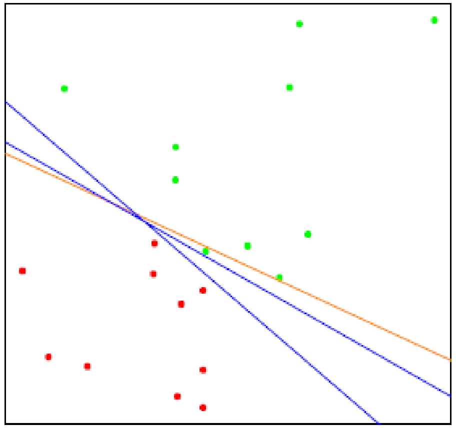


Parallel & Scalable Data Analysis

Introduction to Machine Learning Algorithms

Dr. – Ing. Morris Riedel

Adjunct Associated Professor

School of Engineering and Natural Sciences, University of Iceland

Research Group Leader, Juelich Supercomputing Centre, Germany

LECTURE 6

Validation and Parallelization Benefits

November 24th, 2017

Ghent, Belgium

UNIVERSITY OF ICELAND

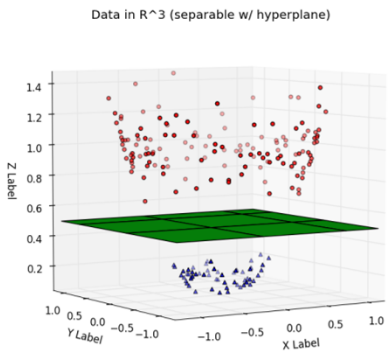
SCHOOL OF ENGINEERING AND NATURAL SCIENCES

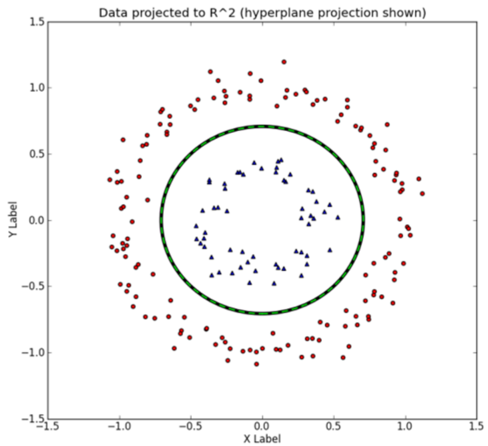
FACULTY OF INDUSTRIAL ENGINEERING, MECHANICAL ENGINEERING AND COMPUTER SCIENCE

JÜLICH FORSCHUNGSZENTRUM

Review of Lecture 5

- Non-linear Transformations
  - Use of a mapping function Φ
  - Hyperplane in higher dimensional space possible
  - Mapping back corresponds to non-linear decision boundary in initial input or x space





- Full Support Vector Machine
  - Full = use of non-linear kernel
  - Take advantage of mapping into a higher-level/infinite space
  - Apply 'kernel trick'
  - Kernels quantify similiarity
  - Different trusted kernels available (RBF, polynomial, etc.)

$$\sum \alpha_i y_i \boxed{\mathbf{x}_i \cdot \mathbf{u}_i} + b \geq 0 \quad \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{u}_i)$$

(dual since primal wi and b removed)

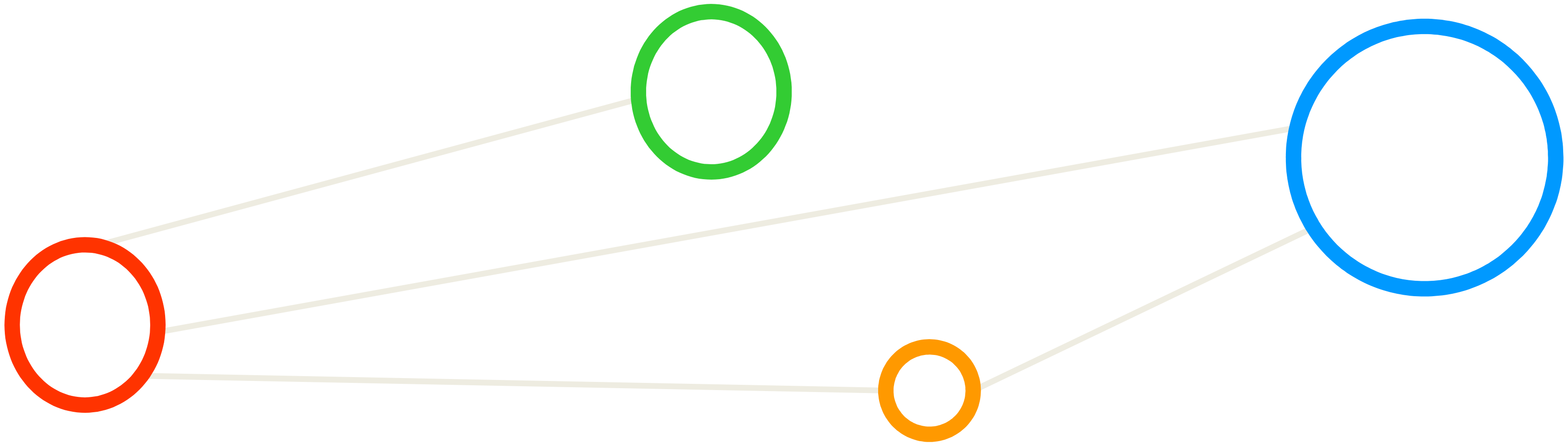
$$\mathcal{L} = \sum \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j \boxed{\mathbf{x}_i \cdot \mathbf{x}_j}$$

(trusted Kernel avoids to know Phi)

$$\Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j)$$

$$\boxed{K(\mathbf{x}_i, \mathbf{x}_j)} = \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j)$$

Outline



Outline of the Course

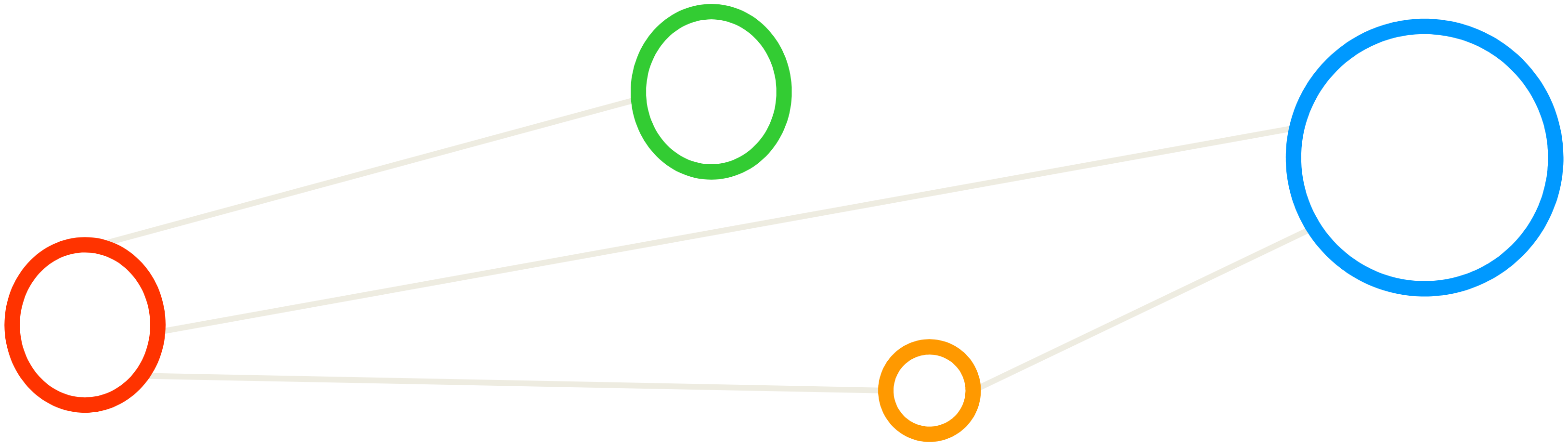
1. Machine Learning Fundamentals
2. Supervised Classification
3. Support Vector Machines
4. Applications and Serial Computing Limits
5. Kernel Methods
6. Applications and Parallel Computing Benefits

Outline

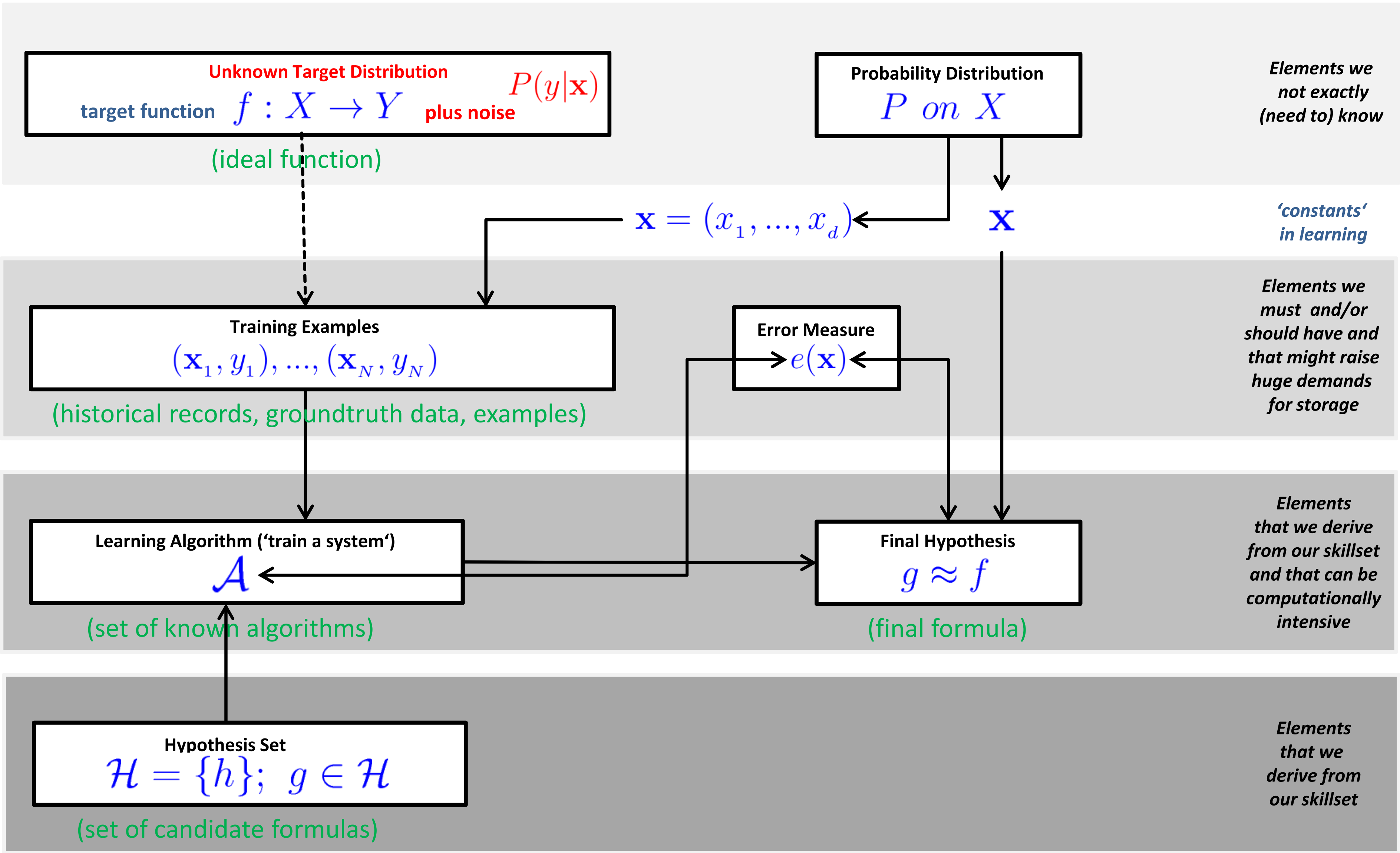
- Validation
  - Validation Set & Validation Error
  - Validation for Model Selection
  - N-Fold Cross-Validation Technique
  - Applying Validation of SVMs to Datasets
  - Experiencing Linear & Serial Limits

- Parallelization Benefits
  - Regularization Parameter Revisited
  - Possibility to work with large datasets
  - Parallelization Impact in Cross-Validation
  - Parallelization Summary & Acknowledgements
  - Complex Applications & Data Contamination

Validation



Mathematical Building Blocks – Revisited



Initial Terminologies – Reviewed w.r.t. Model Decisions

- Target Function $f : X \to Y$ → Target Distribution $f : X \to Y$
  - Ideal 'function' that 'explains' the data we want to learn   plus noise $P(y|\mathbf{x})$
- Labelled Dataset (samples)
  - 'in-sample' data given to us $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)$
- Dataset Part One: Training set (training set is used to make some decisions for model…)
  - Used for training a machine learning algorithms
  - Result after using a training set: a trained system
- Dataset Part Two: Test set (testing set has not been used to make any decisions for model…)
  - Used for testing whether the trained system might work well
  - Result after using a test set: accuracy of the trained model
- Learning vs. Memorizing
  - The goal is to create a system that works well 'out of sample' (future data)

(Another set of data is needed not used in training but that is used for model selection & 'validate decisions')

Training and Testing – Reviewed w.r.t. Model Decisions

(cf. Lecture 1)

- Mathematical notations
  - Testing follows: (hypothesis clear) $\Pr\left[\,|\,E_{in}(g) - E_{out}(g)\,| > \epsilon\,\right] <= 2 \quad e^{-2\epsilon^2 N}$
  - Training follows: (hypothesis search) $\Pr\left[\,|\,E_{in}(g) - E_{out}(g)\,| > \epsilon\,\right] <= 2Me^{-2\epsilon^2 N}$

(e.g. student exam training on examples to get E_{in} ‚down‘, then test via exam)

- Practice on ‘training examples‘
  - Create two disjoint datasets
  - One used for training only (aka training set)
  - Another used for testing only (aka test set)

Training Examples

$(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)$

(historical records, groundtruth data, examples)

- Training & Testing
  - Different phases in the learning process

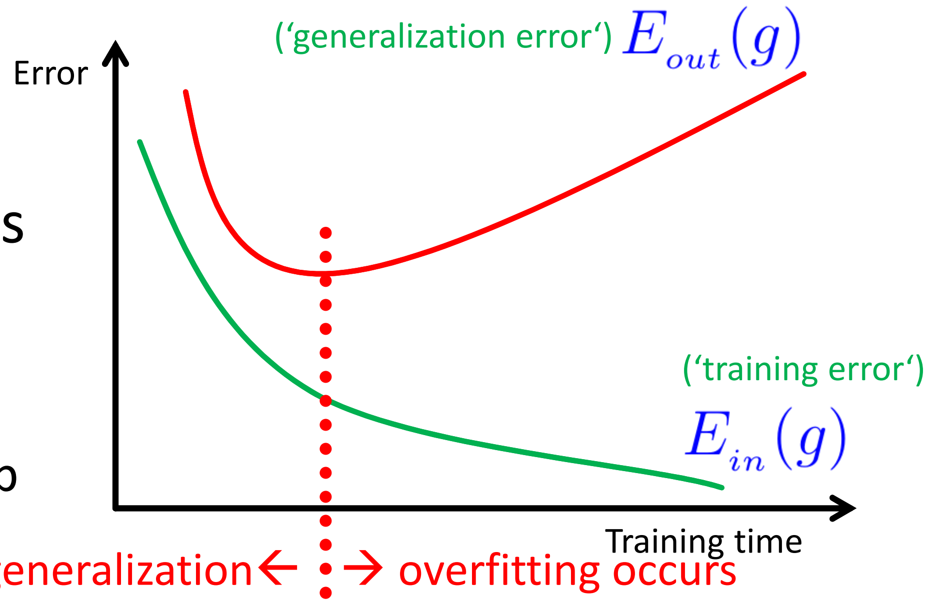
(Another phase in the creation of the whole model is needed where we take ‘validated decisions about the model‘)

Problem of Overfitting – Clarifying Terms

- **A good model must have low training error (E_{in}) and low generalization error (E_{out})**
- **Model overfitting is if a model fits the data too well (E_{in}) with a poorer generalization error (E_{out}) than another model with a higher training error (E_{in})**

[1] Introduction to Data Mining

- Overfitting & Errors
  - $E_{in}(g)$ goes down
  - $E_{out}(g)$ goes up
- 'Bad generalization area' ends
  - Good to reduce $E_{in}(g)$
- 'Overfitting area' starts
  - Reducing $E_{in}(g)$ does not help
  - Reason 'fitting the noise'



- **The two general approaches to prevent overfitting are (1) regularization and <u>(2) validation</u>**

(Decisions about the model are related to the problem of overfitting – need another method to 'select model well')

Problem of Overfitting – Impacts on Learning Revisited

- **The higher the degree of the polynomial (cf. model complexity), the more degrees of freedom are existing and thus the more capacity exists to overfit the training data**

- Understanding deterministic noise & target complexity
  - Increasing target complexity increases deterministic noise (at some level)
  - Increasing the number of data N decreases the deterministic noise
- Finite N case: $\mathcal{H}$ tries to fit the noise
  - Fitting the noise straightforward (e.g. with linear regression)
  - Stochastic (in data) and deterministic (simple model) noise will be part of it
- Two 'solution methods' for avoiding overfitting
  - Regularization: 'Putting the brakes in learning', e.g. early stopping (more theoretical, hence 'theory of regularization')
  - Validation: 'Checking the bottom line', e.g. other hints for out-of-sample (more practical, methods on data that provides 'hints')

(Decisions about the model are related to the model complexity – need another method to 'select model well')

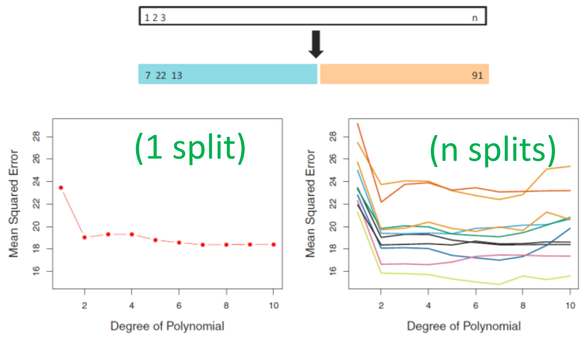
Validation & Model Selection – Terminology

- **The ‘Validation technique‘ should be used in all machine learning or data mining approaches**
- **Model assessment is the process of evaluating a models performance**
- **Model selection is the process of selecting the proper level of flexibility for a model**

modified from [2] ‘An Introduction to Statistical Learning’

- ‘Training error‘
  - Calculated when learning from data (i.e. dedicated training set)
- ‘Test error’
  - Average error resulting from using the model with ‘new/unseen data‘
  - ‘new/unseen data‘ was not used in training (i.e. dedicated test set)
  - In many practical situations, a dedicated test set is not really available
- ‘Validation Set‘
  - Split data into training & validation set
- ‘Variance‘ & ‘Variability‘
  - Result in different random splits (right)

(split creates a two subsets of comparable size)



Validation Technique – Formalization & Goal

- **Validation is a very important technique to estimate the out-of-sample performance of a model**
- **Main utility of regularization & validation is to control or avoid overfitting via model selection**

- Regularization & Validation
  - Approach: introduce a 'overfit penalty' that relates to model complexity
  - Problem: Not accurate values: 'better smooth functions'

(regularization uses a term that captures the overfit penalty)

$$E_{out}(h) = E_{in}(h) + \textbf{overfit penalty}$$

(minimize both to be better proxy for E_{out})

(validation estimates this quantity)

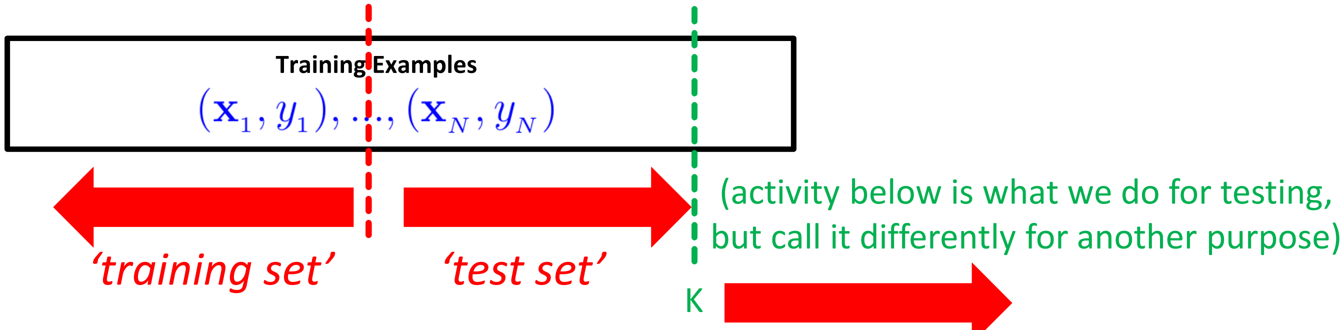
(regularization estimates this quantity)

- Validation

  (measuring E_{out} is not possible as this is an unknown quantity, another quantity is needed that is measurable that at least estimates it)

  - Goal 'estimate the out-of-sample error' (establish a quantity known as validation error)
  - Distinct activity from training and testing (testing also tries to estimate the E_{out})

Validation Technique – Pick one point & Estimate E_{out}



- Understanding 'estimate' E_{out}
  - On one out-of-sample point $(\mathbf{x}, y)$ the error is $e(h(\mathbf{x}), y)$
  - E.g. use squared error: $e(h(\mathbf{x}), f(\mathbf{x})) = (h(\mathbf{x}) - f(\mathbf{x}))^2$
    $$e(h(\mathbf{x}), y) = (h(\mathbf{x}) - y)^2$$
  - Use this quantity as estimate for E_{out} (poor estimate)
  - Term 'expected value' to formalize (probability theory)

(Taking into account the theory of Lecture 1 with probability distribution on X etc.)

Probability Distribution
P on X

$\mathbf{x} = (x_1, \ldots, x_d)$

(aka 'random variable')

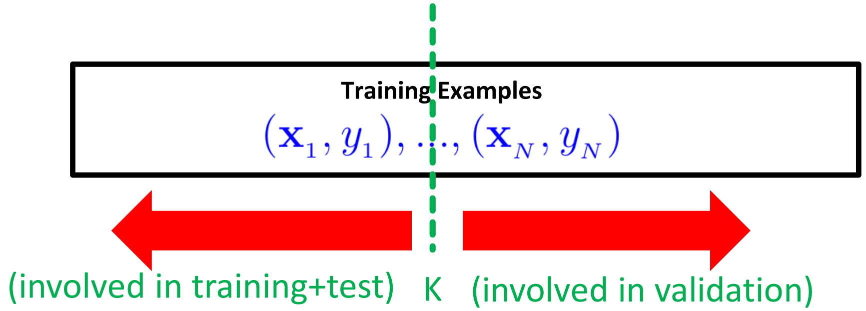
$$\mathbb{E}[e(h(\mathbf{x}), y)] = E_{out}(h)$$ (aka the long-run average value of repetitions of the experiment)

(one point as unbiased estimate of E_{out} that can have a high variance leads to bad generalization)

Validation Technique – Validation Set

- **Validation set consists of data that has been not used in training to estimate true out-of-sample**
- **Rule of thumb from practice is to take 20% (1/5) for validation of the learning model**

- Solution for high variance in expected values $\mathbb{E}[e(h(\mathbf{x}), y)] = E_{out}(h)$
  - Take a 'whole set' instead of just one point $(\mathbf{x}, y)$ for validation



(we need points not used in training
to estimate the out-of-sample performance)

(we do the same approach with the
testing set, but here different purpose)

  - Idea: K data points for validation

$(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_K, y_K)$ (validation set)

$$E_{val}(h) = \frac{1}{K}\sum_{k=1}^{K} e(h(\mathbf{x})_k, y_k)$$ (validation error)

  - Expected value to 'measure' the out-of-sample error

(expected values averaged over set)

$$\mathbb{E}[E_{val}(h)] = \frac{1}{K}\sum_{k=1}^{K} \mathbb{E}[e(h(\mathbf{x})_k, y_k)] = E_{out}$$

  - 'Reliable estimate' if K is large

(on rarely used validation set,
otherwise data gets contaminated)

(this gives a much better (lower) variance than on a single point given K is large)

Validation Technique – Model Selection Process

- **Model selection is choosing (a) different types of models or (b) parameter values inside models**
- **Model selection takes advantage of the validation error in order to decide → 'pick the best'**

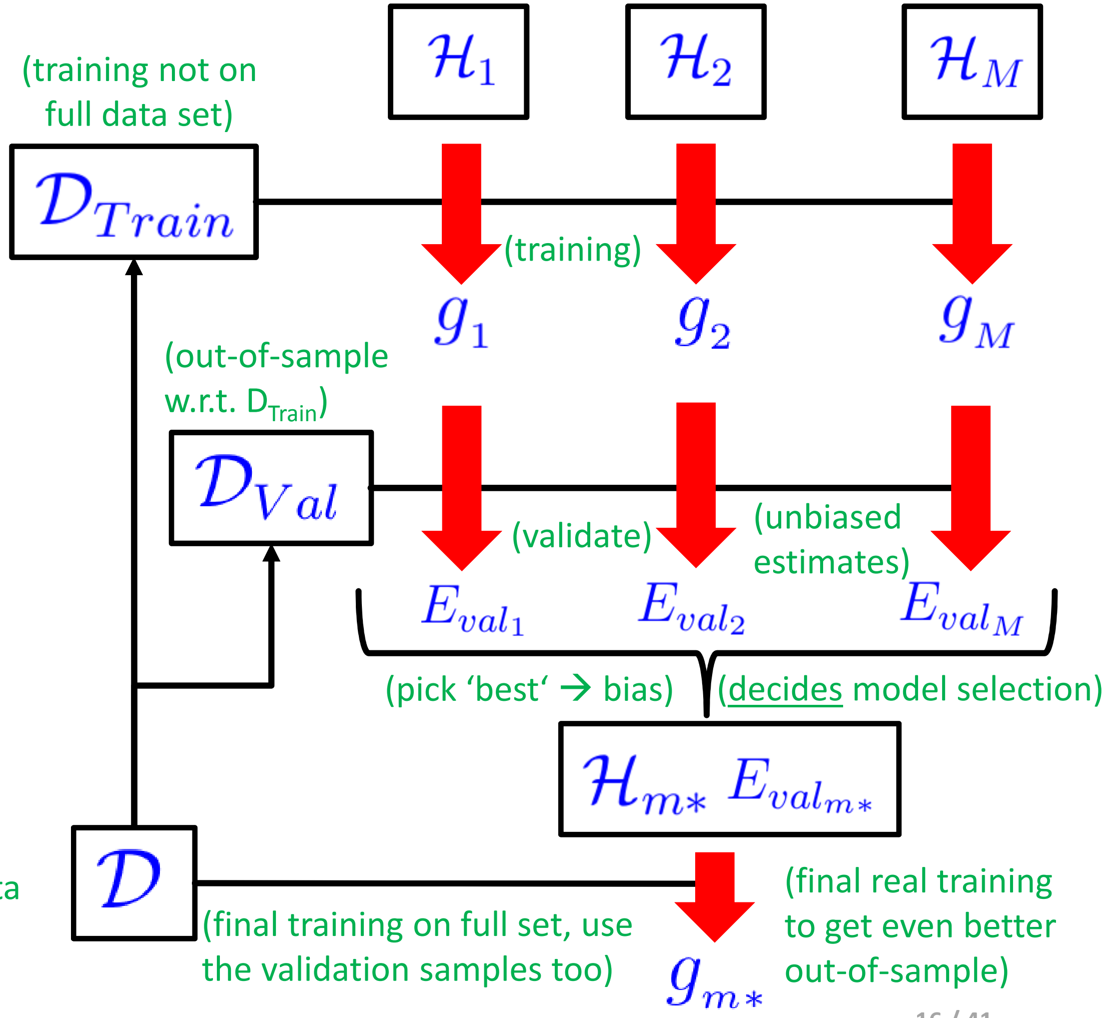
Hypothesis Set
$\mathcal{H} = \{h\}; \ g \in \mathcal{H}$

(set of candidate formulas across models)

- M models
  (cf. Lecture 1)
  - Use validation error to perform select decisions
- Careful consideration:
  - 'Picked means decided' hypothesis has already bias (→ contamination)
  - Using $\mathcal{D}_{Val}$ M times

Final Hypothesis
$g_{m*} \approx f$

(test this on unseen data good, but depends on availability in practice)



Validation Technique – Cross-Validation – Trick

- **Cross-validation the technique of choice in practical situations to perform model selection**
- **Different techniques exist for cross-validation such as leave-one-out, leave-more-out**

(every time a data point is used for validation it is taken away from training)

- Goal (validation data not given on top of training data)
  - Target issue 'choosing K' out of N
  - Issue: K needs to be small & large
  - (cf. Lecture 1 feasibility of learning)

Training Examples

$(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)$

(involved in training) K (involved in validation)

(conflicting requirements on K)

(chain of reasoning so far)

$$E_{out}(g) \approx E_{out}(g^-) \approx E_{val}(g^-)$$

(idea: is there a solution over time?)

(small K for large N – K training delivering good out-of-sample performance)

(large K for validation delivering a good estimate for out-of-sample performance)

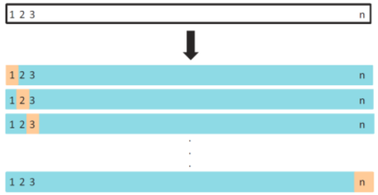
- Apply trick: repeat the number of trainings on different subsets
  - Train multiple times using e.g. leave-one-out or leave-more-out (practice)

- **Cross-validation 'trick' achieves to use N points for training and N points for validation (big gain!)**

Validation Technique – Cross-Validation – Leave-one-out

- Simplest form of cross-validation
  - Use N – 1 data points for training
  - Potential issue: only 1 data point for validation (bad estimate for E_{out})
  - Creates a very ‘small validation‘ set, but very ‘large training‘ set

(split data NOT just in two subsets of comparable size)



Source: [30]

$$\mathcal{D}_n = (\mathbf{x}_1, y_1), (\mathbf{x}_{n-1}, y_{n-1}), \boxed{(\mathbf{x}_n, y_n),} (\mathbf{x}_{n+1}, y_{n+1}), \ldots, (\mathbf{x}_N, y_N)$$

(reduced dataset, but as training set very close to N missing just one)

(one data point left out for validation)

(not trained on full set & depends on point left out)

- Final hypothesis to be ‘selected‘ after training with $\mathcal{D}_n$ is g_n^-

(check error on the point ‘left out‘ → out-of-sample)

$$e_n = E_{val}(g_n^-) = e(g_n^-(\mathbf{x}_n), y_n)$$

(the hypothesis was trained not involving this point)

(validate hypothesis on that data point taken out)

(one point in validation set brings bad estimate for E_{out})

- Apply ‘resampling‘ trick:
  Repeat for different n with $\mathcal{D}_n$
  (obtains different hypothesis g_1^-, g_2^-,…., but all have in common to be obtained by being trained on N – 1 points)

$$E_{CV} = \frac{1}{N} \sum_{n=1}^{N} e_n$$

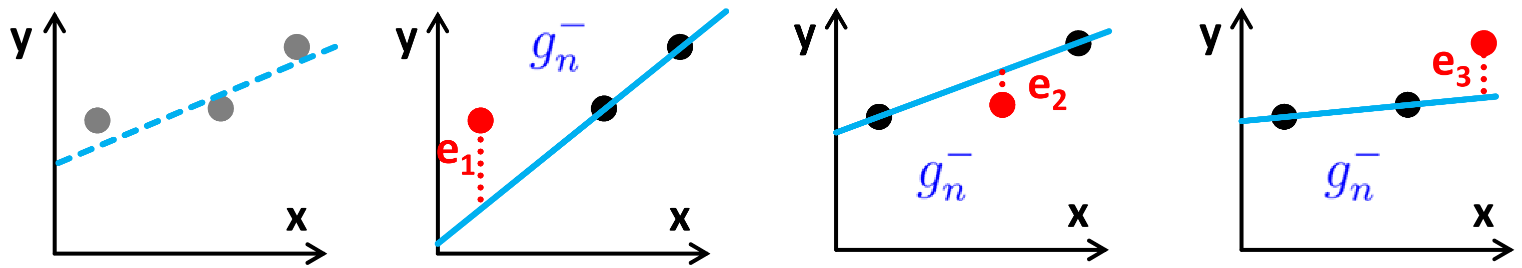
(works well with increasing N)

(cross-validation error on validation set with N points)

Validation Technique – Cross-Validation Error Example

- Example: Create a linear model
  - Assuming there is noise in the target function
  - Cross-validation: evaluate out-of-sample error to choose a model (later)

(red points are validation sets in each run) (black points are training sets in each run)



(the full dataset = 3 points) (n = left point out) (n = middle point out) (n = right point out)

$$E_{CV} = \frac{1}{N}\sum_{n=1}^{N} e_n$$

(simply compute average of all errors, e.g. using squared distance)

$$E_{CV} = \frac{1}{3}(e_1 + e_2 + e_3)$$

(cross-validation error as indication of how well 'the linear model' fits the data → out-of-sample)

(impact on N = small (3) is enormous, but if N = large average works very well)

- **Cross-validation is a 'resampling method' that obtains more information than 'fitting model once'**
- **Compute cross-validation error is possible (via 'in-sample') & a systematic way for model selection**

Validation Technique – Cross-Validation & Model Selection

- Model selection: Perform a ‘decision‘
  - Cross-validation: evaluate out-of-sample error to choose a model (avoiding e.g. heuristics)
  - Example: <u>Decide</u> whether Linear Model or Constant Model is better

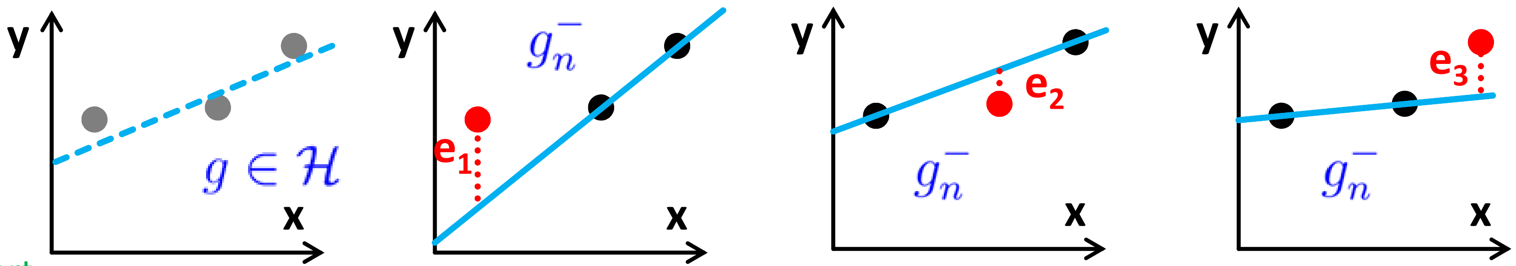
Hypothesis Set

$$\mathcal{H} = \{h\};\ \ g \in \mathcal{H}$$

(set of candidate formulas)

$\mathcal{H} = \{h\};\ g \in \mathcal{H}$

(LINEAR MODEL)



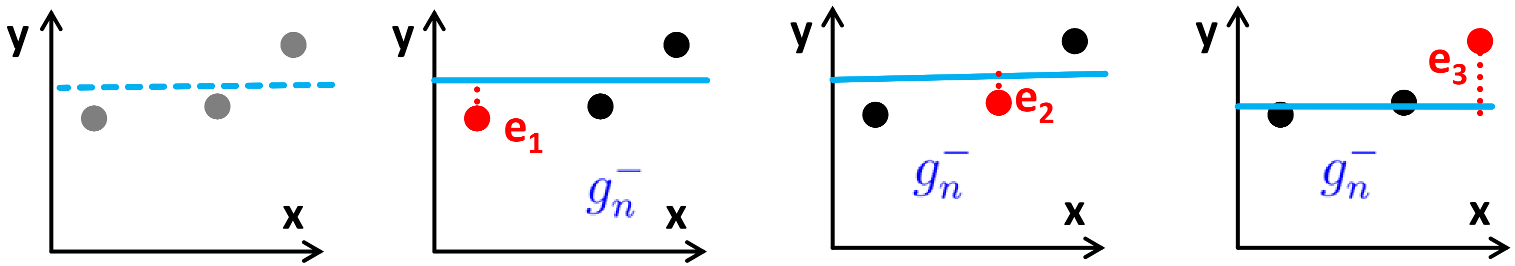
(decision support required, think N huge)

(use the obtained cross-validation error for both models as a ‘grade for each model‘)

(use this ‘grade‘ and compare it for model selection, e.g. constant model wins, less squared error)

$\mathcal{H} = \{h\};\ g \in \mathcal{H}$

(CONSTANT MODEL)



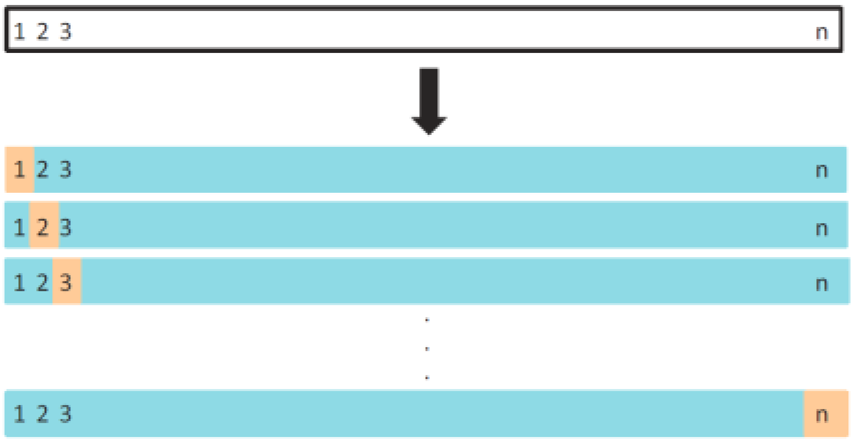
- **Main utility of cross-validation is model selection supporting a decision to choose a model**

Validation Technique – Cross-Validation – K-Fold Approach

(split creates a two subsets of comparable size)

(random strategy, works not particularly well)

Leave-One-Out Cross-Validation (LOOCV) Example

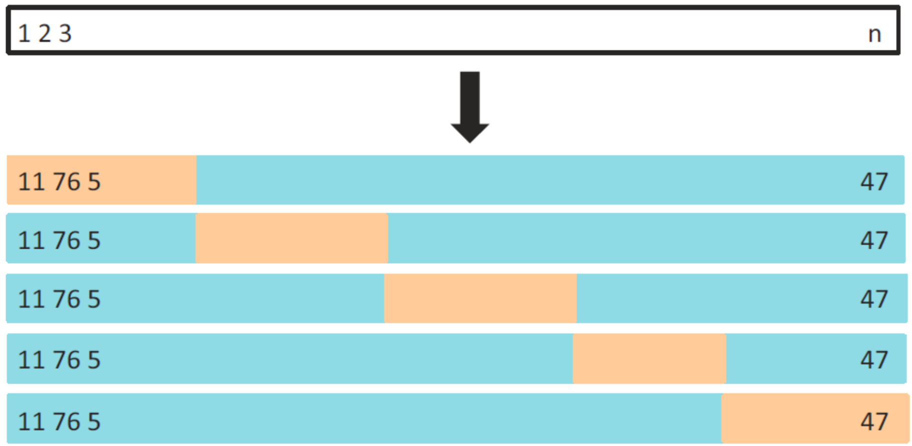


(picking strategy, works well but possible long computing)

pick one point for validation
resulting in possible large
sets when number of points are high

[2] 'An Introduction to Statistical Learning'

5-fold Cross-Validation Example



(picking strategy, works well and reduces computing)

A set of data points is randomly split into
k non-overlapping groups ('k-folds')
of approximately equal size

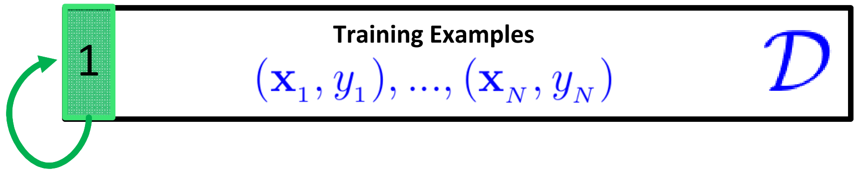
Recommendation in Practice

Validation Technique – Cross-Validation – Leave-more-out

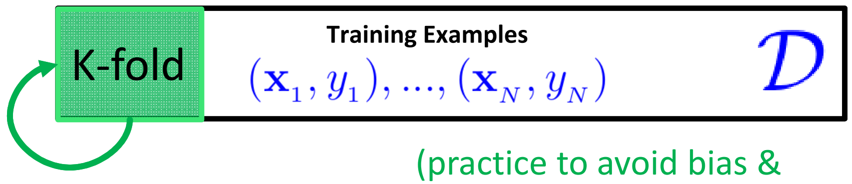
- **10-fold cross validation is mostly applied in practical problems by setting K = N/10 for real data**
- **Having N/K training sessions on N – K points each leads to long runtimes (→ use parallelization)**

- Leave-one-out
  - N training sessions on N – 1 points each time
- Leave-more-out
  - Break data into number of folds
  - N/K training sessions on N – K points each time (fewer training sessions than above)
  - Example: ‘10-fold cross-valdation‘ with K = N/10 multiple times (N/K)
    (use 1/10 for validation, use 9/10 for training, then another 1/10 … N/K times)

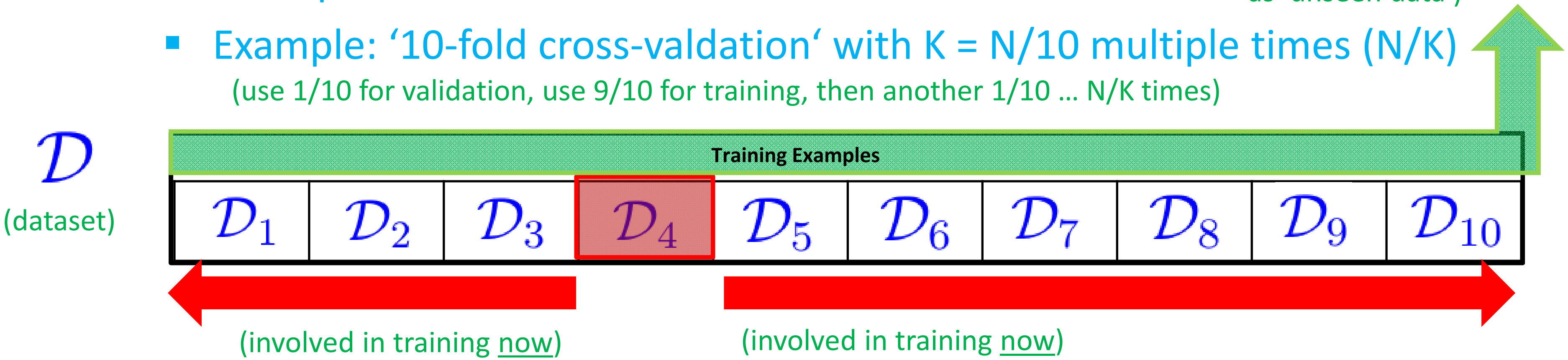
(leave 1 point out at each run → many runs)



(generalization to leave k points out at each run)



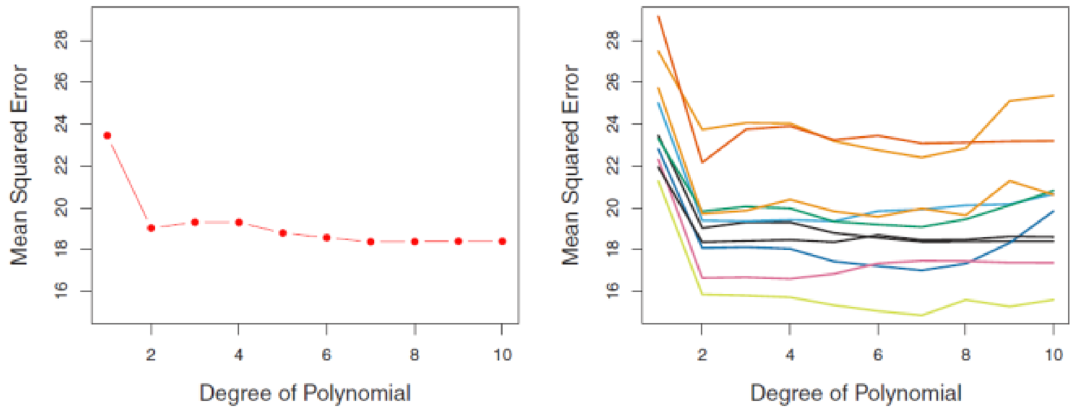
(practice to avoid bias & contamination: some rest for test as ‘unseen data‘)



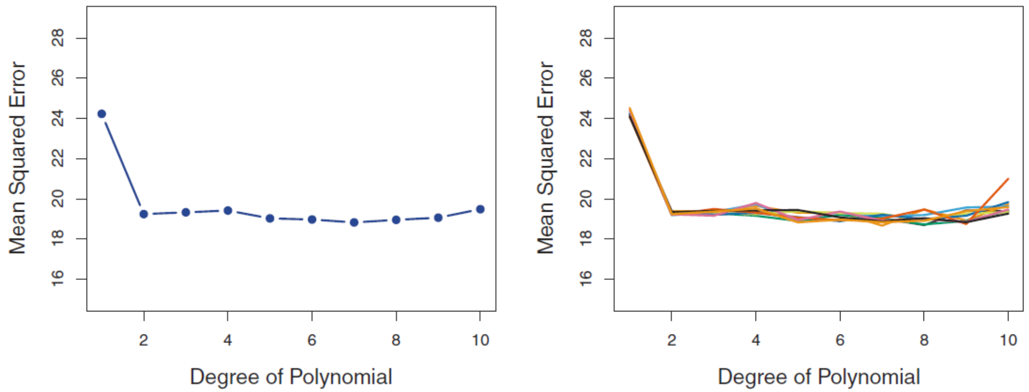
(now is the current example run)

Validation Technique – 10 fold Cross-Validation Example

- 10 times resampling
  - Validation set with 10 x 2 comparable sizes
  - ‘Random splits‘
  - High variability/variance



- 10-fold cross validation
  - Validation set with 10-fold x 2 strategy
  - No ‘random splits‘
  - Lower variability/variance



modified from [2] ‘An Introduction to Statistical Learning’

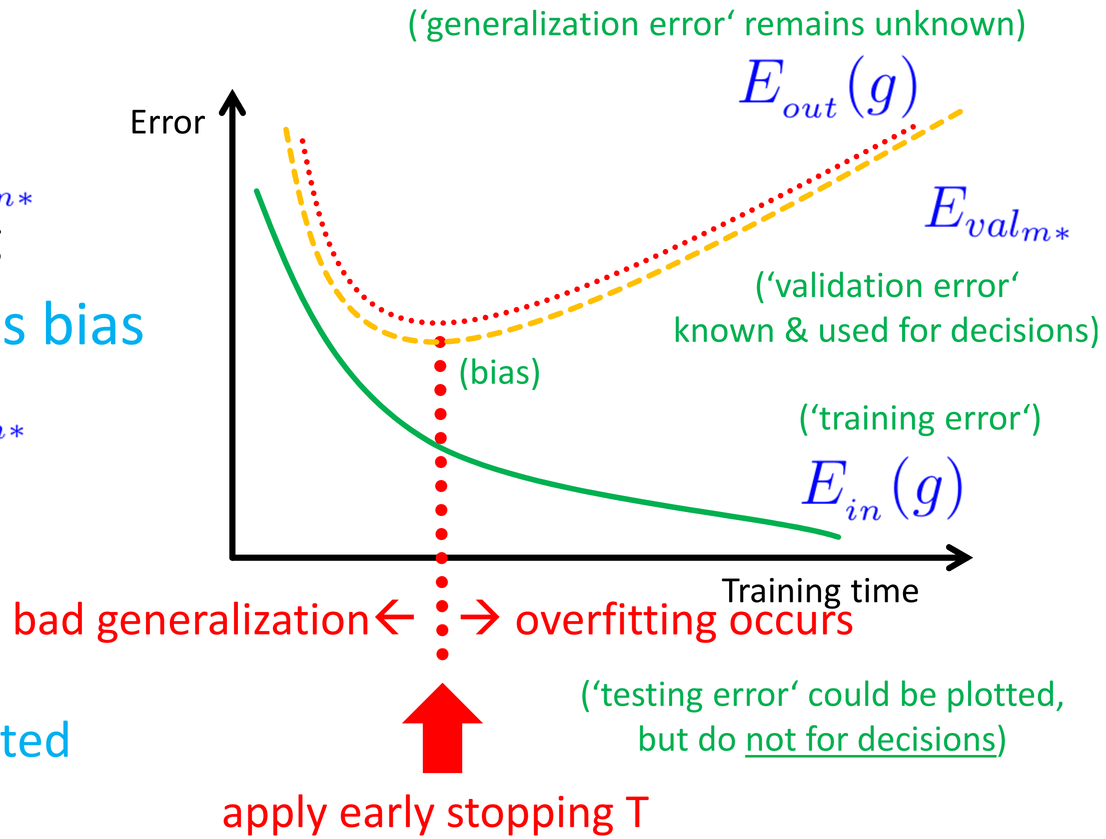
Model Performance – Validation Enables Early Stopping (1)

- Problem of overfitting
  - Issue is that $E_{out}(g)$ is unknown to perform 'early stopping'
  - Apply validation = 'perform decision to make a choice' based on $\mathcal{D}_{Val}$

$\boxed{\mathcal{D}_{Val}}$ (out-of-sample w.r.t. D_{Train})

  - Use validation error $E_{val_{m*}}$ to perform early stopping

- Validation decision brings bias
  - When the estimate $E_{val_{m*}}$ of $E_{out}(g)$ affects the learning process decision
  - Optimistic bias impact brings accuracy higher than in reality (cf. Associated use case with testset)



Model Performance – Validation Enables Early Stopping (2)

- 'Bias' reviewed as 'data contamination'
  - Training set is biased and contaminated (i.e. 'used for train model change')
  - Test set is unbiased and clean (i.e. 'waiting to be used in the final end')
  - Validation set has an optimistic bias (i.s. 'use in model selection decisions')
    ('slightly contaminated since only few choices')

$$\underbrace{E_{val_1} \quad E_{val_2} \quad E_{val_M}}$$

(pick 'best' → bias) (decides model selection)

$$\boxed{\mathcal{H}_{m*} \; E_{val_{m*}}}$$

(reasoning of bias relates to the probability and estimated value of validation errors since 'one is picked' as the minimum of all)

(e is a min function of E_{val1}, E_{val2}, etc.)

$$\mathbb{E}[e(h(\mathbf{x}), y)] = E_{out}(h)$$

$$\mathbb{E}[E_{val}(h)] = \frac{1}{K}\sum_{k=1}^{K}\mathbb{E}[e(h(\mathbf{x})_k, y_k)] = E_{out}$$

(aka the long-run average value of repetitions of the experiment)

(cf. picking the 'best time' in early stopping, also brings optimistic bias since minimum on model creation)

- **Optimistic bias means that there is 'a belief' that the error is smaller as it is actually going to be**
- **Optimistic bias is minor and thus accepted in learning, but perform reporting with unbiased testset**
- **Important in validation is that the validation set stays only 'slightly contaminated' (few choices)**
- **In practice several validation sets can be used for n parameter choices to keep reliable estimate**

piSVM / LibSVM – svm-train Parameters Revisited

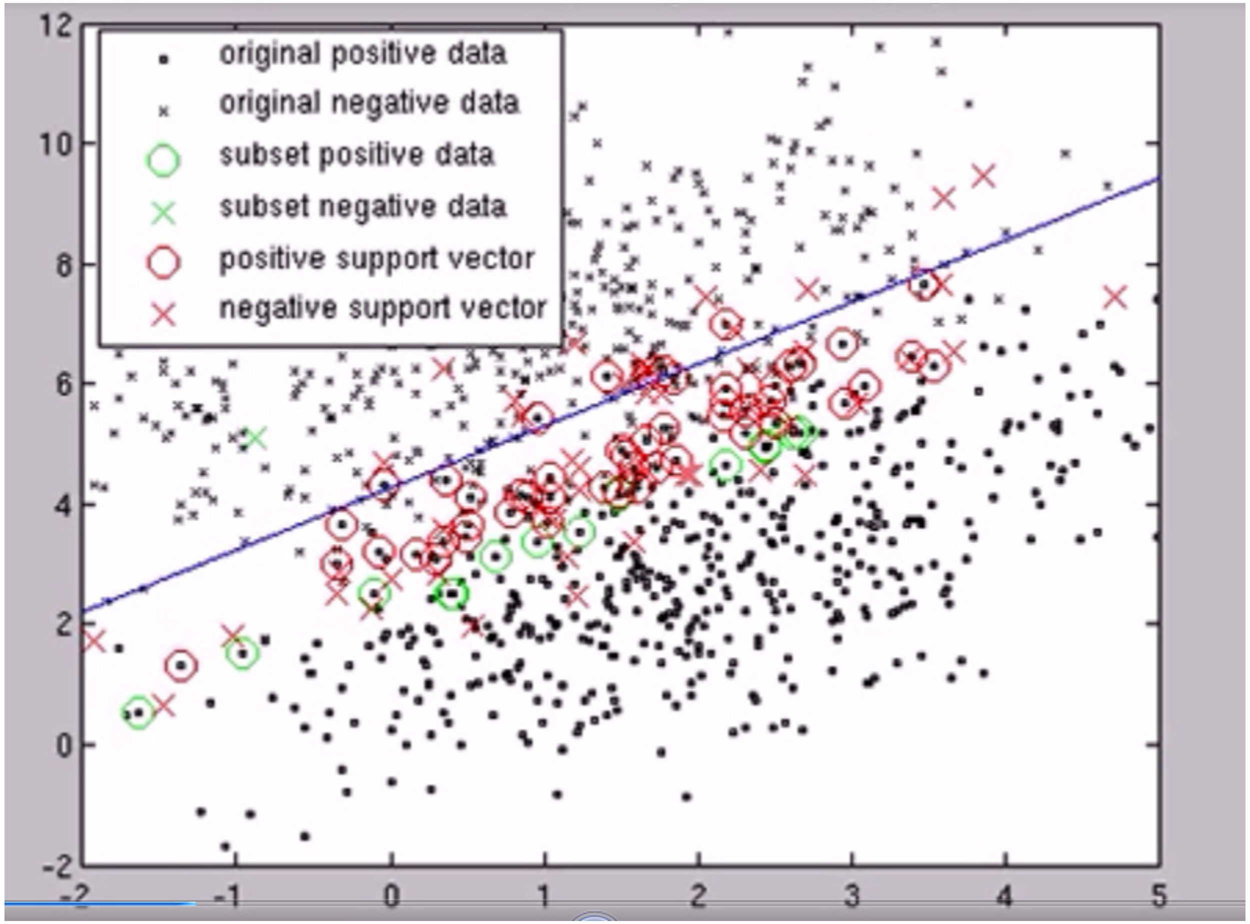
- Important parameters

```
-bash-4.2$ ./svm-train
Usage: svm-train [options] training_set_file [model_file]
options:
-s svm_type : set type of SVM (default 0)
        0 -- C-SVC              (multi-class classification)
        1 -- nu-SVC             (multi-class classification)
        2 -- one-class SVM
        3 -- epsilon-SVR        (regression)
        4 -- nu-SVR             (regression)
-t kernel_type : set type of kernel function (default 2)
        0 -- linear: u'*v
        1 -- polynomial: (gamma*u'*v + coef0)^degree
        2 -- radial basis function: exp(-gamma*|u-v|^2)
        3 -- sigmoid: tanh(gamma*u'*v + coef0)
        4 -- precomputed kernel (kernel values in training_set_file)
-d degree : set degree in kernel function (default 3)
-g gamma : set gamma in kernel function (default 1/num_features)
-r coef0 : set coef0 in kernel function (default 0)
-c cost : set the parameter C of C-SVC, epsilon-SVR, and nu-SVR (default 1)
-n nu : set the parameter nu of nu-SVC, one-class SVM, and nu-SVR (default 0.5)
-p epsilon : set the epsilon in loss function of epsilon-SVR (default 0.1)
-m cachesize : set cache memory size in MB (default 100)
-e epsilon : set tolerance of termination criterion (default 0.001)
-h shrinking : whether to use the shrinking heuristics, 0 or 1 (default 1)
-b probability_estimates : whether to train a SVC or SVR model for probability estimates, 0 or 1 (default 0)
-wi weight : set the parameter C of class i to weight*C, for C-SVC (default 1)
-v n: n-fold cross validation mode
-q : quiet mode (no outputs)
```

(creates not a model, but gives an estimate for unseen data)

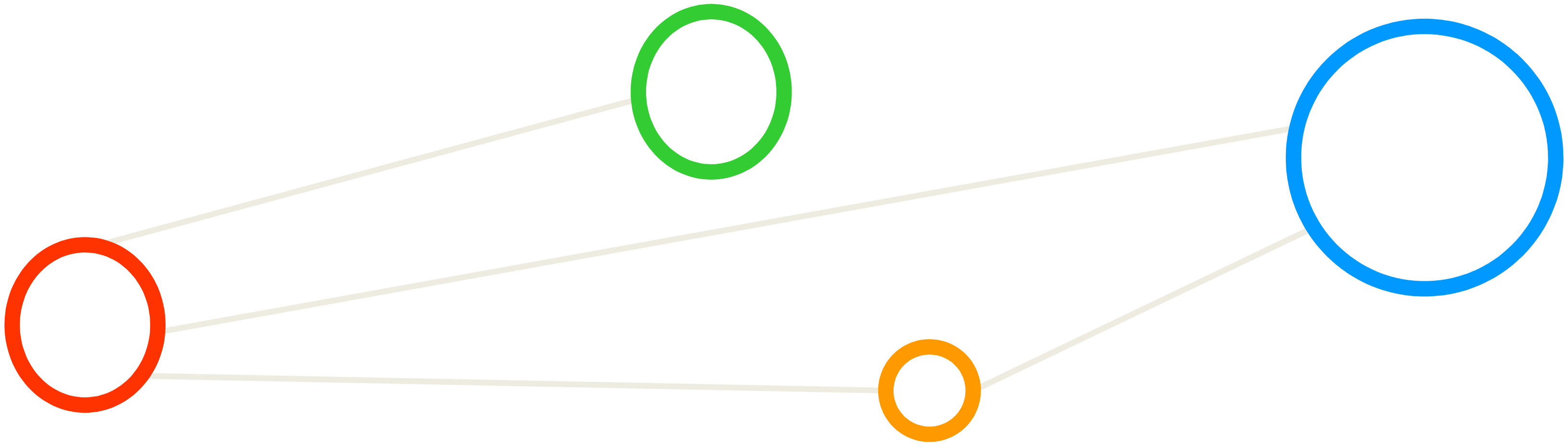
[3] LibSVM Webpage

[Video] Training Process of Support Vector Machines



Exercises

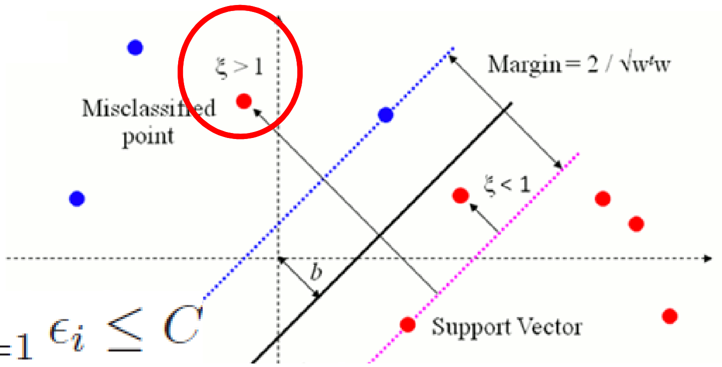
Parallelization Benefits



Regularization Revisited & Rules of Thumb for C

- C = 0 (too rectrictive, potentially bad for E_{out}) $\epsilon_1 = \ldots = \epsilon_n = 0$
  - No budget/costs for violations: comparable to maximal margin classifier
  - Further constraint: only works in linearly seperable cases (less in practice)
- C > 0 (flexible option, better for E_{out})
  - No more than C data points can be on the wrong side of the hyperplane ('how much misclassifications allowed')
  - Reasoning: if an observation is on the wrong side then $\epsilon_i > 1$

$$\sum_{i=1}^{n} \epsilon_i \leq C$$



(rule of thumb)

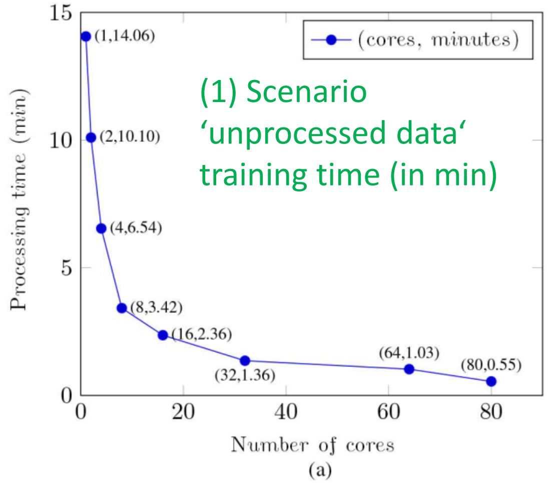
- **regularization parameter C (budget of errors) increase → margins will be wide and more tolerant of violations to the margin (<u>classifier fits data less</u>)**
- **regularization parameter C (budget of errors) descreases → margins will be narraw and less tolerant of violations to the margin (<u>classifier highly fit data</u>)**

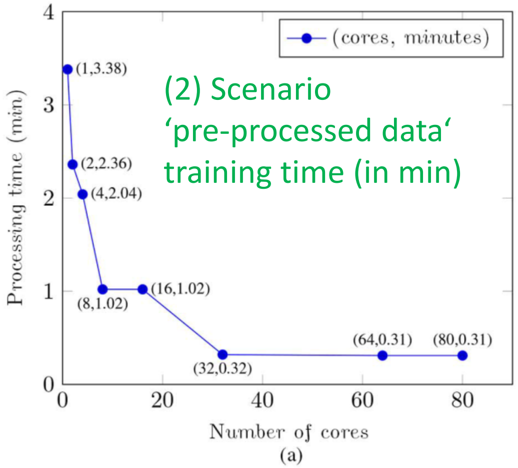
(differently handled in R library)

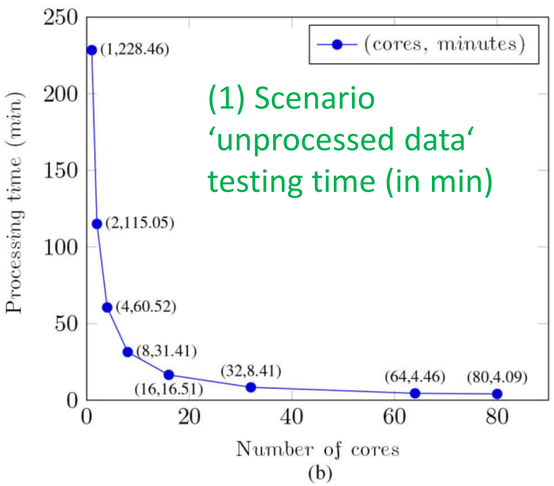
- **Determine the right C parameter for a model can be obtained using parallelization on a HPC system**

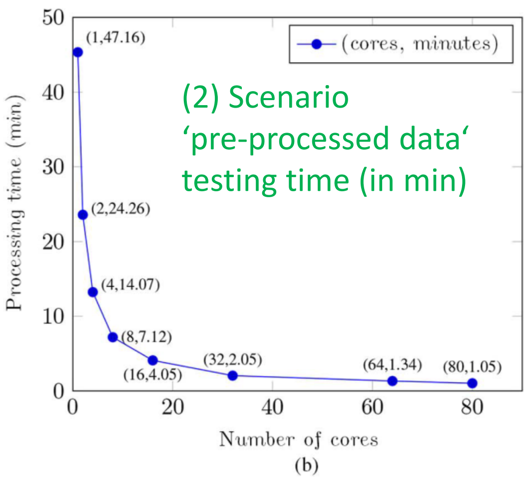
Parallelization Benefit: Lower-Time-To-Solution

- Major speed-ups; ~interactive (<1 min); same accuracy;









manual & serial activities (in min)

	kpca	esdap	nwfe	10x CSV	Training	Test	Total
(1) Scenario	0	0	0	4.47×10^3	10.45	71.08	4.55×10^3
(2) Scenario	5	15.38	1	529.55	1.37	23.25	575.55

'big data' is not always better data

	(1) Scenario	(2) Scenario
Number of features	200	30
Overall Accuracy (%)	40.68	77.96

(cf. Importance of feature engineering above)

[4] G. Cavallaro, M. Riedel, J.A. Benediktsson et al., Journal of Selected Topics in Applied Earth Observation and Remote Sensing, 2015

B2SHARE Store and Share Research Data

Parallelization Benefit: Parallel 10-Fold Cross-Validation

- Example: 2 Parameters, 10-fold cross-validation
  - 2 x benefits of parallelization possible in a so-called ‘gridsearch‘
  - (1) Compute parallel; (2) Do all cross-validation runs in parallel (all cells)
  - Evaluation between Matlab (aka ‘serial laptop‘) & parallel (80 cores)

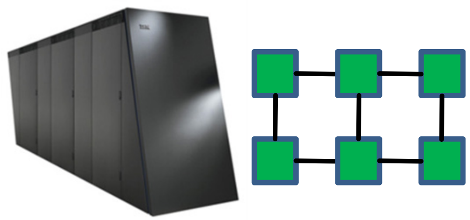
(2) Scenario ‘pre-processed data‘, 10xCV **serial**: accuracy (min)

γ/C	1	10	100	1000	10 000
2	48.90 (18.81)	65.01 (19.57)	73.21 (20.11)	75.55 (22.53)	74.42 (21.21)
4	57.53 (16.82)	70.74 (13.94)	75.94 (13.53)	76.04 (14.04)	74.06 (15.55)
8	64.18 (18.30)	74.45 (15.04)	**77.00** (14.41)	75.78 (14.65)	74.58 (14.92)
16	68.37 (23.21)	76.20 (21.88)	76.51 (20.69)	75.32 (19.60)	74.72 (19.66)
32	70.17 (34.45)	75.48 (34.76)	74.88 (34.05)	74.08 (34.03)	73.84 (38.78)

(1) First Result: best parameter set from 14.41 min to 1.02 min

(2) Second Result: all parameter sets from ~9 hours to ~35 min

[4] G. Cavallaro, M. Riedel, J.A. Benediktsson et al., Journal of Selected Topics in Applied Earth Observation and Remote Sensing, 2015



‘(1) each cell inherent parallel’

‘(2) all cells in parallel’

(2) Scenario ‘pre-processed data‘, 10xCV **parallel**: accuracy (min)

γ/C	1	10	100	1000	10 000
2	75.26 (1.02)	65.12 (1.03)	73.18 (1.33)	75.76 (2.35)	74.53 (4.40)
4	57.60 (1.03)	70.88 (1.02)	75.87 (1.03)	76.01 (1.33)	74.06 (2.35)
8	64.17 (1.02)	74.52 (1.03)	**77.02 (1.02)**	75.79 (1.04)	74.42 (1.34)
16	68.57 (1.33)	76.07 (1.33)	76.40 (1.34)	75.26 (1.05)	74.53 (1.34)
32	70.21 (1.33)	75.38 (1.34)	74.69 (1.34)	73.91 (1.47)	73.73 (1.33)

- **10-fold cross-validation achieves parallelization benefits (1) in each grid cell and (2) across all cells**

Parallelization Summary

- Parallelization benefits are enormous for complex problems
  - Enables feasibility to tackle extremely large datasets & high dimensions
  - Provides functionality for a high number of classes (e.g. #k SVMs)
  - Achieves a massive reduction in time → lower time-to-solution

(1) Scenario 'unprocessed data', 10xCV **serial**: accuracy (min)

γ/C	1	10	100	1000	10 000
2	27.30 (109.78)	34.59 (124.46)	39.05 (107.85)	37.38 (116.29)	37.20 (121.51)
4	29.24 (98.18)	37.75 (85.31)	38.91 (113.87)	38.36 (119.12)	38.36 (118.98)
8	31.31 (109.95)	**39.68** (118.28)	39.06 (112.99)	39.06 (190.72)	39.06 (872.27)
16	33.37 (126.14)	39.46 (171.11)	39.19 (206.66)	39.19 (181.82)	39.19 (146.98)
32	34.61 (179.04)	38.37 (202.30)	38.37 (231.10)	38.37 (240.36)	38.37 (278.02)

(2) Scenario 'pre-processed data', 10xCV **serial**: accuracy (min)

γ/C	1	10	100	1000	10 000
2	48.90 (18.81)	65.01 (19.57)	73.21 (20.11)	75.55 (22.53)	74.42 (21.21)
4	57.53 (16.82)	70.74 (13.94)	75.94 (13.53)	76.04 (14.04)	74.06 (15.55)
8	64.18 (18.30)	74.45 (15.04)	**77.00** (14.41)	75.78 (14.65)	74.58 (14.92)
16	68.37 (23.21)	76.20 (21.88)	76.51 (20.69)	75.32 (19.60)	74.72 (19.66)
32	70.17 (34.45)	75.48 (34.76)	74.88 (34.05)	74.08 (34.03)	73.84 (38.78)

(1) Scenario 'unprocessed data''10xCV **parallel**: accuracy (min)

γ/C	1	10	100	1000	10 000
2	27.26 (3.38)	34.49 (3.35)	39.16 (5.35)	37.56 (11.46)	37.57 (13.02)
4	29.12 (3.34)	37.58 (3.38)	38.91 (6.02)	38.43 (7.47)	38.43 (7.47)
8	31.24 (3.38)	**39.77 (4.09)**	39.14 (5.45)	39.14 (5.42)	39.14 (5.43)
16	33.36 (4.09)	39.61 (4.56)	39.25 (5.06)	39.25 (5.27)	39.25 (5.10)
32	34.61 (5.13)	38.37 (5.30)	38.36 (5.43)	38.36 (5.49)	38.36 (5.28)

First Result: best parameter set from 118.28 min to 4.09 min
Second Result: all parameter sets from ~3 days to ~2 hours

(2) Scenario 'pre-processed data', 10xCV **parallel**: accuracy (min)

γ/C	1	10	100	1000	10 000
2	75.26 (1.02)	65.12 (1.03)	73.18 (1.33)	75.76 (2.35)	74.53 (4.40)
4	57.60 (1.03)	70.88 (1.02)	75.87 (1.03)	76.01 (1.33)	74.06 (2.35)
8	64.17 (1.02)	74.52 (1.03)	**77.02 (1.02)**	75.79 (1.04)	74.42 (1.34)
16	68.57 (1.33)	76.07 (1.33)	76.40 (1.34)	75.26 (1.05)	74.53 (1.34)
32	70.21 (1.33)	75.38 (1.34)	74.69 (1.34)	73.91 (1.47)	73.73 (1.33)

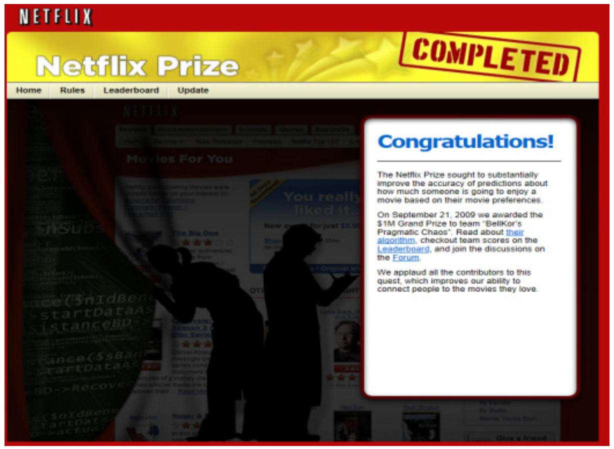
First Result: best parameter set from 14.41 min to 1.02 min
Second Result: all parameter sets from ~9 hours to ~35 min

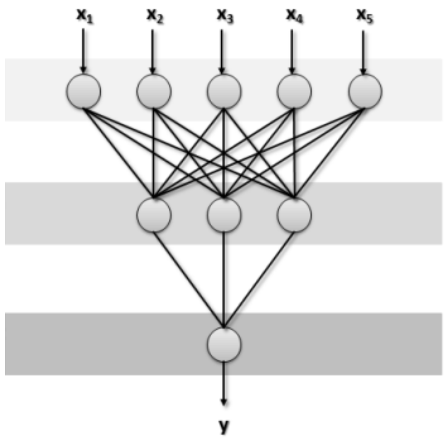
[4] G. Cavallaro, M. Riedel, J.A. Benediktsson et al., Journal of Selected Topics in Applied Earth Observation and Remote Sensing, 2015

B2SHARE Store and Share Research Data

Complex Application Example in Industry – Netflix

- ~2009 - Netflix Prize Challenge 2009
  - Data: Netflix company provided data to learn from previous movie rentals
  - Challenge: Improve Netflix in-house movie recommender system
  - Prize: 1.000.000 US $ for team with 10% improvements
  - Approaches: Machine learning algorithms and collaborative filterings
  - Winner: Prize received by working with Artificial Neural Network (ANNs)





[5] A. Töscher and M. Jahrer, 'The BigChaos Solution to the Netflix Grand Prize', 2009

Complex Application Example in Industry – Windpower

- Predictive & Instant Maintenance Workforce Management

Copa | Meter | **Wind** | Grid | Work | Loss | Churn | Tariff | Forecast | Model

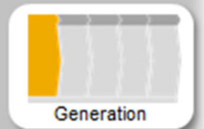
Business Value Driver

- Increase average time between inspections
- Decrease lost power generation factor
- Decrease cost of spare parts

Big Data Impact

- Ad-hoc analysis on large data volumes to predict, monitor and optimize performance and component breakdown
- Enable "Process-to-device"

Value Chain



Process

Turbine Maintenance

Value

Feasibility

Phase

1. | 2. | 3.

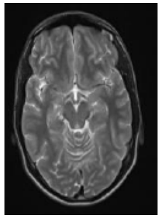
Lead User(s)

> ➢ **Slide courtesy of Dr. S. Fischer, Global Head of Applied Research – SAP AG, Germany SDIL**

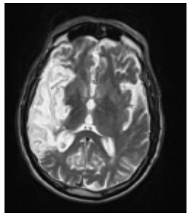
Complex Application Examples in Science & Engineering

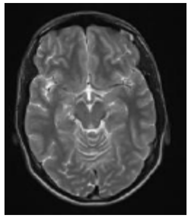
- Classification of Abnormalities in Brain MRI Images
  - Using Support Vector Machines (SVMs)
  - ‘Classify images between normal and abnormal along with type of disease depending upon features.‘

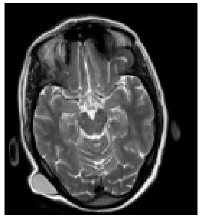
[6] D. Singh et al., 2012

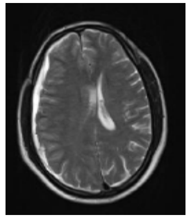

input data

?


class
Infected by clot


class
normal brain


class
Infected by tumor


class
Infected by bleed

- Classification of buildings from multi-spectral satellite data
  - Using Support Vector Machines (SVMs)
  - Classify land cover using image data & data preprocessing methods

[7] G. Cavallaro & M. Riedel et al., 2014

Class	Training	Test
Buildings	18126	163129
Blocks	10982	98834
Roads	16353	147176
Light Train	1606	14454
Vegetation	6962	62655
Trees	9088	81792
Bare Soil	8127	73144
Soil	1506	13551
Tower	4792	43124
Total	77542	697859

?

different types of land cover

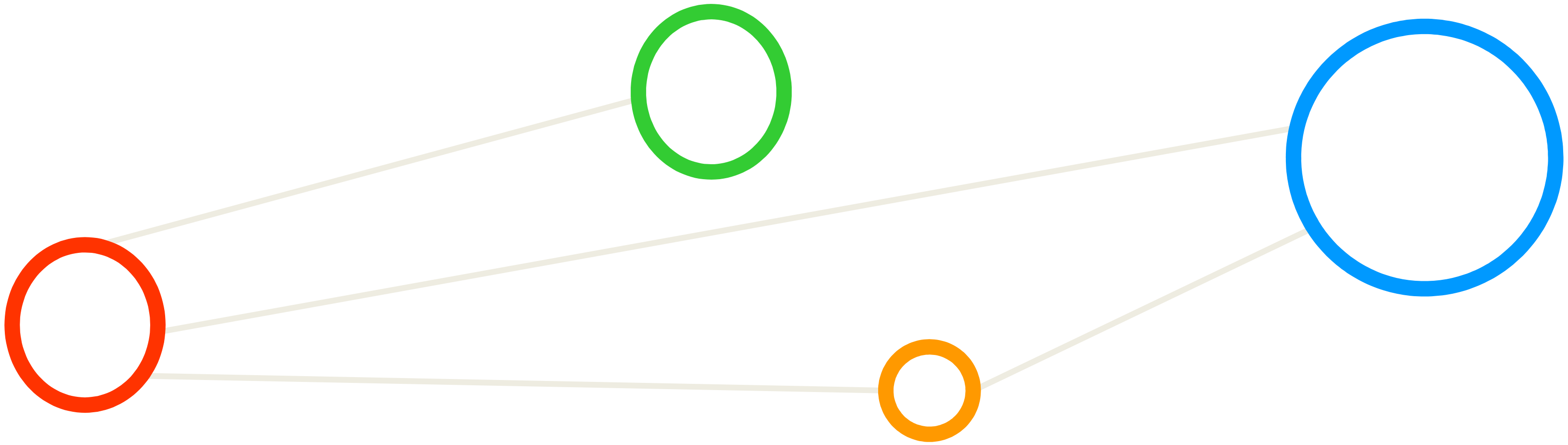
Exercises

[Video] Contamination of Data: Training, Testing, Validation

(relative high-level but captures the essence of unseen data and differences between testing & validation)

[8] YouTube Video, 'Machine Learning : Model Selection & Cross Validation'

Lecture Bibliography



Lecture Bibliography

- [1] Introduction to Data Mining, Pang-Ning Tan, Michael Steinbach, Vipin Kumar, Addison Wesley, ISBN 0321321367, English, ~769 pages, 2005
- [2] An Introduction to Statistical Learning with Applications in R,
  Online: http://www-bcf.usc.edu/~gareth/ISL/index.html
- [3] LibSVM Webpage,
  Online: https://www.csie.ntu.edu.tw/~cjlin/libsvm/
- [4] G. Cavallaro, M. Riedel, J.A. Benediktsson et al., 'On Understanding Big Data Impacts in Remotely Sensed Image Classification using Support Vector Machine Methods', IEEE Journal of Selected Topics in Applied Earth Observation and Remote Sensing, 2015
- [5] Andreas Töscher and Michael Jahrer, The BigChaos Solution to the Netflix Grand Prize, 2009
- [6] D. Singh and K. Kaur, 'Classification of Abnormalities in Brain MRI Images Using', International Journal of Engineering and Advanced Technology, ISSN: 2249 – 8958, Volume 1, Issue-6, 2012
- [7] G. Cavallaro and M. Riedel, 'Smart Data Analytics Methods for Remote Sensing Applications', 35th Canadian Symposium on Remote Sensing (IGARSS), 2014, Quebec, Canada
- [8] YouTube Video, 'Machine Learning :: Model Selection & Cross Validation',
  Online: http://www.youtube.com/watch?v=hihuMBCuSlU

- Acknowledgements and more Information: Yaser Abu-Mostafa, Caltech Lecture series, YouTube


services
measurement
within
fusion
new
funding
Policy-based
concepts
device
Analysis
forms
cross-disciplinary
resources
Centre
climate
Computer
expertise
Approaches
construction
Fusion
year
like
Computational
Test
E.g
Computing
disciplines
results
Enable
Cross-Disciplinary
methods
High
big
nodes
PB
community
Environment
Modelling
different
EU
Science
centers
Key
scientific
important
model
etc
DKRZ
Use
storage
basis
Infrastructure
files
international
simulations
ENES
technologies
SMAQ
engineering
References
formats
physical
brain
increasing
via
EUDAT
IPCC
Big
Project
Data
processing
Common
TB
Methods
MapReduce
create
computing
Using
European
databases
GB
Supercomputing
Work
images
codes
analysis
DLCL
stored
Resources
analyze
Large
Summary
data-intensive
Service
Scientific
levels
often
use
Level
Research
Matters
Concrete
external
performance
JSC
computational
research
e.g
science
HPC
Juelich
system
datasets
acquisition
large-scale
compute
Landscape
management
System
Climate
modelling
AR5
Hardware
distributed
cores
access
hundreds
Nutshell
Services
Understanding
structures
Technologies
Earth
Structure
network
DLCLs
Germany
Simulation
directory
project
General
current
Provide
NASA
Energy
systems
day
Processing
Health