Aalto University
School of Electrical Engineering

CS-C3240 – Machine Learning D

Model validation and Selection

Stephan Sigg

Department of Communications and Networking
Aalto University, School of Electrical Engineering
stephan.sigg@aalto.fi

Version 1.0, January 24, 2022

Learning goals

- Data preparation
- Model performance: Confusion matrices, precision, recall, F-score
- Common Issues: High bias/variance problems, Regularization
- Drawing learning curves
- Comparing different models

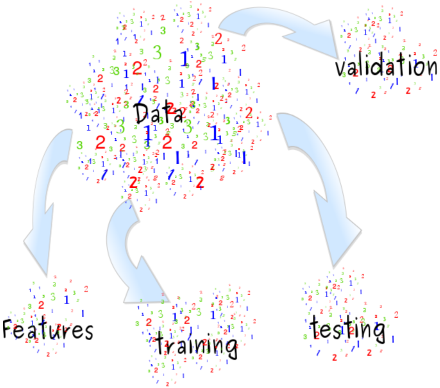
Outline

Data collection and preparation

Bias – Variance tradeoff

Evaluation of model performance

Data, data, data, ... or not (?)



Data, data, data, ... or not (?)

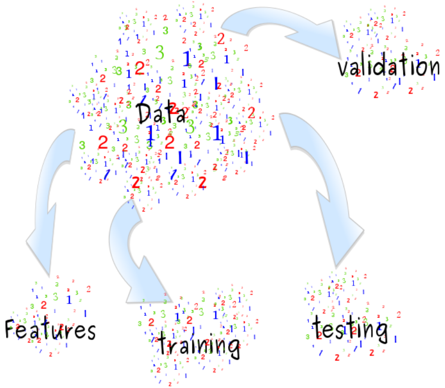
Use separate data sets

Feature selection Identify meaningful features

Training Train a model with given features

Testing Test a trained model and features

Model selection Find a best model given features

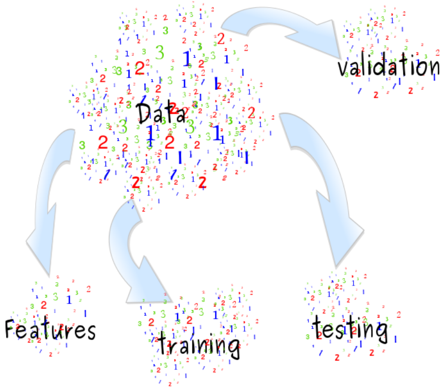


Data, data, data, ... or not (?)

Use separate data sets

Feature selection Identify meaningful features
Training Train a model with given features
Testing Test a trained model and features
Model selection Find a best model given features

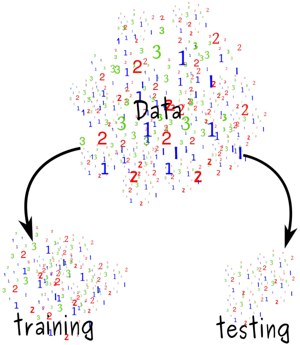
Using the same set for multiple purposes may result in biased results



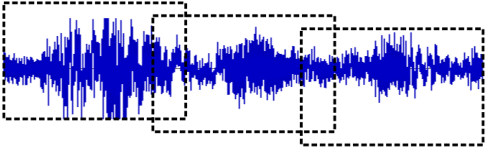
Perparation for training and testing

Separating the data

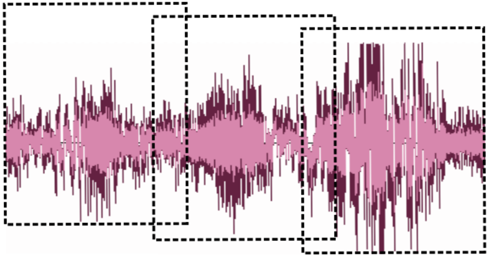
Data scarcity	Most data for training, rest for testing
Diversity	Use several runs with different data sets
Randomness	Avoid deterministic separation of data
Correlation	Training and testing data from different sessions where possible
Domination	Class-sizes during training should be equal



Pitfalls in separating the data



This also contributes to a more general distribution of the collected data, i.e. less biased with respect to a particular experimental setting.

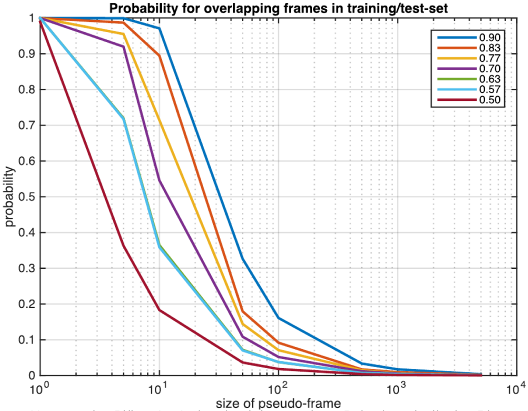


Pitfalls in separating the data

Implications of overlapping:

Overlapping windows for feature computation may cause correlation after data separation

This also contributes to a more general distribution of the collected data, i.e. less biased with respect to a particular experimental setting.



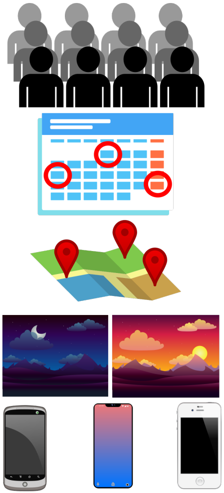
Hammerla, Plötz: Let's (not) stick together: Pairwise similarity Biases cross-validation in activity recognition, Ubicomp 2015

Pitfalls in separating the data

Minimize risk of correlation:

collect data over

- multiple subjects
- multiple environments
- multiple days
- multiple times of day
- diverse sensing hardware



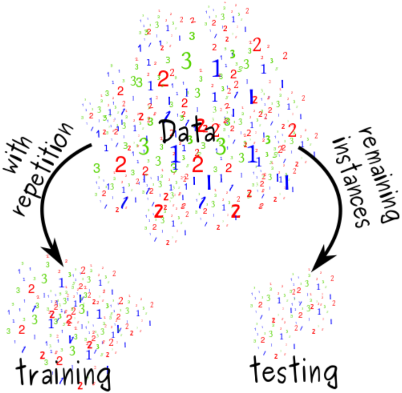
This also contributes to a more general distribution of the collected data, i.e. less biased with respect to a particular experimental setting.

Training the model – random data separation

0.632 Bootstrap

- Training: n instances with replacement
- Testing: all instances not in training
- Prob. to pick a specific instance twice:

$$
\begin{aligned}
& 1-\left(1-\frac{1}{n}\right)^{n-1} \\
\approx\ & 1-e^{-1} \\
\approx\ & 0.632
\end{aligned}
$$

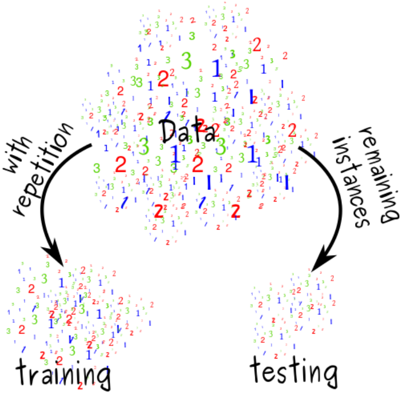


Training the model – random data separation

0.632 Bootstrap

- Training: n instances with replacement
- Testing: all instances not in training
- Prob. to pick a specific instance twice:

$$
\begin{aligned}
& 1-\left(1-\frac{1}{n}\right)^{n-1} \\
\approx\ & 1-e^{-1} \\
\approx\ & 0.632
\end{aligned}
$$



Risk: Correlation of testing and training data

Training the model – varying data distributions

k-fold cross-validation

Builds: Multiple distributions in training and testing data

Training the model – varying data distributions

k-fold cross-validation

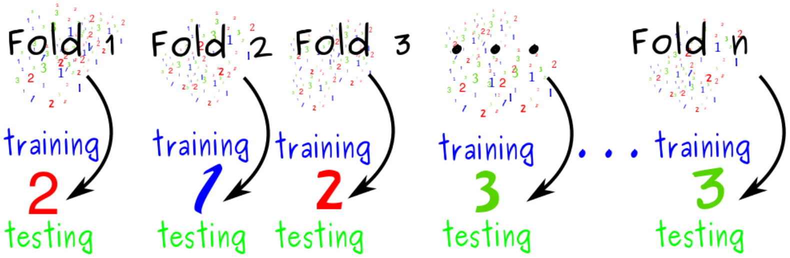
Builds: Multiple distributions in training and testing data

Avoid: random generation of training/testing sets from same data $\rightarrow$ correlation

Training the model on scarce data

Leave-one-out cross-validation

- n-fold cross-validation where n is the number of sample instances
- Leave out each instance once; train model on remaining instances
- Estimate performance on left-out instances (success/failure)

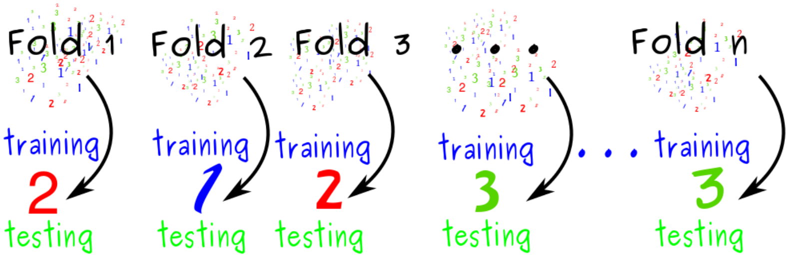


Training the model on scarce data

Leave-one-out cross-validation

- n-fold cross-validation where n is the number of sample instances
- Leave out each instance once; train model on remaining instances
- Estimate performance on left-out instances (success/failure)

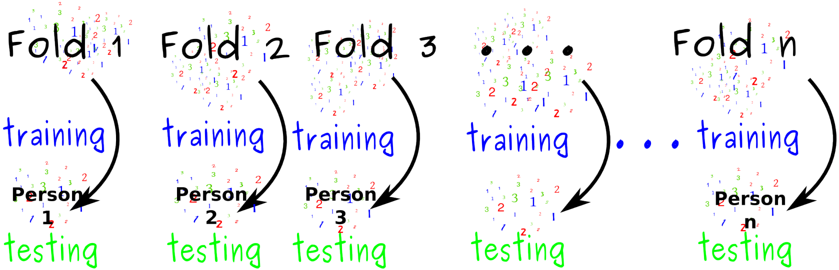
Caution: Possible correlation from data sampled in same condition



Training the model on data with known correlation

Leave-one-person-out cross-validation

- n-fold cross validation where n is the number of subjects
- Repeat: leave out instances from 1 subject; train on remaining data
- Avoids inner-subject correlation
- Left-out condition e.g. person, environment, day, ...



Outline

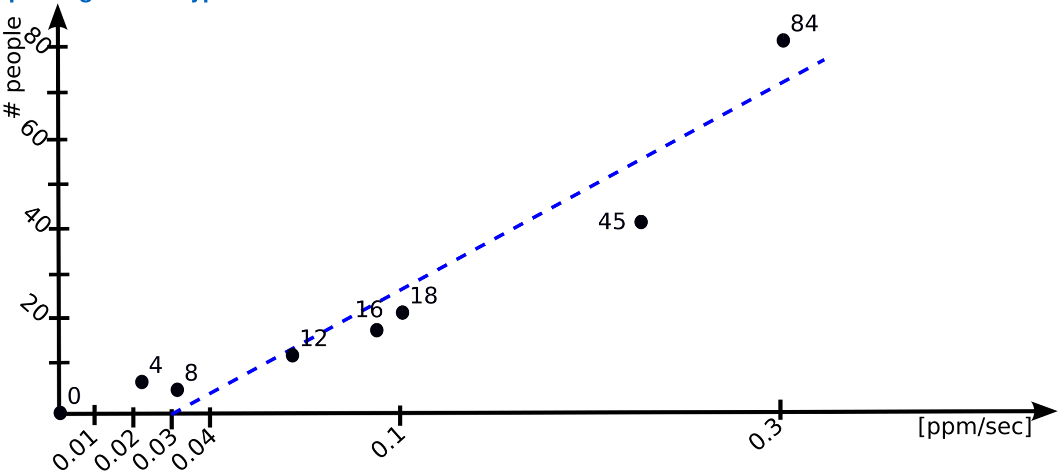
Data collection and preparation

Bias – Variance tradeoff

Evaluation of model performance

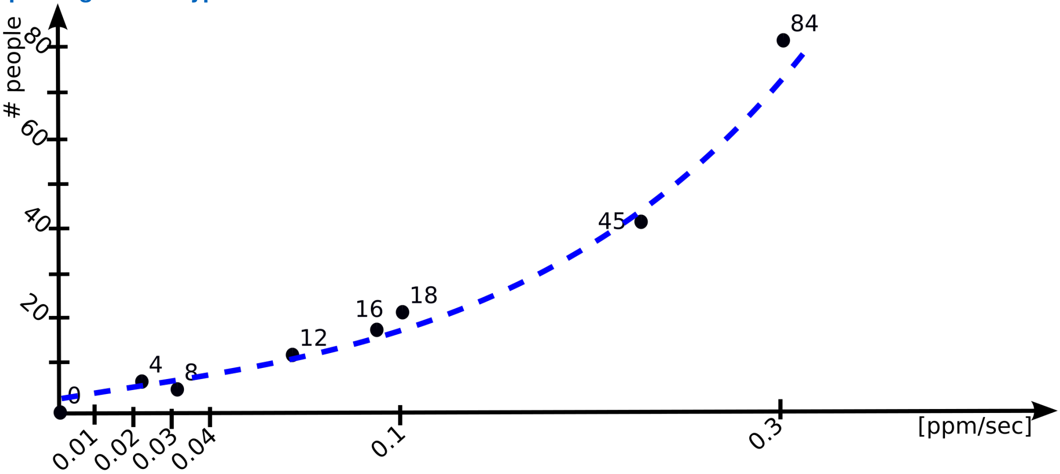
Bias - Variance tradeoff

Example: regression-type model



Bias - Variance tradeoff

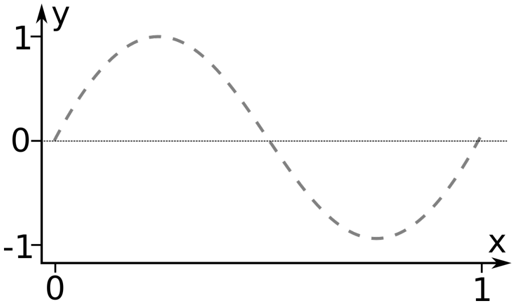
Example: regression-type model



Bias - Variance tradeoff

Example

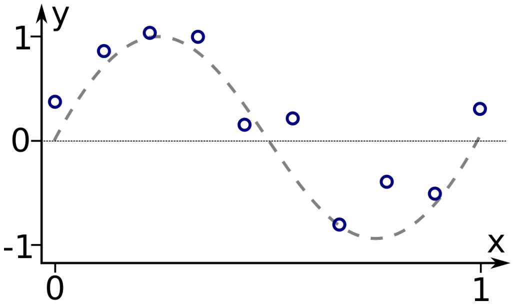
Sample points are created for the function $\sin(2\pi x) + \mathcal{N}$ where $\mathcal{N}$ is a random noise value



Bias - Variance tradeoff

Example

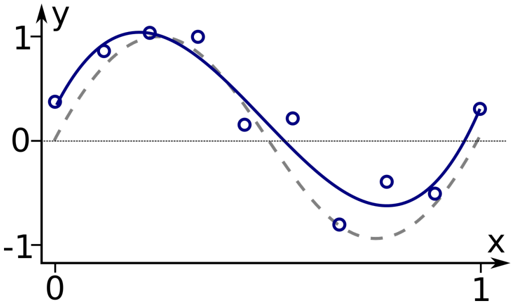
Sample points are created for the function $\sin(2\pi x) + \mathcal{N}$ where $\mathcal{N}$ is a random noise value



Bias - Variance tradeoff

Example

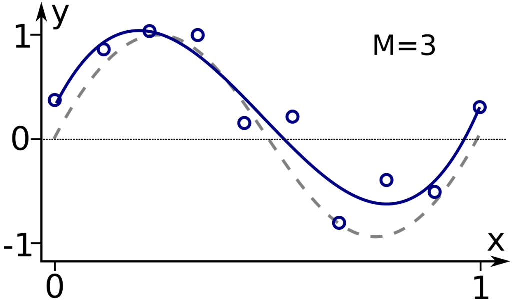
Sample points are created for the function $\sin(2\pi x) + \mathcal{N}$ where $\mathcal{N}$ is a random noise value



Bias - Variance tradeoff

We fit the data points into a polynomial function:

$$h(x, \vec{w}) = w_0 + w_1 x + w_2 x^2 + \cdots + w_M x^M = \sum_{j=0}^{M} w_j x^j$$



Bias - Variance tradeoff

We fit the data points into a polynomial function:

$$h(x, \overrightarrow{w}) = w_0 + w_1 x + w_2 x^2 + \cdots + w_M x^M = \sum_{j=0}^{M} w_j x^j$$

This can be obtained by minimising a loss function which measures the misfit between $h(x, \overrightarrow{w})$ and the training data set:

$$L[(\mathcal{X}, \mathcal{Y}), h(\cdot)] = \frac{1}{2n} \sum_{i=1}^{n} \left[h(x_i, \overrightarrow{w}) - y_i\right]^2$$

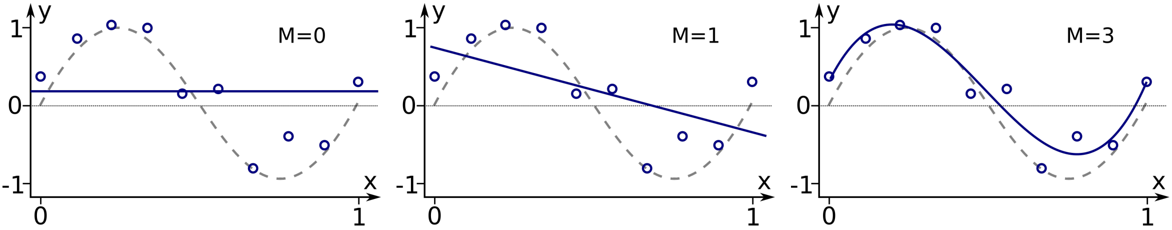
$L[(\mathcal{X}, \mathcal{Y}), h(\cdot)] \geq 0$;
$L[(\mathcal{X}, \mathcal{Y}), h(\cdot)] = 0$ IFF all points are covered by the function

Bias - Variance tradeoff

One problem is the right choice of the dimension M

When M is too small, the approximation accuracy might be bad

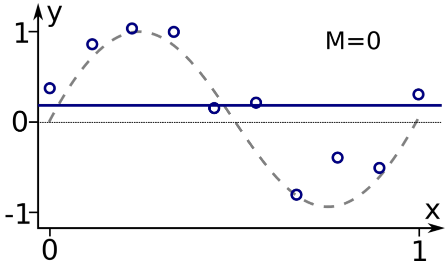
$$h(x, \overrightarrow{w}) = w_0 + w_1 x + w_2 x^2 + \cdots + w_M x^M = \sum_{j=0}^{M} w_j x^j$$

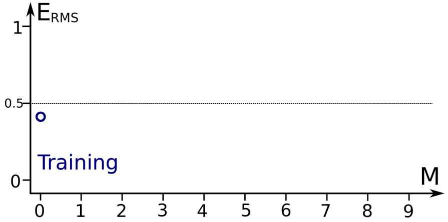


Bias - Variance tradeoff

Visualise loss $L[(\mathcal{X},\mathcal{Y}),h(\cdot)]$ wrt the data by Root of the Mean Squared (RMS)

$$E_{RMS} = \sqrt{\frac{\sum_{i=1}^{n}(L[x_i,y_i),h(x_i,\overrightarrow{w})])^2}{n}}$$

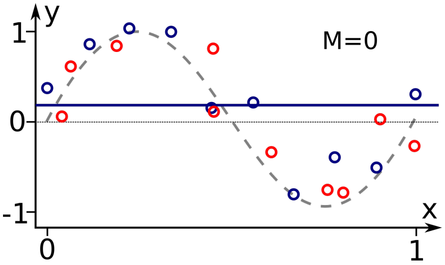


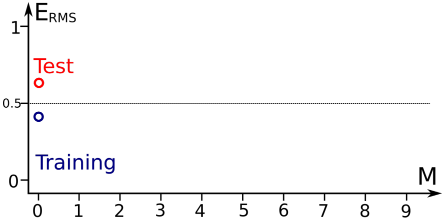


Bias - Variance tradeoff

Visualise loss $L[(\mathcal{X},\mathcal{Y}), h(\cdot)]$ wrt the data by Root of the Mean Squared (RMS)

$$E_{RMS} = \sqrt{\frac{\sum_{i=1}^{n}(L[x_i, y_i), h(x_i, \overrightarrow{w})])^2}{n}}$$

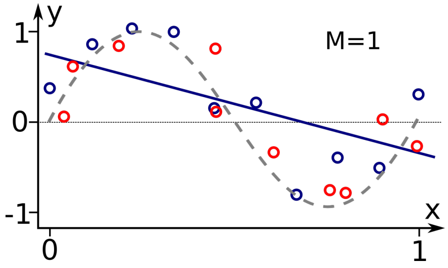


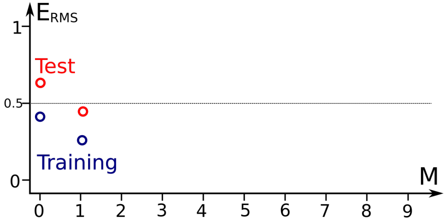


Bias - Variance tradeoff

Visualise loss $L[(\mathcal{X},\mathcal{Y}),h(\cdot)]$ wrt the data by Root of the Mean Squared (RMS)

$$E_{RMS} = \sqrt{\frac{\sum_{i=1}^{n}(L[x_i,y_i),h(x_i,\vec{w})])^2}{n}}$$

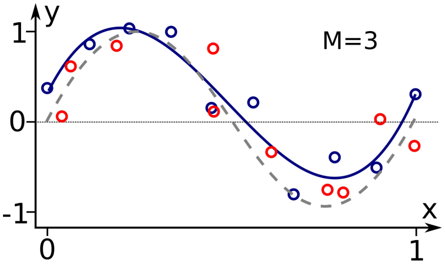


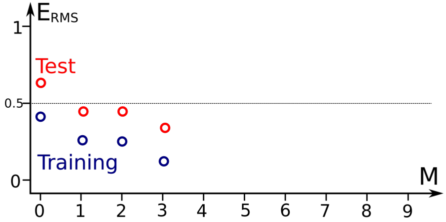


Bias - Variance tradeoff

Visualise loss $L[(\mathcal{X}, \mathcal{Y}), h(\cdot)]$ wrt the data by Root of the Mean Squared (RMS)

$$E_{RMS} = \sqrt{\frac{\sum_{i=1}^{n}(L[x_i, y_i), h(x_i, \overrightarrow{w})])^2}{n}}$$

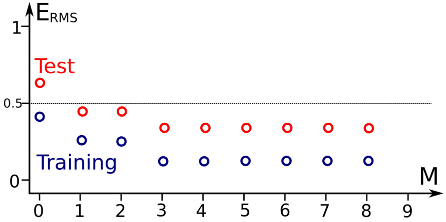




Bias - Variance tradeoff

Visualise loss $L[(\mathcal{X}, \mathcal{Y}), h(\cdot)]$ wrt the data by Root of the Mean Squared (RMS)

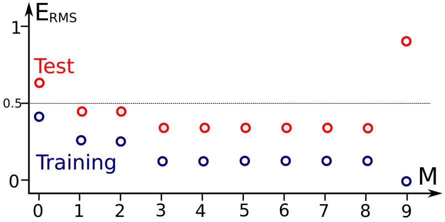
$$E_{RMS} = \sqrt{\frac{\sum_{i=1}^{n}(L[x_i, y_i), h(x_i, \overrightarrow{w})])^2}{n}}$$



Bias - Variance tradeoff

Visualise loss $L[(\mathcal{X}, \mathcal{Y}), h(\cdot)]$ wrt the data by Root of the Mean Squared (RMS)

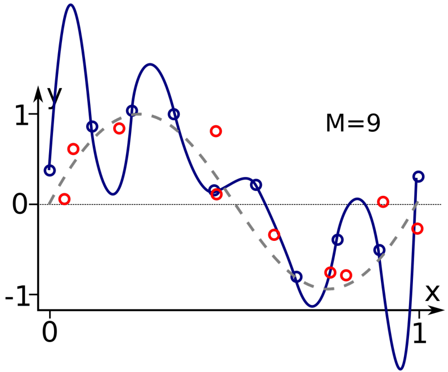
$$E_{RMS} = \sqrt{\frac{\sum_{i=1}^{n}(L[x_i, y_i), h(x_i, \vec{w})])^2}{n}}$$

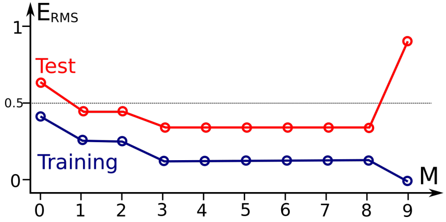


Bias - Variance tradeoff

Visualise loss $L[(\mathcal{X}, \mathcal{Y}), h(\cdot)]$ wrt the data by Root of the Mean Squared (RMS)

$$E_{RMS} = \sqrt{\frac{\sum_{i=1}^{n}(L[x_i, y_i), h(x_i, \overrightarrow{w})])^2}{n}}$$



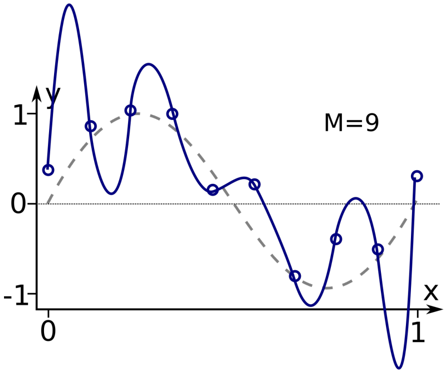


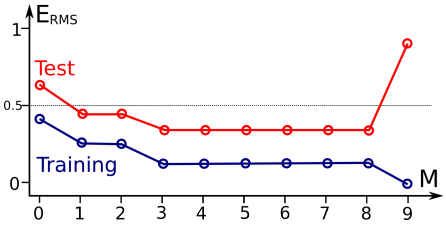
Bias - Variance tradeoff

This event is called overfitting

The polynomial is now trained too well to the training data

It performs badly on test data

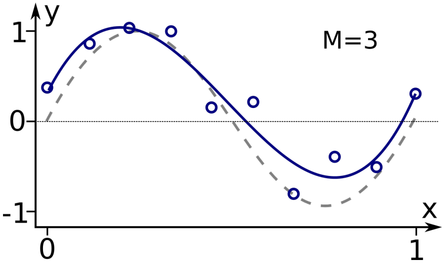


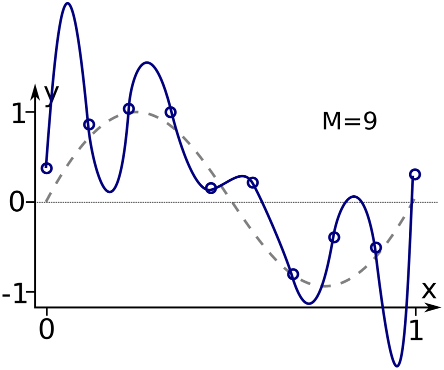


Bias - Variance tradeoff

When M becomes too big, the polynomial will cross all points exactly

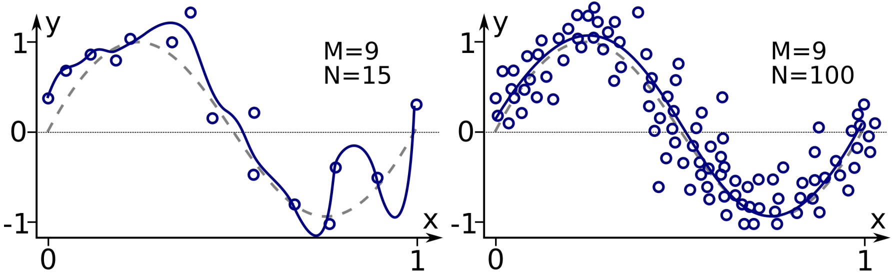
For $M = n$, it is always possible to create a polynomial of order M that contains all values in the data set.





Bias - Variance tradeoff

With increasing number of data points, the problem of overfitting becomes less severe for a given value of M



Bias and Variance in training a learning model

Bias

- inability of machine learning model to capture the true distribution of the data

  Example: Linear model to describe non-linear relationship between data and labels

  e.g. linear regression is expected to have a high bias error; in contrast to other algorithms that take less hard assumptions (e.g. decision trees, k-Nearest Neighbours, Support Vector Machines)

Bias and Variance in training a learning model

Bias

- inability of machine learning model to capture the true distribution of the data

High bias: more assumption in the learning algorithm on the underlying distribution

Low bias: fewer assumptions in the learning algorithm

Bias and Variance in training a learning model

Variance

- model overfits on a particular dataset (learning to fit very closely to the points of a particular dataset)

  Example: Generally, nonlinear machine learnign algorithms like decision trees have a high variance

Bias and Variance in training a learning model

Variance

- model overfits on a particular dataset (learning to fit very closely to the points of a particular dataset)

  Example: Generally, nonlinear machine learnign algorithms like decision trees have a high variance

  Low variance algorithms: Linear regression, logistic regression, linear discriminant analysis

  High variance algorithms: Decision Trees, k-NN, support vector machines

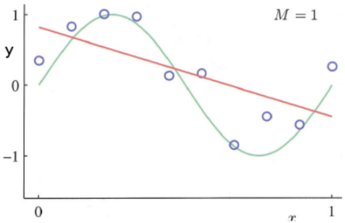
Model selection

High Bias
(underfitting)

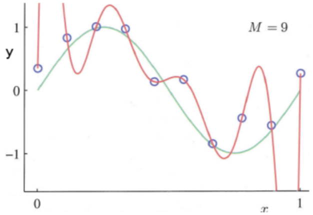
High Variance
(overfitting)

Model selection

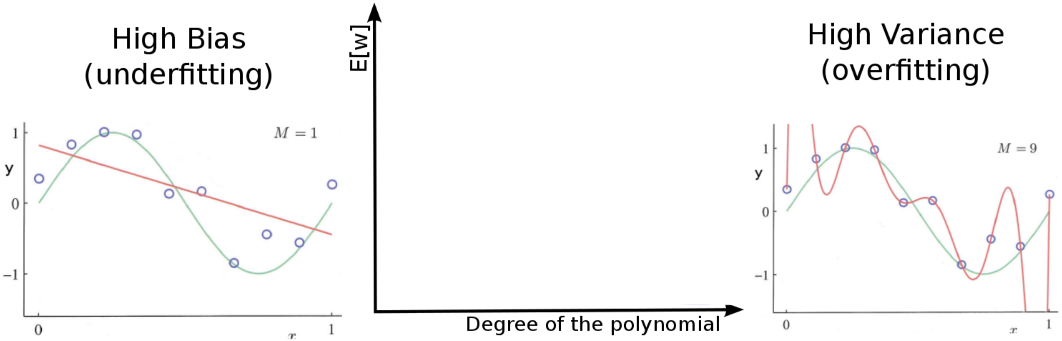
High Bias
(underfitting)



High Variance
(overfitting)

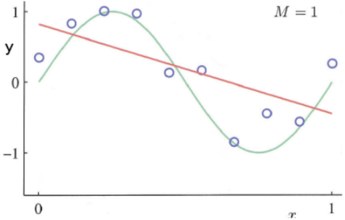


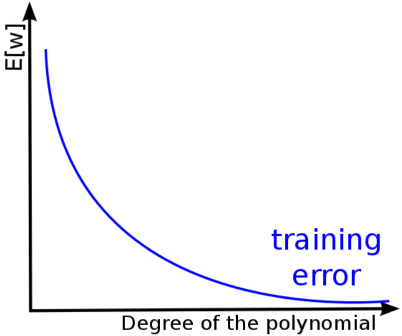
Model selection



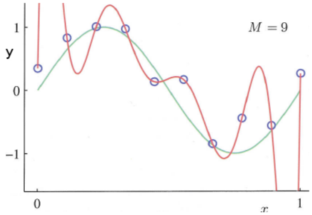
Model selection

High Bias
(underfitting)

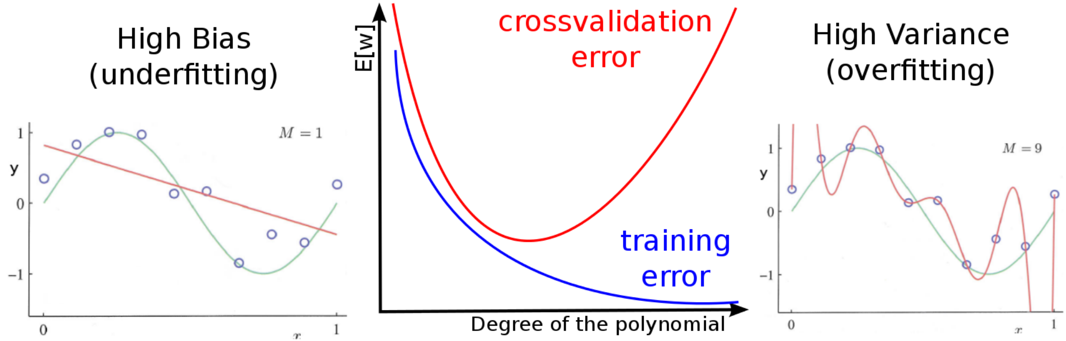




High Variance
(overfitting)

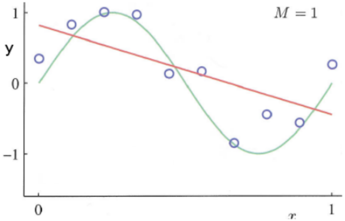


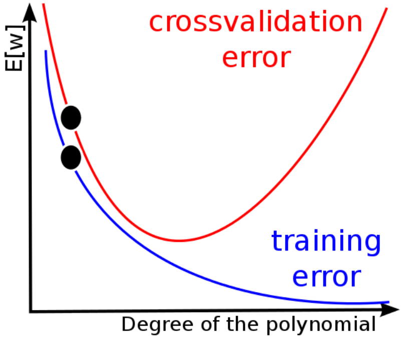
Model selection



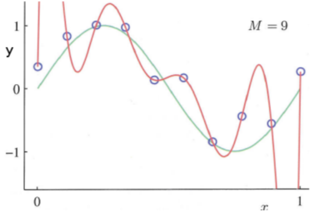
Model selection

High Bias
(underfitting)

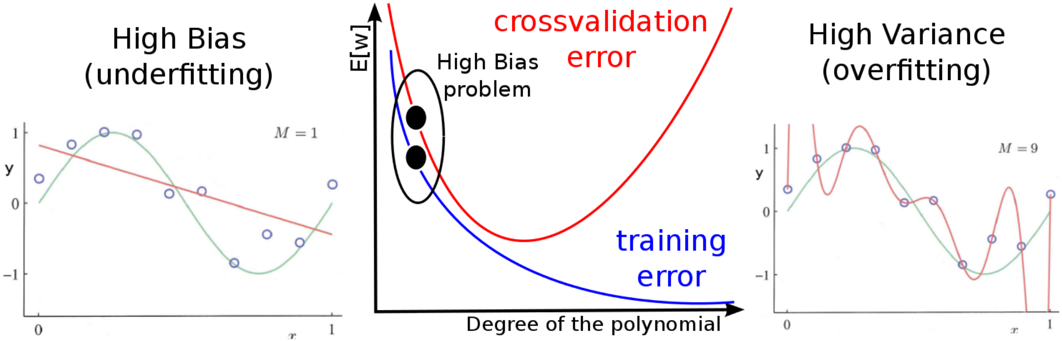




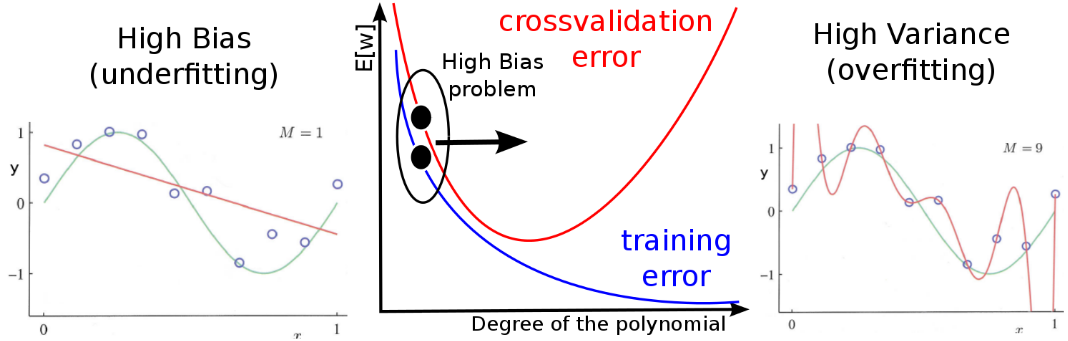
High Variance
(overfitting)



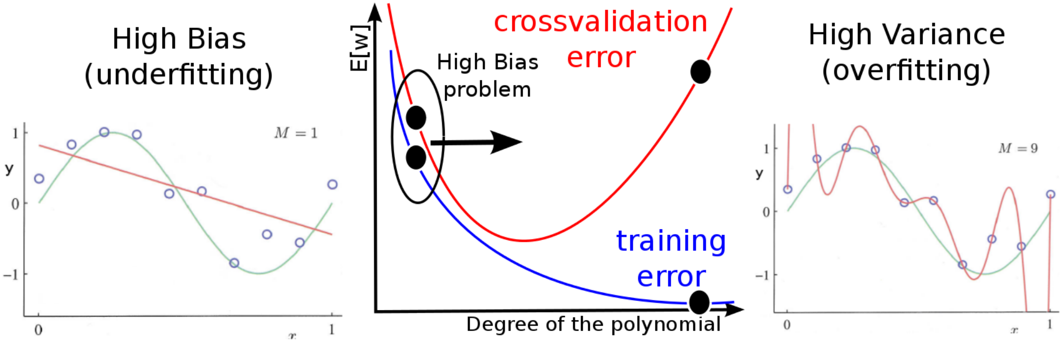
Model selection



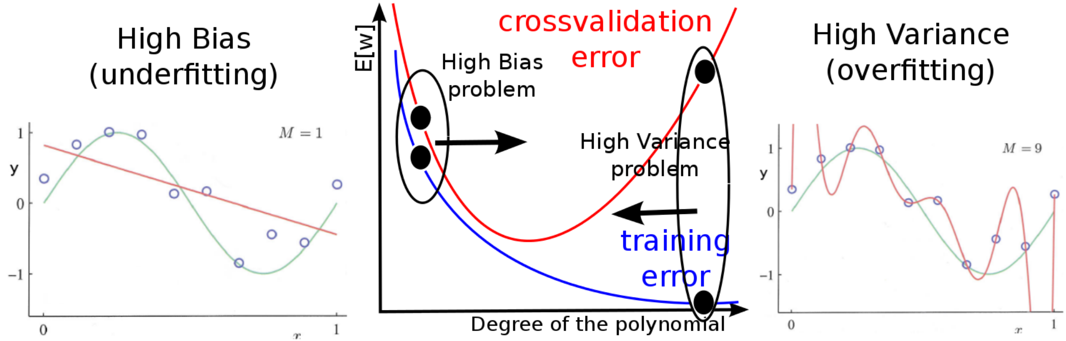
Model selection



Model selection



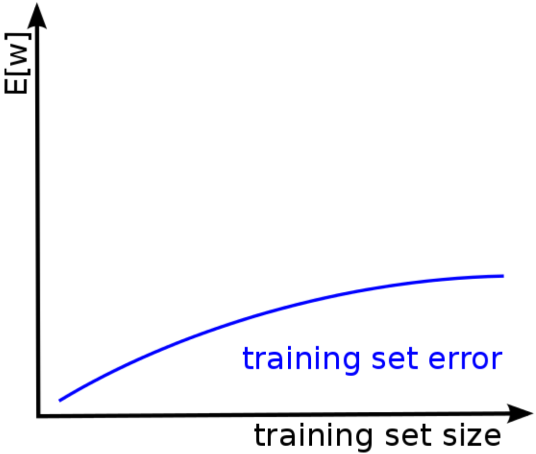
Model selection



Learning curves

Learning Curves

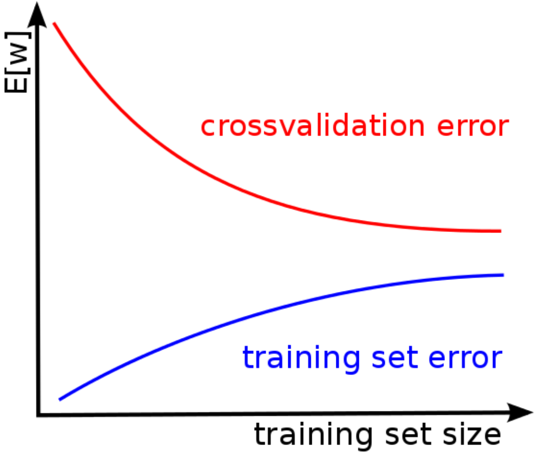
Plotting learning curves helps to find out, whether our algorithm suffers from high variance or high bias

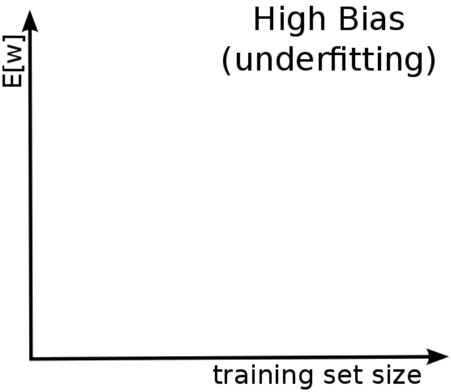


Learning curves

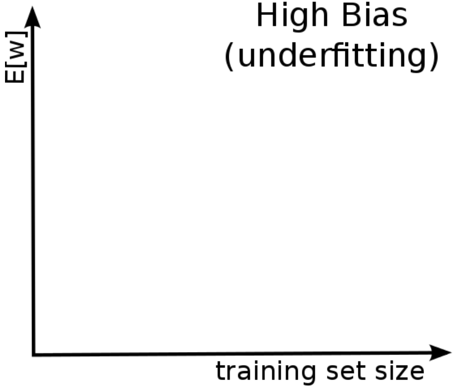
Learning Curves

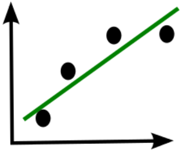
Plotting learning curves helps to find out, whether our algorithm suffers from high variance or high bias

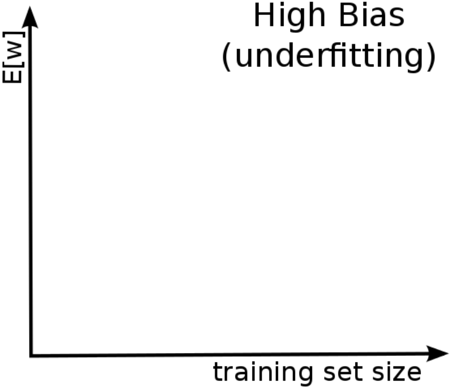



High Bias
(underfitting)
E[w]
training set size

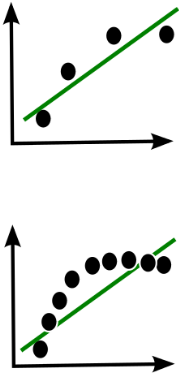
High Bias (underfitting)

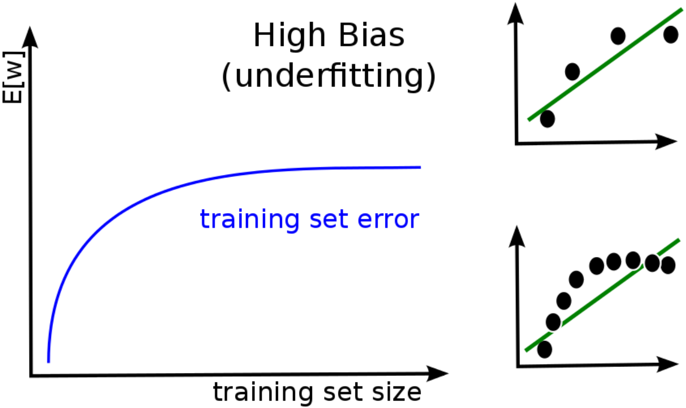


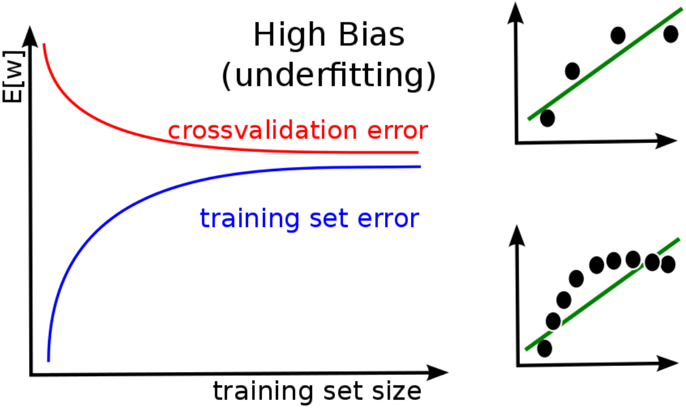




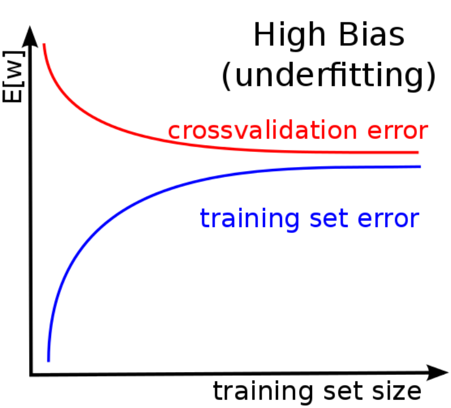
High Bias
(underfitting)

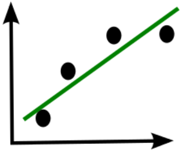


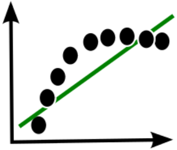

E[w]
High Bias
(underfitting)
training set error
training set size



High Bias
(underfitting)
E[w]
crossvalidation error
training set error
training set size



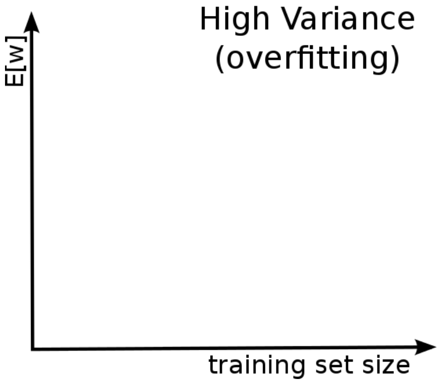




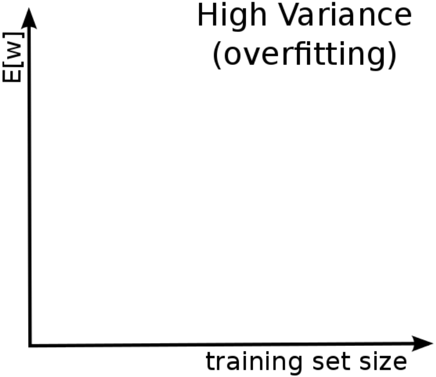
When the algorithm suffers from high Bias...

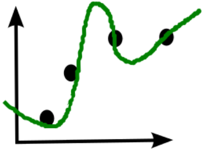
→ crossvalidation error and training error are close

→ Increasing the training set size does not help !

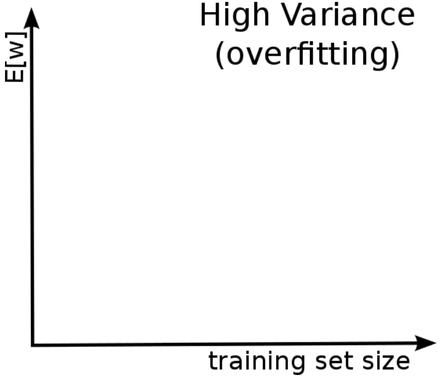

High Variance
(overfitting)
E[w]
training set size

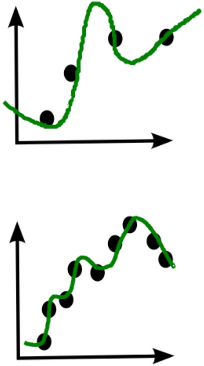
High Variance (overfitting)



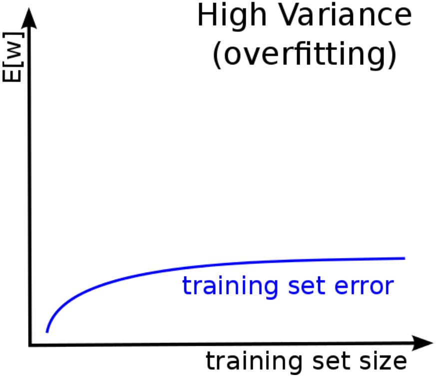


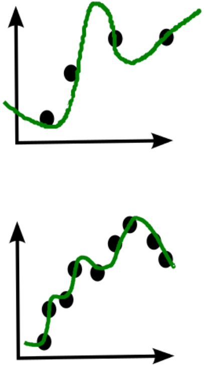
High Variance (overfitting)

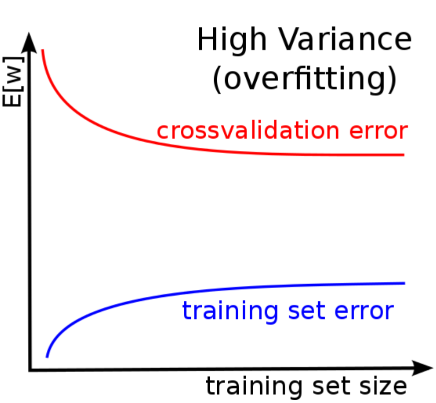


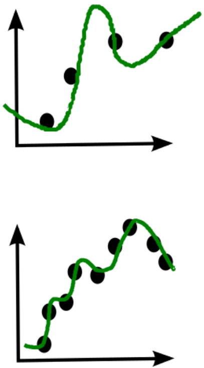


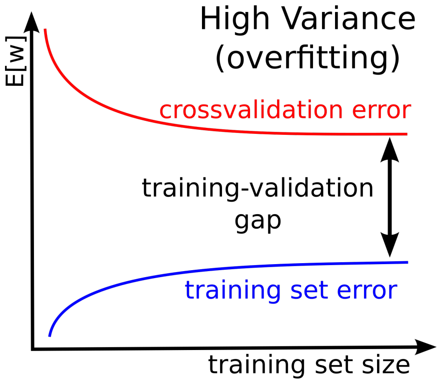
High Variance (overfitting)

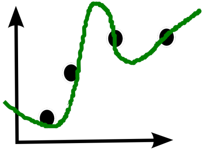


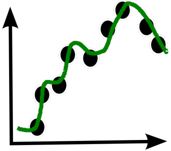



High Variance
(overfitting)
E[w]
crossvalidation error
training set error
training set size









When the algorithm suffers from high variance...

- → crossvalidation error and training error are far apart
- → Increasing the training set size improves the performance

Outline

Data collection and preparation

Bias – Variance tradeoff

Evaluation of model performance

Evaluation of model performance

Evaluation of classification performance

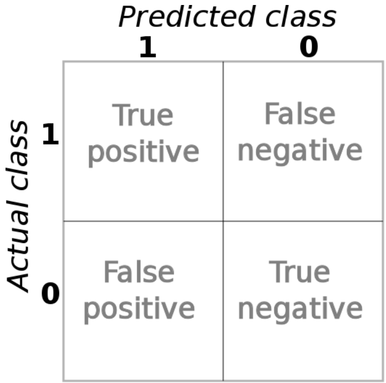
Classification accuracy

- Confusion matrices
- Precision
- Recall
- F_1-score

	Classification							
	Aw	No	To	Sb	Sl	Sr	St	$\sum$
Aw	**52**		3	6	0	17	22	100
No		**436**	25	7	6	17	9	500
To		40	**59**				1	100
Sb	15	22		**32**	4	22	5	100
Sl	12	11	1	6	**48**	8	14	100
Sr	4	15		6	1	**67**	7	100
St	3	18	1	1	24	10	**43**	100
$\sum$	92	551	86	65	94	129	83	

	Classification							
	Aw	No	To	Sb	Sl	Sr	St	recall
Aw	**.58**	.09		.13	.11	.05	.04	.58
No		**.872**	.05	.014	.012	.034	.018	.872
To		.4	**.59**				.01	.59
Sb	.15	.22		**.32**	.04	.22	.05	.32
Sl	.12	.11	.01	.06	**.48**	.08	.14	.48
Sr	.04	.15		.06	.01	**.67**	.07	.67
St	.03	.18	.01	.01	.24	.1	**.43**	.43
prec	.630	.791	.686	.492	.511	.519	.518	

Evaluation of model performance



Evaluation of model performance

Precision

Of all samples that were predicted with $y = 1$, what fraction actually belongs to class 1?

		Predicted class	
		1	**0**
Actual class	**1**	True positive	False negative
	0	False positive	True negative

Evaluation of model performance

Precision

Of all samples that were predicted with $y = 1$, what fraction actually belongs to class 1?

Recall

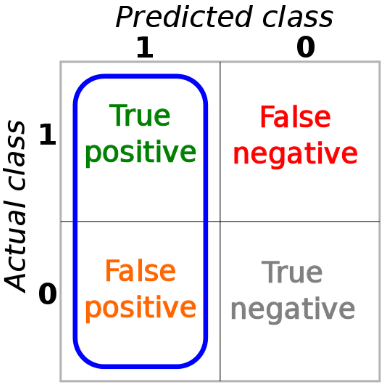
Of all samples that actually belong to class 1, which fraction has been correctly predicted with $y = 1$?

		Predicted class	
		1	**0**
Actual class	**1**	True positive	False negative
	0	False positive	True negative

Evaluation of model performance

Precision

$$\frac{\text{True positive}}{\text{True positive} + \text{False positive}}$$



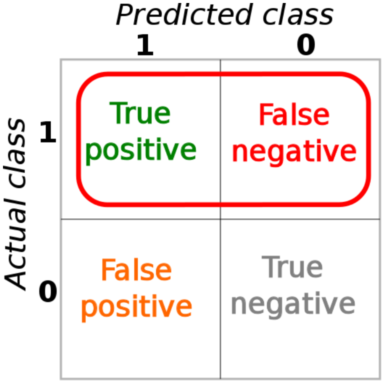
Evaluation of model performance

Precision

$$\frac{\text{True positive}}{\text{True positive} + \text{False positive}}$$

Recall

$$\frac{\text{True positive}}{\text{True positive} + \text{False negative}}$$



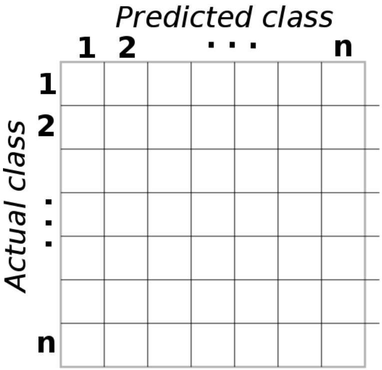
Evaluation of model performance

Precision

$$\frac{\text{True positive}}{\text{True positive} + \text{False positive}}$$

Recall

$$\frac{\text{True positive}}{\text{True positive} + \text{False negative}}$$



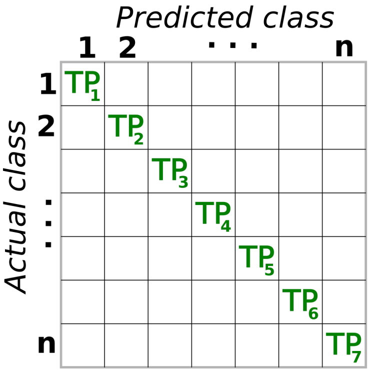
Evaluation of model performance

Precision

$$\frac{\text{True positive}}{\text{True positive} + \text{False positive}}$$

Recall

$$\frac{\text{True positive}}{\text{True positive} + \text{False negative}}$$



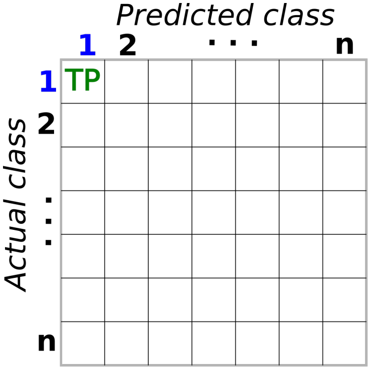
Evaluation of model performance

Precision

$$\frac{\text{True positive}}{\text{True positive} + \text{False positive}}$$

Recall

$$\frac{\text{True positive}}{\text{True positive} + \text{False negative}}$$



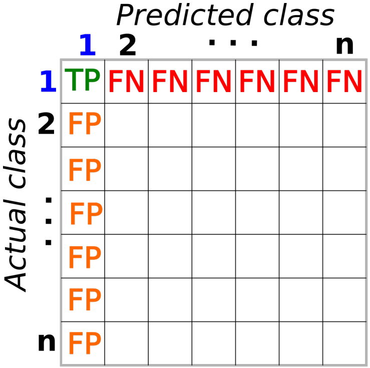
Evaluation of model performance

Precision

$$\frac{\text{True positive}}{\text{True positive} + \text{False positive}}$$

Recall

$$\frac{\text{True positive}}{\text{True positive} + \text{False negative}}$$



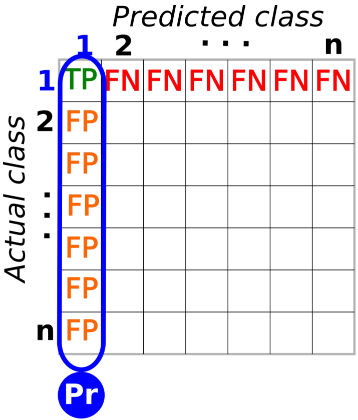
Evaluation of model performance

Precision

$$\frac{\text{True positive}}{\text{True positive} + \text{False positive}}$$

Recall

$$\frac{\text{True positive}}{\text{True positive} + \text{False negative}}$$



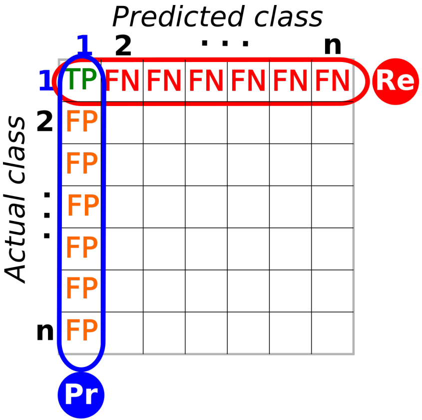
Evaluation of model performance

Precision

$$\frac{\text{True positive}}{\text{True positive} + \text{False positive}}$$

Recall

$$\frac{\text{True positive}}{\text{True positive} + \text{False negative}}$$



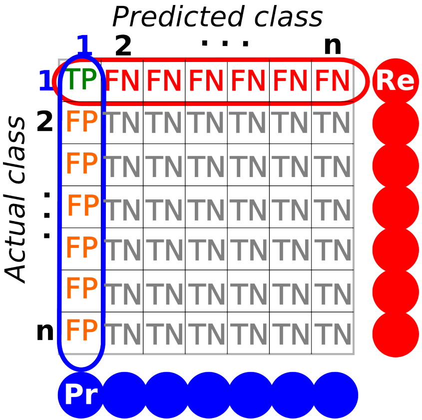
Evaluation of model performance

Precision

$$\frac{\text{True positive}}{\text{True positive} + \text{False positive}}$$

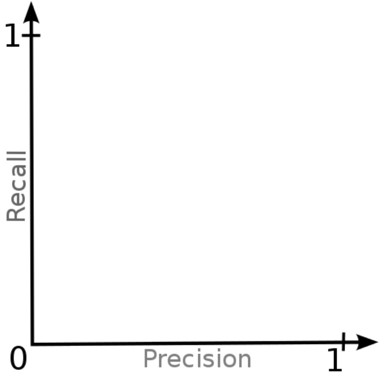
Recall

$$\frac{\text{True positive}}{\text{True positive} + \text{False negative}}$$



Evaluation of model performance

Tradeoff between precision and recall



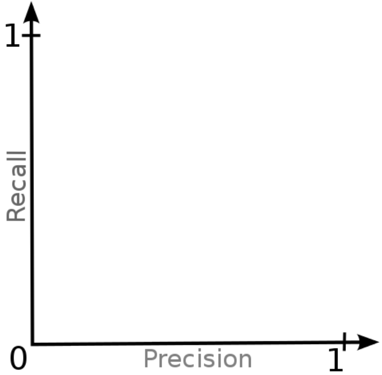
Evaluation of model performance

Tradeoff between precision and recall

Precision

$$\frac{\text{True positive}}{\text{True positive} + \text{False positive}}$$

Predict a particular class only if very confident
$\Rightarrow$ High precision (minimize false positives)



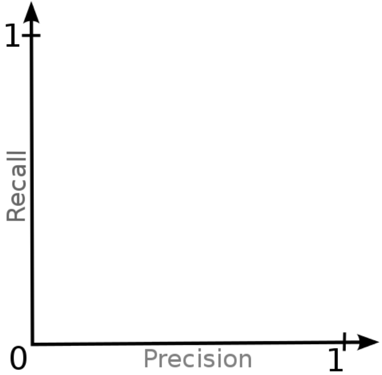
Evaluation of model performance

Tradeoff between precision and recall

Recall

$$\frac{\text{True positive}}{\text{True positive} + \text{False negative}}$$

Minimise false negatives $\Rightarrow$ <u>High recall</u>



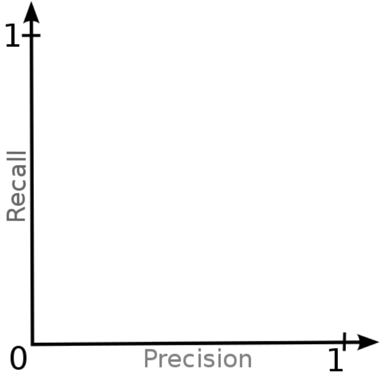
Evaluation of model performance

Tradeoff between precision and recall

F_1 Score

Combines precision and recall into a single decision variable

$$F_1 \text{ Score:} \quad 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$



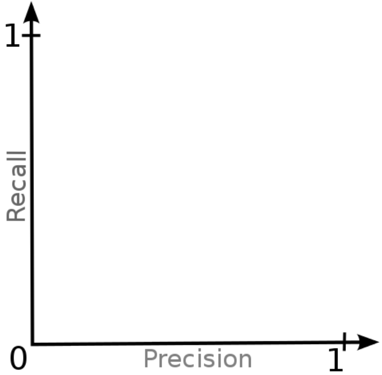
Evaluation of model performance

Tradeoff between precision and recall

F_β Score

Recall is considered β times as important as precision (for $\beta \in \mathbb{R}$)

$$F_\beta\text{Score:} \quad (1+\beta^2) \cdot \frac{\text{Precision} \cdot \text{Recall}}{\beta^2 \cdot \text{Precision} + \text{Recall}}$$



Comparing different models

Why accuracy is not enough

Comparing different models – Information score

Let C be the correct class of an instance and $\mathcal{P}(C)$, $\mathcal{P}'(C)$ be the prior and posterior probability of a classifier to predict that class

Define:[1]

$$I_i = \begin{cases} \log(\mathcal{P}'(C)) - \log(\mathcal{P}(C)) & \text{if } \mathcal{P}'(C) \geq \mathcal{P}(C) \\ -\log(1 - \mathcal{P}'(C)) + \log(1 - \mathcal{P}(C)) & \text{else} \end{cases}$$

The information score *(amount of information gained)* is then

$$\text{IS} = \frac{1}{|\mathcal{C}|} \sum_{i=1}^{|\mathcal{C}|} I_i$$

Note:
$\mathcal{P}(C) = \mathcal{P}'(C) \rightarrow I_i = 0$

[1] I. Kononenko and I. Bratko: Information-Based Evaluation Criterion for Classifier's Performance, Machine Learning, 6, 67-80, 1991.

Comparing different models – Brier score

The Brier score is defined as

$$\text{Brier} = \frac{1}{|\mathcal{C}|} \sum_{i=1}^{|\mathcal{C}|} (t(C_i) - \mathcal{P}(C_i))^2$$

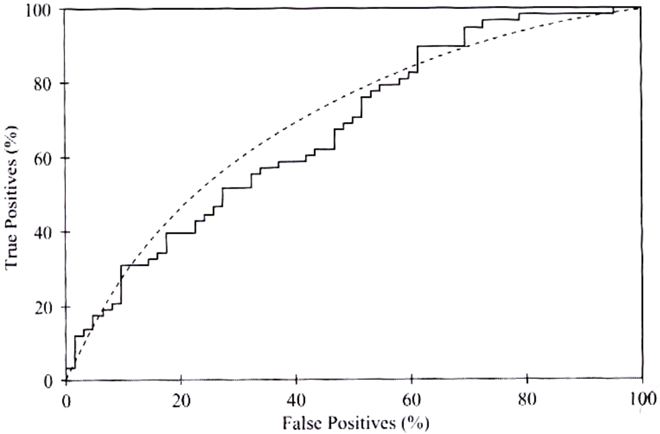
where

$$t(C_i) = \begin{cases} 1 & \text{if } C_i \text{ is the correct class } (C_i = C) \\ 0 & \text{else} \end{cases}$$

and $\mathcal{P}(C_i)$ is the probability the classifier assigned to class C_i.

Comparing different models – ROC curves

Area under the receiver operating characteristic (ROC) curve (AUC)



If probability distributions for TP and FP known, ROC curve is generated by plotting cumulative distribution function of the TP versus CDF of FP

Questions?

Stephan Sigg
stephan.sigg@aalto.fi

Si Zuo
si.zuo@aalto.fi

Literature

- C.M. Bishop: Pattern recognition and machine learning, Springer, 2007.
- R.O. Duda, P.E. Hart, D.G. Stork: Pattern Classification, Wiley, 2001.