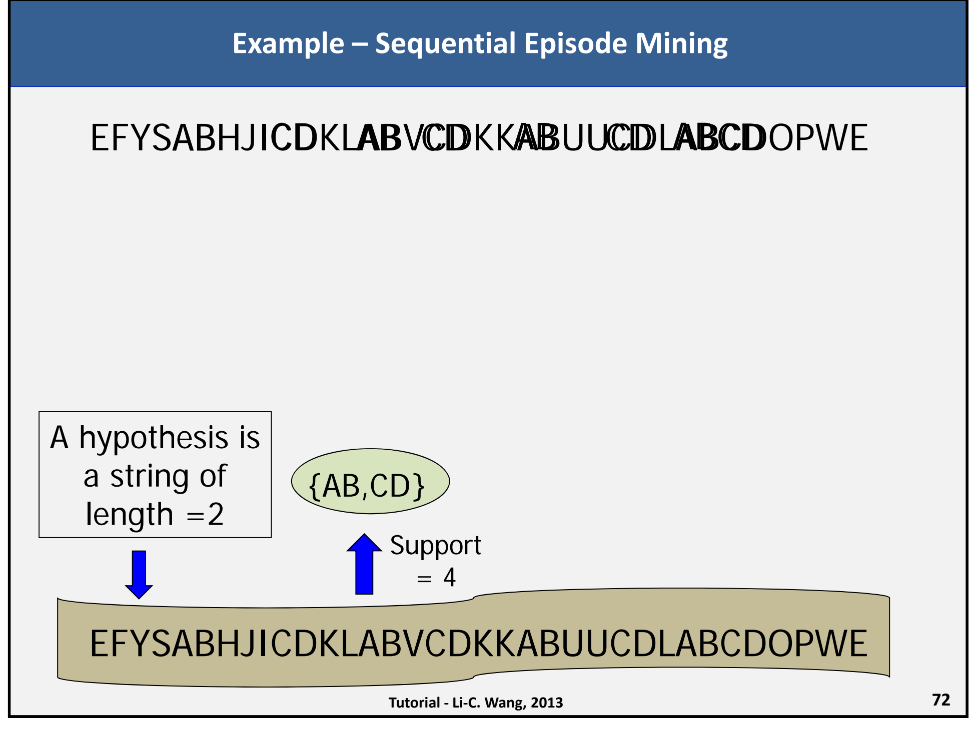
Binary label

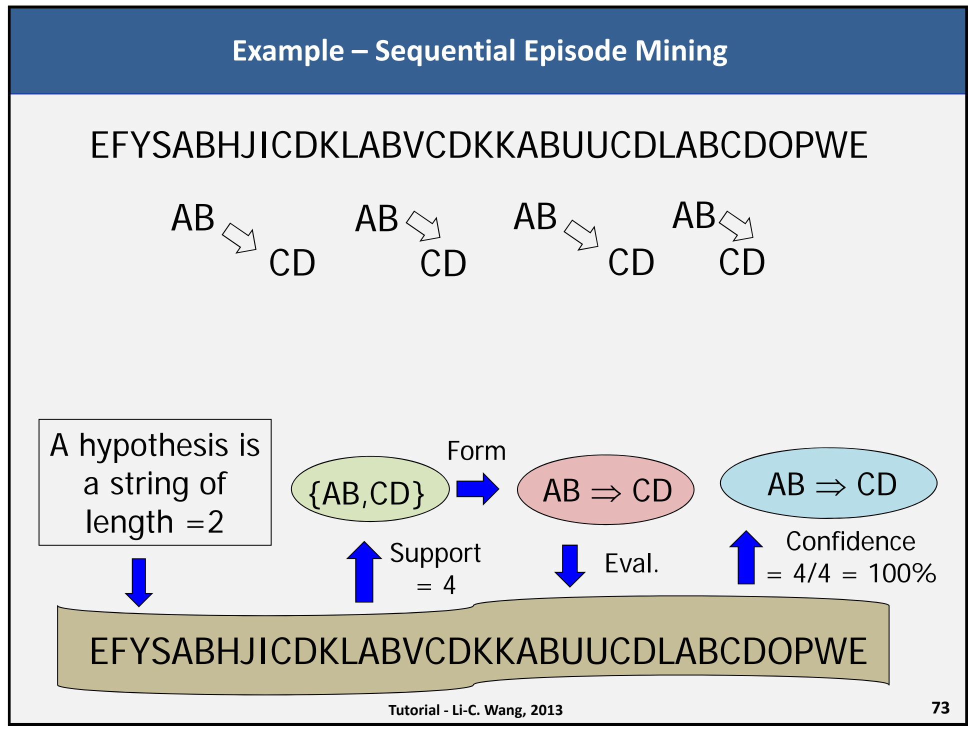
- **With y's label (binary class)**
  - **Classification rule learning**
- **Without y's label (unsupervised)**
  - **Association rule mining**

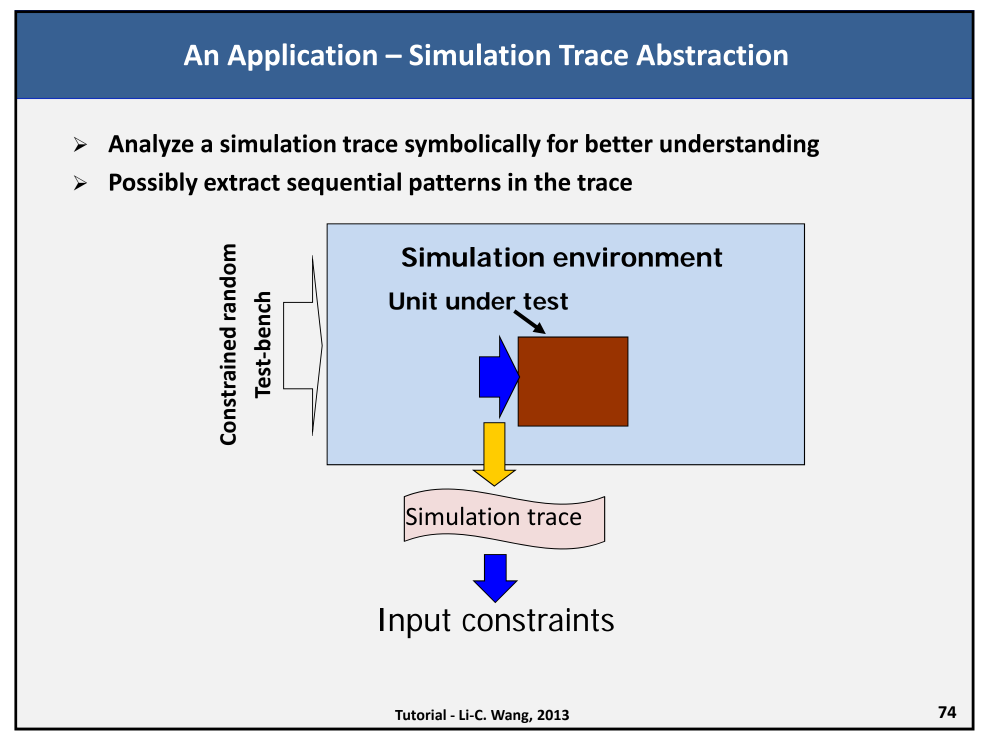
Tutorial - Li-C. Wang, 2013 69

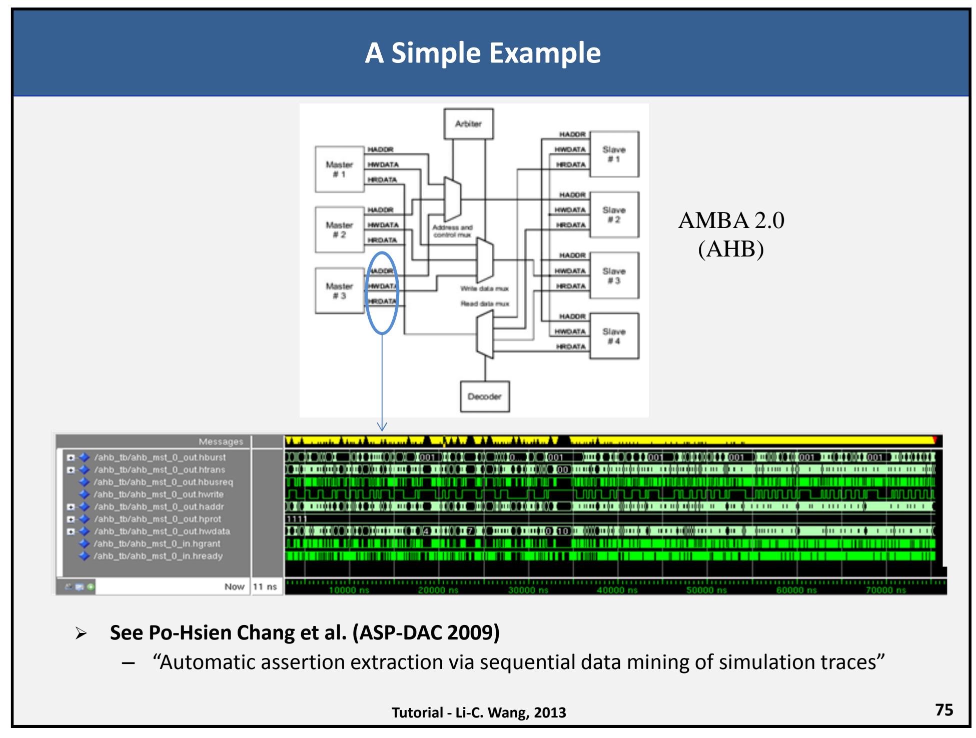


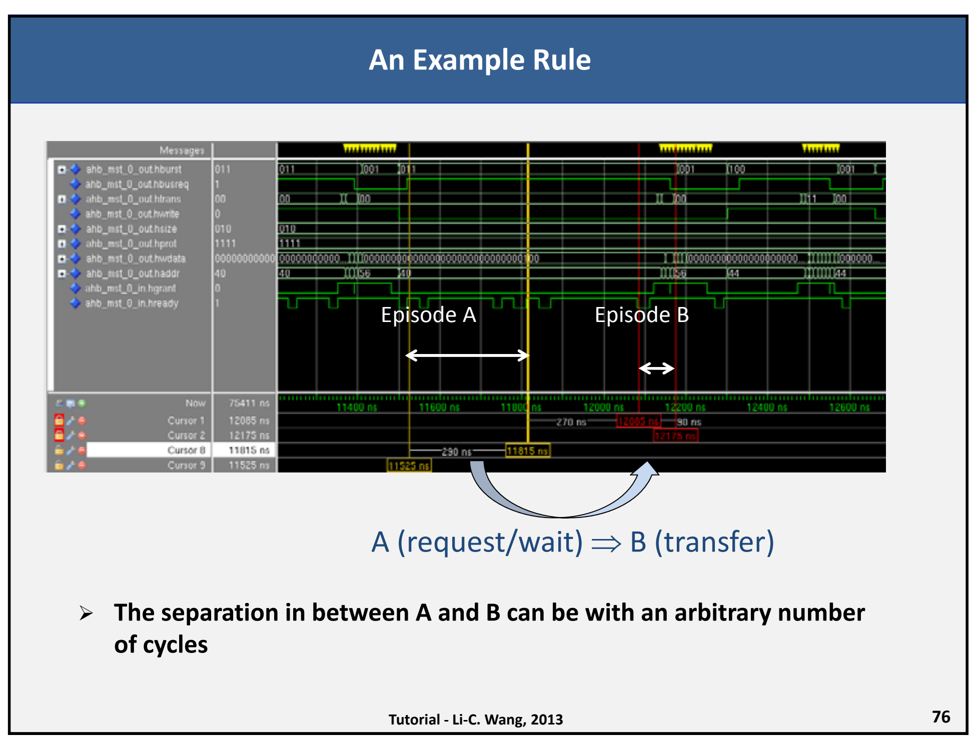

Example – Sequential Episode Mining
A hypothesis is a string of length =2
EFYSABHJICDKLABVCDKKABUUCDLABCDOPWE
Tutorial - Li-C. Wang, 2013
71

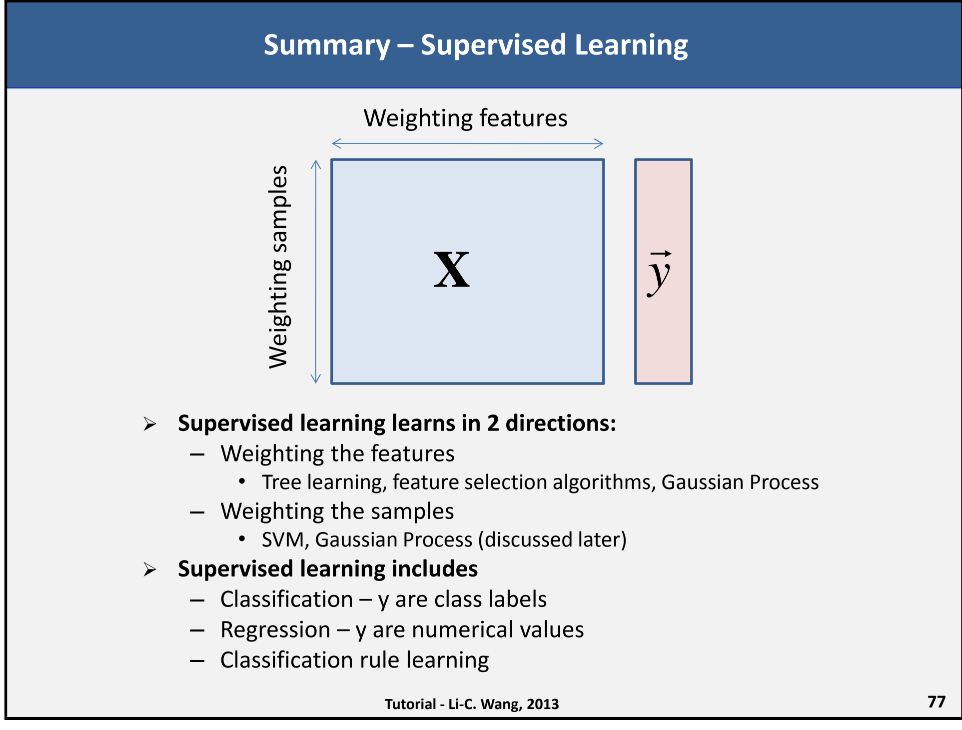

Example – Sequential Episode Mining
EFYSABHJICDKLABVCDKKABUUCDLABCDOPWE
A hypothesis is a string of length =2
{AB,CD}
Support = 4
EFYSABHJICDKLABVCDKKABUUCDLABCDOPWE
Tutorial - Li-C. Wang, 2013
72

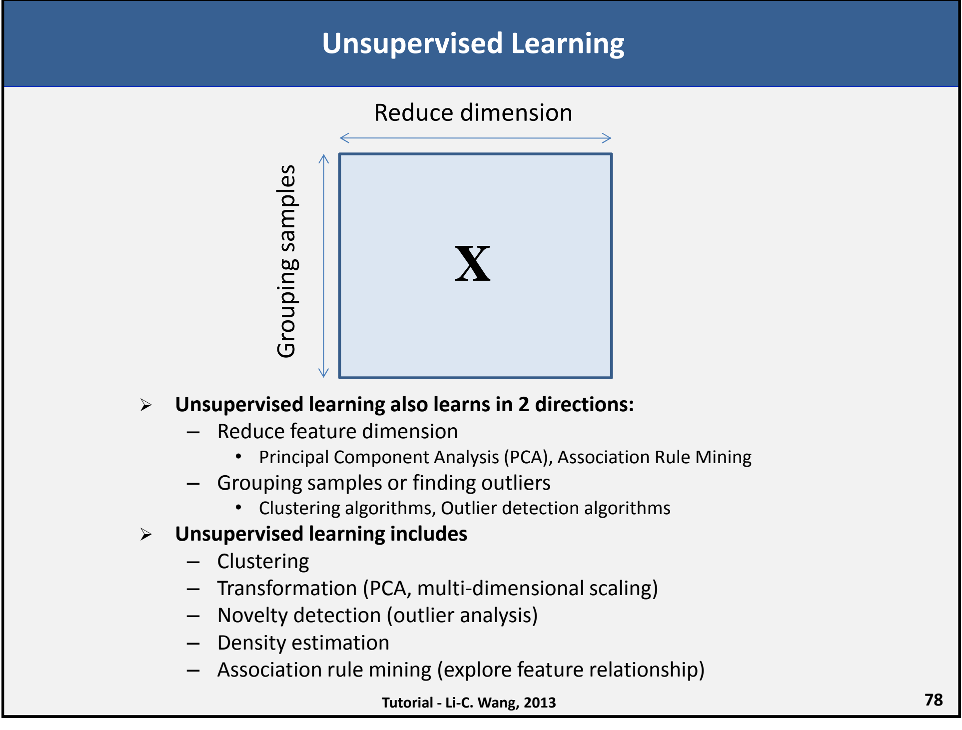

Example – Sequential Episode Mining
EFYSABHJICDKLABVCDKKABUUCDLABCDOPWE
AB
CD
AB
CD
AB
CD
AB
CD
A hypothesis is a string of length =2
{AB,CD}
Form
AB ⇒ CD
AB ⇒ CD
Support = 4
Eval.
Confidence = 4/4 = 100%
EFYSABHJICDKLABVCDKKABUUCDLABCDOPWE
Tutorial - Li-C. Wang, 2013
73


An Application – Simulation Trace Abstraction
Analyze a simulation trace symbolically for better understanding
Possibly extract sequential patterns in the trace
Constrained random
Test-bench
Simulation environment
Unit under test
Simulation trace
Input constraints
Tutorial - Li-C. Wang, 2013
74


A Simple Example
AMBA 2.0
(AHB)
See Po-Hsien Chang et al. (ASP-DAC 2009)
"Automatic assertion extraction via sequential data mining of simulation traces"
Tutorial - Li-C. Wang, 2013
75


An Example Rule
Episode A
Episode B
A (request/wait) ⇒ B (transfer)
The separation in between A and B can be with an arbitrary number of cycles
Tutorial - Li-C. Wang, 2013
76


Summary – Supervised Learning
Weighting features
Weighting samples
X
$\vec{y}$
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
Tutorial - Li-C. Wang, 2013
77


Unsupervised Learning
Reduce dimension
Grouping samples
X
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)
Tutorial - Li-C. Wang, 2013
78

Learning Theory, SVM and Kernel Method (60->15 minutes)

Tutorial - Li-C. Wang, 2013 79

Classification, Machine Learning, Pattern Recognition

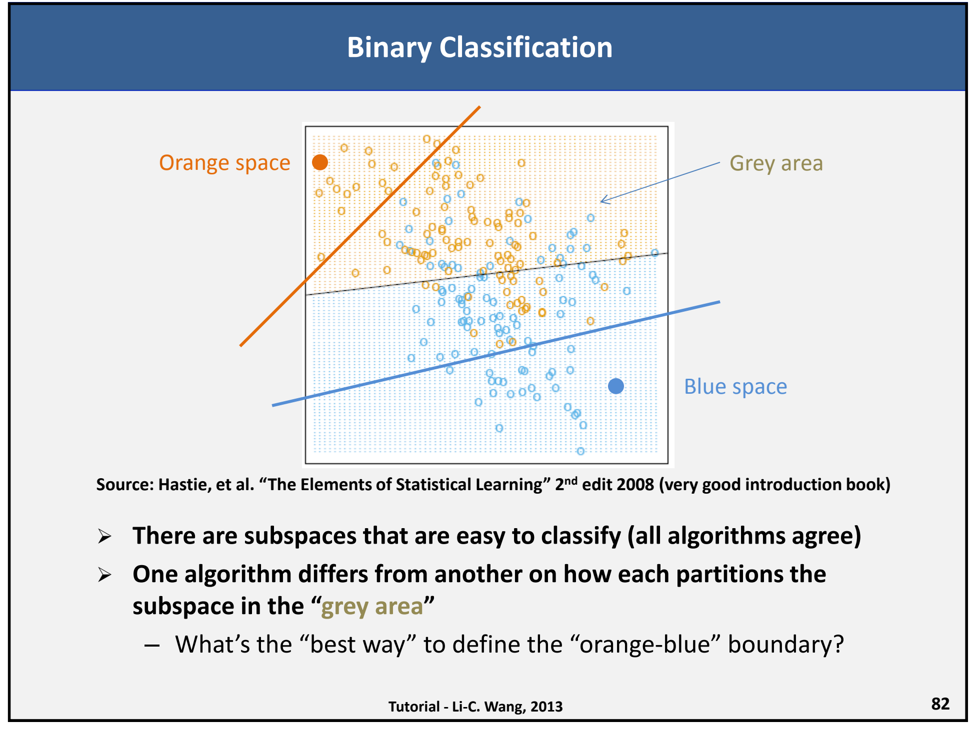
- **In machine learning, Perceptron is widely considered as one of the earliest examples to show that a machine can actually "learn"**
- **SVM is based on statistical learning theory that provides the necessary and sufficient conditions where a machine is guaranteed to "learn"**

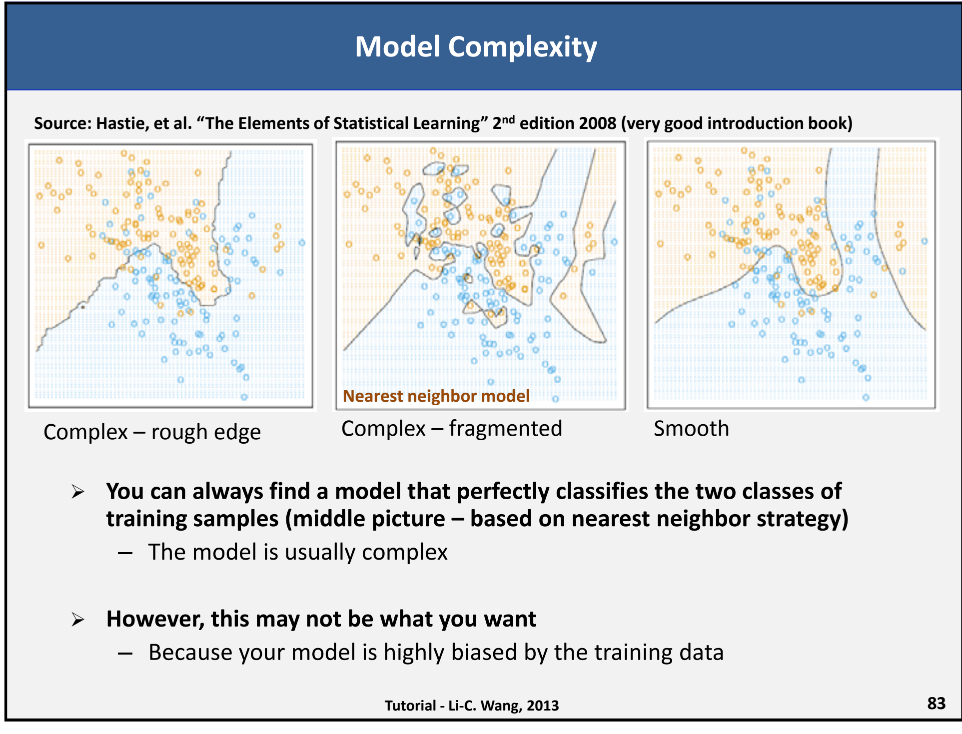
Tutorial - Li-C. Wang, 2013 80

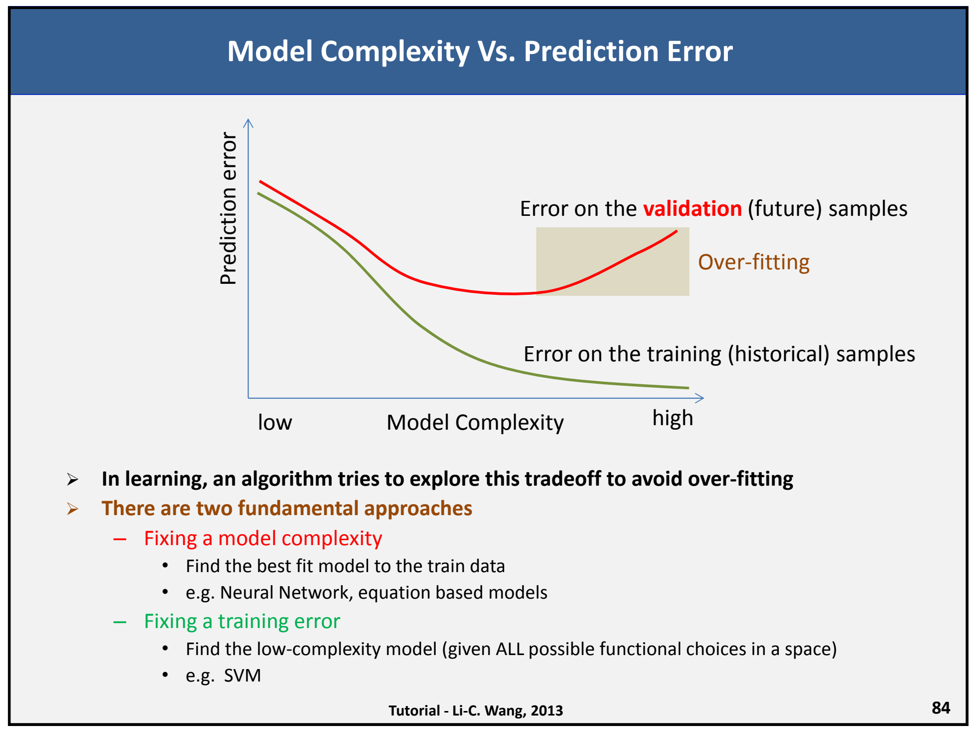
A Popular Dataset For Machine Learning Research

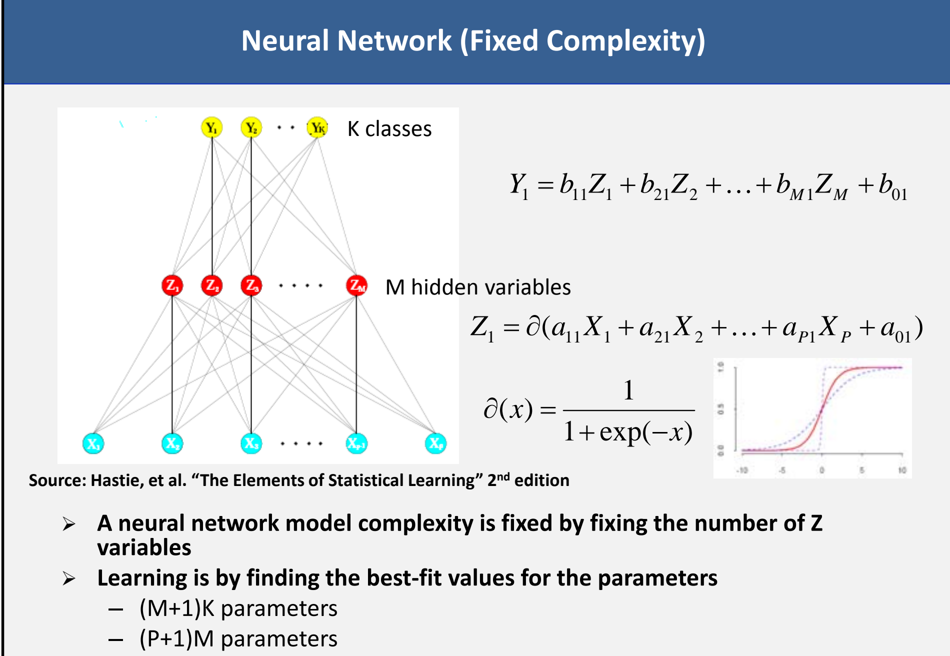
Source: Hastie, et al. "The Elements of Statistical Learning" 2nd edition 2008 (very good introduction book)

- **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?**




Model Complexity
Source: Hastie, et al. "The Elements of Statistical Learning" 2nd edition 2008 (very good introduction book)
Nearest neighbor model
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
Tutorial - Li-C. Wang, 2013
83


Model Complexity Vs. Prediction Error
Prediction error
Error on the validation (future) samples
Over-fitting
Error on the training (historical) samples
low
Model Complexity
high
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
Find the low-complexity model (given ALL possible functional choices in a space)
e.g. SVM
Tutorial - Li-C. Wang, 2013
84



Support Vector Machine

- **Fix the training error, 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?

Tutorial - Li-C. Wang, 2013 86

What Is The Model Like?

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

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

- **An SVM model always take the following form:**

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

Weighted average of similarity measures

Tutorial - Li-C. Wang, 2013 87

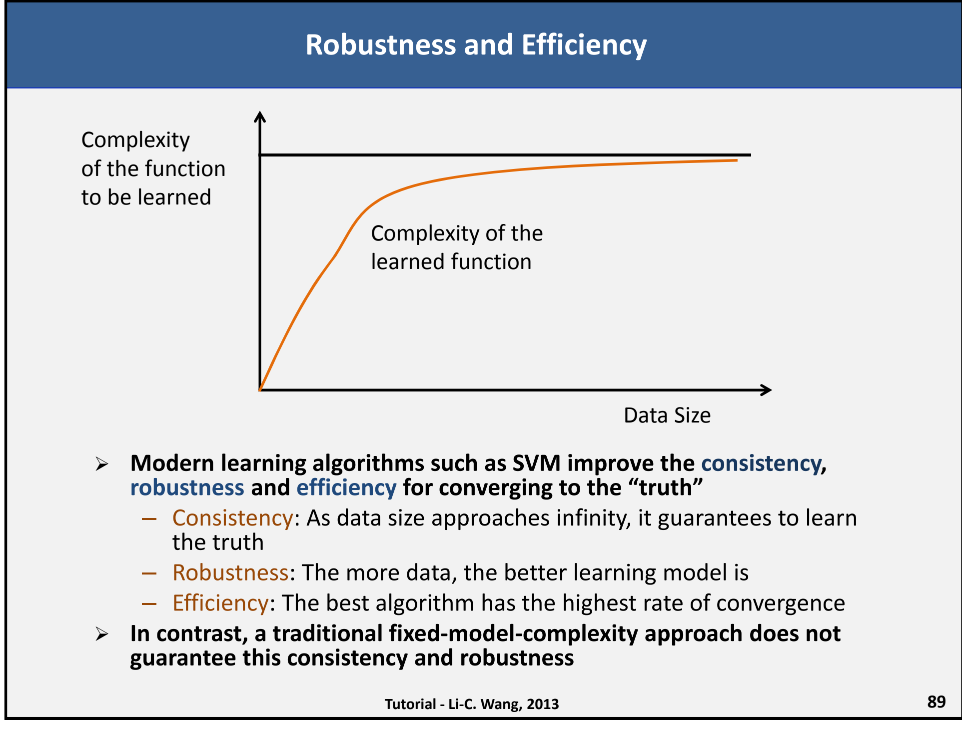
Model Complexity

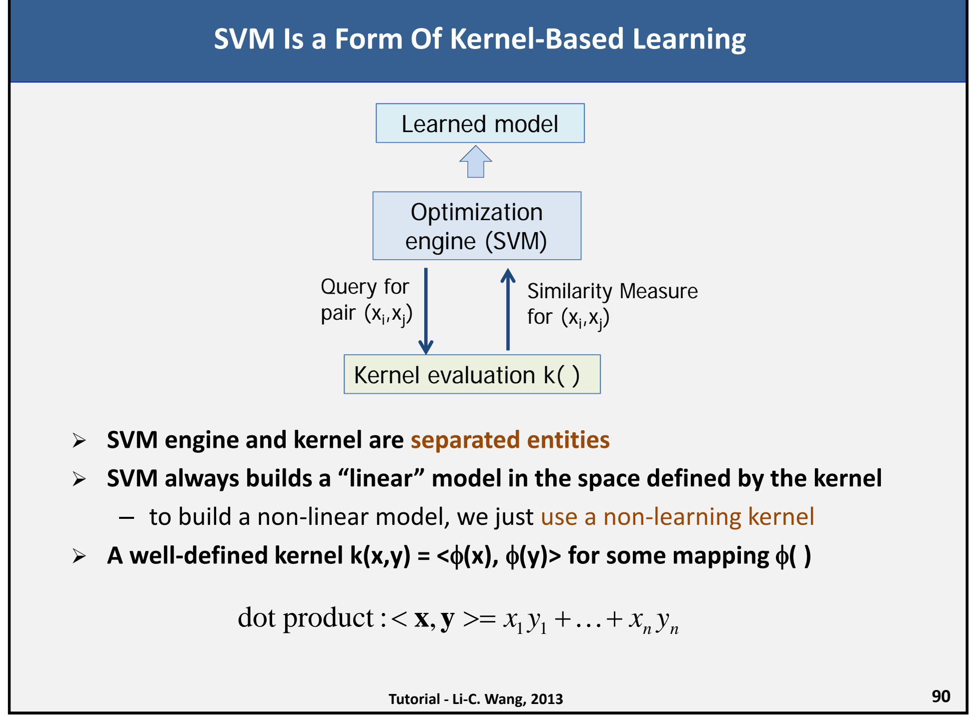
$$\mathbf{X} = \begin{vmatrix} \vec{x}_1 \\ \vec{x}_2 \\ \cdots \\ \vec{x}_m \end{vmatrix} = \begin{vmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{vmatrix} \quad \vec{\alpha} = \begin{vmatrix}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

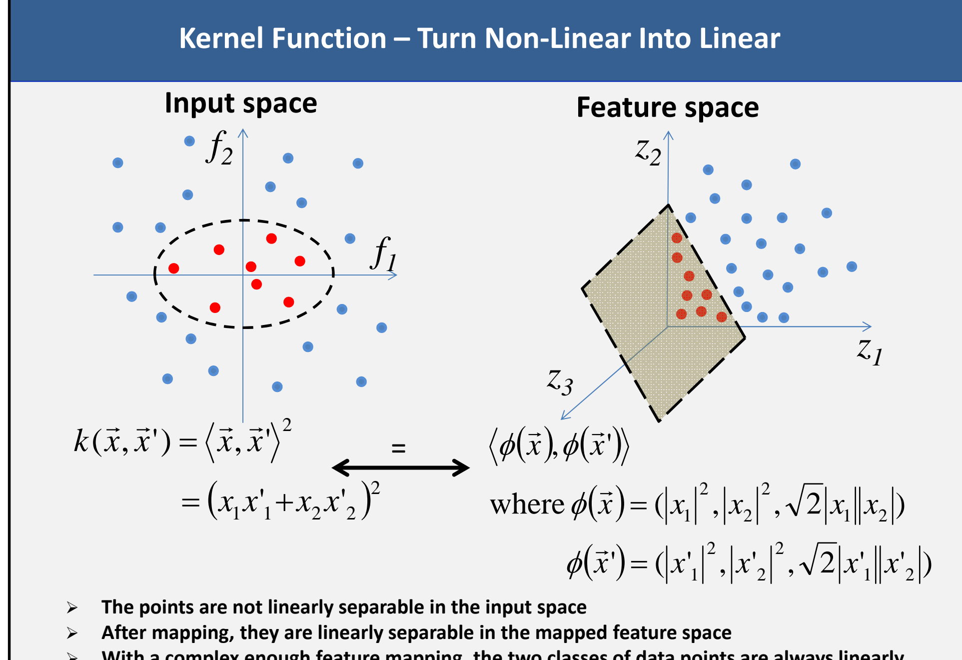
Model complexity $\propto \ (\alpha_1 + \ldots + \alpha_m)$

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

Tutorial - Li-C. Wang, 2013 88


Robustness and Efficiency
Complexity
of the function
to be learned
Complexity of the
learned function
Data Size
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
Tutorial - Li-C. Wang, 2013
89


SVM Is a Form Of Kernel-Based Learning
Learned model
Optimization
engine (SVM)
Query for
pair (xi,xj)
Similarity Measure
for (xi,xj)
Kernel evaluation k( )
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( )
dot product : < x, y >= x1y1 + … + xnyn
Tutorial - Li-C. Wang, 2013
90



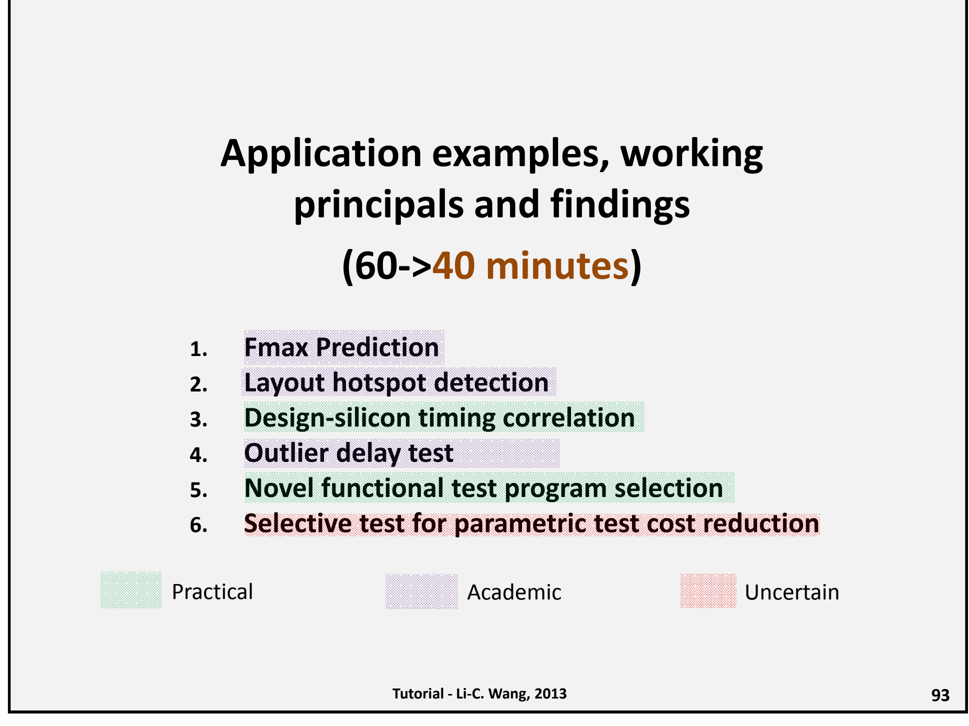
- **The points are not linearly separable in the input space**
- **After mapping, they are linearly separable in the mapped feature space**
- **With a complex enough feature mapping, the two classes of data points are always linearly separable**

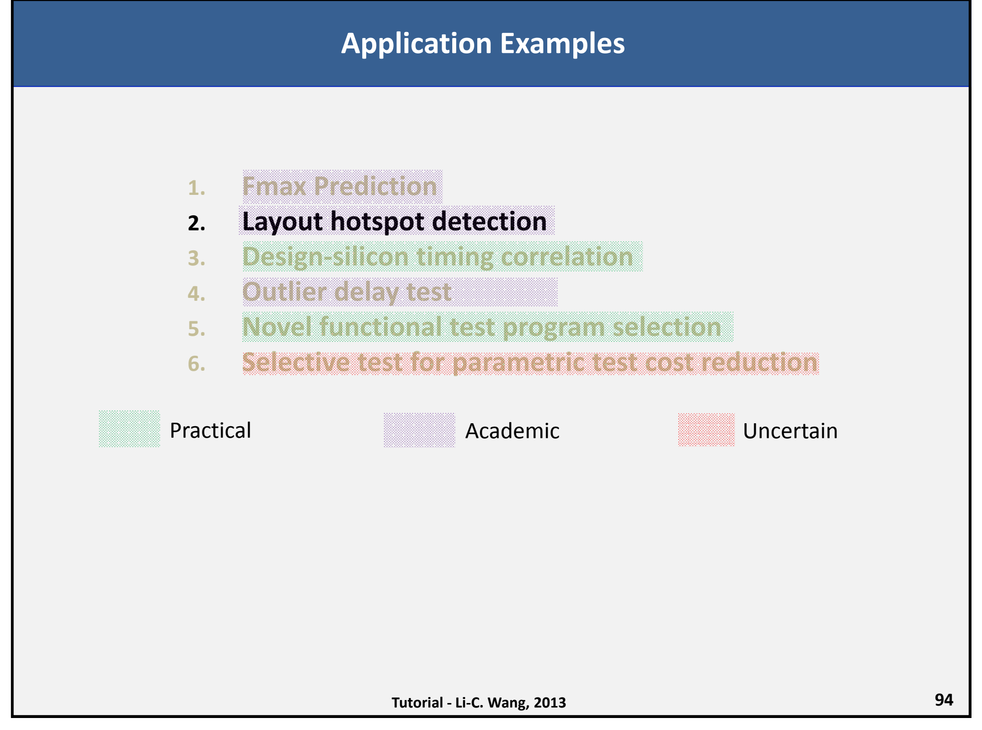
Tutorial - Li-C. Wang, 2013 91

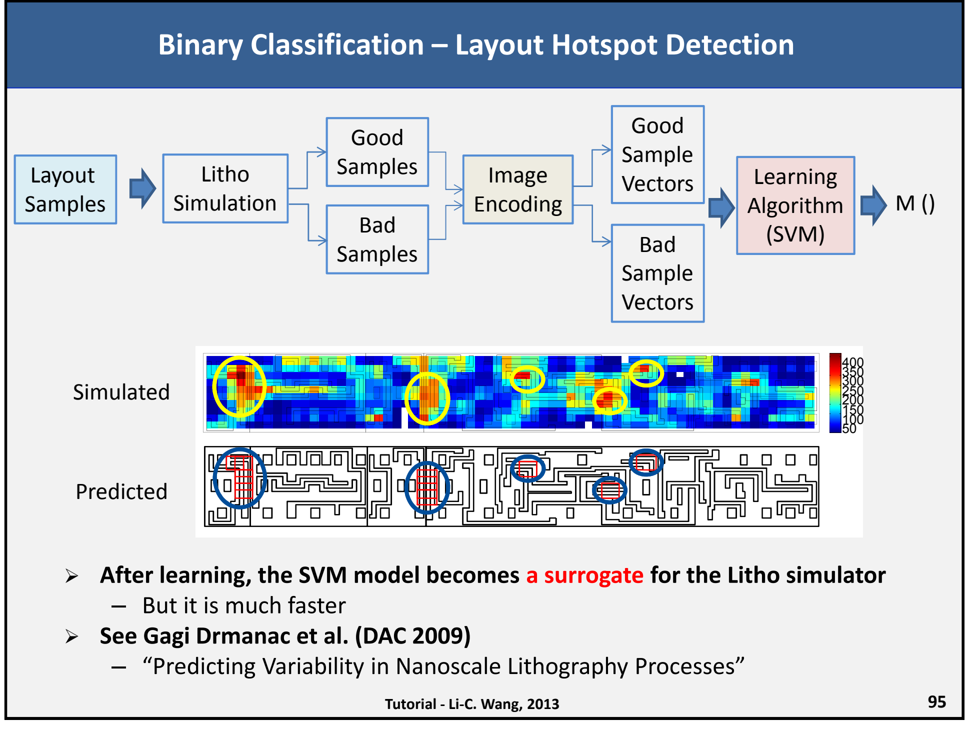
Bayesian SVM

- **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

Tutorial - Li-C. Wang, 2013 92


Application examples, working principals and findings
(60->40 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
Tutorial - Li-C. Wang, 2013
93

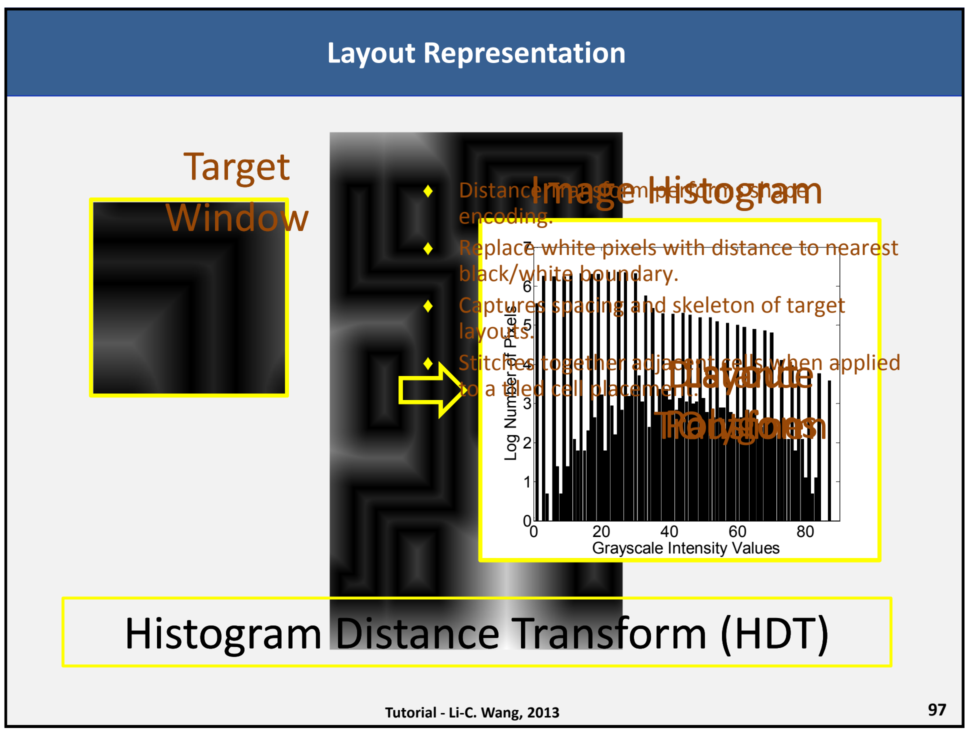

Application Examples
1. Fmax Prediction
2. Layout hotspot detection
3. Design-silicon timing correlation
4. Outlier delay test
5. Novel functional test program selection
6. Selective test for parametric test cost reduction
Practical
Academic
Uncertain
Tutorial - Li-C. Wang, 2013
94

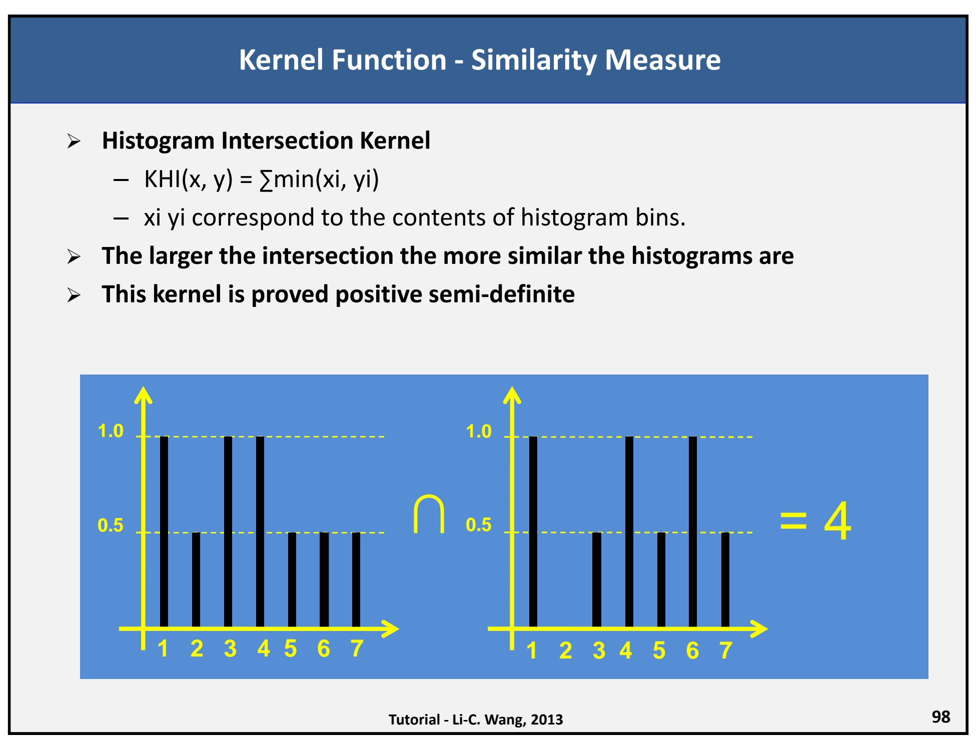


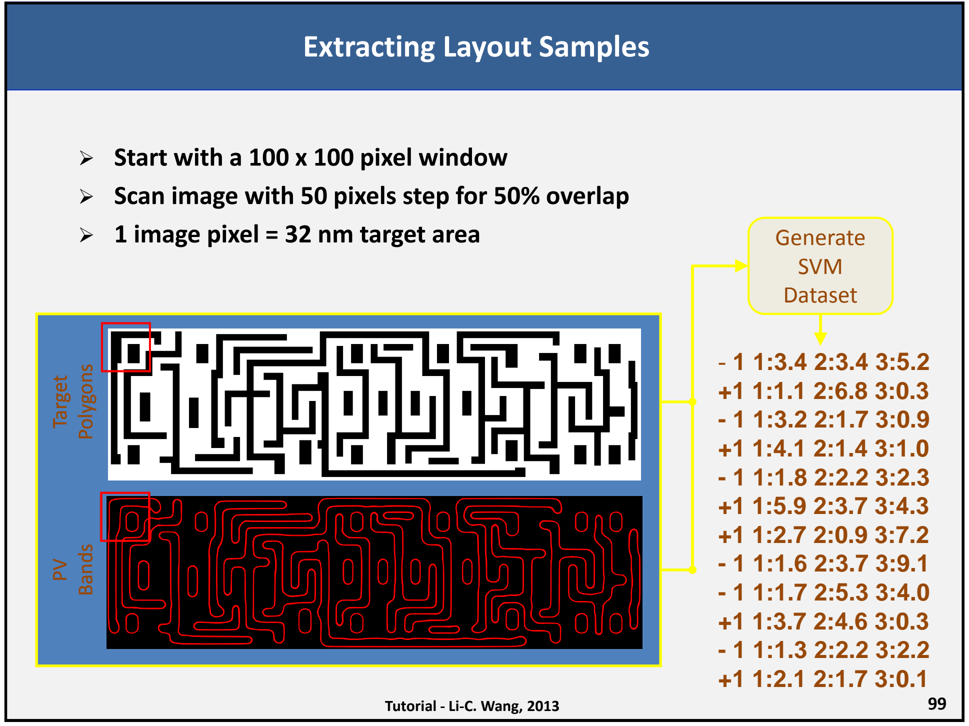
Two Fundamental Issues

- **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**

Tutorial - Li-C. Wang, 2013 96


Layout Representation
Target Window
Image Histogram
Distance transform shape encoding.
Replace white pixels with distance to nearest black/white boundary.
Captures spacing and skeleton of target layouts.
Stitches together adjacent cells when applied to a tiled cell placement.
Log Number of Pixels
Grayscale Intensity Values
Histogram Distance Transform (HDT)
Tutorial - Li-C. Wang, 2013
97

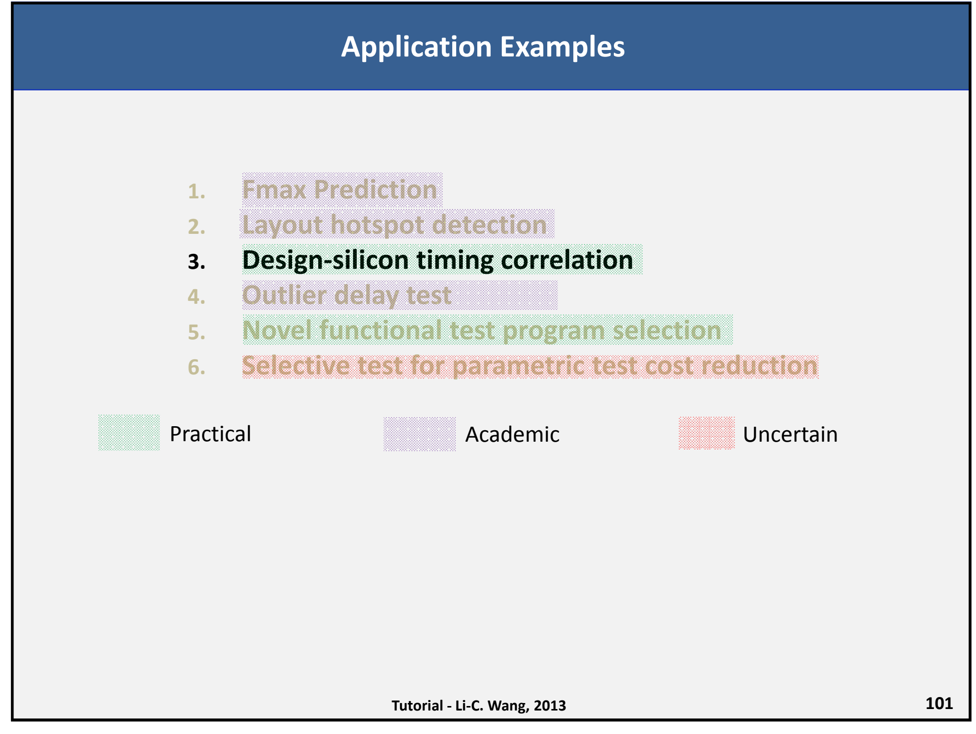

Kernel Function - Similarity Measure
Histogram Intersection Kernel
– KHI(x, y) = ∑min(xi, yi)
– xi yi correspond to the contents of histogram bins.
The larger the intersection the more similar the histograms are
This kernel is proved positive semi-definite
1.0
0.5
1 2 3 4 5 6 7
∩
= 4
Tutorial - Li-C. Wang, 2013
98

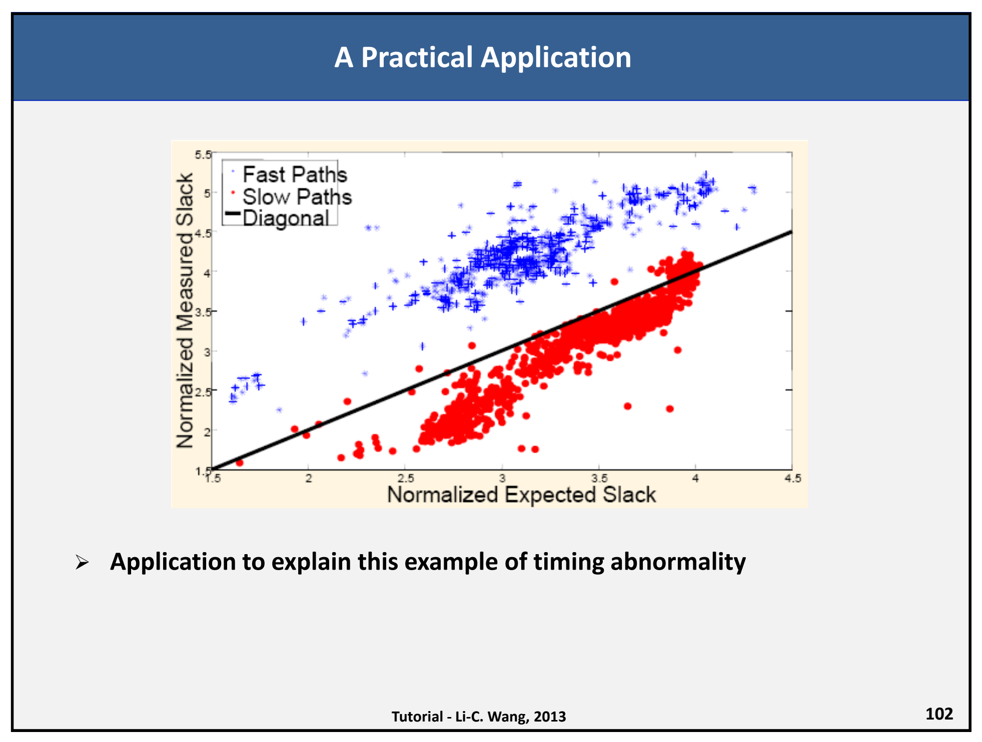


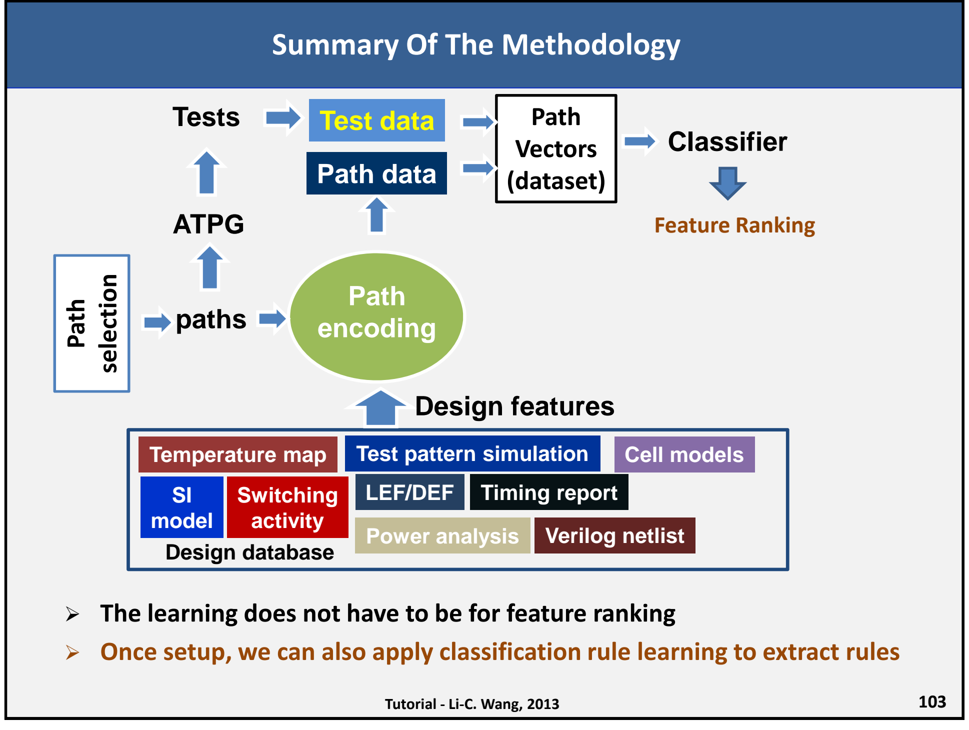
Challenges

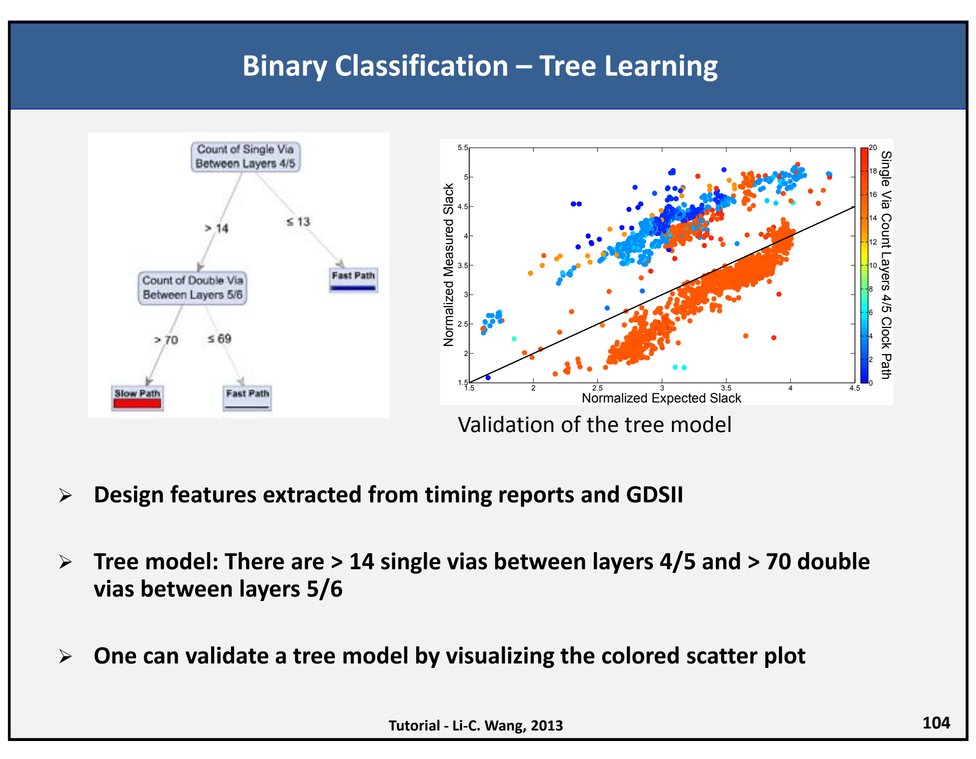
- **The work was discontinued because**
  - Not sure if it provide either **accuracy** and/or **speed** benefit to the rule-based approach
  - Or, learning should be used to extract rules, not just a prediction model
  - It should be applicable to the next technology node – difficult to obtain data

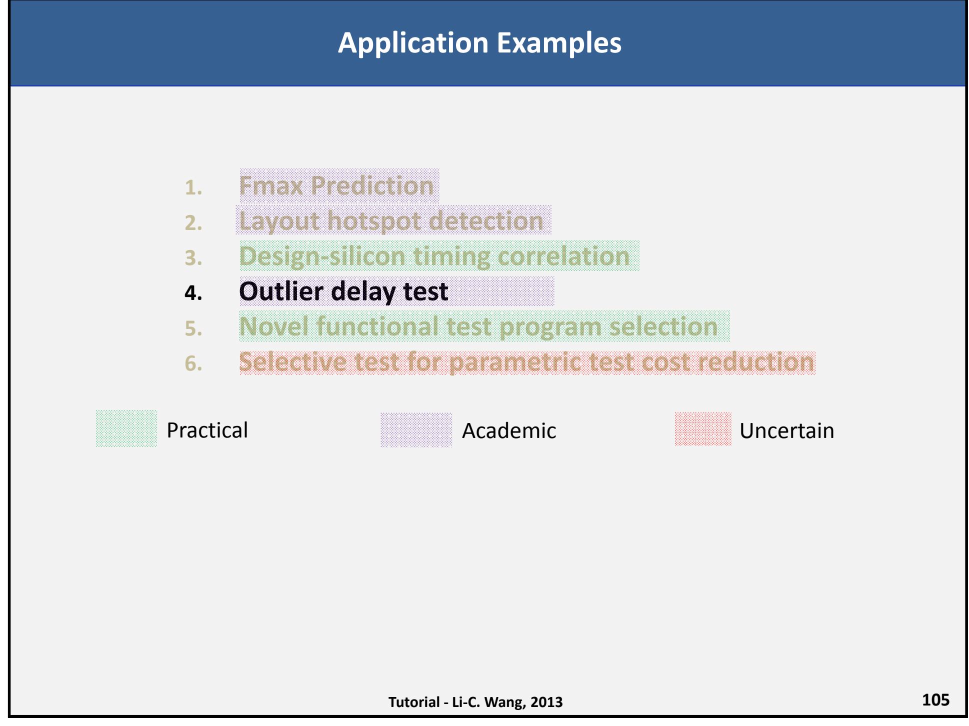
Tutorial - Li-C. Wang, 2013 100

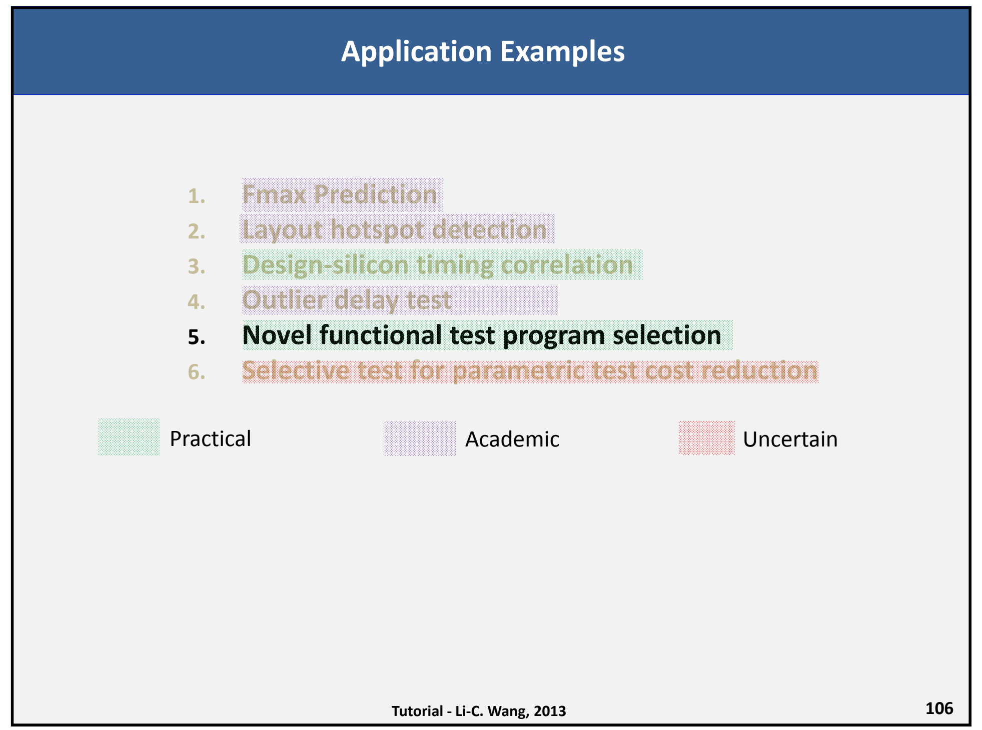

Application Examples
1. Fmax Prediction
2. Layout hotspot detection
3. Design-silicon timing correlation
4. Outlier delay test
5. Novel functional test program selection
6. Selective test for parametric test cost reduction
Practical
Academic
Uncertain
Tutorial - Li-C. Wang, 2013
101

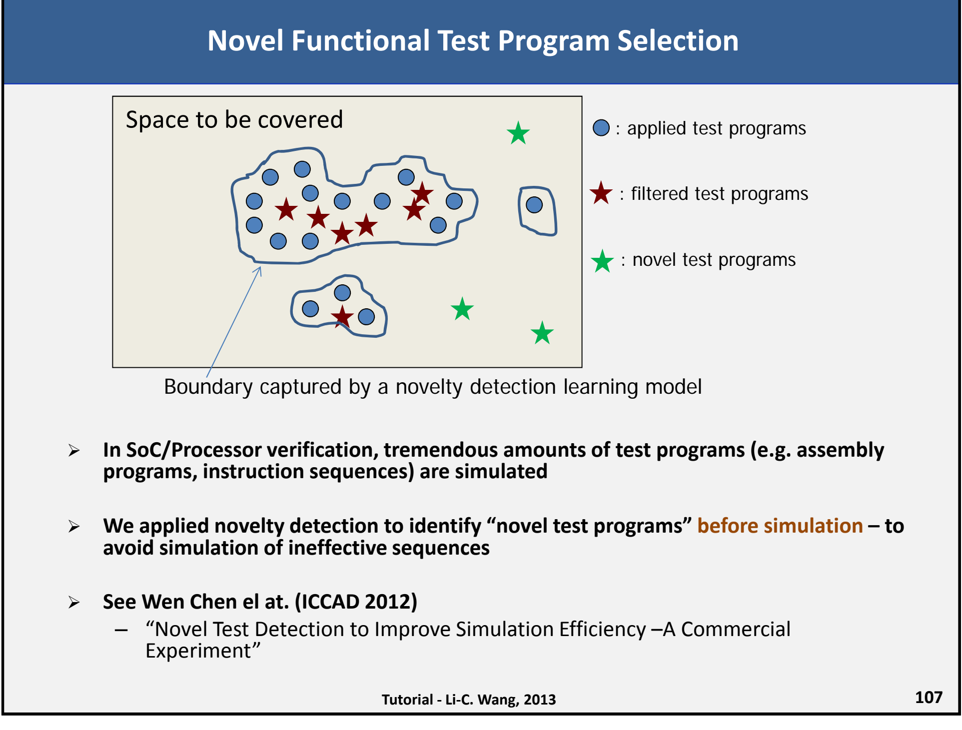

A Practical Application
Fast Paths
Slow Paths
Diagonal
Normalized Measured Slack
Normalized Expected Slack
Application to explain this example of timing abnormality
Tutorial - Li-C. Wang, 2013
102

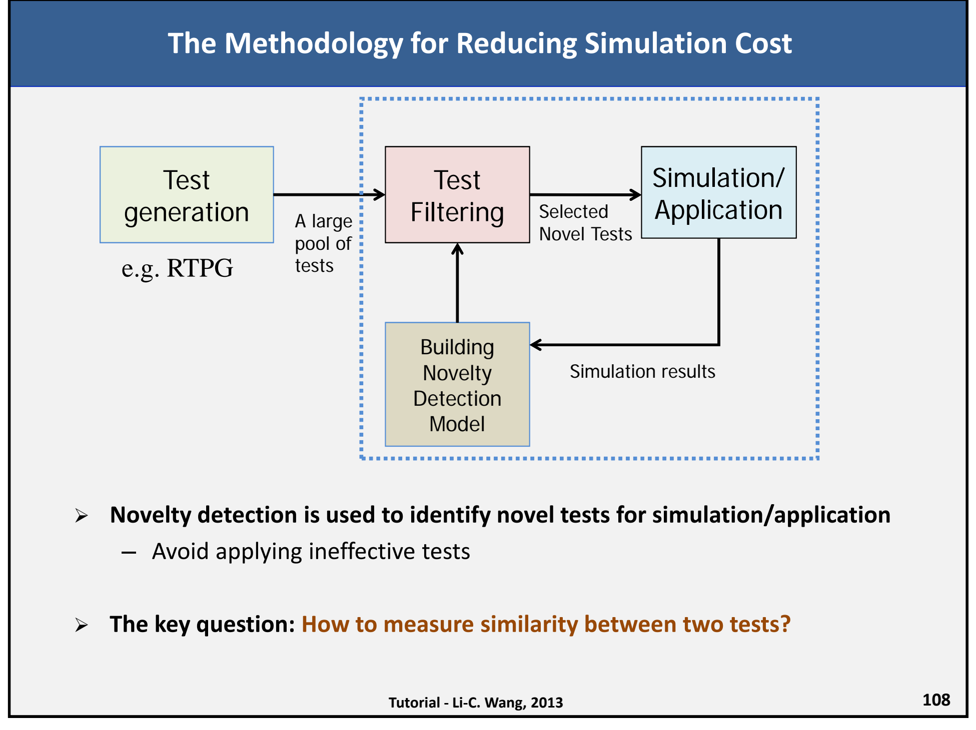

Summary Of The Methodology
Tests
Test data
Path data
Path
Vectors
(dataset)
Classifier
Feature Ranking
ATPG
Path
selection
paths
Path
encoding
Design features
Temperature map
Test pattern simulation
Cell models
SI
model
Switching
activity
LEF/DEF
Timing report
Power analysis
Verilog netlist
Design database
The learning does not have to be for feature ranking
Once setup, we can also apply classification rule learning to extract rules
Tutorial - Li-C. Wang, 2013
103

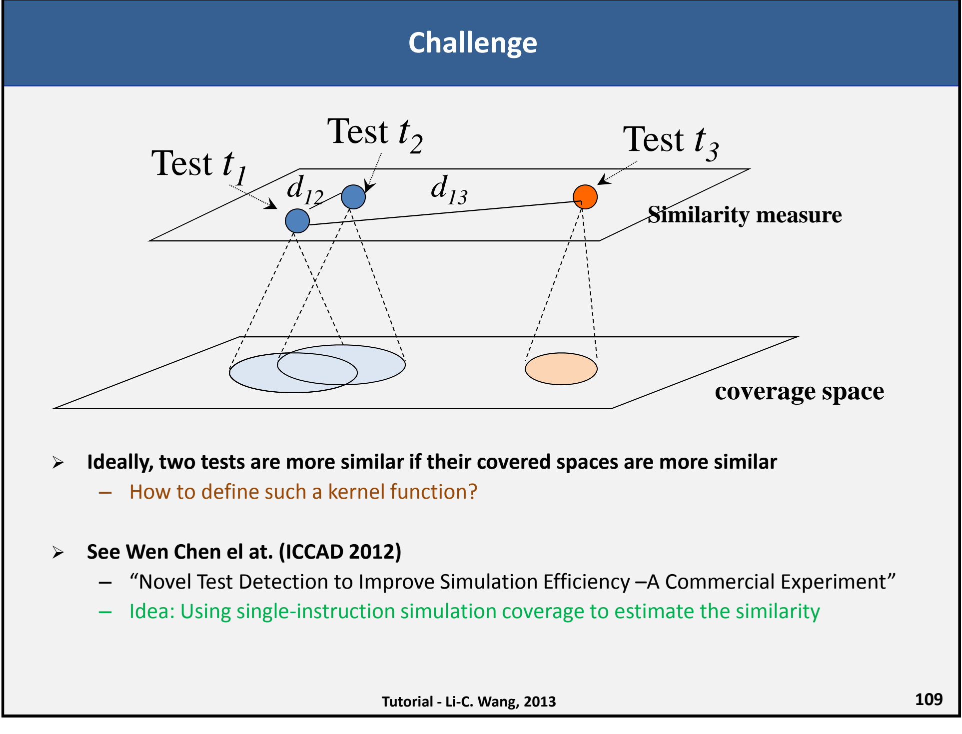

Binary Classification – Tree Learning
Count of Single Via
Between Layers 4/5
> 14
≤ 13
Fast Path
Count of Double Via
Between Layers 5/6
> 70
≤ 69
Slow Path
Fast Path
Normalized Measured Slack
Normalized Expected Slack
Single Via Count Layers 4/5 Clock Path
Validation of the tree model
Design features 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
Tutorial - Li-C. Wang, 2013
104

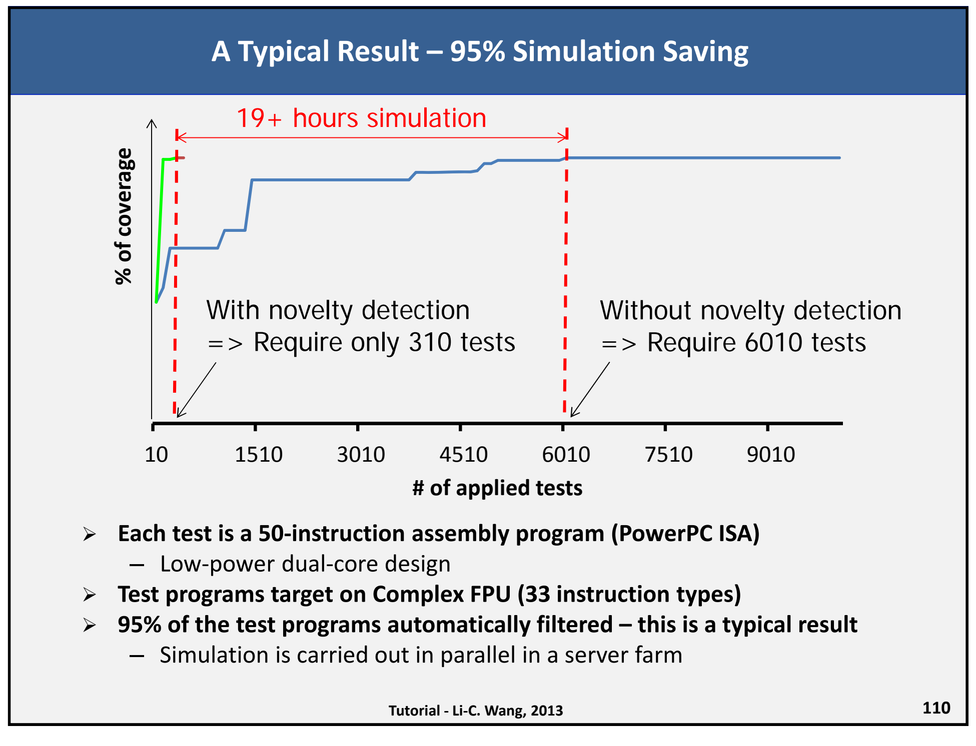

Application Examples
1. Fmax Prediction
2. Layout hotspot detection
3. Design-silicon timing correlation
4. Outlier delay test
5. Novel functional test program selection
6. Selective test for parametric test cost reduction
Practical
Academic
Uncertain
Tutorial - Li-C. Wang, 2013
105

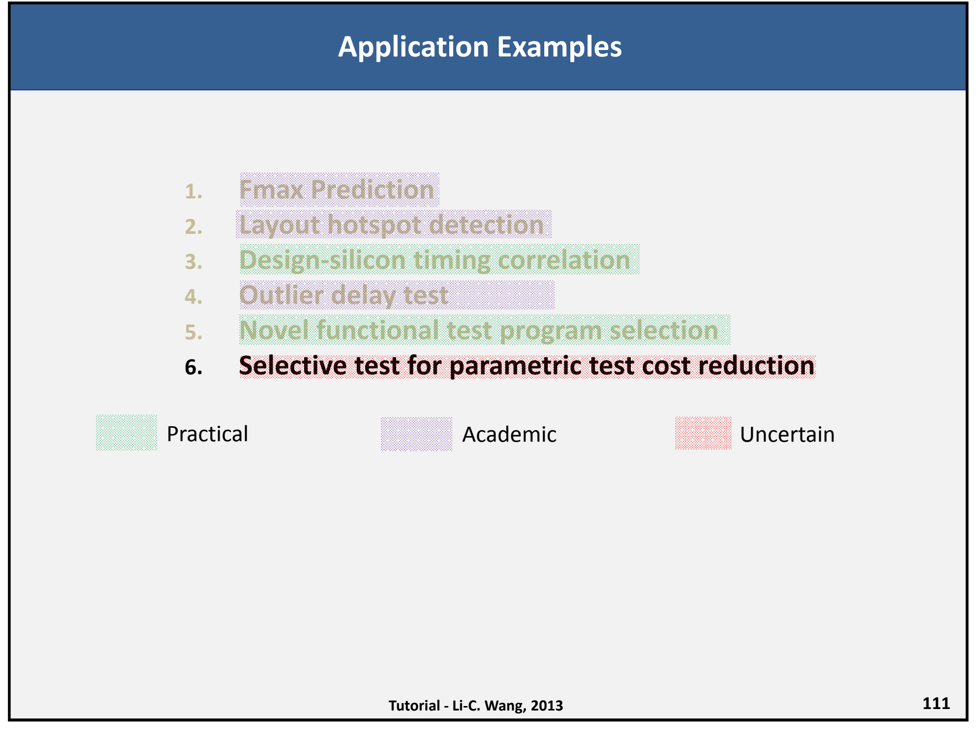

Application Examples
1. Fmax Prediction
2. Layout hotspot detection
3. Design-silicon timing correlation
4. Outlier delay test
5. Novel functional test program selection
6. Selective test for parametric test cost reduction
Practical
Academic
Uncertain
Tutorial - Li-C. Wang, 2013
106

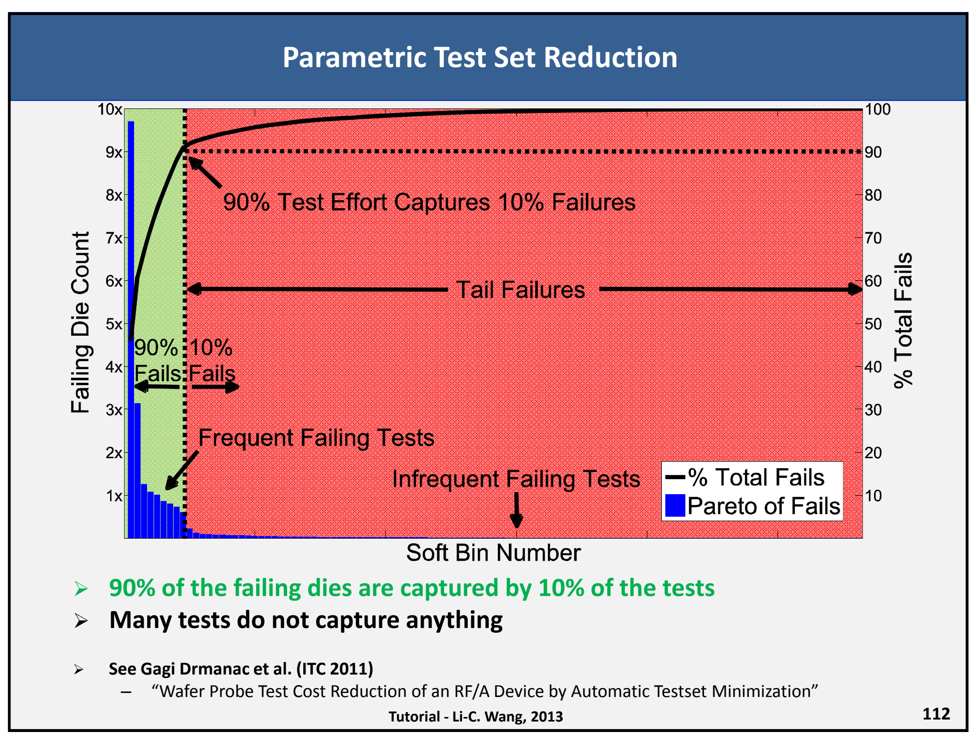

Novel Functional Test Program Selection
Space to be covered
: applied test programs
: filtered test programs
: novel test programs
Boundary captured by a novelty detection learning model
In SoC/Processor verification, tremendous amounts of test programs (e.g. assembly programs, instruction sequences) are simulated
We applied novelty detection to identify "novel test programs" before simulation – to avoid simulation of ineffective sequences
See Wen Chen el at. (ICCAD 2012)
"Novel Test Detection to Improve Simulation Efficiency –A Commercial Experiment"
Tutorial - Li-C. Wang, 2013
107

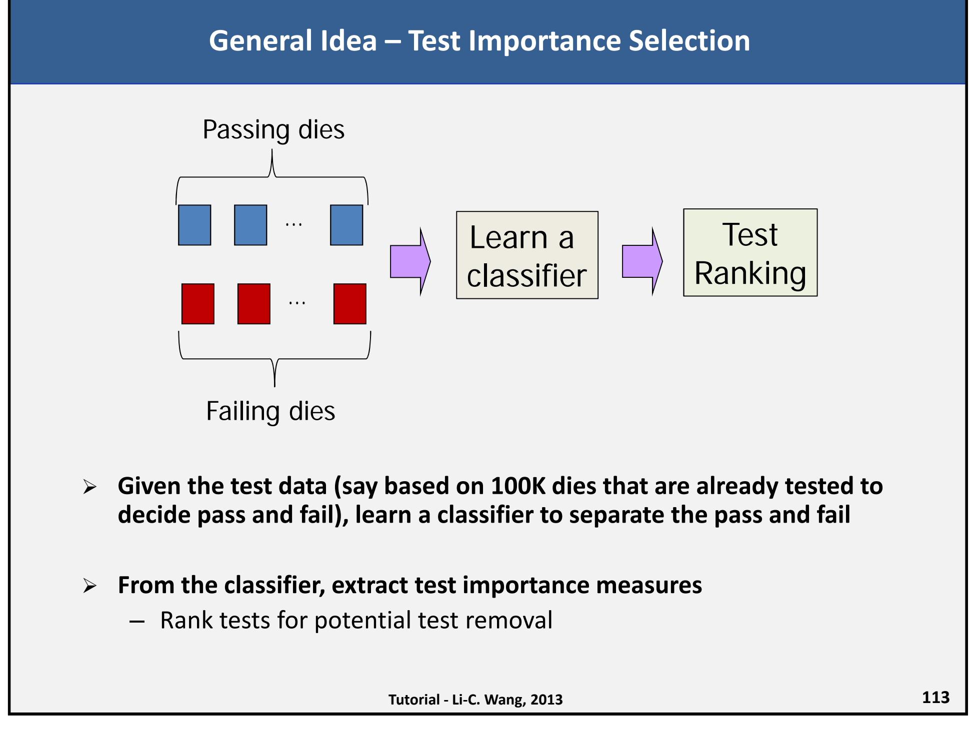

The Methodology for Reducing Simulation Cost
Test generation
e.g. RTPG
A large pool of tests
Test Filtering
Selected Novel Tests
Simulation/ Application
Building Novelty Detection Model
Simulation results
Novelty detection is used to identify novel tests for simulation/application
Avoid applying ineffective tests
The key question: How to measure similarity between two tests?
Tutorial - Li-C. Wang, 2013
108


Challenge
Test t_1
Test t_2
Test t_3
d_{12}
d_{13}
Similarity measure
coverage space
Ideally, two tests are more similar if their covered spaces are more similar
– How to define such a kernel function?
See Wen Chen el at. (ICCAD 2012)
– "Novel Test Detection to Improve Simulation Efficiency –A Commercial Experiment"
– Idea: Using single-instruction simulation coverage to estimate the similarity
Tutorial - Li-C. Wang, 2013
109


A Typical Result – 95% Simulation Saving
19+ hours simulation
% of coverage
With novelty detection
=> Require only 310 tests
Without novelty detection
=> Require 6010 tests
10
1510
3010
4510
6010
7510
9010
# of applied tests
Each test is a 50-instruction assembly program (PowerPC ISA)
– Low-power dual-core design
Test programs target on Complex FPU (33 instruction types)
95% of the test programs automatically filtered – this is a typical result
– Simulation is carried out in parallel in a server farm
Tutorial - Li-C. Wang, 2013
110


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
Tutorial - Li-C. Wang, 2013
111

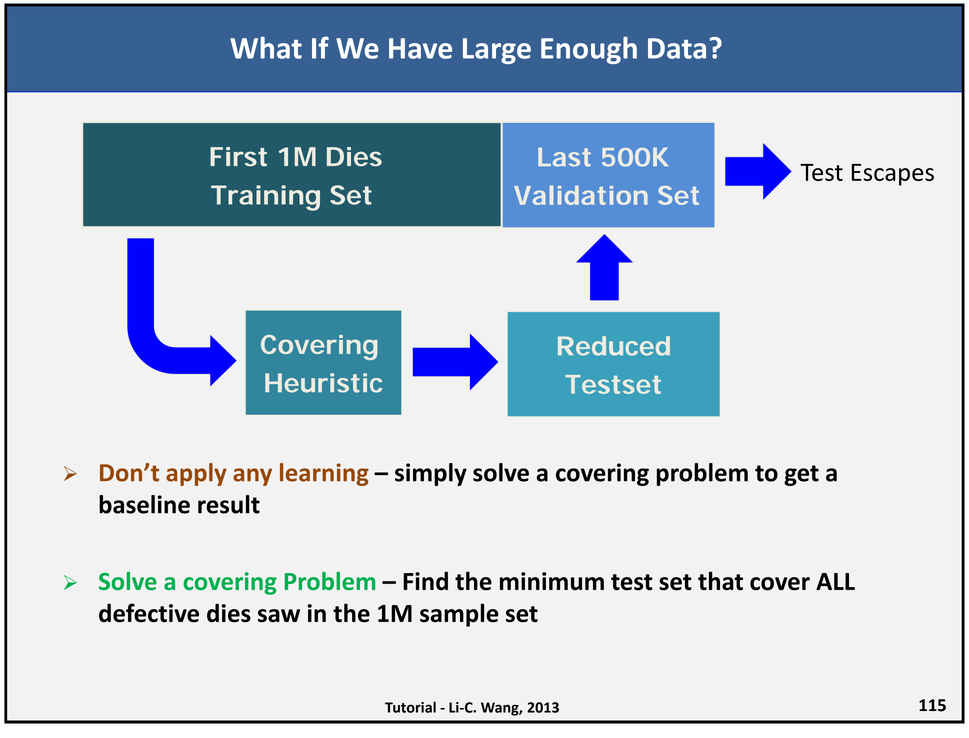

Parametric Test Set Reduction
90% Test Effort Captures 10% Failures
Tail Failures
90% Fails
10% Fails
Frequent Failing Tests
Infrequent Failing Tests
% Total Fails
Pareto of Fails
Failing Die Count
% Total Fails
Soft Bin Number
10x
9x
8x
7x
6x
5x
4x
3x
2x
1x
100
90
80
70
60
50
40
30
20
10
90% of the failing dies are captured by 10% of the tests
Many tests do not capture anything
See Gagi Drmanac et al. (ITC 2011)
– "Wafer Probe Test Cost Reduction of an RF/A Device by Automatic Testset Minimization"
Tutorial - Li-C. Wang, 2013
112

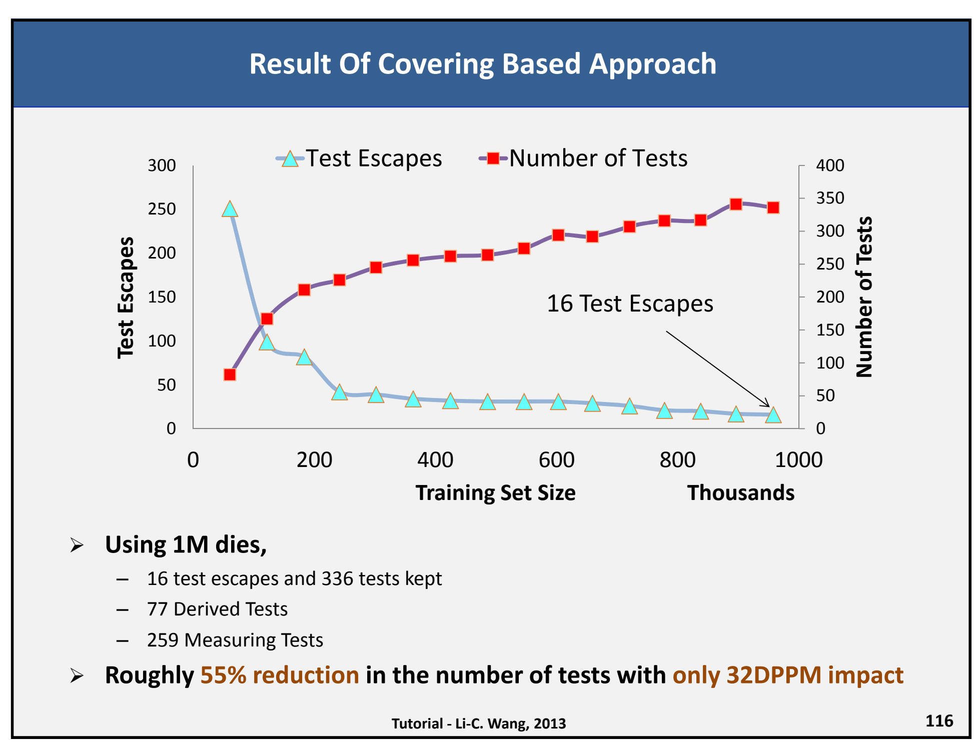


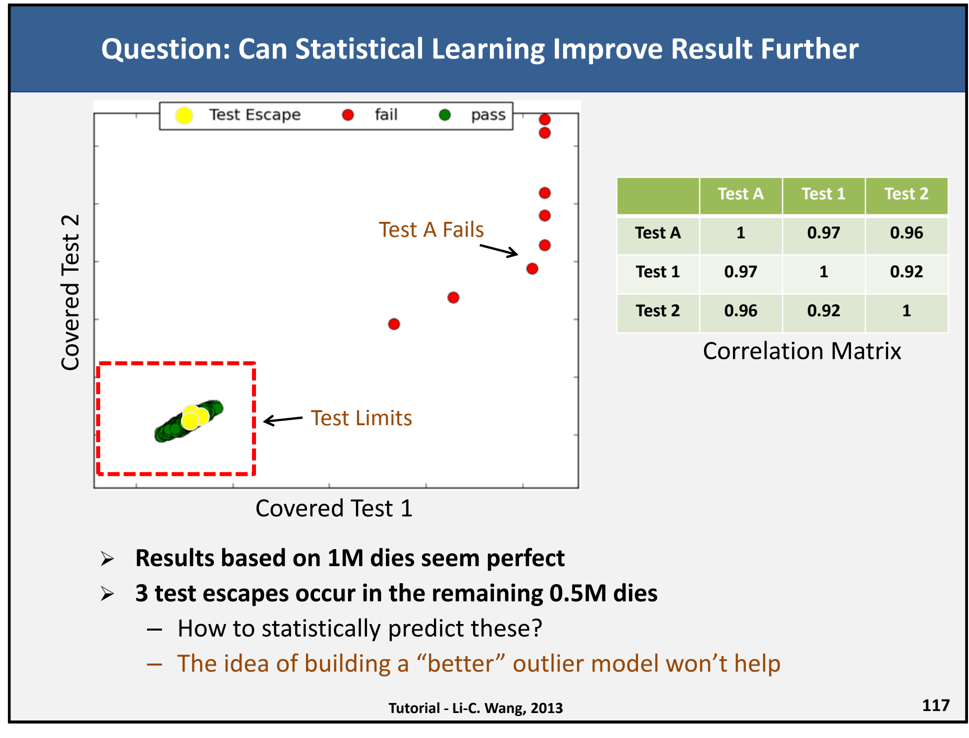
Some Result

- **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"

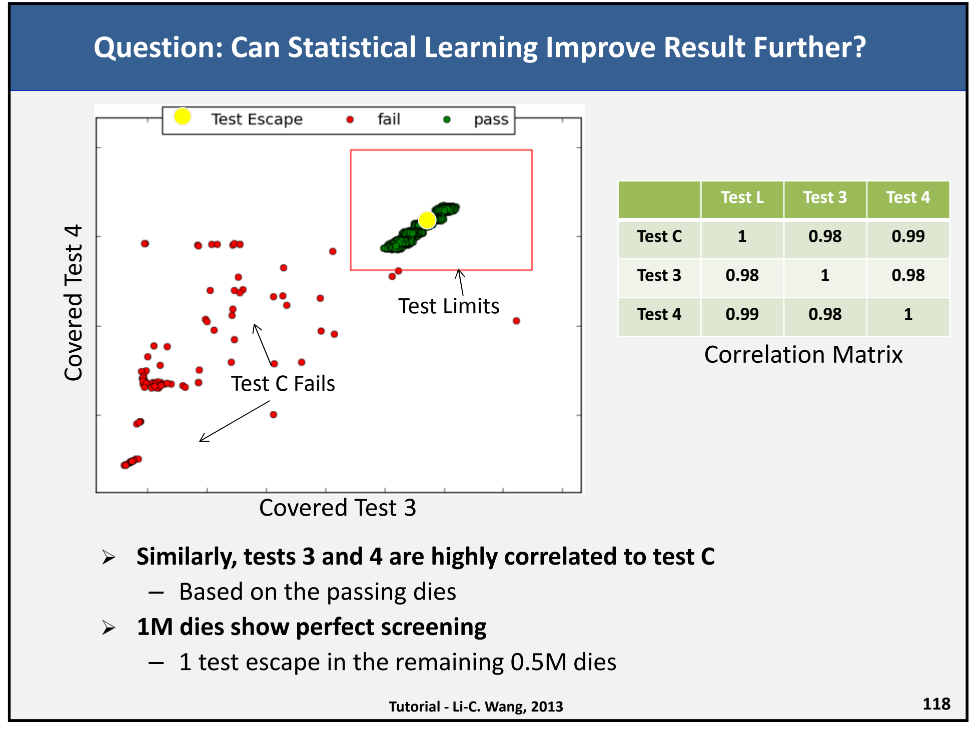
Tutorial - Li-C. Wang, 2013 114


What If We Have Large Enough Data?
First 1M Dies
Training Set
Last 500K
Validation Set
Test Escapes
Covering
Heuristic
Reduced
Testset
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
Tutorial - Li-C. Wang, 2013
115


Result Of Covering Based Approach
Test Escapes
Number of Tests
16 Test Escapes
Training Set Size
Thousands
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
Tutorial - Li-C. Wang, 2013
116



	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

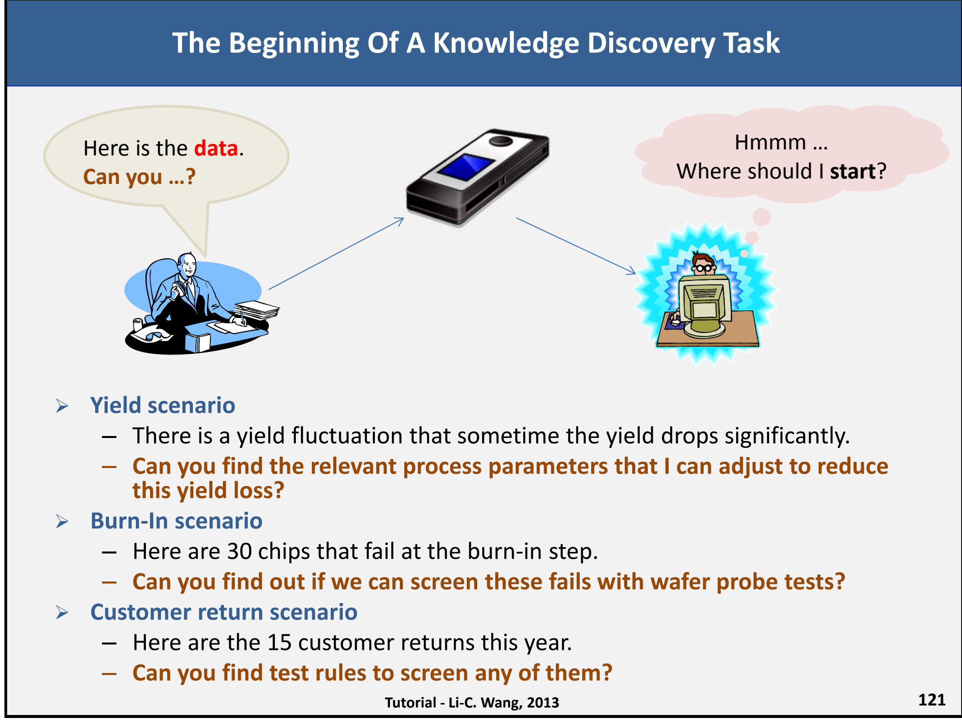


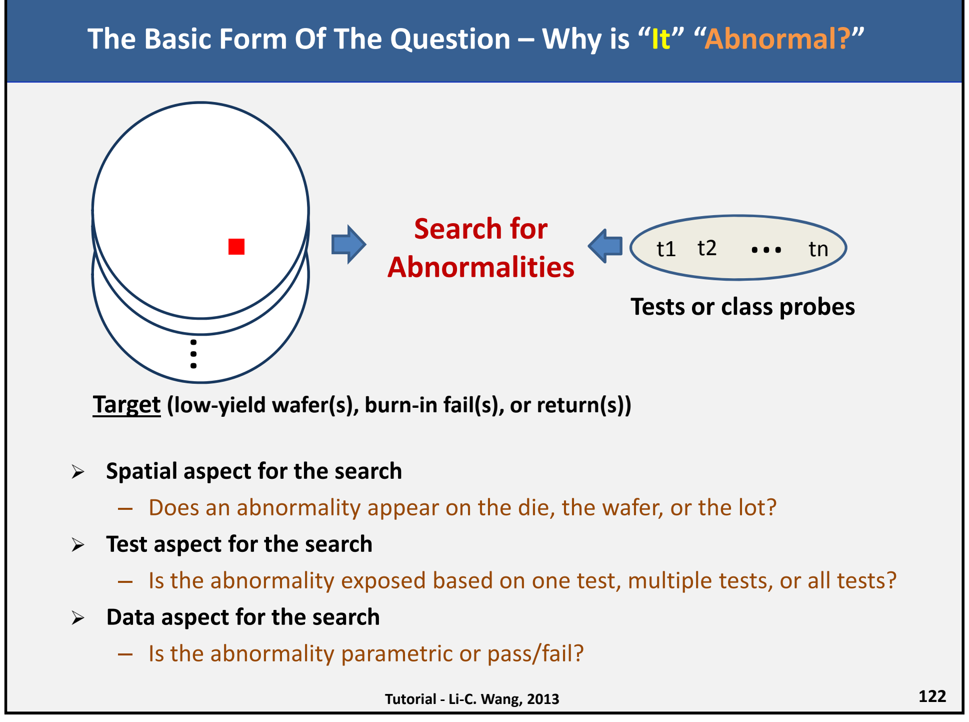
	Test L	Test 3	Test 4
Test C	1	0.98	0.99
Test 3	0.98	1	0.98
Test 4	0.99	0.98	1

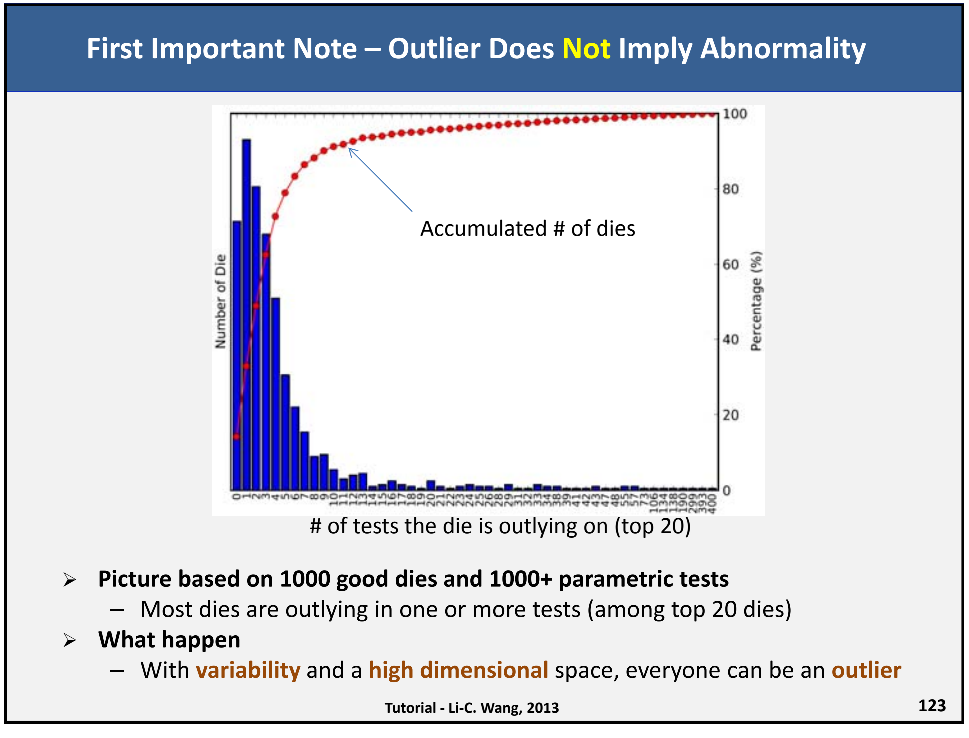
Lessons Learned

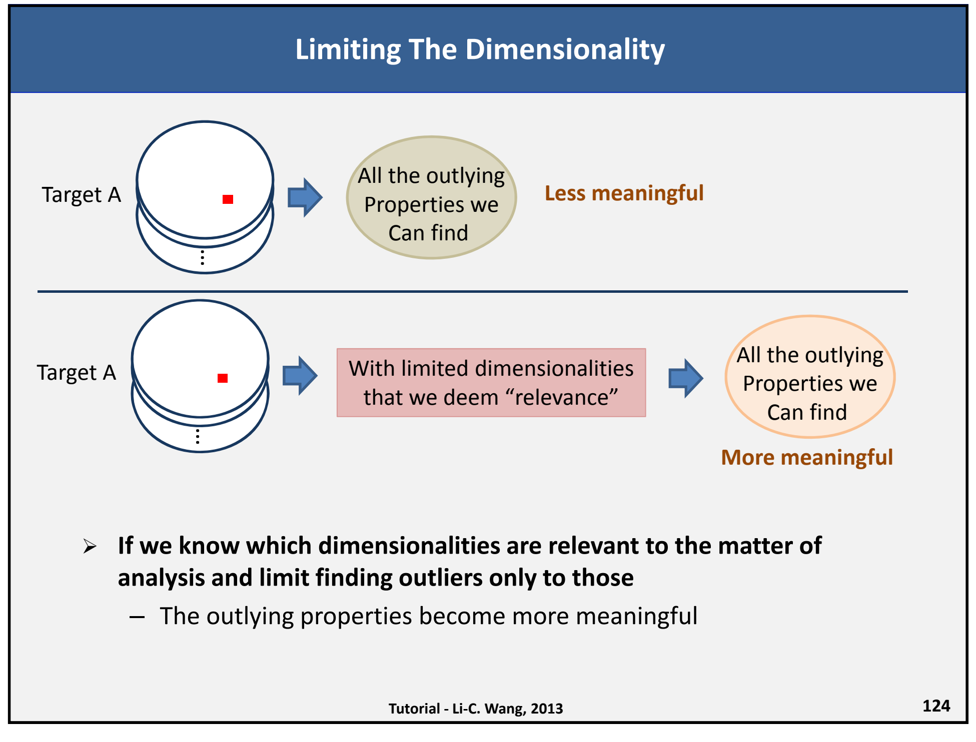
- **A statistical learning approach tries to generalize beyond what it sees in a given dataset**
  - That should be why a statistical approach is better than a simple covering approach that only tries to fit the given data
- **However, even though a statistical approach gives good result, the approach 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**

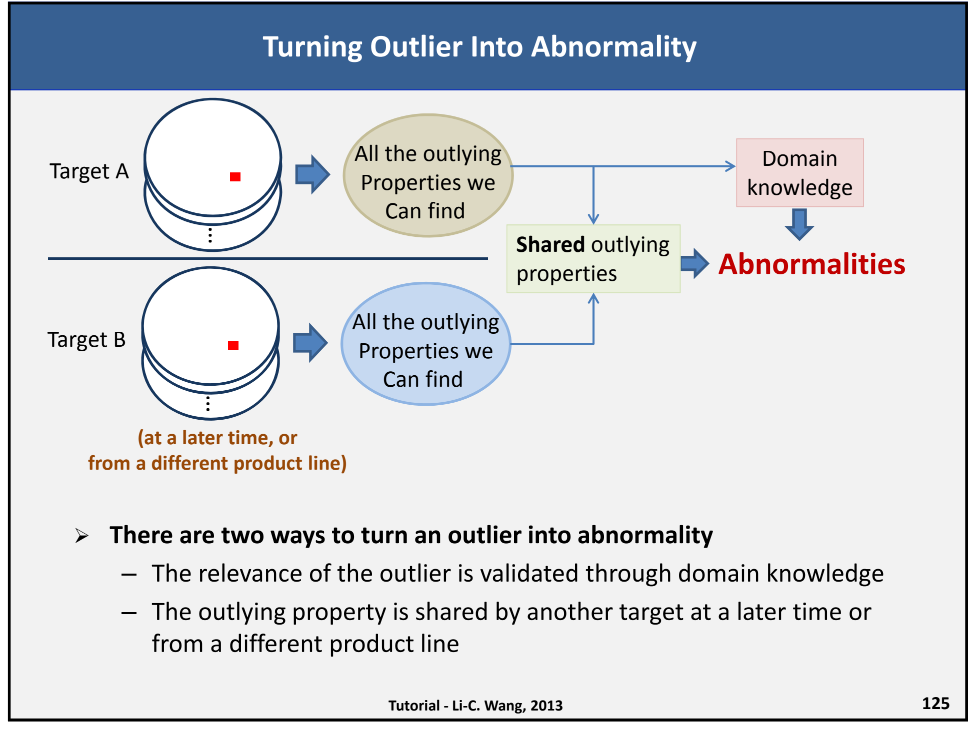
Knowledge Discovery in Test Applications (60->30 minutes)

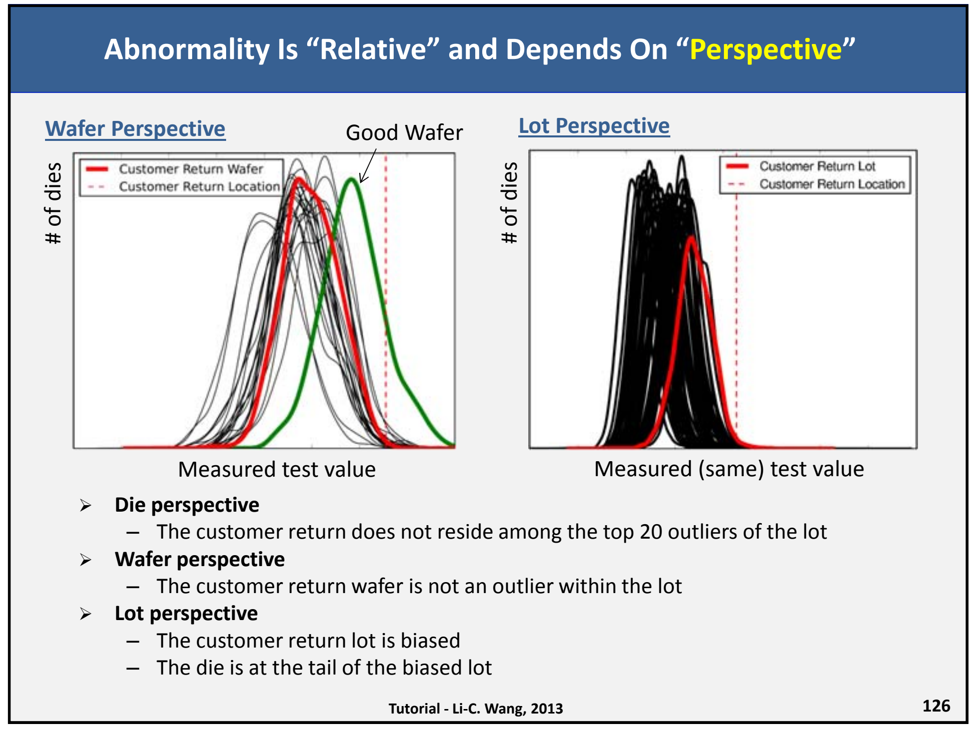

The Beginning Of A Knowledge Discovery Task
Here is the data.
Can you …?
Hmmm …
Where should I start?
Yield scenario
There is a yield fluctuation that sometime the yield drops significantly.
Can you find the relevant process parameters that I can adjust to reduce this yield loss?
Burn-In scenario
Here 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?
Tutorial - Li-C. Wang, 2013
121

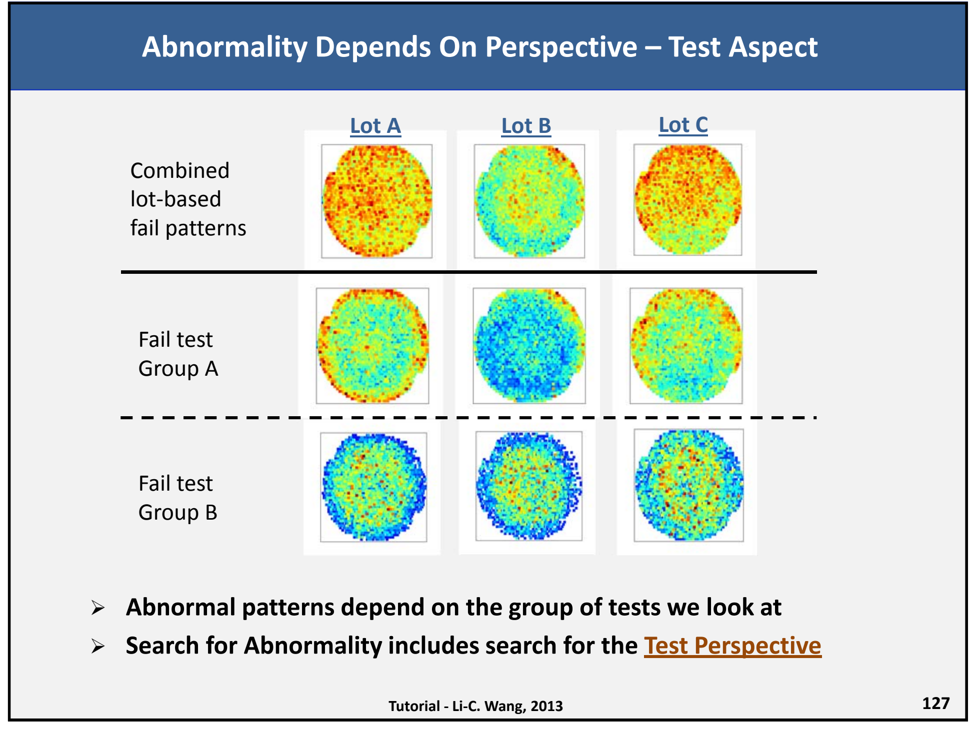

The Basic Form Of The Question – Why is "It" "Abnormal?"
Search for Abnormalities
t1 t2 ••• tn
Tests or class probes
Target (low-yield wafer(s), burn-in fail(s), or return(s))
Spatial aspect for the search
Does an abnormality appear on the die, the wafer, or the lot?
Test aspect for the search
Is the abnormality exposed based on one test, multiple tests, or all tests?
Data aspect for the search
Is the abnormality parametric or pass/fail?
Tutorial - Li-C. Wang, 2013
122

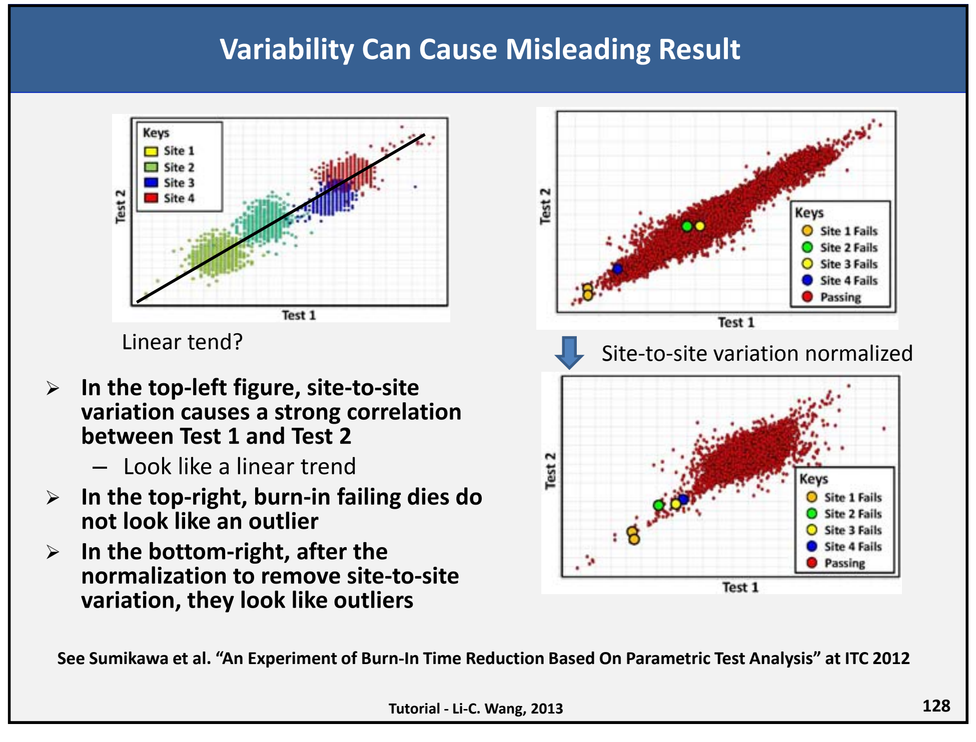

First Important Note – Outlier Does Not Imply Abnormality
Accumulated # of dies
Number of Die
Percentage (%)
# of tests the die is outlying on (top 20)
Picture based on 1000 good dies and 1000+ parametric tests
– Most dies are outlying in one or more tests (among top 20 dies)
What happen
– With variability and a high dimensional space, everyone can be an outlier
Tutorial - Li-C. Wang, 2013
123

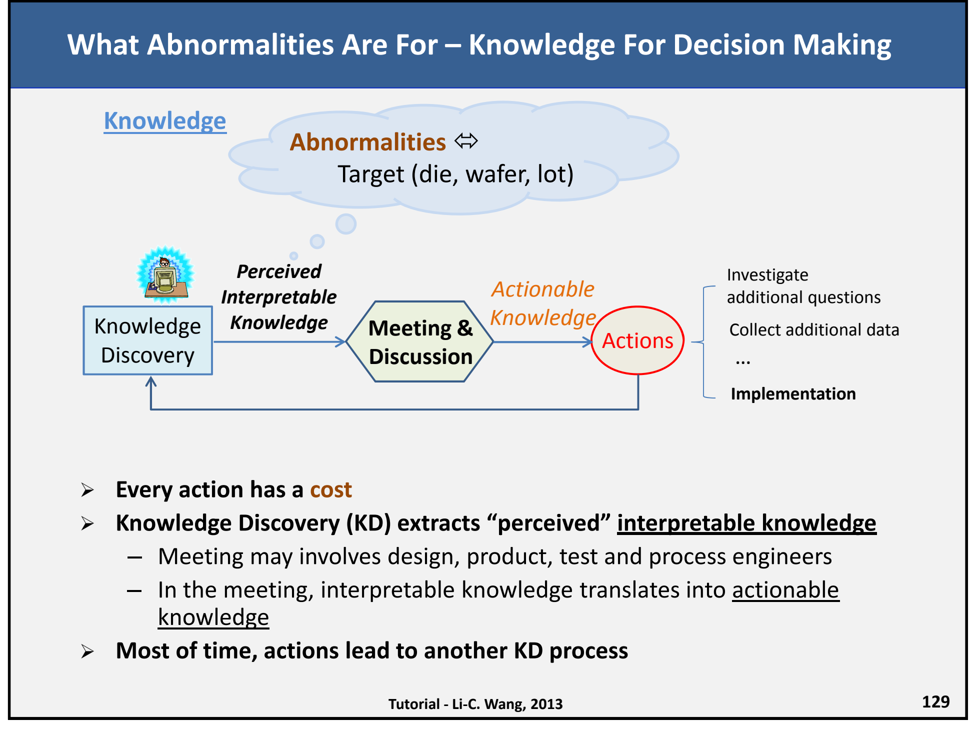

Limiting The Dimensionality
Target A
All the outlying Properties we Can find
Less meaningful
Target A
With limited dimensionalities that we deem "relevance"
All the outlying Properties we Can find
More meaningful
If we know which dimensionalities are relevant to the matter of analysis and limit finding outliers only to those
– The outlying properties become more meaningful
Tutorial - Li-C. Wang, 2013
124

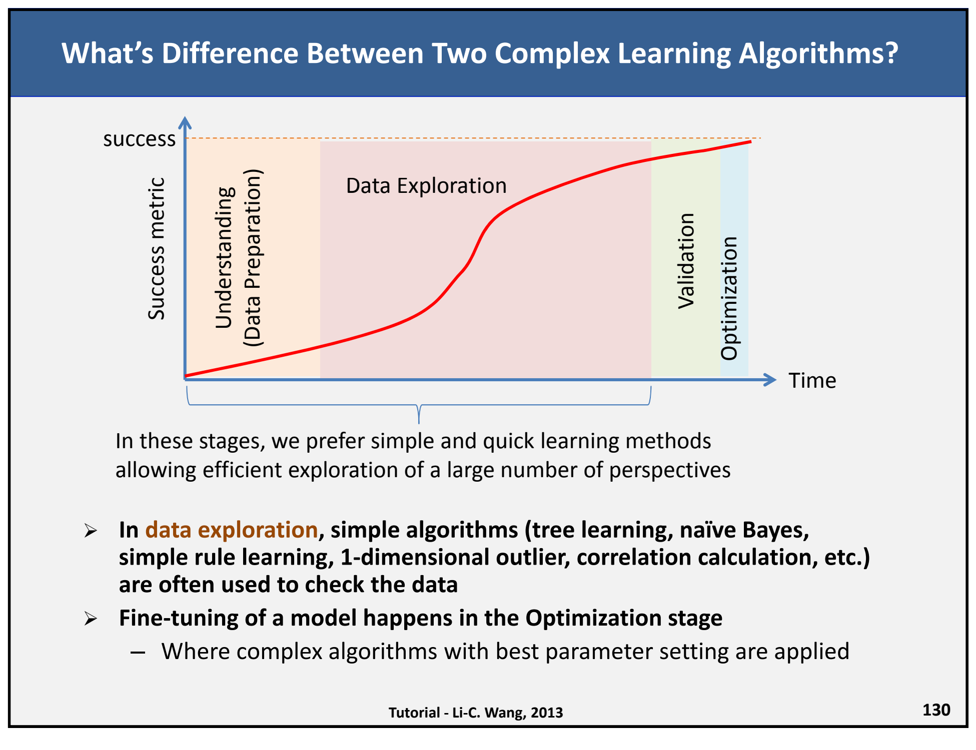

Turning Outlier Into Abnormality
Target A
All the outlying Properties we Can find
Domain knowledge
Shared outlying properties
Abnormalities
Target B
All the outlying Properties we Can find
(at a later time, or from a different product line)
There are two ways to turn an outlier into abnormality
The relevance of the outlier is validated through domain knowledge
The outlying property is shared by another target at a later time or from a different product line
Tutorial - Li-C. Wang, 2013
125

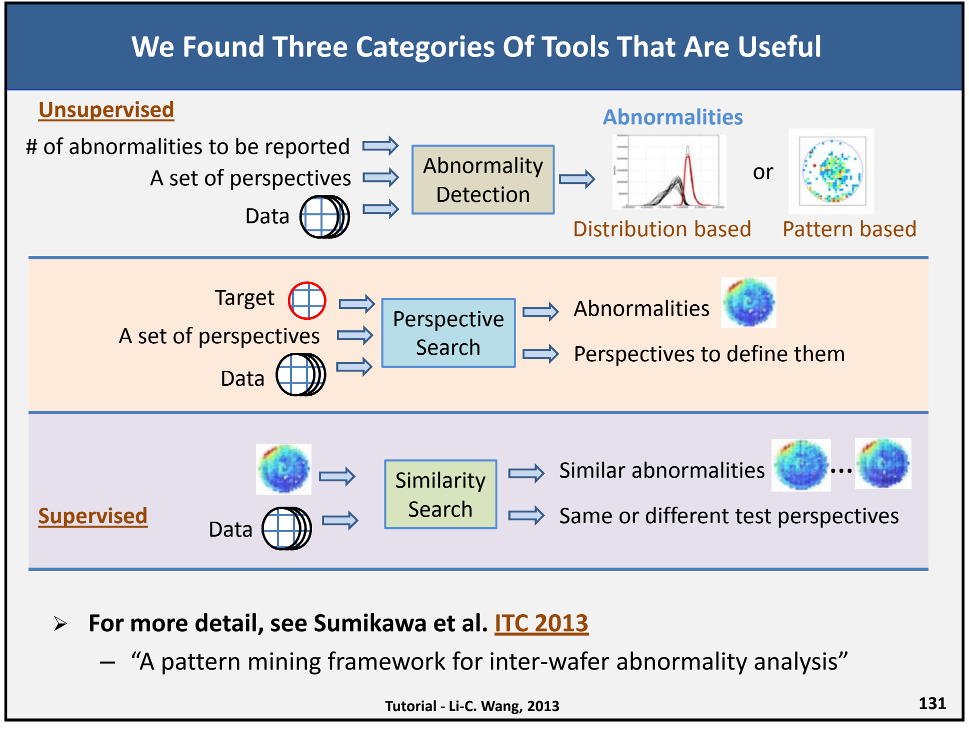

Abnormality Is "Relative" and Depends On "Perspective"
Wafer Perspective
Good Wafer
Customer Return Wafer
Customer Return Location
# of dies
Measured test value
Lot Perspective
Customer Return Lot
Customer Return Location
# of dies
Measured (same) test value
Die perspective
The customer return does not reside among the top 20 outliers of the lot
Wafer perspective
The customer return wafer is not an outlier within the lot
Lot perspective
The customer return lot is biased
The die is at the tail of the biased lot
Tutorial - Li-C. Wang, 2013
126

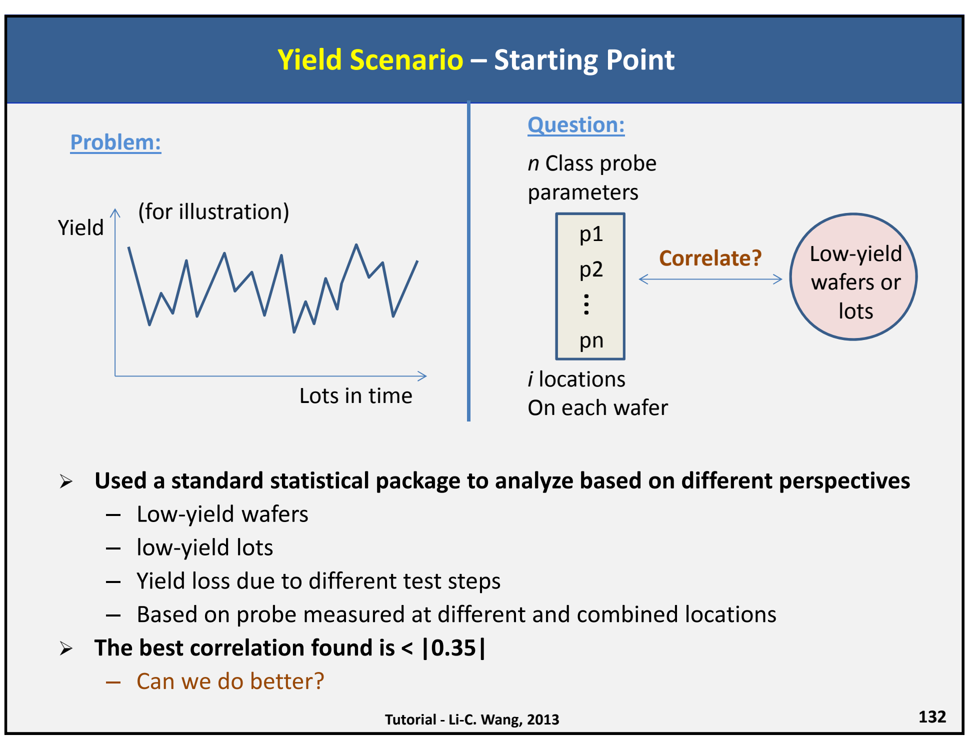

Abnormality Depends On Perspective – Test Aspect
Lot A
Lot B
Lot C
Combined lot-based fail patterns
Fail test Group A
Fail test Group B
Abnormal patterns depend on the group of tests we look at
Search for Abnormality includes search for the Test Perspective
Tutorial - Li-C. Wang, 2013
127

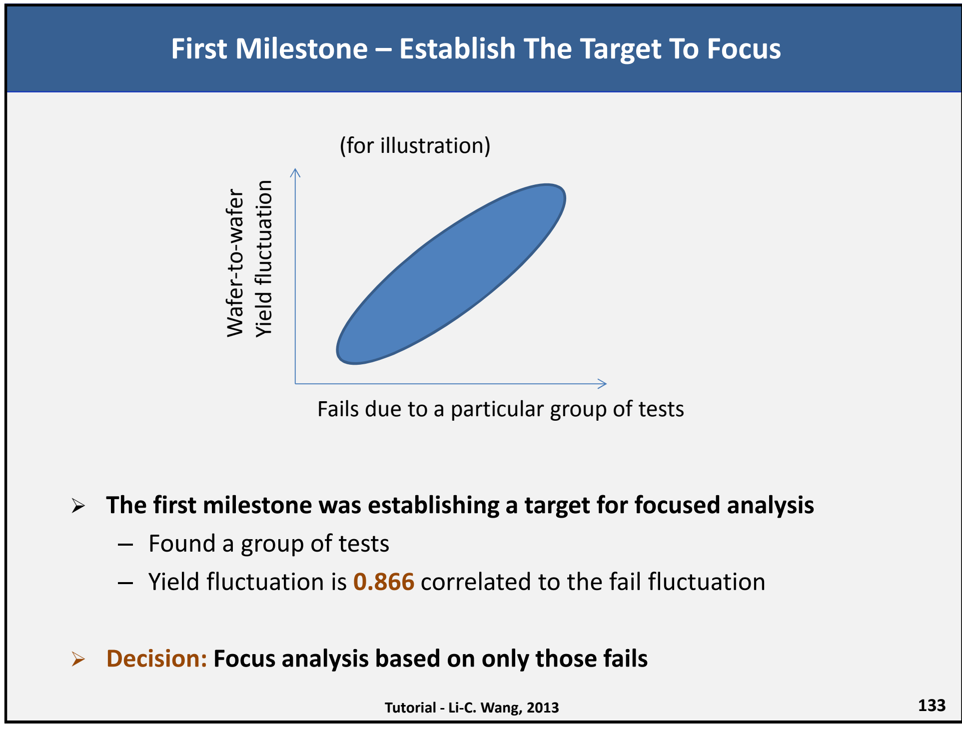

Variability Can Cause Misleading Result
Keys
Site 1
Site 2
Site 3
Site 4
Test 2
Test 1
Linear tend?
Site 1 Fails
Site 2 Fails
Site 3 Fails
Site 4 Fails
Passing
Site-to-site variation normalized
In the top-left figure, site-to-site variation causes a strong correlation between Test 1 and Test 2
Look like a linear trend
In the top-right, burn-in failing dies do not look like an outlier
In the bottom-right, after the normalization to remove site-to-site variation, they look like outliers
See Sumikawa et al. "An Experiment of Burn-In Time Reduction Based On Parametric Test Analysis" at ITC 2012
Tutorial - Li-C. Wang, 2013
128

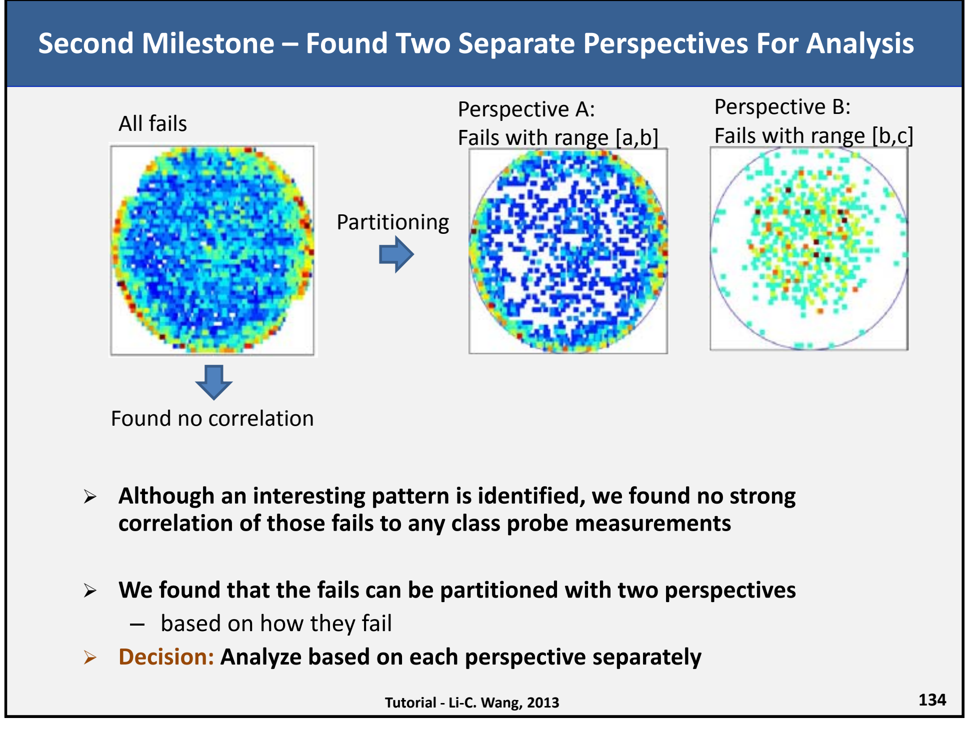

What Abnormalities Are For – Knowledge For Decision Making
Knowledge
Abnormalities ⇔
Target (die, wafer, lot)
Perceived
Interpretable
Knowledge
Knowledge Discovery
Meeting & Discussion
Actionable Knowledge
Actions
Investigate additional questions
Collect additional data
...
Implementation
Every action has a cost
Knowledge Discovery (KD) extracts "perceived" interpretable knowledge
– Meeting may involves design, product, test and process engineers
– In the meeting, interpretable knowledge translates into actionable knowledge
Most of time, actions lead to another KD process
Tutorial - Li-C. Wang, 2013
129

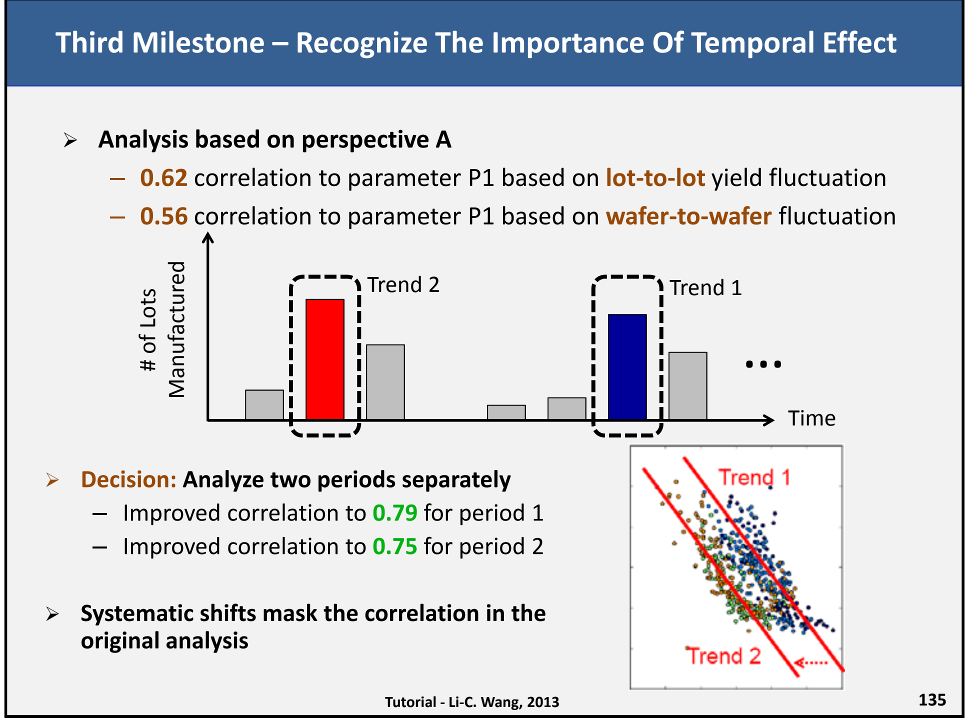

What's Difference Between Two Complex Learning Algorithms?
success
Success metric
Understanding (Data Preparation)
Data Exploration
Validation
Optimization
Time
In these stages, we prefer simple and quick learning methods allowing efficient exploration of a large number of perspectives
In data exploration, simple algorithms (tree learning, naïve Bayes, simple rule learning, 1-dimensional outlier, correlation calculation, etc.) are often used to check the data
Fine-tuning of a model happens in the Optimization stage
– Where complex algorithms with best parameter setting are applied
Tutorial - Li-C. Wang, 2013
130

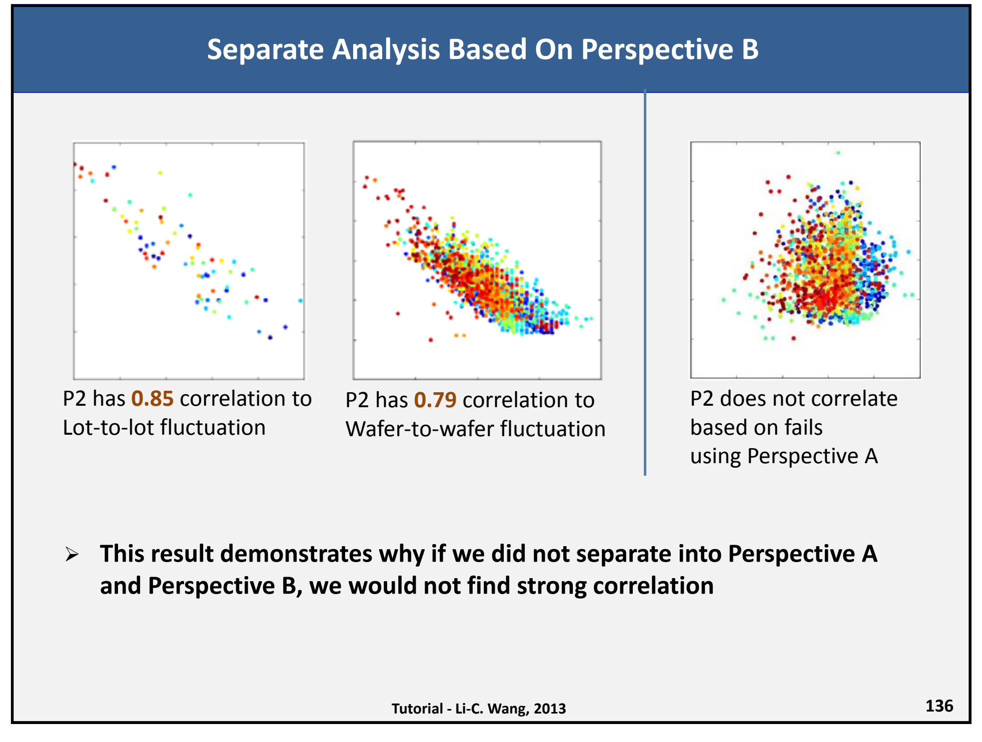

We Found Three Categories Of Tools That Are Useful
Unsupervised
# of abnormalities to be reported
A set of perspectives
Data
Abnormality Detection
Abnormalities
or
Distribution based
Pattern based
Target
A set of perspectives
Data
Perspective Search
Abnormalities
Perspectives to define them
Supervised
Data
Similarity Search
Similar abnormalities
Same or different test perspectives
For more detail, see Sumikawa et al. ITC 2013
– "A pattern mining framework for inter-wafer abnormality analysis"
Tutorial - Li-C. Wang, 2013
131


Yield Scenario – Starting Point
Problem:
Yield
(for illustration)
Lots in time
Question:
n Class probe parameters
p1
p2
pn
Correlate?
Low-yield wafers or lots
i locations
On each wafer
Used a standard statistical package to analyze based on different perspectives
– Low-yield wafers
– low-yield lots
– Yield loss due to different test steps
– Based on probe measured at different and combined locations
The best correlation found is < |0.35|
– Can we do better?
Tutorial - Li-C. Wang, 2013
132


First Milestone – Establish The Target To Focus
(for illustration)
Wafer-to-wafer Yield fluctuation
Fails due to a particular group of tests
The first milestone was establishing a target for focused analysis
– Found a group of tests
– Yield fluctuation is 0.866 correlated to the fail fluctuation
Decision: Focus analysis based on only those fails
Tutorial - Li-C. Wang, 2013
133


Second Milestone – Found Two Separate Perspectives For Analysis
All fails
Partitioning
Perspective A: Fails with range [a,b]
Perspective B: Fails with range [b,c]
Found no correlation
Although an interesting pattern is identified, we found no strong correlation of those fails to any class probe measurements
We found that the fails can be partitioned with two perspectives
– based on how they fail
Decision: Analyze based on each perspective separately
Tutorial - Li-C. Wang, 2013
134


Third Milestone – Recognize The Importance Of Temporal Effect
Analysis based on perspective A
– 0.62 correlation to parameter P1 based on lot-to-lot yield fluctuation
– 0.56 correlation to parameter P1 based on wafer-to-wafer fluctuation
# of Lots Manufactured
Trend 2
Trend 1
Time
Decision: Analyze two periods separately
– Improved correlation to 0.79 for period 1
– Improved correlation to 0.75 for period 2
Systematic shifts mask the correlation in the original analysis
Trend 1
Trend 2
Tutorial - Li-C. Wang, 2013
135


Separate Analysis Based On Perspective B
P2 has 0.85 correlation to Lot-to-lot fluctuation
P2 has 0.79 correlation to Wafer-to-wafer fluctuation
P2 does not correlate based on fails using Perspective A
This result demonstrates why if we did not separate into Perspective A and Perspective B, we would not find strong correlation
Tutorial - Li-C. Wang, 2013
136

More To Do Before Implementing A Process Change

- Made a recommendation for process parameter changes
- **Need to answer additional questions before implementation**
  - There was a suspected weak component – evaluated potential impact from the recommendation based on specific devices in the component
  - There was an earlier unsuccessful split lot experiment – made sure the recommendation do not cause the same problems
  - Made sure no evidence that the recommendation would not cause more fails due to other types of tests
- **After all those questions were cleared => Implemented the changes**

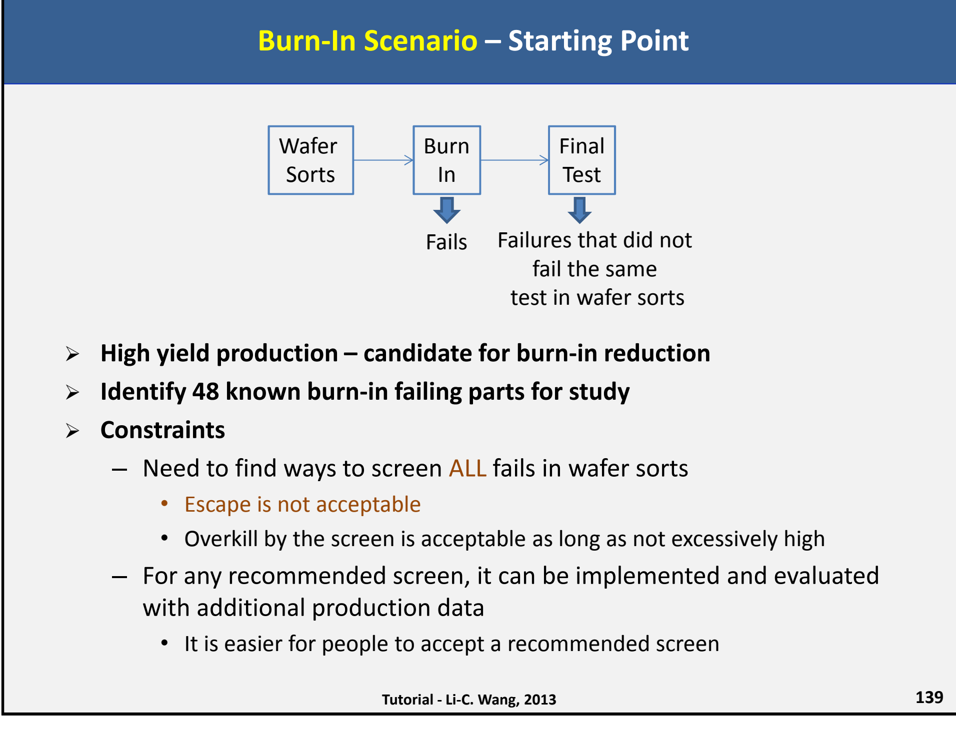
Tutorial - Li-C. Wang, 2013 137

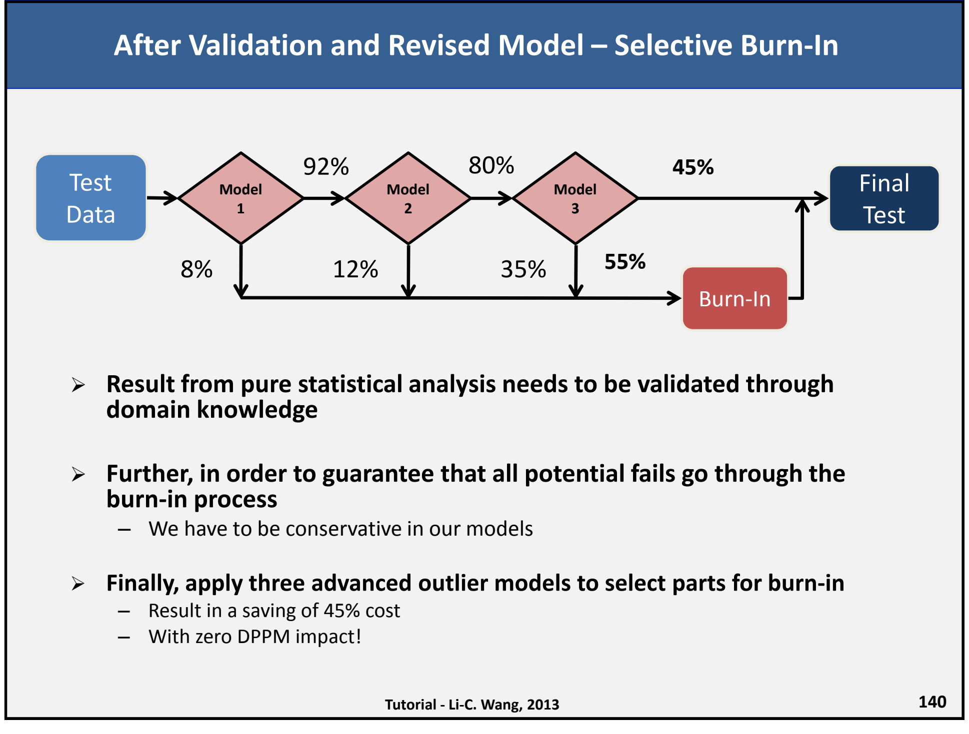
Recall: Six Months Later

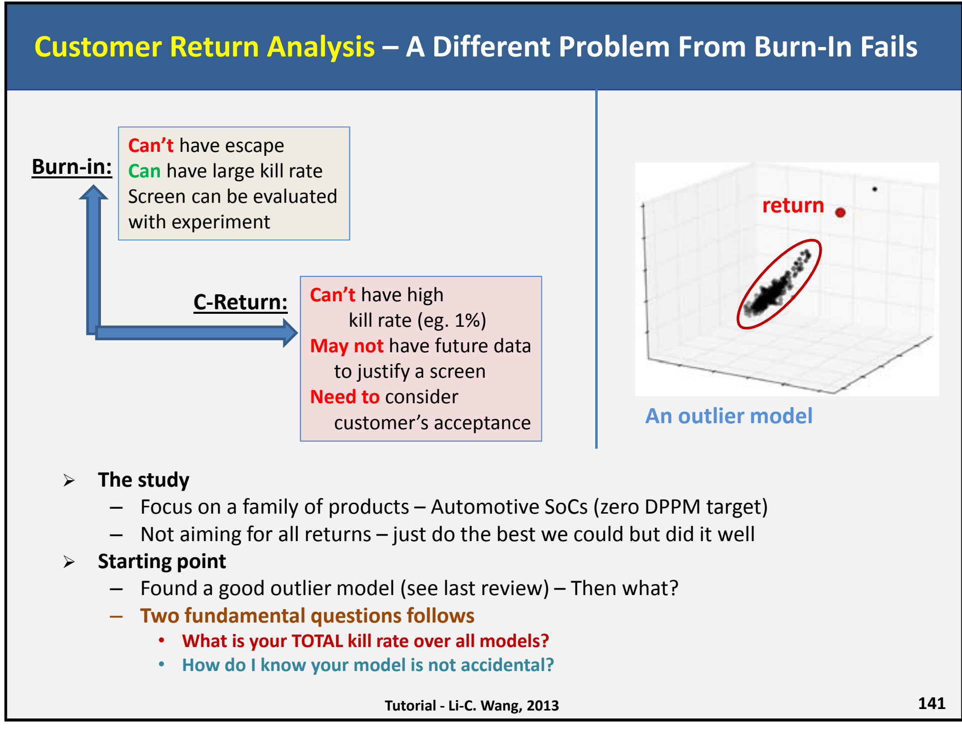
before RMD #1 RMD #2 Both

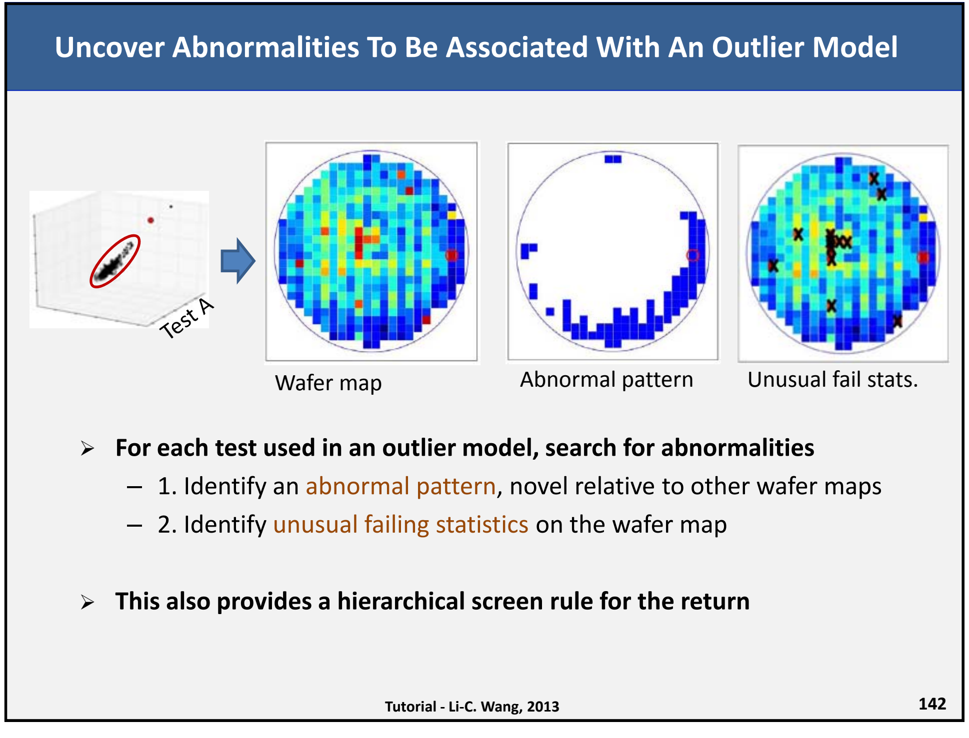
- **After 6-7 weeks of analysis and several meetings**
  - We recommended two process parameter changes
- **Changes were accepted by the product team and foundry to do a split-lot experiment**
- **Result shows significant improvement in yield and reduction of the fluctuation**

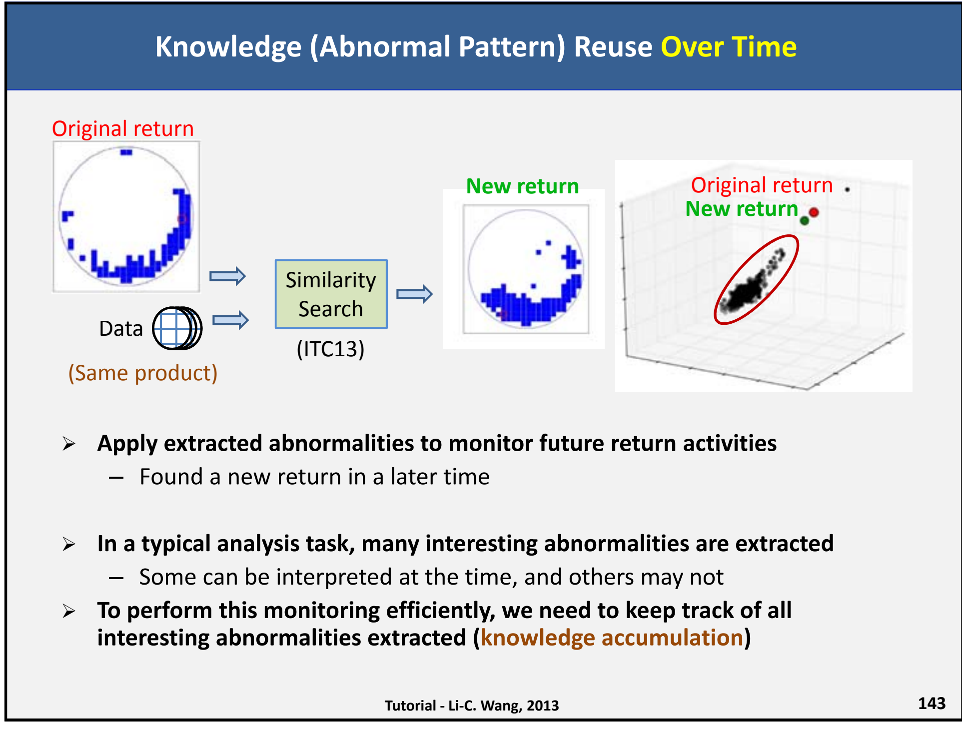
Tutorial - Li-C. Wang, 2013 138

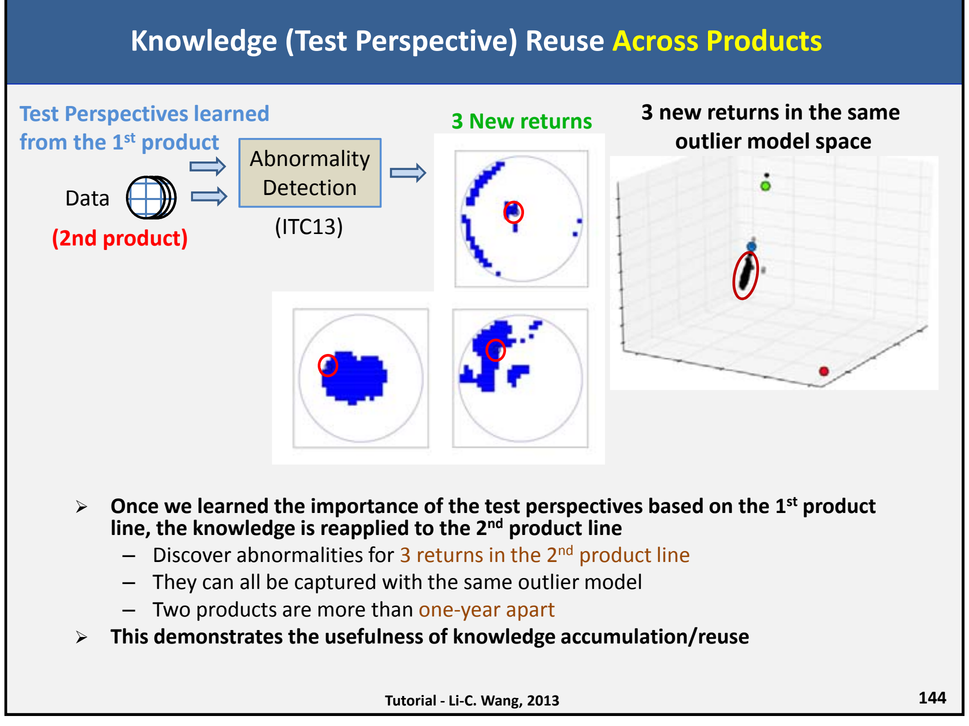

Burn-In Scenario – Starting Point
Wafer Sorts
Burn In
Final Test
Fails
Failures that did not fail the same test in wafer sorts
High yield production – candidate for burn-in reduction
Identify 48 known burn-in failing parts for study
Constraints
Need to find ways to screen ALL fails in wafer sorts
Escape is not acceptable
Overkill by the screen is acceptable as long as not excessively high
For any recommended screen, it can be implemented and evaluated with additional production data
It is easier for people to accept a recommended screen
Tutorial - Li-C. Wang, 2013
139


After Validation and Revised Model – Selective Burn-In
Test Data
Model 1
Model 2
Model 3
92%
80%
45%
Final Test
8%
12%
35%
55%
Burn-In
Result from pure statistical analysis needs to be validated through domain knowledge
Further, in order to guarantee that all potential fails go through the burn-in process
We have to be conservative in our models
Finally, apply three advanced outlier models to select parts for burn-in
Result in a saving of 45% cost
With zero DPPM impact!
Tutorial - Li-C. Wang, 2013
140


Customer Return Analysis – A Different Problem From Burn-In Fails
Burn-in:
Can't have escape
Can have large kill rate
Screen can be evaluated with experiment
C-Return:
Can't have high kill rate (eg. 1%)
May not have future data to justify a screen
Need to consider customer's acceptance
return
An outlier model
The study
Focus on a family of products – Automotive SoCs (zero DPPM target)
Not aiming for all returns – just do the best we could but did it well
Starting point
Found a good outlier model (see last review) – Then what?
Two fundamental questions follows
What is your TOTAL kill rate over all models?
How do I know your model is not accidental?
Tutorial - Li-C. Wang, 2013
141


Uncover Abnormalities To Be Associated With An Outlier Model
Test A
Wafer map
Abnormal pattern
Unusual fail stats.
For each test used in an outlier model, search for abnormalities
1. Identify an abnormal pattern, novel relative to other wafer maps
2. Identify unusual failing statistics on the wafer map
This also provides a hierarchical screen rule for the return
Tutorial - Li-C. Wang, 2013
142


Knowledge (Abnormal Pattern) Reuse Over Time
Original return
Similarity Search
(ITC13)
New return
Original return
New return
Data
(Same product)
Apply extracted abnormalities to monitor future return activities
– Found a new return in a later time
In a typical analysis task, many interesting abnormalities are extracted
– Some can be interpreted at the time, and others may not
To perform this monitoring efficiently, we need to keep track of all interesting abnormalities extracted (knowledge accumulation)
Tutorial - Li-C. Wang, 2013
143


Knowledge (Test Perspective) Reuse Across Products
Test Perspectives learned from the 1st product
Abnormality Detection
(ITC13)
3 New returns
3 new returns in the same outlier model space
Data
(2nd product)
Once we learned the importance of the test perspectives based on the 1st product line, the knowledge is reapplied to the 2nd product line
– Discover abnormalities for 3 returns in the 2nd product line
– They can all be captured with the same outlier model
– Two products are more than one-year apart
This demonstrates the usefulness of knowledge accumulation/reuse
Tutorial - Li-C. Wang, 2013
144

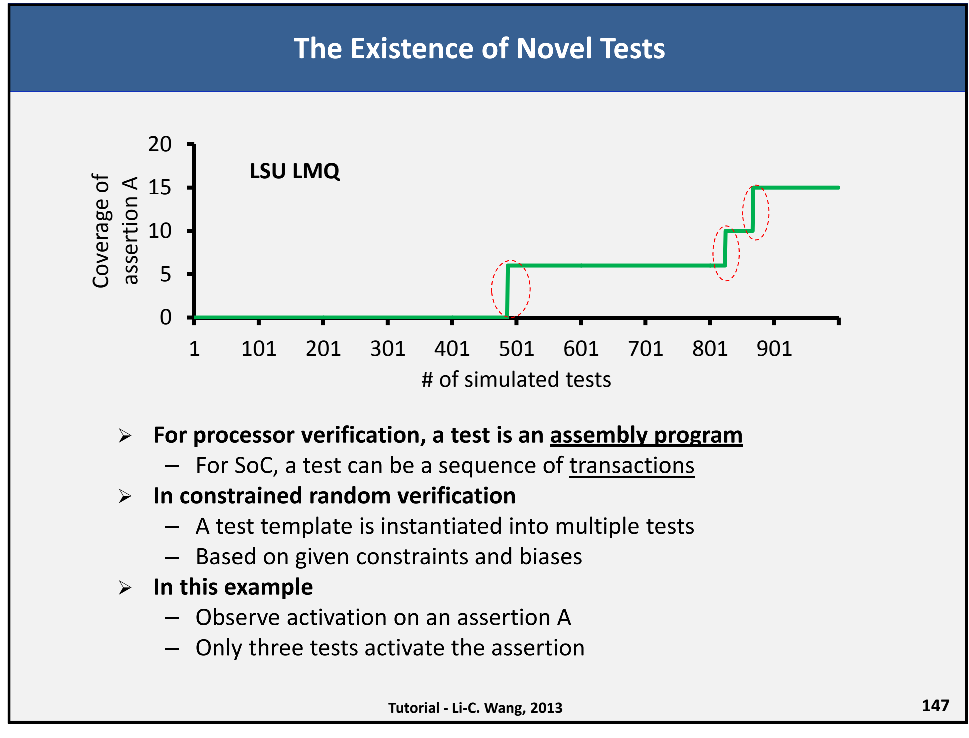
Knowledge Discovery in Functional Verification (15 minutes)

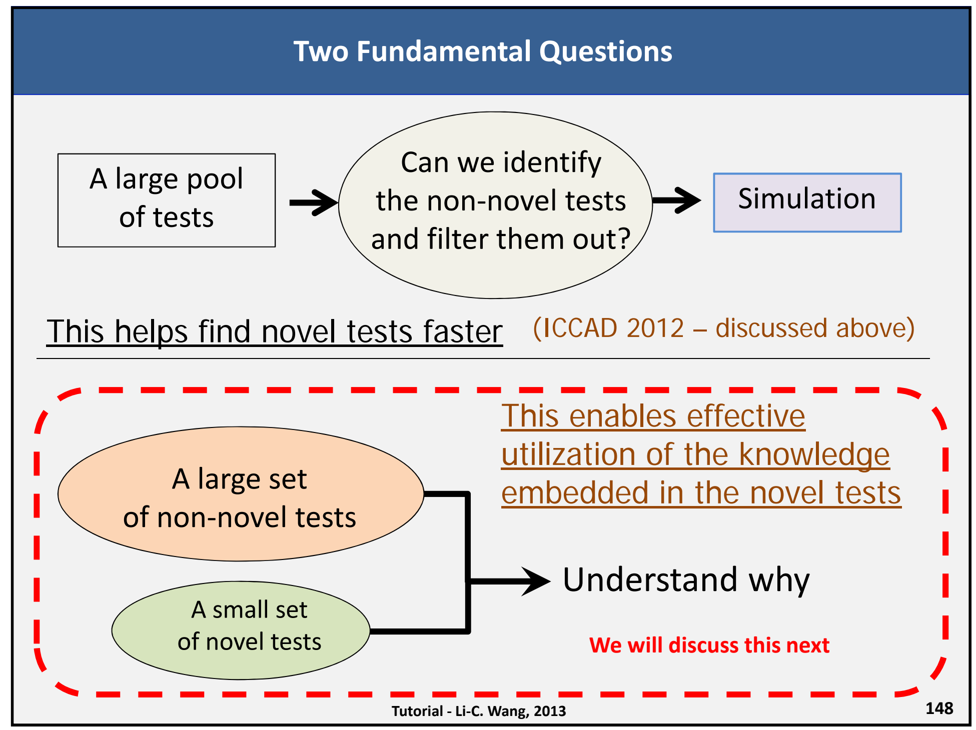
Tutorial - Li-C. Wang, 2013 145

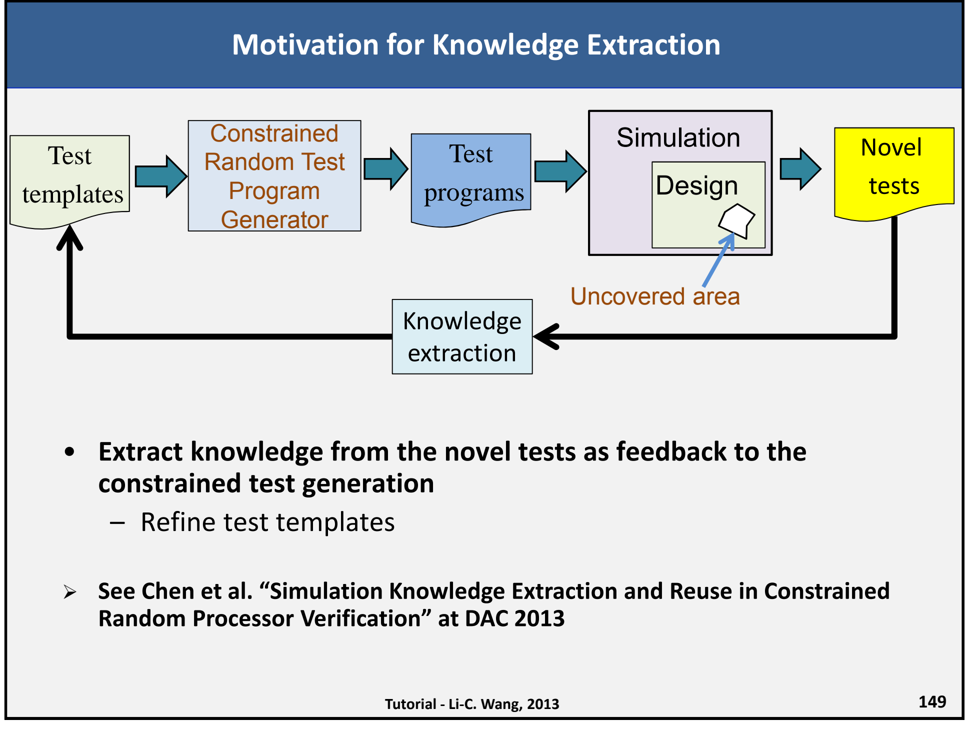
Application Context

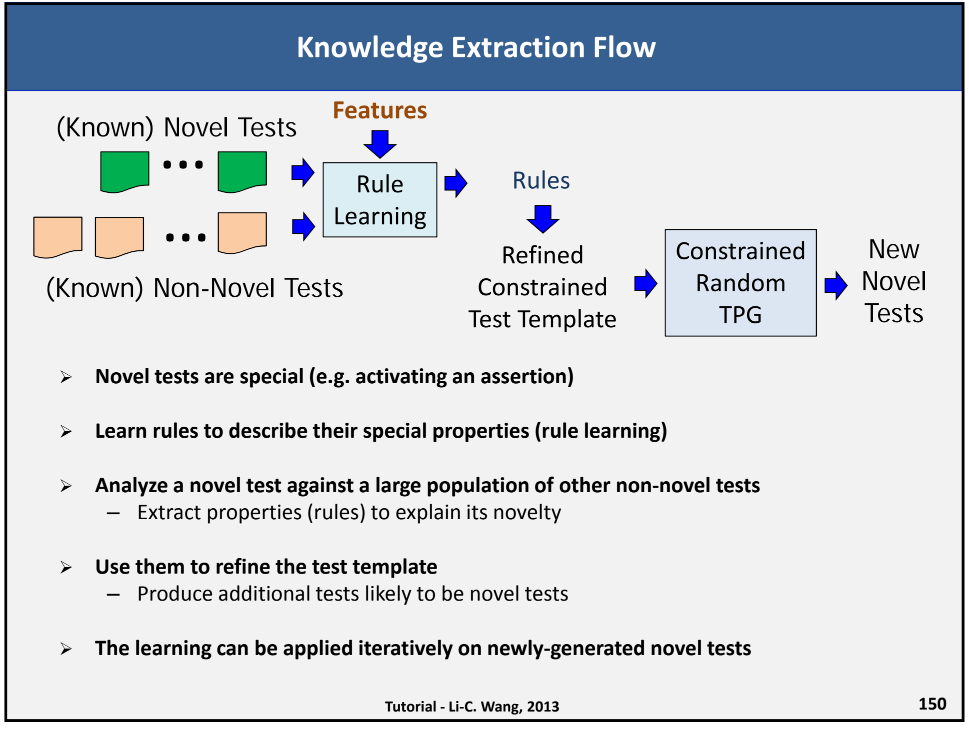
- **Focus on simulation based functional verification**
  - Based on constrained random verification environment
- **Functional verification is an iterative process**
  - Design changes over time
  - Verification restarts when a new version is released
- **Two assets are kept from one iteration to the next**
  - 1. Important (NOVEL) tests collected through simulation
    - For example, tests activating assertions of interest or capturing bugs
  - 2 . Test templates that produce those NOVEL tests
- **These two assets embed the knowledge accumulated the iterations of verification effort**

Tutorial - Li-C. Wang, 2013 146


The Existence of Novel Tests
Coverage of assertion A
20
15
10
5
0
LSU LMQ
1
101
201
301
401
501
601
701
801
901
# of simulated tests
For processor verification, a test is an assembly program
– For SoC, a test can be a sequence of transactions
In constrained random verification
– A test template is instantiated into multiple tests
– Based on given constraints and biases
In this example
– Observe activation on an assertion A
– Only three tests activate the assertion
Tutorial - Li-C. Wang, 2013
147


Two Fundamental Questions
A large pool of tests
Can we identify the non-novel tests and filter them out?
Simulation
This helps find novel tests faster
(ICCAD 2012 – discussed above)
This enables effective utilization of the knowledge embedded in the novel tests
A large set of non-novel tests
A small set of novel tests
Understand why
We will discuss this next
Tutorial - Li-C. Wang, 2013
148


Motivation for Knowledge Extraction
Test templates
Constrained Random Test Program Generator
Test programs
Simulation
Design
Novel tests
Uncovered area
Knowledge extraction
Extract knowledge from the novel tests as feedback to the constrained test generation
Refine test templates
See Chen et al. "Simulation Knowledge Extraction and Reuse in Constrained Random Processor Verification" at DAC 2013
Tutorial - Li-C. Wang, 2013
149


Knowledge Extraction Flow
(Known) Novel Tests
Features
Rule Learning
Rules
(Known) Non-Novel Tests
Refined Constrained Test Template
Constrained Random TPG
New Novel Tests
Novel tests are special (e.g. activating an assertion)
Learn rules to describe their special properties (rule learning)
Analyze a novel test against a large population of other non-novel tests
Extract properties (rules) to explain its novelty
Use them to refine the test template
Produce additional tests likely to be novel tests
The learning can be applied iteratively on newly-generated novel tests
Tutorial - Li-C. Wang, 2013
150

Two-level of Features

Instruction sequence | Arch. feature vector | Instr. feature vector

...

- **Architecture features (A-features)**
  - Based on architecture states from architectural simulation
  - Based on micro-architecture states from the workbook
- **Instruction features (I-features)**
  - Describe important characteristics of an instruction
- **See Wen Chen et al. (DAC 2013)**
  - "Simulation Knowledge Extraction and Reuse for Processor Verification"
  - Also "A Two-level Learning Framework for Knowledge Discovery in Constrained Random Processor Verication" Manuscript 2013

Tutorial - Li-C. Wang, 2013 151

A-Features

Feature	Rules
STQFWD enable (-1, i); i=0,1, … 10	Store => Load ∧ address collision ∧ no more than i instruction in between the store load pair
LMQ enable	Load ∧ CacheInhibited=1
	Load ∧ CacheInhibited=0 ∧ folding=0
	waitrsv
	…
Cflush enable	Multiply ∧ result overflow ∧ XER[o]=0
	Mispredicted branch
	isync
	…
TLB invalid	tlbivax
ST queue full	Stmw ^ RT<23
	…

- **Each feature corresponds to a state variable described in the workbook**
  - Rules to activate the feature are recorded in the tool, and used to check if a test program activates the feature

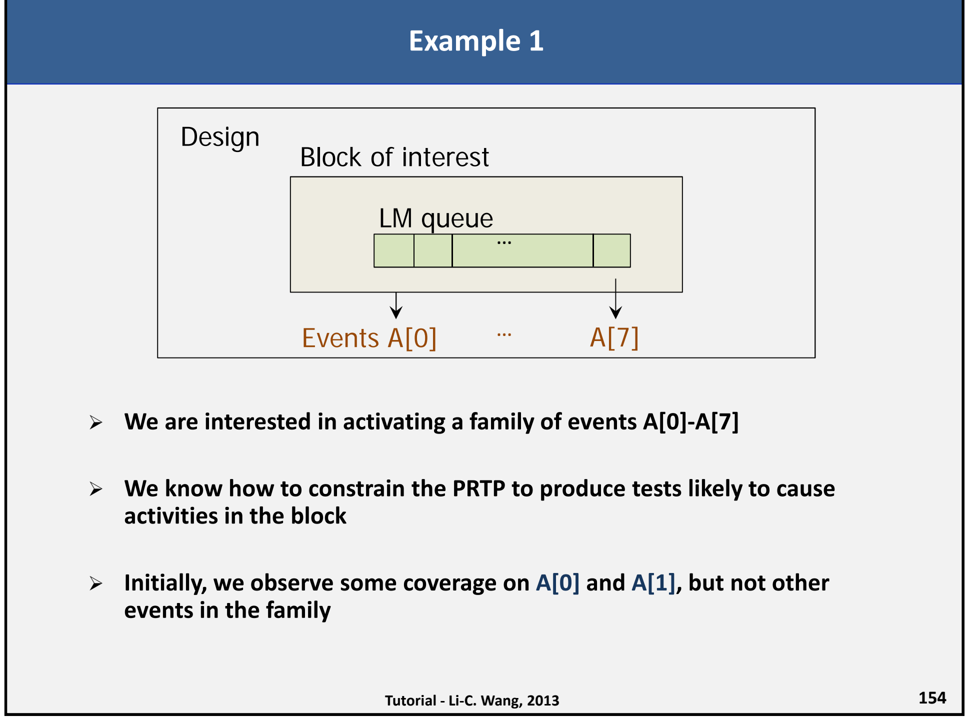
Tutorial - Li-C. Wang, 2013 152

I-Features

Instructions	features
lmw	RT, EA, RA, misaligned, address collision
stmw	RT, EA, RA, misaligned, address collision
mulld	RA, RB, execution result, overflow, data dependency
divd	RA, RB, execution result, divide-by-zero, data dependency
add	RA, RB, execution result, overflow, data dependency
Sub	RA, RB, execution result, underflow, data dependency
branch	mispredicted
…	

- **Features are to describe the important characteristics of an instruction**
  - These features are used for refining the learning result based on A-features

Tutorial - Li-C. Wang, 2013 153



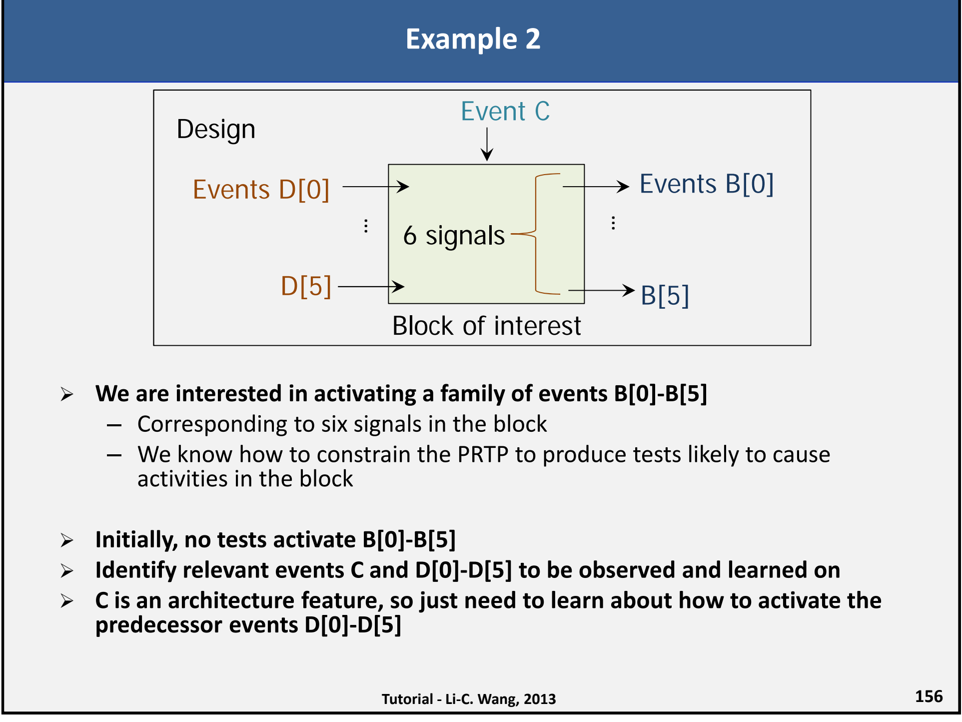
Result

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

Tutorial - Li-C. Wang, 2013 155

Example 2



- **We are interested in activating a family of events B[0]-B[5]**
  - Corresponding to six signals in the block
  - We know how to constrain the PRTP to produce tests likely to cause activities in the block
- **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]**

Tutorial - Li-C. Wang, 2013 156

Result

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]
  - Fortuitously, 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

Final Remark and Questions
(10->5 Minutes)

Final Remarks – BIG Data (Medium Data)

- **Collection of data sets (Big)**
  - Extremely large and complex
  - Difficult for traditional database and/or data processing tools
- **Challenges in multi-fronts (Big/Medium)**
  - Capture
  - Storage
  - Search
  - Sharing
  - Transfer
  - **Model/analysis**
  - Visualization
- **Let's focus on the Model/Analysis aspect**
  - **Do Design and Test have the "BIG data" problems?**

Tutorial - Li-C. Wang, 2013 159

Model and Analysis With "Big" Data

- **Modeling consumer behavior**
  - The underlying "function" is rather steady
  - We have time to accumulate enough data
- **Medical diagnosis**
  - The underlying "function" is rather steady
  - We have time to accumulate enough data
- **Social network mining**
  - The underlying "function" is rather steady
  - We have time to accumulate enough data
- **Other examples?**

Tutorial - Li-C. Wang, 2013 160

What Are Our Problems Like?

- Why silicon timing does not match my predicted timing?
  - Very much case dependent – underlying reasons can be many
  - There is a time limit for the answer to be valuable
  - Data is limited (additional data may be costly or prohibited)

- Are my defects caused by DFM issues? Which?
- Can we find actions to contain these 15 customer returns?
- Can we find way to screen these 50 burn-in fails so that we don't need to run burn-in?
- Can we find a recipe to adjust the process for improving yield?
- Can we learn how to effectively activate this functional state?
- Can we optimize the functional tests for silicon power worsening?
- …

- **We have a "Small" Data Model and Analysis Problem!!**

Tutorial - Li-C. Wang, 2013 161

Something To Think About …

Small (Specific)	Big (Asymptotic)
The underlying "function" to learn is very case-dependent	The underlying "function" to learn is rather steady
Getting new data can be costly or prohibited	If data is not enough, wait and get more
While we may large amounts of data, we have little information on the care space	Data can be accumulated over time – hence the data is almost "unlimited"
Look for novelty (specialty, abnormality)	Look for trends (frequent patterns)
Trends are often obvious to the domain experts	Trends are new knowledge
There is a strict time constraint for the answer to be valuable	These is less time constraint to solve the problem
Research focuses on ???	Much research focuses on optimizing the learning algorithms

- … There can be other angles to differentiate the two paradigms

Tutorial - Li-C. Wang, 2013 162

Five Key Messages To Take Away

- **5. A complex algorithm may not perform better in a specific scenario – in most of the cases a simple algorithm (like CART) may be enough**
- **4. Data mining in design and test is a Knowledge Discovery process – uncover Interpretable and Actionable knowledge**
- **3. In a Knowledge Discovery process, data preparation and data exploration consumes most of the time**
- **2. Before you try learning, try some simple non-learning based heuristic first – that may give you the best result already**
- **1. We can only declare a success when people accept the result, action is taken, and improvement is observed over the existing flow**

Tutorial - Li-C. Wang, 2013 163

The End

THANK YOU!

Tutorial - Li-C. Wang, 2013 164