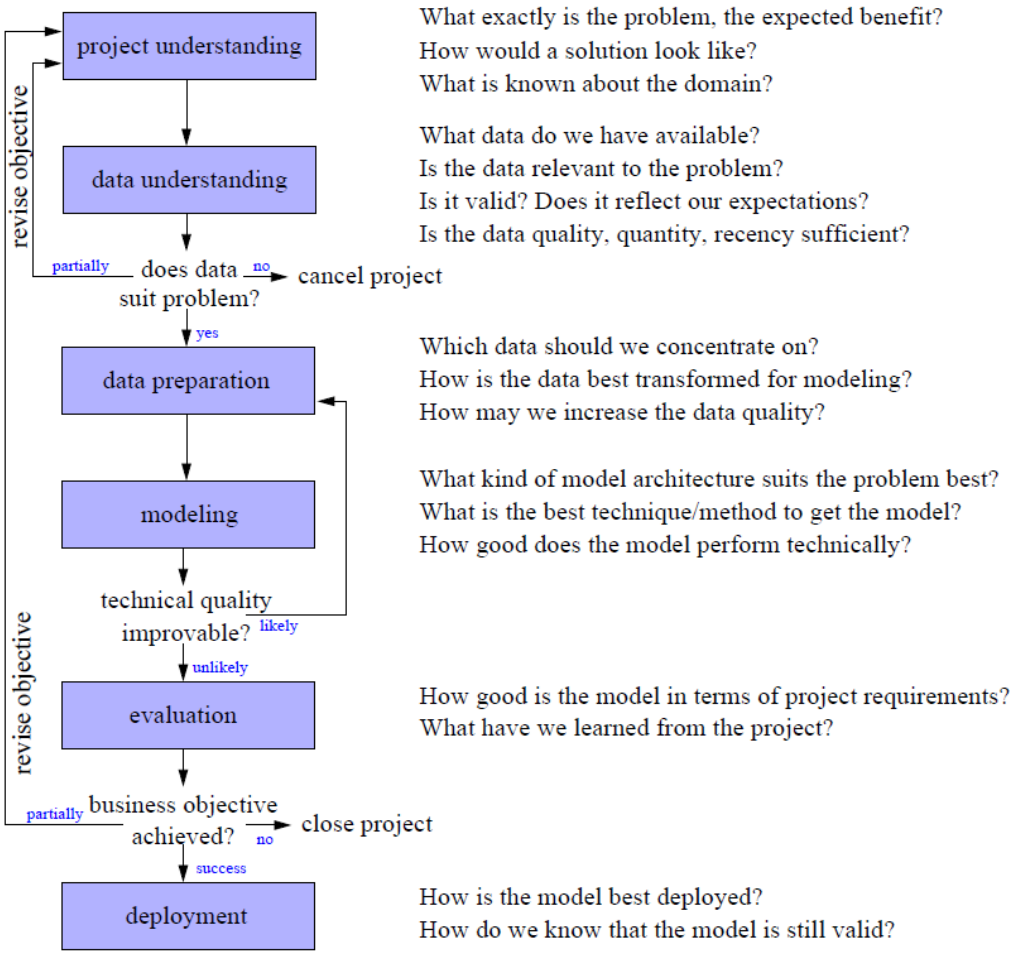
Data Analysis and Knowledge Discovery

Jukka Heikkonen

University of Turku
Department of Information Technology

Where are we now?



Outline

Evaluating the predictive accuracy

- Using a supervised learning method (like k-nn), we are constructing a classifier or regressor
- How accurately does it predict?
- Learning: the learning algorithm tries to find a predictor that fits well the data (e.g. by choosing the weights of a linear model), how can one measure this fit?
- Model selection: many algorithms have additional hyperparameters (e.g. k and distance function for k-nn) that need to be set before learning. How can one compare models learned with different hyperparameters, which one is better?
- Final evaluation: our customer wants to know how well the final model works.

Evaluating the predictive accuracy

- Real thing we would like to measure is how much money we make from using the model, how many lives are saved, how many people are happier...
- Maybe sometimes possible for final evaluation of the model, but not during development
- For a predictive model, the natural criterion to measure is how accurately it predicts
- Basic measures: misclassification rate, squared error
- Advanced: area under ROC curve (AUC), for binary classification problems
- Many more in different application domains

Evaluating the predictive accuracy

We are given a test set of input-output pairs

$$\{(x_1, y_1), \ldots, (x_n, y_n)\}$$

- let $g(\cdot)$ be our model, x an input and y the correct output
- We would hope that $g(x) \approx y$, for any randomly chosen new (x, y) input-output pair
- Let $m(y', y)$ be some function that measures how well a predicted value y'matches the true value y
- Obviously, average performance on this test set is

$$\frac{1}{n} \sum_{i=1}^{n} m(g(x_i), y_i)$$

Misclassification rate

$$m(y', y) = \begin{cases} 1 & \text{if } y' \neq y, \\ 0 & \text{otherwise} \end{cases}$$

- misclassification rate is the fraction of incorrectly classified examples, a number between zero and one
- if 10% classified incorrectly, then misclassification rate 0.1
- conversely, classification accuracy defined as the fraction of correctly classified instances

Misclassification rate

- A low misclassification rate does not necessarily tell anything about the quality of a classifier.
- A low misclassification rate might be easily achieved when classes are extremely unbalanced, i.e. when the distribution of classes deviates strongly from a uniform distribution.
- Example. Data from production of tea cups where the classes are *broken* and *ok*.
  When 99% of the production are *ok*, a classifier always predicting *ok* will have a misclassification rate of 1%.

Cost matrix

- More general approach than the misclassification rate: cost function or cost matrix.
- The consequences (costs) for misclassifications for one class might be different than for another class.

Cost matrix

- Example: Tea cup production.

Cost matrix

	predicted class	
true class	OK	broken
OK	0	c_1
broken	c_2	0

- When an intact cup is classified as broken, the cup must be produced again. Therefore, the costs c_1 caused by this error causes are equal to the production costs for one cup.
- The costs c_2 for a broken cup classified as intact are more difficult to estimate.
  - c_2 must cover the mailing costs for the reproduced cup.
  - costs for loss of reputation of the company for delivering broken cups very difficult to estimate.

Cost matrix

General form of a cost matrix for a multi-class classification problem:

	predicted class			
true class	c_1	c_2	$\dots$	c_m
c_1	0	$c_{1,2}$	$\dots$	$c_{1,m}$
c_1	$c_{2,1}$	0	$\dots$	$c_{2,m}$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
c_m	$c_{m,1}$	$c_{m,2}$	$\dots$	0

Cost matrix

When such a cost matrix is provided, the expected loss given by

$$\text{loss}(c_i|E) = \sum_{j=1}^{m} P(c_j|E)c_{ji}$$

should be minimized.

- E is the evidence, i.e. the observed values of the predictor attributes used for the classification.
- $P(c_j|E)$ is the predicted probability that the true class is c_j given observation E.

Cost matrix

Example. (Hypothecial) cost matrix for the tea cup production problem

	predicted class	
true class	OK	broken
OK	0	1
broken	10	0

A classifier might classify a specific cup with 80% to the class *ok* and with 20% to the class *broken*.

- Expected loss for choosing *ok*: $0.8 \cdot 0 + 0.2 \cdot 10 = 2$.
- Expected loss for choosing *broken*: $0.8 \cdot 1 + 0.2 \cdot 0 = 0.8$.

Cost matrix

Using the cost matrix

	predicted class			
true class	c_1	c_2	$\ldots$	c_m
c_1	0	1	$\ldots$	1
c_1	1	0	$\ldots$	1
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
c_m	1	1	$\ldots$	0

corresponds to minimising the misclassification rate.

Confusion matrix

Classification result can be represented by a confusion matrix which is a table where the rows represent the true classes and the columns the predicted classes.

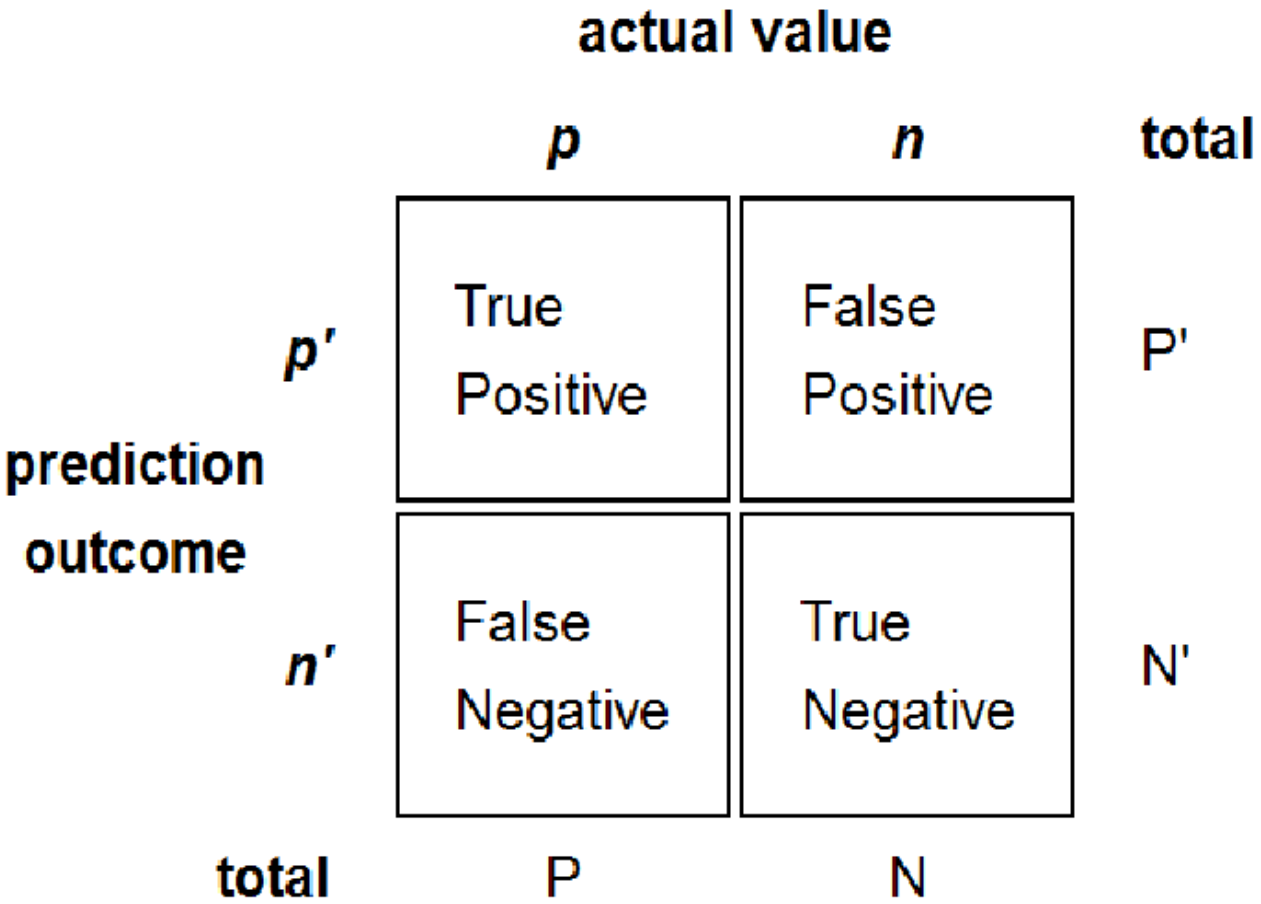
Each entry specifies how many objects from a given class are classified into the class of the corresponding column.

true class	predicted class		
	Iris setosa	Iris versicolor	Iris virginica
Iris setosa	50	0	0
Iris versicolor	0	47	3
Iris virginica	0	2	48

A possible confusion matrix for the Iris data set

Confusion matrix: true/false positives and negatives

A binary classifier case:



Squared error

$$m(y', y) = (y' - y)^2$$

- regression error, zero if the prediction correct, otherwise grows quadratically, value not bounded
- sensitive to outliers, a single very bad prediction can lead to really bad mean squared error
- mean absolute error $m(y', y) = |y' - y|$ less sensitive to outliers, but less used in practice (harder to analyze theoretically and optimize in practical algorithms)

Regression for classification

- Classification can be considered as a special case of regression.
- Example. Classification problem with two classes coded as 0 and 1.
- A classifier is a regression function that should only yield the values 0 and 1 (in the ideal case).
- The regression function g derived from the might not yield the exact values 0 and 1. This could be amended by defining

$$\tilde{g}(\mathbf{x}) = \begin{cases} 0 & \text{if } g(\mathbf{x}) \leq 0.5, \\ 1 & \text{if } g(\mathbf{x}) > 0.5. \end{cases}$$

Regression for classification

Problem. The objective function for regression does not aim at minimising the misclassification rate or a given cost function.

Example. Classification problem with 1000 instances, half of them belonging to class 0 and the other half to class 1. Compare the two regression functions g_1 and g_2 w.r.t. the mean square error.

- Regression function g_1 yields 0.1 for all data from class 0 and 0.9 for all data from class 1.
- Regression function g_2 always yields the exact and correct values 0 and 1, except for 9 data objects where it yields 1 instead of 0 and vice versa.

	Misclassification rate	Mean square error
g_1	0.000	0.010
g_2	0.009	0.009

g_1 is better than g_2 in terms of the misclassification rate, but worse in terms of the mean square error.

Scoring

How can a classifier be judged when no explicit cost matrix is known and the misclassification rate might not be a good choice?

two class problem: 'positive' and 'negative' class, encoded as '1' and '0' (often '+1' and '-1', hence the names)

- classifier may predict probability of belonging to positive class
- more generally, real-valued confidence score
- rule minimizing misclassification rate: if $f(x) > 0.5$ predict positive class, else negative
- problems with imbalanced classes, and different misclassification costs
- $f(x) > t$, moving $t \in \mathbb{R}$ away from 0.5 means favoring one class over another

Scoring example

classification problem: Iris virginica (1) vs. Iris versicolor (0)

(a very naive) classifier: simply outputs the value of attribute *Sepal Length* as confidence score to distinguish Iris virginica from Iris versicolor

$$\text{If Sepal Length} > t\text{, then virginica, otherwise versicolor}$$

where t can be chosen in the range of the values of the attribute *Sepal Length*.

IRIS data

S. Length	Species	S. Length	Species	S. Length	Species
4.9	versicolor	5.6	versicolor	5.9	versicolor
4.9	virginica	5.6	versicolor	5.9	versicolor
5.0	versicolor	5.6	virginica	5.9	virginica
5.0	versicolor	5.7	versicolor	6.0	versicolor
5.1	versicolor	5.7	versicolor	6.0	versicolor
5.2	versicolor	5.7	versicolor	6.0	versicolor
5.4	versicolor	5.7	versicolor	6.0	versicolor
5.5	versicolor	5.7	versicolor	6.0	virginica
5.5	versicolor	5.7	virginica	6.0	virginica
5.5	versicolor	5.8	versicolor	6.1	versicolor
5.5	versicolor	5.8	versicolor	6.1	versicolor
5.5	versicolor	5.8	versicolor	6.1	versicolor
5.6	versicolor	5.8	virginica	6.1	versicolor
5.6	versicolor	5.8	virginica	6.1	virginica

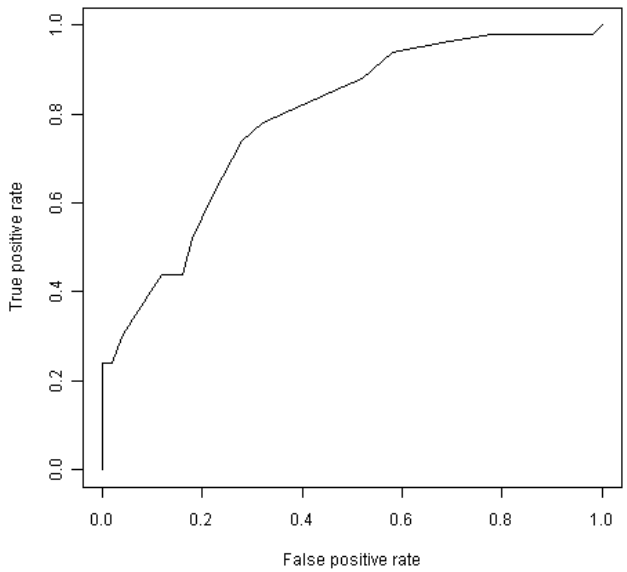
true/false positives and negatives

- Consider virginica as the "positive" and versicolor as the "negative" class.
- The higher the threshold t is chosen, the more instances are classified as "negative" (versicolor).
- Those instances classified as "positive" that belong to the class "positive", are called true positives.
- Those instances classified as "positive" that belong to the class "negative", are called false positives.
- analogously, true negatives and false negatives
- decreasing the threshold t, will increase the true positives as well as false positives
- In the ideal case, we would first get only true positives and no false negatives (if the sepal length of versicolor would always be smaller than the sepal length of virginica).

ROC curves

The ROC curve (receiver operating characteristic curve) draws the true positive rate (TPR) against the false positive rate (FPR) illustrating the performance of a binary classifier.

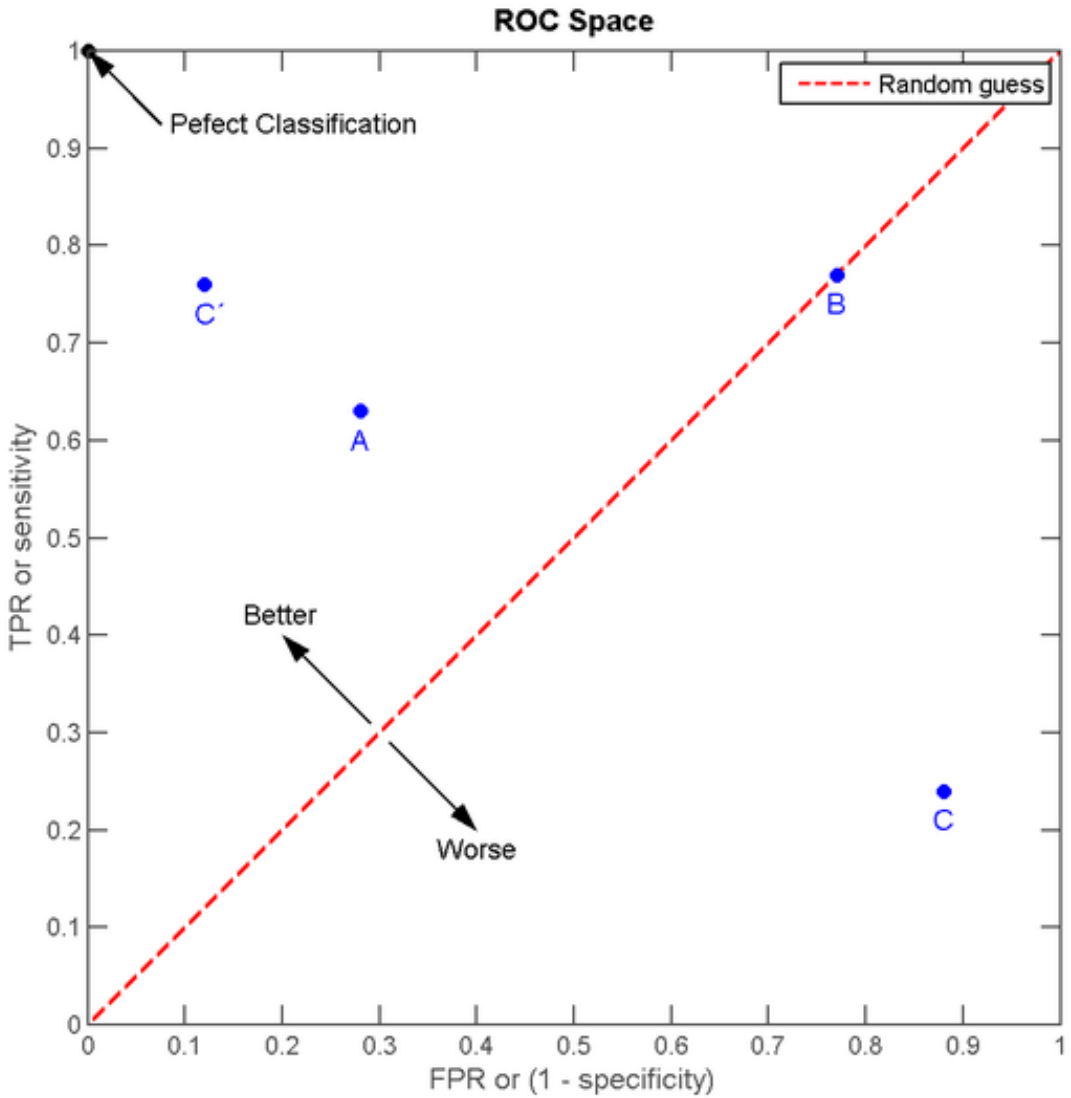
- A ROC space is defined by FPR and TPR as x and y axes respectively, which depicts relative trade-offs between TP (benefits) and FP (costs).



ROC curves

- The best possible prediction method would yield a point in the upper left corner or coordinate (0,1) of the ROC space representing 100% sensitivity (no false negatives) and 100% specificity (no false positives).
- The (0,1) point is also called a perfect classification.
- A completely random guess would give a point along a diagonal line (the so-called line of no-discrimination) from the left bottom to the top right corners. An intuitive example of random guessing is a decision by flipping coins (head or tail).

ROC curves


ROC Space
Random guess
Pefect Classification
C´
B
A
Better
Worse
C
TPR or sensitivity
FPR or (1 - specificity)
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

ROC curves

- The area under curve (AUC), i.e. the area under the ROC curve, is an indicator how well the classifier solves the problem. The larger the area, the better the solution for the classification problem.
- traditionally used in medicine for comparing diagnostic tests
- increasingly popular in data mining / machine learning
- compares two classifiers over all possible thresholds simultaneuosly
- invariant to relative class distributions
- useful when the proper misclassification costs are unknown

Area under ROC curve

$$\frac{1}{|X_+||X_-|} \sum_{x_i \in X_+} \sum_{x_j \in X_-} H(g(x_i) - g(x_j)),$$

$$H(a) = \begin{cases} 1, & \text{if } a > 0 \\ 1/2, & \text{if } a = 0 \\ 0, & \text{if } a < 0 \end{cases} .$$

- AUC: probability, that randomly chosen positive example ranked higher, than randomly chosen negative one
- similar to ranking based correlation measures considered earlier

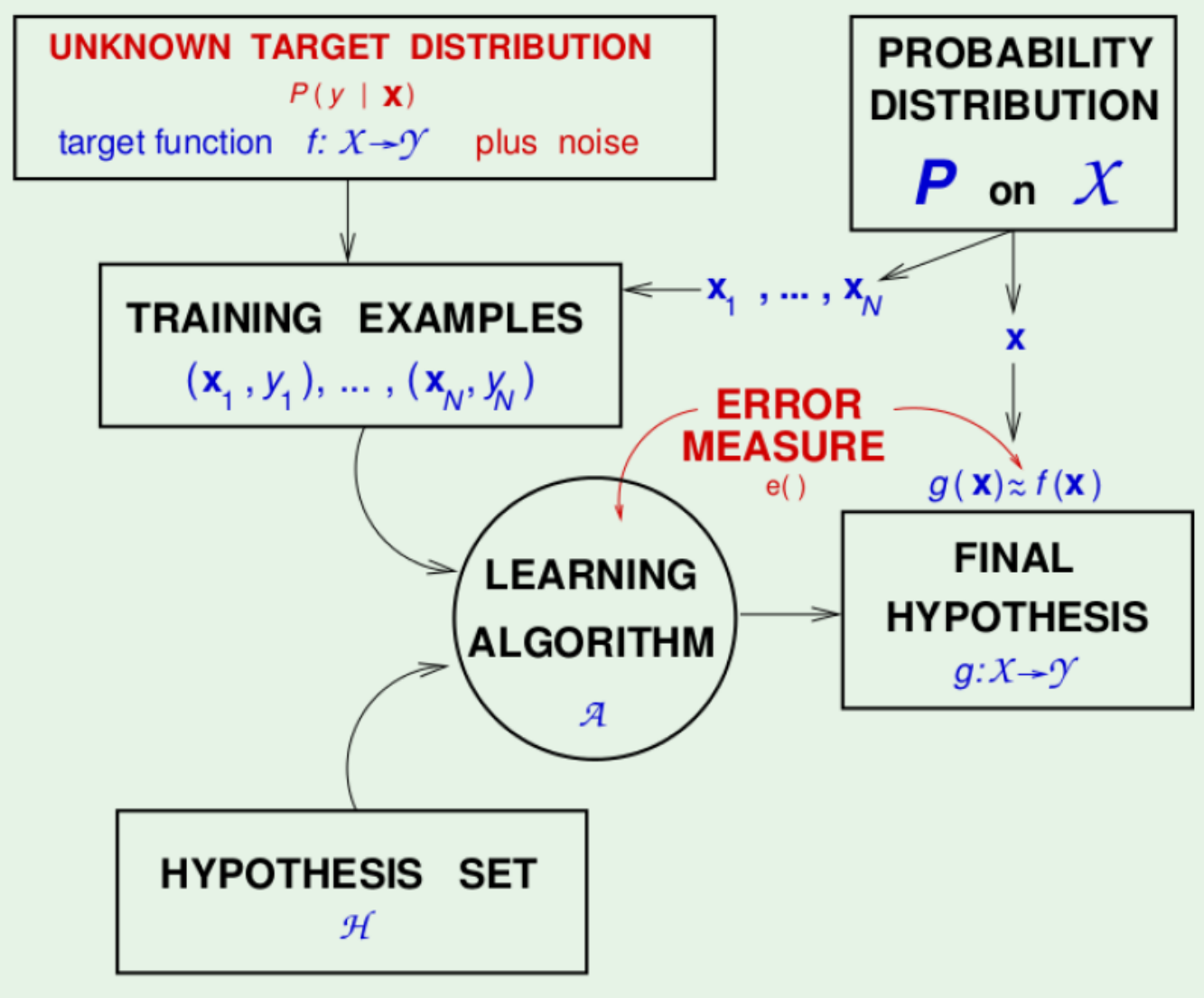
Notation

g	classifier function
x_i	i:th training instance
X_+	set of positive instances
X_-	set of negative instances

Area under ROC curve

- AUC produces a value that is between 0 and 1
- 1.0 perfect ranking, all instances of positive class receive higher predicted values than those of negative class
- 0.5 random performance, you can get the same AUC by flipping a coin or predicting a constant value always
- sanity check for binary classifier: is AUC (much) higher than 0.5? If not, you have not captured any meaningful pattern.
- (on most real problems AUCs very close to 1.0 also suspicous, usually 'too good to be true', sign that you have messed up somewhere in your evaluation protocol, e.g. overfitting and then testing on training data...)

Components Of Learning

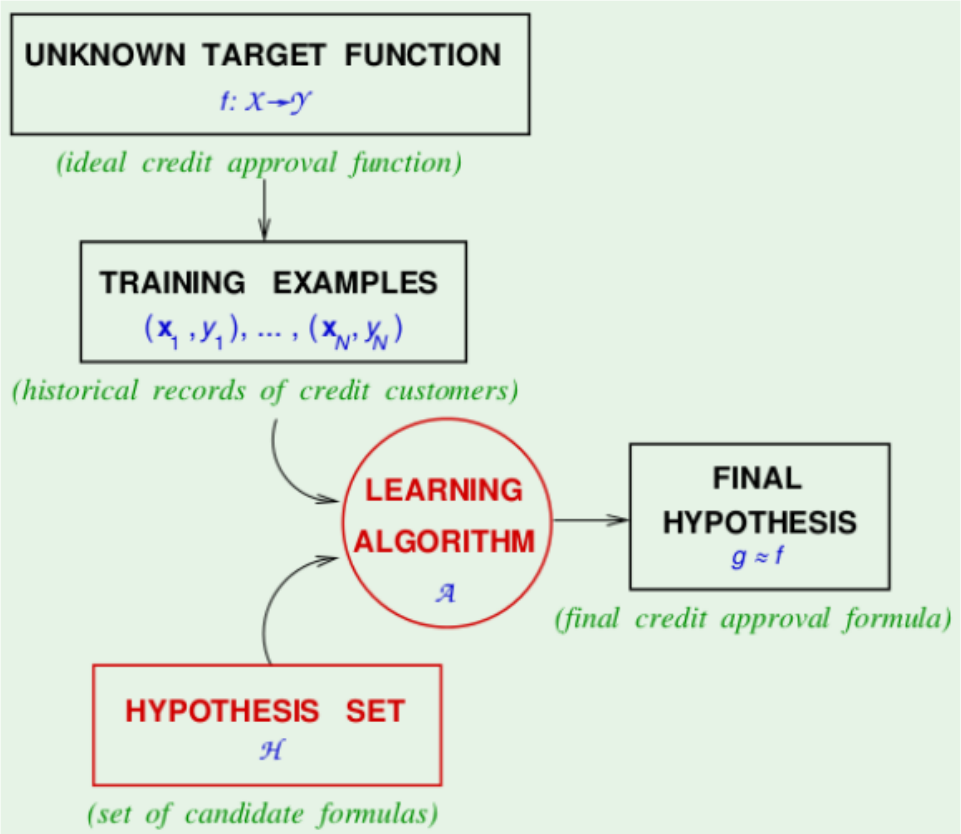


Components Of Learning

The 2 solution components of the learning problem:

- The Hypothesis Set $\mathcal{H}$
- The Learning Algorithm $\mathcal{A}$

Together they are referred to as the learning model



Approximation-generalization trade-off

- The aim of learning is to approximate the target function f as closely as possible.
- Small prediction error: good approximation of the target function f outside of the training set.
- More complex hypothesis set $\mathcal{H} \Rightarrow$ better change of approximating f.
- Less complex hypothesis set $\mathcal{H} \Rightarrow$ better change of generalizing f outside of the training set.
- Having the ideal hypothesis set $\mathcal{H} = \{f\}$ means we already know the answer and there is no need for machine learning.

Grue Emerald Paradox

What if we allow $\mathcal{H}$ to contain all possible functions in the hypothesis set?

Grue Emerald Paradox

Suppose we have seen a large set of emeralds and they are all green. Consider the followig two alternative hypotheses:

- Hypothesis 1: All emeralds are green.
- Hypothesis 2: All emeralds are blue except the ones we have seen so far.

Based on the training set alone, there is no means of choosing which one is better. On the test data, however, they give completely opposite results.

Based on the empirical evidence there is no justification for regarding the evidence as ever confirming one hypothesis more than another one!

This is a paradox" because of course scientists frequently make judgments that the evidence confirms one hypothesis more favorably than another.

Grue Emerald Paradox

Suppose we have seen a large set of emeralds and they are all green. Consider the followig two alternative hypotheses:

- Hypothesis 1: All emeralds are green.
- Hypothesis 2: All emeralds are blue except the ones we have seen so far.

Based on the training set alone, there is no means of choosing which one is better. On the test data, however, they give completely opposite results.

Based on the empirical evidence there is no justification for regarding the evidence as ever confirming one hypothesis more than another one!

This is a paradox" because of course scientists frequently make judgments that the evidence confirms one hypothesis more favorably than another.

$\rightarrow$ A restriction must be placed on the functions that we allow.

Which hypothesis is more plausible? Justification? Occam's razor

Occam's Razor

- Occam's razor is a principle that favors the simplest hypothesis that can well explain a given set of observations
- Given a simple hypothesis A, and a much more complex hypothesis B which explains the existing data only slightly better than A, A is likely to explain future data better than B.
- Works also on hypothesis sets $\mathcal{H}$

Another example: Polynomial Curve Fitting

M^{th} Order Polynomial

$$f_{\mathbf{w}}(x) = w_0 + w_1 x + w_2 x^2 + \ldots + w_M x^M$$

Denoting with

$$\mathbf{w} = (w_0, w_1, \ldots, w_M)$$

the $M + 1$ dimensional vector of polynomial weights, the hypothesis set of all Mth oder polynomials corresponds to the set

$$\mathbb{R}^{M+1}$$

of $M + 1$ dimensional real valued vectors.

Recap: Polynomial Curve Fitting

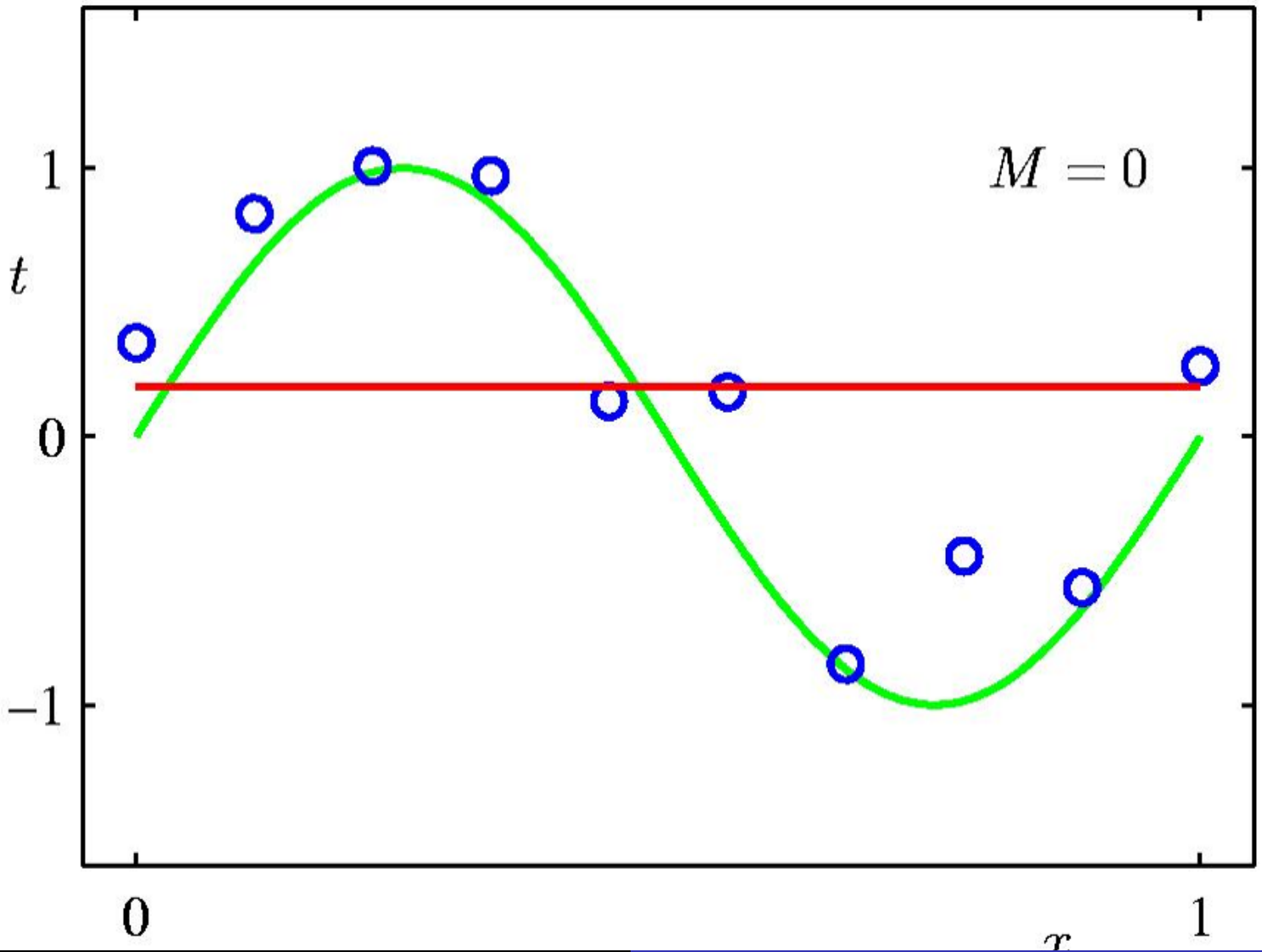
Error on the training set with the mean squared error measure

$$R(f_{\mathbf{w}}) = \frac{1}{N} \sum_{i=1}^{N} \left(f\left(x_i\right) - y_i \right)^2$$

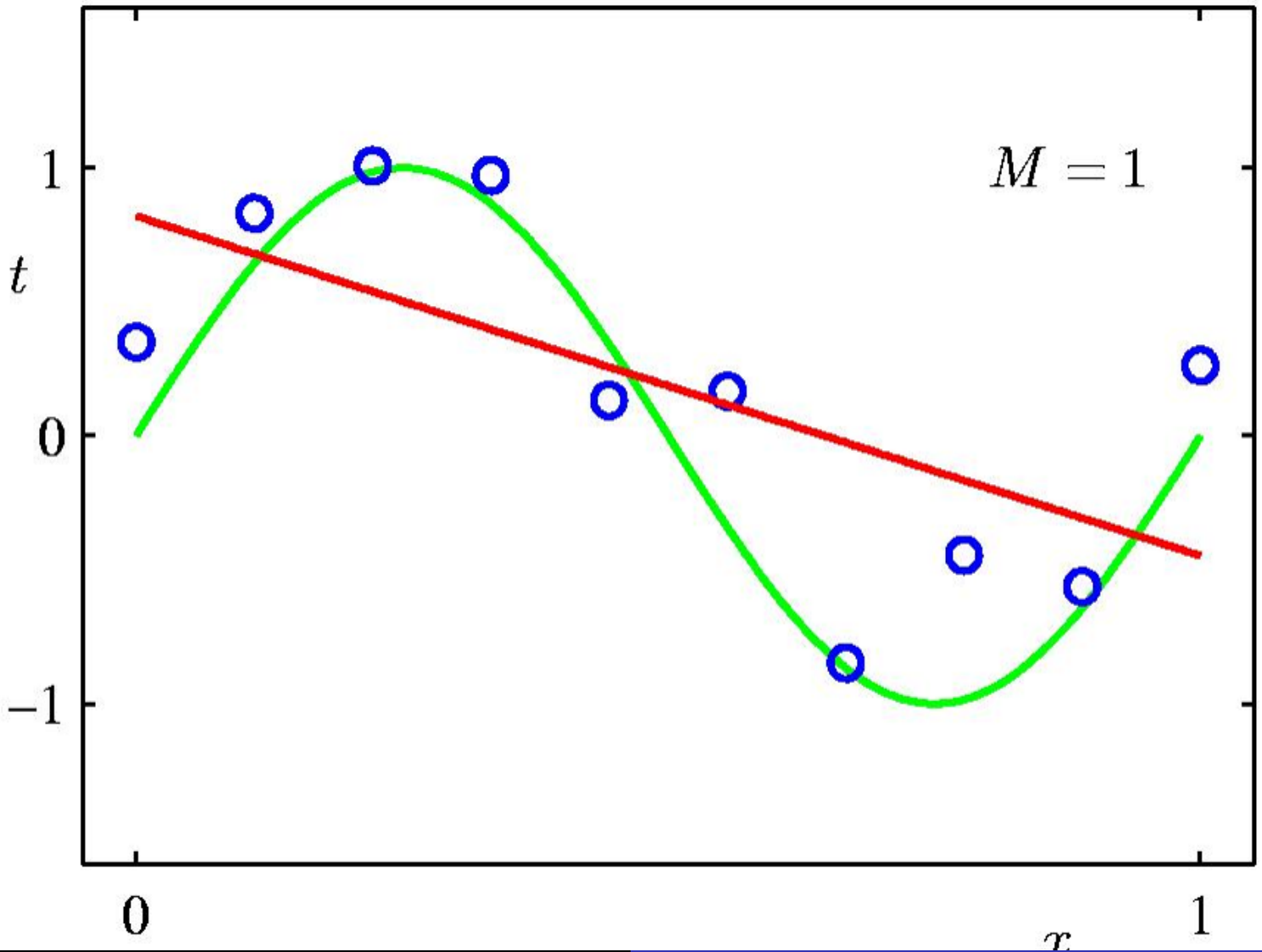
Select the hypothesis (e.g. the polynomial of degree M) with the smallest error:

$$\arg\min_{\mathbf{w} \in \mathbb{R}^{M+1}} R(f_{\mathbf{w}})$$

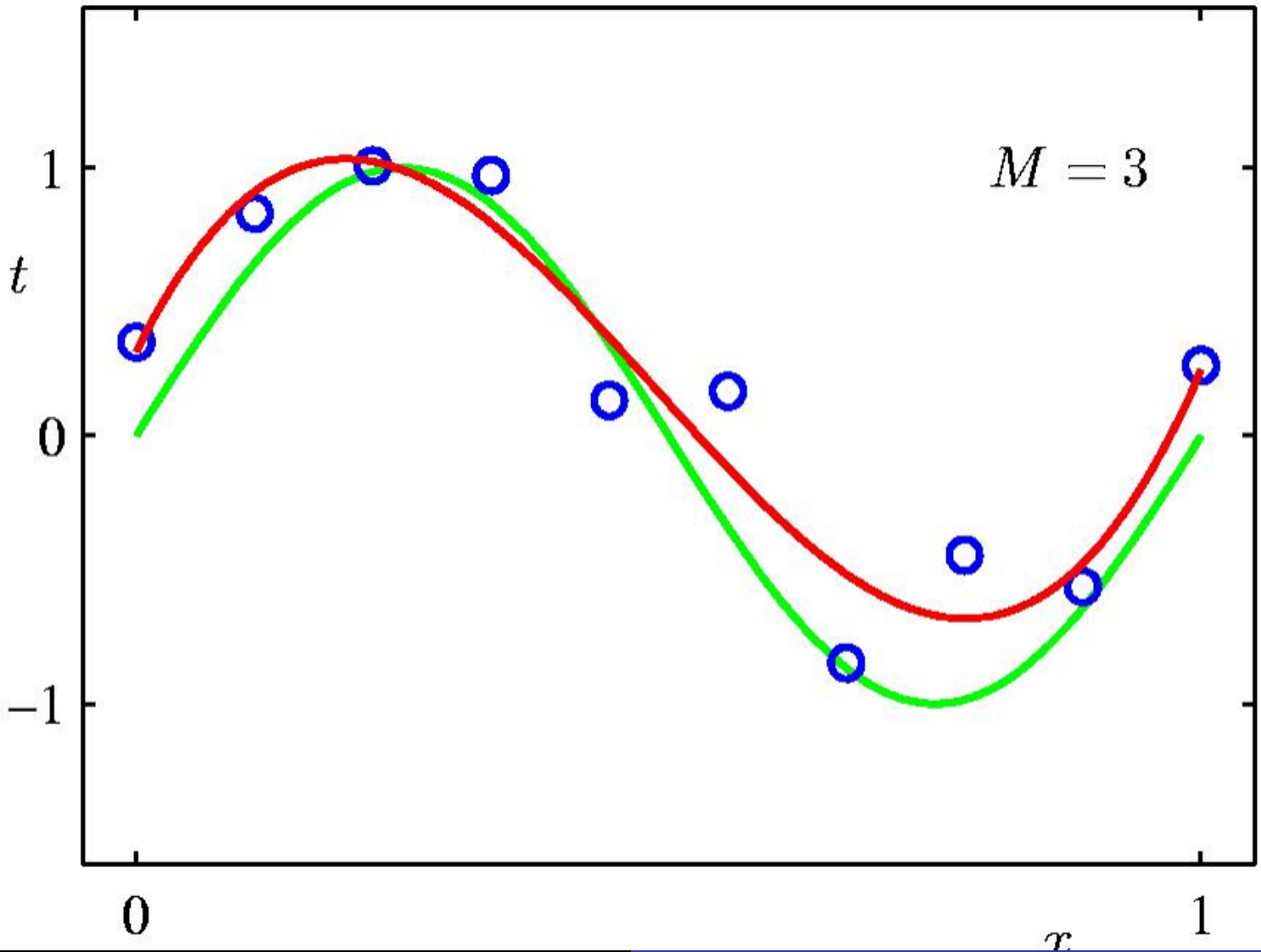
0^{th} Order Polynomial



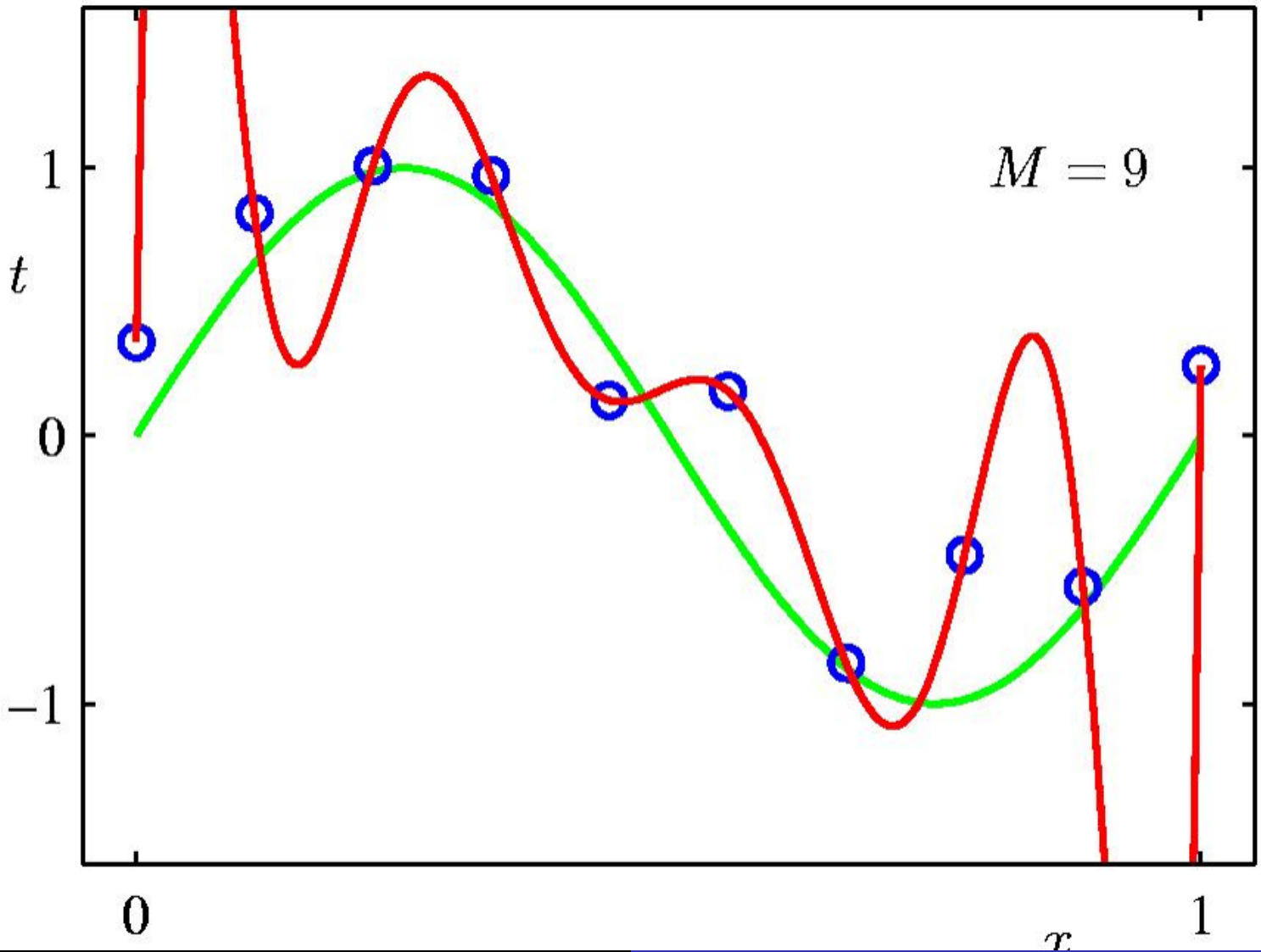
1st Order Polynomial



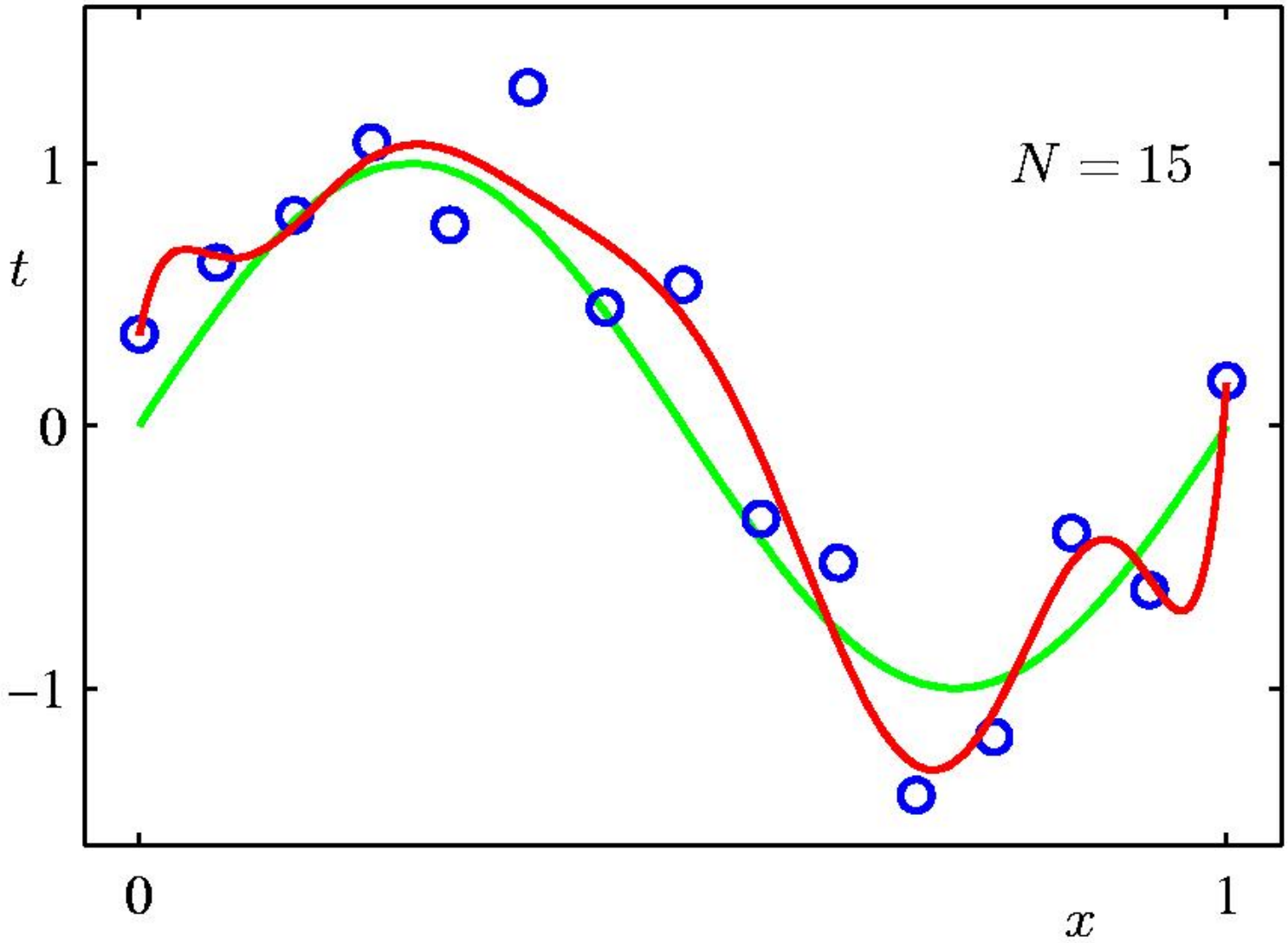
3rd Order Polynomial



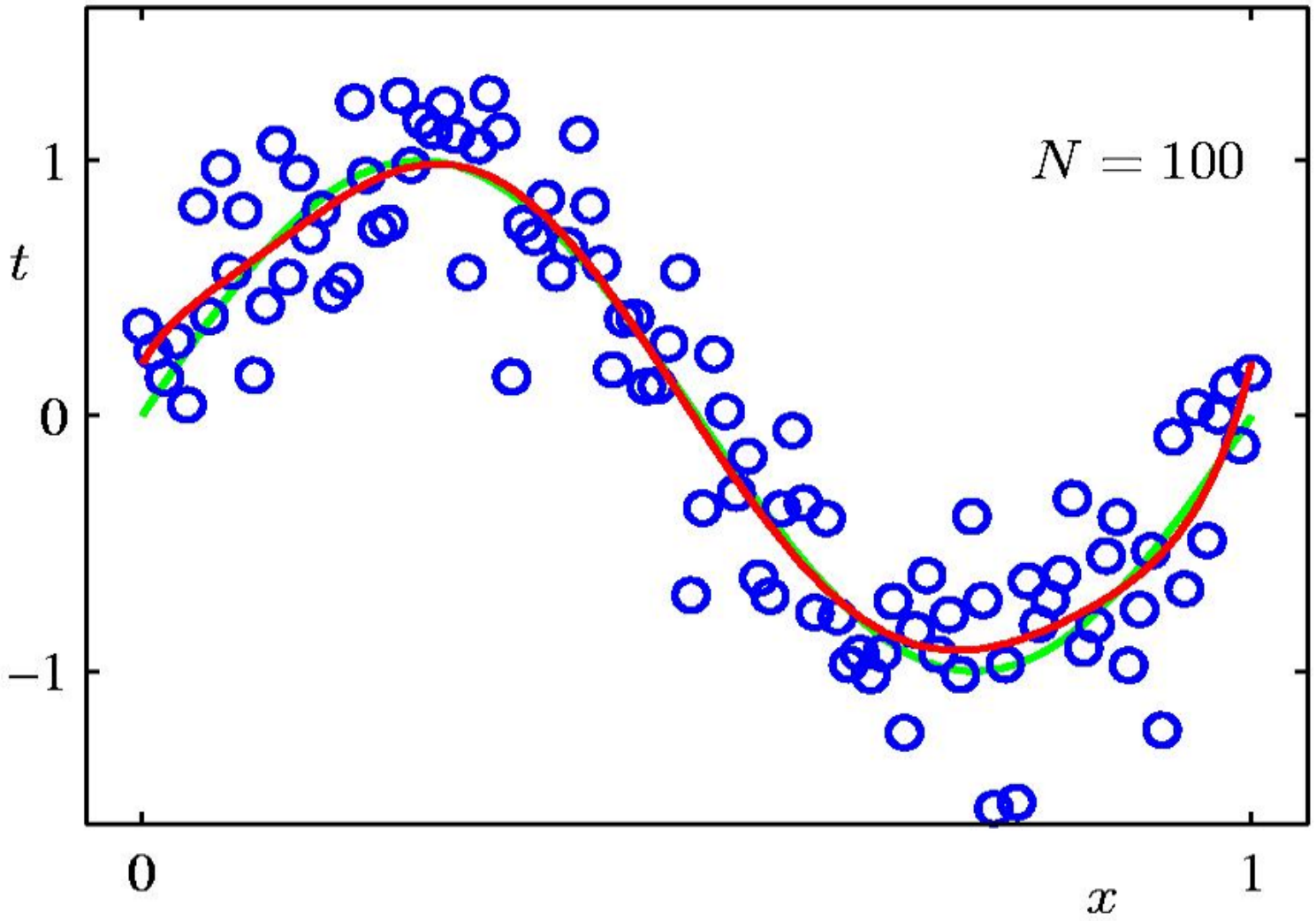
9^{th} Order Polynomial



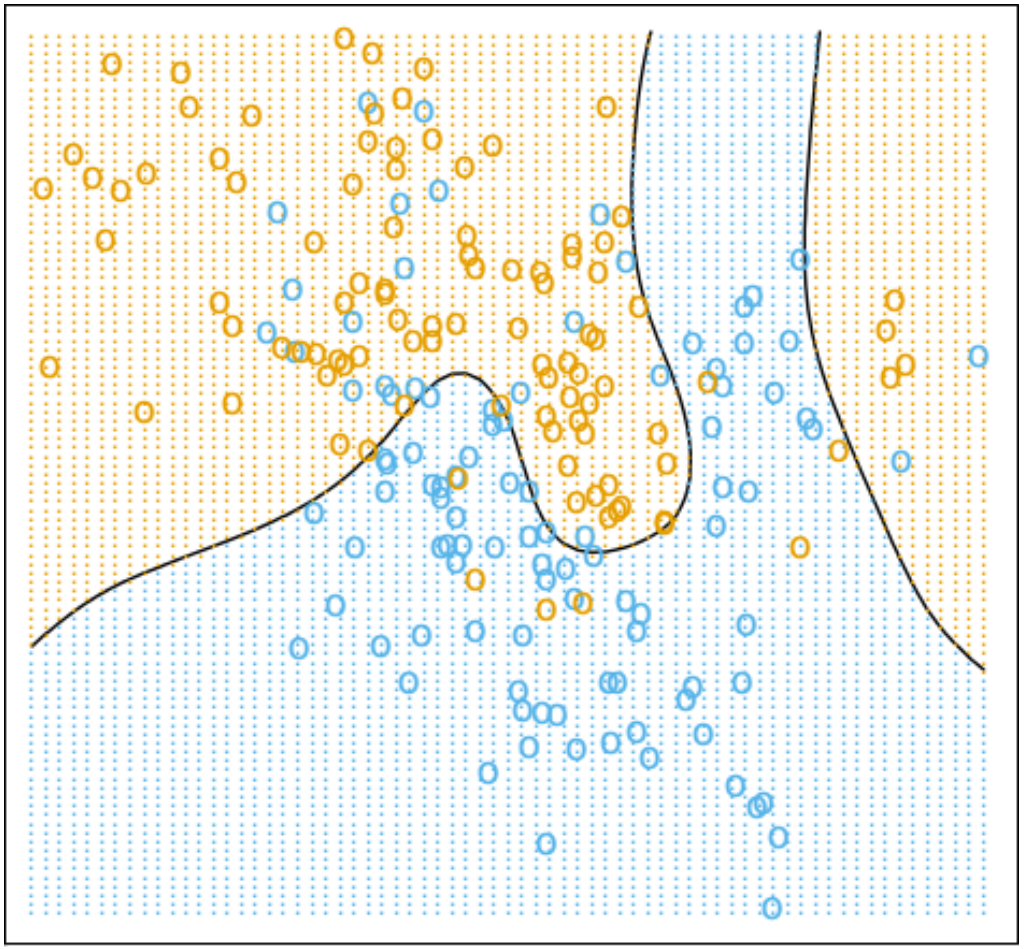
9^{th} Order Polynomia, N=15 data points



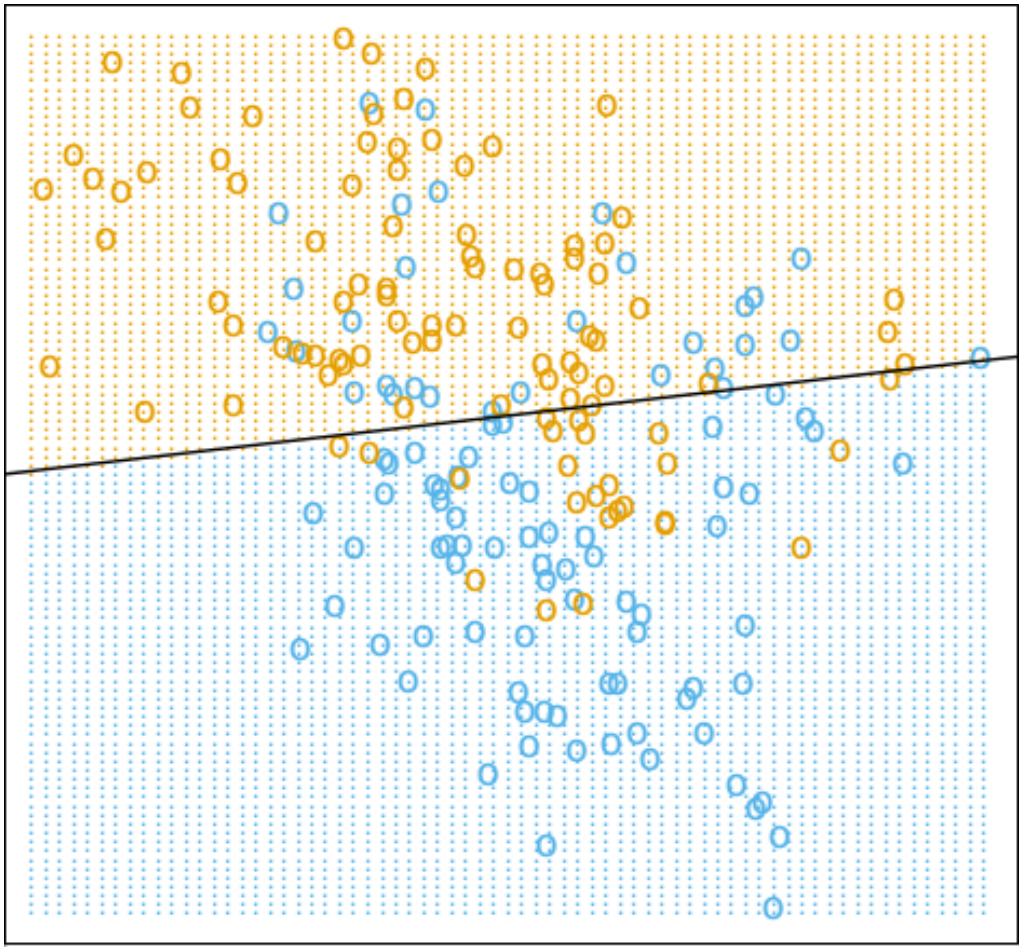
9^{th} Order Polynomia, N=100 data points



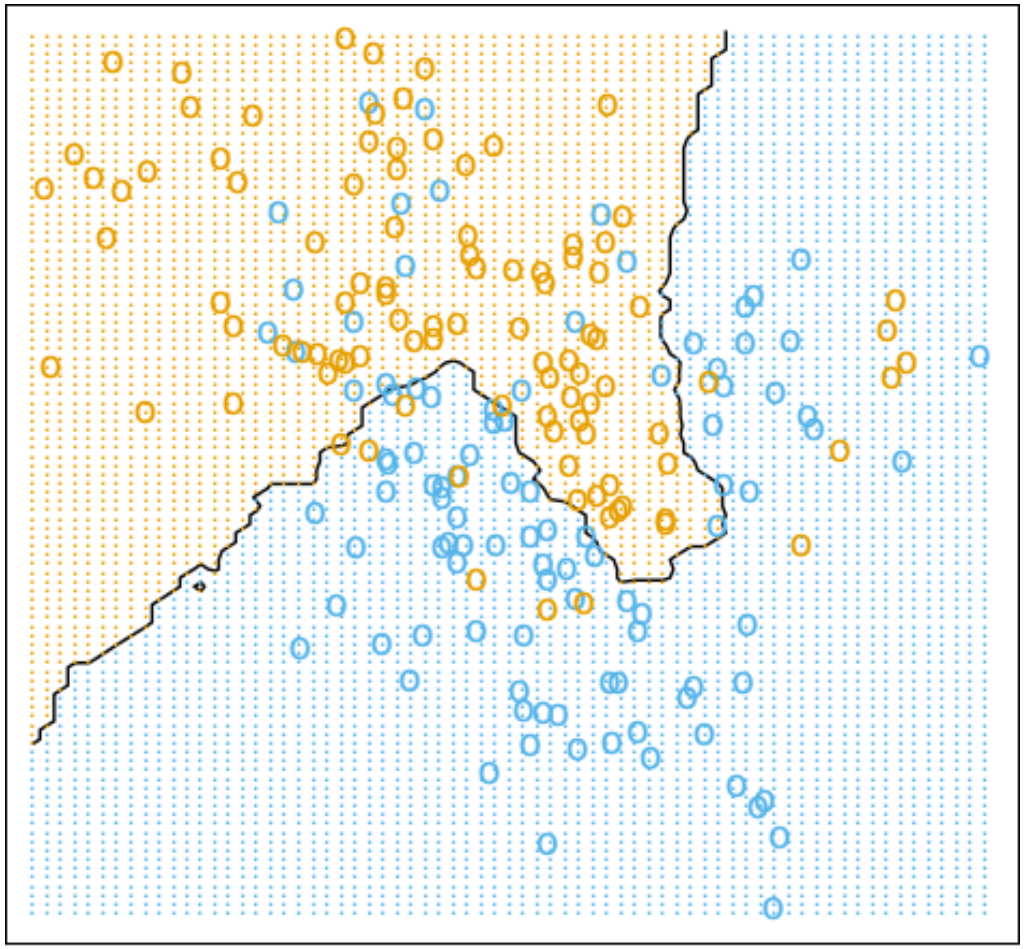
True model



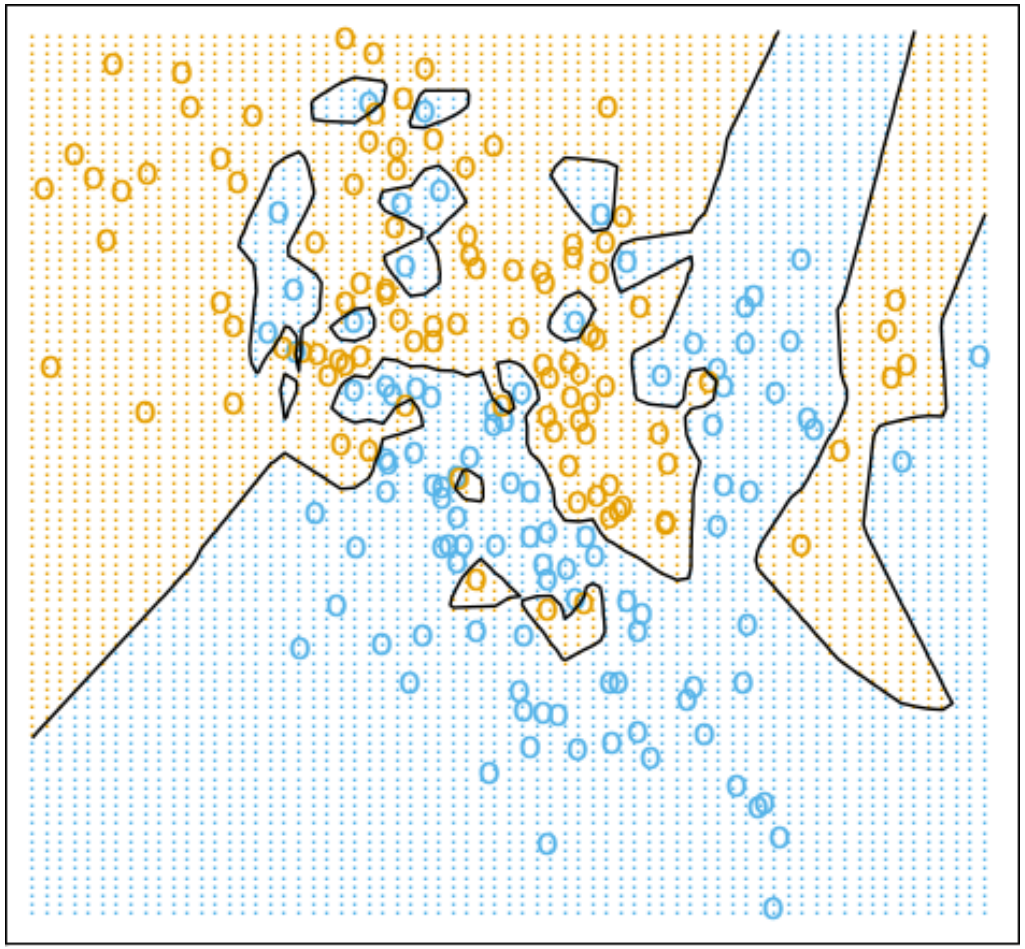
Underfitting (linear model)



Reasonable model (15-nearest neighbour)



Overfitting (1-nearest neighbour)



Moral of the story

- Two criteria need to be balanced in learning, fit to data (low error) and model complexity
- Underfitting: too simple models may not be able to capture the underlying concept well
- Overfitting: too complex models may simply model the noise in the data
- The more data you have, the more complex models you can afford to use
- Many theoretical approaches to defining what we mean by complex: regularization theory, minimum description length principle, Bayesian priors...

Controlling the complexity of $\mathcal{H}$

The complexity of the hypothesis set $\mathcal{H}$ is usually controlled with hyper-parameters.

- The max degree of polynomials on the above regression example
- The number of neighbours in the k-nearest neighbour method
- The regularization parameter value for many methods (regularized linear regression models, support vector machines,...)
- Depth of decision tree
- The number of layers and the number of neurons per hidden layer with neural networks
- You might also be comparing models produced by different algorithms altogether
- ...

Validation

- How to tell which model is the best?
- Simplest idea: Use the training set error.
- This invariably overfits. Don't do this!
- Validation means measuring the predictor's behavior on data points other than those in the training set (often known as the test set)
- One of the simplest and most practical approaches.
- Basic idea of cross-validation: split data randomly to a training and test set, construct model on training set, compute performance (misclassification error / squared error / AUC) on test set, repeat e.g. 10 times, take average value
- Needed when you have so little data that you cannot afford a big separate test set

Validation

- seriously, never ever report your model's performance on its training data (sometimes called the resubstitution error) and claim that this is how well it predicts!
- if you use very restricted and simple models and you have lots of data, you might get from this a reasonable estimate of your true accuracy, but...
- when you have overfitted a complex model to your data, you may get close to zero error, even when the model is no better than random
- (machine) learning is not about memorization, learning is about generalization