NeuroComp
Machine Learning and Validation

Michèle Sebag

http://tao.lri.fr/tiki-index.php

Nov. 16th 2011

Validation, the questions

1. What is the result ?
2. My results look good. Are they ?
3. Does my system outperform yours ?
4. How to set up my system ?

Contents

Contents

Supervised Machine Learning

Context

World $\rightarrow$ instance $\mathbf{x}_i$ $\rightarrow$ Oracle $\downarrow$ y_i

Input: Training set $\mathcal{E} = \{(\mathbf{x_i}, y_i), i = 1 \dots n, x_i \in \mathcal{X}, y_i \in \mathcal{Y}\}$

Output: Hypothesis $h : \mathcal{X} \mapsto \mathcal{Y}$

Criterion: few mistakes (details later)

Definitions

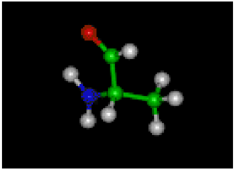
Example

- row : example/ case
- column : feature/variables/attribute
- attribute : class/label

age	employme	education	edun	marital	...	job	relation	race	gender	hour	country	wealth
					...							
39	State_gov	Bachelors	13	Never_mar	...	Adm_cleric	Not_in_fam	White	Male	40	United_Sta	poor
51	Self_emp_	Bachelors	13	Married	...	Exec_man	Husband	White	Male	13	United_Sta	poor
39	Private	HS_grad	9	Divorced	...	Handlers_c	Not_in_fam	White	Male	40	United_Sta	poor
54	Private	11th	7	Married	...	Handlers_c	Husband	Black	Male	40	United_Sta	poor
28	Private	Bachelors	13	Married	...	Prof_speci	Wife	Black	Female	40	Cuba	poor
38	Private	Masters	14	Married	...	Exec_man	Wife	White	Female	40	United_Sta	poor
50	Private	9th	5	Married_sp	...	Other_serv	Not_in_fam	Black	Female	16	Jamaica	poor
52	Self_emp_	HS_grad	9	Married	...	Exec_man	Husband	White	Male	45	United_Sta	rich
31	Private	Masters	14	Never_mar	...	Prof_speci	Not_in_fam	White	Female	50	United_Sta	rich
42	Private	Bachelors	13	Married	...	Exec_man	Husband	White	Male	40	United_Sta	rich
37	Private	Some_coll	10	Married	...	Exec_man	Husband	Black	Male	80	United_Sta	rich
30	State_gov	Bachelors	13	Married	...	Prof_speci	Husband	Asian	Male	40	India	rich
24	Private	Bachelors	13	Never_mar	...	Adm_cleric	Own_child	White	Female	30	United_Sta	poor
33	Private	Assoc_acc	12	Never_mar	...	Sales	Not_in_fam	Black	Male	50	United_Sta	poor
41	Private	Assoc_voc	11	Married	...	Craft_repai	Husband	Asian	Male	40	*MissingV	rich
34	Private	7th_8th	4	Married	...	Transport_	Husband	Amer_India	Male	45	Mexico	poor
26	Self_emp_	HS_grad	9	Never_mar	...	Farming_fi	Own_child	White	Male	35	United_Sta	poor
33	Private	HS_grad	9	Never_mar	...	Machine_c	Unmarried	White	Male	40	United_Sta	poor
38	Private	11th	7	Married	...	Sales	Husband	White	Male	50	United_Sta	poor
44	Self_emp_	Masters	14	Divorced	...	Exec_man	Unmarried	White	Female	45	United_Sta	rich
41	Private	Doctorate	16	Married	...	Prof_speci	Husband	White	Male	60	United_Sta	rich
:	:	:	:	:	:	:	:	:	:	:	:	:

Instance space $\mathcal{X}$

- Propositionnal : $\mathcal{X} \equiv \mathbb{R}^d$
- Relational : ex. chemistry.



molecule: alanine

Contents

Difficulty factors

Quality of examples / of representation

+ Relevant features — Feature extraction
− Not enough data
− Noise ; missing data
− Structured data : spatio-temporal, relational, textual, videos ..

Distribution of examples

+ Independent, identically distributed examples
− Other: robotics; data stream; heterogeneous data

Prior knowledge

+ Constraints on sought solution
+ Criteria; loss function

Difficulty factors, 2

Learning criterion

+ Convex function: a single optimum
↘ Complexity : n, $nlogn$, n^2 — Scalability
− Combinatorial optimization

What is your agenda ?

- Prediction performance
- Causality
- INTELLIGIBILITY
- Simplicity
- Stability
- Interactivity, visualisation

Difficulty factors, 3

Crossing the chasm

- There exists no *killer algorithm*
- Few general recommendations about algorithm selection

Performance criteria

- Consistency

When number n of examples goes to ∞
and the target concept h^* is in $\mathcal{H}$
Algorithm finds $\hat{h}_n$, with

$$lim_{n \to \infty} h_n = h^*$$

- Convergence speed

$$||h^* - h_n|| = \mathcal{O}(1/n), \mathcal{O}(1/\sqrt{n}), \mathcal{O}(1/\ln n)$$

Contents

Context

Related approaches — criteria

- Data Mining, KDD
  scalability
- Statistics and data analysis
  Model selection and fitting; hypothesis testing
- Machine Learning
  Prior knowledge; representations; distributions
- Optimisation
  well-posed / ill-posed problems
- Computer Human Interface
  No ultimate solution: a dialog
- High performance computing
  Distributed data; privacy

Methodology

Phases

1. Collect data — expert, DB
2. Clean data — stat, expert
3. Select data — stat, expert
4. Data Mining / Machine Learning
   - Description — *what is in data ?*
   - Prediction — *Decide for one example*
   - Agregate — *Take a global decision*
5. Visualisation — chm
6. Evaluation — stat, chm
7. Collect new data — expert, stat

An interative process

depending on expectations, data, prior knowledge, current results

Contents

Supervised Machine Learning

Context

$$\text{World} \rightarrow \text{instance } \mathbf{x}_i \rightarrow \begin{array}{c}\text{Oracle}\\ \downarrow \\ y_i\end{array}$$

Input

Training set $\mathcal{E} = \{(\mathbf{x}_i, y_i), i = 1 \ldots n, \mathbf{x}_i \in \mathcal{X}, y_i \in \mathcal{Y}\}$

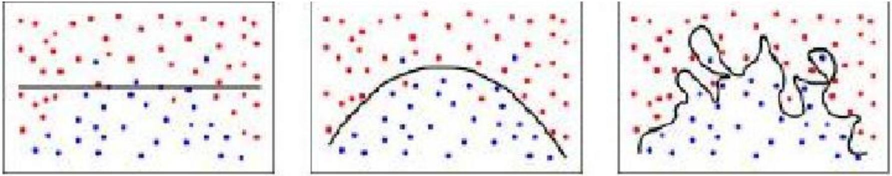
Tasks

- Select hypothesis space $\mathcal{H}$
- Assess hypothesis $h \in \mathcal{H}$ $\quad$ $score(h)$
- Find best hypothesis h^*

What is the point ?

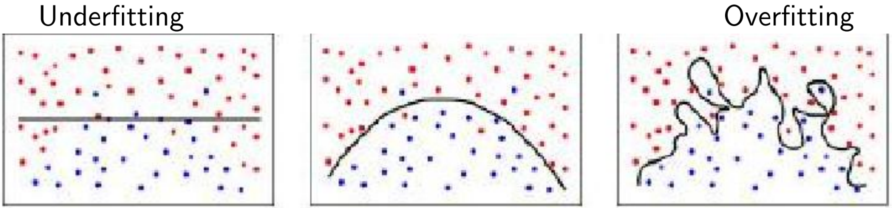
Underfitting

Overfitting



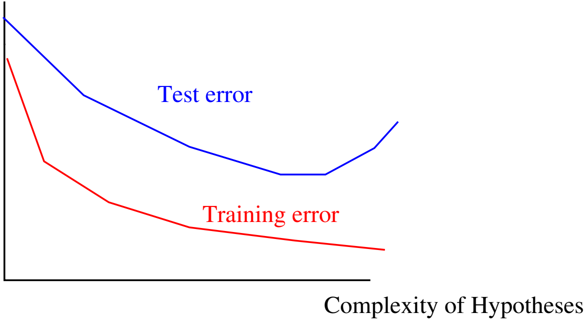
The point is not to be perfect on the training set

What is the point ?



The point is not to be perfect on the training set

The villain: overfitting



What is the point ?

Prediction good on future instances

Necessary condition:

Future instances must be similar to training instances

"identically distributed"

Minimize (cost of) errors $\ell(y, h(x)) \geq 0$

not all mistakes are equal.

Error: theoretical approach

Minimize expectation of error cost

$$\text{Minimize } E[\ell(y, h(x))] = \int_{X \times Y} \ell(y, h(x)) p(x, y) dx\, dy$$

Error: theoretical approach

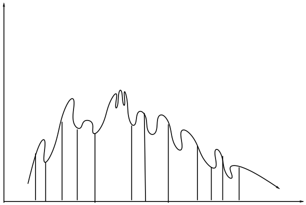
Minimize expectation of error cost

$$\text{Minimize } E[\ell(y, h(x))] = \int_{X \times Y} \ell(y, h(x)) p(x, y) dx\, dy$$

Principle

Si h “is well-behaved“ on $\mathcal{E}$, and h is "sufficiently regular" h will be well-behaved in expectation.

$$E[F] \leq \frac{\sum_{i=1}^{n} F(x_i)}{n} + c(F, n)$$



Classification, Problem posed

INPUT $\sim P(x, y)$

$$\mathcal{E} = \{(x_i, y_i), x_i \in \mathcal{X}, y_i \in \{0, 1\}, i = 1 \ldots n\}$$

HYPOTHESIS SPACE SEARCH SPACE

$$\mathcal{H} \quad h : \mathcal{X} \mapsto \{0, 1\}$$

LOSS FUNCTION

$$\ell : \mathcal{Y} \times \mathcal{Y} \mapsto \mathbb{R}$$

OUTPUT

$$h^* = arg\ max\{score(h), h \in \mathcal{H}\}$$

Classification, criteria

Generalisation error

$$Err(h) = E[\ell(y, h(x))] = \int \ell(y, h(x)) dP(x, y)$$

Empirical error

$$Err_e(h) = \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, h(x_i))$$

Bound

risk minimization

$$Err(h) < Err_e(h) + \mathcal{F}(n, d(\mathcal{H}))$$

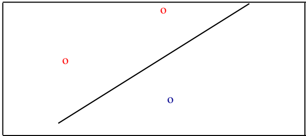
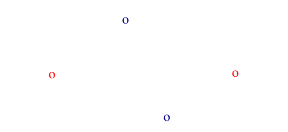
$d(\mathcal{H})$ = VC-dimension of $\mathcal{H}$

Dimension of Vapnik Cervonenkis

Principle Given hypothesis space $\mathcal{H}$: $\mathcal{X} \mapsto \{0,1\}$ Given n points $x_1, \ldots, x_n$ in $\mathcal{X}$.
If, $\forall (y_i)_{i=1}^n \in \{0,1\}^n, \exists h \in \mathcal{H} / h(x_i) = y_i$,
$\mathcal{H}$ shatters $\{x_1, \ldots, x_n\}$

Example: $\mathcal{X} = \mathbb{R}^p$
$d(\text{hyperplanes in } \mathbb{R}^p) = p + 1$

WHY: if $\mathcal{H}$ shatters $\mathcal{E}$, $\mathcal{E}$ doesn't tell anything

	
3 pts shattered by a line	4 points, non shattered

Definition

$$d(\mathcal{H}) = max\{n / \exists (x_1 \ldots, x_n) \text{ shattered by } \mathcal{H}\}$$

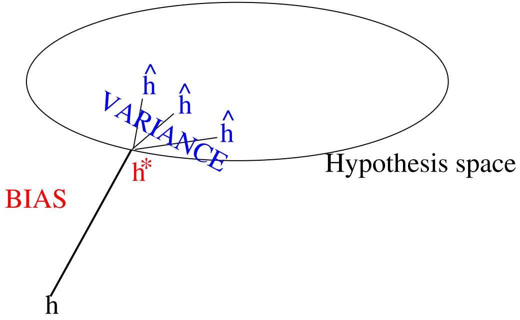
Classification: Ingredients of error

Bias

Bias $(\mathcal{H})$: error of the best hypothesis h^* in $\mathcal{H}$

Variance

Variance of h_n depending on $\mathcal{E}$



Optimization

negligible in small scale
takes over in large scale (Google)

Contents

Validation: Three questions

Define a good indicator of quality

- Misclassification cost
- Area under the ROC curve

Computing an estimate thereof

- Validation set
- Cross-Validation
- Leave one out
- Bootstrap

Compare estimates: Tests and confidence levels

Which indicator, which estimate: it depends.

Settings

- Large/few data

Data distribution

- Dependent/independent examples
- balanced/imbalanced classes

Contents

Performance indicators

Binary class

- h^* the truth
- $\hat{h}$ the learned hypothesis

Confusion matrix

$\hat{h}$ / h^*	1	0	
1	a	b	a + b
0	c	d	c+d
	a+c	b+d	a + b + c + d

Performance indicators, 2

$\hat{h}$ / h^*	1	0	
1	a	b	a + b
0	c	d	c+d
	a+c	b+d	a + b + c + d

- Misclassification rate $\frac{b+c}{a+b+c+d}$
- Sensitivity, True positive rate (TP) $\frac{a}{a+c}$
- Specificity, False negative rate (FN) $\frac{b}{b+d}$
- Recall $\frac{a}{a+c}$
- Precision $\frac{a}{a+b}$

Note: always compare to random guessing / baseline alg.

Performance indicators, 3

The Area under the ROC curve

- ROC: Receiver Operating Characteristics
- Origin: Signal Processing, Medicine

Principle

$$h : X \mapsto \mathbb{R} \qquad h(x) \text{ measures the risk of patient } x$$

h leads to order the examples:

$$+ + + - + - + + + + - - - + - - - + - - - - - - - - - - - -$$

Performance indicators, 3

The Area under the ROC curve

- ROC: Receiver Operating Characteristics
- Origin: Signal Processing, Medicine

Principle

$$h : X \mapsto \mathbb{R} \qquad h(x) \text{ measures the risk of patient } x$$

h leads to order the examples:

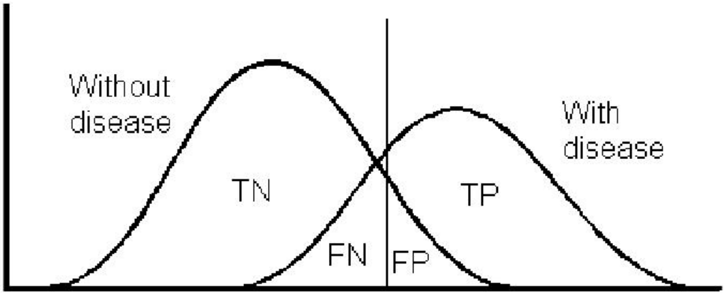
$$+ + + - + - + + + + - - - + - - - + - - - - - - - - - - - -$$

Given a threshold θ, h yields a classifier: Yes iff $h(x) > \theta$.

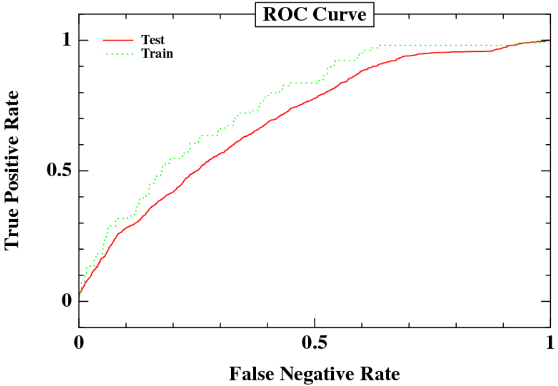
$$+ + + - + - + + + + \mid - - - + - - - + - - - - - - - - - - - -$$

Here, TP $(\theta) = .8$; FN $(\theta) = .1$

ROC



The ROC curve



Ideal classifier: (0 False negative,1 True positive)
Diagonal (True Positive = False negative) $\equiv$ nothing learned.

ROC Curve, Properties

Properties

ROC depicts the trade-off True Positive / False Negative.

Standard: misclassification cost (Domingos, KDD 99)

$$\text{Error} = \#\ \text{false positive} + c \times\ \#\ \text{false negative}$$

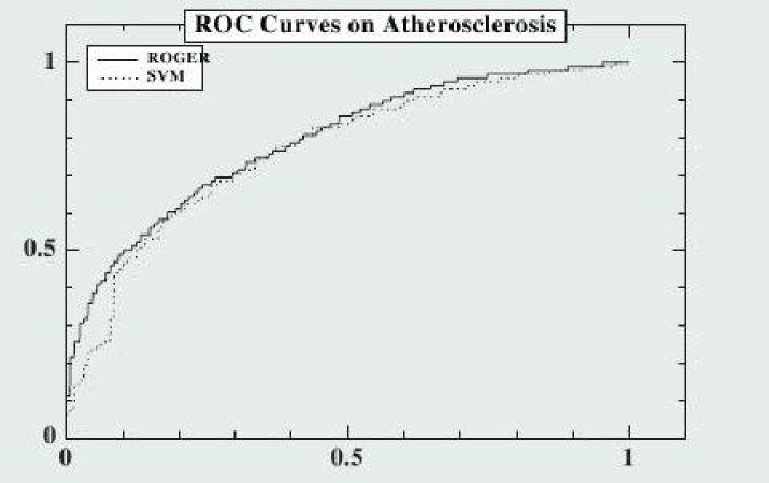
In a multi-objective perspective, ROC = Pareto front.

Best solution: intersection of Pareto front with $\Delta(-c, -1)$

ROC Curve, Properties, foll'd

Used to compare learners — Bradley 97

- multi-objective-like
- insensitive to imbalanced distributions
- shows sensitivity to error cost.



Area Under the ROC Curve

Often used to select a learner

Don't ever do this ! Hand, 09

Sometimes used as learning criterion Mann Whitney Wilcoxon

$$AUC = Pr(h(x) > h(x')|y > y')$$

WHY Rosset, 04

- More stable $\mathcal{O}(n^2)$ vs $\mathcal{O}(n)$
- With a probabilistic interpretation Clemençon et al. 08

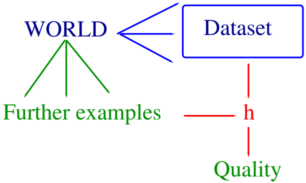
HOW

- SVM-Ranking Joachims 05; Usunier et al. 08, 09
- Stochastic optimization

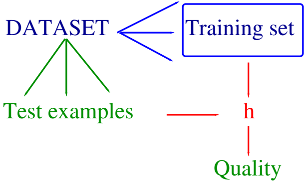
Contents

Validation, principle

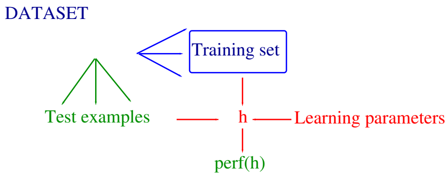
Desired: performance on further instances



Assumption: Dataset is to World, like Training set is to Dataset.

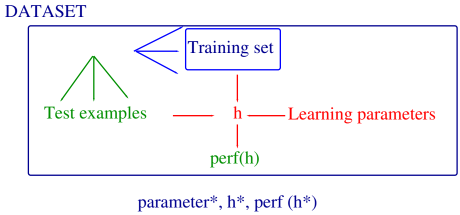


Validation, 2



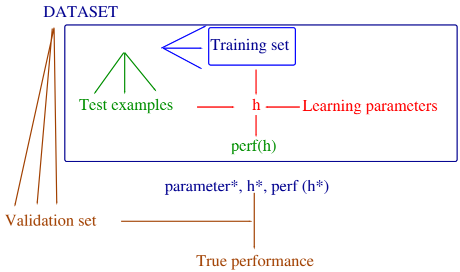
Unbiased Assessment of Learning Algorithms
T. Scheffer and R. Herbrich, 97

Validation, 2



Unbiased Assessment of Learning Algorithms
T. Scheffer and R. Herbrich, 97

Validation, 2



Unbiased Assessment of Learning Algorithms
T. Scheffer and R. Herbrich, 97

Contents

Confidence intervals

Definition

Given a random variable X on $\mathbb{R}$, a p%-confidence interval is $I \subset \mathbb{R}$ such that

$$Pr(X \in I) > p$$

Binary variable with probability ϵ

Probability of r events out of n trials:

$$P_n(r) = \frac{n!}{r!(n-r)!}\epsilon^r(1-\epsilon)^{n-r}$$

- Mean: $n\epsilon$
- Variance: $\sigma^2 = n\epsilon(1-\epsilon)$

Gaussian approximation

$$P(x) = \frac{1}{\sqrt{2\pi\sigma^2}} exp^{-\frac{1}{2}\frac{x-\mu}{\sigma}^2}$$

Confidence intervals

Bounds on (true value, empirical value) for n trials, $n > 30$

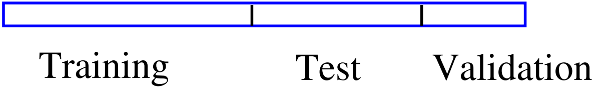
$$Pr(|\hat{x}_n - x^*| > \underbrace{1.96}_{z} \sqrt{\frac{\hat{x}_n.(1-\hat{x}_n)}{n}}) < \underbrace{.05}_{\varepsilon}$$

Table

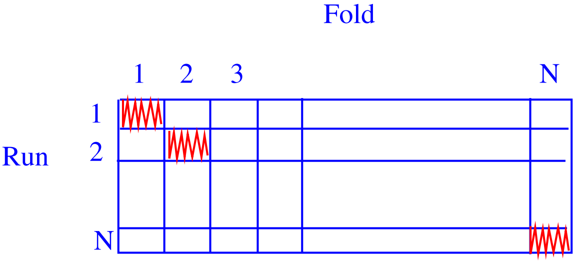
z	.67	1.	1.28	1.64	1.96	2.33	2.58
ε	50	32	20	10	5	2	1

Empirical estimates

When data abound (MNIST)



Cross validation



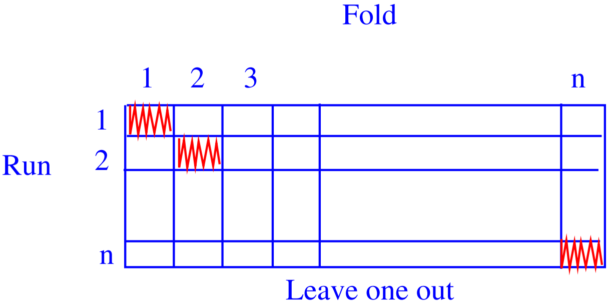
N–fold Cross Validation

Error = Average (error on  of h

learned from )

Empirical estimates, foll'd

Cross validation → Leave one out



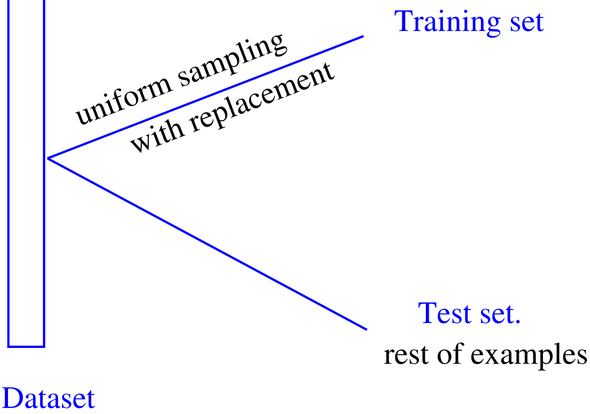
Same as N-fold CV, with N = number of examples.

Properties

Low bias; high variance; underestimate error if data not independent

Empirical estimates, foll'd

Bootstrap



Average indicator over all (Training set, Test set) samplings.

Contents

Is $\hat{h}$ better than random ?

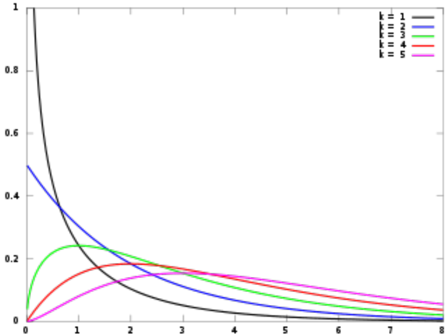
The McNemar test

McNemar 47

$\hat{h}$ / h^*	1	0	
1	a	b	a + b
0	c	d	c+d
	a+c	b+d	a + b + c + d

Property

$\frac{|b-c|-1}{b+c}$ follows a χ^2 law with degre of freedom 1



Types of test error

Type I error
The hypothesis is not significant, and the test thinks it's significant

Type II error
The hypothesis is valid, and the test discards it.

Comparing algorithms A and B

	A	B	A-B
run 1	30	28	2
run 2	17	25	-8
	28	25	3
	17	28	-11
	30	26	4

Assumption

A and B have normal distribution

Simplest case

two samples with same size, (quasi) same variance.

Define

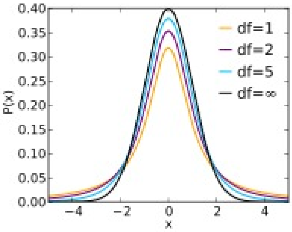
$$t = \frac{\bar{A} - \bar{B}}{S_{A,B} \cdot \sqrt{\frac{2}{n}}}$$

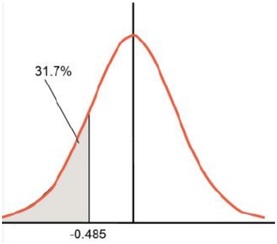
with $S_{A,B} = \sqrt{\frac{1}{2}(S_A^2 + S_B^2)}$ and $S_A^2 = \frac{1}{n}\sum(A_i - \bar{A})^2$

Comparing algorithms A and B

t follows a Student law with (2n-2)-dof

- Compute t
- See confidence of t





Comparing algorithms A and B

Recommended: Use paired t-test

- Apply A and B with same (training, test) sets
- Variance is lower:

$$Var(A - B) = Var(A) + Var(B) - 2coVar(A, B)$$

- Thus easier to make significant differences

What if variances are different ?

See Welch' test:

$$\frac{\bar{A} - \bar{B}}{\sqrt{\frac{S_A^2}{N_A} + \frac{S_B^2}{N_B}}}$$

Summary: single dataset (if we had enough data...)

The 5 x 2CV — Dietterich 98

- 5 times
- split the data into 2 halves
- gives 10 estimates of error indicator

\+ More independent

− Each training set is 1/2 data.

With a single dataset

- 5x2 CV
- paired t-test
- McNemar test on a validation set

Multiple datasets

If A and B results don't follow a normal distribution

$$Z_i = A_i - B_i$$

A	B	|Z|	rank	sign
19	23	4	6th	$-$
22	21	1	1st	$+$
21	19	2	2nd	$+$
25	28	3	4th	$-$
24	22	2	2nd	$+$
23	20	3	4th	$+$

Wilcoxon signed rank test

1. Rank the $|Z_i|$
2. $W_+ =$ sum of ranks when $Z_i > 0$
3. $W_- =$ sum of ranks when $Z_i < 0$
4. $W_{min} = min(W_+, W_-)$

$$z = \frac{1/4n(n+1) - W_{min} - 1/2}{\sqrt{1/24n(n+1)(2n+1)}}$$

5. $z \sim \mathcal{N}(0,1)$ $\quad n > 20$

Multiple hypothesis testing

Beware

- If you test many hypotheses on the same dataset
- one of them will appear confidently true...
  increase in type I error

Corrections Over n tests, the global significance level α_{global} is related to the elementary significance level α_{unit}:

$$\alpha_{global} = 1 - (1 - \alpha_{unit})^n$$

- Bonferroni correction pessimistic

$$\alpha_{unit} = \frac{\alpha_{global}}{n}$$

- Sidak correction

$$\alpha_{unit} = 1 - (1 - \alpha_{global})^{\frac{1}{n}}$$

Contents

How to set up my system ?

Parameter tuning

- Setting the parameters for feature extraction
- Select the best learning algorithm
- Setting the learning parameters (e.g. type of kernel, the parameters in SVMs)
- Setting the validation parameters

Goal: find the best setting   a pervasive concern

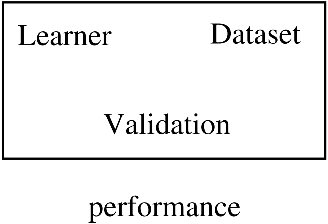
- Algorithm selection in Operational Research
- Parameter tuning in Stochastic Optimization
- Meta-Learning in Machine Learning

From Design of Experiments to ...

Main approaches

1. Design of experiments (Latin square)
2. Anova (Analysis of variance)-like methods:
   - Racing
   - Sequential parameter optimization

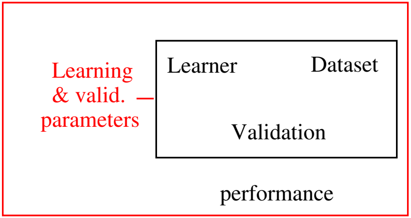
Parameter Tuning: A Meta-Optimization problem



Optimization: the Black-Box Scenario

- Need to perform several runs to compute performance
  Cross-Validation
- Need to specify the # runs and tune it optimally
- Overall cost is the total number of evaluations
- And don't forget to tune the parameters of the meta-optimizer!

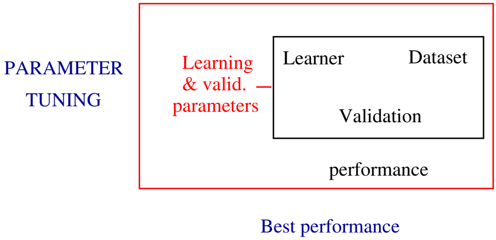
Parameter Tuning: A Meta-Optimization problem



Optimization: the Black-Box Scenario

- Need to perform several runs to compute performance
  Cross-Validation
- Need to specify the # runs and tune it optimally
- Overall cost is the total number of evaluations
- And don't forget to tune the parameters of the meta-optimizer!

Parameter Tuning: A Meta-Optimization problem



Optimization: the Black-Box Scenario

- Need to perform several runs to compute performance
  Cross-Validation
- Need to specify the # runs and tune it optimally
- Overall cost is the total number of evaluations
- And don't forget to tune the parameters of the meta-optimizer!

Ingredients

Design Of Experiments (DOE)

- A long-known method from statistics
- Choose a finite number of parameter sets
- Compute their performance
- Return the *statistically significantly* best sets

Analysis of Variance (ANOVA)

- Assumes normally distributed data
- Tests if means are significantly different

  for a given confidence level; generalizes T-Test
- Perform pairwise tests if ANOVA reports some difference

  T-Test, rank-based tests, ...

DOE: Issues

Choice of sample parameter sets

- *Full Factorial Design*
  - Discretize all parameters if continuous
  - Choose all possible combinations
- *Latin Hypercube Sampling*: to generate k sets,
  - Discretize all parameters in k values
  - Repeat k times:
    for each parameter, (uniformly) choose one value out of k
  - For each parameter, each value is taken once

fine if no correlation

Cost

- For each parameter set, the full cost of learning validation
- Combinatorial explosion with number of parameters and precision

Racing algorithms

Birattari & al. 02, Yuan & Gallagher 04

Rationale

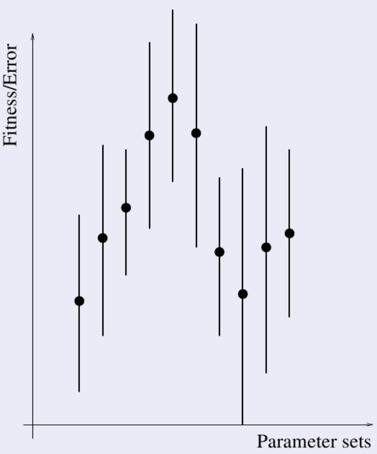
- All parameter settings are run the same number of times
  whereas very bad settings could be detected earlier

Implementation

- Repeat
  - Perform only a few runs per parameter set
  - Statistically check all sets against the best one
    at given confidence level
  - Discard the bad ones
- Until only survivor, or maximum number of runs per setting reached

Racing algorithms

How?

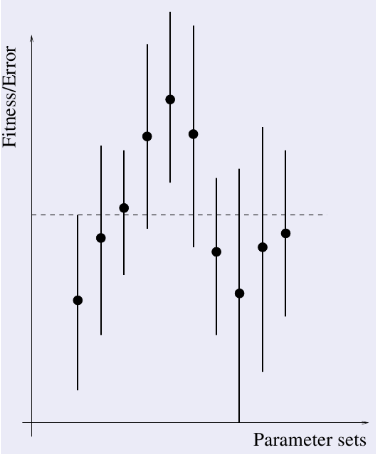


Example: Initialization

- $R = 0$
- while $R < R_{max}$ and more than 1 set
  - Compute empirical value of performance for all sets doing r additional runs
    average, median, ...
  - Compute X% confidence intervals
    Hoeffding bounds, Friedman tests, ...
  - Remove sets whose best possible value is worse than worse possible value of the best empirical set.
  - $R+ = r$

Racing algorithms

How?

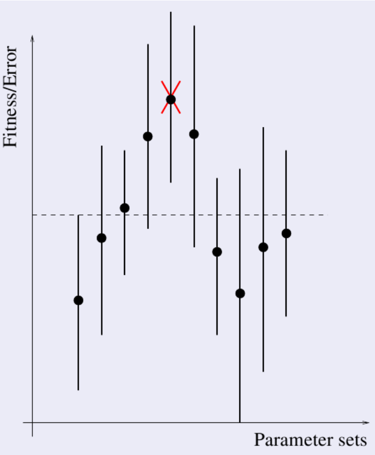


Example: Initialization

- $R = 0$
- while $R < R_{max}$ and more than 1 set
  - Compute empirical value of performance for all sets doing r additional runs
    average, median, ...
  - Compute X% confidence intervals
    Hoeffding bounds, Friedman tests, ...
  - Remove sets whose best possible value is worse than worse possible value of the best empirical set.
  - $R+ = r$

Racing algorithms

How?

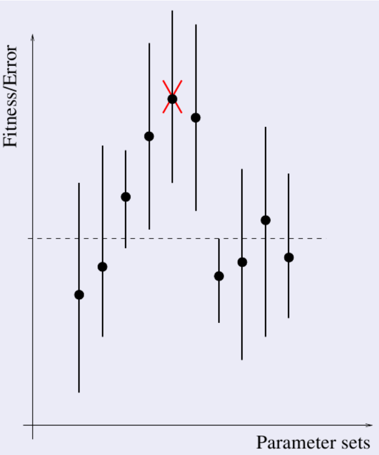


Example: Initialization

- $R = 0$
- while $R < R_{max}$ and more than 1 set
  - Compute empirical value of performance for all sets doing r additional runs
    average, median, ...
  - Compute X% confidence intervals
    Hoeffding bounds, Friedman tests, ...
  - Remove sets whose best possible value is worse than worse possible value of the best empirical set.
  - $R+ = r$

Racing algorithms

How?

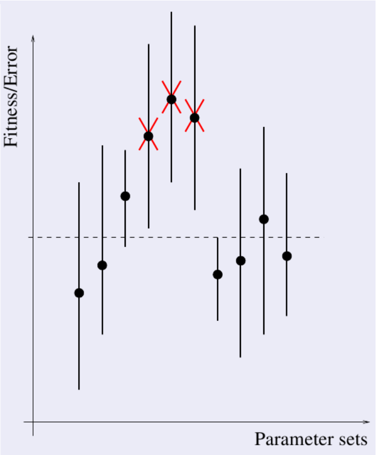


Example: Iteration 1

- $R = 0$
- while $R < R_{max}$ and more than 1 set
  - Compute empirical value of performance for all sets doing r additional runs
    average, median, ...
  - Compute X% confidence intervals
    Hoeffding bounds, Friedman tests, ...
  - Remove sets whose best possible value is worse than worse possible value of the best empirical set.
  - $R+ = r$

Racing algorithms

How?

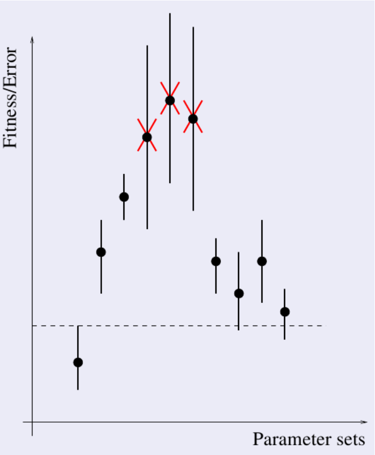


Example: Iteration 1

- $R = 0$
- while $R < R_{max}$ and more than 1 set
  - Compute empirical value of performance for all sets doing r additional runs
    average, median, ...
  - Compute X% confidence intervals
    Hoeffding bounds, Friedman tests, ...
  - Remove sets whose best possible value is worse than worse possible value of the best empirical set.
  - $R+ = r$

Racing algorithms

How?



Example: Iteration N

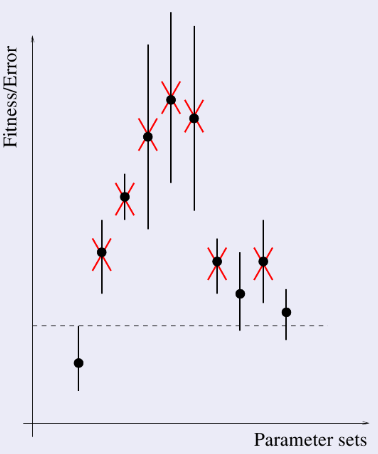
- $R = 0$
- while $R < R_{max}$ and more than 1 set
  - Compute empirical value of performance for all sets doing r additional runs

    average, median, . . .
  - Compute X% confidence intervals

    Hoeffding bounds, Friedman tests, . . .
  - Remove sets whose best possible value is worse than worse possible value of the best empirical set.
  - $R+ = r$

Racing algorithms

How?

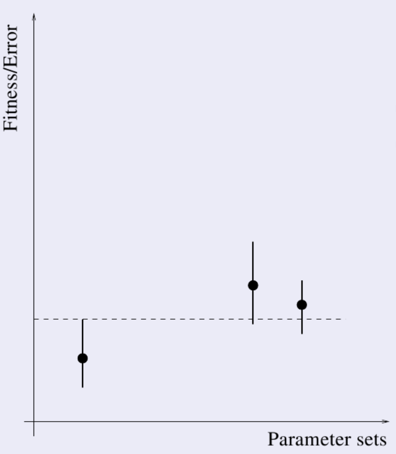


Example: Iteration N

- $R = 0$
- while $R < R_{max}$ and more than 1 set
  - Compute empirical value of performance for all sets doing r additional runs
    average, median, ...
  - Compute X% confidence intervals
    Hoeffding bounds, Friedman tests, ...
  - Remove sets whose best possible value is worse than worse possible value of the best empirical set.
  - $R+ = r$

Racing algorithms

How?



Example: Best parametere sets

- $R = 0$
- while $R < R_{max}$ and more than 1 set
  - Compute empirical value of performance for all sets doing r additional runs
    average, median, ...
  - Compute X% confidence intervals
    Hoeffding bounds, Friedman tests, ...
  - Remove sets whose best possible value is worse than worse possible value of the best empirical set.
  - $R+ = r$

Racing algorithms: Discussion

Results

- Published results claim saving between 50 and 90% of the runs

Useful for

- Multiple algorithms on single problem — for efficiency
- Single algorithm on multiple problems — to assess problem difficulties
- Multiple algorithms on multiple problems — for robustness

Issues

- Nevertheless costly
- Can only find the best one in initial sample

Sequential Parameter Optimization

Bartz-Beielstein & al. 05-07

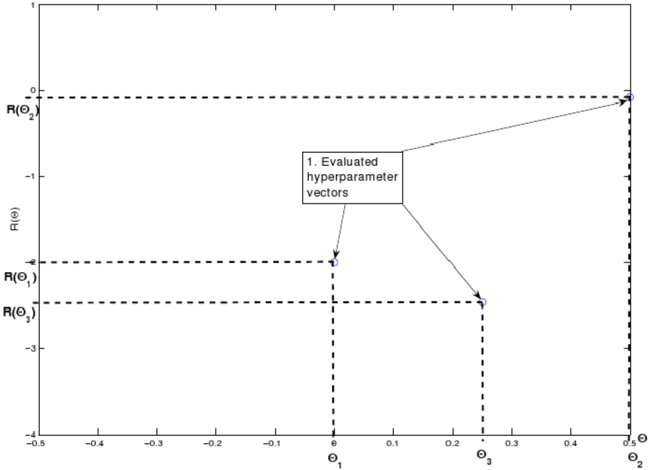
Rationale

- Start with some very coarse sampling DOE
- Evaluate performance using few runs per set
- Build a model of the performance landscape using *Gaussian Processes* aka Kriging
- Select best points based on *Expected Improvement* according to current model Monte-Carlo sampling
- Compute actual performance of best estimates using same number of runs as current best
- Increase # runs of best if unchanged

Gaussian Processes in one slide

An optimization algorithm for expensive functions

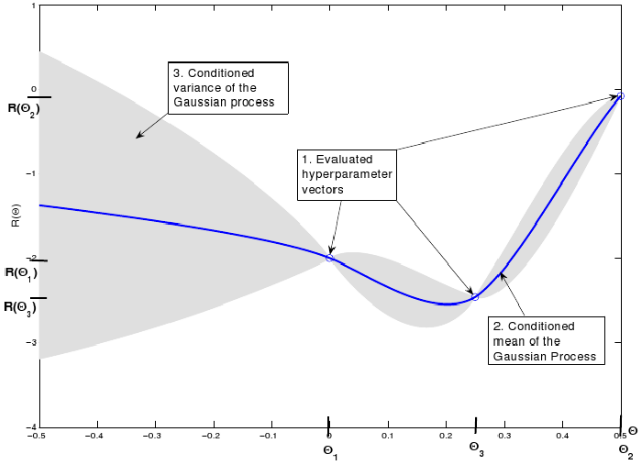
D.R. Jones, Schonlau, & Welch, 98



Gaussian Processes in one slide

An optimization algorithm for expensive functions

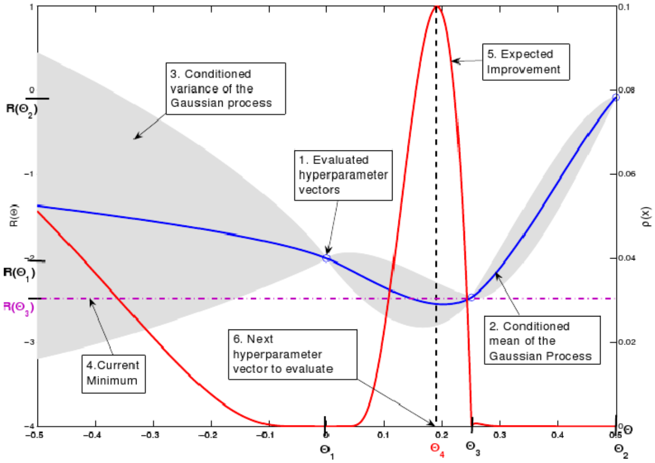
D.R. Jones, Schonlau, & Welch, 98



Gaussian Processes in one slide

An optimization algorithm for expensive functions

D.R. Jones, Schonlau, & Welch, 98



SPO: Discussion

Pros

- Similar ideas as racing,
- but allows to *refine initial sampling* a true optimization algorithm
- Compatible with a *fixed budget* scenario racing is not
- Authors also report gains up to 90%

Cons

- Works best with . . . some tuning

Take home messages

What is the performance criterion

- Cost function
- Account for class imbalance
- Account for data correlations

Assessing a result

- Compute confidence intervals
- Consider baselines
- Use a validation set

If the result looks too good, beware