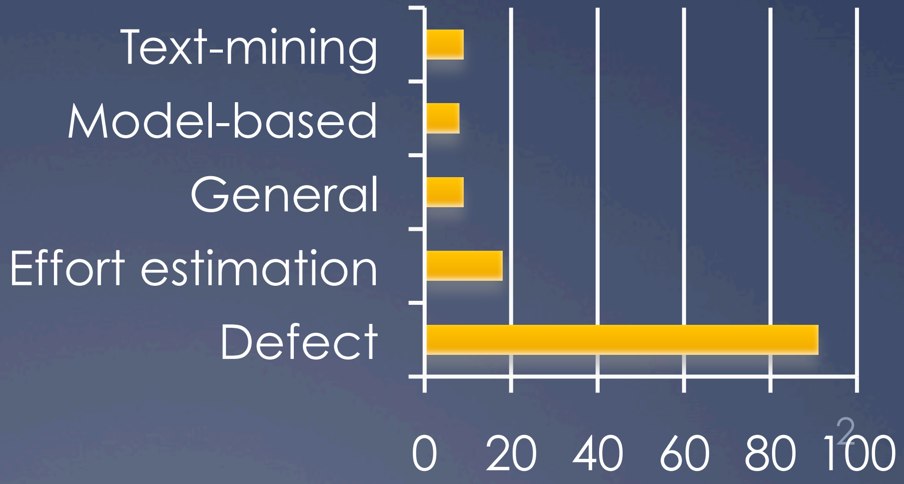
Tim Menzies, WVU, USA
Forrest Shull, Fraunhofer , USA
(with John Hoskings, UoA, NZ)
Jan 27-2011

Empirical Software Engineering, Version 2.0

About us

- Curators of large repositories of SE data
  - Searched for conclusions
- Shull: NSF-funded CeBase 2001- 2005
  - No longer on-line
- Menzies: PROMISE 2006-2011
  - If you publish, offer data used in that pub
  - http://promisedata.org/data
- Our question:
  - What's next?



Summary

- We need to do more "data mining"
  - Not just on different projects
  - But again and again on the same project
- And by "data Mining" we really mean
  - Automated agents that implement
    - prediction
    - monitoring
    - diagnosis,
    - Planning
    - Adaptive business intelligence

Adaptive Business Intelligence

* learning, and re-learning,
* How to….
  * Detect death march project
  * Repair death march projects
  * Find best sell/buy point for software artifacts
  * Invest more (or less) in staff training/dev programs
  * Prioritize software inspections
  * Estimate development cost
  * Change development costs
  * etc

This talk

* A plea for industrial partners to join in
* A roadmap for my next decade of research
  * Many long term questions
  * A handful of new results

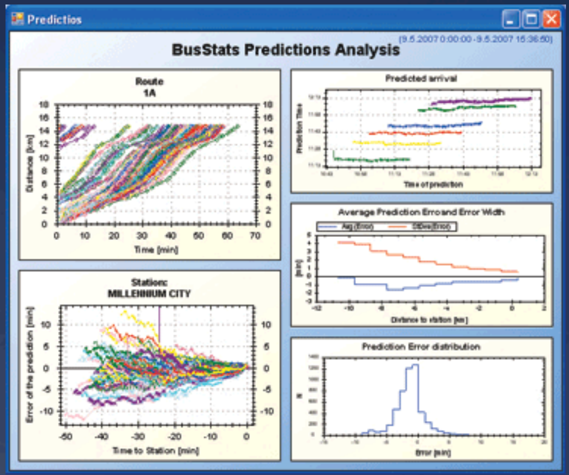
Data Mining & Software Engineering

So many applications of data mining to SE

- Process data
  - Input: Developer skills, platform stability
  - Output: effort estimation
- Social data
  - Input: e.g. which tester do you most respect?
  - Output: predictions of what bugs gets fixed first
- Product data
  - Input: static code descriptions
  - Output: defect predictors
- Trace data
  - Input: what calls what?
  - Output: call sequences that lead to a core dump
- Usage data
  - Input: what is everyone using?
  - Output: recommendations on where to browse next
- Any textual form
  - Input: text of any artifact
  - Output: e.g. fault localization

The State of the Art

* If data collected, then usually forgotten
* Dashboards : visualizations for feature extraction; intelligence left the user
* MapReduce, Hadoop et. al : systems support for massive, parallel execution.
  * http://hadoop.apache.org
  * Implements the bus, but no bus drivers
* Many SE data mining publications
  * e.g. Bird, Nagappan, Zimmermann and last slide
  * But, no agents that recognize when old models are no longer relevant,
  * Or to repair old models using new data



hadoop

Of course, DM gets it wrong, sometimes

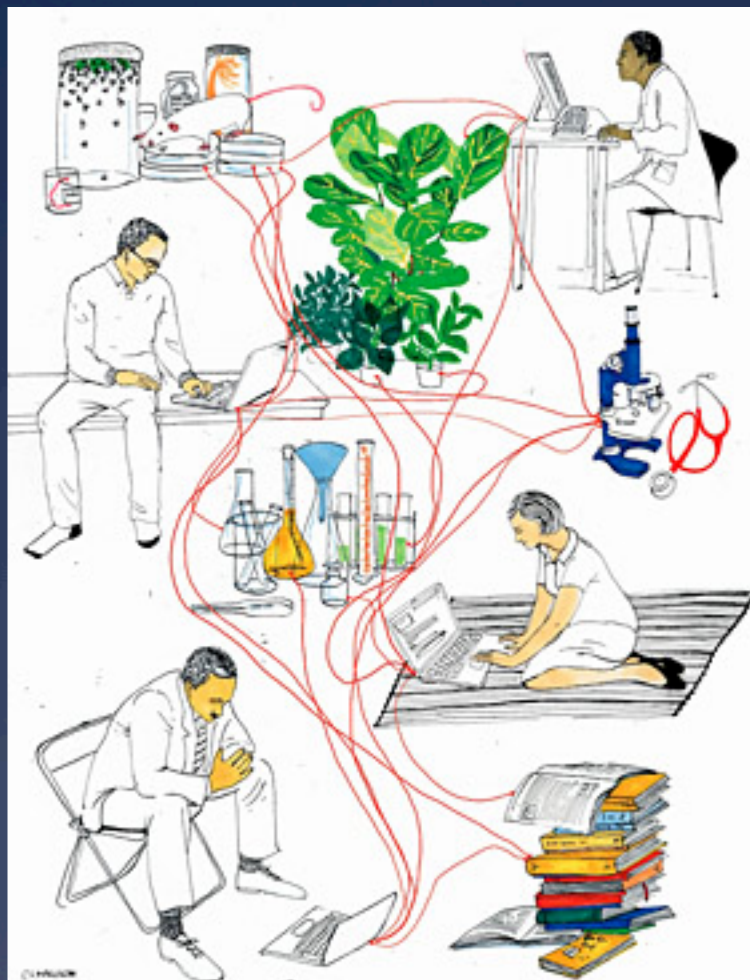
- Heh, nobody's perfect
- E.g. look at all the mistakes people make:
  - Wikipidea: list of cognitive biases
  - 38 decision making biases
  - 30 biases in probability
  - 18 social biases
  - 10 memory biases
- At least with DM, can repeat the analysis, audit the conclusion.

- Create agent communities, each with novel insights and limitations
  - Data miners working with humans
  - See more together than separately
  - Partnership

Does this change empirical SE research?

Ben Shneiderman, Mar'08

- The growth of the World Wide Web ... continues to reorder whole disciplines and industries. ...
- It is time for researchers in science to take network collaboration to the next phase and reap the potential intellectual and societal payoffs.
  - -B. Shneiderman.
  - Science 2.0.
  - Science, 319(7):1349–1350, March 2008



SCIENCE 2.0: YOU SAY YOU WANT A REVOLUTION?

The collaborative online tools people are using in other parts of their lives, such as Facebook, YouTube, and blogs, are roiling the disciplined world of scientific communication.

A proposal

- Add roller skates to software engineering
- Always use DM (data mining) on SE data

What's the difference?

SE research v1.0

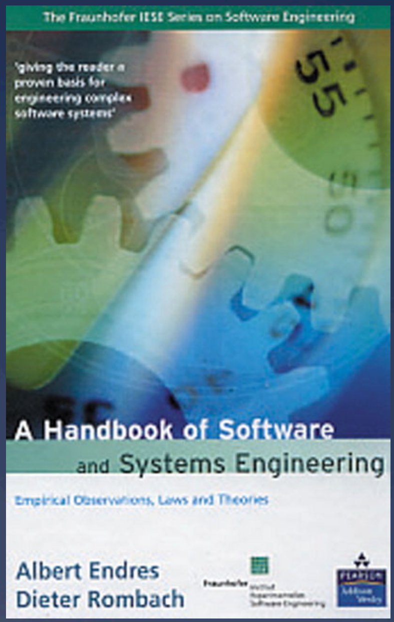
* Case studies
  * Watch, don't touch
* Experiments
  * Vary a few conditions in a project
* Simple analysis
  * A little ANOVA, regression, maybe a t-test

SE research v2.0

* Data generators
  * Case studies
  * Experiments
* Data analysis
  * 10,000 of possible data miners
* Crowd-sourcing
  * 10,000 of possible analysts

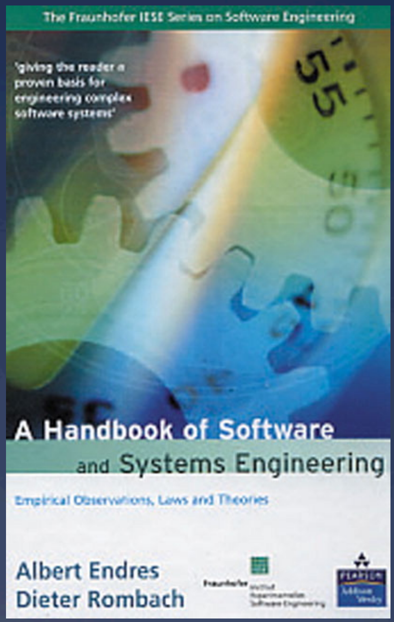
Value-added (to case-study-based research)

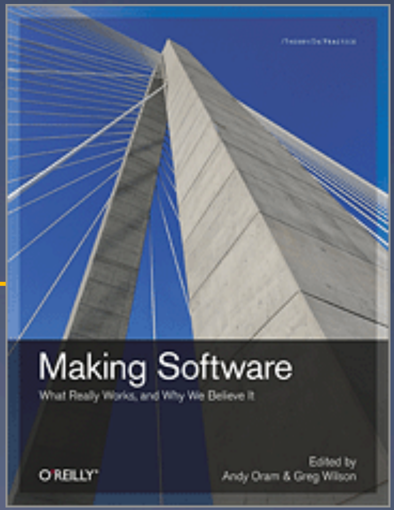
* Case studies: powerful for defining problems, highlighting open issues
* Has documented 100s of candidate methods for improving SE development
* e.g. Kitchenham et. Al IEEE TSE, 2007,
  * Cross versus Within-Company Cost Estimation
  * Spawned a sub-culture of researchers
    * checking if what works <u>here </u>also works <u>there</u>.



Case-Study-based Research: Has Limits

- Too slow
  - Years to produce conclusions
  - Meanwhile, technology base changes
- Too many candidate methods
  - No guidance on what methods to apply to particular projects
- Little generality
  - Zimmermann et. al, FSE 2009
    - 662 times : learn <u>here</u>, test <u>there</u>
    - Worked in 4% of pairs
  - Many similar no-generality results
    - Chpt1, Menzies & Shull





Case-studies + DM = Better Research

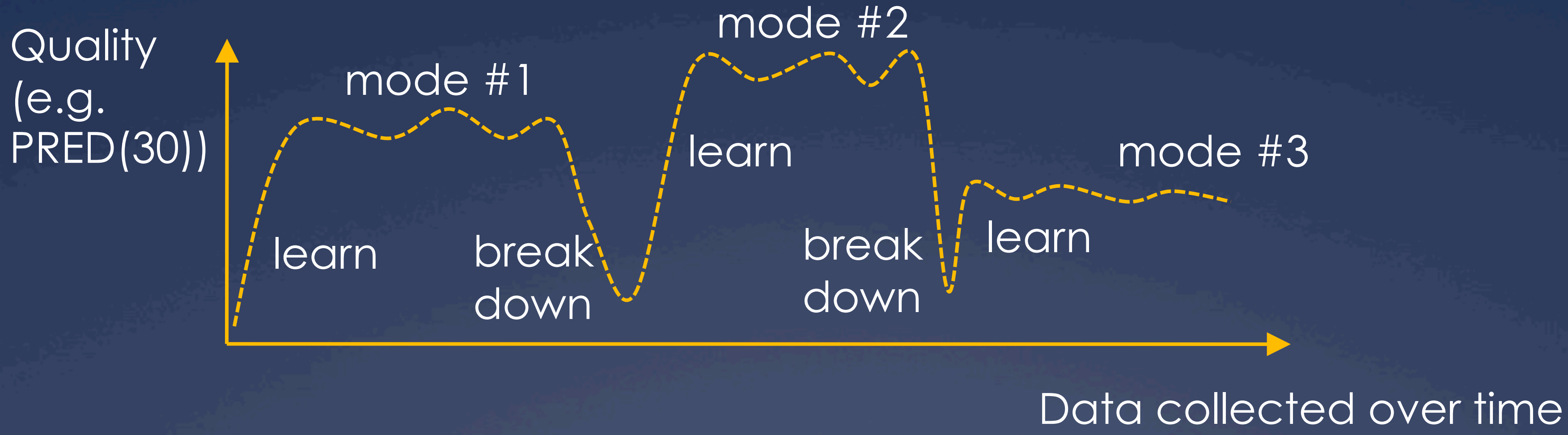
* Propose a partnership between
  * case study research
  * And data mining
* Data mining is stupid
  * Syntactic, no business knowledge
* Case studies are too slow
  * And to check for generality? Even slower
* Case study research (on one project) to raise questions
  * Data mining (on many projects) to check the answers

Adaptive Agents

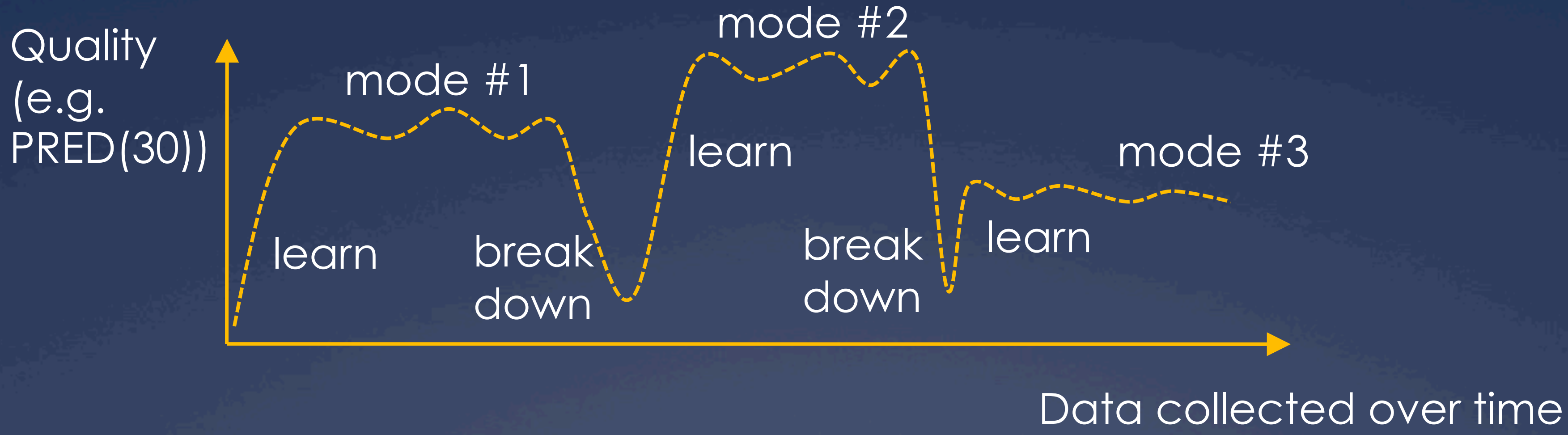
Need for adaptive agents

* No general rules in SE
  * Zimmermann FSE, 2009
* But general methods to find the local rules
* Issues:
  * How quickly can we learn the local models?
  * How to check when local models start failing?
  * How to repair local models?
* An adaptive agent watching a stream of data, learning and relearning as appropriate

Agents for adaptive business intelligence

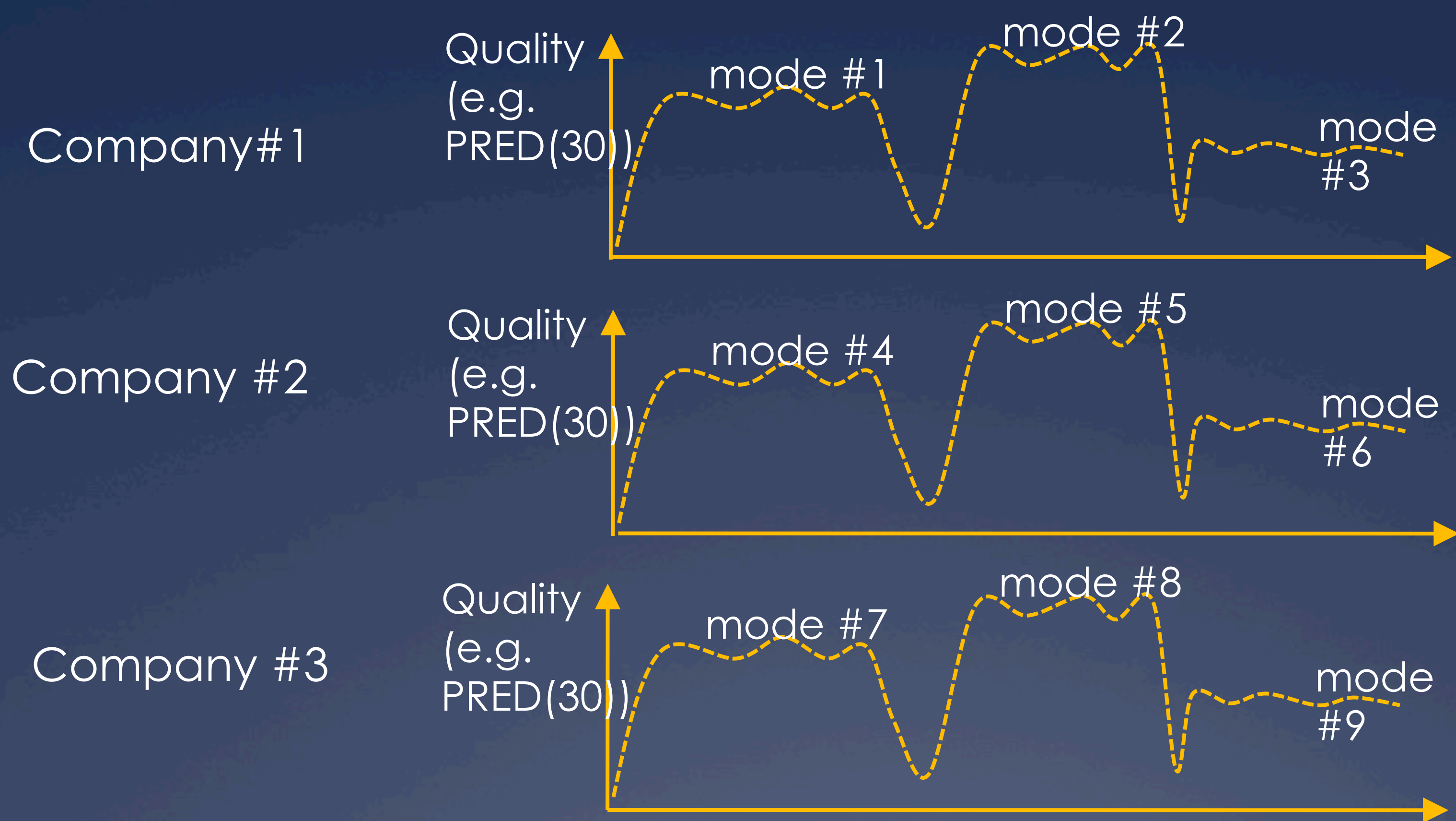


Agents for adaptive business intelligence



* What is different here?
  * Not "apply data mining to build a predictor"
  * But add monitor and repair tools to recognize and handle the breakdown of old predictors
  * Trust = data mining + monitor + repair

If crowd sourcing



With DM, we could recognize that e.g. 1=4=7

- i.e. when some "new" situation has happened before
- So we'd know what experience base to exploit

Research Questions. How to handle….

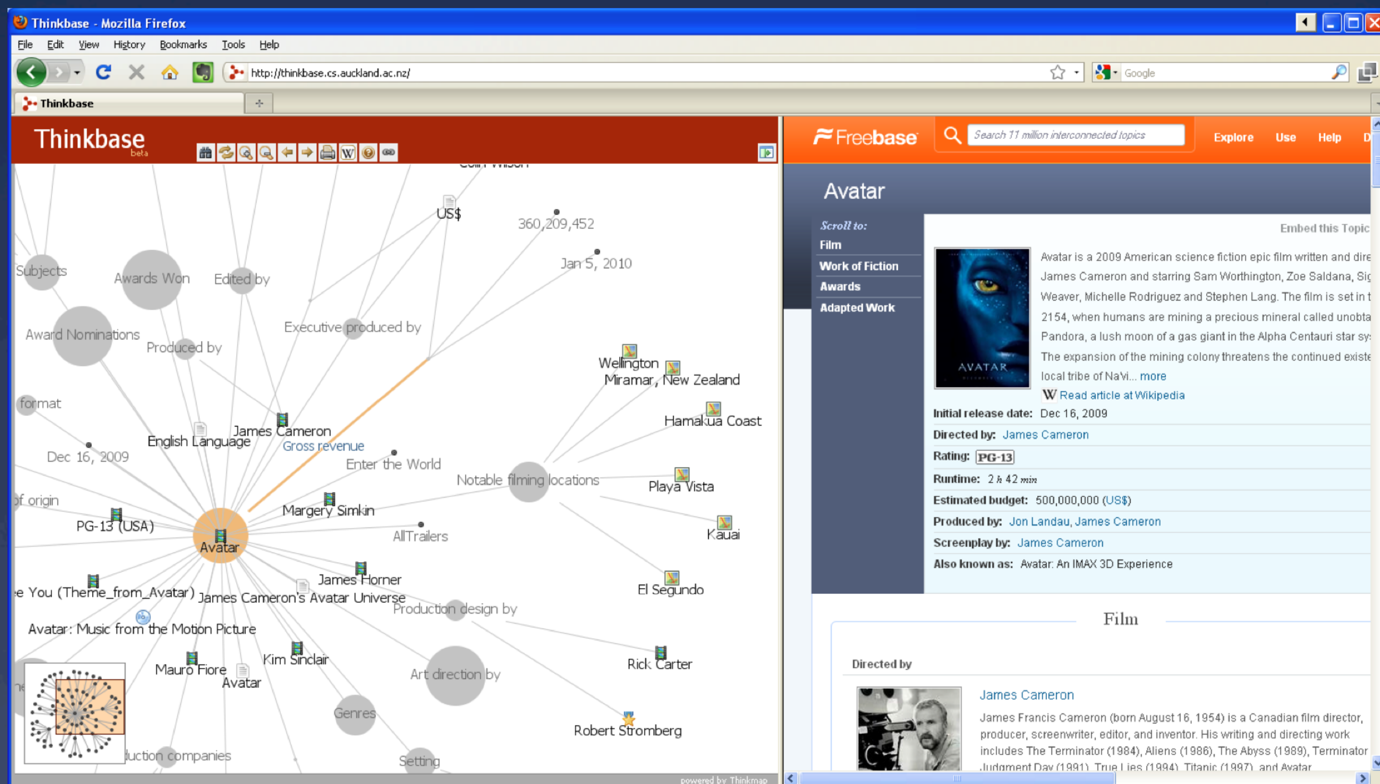
* Anonymization
  * Make data public, without revealing private data
* Volume of data
  * Especially if working from "raw" project artifacts
  * Especially if crowd sourcing
* Explanation : of complex patterns
* Noise: from dad data collection
* Mode recognition
  * Is when new is stuff is new, or a a repeat of old stuff
* Trust : you did not collect the data
  * Must surround the learners with assessment agents
    * Anomaly detectors
    * Repair

Most of the technology required for this approach can be implemented via data mining

So it would scale to large data sets

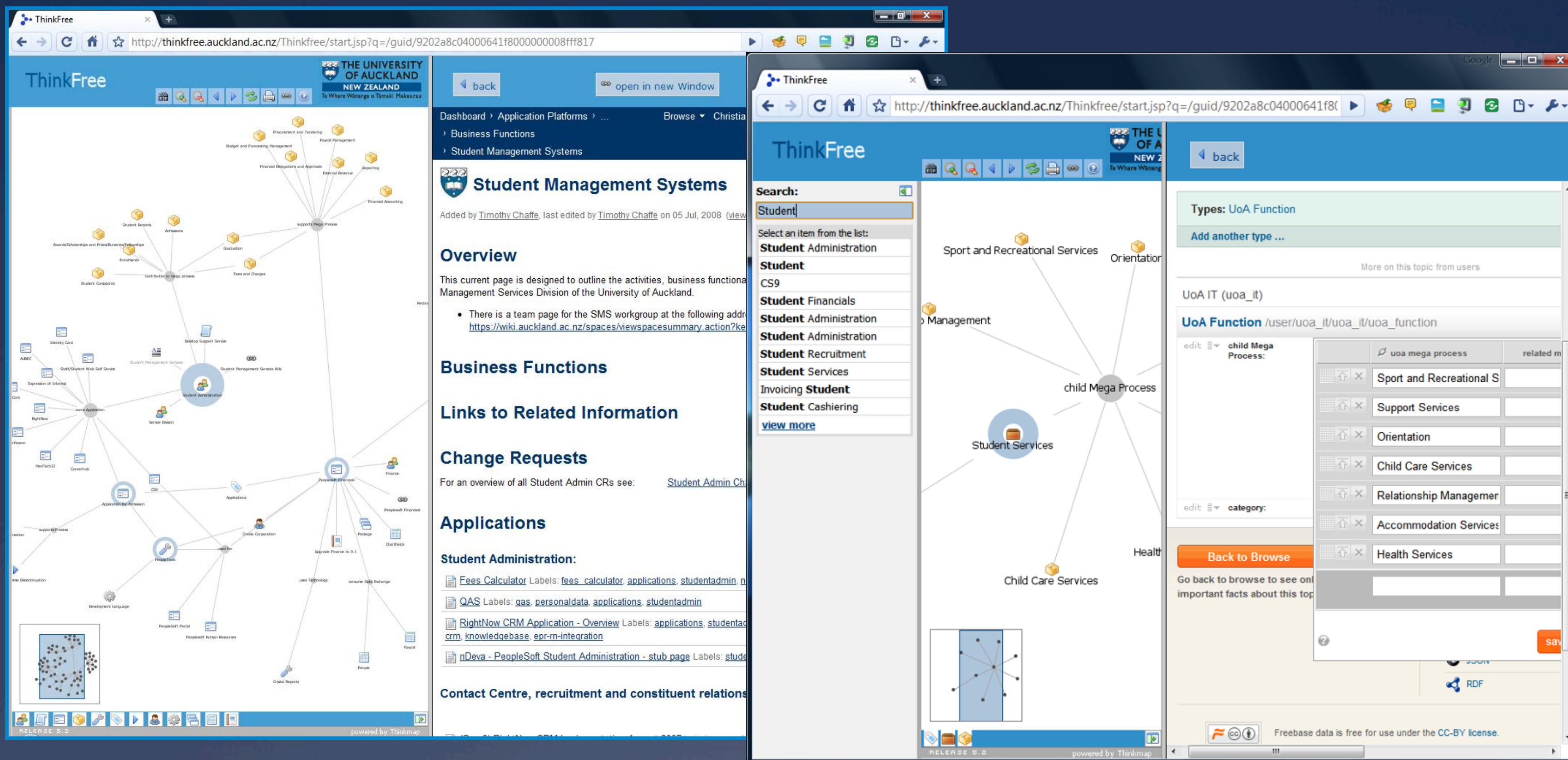
Organizing the artifacts

Visual Wiki ("Viki") concept



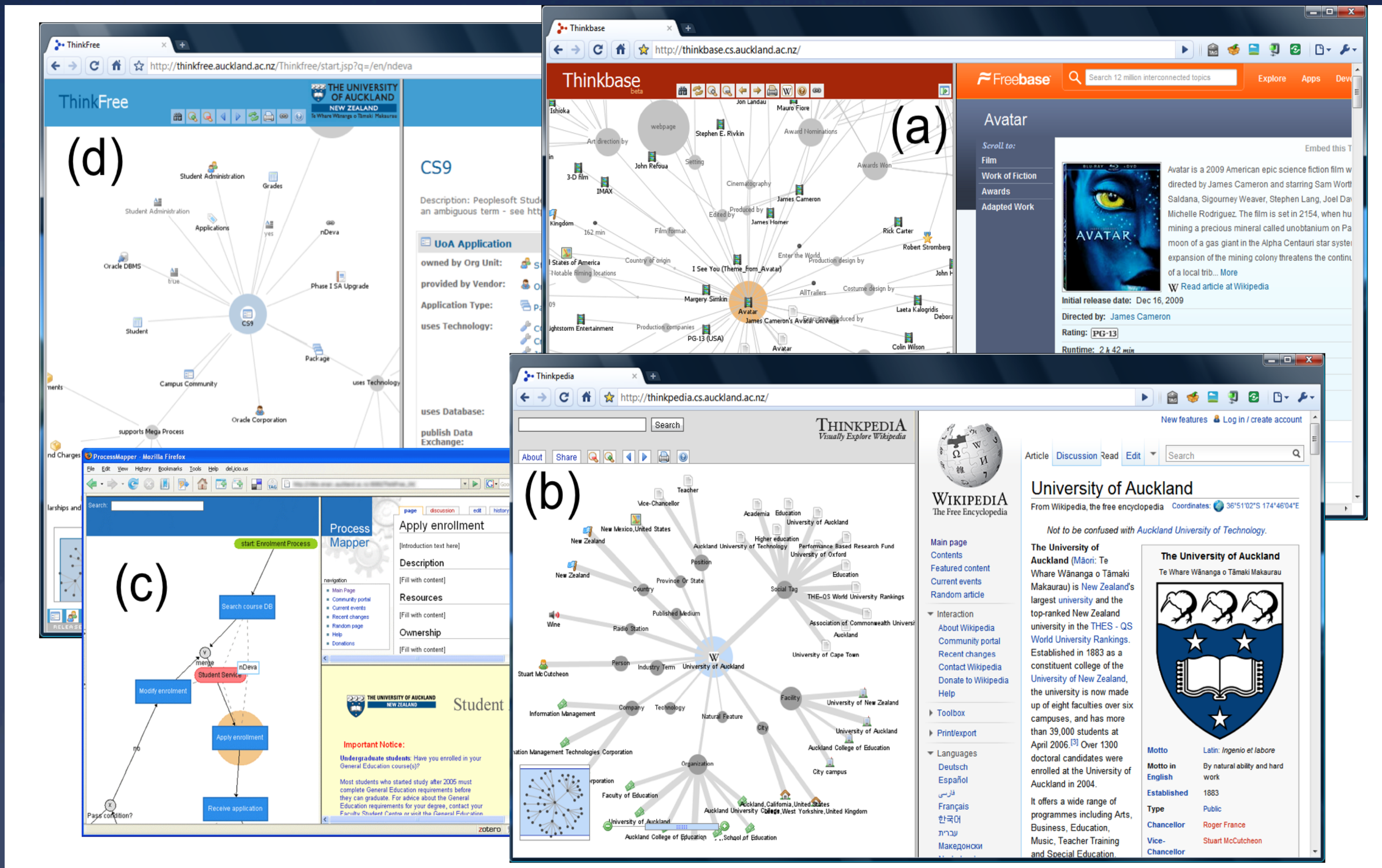
Visualization	Mapping	Text
Concept		

Enterprise Artifacts example



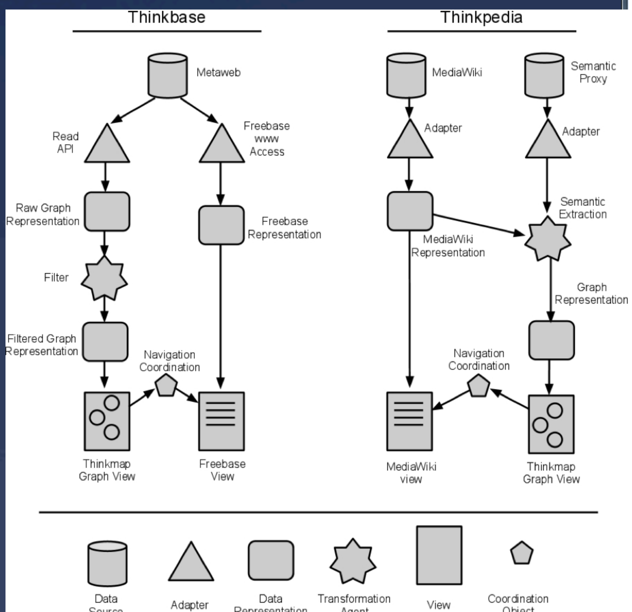
- Add documents; organize; search; navigate
- Edit properties, documents, add links, extract links

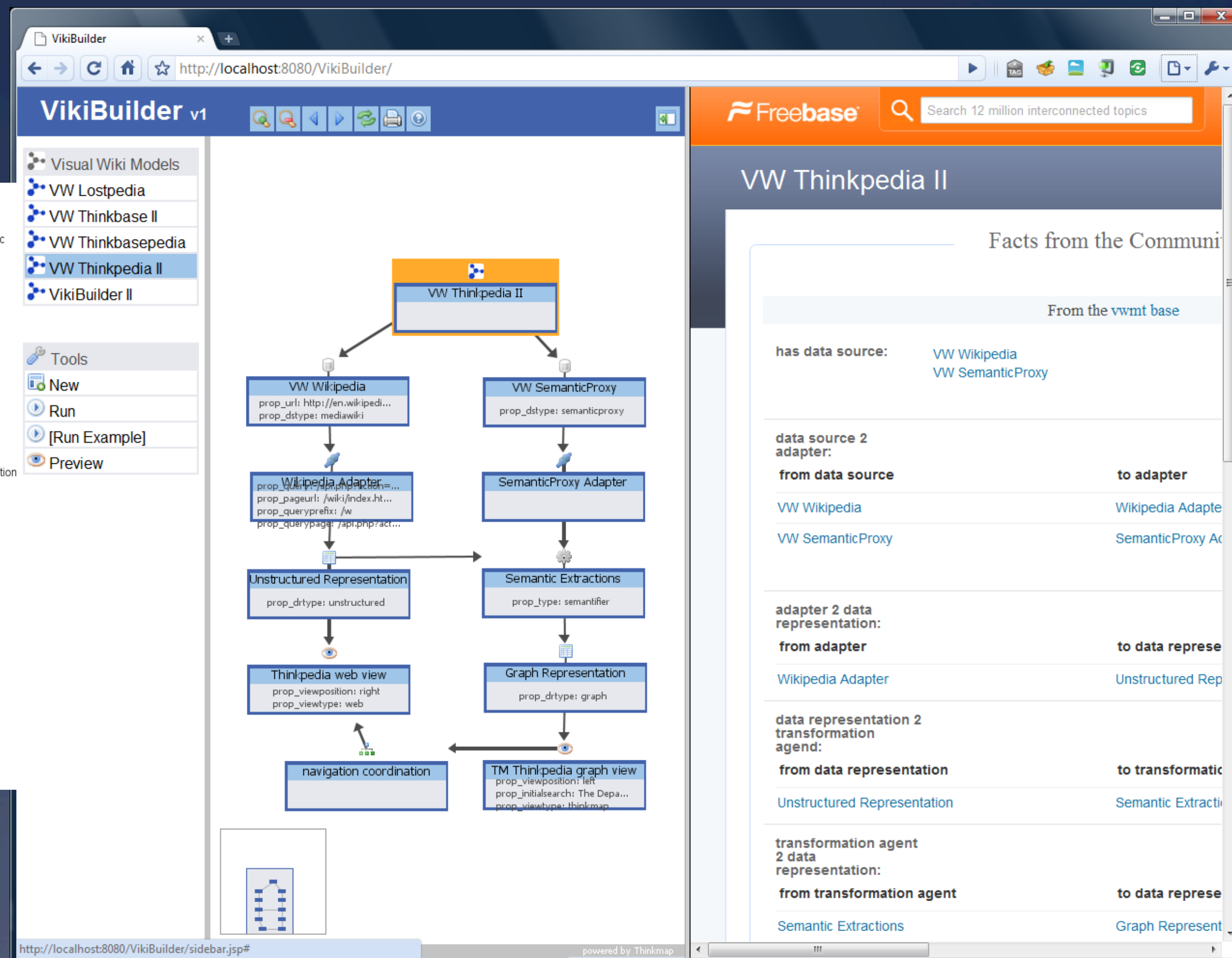
Various Vikis



* Bottom left: business process descriptions

VikiBuilder – generating Vikis



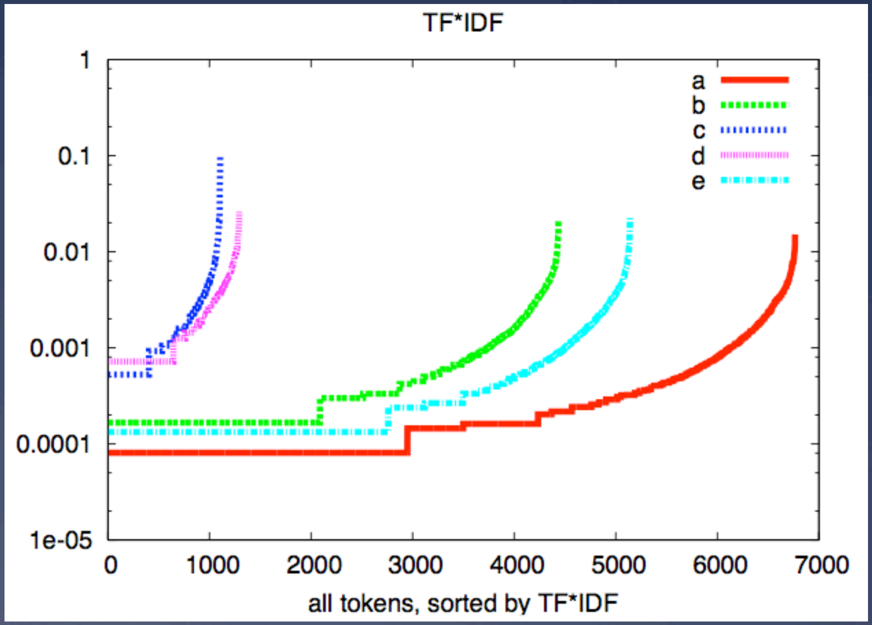


Text mining

* Key issue: dimensionality reduction
* In some domains, can be done in linear time

$$w_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_i}\right)$$

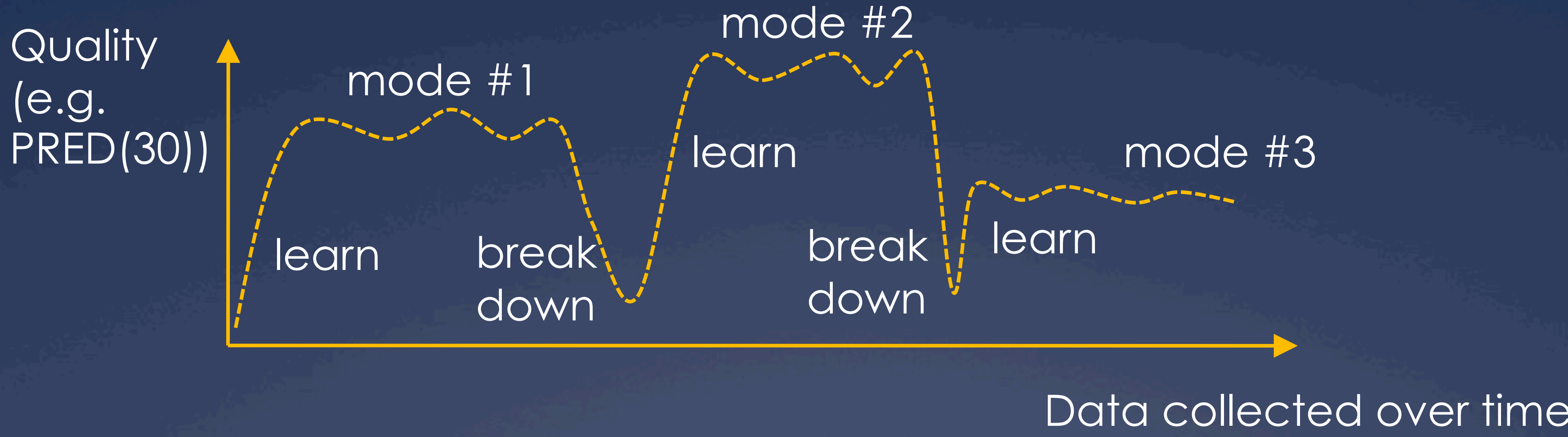
$tf_{i,j}$ = number of occurrences of i in j
df_i = number of documents containing i
N = total number of documents



* Use standard data miners, applied to top 100 terms in each corpus

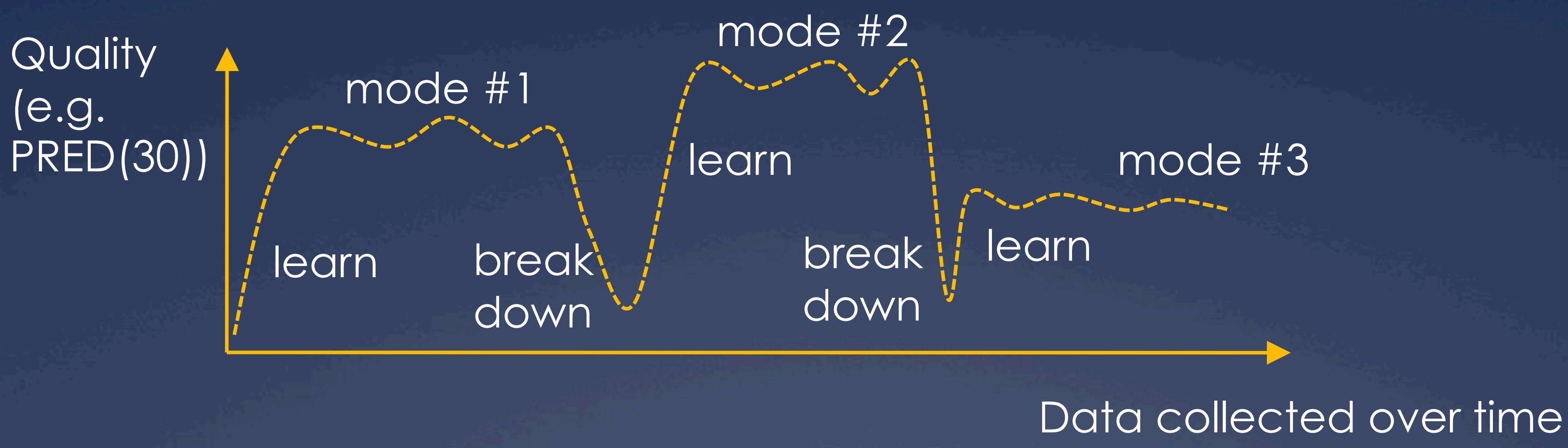
Details

Agents for adaptive business intelligence



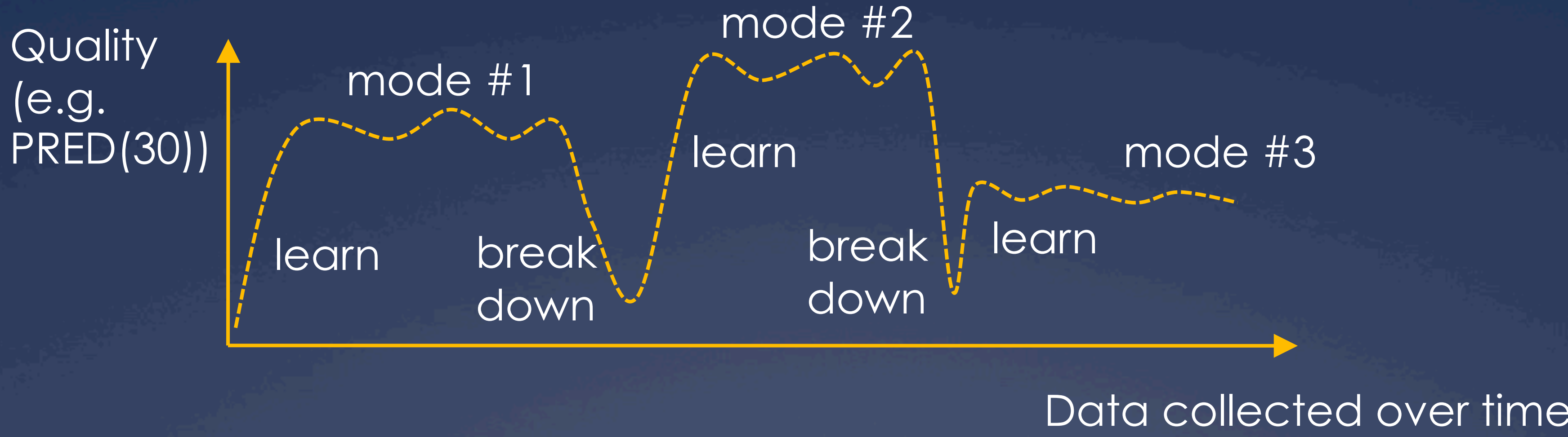
- Q1: How to learn faster?
  - Technology: active learning: reflect on examples to date to ask most informative next question
- Q2: How to recognize breakdown?
  - Technology: bayesian anomaly detection
  - Focusing on frequency counts of contrast sets

Agents for adaptive business intelligence



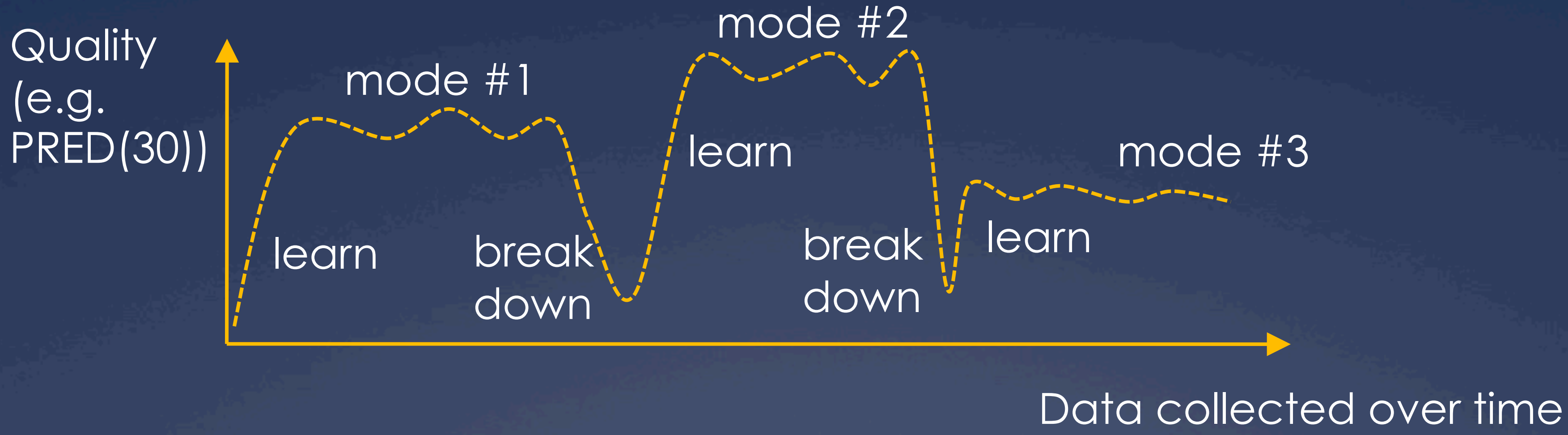
- Q3: How to classify a mode?
  - Recognize if you've arrived at a mode seen before
  - Technology: Bayes classifier
- Q4: How to make predictions?
  - Using the norms of a mode, report expected behavior
  - Technology: table look-up of data inside Bayes classifier

Agents for adaptive business intelligence



* Q5: What went wrong? (diagnosis)
  * Delta between current and prior, better, mode
* Q6: What to do? (planning)
  * Delta between current and other, better, mode
* Technology: contrast set learning

Agents for adaptive business intelligence



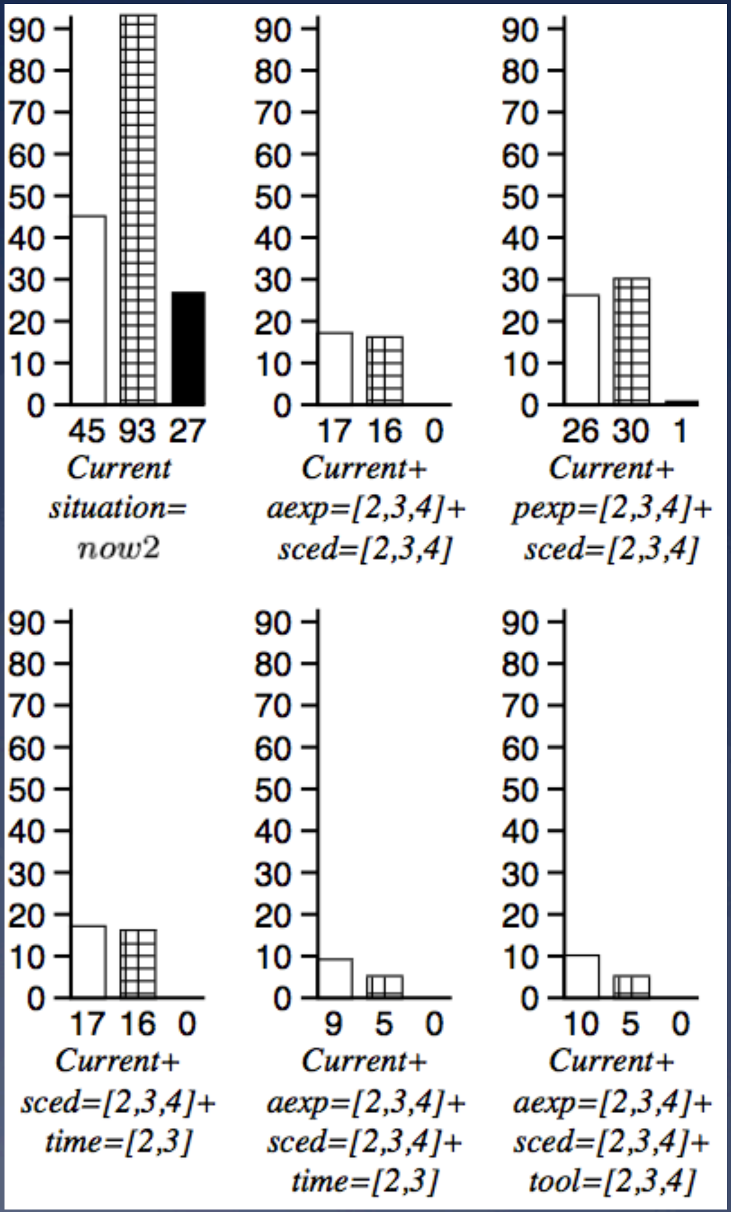
- Q7: How to understand a mode (explanation)
  - Presentation of essential features of a mode
- Technology: contrast set learning

Bits and pieces

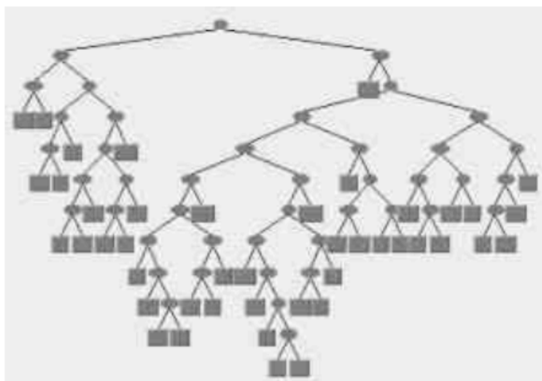
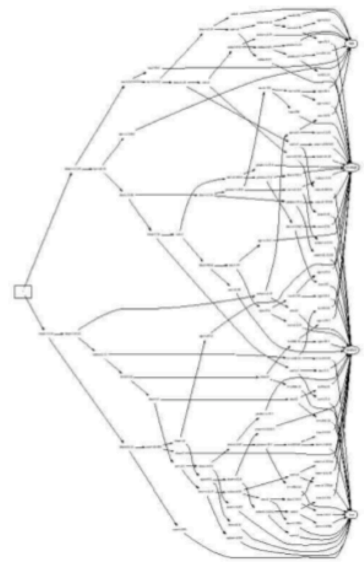
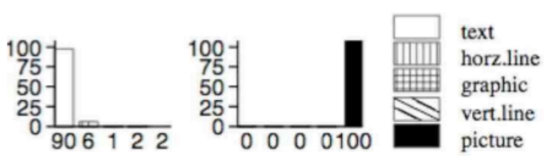
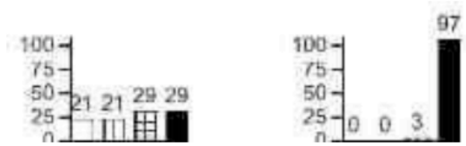
Prototypes

Contrast set learning

* Minimal contrast set learning = diagnosis &planning
* A decision tree with weighted leaves
  * Treatment = decisions that prune branches
    * Culls bad weights
    * Keeps good weights
* E.g. Simulator + C4.5 + 10-way
  * 10 * 1000 node trees
  * TAR1: tiny rules: decision on 4 ranges
* Why so small?
  * Higher decisions prune more branches
    * touch fewer nodes

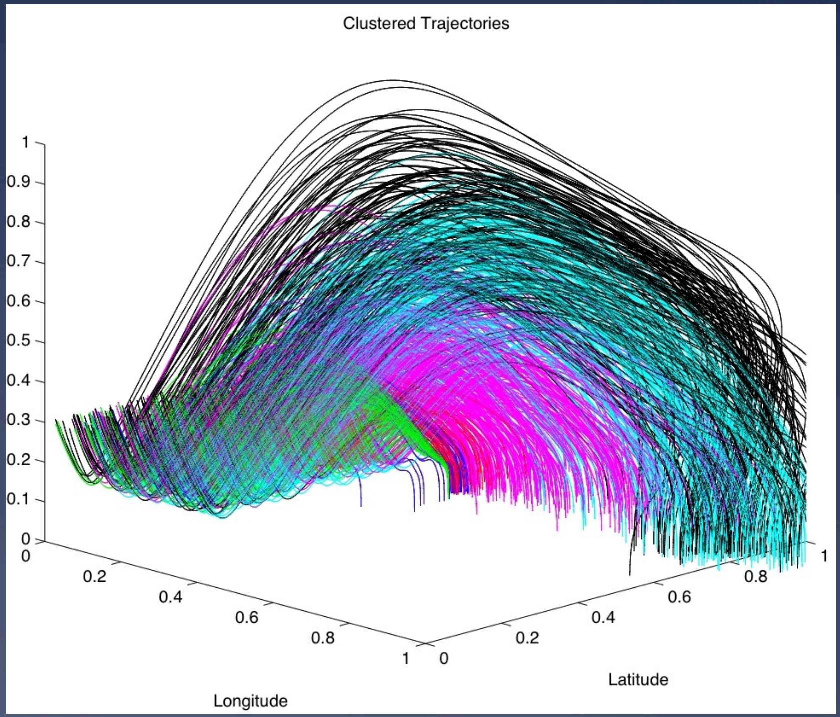


Contrast Set Learning → Succinct Explanations

find graphics on a page from 11 features	find good housing in Boston
	
	
$34 \le height < 86 \wedge$ $3.9 \le mean\_tr < 9.5$	$6.7 \le RM < 9.8 \wedge$ $12.6 \le PTRATION < 15.9$

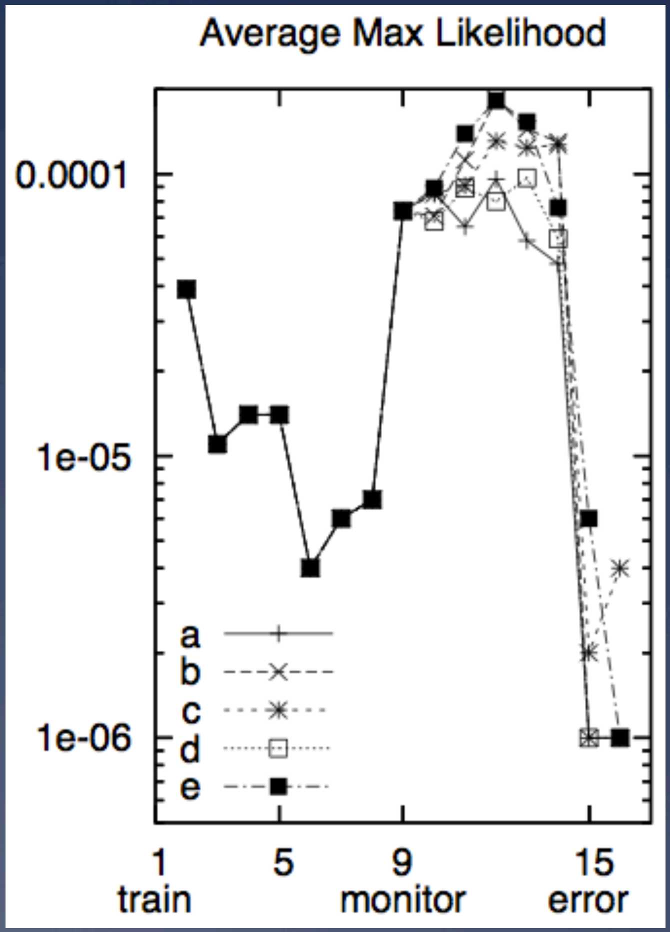
Contrast Set Learning (10 years later)

- No longer a post-processor to a decision tree learner
  - TAR3: Branch pruning operators applied directly to discretized data
- Summer'09
  - Shoot 'em up at NASA AMES
  - State-of-the-art numerical optimizer
  - TAR3
    - Ran 40 times faster
    - Generated better solutions
- Powerful succinct explanation tool



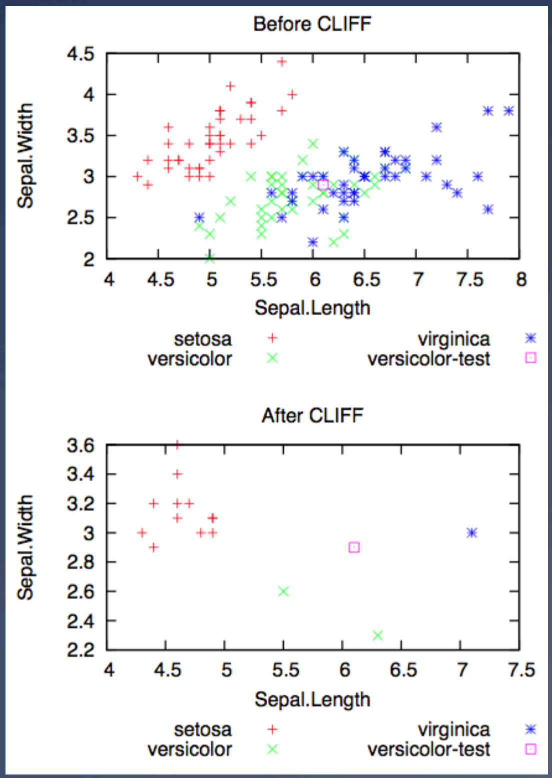
Contrast Set Learning ➔ Anomaly Detection

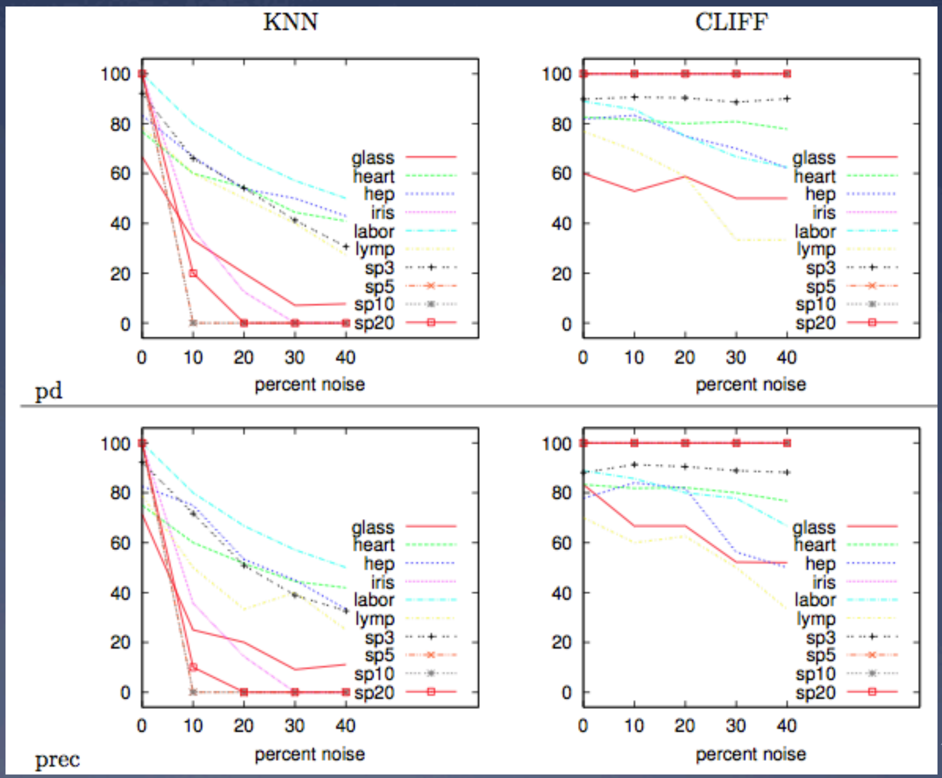
- Recognize when old ideas are now out-dated
- SAWTOOTH:
  - read data in "eras" of 100 instances
  - Classify all examples as "seen it"
- SAWTOOTH1:
  - Report average likelihood of examples belong to "seen it"
  - Alert if that likelihood drops
- SAWTOOTH2:
  - Back-end to TAR3
  - Track frequency of contrast sets
  - Some uniformity between contrast sets and anomaly detection



Contrast sets ➔ noise management

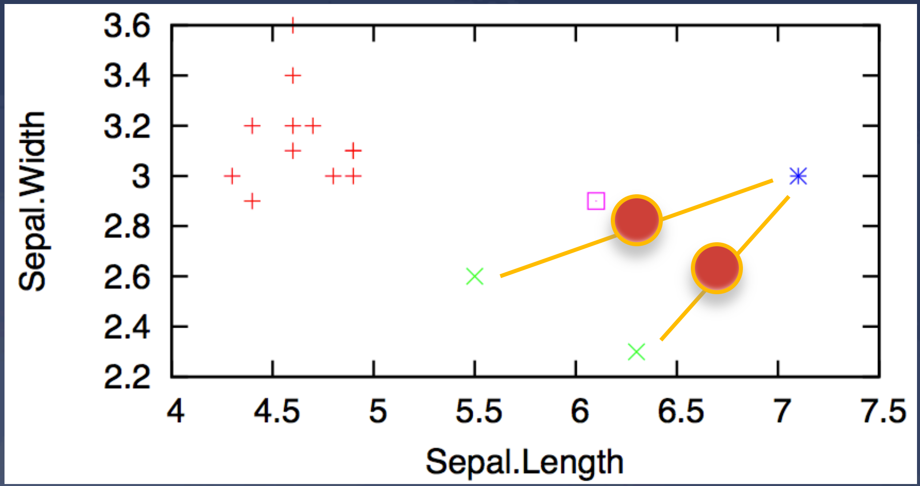
* CLIFF: post-processor to TAR3
  * Linear time instance selector
* Finds the attribute ranges that change classification
* Delete all instances that lack the "power ranges"





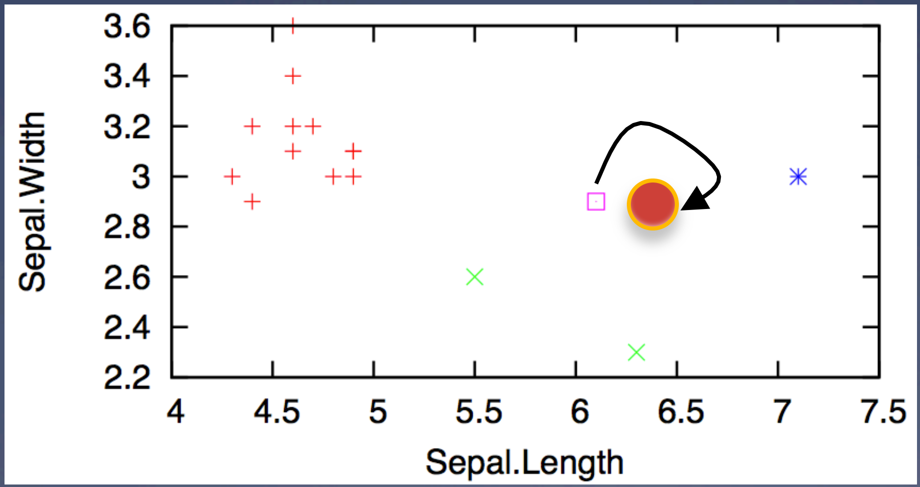
Contrast Sets → CLIFF → Active Learning

- Many examples, too few labels
- Reduce the time required for business users to offer judgment on business cases
- Explore the reduced space generated by CLIFF.
  - Randomly ample the instances half-way between different classes
- Fast (in the reduced space)



Contrast sets → CLIFF → Statistical databases

* Anonymize the data: Preserving its distributions
* For KNN, that means keep the boundaries between classes
  * Which we get from CLIFF
* Also, CLIFF empties out the instance space
  * Leaving us space to synthesize new instances



And so…

We seek industrial partners

1. That will place textual versions of their products in a wiki
2. That will offer joins of those products to quality measures
3. That will suffer us interviewing their managers, from time to time, to learn the control points.

(Note: 1,2 can be behind your firewalls.)

In return, we offer

* Agents for
  * automatic, adaptive, business intelligence
  * that tunes itself to your local domain

Questions?
Comments?