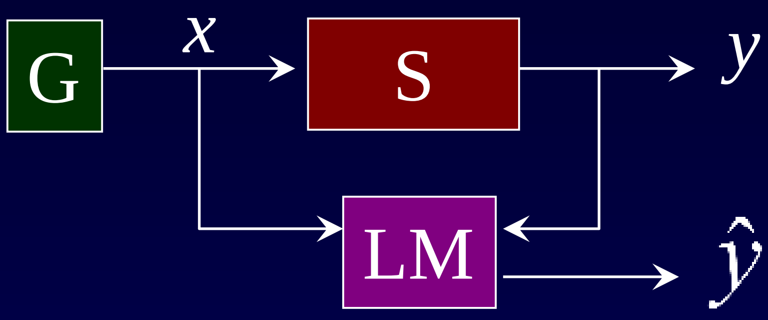
Data Mining In Design and Test Processes – Basic Principles and Promises

Li-C. Wang
UC-Santa Barbara

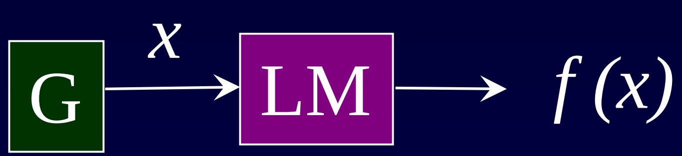
Outline

- Machine learning basics
- Application examples
- Data mining is knowledge discovery
- Some results
  - Analyzing design-silicon mismatch
  - Improve functional verification
  - Analyzing customer returns

Supervised vs. Unsupervised learning



Supervised



Unsupervised

- A generator G of random vector $x \in R^n$, drawn independently from a fixed but unknown distribution $F(x)$
  - *This is the iid assumption*
- Supervised learning
  - A supervisor S who returns an output value y on every input x, according to the conditional distribution function $F(y \mid x)$, also fixed and unknown
- A learning machine LM, capable of implementing a set of functions $f(x, \alpha)$, where $\alpha \in \Lambda$ that is a set of parameters

Dataset *usually* look like

$$\begin{array}{cccc} f_1 & f_2 & \dots & f_n \end{array} \longleftarrow \text{features}$$

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

supervised

- m samples are given for learning
- Each sample is represented as a vector based on n features
- In supervised case, there is a y vector

Learning algorithms

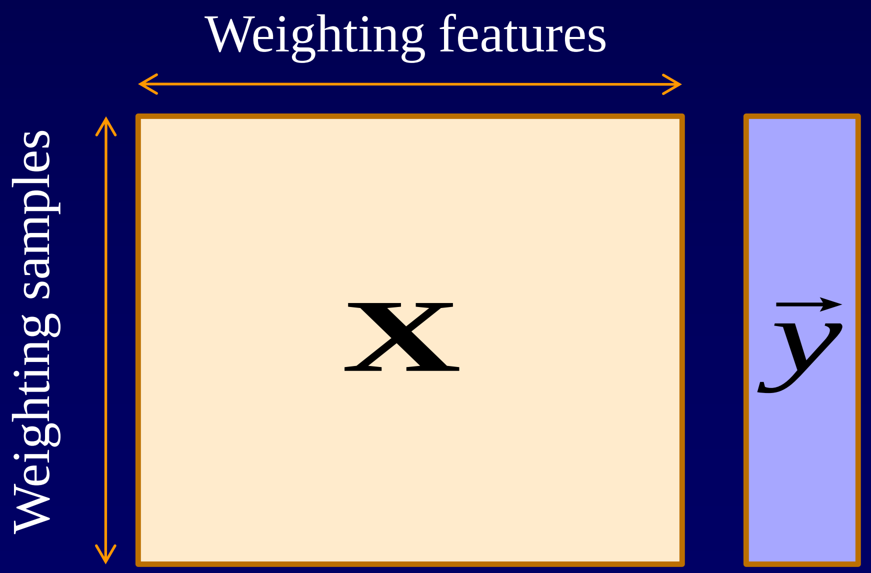
- Supervised learning
  - Classification (y represents a list of classes)
  - Regression (y represents a numerical output)
  - Feature ranking
  - Classification (regression) rule learning

- Unsupervised learning
  - Transformation (PCA, ICA, etc.)
  - Clustering
  - Novelty detection (outlier analysis)
  - Association rule mining

- In between, we have
  - Rule (diagnosis) learning (classification with extremely unbalanced dataset – one/few vs. many)

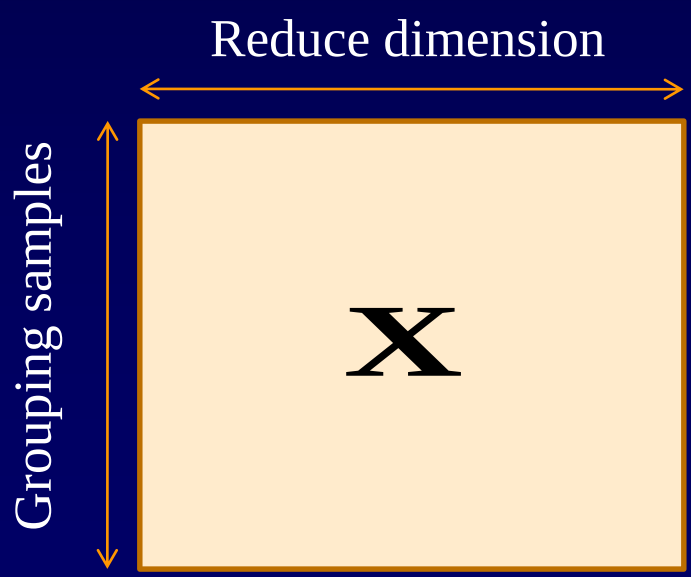
Supervised learning

- Supervised learning learns in 2 directions:
  - Weighting the features
  - Weighting the samples
- Supervised learning includes
  - Classification – y are class labels
  - Regression – y are numerical values
  - Feature ranking – select important features
  - Classification rule learning – select a combination of features

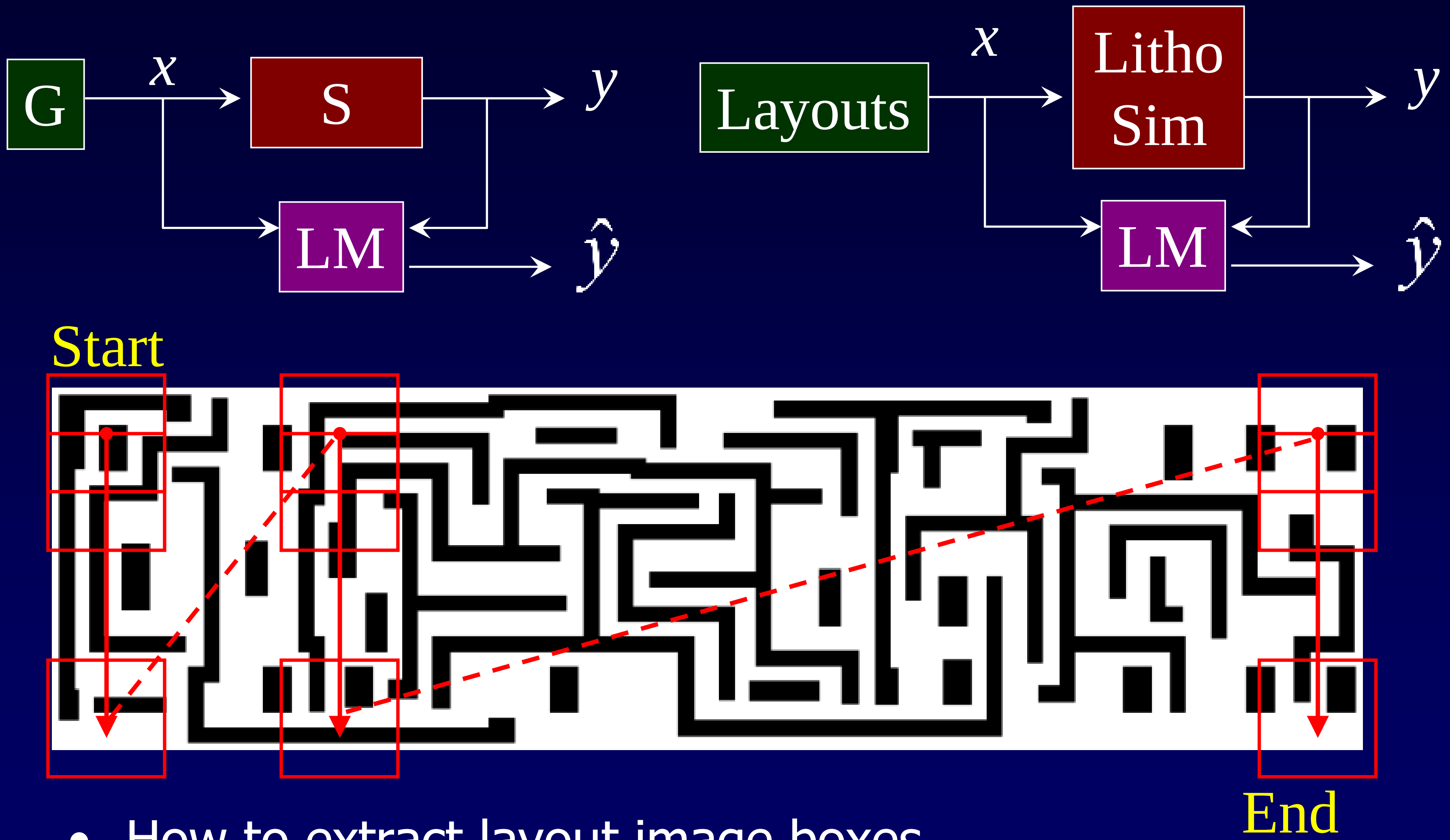


Unsupervised learning

- Unsupervised learning also learns in 2 directions:
  - Reduce feature dimension
  - Grouping samples
- Unsupervised learning includes
  - Transformation (PCA, multi-dimensional scaling)
  - Association rule mining (explore feature relationship)
  - Clustering (grouping similar samples)
  - Novelty detection (identifying outliers)

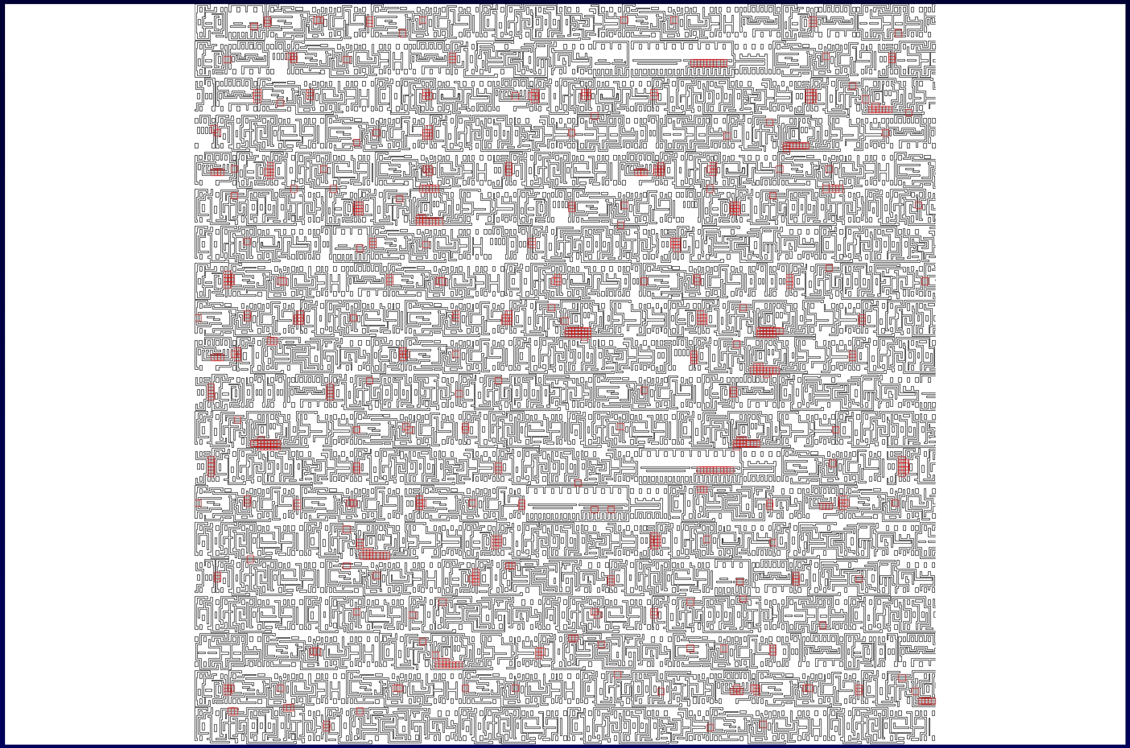


Supervised learning example



- How to extract layout image boxes
- How to represent a image box
- Where to get training samples?

DAC 2009



- Based on IBM in-house litho simulation (Frank Liu)
- Learn from cell-based examples
- Scan chip layout for spots sensitive to post-OPC lithographic variability
- Identify spots almost the same as using a lithographic simulator
- But orders-of-magnitude faster

Supervised - Fmax prediction

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

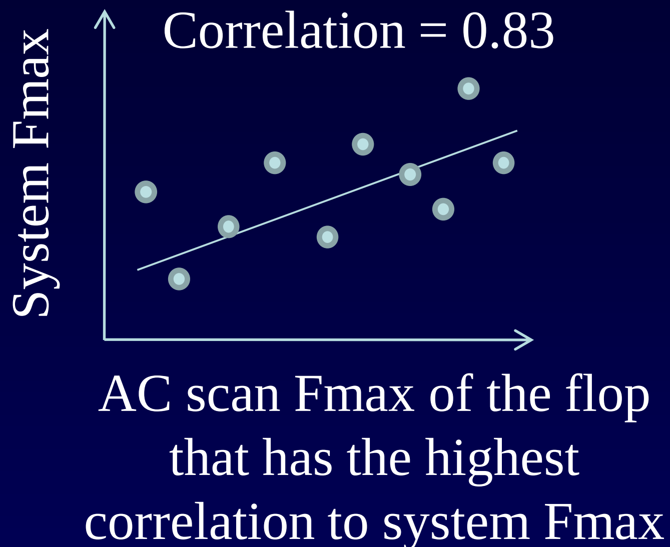
Dataset

$$
\begin{array}{cc}
 & \overbrace{\begin{array}{cccc} M_1 & M_2 & \ldots & M_n \end{array}}^{n \text{ delay measurements}} \\
\mathbf{X} = \begin{vmatrix} \vec{x}_1 \\ \vec{x}_2 \\ \cdots \\ \vec{x}_m \end{vmatrix} = & \begin{vmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{vmatrix}
\end{array}
\qquad
\begin{array}{c}
\text{Fmax} \\
\mathbf{y} = \left.\begin{vmatrix} y_1 \\ y_2 \\ \cdots \\ y_m \end{vmatrix}\right\} m \text{ samples chips}
\end{array}
$$

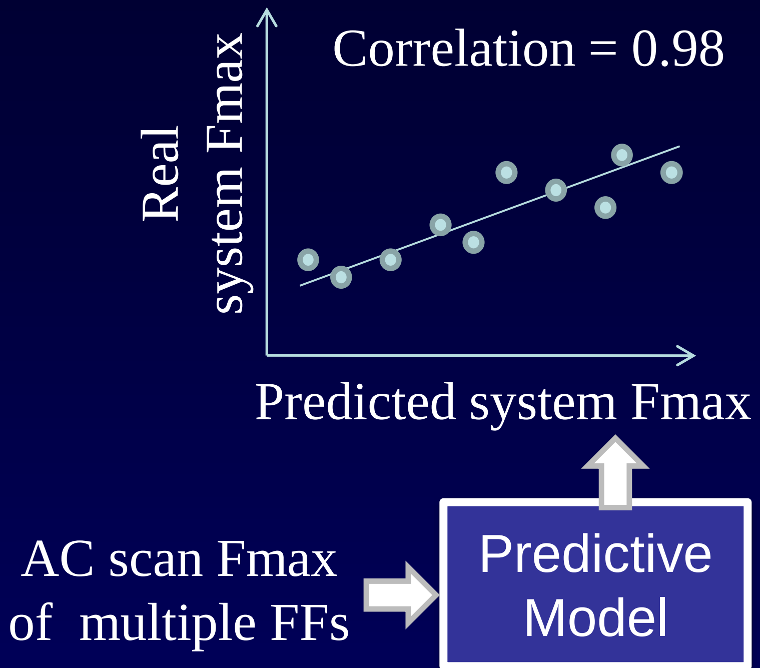
Fmax of c?

- Fmax prediction is to generalize the correlation in between a random vector of (cheap) delay measurements and the random variable Fmax

Predicting system Fmax (ITC 2010)



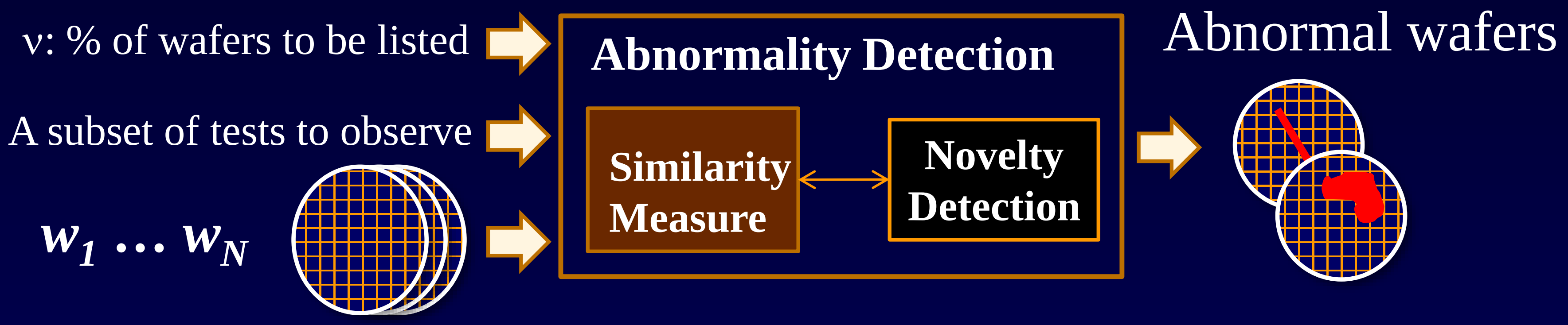
(a). 1-dimensional correlation



(b). Multi-dimensional correlation

- A predictive model can be learned from data
  - This model takes multiple structural frequency measurements as inputs and calculate a predicted system Fmax
- For practical purpose, this model needs to be interpretable

Unsupervised learning example



- In order to perform novelty detection, we need to have a similarity measure
  - Similarity between given two wafer maps

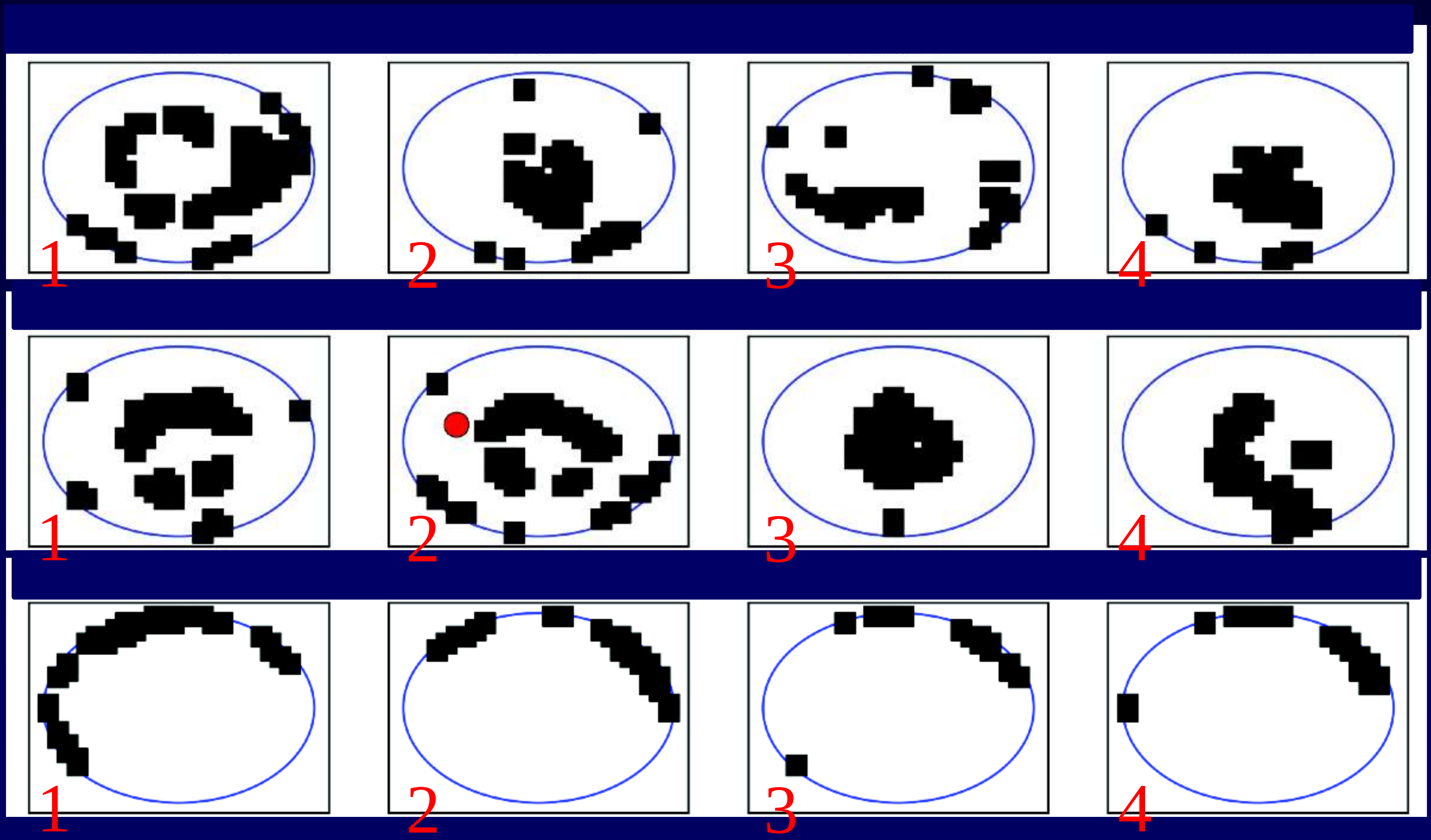
- Then, the objective is to identify wafers whose patterns are very different from others

Example results

BIST

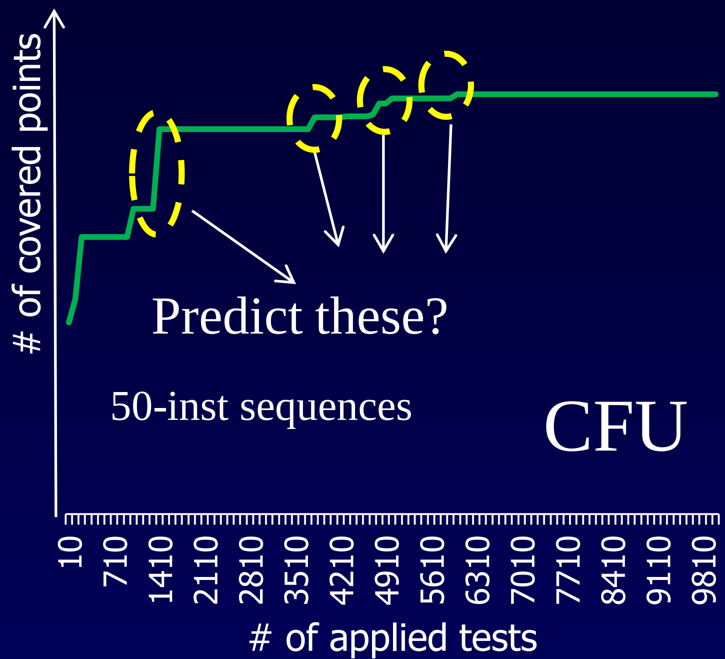
Scan

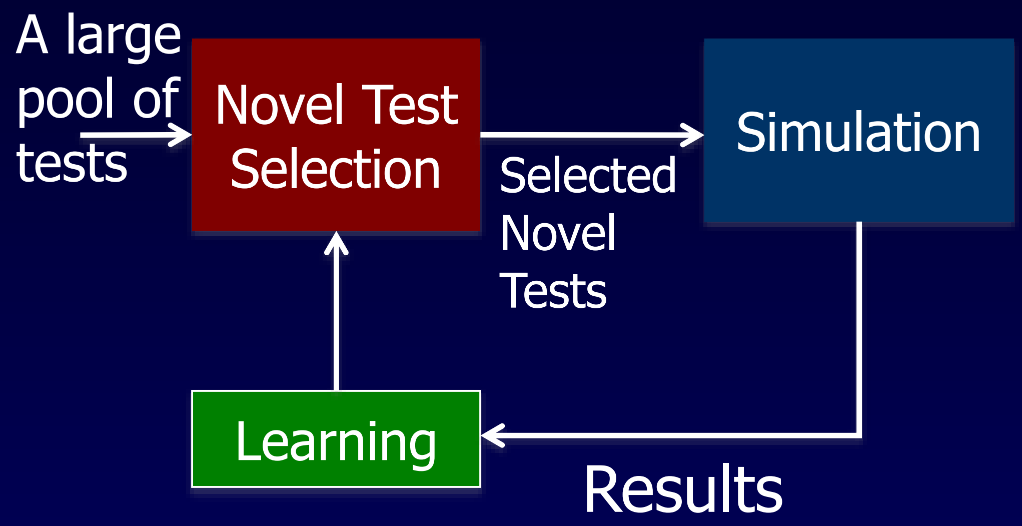
Flash



- Help understand unexpected test behavior based on a particular test perspective

Unsupervised learning example

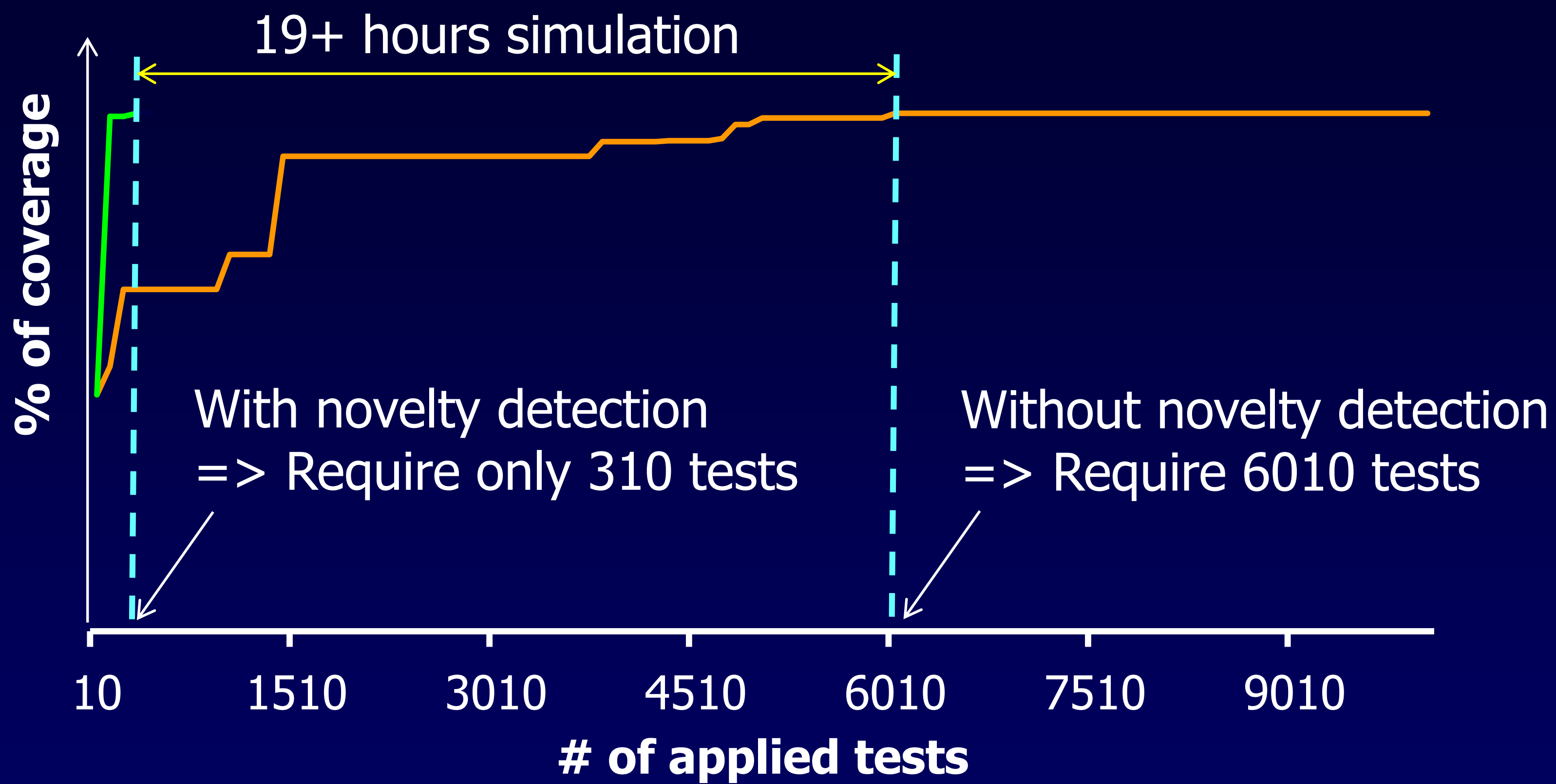




- In constrained random verification, simulation cycles are wasted on ineffective tests (assembly programs)

- Apply novelty detection to identify "novel" tests for simulation (tests different from those simulated)

Example result (ICCAD 2012)



- The novelty detection framework results in a dramatic cost reduction
  - Saving 19 hours in parallel machine simulation
  - Saving days if ran on single machine simulation

Simplistic view of “data mining”

- Data are well organized
- Data are planned for the mining task
- Our job
  - Apply the best mining algorithm
  - Obtain statistical significant results

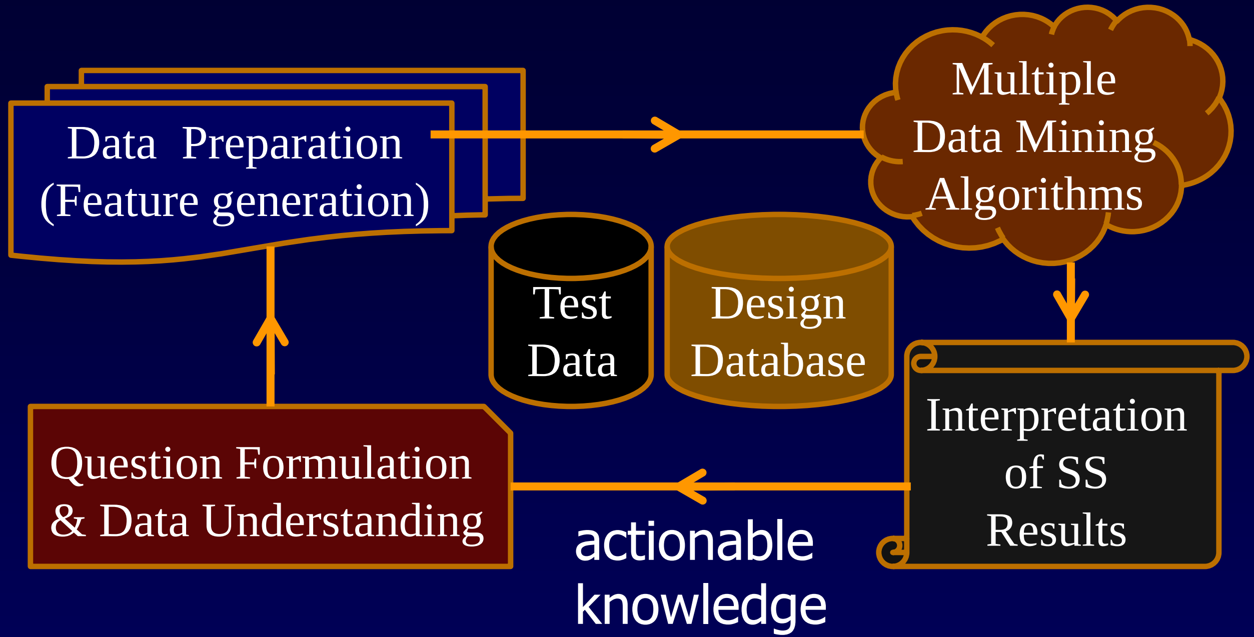
What happened in reality

- Data are not well organized (missing values, not enough data, etc.)
- Initial data are not prepared for the mining task
- Questions are not well formulated
- One algorithm is not enough

- More importantly, the user need to know why before taking an important action
  - Drop a test or remove a test insertion
  - Make a design change
  - Tweak process parameters to a corner

- Interpretable evidence is required for an action

Data mining ⇒ Knowledge Discovery



- The mining process is iterative
- Questions are refined in the process
- Multiple datasets are produced
- Multiple algorithms are applied
- Statistical significant (SS) results are interpreted through domain knowledge
- Discover actionable and interpretable knowledge

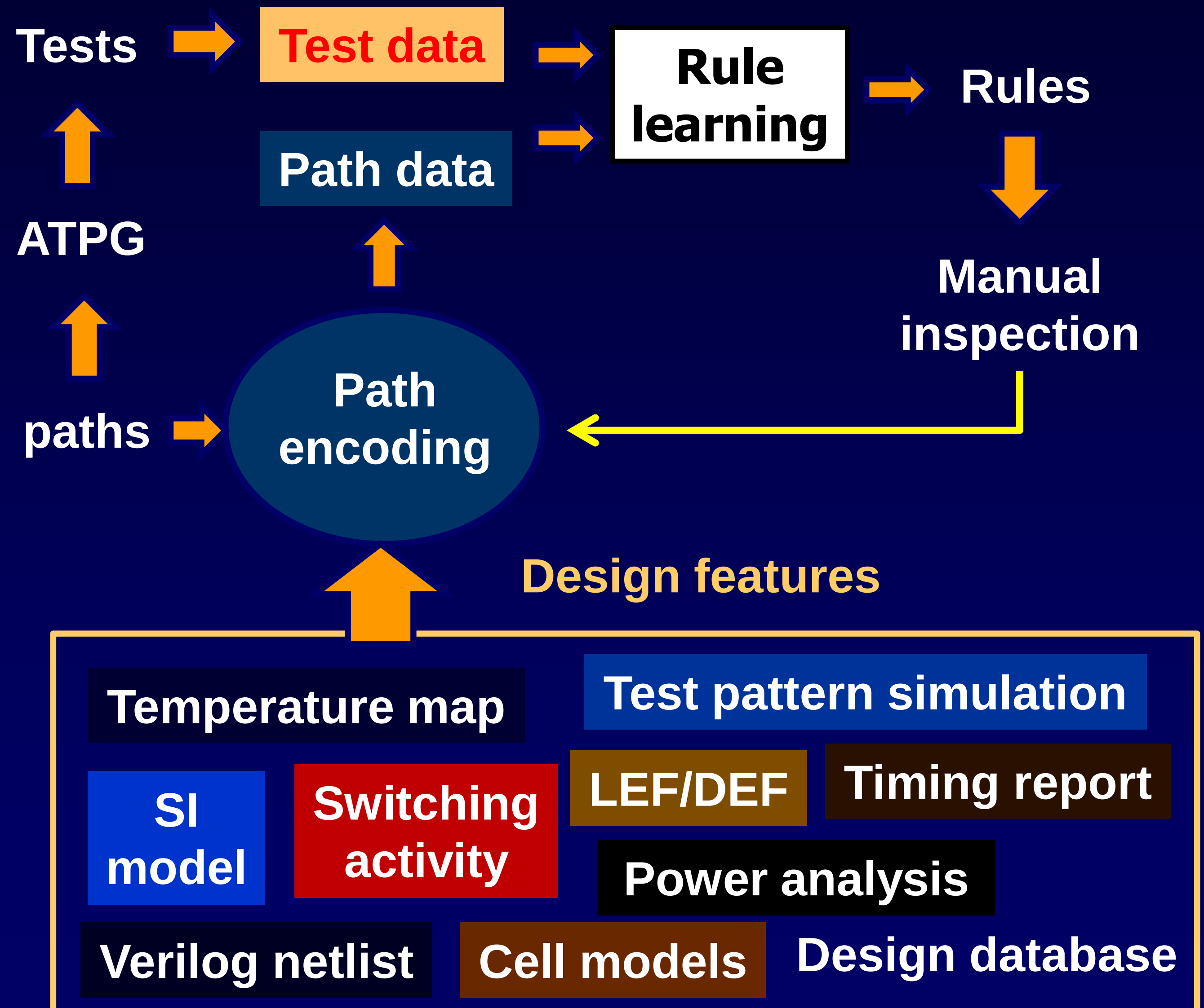
Example – analyzing design-silicon mismatch

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

- Based on AMD quad-core processor (ITC 2010)
- There are 12,248 STA-long paths activated by patterns
  - They don't show up as silicon critical paths
- 158 silicon critical but STA non-critical paths

- Question: Why are the 158 paths so special?
  - Use 12,248 silicon non-critical paths as the basis for comparison

Overview of the infrastructure



Example result

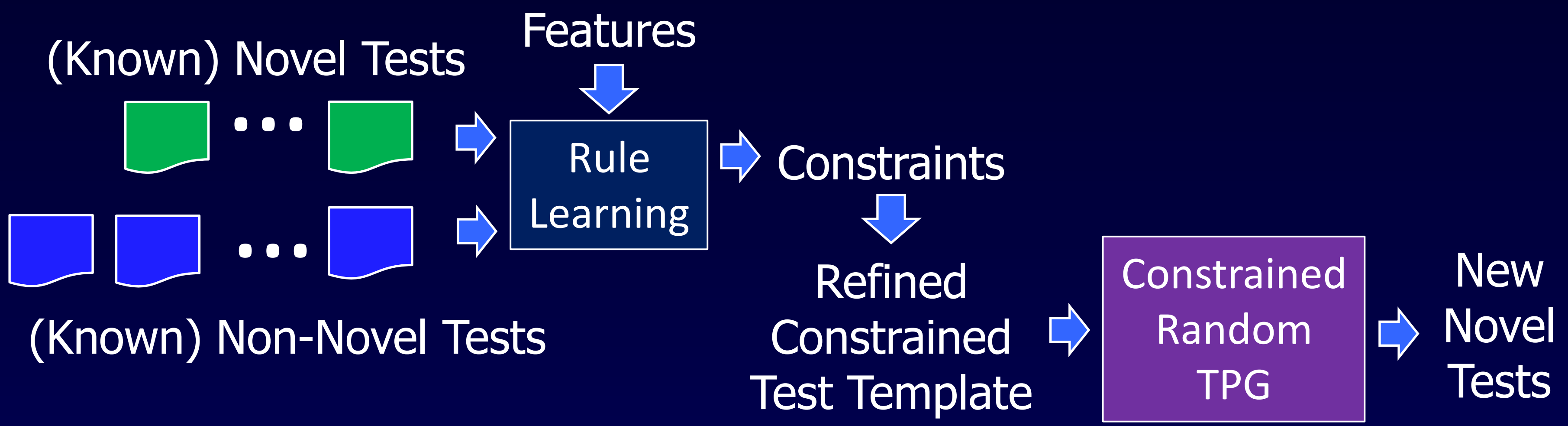
Rule#	Rule	$S_{critical}$	$S_{non-critical}$
#1	$CMAC \in [8,14] \wedge$ $CCT \in [6,7]$	68	1
#2	$CMAC \in [8,14] \wedge$ $PS \in [247,\infty]$	47	0
#3	$VRC \in [0.565,0.571] \wedge$ $PS \in [247,\infty]$	47	0
#4	$CMAC \in [8,14] \wedge$ $VRC \in [0.565,0.571]$	46	0
#5	$CMAC \in [8,14] \wedge$ $VRD \in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

Manual inspection of rules #1,2,4,5 led to
Explanation of 68 paths; Then, for the rest, run again

Rule#	Rule	$S_{critical}$	$S_{non-critical}$
#1	$CID \in [102,148] \wedge$ $TS \in [378,404]$	26	0
#2	$CBC \in [0,\infty] \wedge$ $TS \in [378,404]$	25	0
#3	$CBC \in [0,\infty] \wedge$ $CFD \in [38,39]$	24	0
#4	$CFD \in [38,39] \wedge$ $TS \in [378,404]$	24	0

Manual inspection
Explains additional
25 paths

Rule learning for analyzing functional tests



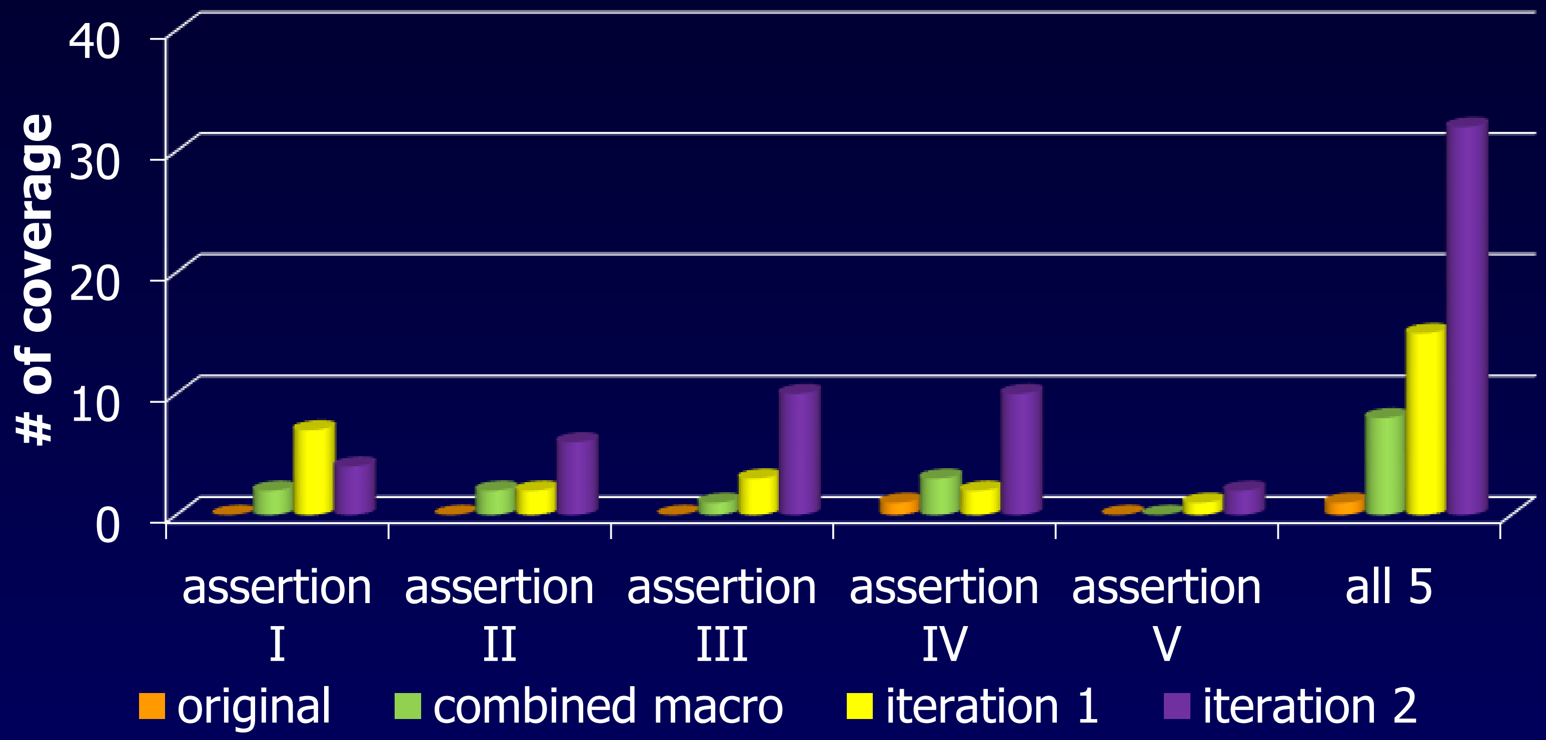
- Novel tests are special (e.g. hitting an assertion)
  - Learn rules to describe their special properties
- Analyze a novel test against a large population of other non-novel tests
  - Extract properties to explain its novelty
- Use them to refine the test template
- Produce additional tests similar to the novel tests
- The learning can be applied iteratively on newly-generated novel tests

Example result (DAC 2013)

- Five assertions of interest-I, II, III, IV, V
  - Comprise the same two condition c_1 and c_2
  - Temporal constraints between c_1 and c_2 are different across different assertions
  - Initially, only assertion IV was hit by one test out of 2000
  - Learn rules for c_1 and c_2 respectively, and combine the rule macro m_1(for c_1) and rule macro m_2(for c_2) based on the ordering in the novel test

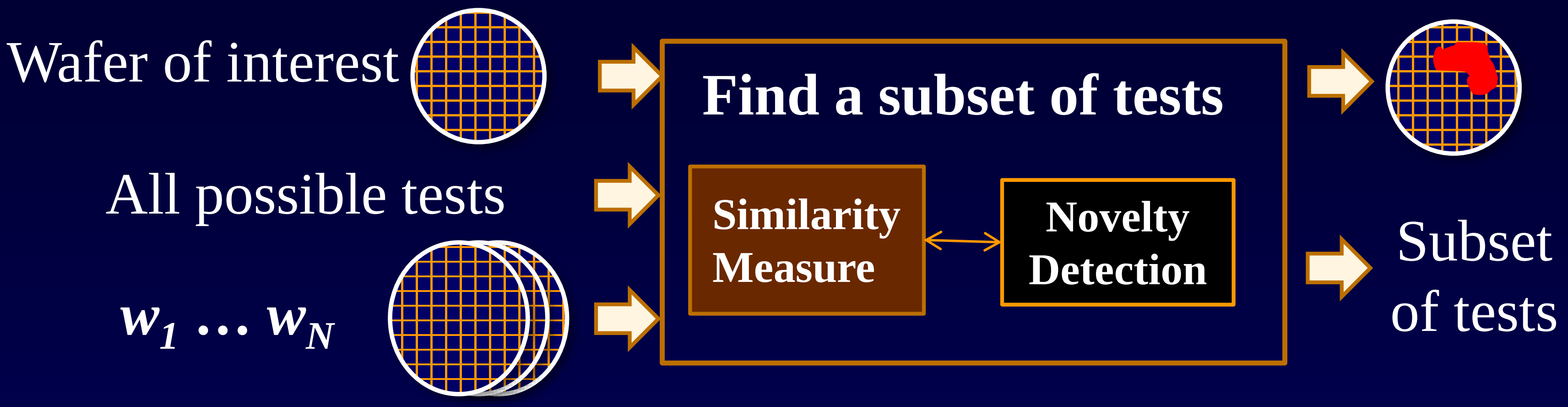
Rule for m1	There is a mulld instruction and the two multiplicands are larger than 2^{32}
Rule for m2	There is a lfd instruction and the instructions prior to the lfd are not memory instructions whose addresses collide with the lfd

Coverage improvement



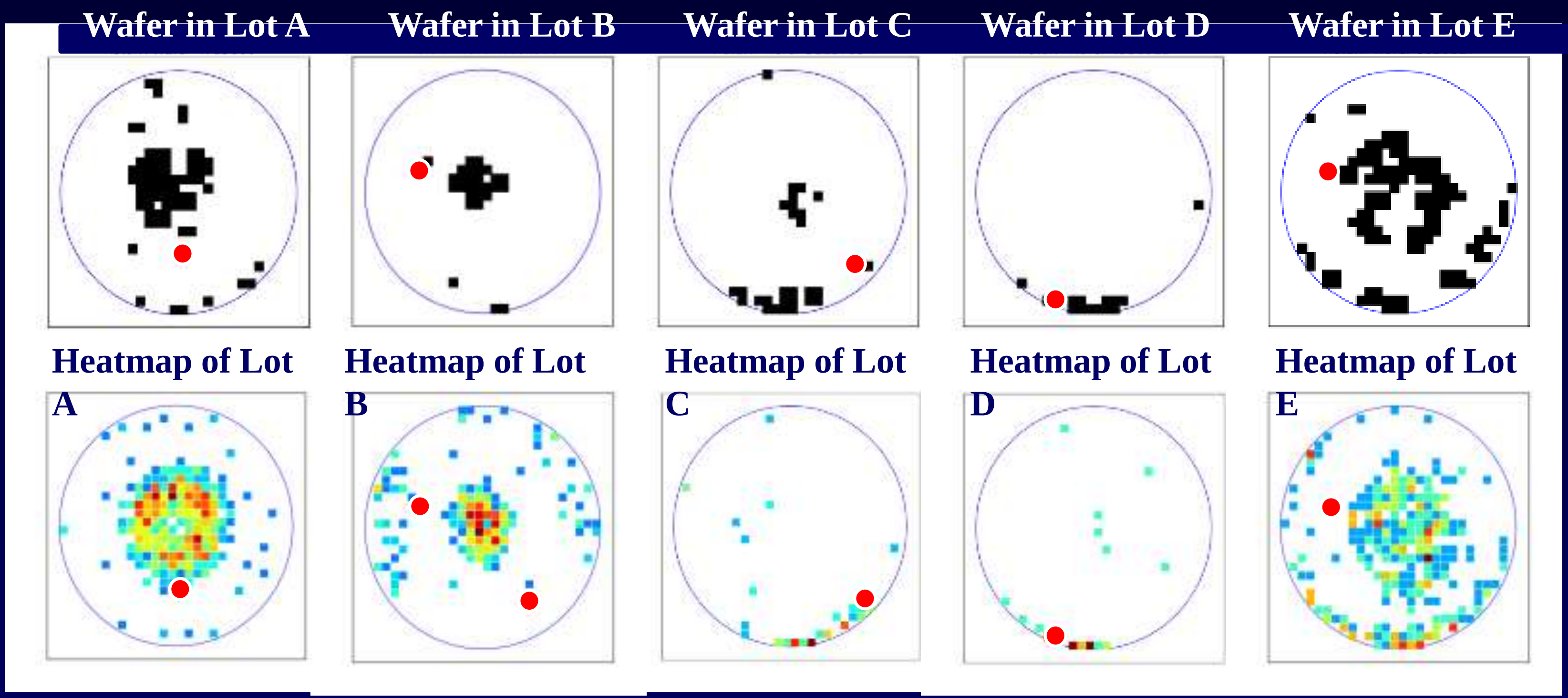
- After initial learning, 100 tests produced by the combined rule macro cover 4 out of 5 assertions
- Refining the rules result in coverage improvement
  - All 5 assertions are hit and the coverage increase in iteration 1 and 2, 100 tests each iteration

Search for a test perspective



- Given a wafer of interest, a set of tests, and a set of wafers
  - For example, the wafer contains a customer return
- Find a test perspective (a subset of tests)
- Such that the wafer shows abnormal failing pattern
- Output the test perspective and the wafer map for further analysis

Customer return analysis



- Applied to analyze customer returns from an automotive SoC product line
- Extract abnormal wafer maps for further inspection

Summary

- Data mining is not a one-step task
  - It is an iterative process
  - In each iteration, the goal is to discover interpretable and actionable knowledge
- Data mining is not fully automatic
  - It provides guides to user
  - Manual inspection and decision is required
- Effective data mining cannot be implemented without some domain knowledge
  - Feature generation is often the key
  - Methodology development is crucial
- Data mining is best for improving efficiency
  - User takes a long time to solve the problem
  - Data mining make the process much faster