Understanding Machine Learning for Empirical Software Engineering

tim@menzies.us
http://menzies.us
usa, wvu, csee, ai
march 2012

Do you understand data mining?

- Can you map between data miners and business needs?
- Can you make them run fast? Scale to large data sets?
  - Linear, or logLinear, approximations
  - Random sampling
- Can you code them?
  - in 1,000 LOC (or less)?
- Can you take M data mining methods and remix them?
  - Not M methods
  - But $2^{M!}$ Combos
- Can you explain them to other people?
  - Empower them to explore new miners for new domains?
- Can you avoid bogus complexity?

More complex methods aren't making us better

- Dejaeger, K.; Verbeke, W.; Martens, D.; Baesens, B.; , "Data Mining Techniques for Software Effort Estimation: A Comparative Study," *Software Engineering, IEEE Transactions, doi: 10.1109/TSE.2011*
  - Simple, understandable techniques like Ordinary least squares regressions with log transformation of attributes and target perform as well as (or better than) nonlinear techniques.
- Hall, T.; Beecham, S.; Bowes, D.; Gray, D.; Counsell, S.; , "A Systematic Review of Fault Prediction Performance in Software Engineering," *Software Engineering, IEEE Transactions,* doi: 10.1109/TSE.2011.103
  - Support Vector Machine (SVM) perform less well.
  - Models based on C4.5 seem to under-perform if they use imbalanced data.
  - Models performing comparatively well are relatively simple techniques that are easy to use and well understood.. E.g. Naïve Bayes and Logistic regression

And we aren't so good at the simpler methods

- Data miners (WEKA, R, MATLAB, …)
  - Quick and easy to use
  - Quick and easy to use … poorly
- Hall (2011) :
  - IEEE TSE pre-prints
  - Large survey on defect prediction via data mining.
  - What explain the variance in performance results?

A. How the data is mined (the algorithms)?
B. What data is mined
C. Who does the data mining

- Overwhelmingly:
  - "C"
  - see Shepperd (2011)
- Not enough to use these tools black box
  - Not enough to poke & pray

More "data mining" and less "algorithm mining"

- We do data mining not to study algorithms.
  - But to study data
- Our results should be insights about data,
  - not trivia about (say) decision tree algorithms
- Besides, the thing that most predicts for performance is the data, not the algorithm,
  - Pedro Domingos and Michael J. Pazzani, On the Optimality of the Simple Bayesian Classifier under Zero-One Loss, Machine Learning, Volume 29, number 2-3, pages 103-130, 1997

Table 1. Classification accuracies and sample standard deviations, averaged over 20 random training/test splits. "Bayes" is the Bayesian classifier with discretization and "Gauss" is the Bayesian classifier with Gaussian distributions. Superscripts denote confidence levels for the difference in accuracy between the Bayesian classifier and the corresponding algorithm, using a one-tailed paired t test: 1 is 99.5%, 2 is 99%, 3 is 97.5%, 4 is 95%, 5 is 90%, and 6 is below 90%.

Data Set	Bayes	Gauss	C4.5	PEBLS	CN2	Def.
Audiology	73.0±6.1	73.0±6.1^{6}	72.5±5.8^{6}	75.8±5.4^{3}	71.0±5.1^{5}	21.3
Annealing	95.3±1.2	84.3±3.8^{1}	90.5±2.2^{1}	98.8±0.8^{1}	81.2±5.4^{1}	76.4
Breast cancer	71.6±4.7	71.3±4.3^{6}	70.1±6.8^{5}	65.6±4.7^{1}	67.9±7.1^{1}	67.6
Credit	84.5±1.8	78.9±2.5^{1}	85.9±2.1^{3}	82.2±1.9^{1}	82.0±2.2^{1}	57.4
Chess endgames	88.0±1.4	88.0±1.4^{6}	99.2±0.1^{1}	96.9±0.7^{1}	98.1±1.0^{1}	52.0
Diabetes	74.5±2.4	75.2±2.1^{6}	73.5±3.4^{5}	71.1±2.4^{1}	73.8±2.7^{6}	66.0
Echocardiogram	69.1±5.4	73.4±4.9^{1}	64.7±6.3^{1}	61.7±6.4^{1}	68.2±7.2^{6}	67.8
Glass	61.9±6.2	50.6±8.2^{1}	63.9±8.7^{6}	62.0±7.4^{6}	63.8±5.5^{6}	31.7
Heart disease	81.9±3.4	84.1±2.8^{1}	77.5±4.3^{1}	78.9±4.0^{1}	79.7±2.9^{3}	55.0
Hepatitis	85.3±3.7	85.2±4.0^{6}	79.2±4.3^{1}	79.0±5.1^{1}	80.3±4.2^{1}	78.1
Horse colic	80.7±3.7	79.3±3.7^{1}	85.1±3.8^{1}	75.7±5.0^{1}	82.5±4.2^{2}	63.6
Hypothyroid	97.5±0.3	97.9±0.4^{1}	99.1±0.2^{1}	95.9±0.7^{1}	98.8±0.4^{1}	95.3
Iris	93.2±3.5	93.9±1.9^{6}	92.6±2.7^{6}	93.5±3.0^{6}	93.3±3.6^{6}	26.5
Labor	91.3±4.9	88.7±10.6^{6}	78.1±7.9^{1}	89.7±5.0^{6}	82.1±6.9^{1}	65.0
Lung cancer	46.8±13.3	46.8±13.3^{6}	40.9±16.3^{5}	42.3±17.3^{6}	38.6±13.5^{3}	26.8
Liver disease	63.0±3.3	54.8±5.5^{1}	65.9±4.4^{1}	61.3±4.3^{6}	65.0±3.8^{3}	58.1
LED	62.9±6.5	62.9±6.5^{6}	61.2±8.4^{6}	55.3±6.1^{1}	58.6±8.1^{2}	8.0
Lymphography	81.6±5.9	81.1±4.8^{6}	75.0±4.2^{1}	82.9±5.6^{6}	78.8±4.9^{3}	57.3
Post-operative	64.7±6.8	67.2±5.0^{3}	70.0±5.2^{1}	59.2±8.0^{2}	60.8±8.2^{4}	71.2
Promoters	87.9±7.0	87.9±7.0^{6}	74.3±7.8^{1}	91.7±5.9^{3}	75.9±8.8^{1}	43.1
Primary tumor	44.2±5.5	44.2±5.5^{6}	35.9±5.8^{1}	30.9±4.7^{1}	39.8±5.2^{1}	24.6
Solar flare	68.5±3.0	68.2±3.7^{6}	70.6±2.9^{1}	67.6±3.5^{6}	70.4±3.0^{2}	25.2
Sonar	69.4±7.6	63.0±8.3^{1}	69.1±7.4^{6}	73.8±7.4^{1}	66.2±7.5^{5}	50.8
Soybean	100.0±0.0	100.0±0.0^{6}	95.0±9.0^{3}	100.0±0.0^{6}	96.9±5.9^{3}	30.0
Splice junctions	95.4±0.6	95.4±0.6^{6}	93.4±0.8^{1}	94.3±0.5^{1}	81.5±5.5^{1}	52.4
Voting records	91.2±1.7	91.2±1.7^{6}	96.3±1.3^{1}	94.9±1.2^{1}	95.8±1.6^{1}	60.5
Wine	96.4±2.2	97.8±1.2^{3}	92.4±5.6^{1}	97.2±1.8^{6}	90.8±4.7^{1}	36.4
Zoology	94.4±4.1	94.1±3.8^{6}	89.6±4.7^{1}	94.6±4.3^{6}	90.6±5.0^{1}	39.4

Data mining = data “carving”

- Data is like a block of marble,
  - waiting for a sculptor (that’s you)
  - to find the shape within
- To build a data miner, throw stuff away
  - Chip away the irrelevancies
  - To find what lies beneath.

Understanding data mining for SE

- SE information needs:
  - Uncovering trends in data;
  - Learning when to raise an alert;
  - Forecasting the future;
  - Summarizing the current situation;
  - Planning;
  - Modeling;
  - Benchmarking;
  - Running what-if queries.
- Standard machine learning algorithms:
  - Clustering,
  - Dendograms,
  - Active learning
  - Multi-objective optimization
  - Data stream mining,
  - Anomaly detectors,
  - Discretization,
  - Decision-tree learning,
  - Contrast rule learning,
  - Bayes classifiers,
  - Scenario generation and simulation

- While the above list of learning algorithms seems very long..
  - Once an analyst understand a set of core functionality of ML
  - Straight-forward to combine and tune and apply these algorithms
  - to a wide range of software engineering tasks.

For more information

- For answers to these questions:
  - What software engineering tasks can be helped by data mining?
  - What kinds of software engineering data can be mined?
  - How are data mining techniques used in software engineering?
- See Tao Xie's excellent Bibliography
  - "Mining Software Engineering Data"
  - http://goo.gl/14cAs

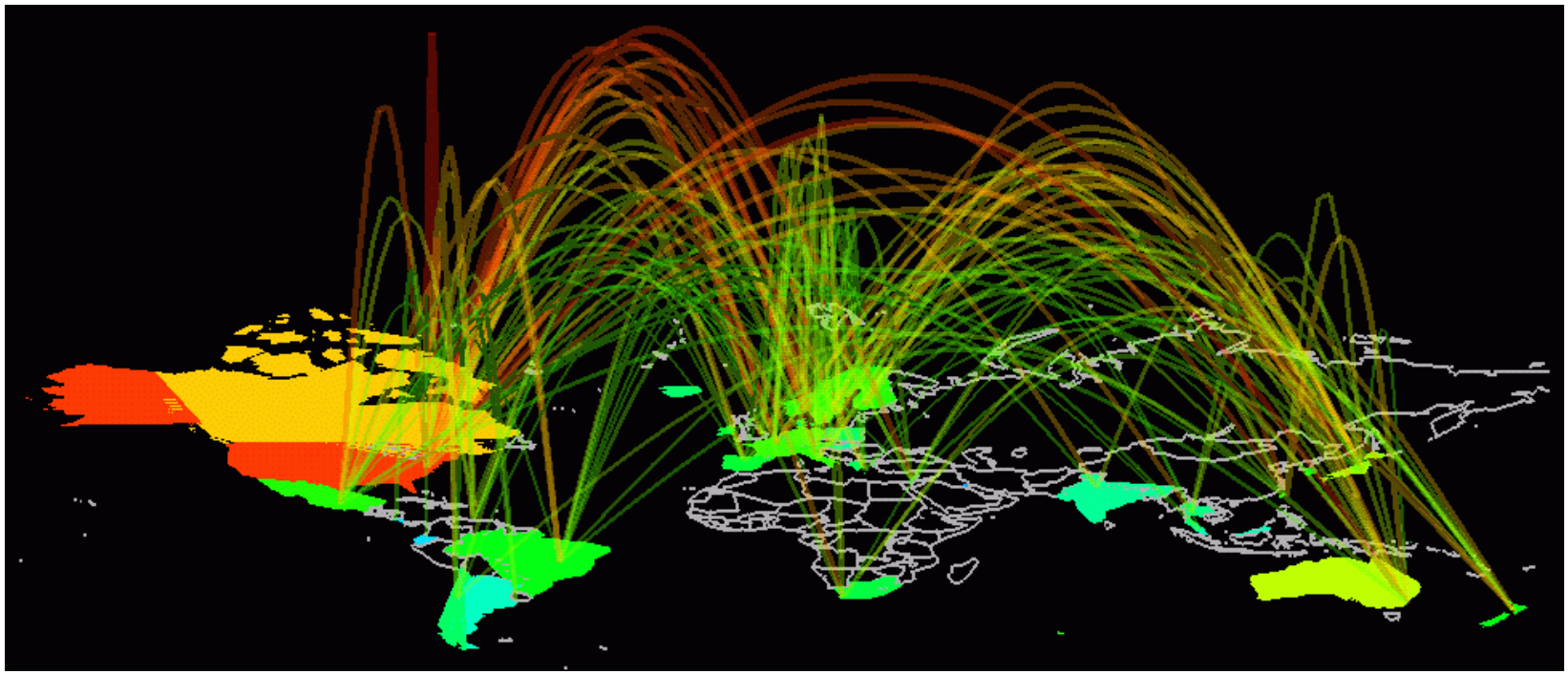
Roadmap

- Introduction
- Throwing stuff away
- Business info needs
- IDEA
- Dimensionality reduction
- Row reduction
- Column reduction
- Rule reduction
- Sanity Check
- The End

Roadmap

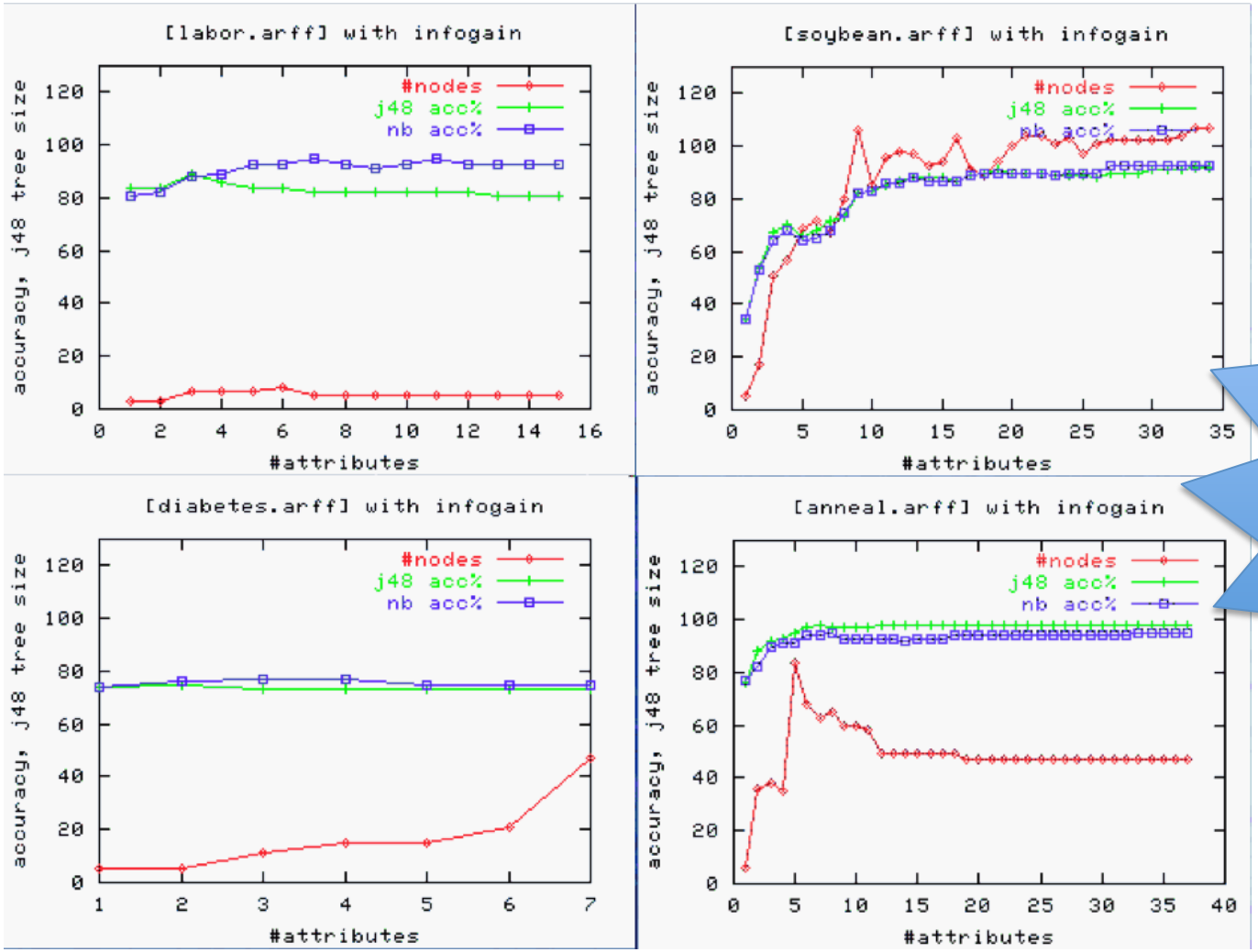
- Introduction
- Throwing stuff away
- Business info needs
- IDEA
- Dimensionality reduction
- Row reduction
- Column reduction
- Rule reduction
- Sanity Check
- The End

The world is a complex place, right?



- If so, then …
  - How do dumb apes (like me) managed to gain (some) control over a (seemingly) impossibly complex world?
- So few Einsteins, so many Menziess

Are some details superfluous?



X-axis ordered by information content of each attribute

Why are some details superfluous?

- N-sphere: the size of the region of similar examples

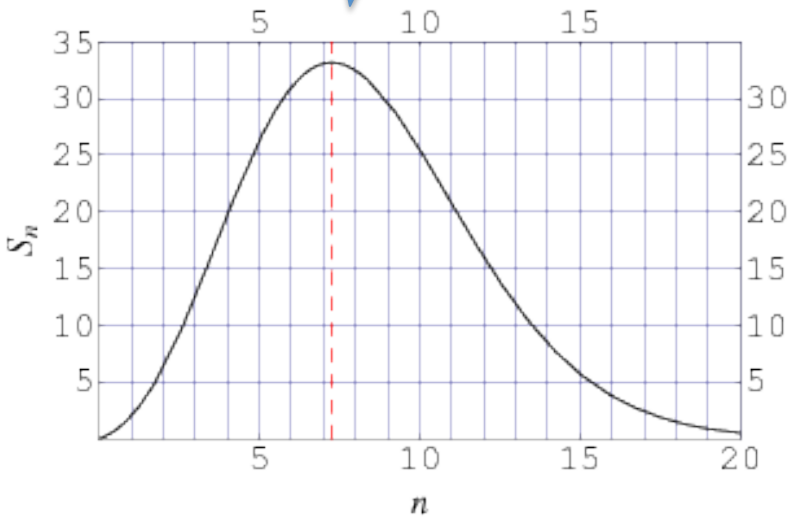
$V_2 = \pi r^2$

$V_3 = 4/3\ \pi r^3$

$V_n = V_{n-2} * 2\pi r^2 / n$

- Volume decreases after $n > 2\pi r^2$

- For the unit sphere (r = 1), size is zero after 2 dozen dimensions



- Repeated effects can't use many dimensions: else, no supporting evidence

- Lofti Zadeh:
  - As the complexity of a system increase, a threshold is reached beyond which precision and significance become mutually exclusive properties.

Data mining = data “carving”

- Data is like a block of marble,
  - waiting for a sculptor (that’s you)
  - to find the shape within
- To build a data miner, throw stuff away
  - Chip away the irrelevancies
  - To find what lies beneath.

Roadmap

- Introduction
- Throwing stuff away
- Business info needs
- IDEA
- Dimensionality reduction
- Row reduction
- Column reduction
- Rule reduction
- Sanity Check
- The End

So what are "the questions the user cares about"?

- Instead of describing data miners as
  - Classifiers
  - Association rule learners
  - Contrast set learners
  - Clusterers
  - Etc etc
- FIRST ask "what are the information needs of industrial managers?"
  - Then check how the miners fit the info needs.

Information Needs for Software Development Analytics.

- Raymond P.L. Buse, Thomas Zimmermann.
  - Proceedings of the 34th International Conference on Software Engineering (ICSE 2012 SEIP Track)
  - Zurich, Switzerland, June 2012.
- Survey of 100+ developers and managers at Msoft

The Busemann-9

	Past	Present	Future
Exploration (find)	Trends	Alerts	Forecasts
Analysis (explain)	Summarize	Overlays	Goals
Experiment (what-if)	Model	Bench marks	Simulate

Is it ***sufficient*** to automate Busemann-9 with machine learning?

- Any sane analyst should
  - augment these automatic tools
  - with their own serious domain reflection
- No survey is complete.
  - All surveys have sample biases.
  - It's a place to start
    - With some hope that these tools are relevant to something.

Is it ***necessary*** to automate Busemann-9 with machine learning?

- Before spending months on algorithms…
  - … try spending a few days just talking to people

Are there ***better ways*** than the Busemann-9 to describe info needs?

- Probably
  - But what?
- Question to you:
  - Got surveys of info needs?
- Challenge to me:
  - Map those needs to data mining tasks.

Roadmap

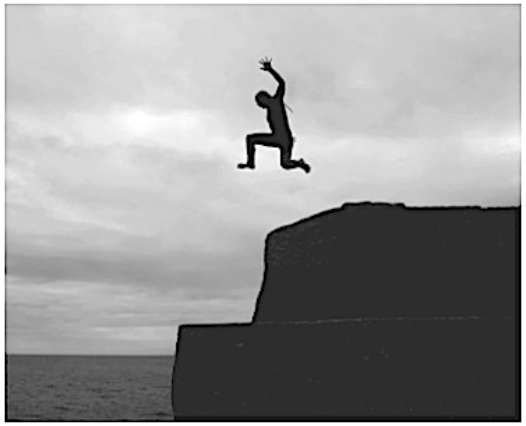
- Introduction
- Throwing stuff away
- Business info needs
- IDEA
- Dimensionality reduction
- Row reduction
- Column reduction
- Rule reduction
- Sanity Check
- The End

Design principle #1: look before your leap

- Report what is true about the data
  - Not trivia on how algorithms walk that data
- Map the landscape
  - Reason on each part of map

- E.g. IDEA
  - Unsupervised iterative dichotomization
  - Cluster, prune
  - Then generate rules

- Different to "leap before you look"
  - i.e. skew learning by class variable
  - then study the results



- E.g. C4.5, CART, Fayya-Iranni, etc
  - Supervised iterative dichotomization
- E.g. 61% * 300+effort estimation papers
  - Algorithm tinkering, without end

Design principle #2
"data mining" = "data pruning"

1. Dimensionality reduction
   - Fastmap (simple, linear time)
2. Row reduction
   - Cluster via recursive fastmap,
   - find centroids
   - IDEA v1.0
     - **I**terative **D**ichotomization on **E**very **A**ttribute
3. Column reduction
   - Ignore ranges found in many centroids
4. Rule reduction
   - Constrast sets to generate tiny rules

Preliminaries: Distance between rows

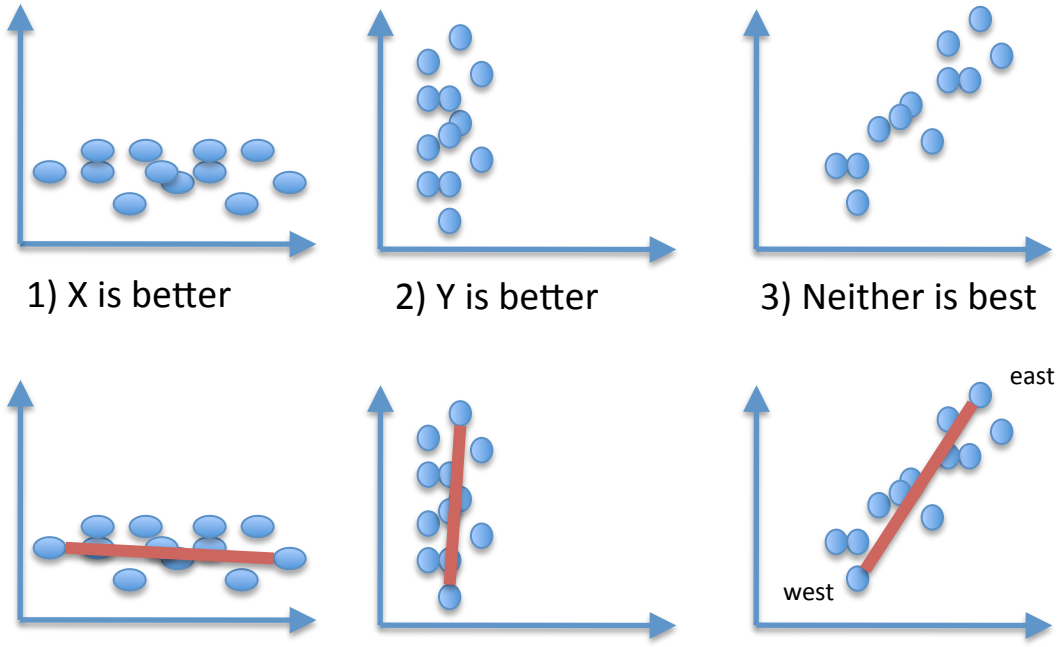
- Two rows X,Y have columns col1, col2,…
  - Some cols are numeric
  - Some cols are goals (e.g. class variables)
  - Some rows have missing values
- Aha (1991)'s unsupervised distance measure:
  - sqrt of sum of
    (c in (Cols – goals)
    ⋀ diff(Xc,Yc))
- function diff (Xc, Yc)
  - If both missing,
    - return "1" (max value)
  - Else If non-numerics:
    - If one missing , return 1
    - Else return Xc == Yc
  - Else if numeric
    - normalize each one via (one- min) / (max - min)
    - If none missing return $(Xc – Yc)^2$
    - Else if present < 0.5 return $(1 – present)^2$
    - Else return $(present)^2$

Roadmap

- Introduction
- Throwing stuff away
- Business info needs
- IDEA
- Dimensionality reduction
- Row reduction
- Column reduction
- Rule reduction
- Sanity Check
- The End

Dimensionality reduction

- Trick:
  - Dimension of greatest interest is the line of most variance



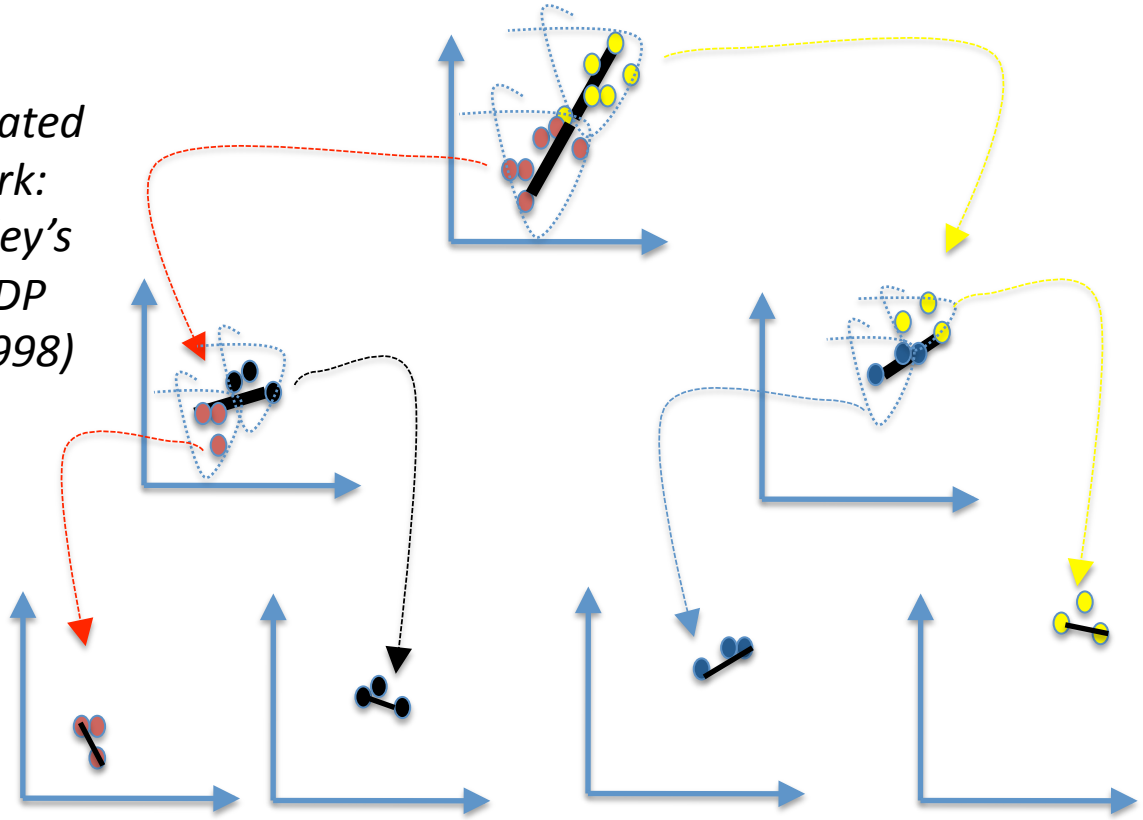
To find east west:

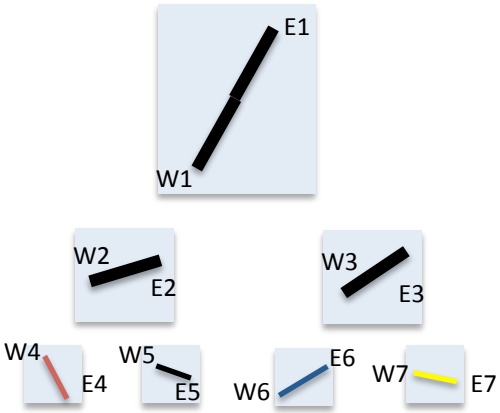
The slow way: $O(N^2)$
- compare all points
- or matrix methods eg PCA

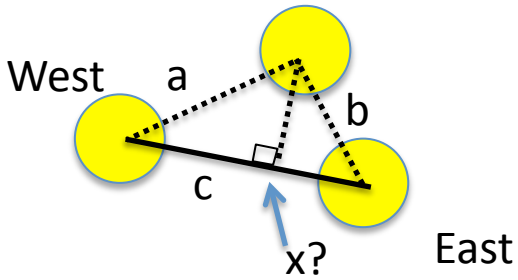
The fast way: O(2N) Faloutsos (1997)
- pick any point X ;
- East is furthest from X ;
- West is furthest from East

Recursive dimensionality reduction = clustering

Related work: Boley's PDDP (1998)



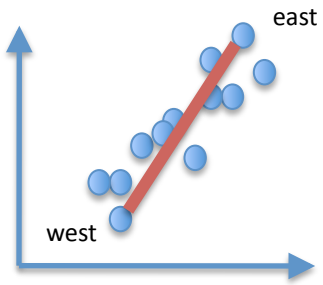




To find median point:

- a = dist(p,West);
- b= dist(p,East);
- c = dist(West,East);
- $x = (a^2 + c^2 – b^2) / (2c)$
- break at median of all x

But aren't I adding dimensions?

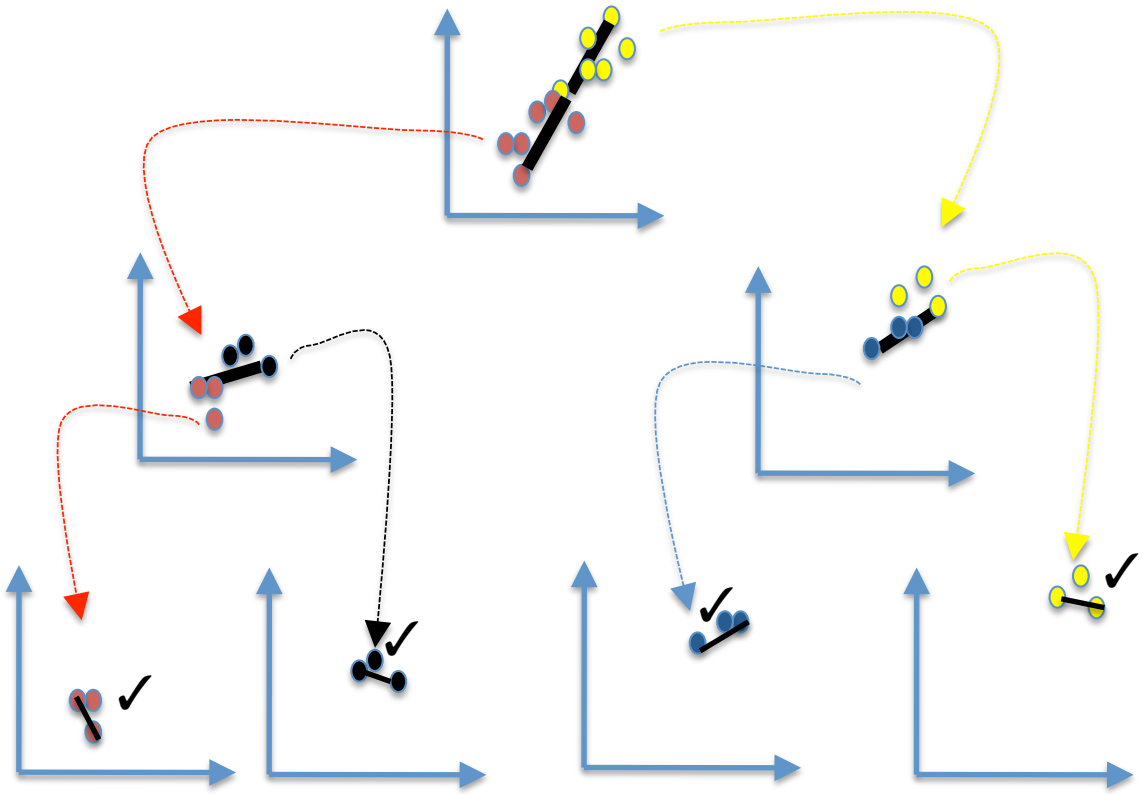


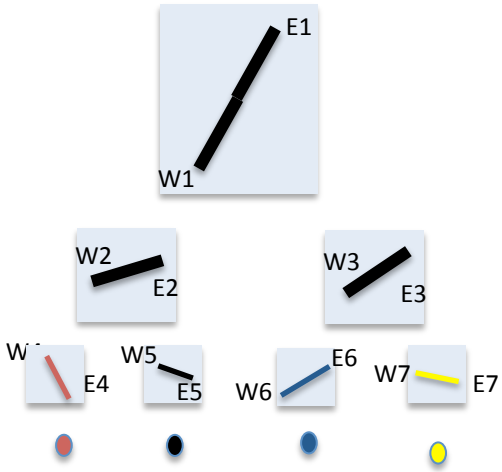
- No
- Consider a data set with 40 columns
  - Dimensionality = 40 + 1 class variable
  - At each level of the recursive Fastmap
    - Those 40 mapped to one

Roadmap

- Introduction
- Throwing stuff away
- Business info needs
- IDEA
- Dimensionality reduction
- Row reduction
- Column reduction
- Rule reduction
- Sanity Check
- The End

Row Pruning via Clustering: keep on M rows per leaf





Selection strategy

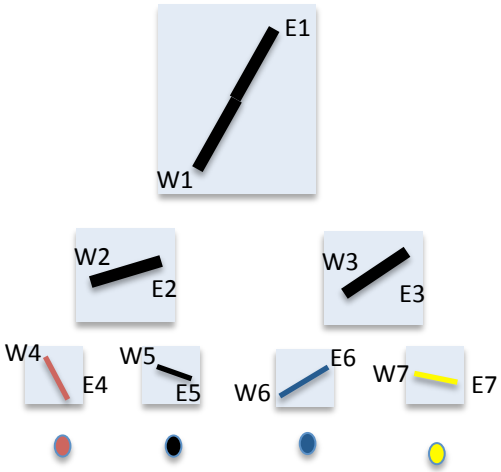
- M=1. Use mean and mode of each feature
- M = 2. Just use West, East (not recommended)
- M ≥ 1. Select points equi-distance along East-West line
- etc

East

Roadmap

- Introduction
- Throwing stuff away
- Business info needs
- IDEA
- Dimensionality reduction
- Row reduction
- Column reduction
- Rule reduction
- Sanity Check
- The End

Column pruning via Entropy



Sort columns by the probability they select for fewer clusters

Delete the more ambiguous ones

analyst capability	programmer capability	leaf cluster
2	3	c1
3	3	c1
3	4	c1
3	5	c1
3	5	c1
2	3	c2
2	4	c2
2	4	c2
3	5	c2

analyst capability	leaf cluster	entropy= sum(r* sum(-p*log(p)))
2	c1	
2	c2	0.44 * (-1/4*log(1/4) - 3/4*log(3/4)) = 0.107
2	c2	
2	c2	
3	c1	
3	c1	0.56 * (-1/5*log(1/5) - 4/5 *log(4/5)) = 0.132
3	c1	
3	c1	
3	c2	entropy= 0.107+0.132 = 0.239

Prog. capability	leaf cluster	entropy= sum(r * sum(-p*log(p)))
3	c1	0.33* (-1/3*log(1/3) -2/3*log(2/3)) = 0.091
3	c1	
3	c2	
4	c1	0.33* (-1/3*log(1/3) - 2/3*log(2/3)) = 0.091
4	c2	
4	c2	
5	c1	0.33* (-1/3*log(1/3) -2/3*log(2/3)) = 0.091
5	c1	
5	c2	entropy = 0.091*3 =0.273

Input: 93 rows * 24 cols
Output: 13 rows * 11 cols

Cluster	*effort*	*apex*	*plex*	*pmat*	*rely*	*data*	*cplx*	*time*	*stor*	*kloc*	*acap*	*pcap*
[113]	38	3	2	3	4	3	5	5	5	6.2	3	3
[112]	37	3	3	4	4	2	4	3	3	8.1	3	3
[110]	206	3	3	4	4	3	4	3	3	28.25	3	3
[120]	300	5	2	4	3	5	4	5	5	28.3	5	5
[115]	156	4	3	4	3	3	4	3	3	30.75	3	3
[121]	192	4	2	4	3	5	4	5	5	35.5	5	3
[118]	290	4	4	2	4	4	4	3	5	43.5	4	4
[116]	166	3	3	3	3	2	4	3	3	66.25	4	3
[125]	1645	4	4	4	5	4	5	6	6	70	4	4
[119]	300	4	4	2	3	3	4	3	5	77.5	4	4
[114]	304	4	3	3	4	3	3	3	3	84.5	4	4
[123]	342	5	3	3	3	2	4	3	3	125	4	5
[122]	144	4	3	3	3	2	4	3	3	131	4	4

Overlay

	Past	Present	Future
Exploration (find)			
Analysis (explain)		Overlay	
Experiment (what-if)			

- For each cluster
- Print the distributions of features of instances in that cluster

Goals & Benchmarks

	Past	Present	Future
Exploration (find)			
Analysis (explain)		Overlay	**Goals**
Experiment (what-if)		**Benchmark**	

- For each cluster
- Benchmarks: compare its distributions to industrial standards
- Goals: compare its distributions to desired outcomes

Cluster	effort	apex	plex	pmat	rely	data	cplx	time	stor	kloc	acap	pcap
[113]	38	3	2	3	4	3	5	5	5	6.2	3	3
[112]	37	3	3	4	4	2	4	3	3	8.1	3	3
[110]	206	3	3	4	4	3	4	3	3	28.25	3	3
[120]	300	5	2	4	3	5	4	5	5	28.3	5	5
[115]	156	4	3	4	3	3	4	3	3	30.75	3	3
[121]	192	4	2	4	3	5	4	5	5	35.5	5	3
[118]	290	4	4	2	4	4	4	3	5	43.5	4	4
[116]	166	3	3	3	3	2	4	3	3	66.25	4	3
[125]	1645	4	4	4	5	4	5	6	6	70	4	4
[119]	300	4	4	2	3	3	4	3	5	77.5	4	4
[114]	304	4	3	3	4	3	3	3	3	84.5	4	4
[123]	342	5	3	3	3	2	4	3	3	125	4	5
[122]	144	4	3	3	3	2	4	3	3	131	4	4

In numerous experiments:

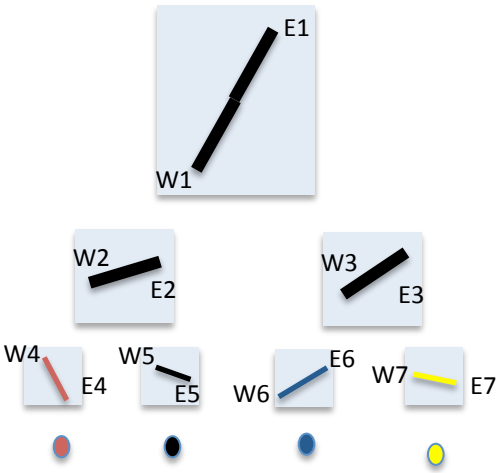
- Good predictions via
- a simple k=1 nearest neighbor in this reduced space

Forecasts

	Past	Present	Future
Exploration (find)			**Forecasts**
Analysis (explain)		Overlay	Goals
Experiment (what-if)		Bench marks	

Also, to get expectations for non-class variables, find nearest cluster and look at the column distributions of the M rows in that cluster

BTW, forecasts takes linear time



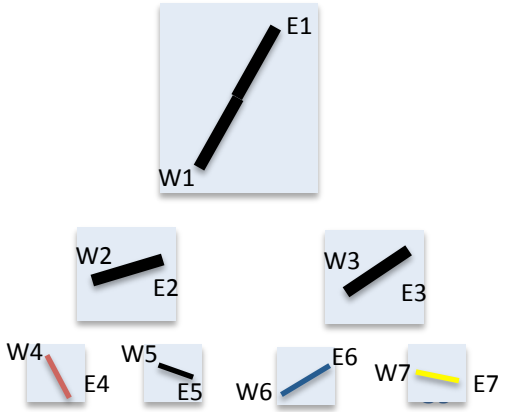
- Pre-processor : linear time
  - Generates tree of clusters
  - At each level: O(2N) search for East-West
  - At each level: find the median of the x-values on the East-West axis O(N)

- After the pre-processor
  - O(log(N)) to find your leaf cluster
  - If leaves have $M=N^{0.5}$ rows, then $O(N^{0.5})$ to find nearest nearest neighbors
  - If leaves condensed to M=1 rows, then O(1) to find nearest nearest neighbors

Alerts

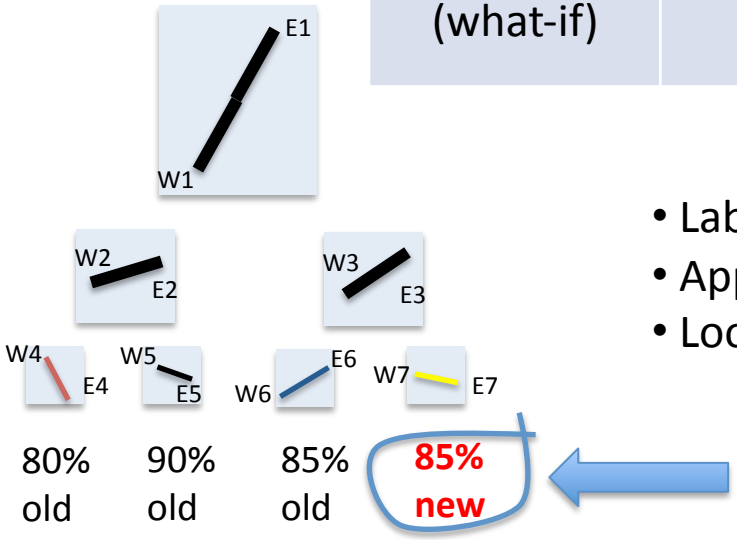
	Past	Present	Future
Exploration (find)		**Alerts (1)**	Forecasts
Analysis (explain)		Overlay	Goals
Experiment (what-if)		Benchmark	

- IDEA's fast mapping recursively finds EAST-WEST point
- Shows the extreme points of what data found too date
- Anomaly detection:
  - report new data that falls outside these points



Alerts (again)

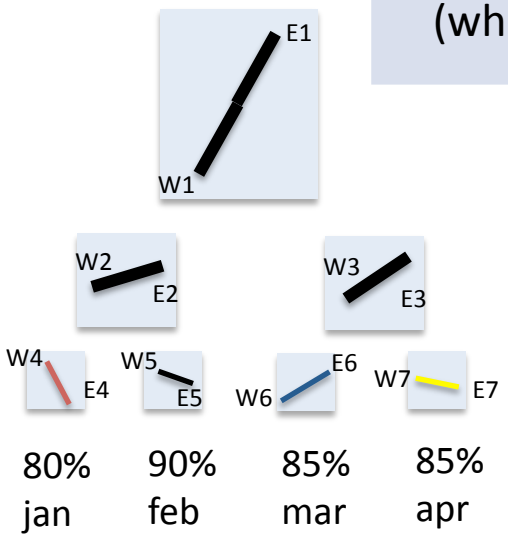
	Past	Present	Future
Exploration (find)		**Alerts (2)**	Forecasts
Analysis (explain)		Overlay	Goals
Experiment (what-if)		Benchmark	



- Label data "old, new"
- Apply IDEA, ignoring labels.
- Look for clusters where things are mostly "new"

Trends

	Past	Present	Future
Exploration (find)	**Trends**	Alerts	Forecasts
Analysis (explain)		Overlay	Goals
Experiment (what-if)		Benchmark	



- Time stamp data “jan, feb, mar, apr,…”
- Apply IDEA, ignoring labels.
- Look for clusters where one month predominates or is absent

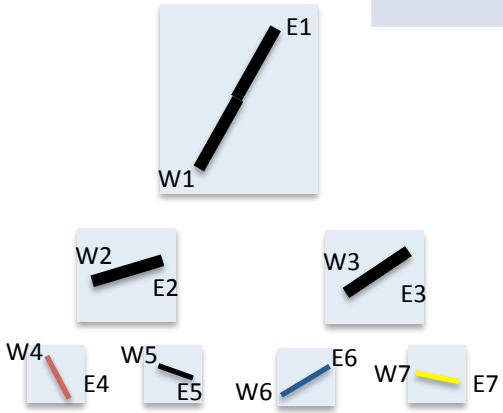
Simulate

	Past	Present	Future
Exploration (find)	Trends	Alerts	Forecasts
Analysis (explain)		Overlay	Goals
Experiment (what-if)		Benchmark	**Simulate (1)**

- Clusters map out the space of options
- So to "simulate" just report all the forecasts in different clusters

Simulate
(and this time, we mean it)

	Past	Present	Future
Exploration (find)	Trends	Alerts	Forecasts
Analysis (explain)		Overlay	Goals
Experiment (what-if)		Benchmark	**Simulate (2)**



- IDEA's fast mapping recursively finds EAST-WEST point
- To replay old data (e.g. during regression testing) sample within EAST-WEST
- When looking for new simulations, step outside the old EAST-WEST boundaries

Roadmap

- Introduction
- Throwing stuff away
- Business info needs
- IDEA
- Dimensionality reduction
- Row reduction
- Column reduction
- Rule reduction
- Sanity Check
- The End

Intra-cluster contrast sets: Very small rules, found in logLinear time

- Divide each cluster
  - best = one-third lowest effort;
  - rest = others
  - b = f(range|best) / 0.33*n
  - r = f(range|rest) / 0.66*n
- Rank via $b^2/(b+r)$: best ranges more frequent in best than rest
- Search over ranked ranges 1..n,
  - Use ranges "1..i" (i ≤ n) where no "i+1" isbetter than "I"

	effort		defect						
cluster	NasaCoc	china	lucene2.4	xalan2.6	jedit4.0	velocity1.6	synapse1.2	tomcat	xerces1.4
global	kloc=1	afp=1	rfc=2	loc=1	rfc=2	cam=7	amc=1	loc=2	cbo=1
C0									
C1	rely=n	added=4	amc=7	amc=1	ic=7	noc=1	dit=4	cbm=1	dit=1
C2	prec=h	deleted=1	ca=1	cam=2	noc=1	dam=1 or 5		dam=1	dam=1
C3		deleted=1	dam=5	cam=3	amc=6	avg_cc=4		noc=1	ca=1 or 7
C4			mfa=1	dit=2 or 4	noc=1	moa=1		rfc=5	**cbo=1**
C5			moa=1	**loc=1**				lcom3=5	
C6				**loc =1 or 2**				max_cc=1	
C7				moa=1				cbm=1	

Summary

	Past	Present	Future
Exploration (find)	Trends	Alerts	Forecasts
Analysis (explain)	**Summary**	Overlay	Goals
Experiment (what-if)		Benchmark	Simulate

- Intra-cluster contrast sets
- Divide each cluster into best and worst results
- Find what is difference about best and rest

Inter-cluster contrast sets

Cluster	effort	apex	plex	pmat	rely	data	cplx	time	stor	kloc	acap	pcap
[113]	38	3	2	3	4	3	5	5	5	6.2	3	3
[112]	37	3	3	4	4	2	4	3	3	8.1	3	3
[110]	206	3	3	4	4	3	4	3	3	28.25	3	3
[120]	300	5	2	4	3	5	4	5	5	28.3	5	5
[115]	156	4	3	4	3	3	4	3	3	30.75	3	3
[121]	192	4	2	4	3	5	4	5	5	35.5	5	3
[118]	290	4	4	2	4	4	4	3	5	43.5	4	4
[116]	166	3	3	3	3	2	4	3	3	66.25	4	3
[125]	1645	4	4	4	5	4	5	6	6	70	4	4
[119]	300	4	4	2	3	3	4	3	5	77.5	4	4
[114]	304	4	3	3	4	3	3	3	3	84.5	4	4
[123]	342	5	3	3	3	2	4	3	3	125	4	5
[122]	144	4	3	3	3	2	4	3	3	131	4	4

- What if you were building projects like cluster 121?
  - Lets look at all projects with similar klocs

Inter-cluster contrast sets

- Q: What should worry you the most?

Cluster	effort	apex	plex	pmat	rely	data	cplx	time	stor	kloc	acap	pcap
[110]	206	3	3	4	4	3	4	3	3	28.25	3	3
[120]	300	5	2	4	3	5	4	5	5	28.3	5	5
[115]	156	4	3	4	3	3	4	3	3	30.75	3	3
[121]	192	4	2	4	3	5	4	5	5	35.5	5	3
[118]	290	4	4	2	4	4	4	3	5	43.5	4	4

- A: Contrast with closest row with worst effort

Cluster	effort	apex	plex	pmat	rely	data	cplx	time	stor	kloc	acap	pcap
[120]	300	5	2	4	3	5	4	5	5	28.3	5	5
[121]	192	4	2	4	3	5	4	5	5	35.5	5	3
Contrast	+108	+1	0	0	0	0	0	0	0	-7.2	0	+2

- What is being said here?
  - With a little more apex (application experience), highly capably programmers (pcap) will get very clever and greatly increase development time.
  - Management question: does this code deserve such cleverness?

What-if

	Past	Present	Future
Exploration (find)	Trends	Alerts	Forecasts
Analysis (explain)	Summary	Overlay	Goals
Experiment (what-if)	**What-if**	Benchmark	Simulate

- What-if = inter-cluster contrast sets.

Cluster	*effort*	*apex*	*plex*	*pmat*	*rely*	*data*	*cplx*	*time*	*stor*	*kloc*	*acap*	*pcap*
[120]	300	5	2	4	3	5	4	5	5	28.3	5	5
[121]	192	4	2	4	3	5	4	5	5	35.5	5	3
Contrast	+108	+1	0	0	0	0	0	0	0	-7.2	0	+2

	Past	Present	Future
Exploration (find)	Trends	Alerts	Forecasts
Analysis (explain)	Summary	Overlay	Goals
Experiment (what-if)	What-if	Benchmark	Simulate

Roadmap

- Introduction
- Throwing stuff away
- Business info needs
- IDEA
- Dimensionality reduction
- Row reduction
- Column reduction
- Rule reduction
- Sanity Check
- The End

Is it ***insightful*** to use the Busemann-9 to discuss machine learning?

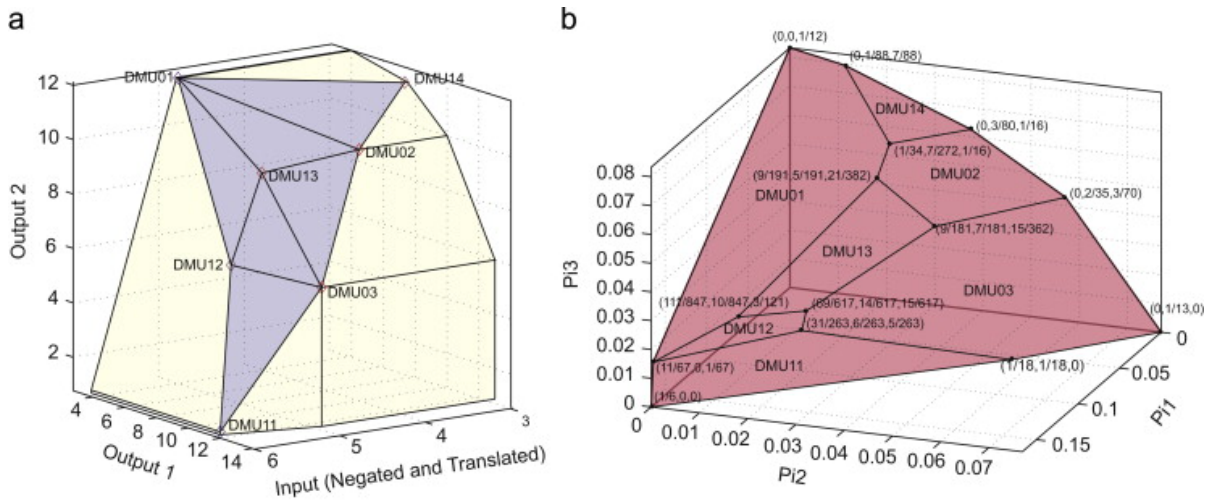
	Past	Present	Future
Exploration (find)	Trends	Alerts	Forecasts
Analysis (explain)	Summarize	Overlays	Goals
Experiment (what-if)	Model	Bench marks	Simulate

OOPS: most ML evaluated on hold-out sets (to predict future performance)

BUT: 2/3rs of these needs are about prior and current performance

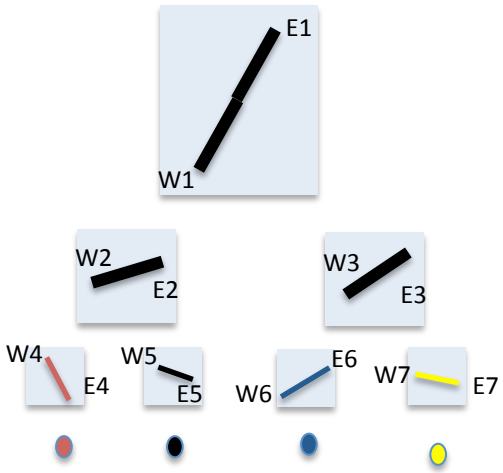
Simulation and Multi-objective optimization

- Speed up multi-objective optimization
  - Ostrouchov (2005): Fastmap's East-West are the vertices of the convex hull



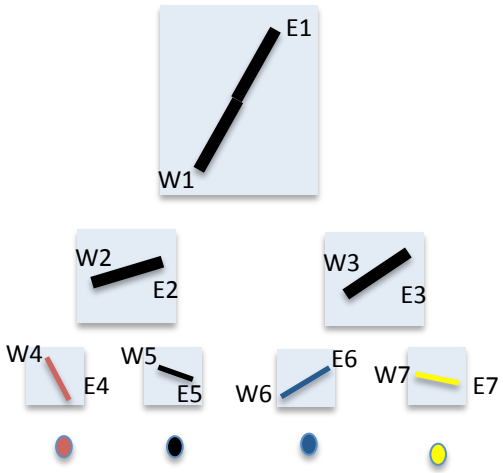
  - Menzies (2013): Prune all except the non-dominated East-West values
  - For a GA, use these as parents for next generation
- Work in progress

Working memory for active learning



- Unsupervised discretization
- Ask for class labels for medians of each each leaf cluster
- Effective conclusions after a small number of labels

Working memory for incremental data mining



- Rows at each leaf
  - a sample of things seen so far

- Read (say) 1000 rows
- Cluster, keeping (say) 20 rows at random on each leaf
- Read one more row
  - If it falls to an existing leaf, replace at random anything
  - If it falls outside the east-west pairs, raise an anomaly alert.
- If we get too many anomalies
  - Re-cluster using leaf contents & east-west pairs & anomalies

Learn local lessons; eschew trite generalities

- Standard procedure: remove outliers
  - 10 to 30% of rows
- But what it its all outliers?

	effort		defect						
cluster	NasaCoc	china	lucene2.4	xalan2.6	jedit4.0	velocity1.6	synapse1.2	tomcat	xerces1.4
global	kloc=1	afp=1	rfc=2	loc=1	rfc=2	cam=7	amc=1	loc=2	cbo=1
C0									
C1	rely=n	added=4	amc=7	amc=1	ic=7	noc=1	dit=4	cbm=1	dit=1
C2	prec=h	deleted=1	ca=1	cam=2	noc=1	dam=1 or 5		dam=1	dam=1
C3		deleted=1	dam=5	cam=3	amc=6	avg_cc=4		noc=1	ca=1 or 7
C4			mfa=1	dit=2 or 4	noc=1	moa=1		rfc=5	**cbo=1**
C5			moa=1	**loc=1**				lcom3=5	
C6				**loc =1 or 2**				max_cc=1	
C7				moa=1				cbm=1	

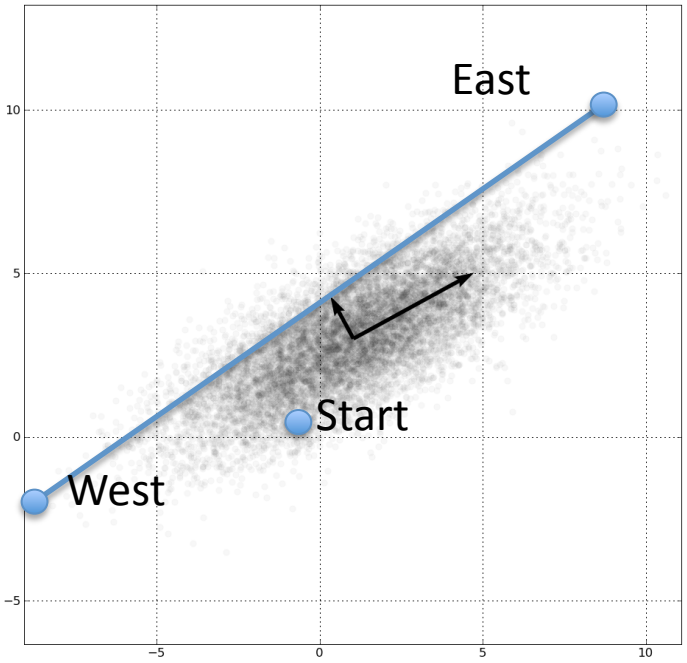
- Menzies (2011)
  - Let all data be one “cluster” and apply the same single-range learner
  - What is true globally is rarely true for specific projects
  - Same conclusion as Posnett (2011) and Hassan (2012)

Aside : for educators

- Top-down hierarchical clustering (a.k.a. divisive clustering)
- Find a way to split the data, then split the splits

Algorithm	Split	Next splitter
IDEA	Median of Fastmap's longest axis	longest axis of each split

Fastmap = an approximation to PCA's first component



- Fastmap finds an approximation to the eigenvectors of a matrix
  - FastMap, MetricMap, and Landmark MDS are all Nystrom Algorithms
    - John C. Platt, Microsoft Research, 2005,
    - http://goo.gl/DoMzg

Aside : for educators

- Top-down hierarchical clustering (a.k.a. divisive clustering)
- Find a way to split the data, then split the splits

Algorithm	Split	Next splitter
IDEA	Median of Fastmap's longest axis	longest axis of each split
PDDP (principle direction divisive partitioning)	?median of principle component	principle components of each split
KD-trees	At median	anything else
Fayyad-Irani discretiztion	To minimizes class attribute entropy	same attribute
(C4.5, CART)	To minimizes class attribute (entropy, variance)	attribute that produces the best splits

Challenge questions

- How is IDEA same and different to Nbtrees?
- How to implement M5' using IDEA?

Roadmap

- Introduction
- Throwing stuff away
- Business info needs
- IDEA
- Dimensionality reduction
- Row reduction
- Column reduction
- Rule reduction
- Sanity Check
- The End

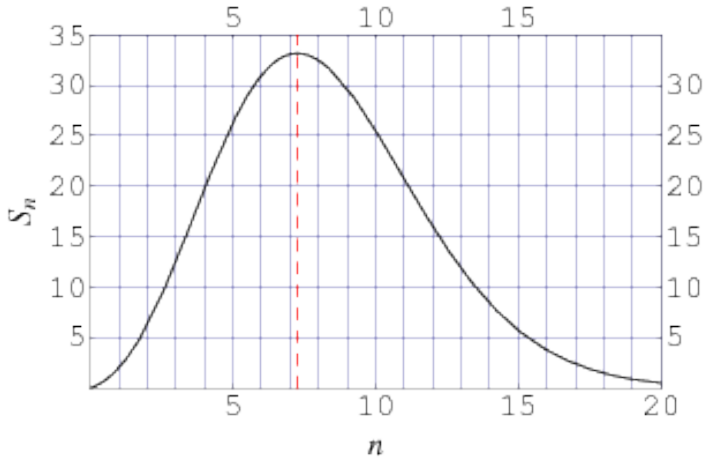
Data mining = data carving

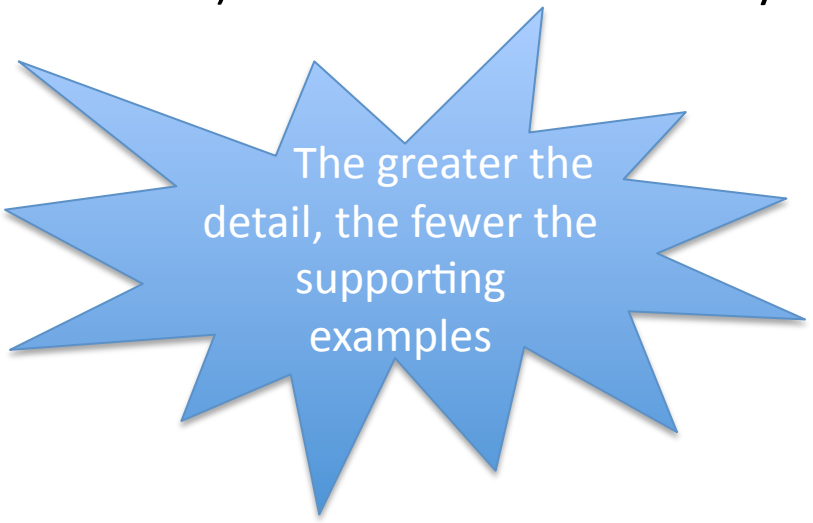
- Data is like a block of marble,
  - waiting for a sculptor (that's you)
  - to find the shape within
- Chipping away the irrelevancies
  - To find what lies beneath
- IDEA:
  - Dimensionality reduction via Fastmap
  - Row reduction via clustering
  - Column reduction via cluster reflection
  - Succinct reporting via contrast sets

Throwing away stuff is a good idea

Lofti Zadeh:

- As the complexity of a system increase, a threshold is reached beyond which precision and significance (or relevance) become almost mutually exclusive properties.





- Data mining is data carving. "Throwing stuff away"
  - Is a model of human cognition
  - Is an engineering principle
  - Cures the curse of complexity

IDEA v1.0
(my new data carver)

- Mostly, it just throws away stuff
- A framework where we can meet many mining methods.
- A mapping of data miners to business information needs.
- LogLinear time learning: suitable for rapid experimentation
- A fruitful error?
  - So many paths not taken, begging to be visited

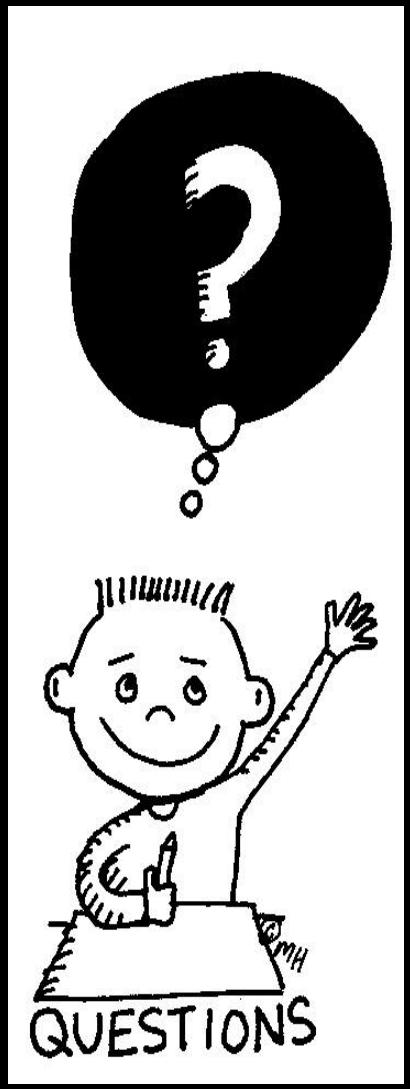
Vilfredo Pareto:

- "Give me the fruitful error any time, full of seeds, bursting with its own corrections."
- "You can keep your sterile truth for yourself."

Other comments

- Start studying the data
  - The field is called “data mining”
  - Yet most folks do “algorithm mining”
- Look before you leap
  - Map topology of data
  - Restrict learning to regions in the topology

- More to business than classification, regression, etc.
  - These are “how”, not “what”
  - Insightful to start with biz needs
- Stop studying algorithms
  - Start studying connection between algorithms
- Simplicity matters
  - K.I.S.S.
  - What can you do in linear time?
  - Then experiment with elaborations


QUESTIONS