Issues in Empirical Machine Learning Research

Antal van den Bosch

ILK / Language and Information Science
Tilburg University, The Netherlands

SIKS - 22 November 2006

Issues in ML Research

- A brief introduction
- (Ever) progressing insights from past 10 years:
  - The curse of interaction
  - Evaluation metrics
  - Bias and variance
  - There's no data like more data

Machine learning

- Subfield of artificial intelligence
  - Identified by Alan Turing in seminal 1950 article *Computing Machinery and Intelligence*
- (Langley, 1995; Mitchell, 1997)
- Algorithms that learn from examples
  - Given task T, and an example base E of examples of T (input-output mappings: supervised learning)
  - Learning algorithm L is better in task T after learning

Machine learning: Roots

- Parent fields:
  - Information theory
  - Artificial intelligence
  - Pattern recognition
  - Scientific discovery
- Took off during 70s
- Major algorithmic improvements during 80s
- Forking: neural networks, data mining

Machine Learning: 2 strands

- Theoretical ML (what can be proven to be learnable by what?)
  - Gold, *identification in the limit*
  - Valiant, *probably approximately correct learning*
- Empirical ML (on real or artificial data)
  - Evaluation Criteria:
    - Accuracy
    - Quality of solutions
    - Time complexity
    - Space complexity
    - Noise resistance

Empirical machine learning

- Supervised learning:
  - Decision trees, rule induction, version spaces
  - Instance-based, memory-based learning
  - Hyperplane separators, kernel methods, neural networks
  - Stochastic methods, Bayesian methods
- Unsupervised learning:
  - Clustering, neural networks
- Reinforcement learning, regression, statistical analysis, data mining, knowledge discovery, …

Empirical ML: 2 Flavours

- Greedy
  - Learning
    - abstract model from data
  - Classification
    - apply abstracted model to new data
- Lazy
  - Learning
    - store data in memory
  - Classification
    - compare new data to data in memory

Greedy vs Lazy Learning

Greedy:
- Decision tree induction
  - CART, C4.5
- Rule induction
  - CN2, Ripper
- Hyperplane discriminators
  - Winnow, perceptron, backprop, SVM / Kernel methods
- Probabilistic
  - Naïve Bayes, maximum entropy, HMM, MEMM, CRF
- (Hand-made rulesets)

Lazy:
- *k*-Nearest Neighbour
  - MBL, AM
  - Local regression

Empirical methods

- Generalization performance:
  - How well does the classifier do on UNSEEN examples?
  - (test data: i.i.d - independent and identically distributed)
  - Testing on training data is not *generalization*, but *reproduction* ability
- How to measure?
  - Measure on separate test examples drawn from the same population of examples as the training examples
  - But, avoid single luck; the measurement is supposed to be a trustworthy estimate of the real performance on *any* unseen material.

n-fold cross-validation

- (Weiss and Kulikowski, *Computer systems that learn*, 1991)
- Split example set in *n* equal-sized partitions
- For each partition,
  - Create a training set of the other *n*-1 partitions, and train a classifier on it
  - Use the current partition as test set, and test the trained classifier on it
  - Measure generalization performance
- Compute average and standard deviation on the *n* performance measurements

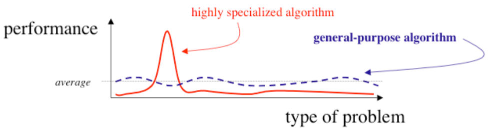
Significance tests

- Two-tailed paired *t*-tests work for comparing 2 10-fold CV outcomes
  - But many type-I errors (false hits)
- Or 2 x 5-fold CV (Salzberg, *On Comparing Classifiers: Pitfalls to Avoid and a Recommended Approach*, 1997)
- Other tests: McNemar, Wilcoxon sign test
- Other statistical analyses: ANOVA, regression trees
- Community determines what is *en vogue*

No free lunch

- (Wolpert, Schaffer; Wolpert & Macready, 1997)
  - No single method is going to be best in all tasks
  - No algorithm is always better than another one
  - No point in declaring victory
- But:
  - Some methods are more suited for some types of problems
  - No rules of thumb, however
  - Extremely hard to meta-learn too

No free lunch



(From Wikipedia)

Issues in ML Research

- A brief introduction
- (Ever) progressing insights from past 10 years:
  - **The curse of interaction**
  - Evaluation metrics
  - Bias and variance
  - There's no data like more data

Algorithmic parameters

- Machine learning meta problem:
  - Algorithmic parameters change bias
    - Description length and noise bias
    - Eagerness bias
  - Can make quite a difference (Daelemans, Hoste, De Meulder, & Naudts, ECML 2003)
  - Different parameter settings = functionally different system
  - But good settings not predictable

Daelemans *et al.* (2003): Diminutive inflection

	Ripper	TiMBL
Default	96.3	96.0
Feature selection	96.7	97.2
Parameter optimization	97.3	97.8
Joint	97.6	97.9

WSD (line)

Similar: little, make, then, time, …

	Ripper	TiMBL
Default	21.8	20.2
Optimized parameters	22.6	27.3
Optimized features	20.2	34.4
Optimized parameters + FS	33.9	38.6

Known solution

- Classifier wrapping (Kohavi, 1997)
  - Training set → train & validate sets
  - Test different setting combinations
  - Pick best-performing
- Danger of overfitting
  - When improving on training data, while *not* improving on test data
- Costly

Optimizing wrapping

- Worst case: exhaustive testing of "all" combinations of parameter settings (pseudo-exhaustive)
- Simple optimization:
  - Not test all settings

Optimized wrapping

- Worst case: exhaustive testing of "all" combinations of parameter settings (pseudo-exhaustive)
- Optimizations:
  - ~~Not test all settings~~
  - Test all settings in less time

Optimized wrapping

- Worst case: exhaustive testing of "all" combinations of parameter settings (pseudo-exhaustive)
- Optimizations:
  - ~~Not test all settings~~
  - Test all settings in less time
  - With less data

Progressive sampling

- Provost, Jensen, & Oates (1999)
- Setting:
  - 1 algorithm (parameters already set)
  - Growing samples of data set
- Find point in learning curve at which no additional learning is needed

Wrapped progressive sampling

- (Van den Bosch, 2004)
- Use **increasing** amounts of data
- While validating **decreasing** numbers of setting combinations
- E.g.,
  - Test "all" settings combinations on a small but sufficient subset
  - Increase amount of data stepwise
  - At each step, discard lower-performing setting combinations

Procedure (1)

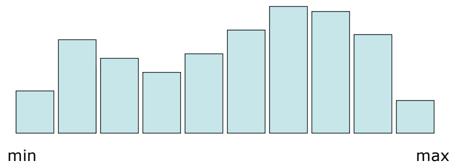
- Given training set of labeled examples,
  - Split internally in 80% training and 20% held-out set
  - Create clipped parabolic sequence of sample sizes
    - *n* steps → multipl. factor n^{th} root of 80% set size
    - Fixed start at 500 train / 100 test
    - E.g. {500, 698, 1343, 2584, 4973, 9572, 18423, 35459, 68247, 131353, 252812, 486582}
    - Test sample is always 20% of train sample

Procedure (2)

- Create pseudo-exhaustive pool of all parameter setting combinations
- Loop:
  - Apply current pool to current train/test sample pair
  - Separate good from bad part of pool
  - Current pool := good part of pool
  - Increase step
- Until one best setting combination left, or all steps performed (random pick)

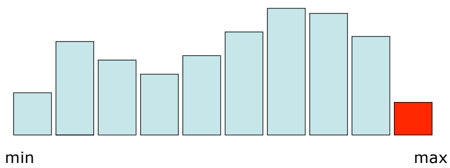
Procedure (3)

- Separate the good from the bad:



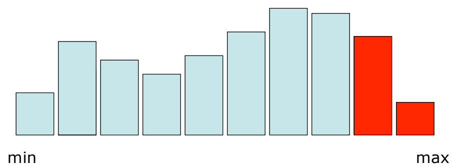
Procedure (3)

- Separate the good from the bad:



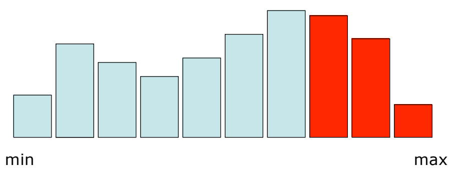
Procedure (3)

- Separate the good from the bad:



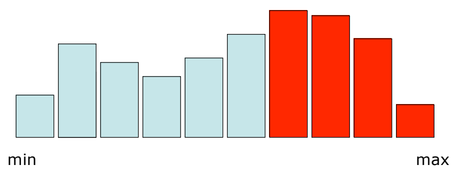
Procedure (3)

- Separate the good from the bad:



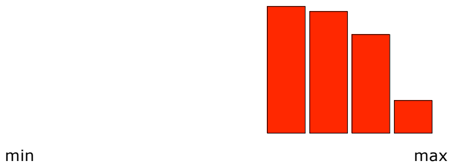
Procedure (3)

- Separate the good from the bad:

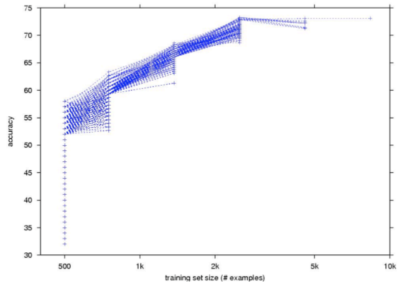


Procedure (3)

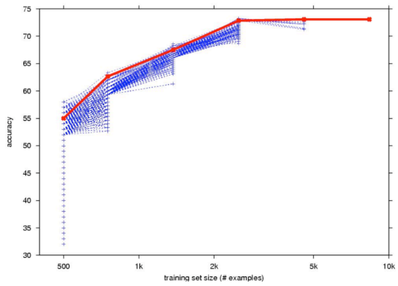
- Separate the good from the bad:



"Mountaineering competition"



"Mountaineering competition"



Customizations

algorithm	# parameters	Total # setting combinations
Ripper (Cohen, 1995)	6	648
C4.5 (Quinlan, 1993)	3	360
Maxent (Giuasu et al, 1985)	2	11
Winnow (Littlestone, 1988)	5	1200
IB1 (Aha et al, 1991)	5	925

Experiments: datasets

Task	# Examples	# Features	# Classes	Class entropy
audiology	228	69	24	3.41
bridges	110	7	8	2.50
soybean	685	35	19	3.84
tic-tac-toe	960	9	2	0.93
votes	437	16	2	0.96
car	1730	6	4	1.21
connect-4	67559	42	3	1.22
kr-vs-kp	3197	36	2	1.00
splice	3192	60	3	1.48
nursery	12961	8	5	1.72

Experiments: results

	normal wrapping		WPS	
Algorithm	Error reduction	Reduction/ combination	Error reduction	Reduction/ combination
Ripper	16.4	0.025	27.9	0.043
C4.5	7.4	0.021	7.7	0.021
Maxent	5.9	0.536	0.4	0.036
IB1	30.8	0.033	31.2	0.034
Winnow	17.4	0.015	32.2	0.027

Discussion

- Normal wrapping and WPS improve generalization accuracy
  - A bit with a few parameters (Maxent, C4.5)
  - More with more parameters (Ripper, IB1, Winnow)
  - 13 significant wins out of 25;
  - 2 significant losses out of 25
- Surprisingly close ([0.015 - 0.043]) average error reductions per setting

Issues in ML Research

- A brief introduction
- (Ever) progressing insights from past 10 years:
  - The curse of interaction
  - **Evaluation metrics**
  - Bias and variance
  - There's no data like more data

Evaluation metrics

- Estimations of generalization performance (on unseen material)
- Dimensions:
  - Accuracy or more task-specific metric
    - Skewed class distribution
    - Two classes vs multi-class
  - Single or multiple scores
    - *n*-fold CV, leave_one_out
    - Random splits
    - Single splits
  - Significance tests

Accuracy is bad

- Higher accuracy / lower error rate does not necessarily imply better performance on target task
- "*The use of error rate often suggests insufficiently careful thought about the real objectives of the research*" - David Hand, *Construction and Assessment of Classification Rules* (1997)

Other candidates?

- Per-class statistics using true and false positives and negatives
  - Precision, recall, F-score
  - ROC, AUC
- Task-specific evaluations
- Cost, speed, memory use, accuracy within time frame

True and false positives

True class

Hypothesized class		p	n
	Y	True Positives	False Positives
	N	False Negatives	True Negatives
Column totals:		P	N

$\text{fp rate} = \frac{FP}{N}$ $\text{tp rate} = \frac{TP}{P}$

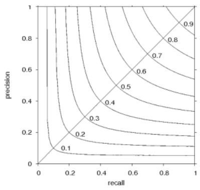
$\text{precision} = \frac{TP}{TP+FP}$ $\text{recall} = \frac{TP}{P}$

$\text{accuracy} = \frac{TP+TN}{P+N}$ $\text{F-measure} = \frac{2}{1/\text{precision}+1/\text{recall}}$

F-score is better

- When your problem is expressible as a per-class precision and recall problem
- (like in IR, Van Rijsbergen, 1979)

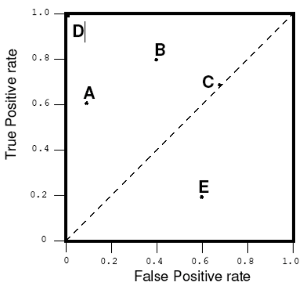
$$F_{\beta=1} = \frac{2pr}{p+r}$$



ROC is the best

- **R**eceiver **O**perating **C**haracteristics
- E.g.
  - ECAI 2004 workshop on ROC
  - Fawcett's (2004) ROC 101
- Like precision/recall/F-score, suited "for domains with skewed class distribution and unequal classification error costs."

ROC curve



True and false positives

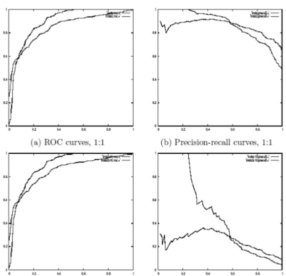
		True class: p	True class: n
Hypothesized class	Y	True Positives	False Positives
	N	False Negatives	True Negatives
Column totals:		P	N

fp rate $= \frac{FP}{N}$ tp rate $= \frac{TP}{P}$

precision $= \frac{TP}{TP+FP}$ recall $= \frac{TP}{P}$

accuracy $= \frac{TP+TN}{P+N}$ F-measure $= \frac{2}{1/\text{precision}+1/\text{recall}}$

ROC is better than p/r/F

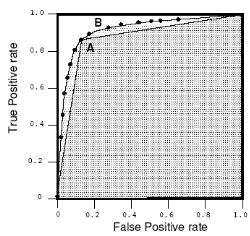


(a) ROC curves, 1:1 (b) Precision-recall curves, 1:1

(c) ROC curves, 1:10 (d) Precision-recall curves, 1:10

AUC, ROC's F-score

- **A**rea **U**nder the **C**urve



Multiple class AUC?

- AUC per class, *n* classes:
- Macro-average: sum(AUC (c_1) + … + AUC(c_n))/*n*
- Micro-average:

$$AUC_{total} = \sum_{c_i \in C} AUC(c_i) \cdot p(c_i)$$

F-score vs AUC

- Which one is better actually depends on the task.
- Examples by Reynaert (2005), spell checker performance on fictitious text with 100 errors:

System	Flagged	Corrected	Recall	Precision	F-score	AUC
A	10,000	100	1	0.01	0.02	0.750
B	100	50	0.5	0.5	0.5	0.747

Significance & F-score

- *t*-tests are valid on accuracy and recall
- But are invalid on precision and F-score
- Accuracy is bad; recall is only half the story
- Now what?

Randomization tests

- (Noreen, 1989; Yeh, 2000; Tjong Kim Sang, CoNLL shared task; *stratified shuffling*)
- Given classifier's output on a *single* test set,
  - Produce many small subsets
  - Compute standard deviation
- Given two classifiers' output,
  - Do as above
  - Compute significance (Noreen, 1989)

So?

- Does Noreen's method work with AUC? We tend to think so
- Incorporate AUC in evaluation scripts
- Favor Noreen's method in
  - "shared task" situations (single test sets)
  - F-score / AUC estimations (skewed classes)
- Maintain matched paired *t*-tests where accuracy is still OK.

Issues in ML Research

- A brief introduction
- (Ever) progressing insights from past 10 years:
  - The curse of interaction
  - Evaluation metrics
  - **Bias and variance**
  - There's no data like more data

Bias and variance

Two meanings!

1. **Machine learning bias and variance** - the degree to which an ML algorithm is flexible in adapting to data
2. **Statistical bias and variance** - the balance between systematic and variable errors

Machine learning bias & variance

- Naïve Bayes:
  - High bias (strong assumption: feature independence)
  - Low variance
- Decision trees & rule learners:
  - Low bias (adapt themselves to data)
  - High variance (changes in training data can cause radical differences in model)

Statistical bias & variance

- Decomposition of a classifier's error:
  - Intrinsic error: intrinsic to the data. Any classifier would make these errors (*Bayes error*)
  - Bias error: recurring error, systematic error, independent of training data.
  - Variance error: non-systematic error; variance in error, averaged over training sets.
- E.g. Kohavi and Wolpert (1996), Bias Plus Variance Decomposition for Zero-One Loss Functions, Proc. of ICML
  - Keep test set constant, and vary training set many times

Variance and overfitting

- Being too faithful in reproducing the classification in the training data
  - Does not help generalization performance on unseen data - **overfitting**
  - Causes high **variance**
- Feature selection (discarding unimportant features) helps avoiding overfitting, thus lowers variance
- Other "smoothing bias" methods:
  - Fewer nodes in decision trees
  - Fewer units in hidden layers in MLP

Relation between the two?

- Suprisingly, NO!
  - A high machine learning bias does not lead to a low number or portion of bias errors.
  - A high bias is not necessarily good; a high variance is not necessarily bad.
  - In the literature: bias error often surprisingly equal for algorithms with very different machine learning bias

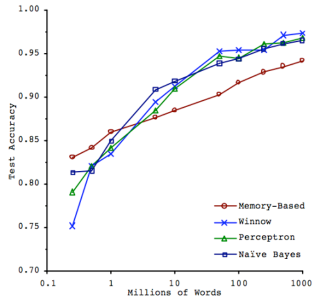
Issues in ML Research

- A brief introduction
- (Ever) progressing insights from past 10 years:
  - The curse of interaction
  - Evaluation metrics
  - Bias and variance
  - **There's no data like more data**

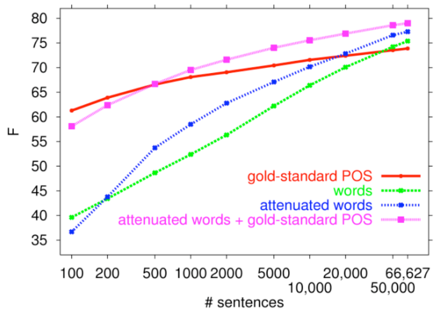
There's no data like more data

- Learning curves
  - At different amounts of training data,
  - algorithms attain different scores on test data
  - (recall Provost, Jensen, Oats 1999)
- Where is the ceiling?
- When not at the ceiling, do differences between algorithms matter?

Banko & Brill (2001)



Van den Bosch & Buchholz (2002)



Learning curves

- Tell more about
  - the task
  - features, representations
  - how much more data needs to be gathered
  - scaling abilities of learning algorithms
- Relativity of differences found at point when learning curve has not flattened

Closing comments

- Standards and norms in experimental & evaluative methodology in empirical research fields always on the move
- *Machine learning* and *search* are sides of the same coin
- Scaling abilities of ML algorithms is an underestimated dimension

Software available at http://ilk.uvt.nl

- paramsearch 1.0 (WPS)
- TiMBL 5.1

Antal.vdnBosch@uvt.nl

Universiteit van Tilburg

Credits

- Curse of interaction: Véronique Hoste and Walter Daelemans (University of Antwerp)
- Evaluation metrics: Erik Tjong Kim Sang (University of Amsterdam), Martin Reynaert (Tilburg University)
- Bias and variance: Iris Hendrickx (University of Antwerp), Maarten van Someren (University of Amsterdam)
- There's no data like more data: Sabine Buchholz (Toshiba Research)