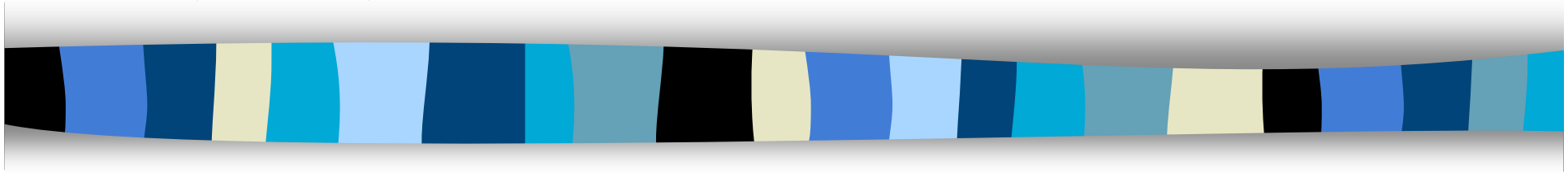
The Data Maturity Model (v0.1)



Feb-16-06

Tim@Menzies.Us
WVU, USA

(For more on this paper, see http://menzies.us/pdf/06learnPredict.pdf)

Executive Summary

- New baselines:
  - In experimental methodology
    - **Data maturity model**, “quartile charts”, “M*N-ways”
  - In learned detectors
    - Mean probability of detection: over 2/3 rds
    - Mean probability of failure: under 1/4 th
- Debating Halstead vs McCabe is an irrelevancy
  - The *learning method* is more important than *attribute subset* used during learning
- Errors follow a “log-normal distribution”:
  - So the next version of PREDICT needs another learner.
    - “Naïve Bayes with single Gaussian kernel estimator” with a “logNums” pre-filter: replace all nums with log(nums)

Is this data useful?

- How has it been used (in the past) to make generalizable conclusions?
- How can it be used (in the future) to make new conclusions?

May you never get what you wish for

http://unbox.org/promise/2006

Title[3] CM1/Software defect prediction
Title[4] JM1/Software defect prediction
Title[5] KC1/Software defect prediction
Title[6] KC2/Software defect prediction
Title[7] PC1/Software defect prediction
Title[8] Cocomo81/Software cost estimation
Title[9] Cocomo NASA/Software cost estimation
Title[10] Reuse/Predicting successful reuse
Title[11] DATATRIEVE Transition/Software defect prediction
Title[12] Class-level data for KC1 (Defect Count)/Software
Title[14] Class-level data for KC1 (Defective or Not)/Software
Title[15] Class-level data for KC1 (Top 5% Defect Count Ranking
Title: [16] Nickle Repository Transaction Data
Title: [17] XFree86 Repository Transaction Data
Title: [18] X.org Repository Transaction Data
Title[19] MODIS/Requirements Tracing
Title[21] CM1/Requirements Tracing
Title[23] Desharnais Software Cost Estimation

So, we are collecting the data

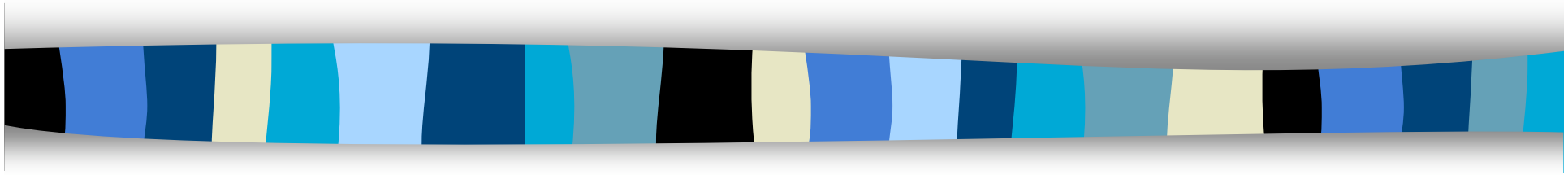
New problem:

- Q: How we need to enforce reasonable standards of analysis?
- A: Lead by example

Honorable mention

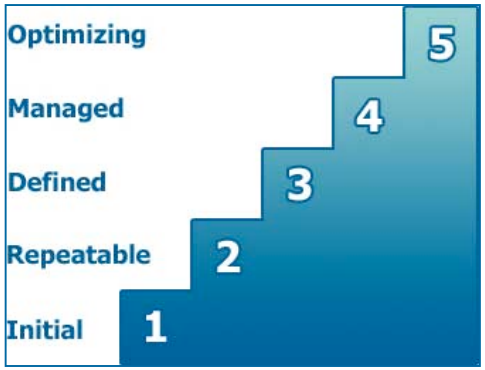
- Wilson & Merritt's analysis of language bias in "ROCKY"
- Good stuff
- Should extend that study:
  - More data, broader discretization methods, separate train/test sets,
  - See below

Methodology notes



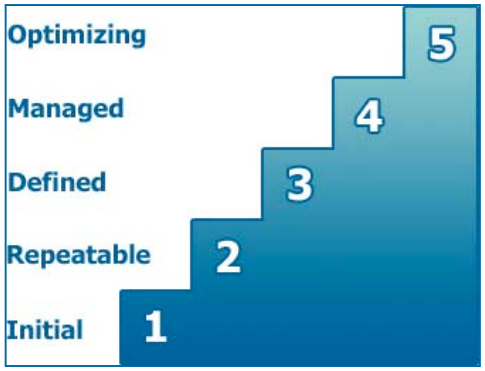
Introducing the
Data Maturity Model

The Data Maturity Model: a better standard for data processing



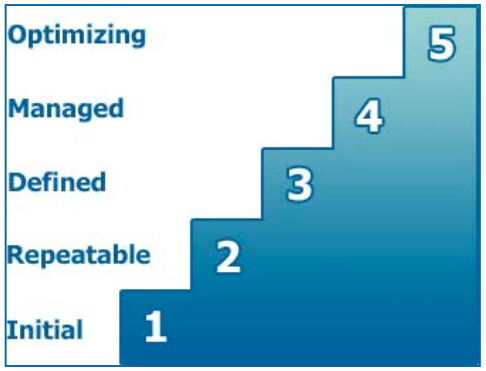
- The lower the levels, the less effort in creating and the data;
  - i.e. 1 is lazier than 5.
- The higher the levels, the more the data has been used and is useful;
  - i.e. 5 is better than 1.
- Each level has steps.
  - Reaching Level 1 means achieving all its steps.
  - To reach the higher levels *l>1*:
    - the lower level *l-1* must be reached
    - but only ***ENOUGH%*** the steps for this level must be achieved.
- **How much is enough?**
  - For a standard to be practical, it can't be too dogmatic. Hence, *ENOUGH=66%*
  - Everyone can play

The Data Maturity Model: level 1 = initial



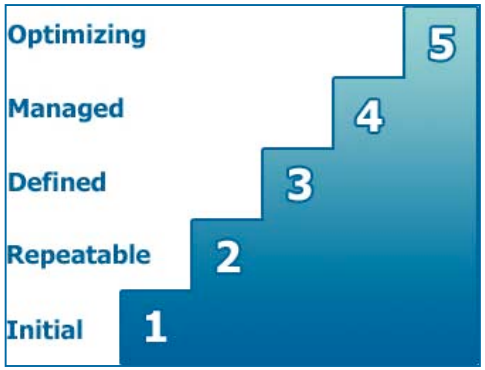
- ✓ Data is in some defined data format (csv, xml, arff, ...).
- ✓ Data has been run through any automatic learner.
- ✓ The learned theory has been automatically applied to some data to return some conclusion without human intervention.
  - Note that manual browsing of some on-screen visualization does *not* constitute automatic application.

The Data Maturity Model: level 2 = repeatable



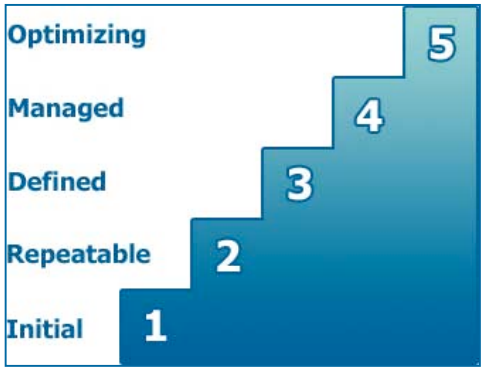
- ✓ A theory learned from some data D1 has been run on some other data D2 and D1 is not D2 ;
  - e.g. via a N-way cross-validation study (defined in http://menzies.us/pdf/04ivv.pdf, page 31)
- ✓ Data is in the public domain;
  - e.g. on a web site with free registration or, better yet, no registration.
  - So someone else can repeat/ refute/ improve the results
- ✓ Data has been run through learners that are public domain.
- ✓ Someone else has processed this data rather than the original users.

The Data Maturity Model: level 3 = defined



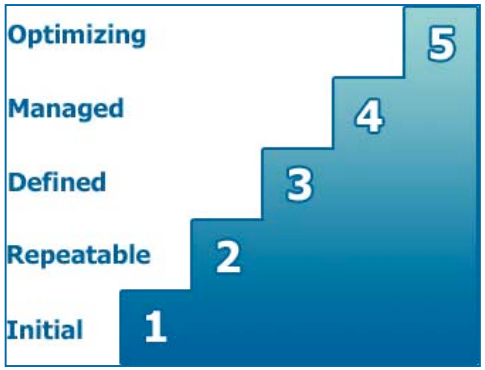
- ✓ A goal for the learning is recorded;
  - e.g. a business situation has been specified in which solutions of type X are useful but solutions of type "Y" are not.
  - Here, goal= find predictors with low pf and high pd
- ✓ The meaning of most attributes are defined;
  - e.g. comments explaining as much as is known about how those values were collected, what they mean, etc.
- ✓ The meaning of each instance is defined;
  - e.g. how is one instance different to another? how were each instances collected? to what extent do we trust the data collection process?
- ✓ Statistics are available on the distribution of each attribute.
  - Statistics include information on how many missing values exist (and some explanation is offered for the missing values).
- ✓ Attribute subsets are identified that have differing effects on the goals;
  - e.g. if the goal is cheap defect detection, then the attributes could be grouped into the cost of their data collection.
- ✓ Instance subsets are identified which domain knowledge observes tells us is very different to the other instances;
  - e.g. we use data from 8 sources

The Data Maturity Model: level 4 = managed



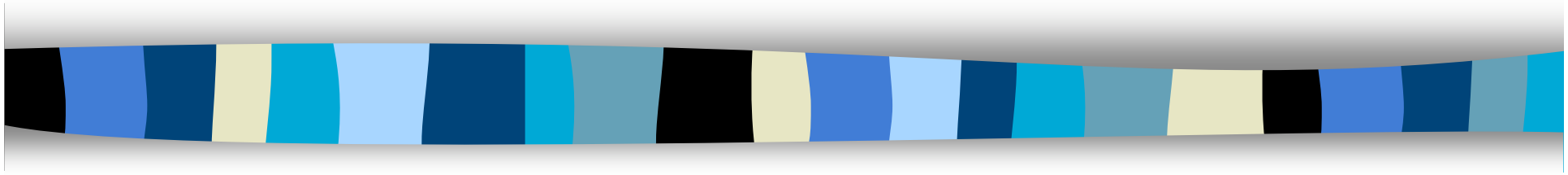
- ✓ Simple attribute distributions studied have been performed:
  - e.g. outliers determined by a manual browsing graphs of the distributions of individual attributes.
- ✓ Data is run though multiple pre-processors;
  - e.g. RemoveOutliers, BinLogging , NBins, LogTransforms, etc.
- ✓ Data is run though multiple learners.
- ✓ Data with different attribute/instance subsets have been run through different learners after different pre-processing.
- ✓ Prior results with this data set are identified; (see ICSE 2005)
- ✓ Results compared to prior results;
  - e.g. using some widely used measure like pred(25) discussing similarities,differences, and advances over previous work.
- ✓ The results from learning from different attributes/instances/pre-processing/learners has been compared in some way (e.g. via t-tests or delta diagrams). Here, we use “quartile charts”
- ✓ Some trade-off study has been performed; e.g. roc curves where the learning goals are used to comment where in the roc curves this learner should fall.
- ✓ Some straw man study has been performed; e.g. data compared to much simpler learners10.Some reduction studies has been performed; e.g.IncrementalCrossValidation or Feature subset selection.

The Data Maturity Model: level 5 = optimized



- Issues with the current high-water mark with this learner are identified.
- Any differences in the learner performance has been analyzed and explained;
  - e.g. via studies on synthetic data sets and/or lesion studies such as where does the current learner stop working when the variance in the continuous variables is increased.
- The limits of the current approach have been stated.
- A future direction for processing this data is defined.
- Going beyond the list of problems, a tentative solution has been proposed.

Experiments



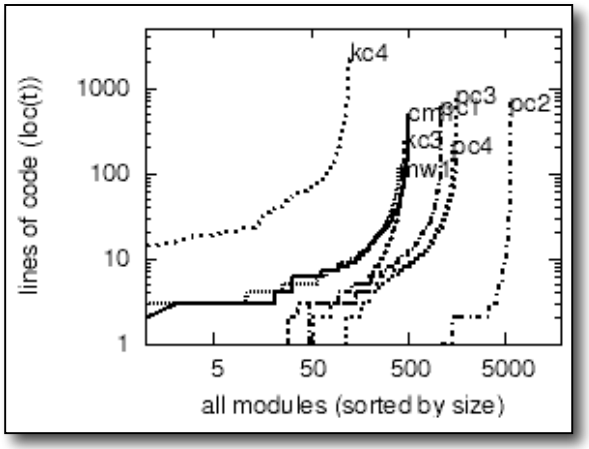
Methods

& results

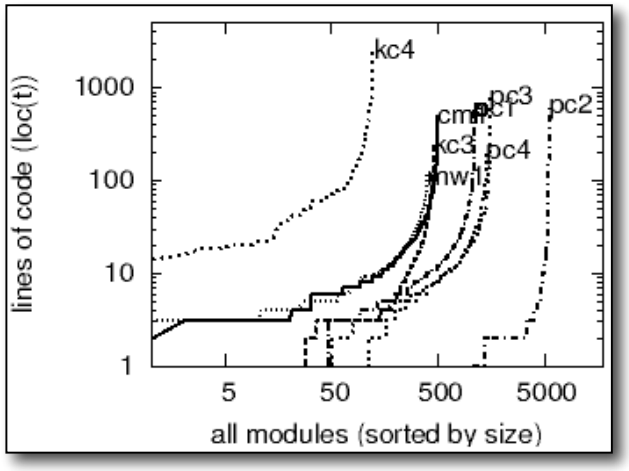
Method

- 10 MDP data sets
  - 8 with 43 attributes: used in this study
  - 2 with 21 attributes: not used

system	language	sub-system data set	# instances	% defective
spacecraft instrument	C	cm1-05	506	9
storage management for ground data	C++	kc3	459	9
		kc4	126	49
Db	C	mw1	404	7
Flight software for earth orbiting satellite	C	pc1-05	1,108	6
		pc2	5,590	0.4
		pc3	1,564	10
		pc4	1,458	12



LogNum: handling exponential distributions

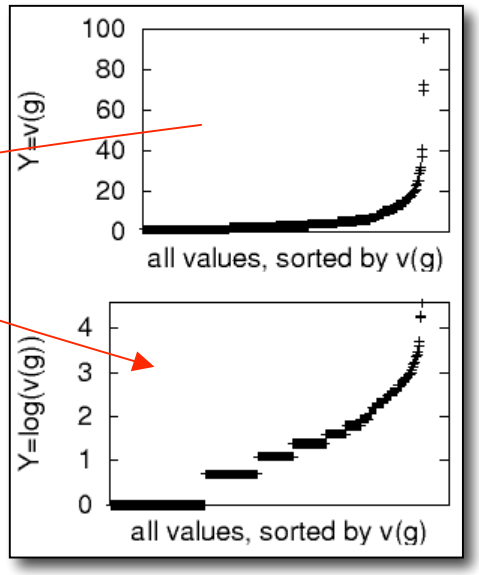


Note all the exponential distributions

```
function logNums(x) {
   min = 0.000001;
   if x<=min
         return log(min)
   else return log(x) }
```

change this to that

Avoids errors with log(0)



most	m = Mccabe		$v(g)$	cyclomatic_complexity
			$iv(G)$	design_complexity
			$ev(G)$	essential_complexity
	locs	loc		loc_total (one line = one count
		loc(other)		loc_blank
				loc_code_and_comment
				loc_comments
				loc_executable
				number_of_lines (opening to closing brackets)
	Halstead	h	N_1	num_operators
			N_2	num_operands
			μ_1	num_unique_operators
			μ_2	num_unique_operands
		H	N	length: $N = N_1 + N_2$
			V	volume: $V = N * log_2\mu$
			L	level: $L = V^*/V$ where $V^* = (2 + \mu_2{}^*)log_2(2 + \mu_2{}^*)$
			D	difficulty: $D = 1/L$
			I	content: $I = \hat{L} * V$ where $\hat{L} = \frac{2}{\mu_1} * \frac{\mu_2}{N_2}$
			E	effort: $E = V/\hat{L}$
			B	error_est
			T	prog_time: $T = E/18$ seconds
	misc = Miscellaneous			branch_count
				call_pairs
				condition_count
				decision_count
				decision_density
				design_density
				edge_count
				global_data_complexity
				global_data_density
				maintenance_severity
				modified_condition_count
				multiple_condition_count
				node_count
				normalized_cyclomatic_complexity
				parameter_count
				pathological_complexity
				percent_comments

Attribute subsets: which subset matters?

- *most*: 38 attributes- all of the following;
- *m=McCabe*: 3 attributes- basic McCabe measures;
- *loc*: 1 attribute- simplest line counts;
- *loc(other)*: 5 attributes - other line counts;
- *locs=loc+loc(other)*: 6 attributes - all the line counts;
- *h*: 4 attributes- the 4 core Halstead measures;
- *H*: 8 attributes- calculated from h;
- *hH* = *h* + *H*: 12 attributes- all Halstead values;
- *misc*: 17 attributes- other attributes found in the data.

subsets explored here

(Foreshadowing: attribute subsets will be shown to be **less** important than the learning method)

Three learning method: which one is best?

1. OneR: simple single attribute tests
   E.g. if v(g)<=10 then *safe* else *defects*

2. J48: complex combinations of many attribute tests in decision trees:
   E.g. if v(g)<=10 then
   if iv(g) < 4 then safe
   else defect
   else v(g)>10 then defect

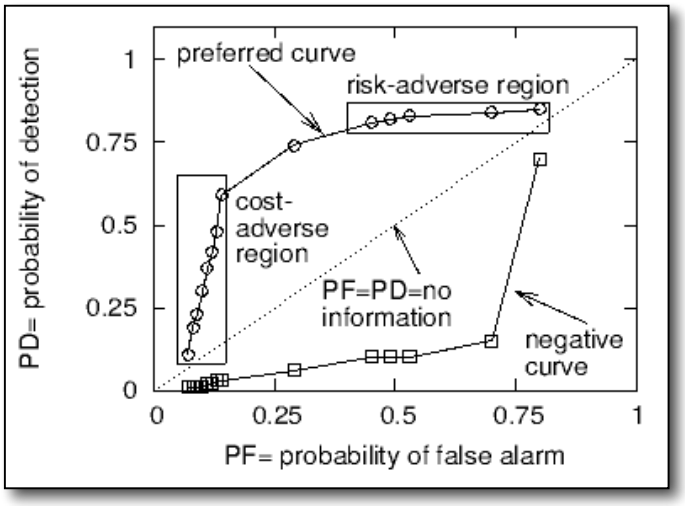
3. Naïve Bayes: conclusions based on multiplying attribute range frequencies

Compatible with the MW study

Best if theories need conjunctions and disjunctions

Best if theories need continuous distributions

Success criteria: maximize “pd”, notPf”, “bal”



		module found in defect logs?	
		no	yes
signal detected?	no (i.e. $v(g) < 10$)	A = 395	B = 67
	yes (i.e. $v(g) \geq 10$)	C = 19	D = 39

$pd =$	$Prop.detected =$	37%
$pf =$	$Prob.falseAlarm =$	5%
$notPf =$	$1 - pf =$	95%
$bal =$	$Balance =$	45%
$Acc =$	$accuracy =$	83%

$$\text{probability detection} = pd = recall = \frac{D}{B+D}$$

$$\text{notPf} = 1 - \frac{C}{A+C}$$

$$balance = bal = 1 - \frac{\sqrt{(\sqrt{0} - pf)^2 + (\sqrt{1} - pd)^2}}{\sqrt{2}}$$

The Wilson&Merritt study: PD vs “effort” (PF does not matter)

Experimental rig:

```
M         = 10
N         = 10
DATAS     = (cm1 kc3 kc4 mw1 pc1 pc2 pc3 pc4)
SUBSETS   = (loc locs m h hH locs+m locs+h locs+hH most)
FILTERS   = (none logNums)
LEARNERS  = (oneR j48 nb)

for data in DATAS
   for filter in FILTERS
       data' = filter(data)
      for attributes in SUBSETS
          some = just the attributes of data'
         repeat M times
             randomized order from "some"
             generate N bins from "some"
            for i in 1 to N
               tests  = bin[i]
               trainingData = some - tests
              for learner in LEARNERS
                 METHOD = (filter attributes learner)
                 theory = learner(trainingData)
                 RESULT[METHOD] = use theory on tests
```

Randomization protects against “order” effects

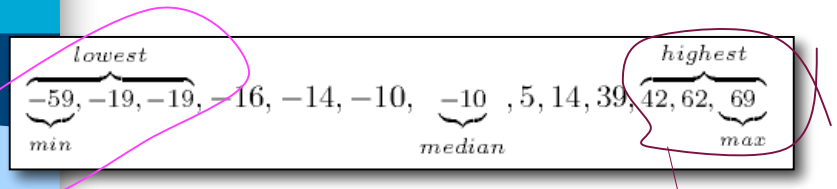
N-way cross-validation:
- Train on (N - 1)/N of the data
- Test of remaining 1/N th of the data

43,200 calls to data miner

This should be in the tools lab so folks can test their own filters/learners, etc

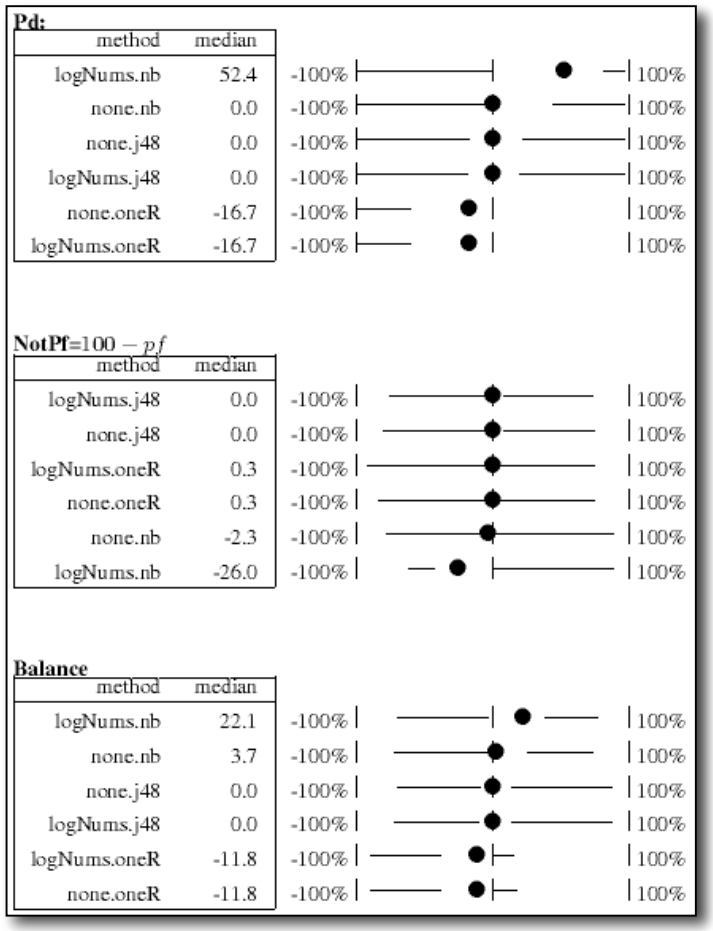
Quartile plots

- 43,200 experiments
- For all pairs of methods <M1,M2>
  - Report deltas in performance: delta= pd(M1) - pd(M2)
  - Delta= -100% if M1 always got 0% and M2 got 100%
  - Delta=100% if M1 always got 100% and M2 got 0%
  - Sort deltas, show medians and upper/lower quartiles



lower quartile:
Here, this method is doing WORSE than others

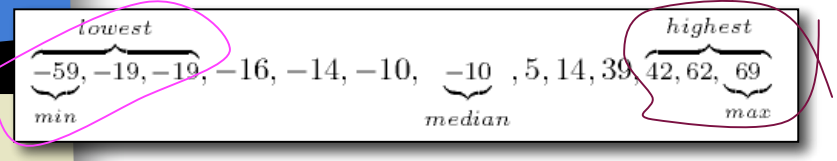
upper quartile:
here, this method is doing BETTER than others



lower quartile:doing WORSE than others

upper quartile: doing BETTER than others

Which learner matters?



Pd:

method	median
logNums.nb	52.4
none.nb	0.0
none.j48	0.0
logNums.j48	0.0
none.oneR	-16.7
logNums.oneR	-16.7

NotPf=100 − *pf*

method	median
logNums.j48	0.0
none.j48	0.0
logNums.oneR	0.3
none.oneR	0.3
none.nb	-2.3
logNums.nb	-26.0

Balance

method	median
logNums.nb	22.1
none.nb	3.7
none.j48	0.0
logNums.j48	0.0
logNums.oneR	-11.8
none.oneR	-11.8

- logNums(NaiveBayes)
  - Performs very well indeed
- Single attribute rules (e.g. vg<10) performs poorly

- Most learners strive to avoid false alarms, don't take risks.
- logNums(NaiveBayes) achieves high PDs by taking risks. So has worse PFs

- On balance, logNums(NaiveBayes) has a much higher median performance
- Single attribute rules perform badly (on all of pd,notPf, and balance)

Results using NaiveBayes + logNums

As far as we know: the results in this report are the best ever seen

An ICSE 2005 results of 88% accuracy using churn-based metrics:

- but only one data set
- results were from "self-test", not cross-validation
- pd/pf not reported)

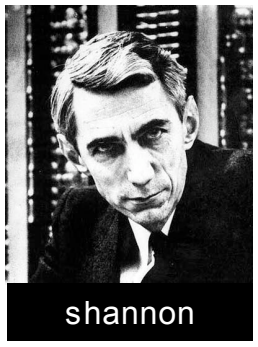
data	N	% pd	% pf	% acc	selected attributes (see Figure 11)	selection method
cm1	100	71	27	73	5, 35, 36	iterative FSS
kc3	100	69	28	72	16, 24, 26	iterative FSS
kc4	100	79	32	73	3, 13, 31	iterative FSS
mw1	100	52	15	82	23, 31, 35	iterative FSS
pc1	100	48	17	81	3, 35, 37	exhaustive FSS
pc2	100	72	14	86	5, 39	iterative FSS
pc3	100	80	35	67	1, 20, 37	iterative FSS
pc4	100	98	29	74	1, 4, 39	iterative FSS
all	800	71	25	76		

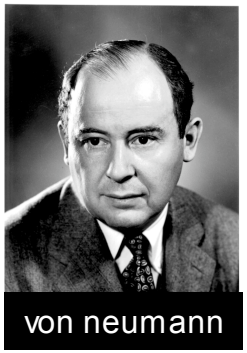
Huh? Results from using 3 (of 38) attributes?

And those attributes are different all the time?

ID	frequency in Figure 10	what	type
1	2	loc_blanks	locs
3	2	call_pairs	misc
4	1	loc_code_and_command	locs
5	2	loc_comments	locs
13	1	edge_count	misc
16	1	loc_executable	locs
20	1	I	H (derived Halstead)
23	1	B	H (derived Halstead)
24	1	L	H (derived Halstead)
26	1	T	H (derived Halstead)
31	2	node_count	misc
35	3	μ_2	h (raw Halstead)
36	1	μ_1	h (raw Halstead)
37	2	number_of_lines	locs
39	2	percent_comments	misc

Entropy: less is more


shannon

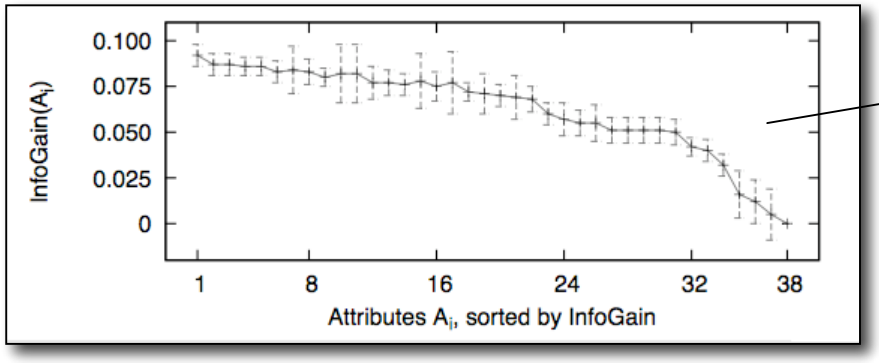

von neumann

- Sample has classes c(1), c(2)..
- Occurring at frequency n(1), n(2)
- Mow much does attribute Ai shrinks the encoding?

$$
\begin{aligned}
N &= \textstyle\sum_{c \in C} n(c) \\
p(c) &= n(c)/N \\
H(C) &= -\textstyle\sum_{c \in C} p(c) log_2 p(c)
\end{aligned}
$$

$$H(C|A) = -\sum_{a \in A} p(a) \sum_{c \in C} p(c|a) log_2 (p(c|a)$$

$$InfoGain(A_i) = H(C) - H(C|A_i)$$



Feature subset selection:
- Sort all attributes by InfoGain(Ai)
- For n = 1,2,3,4,5,….
  - Build theories using the top N ranked attributes

InfoGain Feature subset selection finds two or three attributes that work as well as 38

Standard deviation over ten 90% random sub-samples

Many candidates for “king”

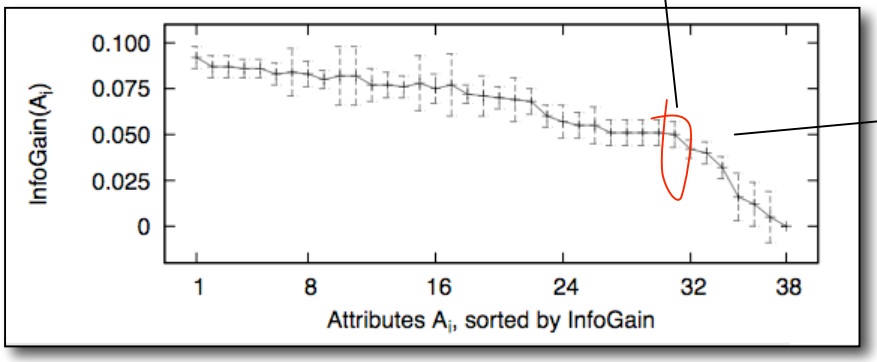
Explains why prior results so variable

Explains success of NaiveBayes listening to just 2 or 3 variables

data	N	% pd	% pf	% acc	selected attributes (seeFigure 11)	selection method
cm1	100	71	27	73	5, 35, 36	iterative FSS
kc3	100	69	28	72	16, 24, 26	iterative FSS
kc4	100	79	32	73	3, 13, 31	iterative FSS
mw1	100	52	15	82	23, 31, 35	iterative FSS
pc1	100	48	17	81	3, 35, 37	exhaustive FSS
pc2	100	72	14	86	5, 39	iterative FSS
pc3	100	80	35	67	1, 20, 37	iterative FSS
pc4	100	98	29	74	1, 4, 39	iterative FSS
all	800	71	25	76		

Balance:

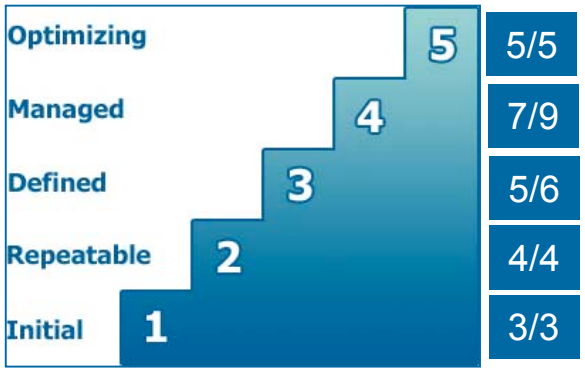
#attributes	median	
3	10.0	-100% … 100%
38	8.0	-100% … 100%
2	0.0	-100% … 100%
1	-30.7	-100% … 100%



Feature subset selection:
- Sort all attributes by InfoGain(Ai)
- For n = 1,2,3,4,5,….
  - Build theories using the top N ranked attributes

A DMM audit

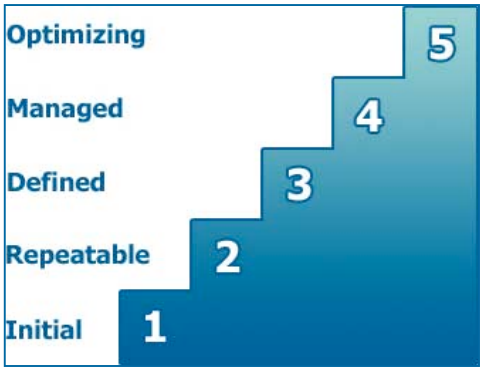
The DMM score of this work



5/5
7/9
5/6
4/4
3/3

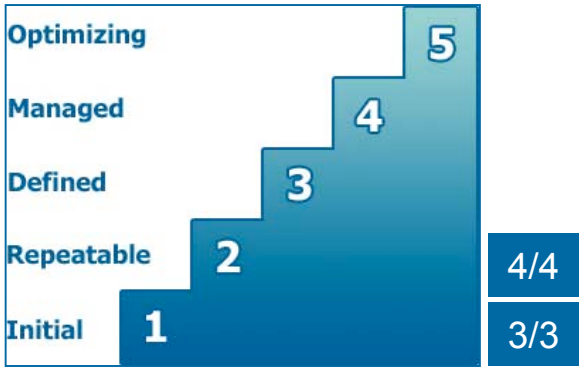
These scores justified on the following pages

The Data Maturity Model: level 1 = initial



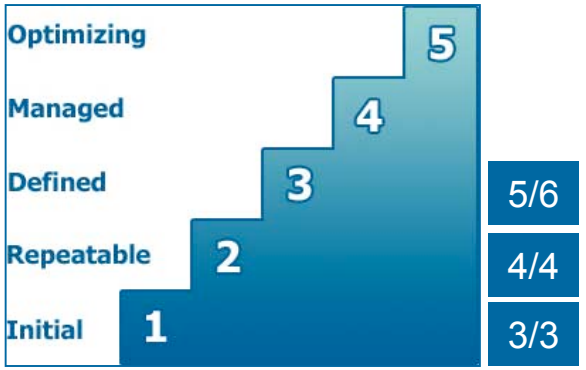
- ✓ Data is in some defined data format (csv, xml, arff, ...).
  - Here, ARFF format (an international standard)
- ✓ Data has been run through any automatic learner.
  - Actually, three learners * 2 pre-filters
- ✓ The learned theory has been automatically applied to some data to return some conclusion without human intervention.
  - Yes. See slide 20

The Data Maturity Model: level 2 = repeatable



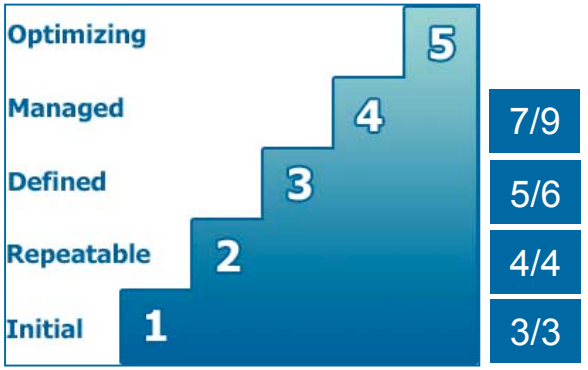
- ✓ A theory learned from some data D1 has been run on some other data D2 and D1 is not D2 ;
  - Yes: see the N-way cross-validation study defined, slide 19
- ✓ Data is in the public domain;
  - Yes: in csv format in multiple tables (see mdp.ivv.nasa.gov)
  - Yes: in arff format (see http://unbox.org/data/arff/mdp43/)
- ✓ Data has been run through learners that are public domain.
  - Three learners from the WEKA toolkit (downloadable from http://sourceforge.net/projects/weka/)
    - J48
    - OneR
    - NaiveBayes
- ✓ Someone else has processed this data rather than the original users.
  - Many authors are working on the MDP data.

The Data Maturity Model: level 3 = defined



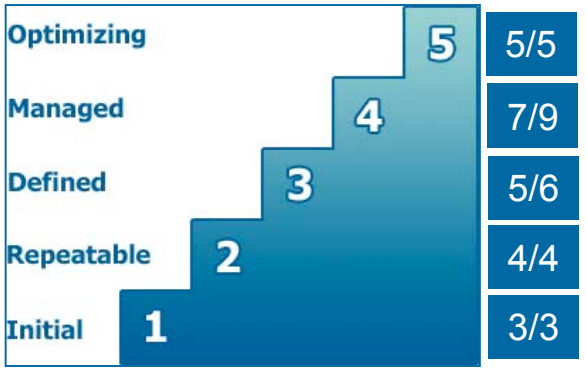
- ✓ A goal for the learning is recorded;
  - Here, goal= find predictors with low pf and high pd
- ✓ The meaning of most attributes are defined;
  - See on-line notes at http://mdp.ivv.nasa.gov/repository.html
- ✘ The meaning of each instance is defined;
  - Sort of: some access to the source of each instance but not much public domain information available here.
- ✓ Statistics are available on the distribution of each attribute.
  - Yes: not shown here.
- ✓ Attribute subsets are identified that have differing effects on the goals;
  - Sort of: we found that most attribute subsets have little effect on the goals..
- ✓ Instance subsets are identified which domain knowledge observes tells us is very different to the other instances;
  - e.g. we use data from 8 sources

The Data Maturity Model: level 4 = managed



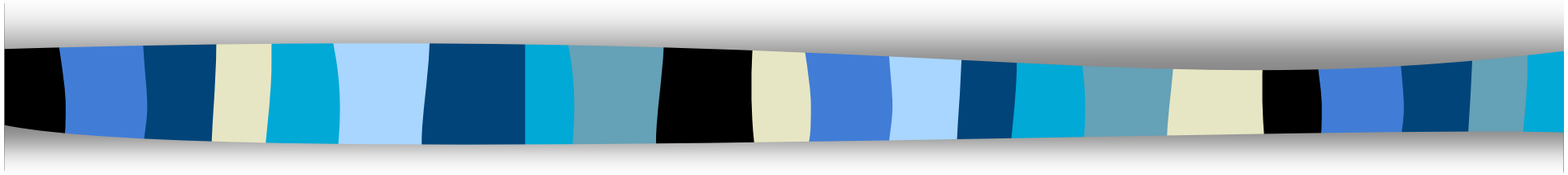
- ✘ Simple attribute distributions studied have been performed:
  - no
- ✓ Data is run though multiple pre-processors;
  - Yes: logNums, none
- ✓ Data is run though multiple learners.
  - Yes
- ✓ Data with different attribute/instance subsets have been run through different learners after different pre-processing.
  - Yes
- ✓ Prior results with this data set are identified;
  - Yes, see ICSE 2005
- ✓ Results compared to prior results;
  - See notes on page 18
  - Also, in the support paper for this presentation, there are numerous references to papers supporting / arguing against those hypotheses.
- ✓ The results from learning from different attributes/instances/pre-processing/learners has been compared in some way (e.g. via t-tests or delta diagrams).
  - Here, we use “quartile charts”
  - See also, support paper for other statistical tests.
- ✘ Some trade-off study has been performed; e.g. roc curves where the learning goals are used to comment where in the roc curves this learner should fall.
  - Nope. Only Allah is perfect.
- ✓ Some straw man study has been performed;
  - Compared to OneR

The Data Maturity Model: level 5 = optimized



- ✓ Issues with the current high-water mark with this learner are identified.
  - Our false alarm rates are still a worry. Need to reduce them.
  - Explanation is a problem: see below.
- ✓ Any differences in the learner performance has been analyzed and explained;
  - If defects really follow a log-normal distribution, then NaiveBayes works best since it is the only method that can directly exploit those distributions
- ✓ The limits of the current approach have been stated.
  - need more learners
  - Need more discretiization methods
- ✓ A future direction for processing this data is defined.
  - What about other business cases? E.g. Use this rig for the Merritt/Wilson study
- ✓ Going beyond the list of problems, a tentative solution has been proposed.
  - Two ideas: feature subset selection using WRAPPER on NaiveBayes
    - Might reduce false alarm rate AND generate attribtue sets small enough to manuall visualize

And so… ?



Summary

- Good idea: learn defect detectors from static code measures.
  - Prior pessimism unfounded
- New baselines:
  - In experimental methodology
    - Data maturity model, “quartile charts”, “M*N-ways”
  - In learned detectors
    - Mean pd: = 2/3 rds
    - Mean pf: = 1/4 th
- Debating Halstead vs McCabe is an irrelevancy
  - Learning method is more important than attribute subset used during learning
- Errors follow a “log-normal distribution”:
  - So the next version of PREDICT needs another learner.
    - “Naïve Bayes with single Gaussian kernel estimator” with a “logNums” pre-filter: replace all nums with log(nums)

Do you disagree with these results?

- On what basis do you disagree?
- This talk:
  - Conclusions from level5 of the data maturity model
- The data used for your conclusions:
  - As mature?