Lecture 12: Model Selection

Andreas Wichert
Department of Computer Science and Engineering
Técnico Lisboa

Models

- Which learning models did we already introduce?

- Nonlinear regression, number of basis functions.
  - Example polynomials.
- Nonlinear regression, regularisation parameter

- Stochastic Gradient Descent (Perceptron), regularisation parameter

Overfitting

Consider error of hypothesis h over

- Training data: $error_{train}(h)$
- Entire distribution D of data: $error_D(h)$ *(or on test alone)*

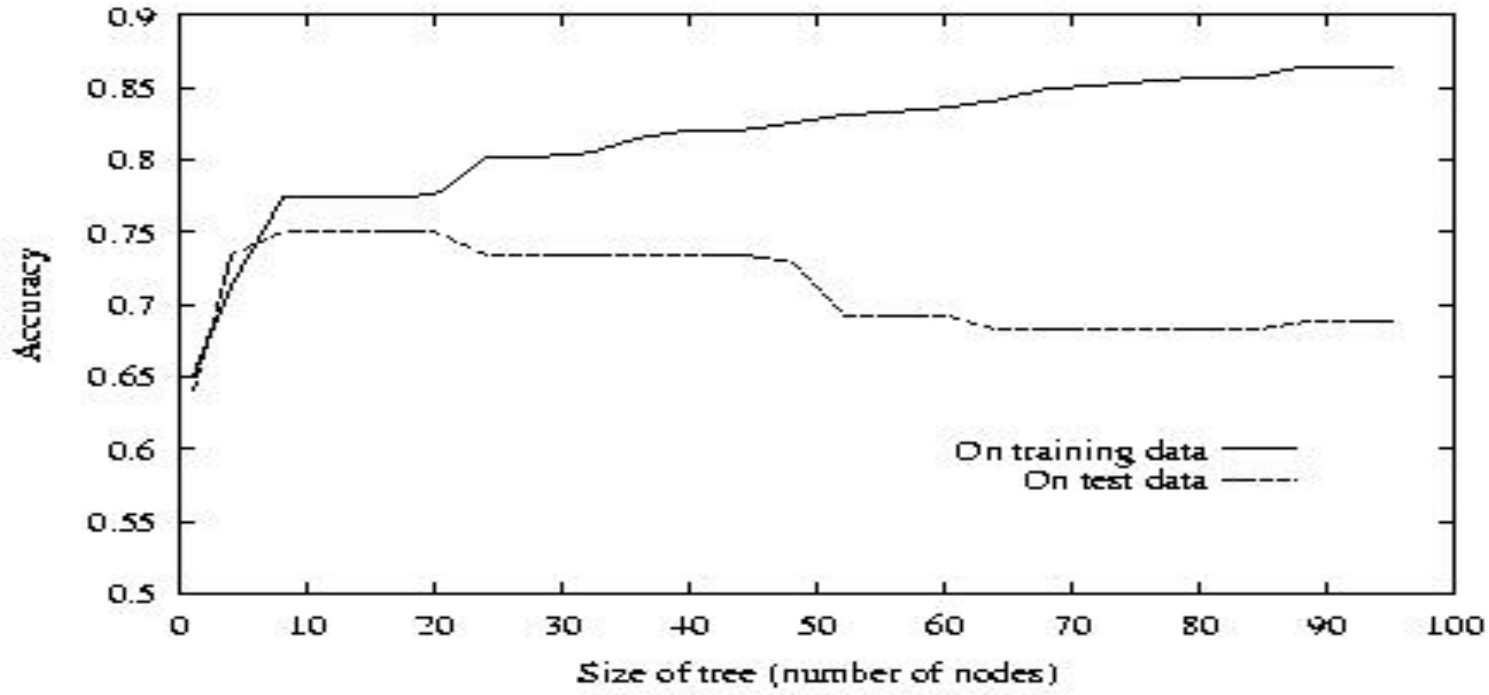
Hypothesis $h \in H$ *overfits* training data if there is an alternative hypothesis $h' \in H$ such that

$$error_{train}(h) < error_{train}(h')$$

and

$$error_D(h) > error_D(h')$$

Overfitting

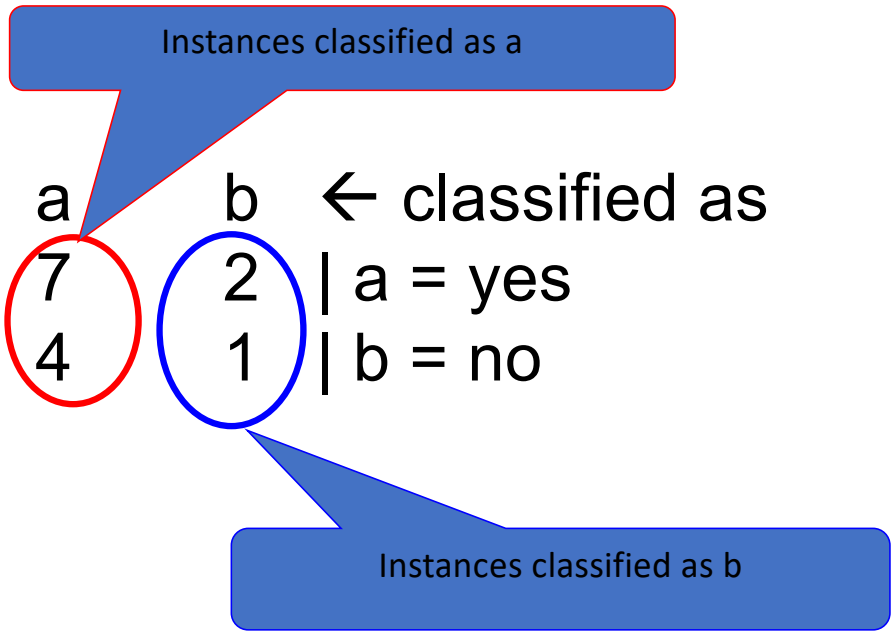


Avoid Overfitting

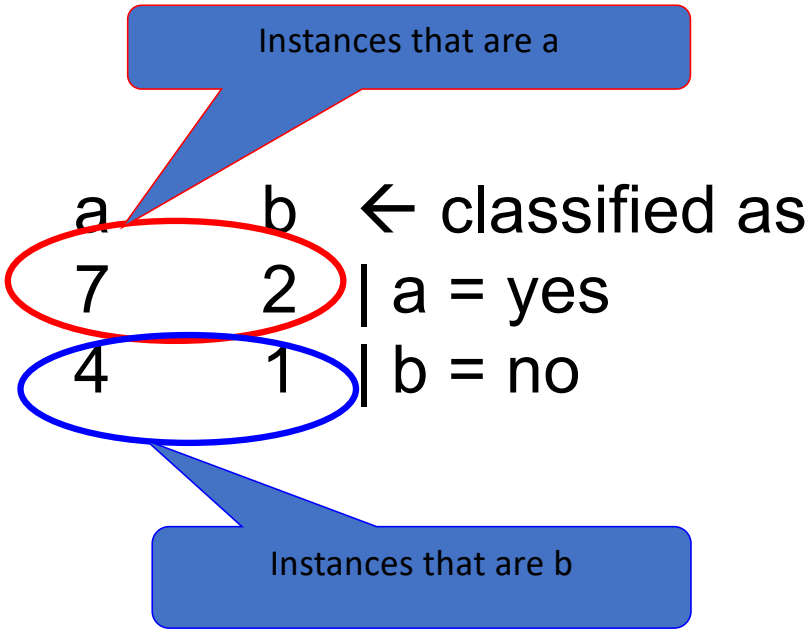
- How can we avoid overfitting?
  - Stop growing when data split not statistically significant
  - Grow full tree then post-prune

- How to select ``best'' model:
  - Measure performance over training data
  - Measure performance over separate validation data set

Evaluation: The confusion matrix



Evaluation: The confusion matrix



Evaluation: The confusion matrix

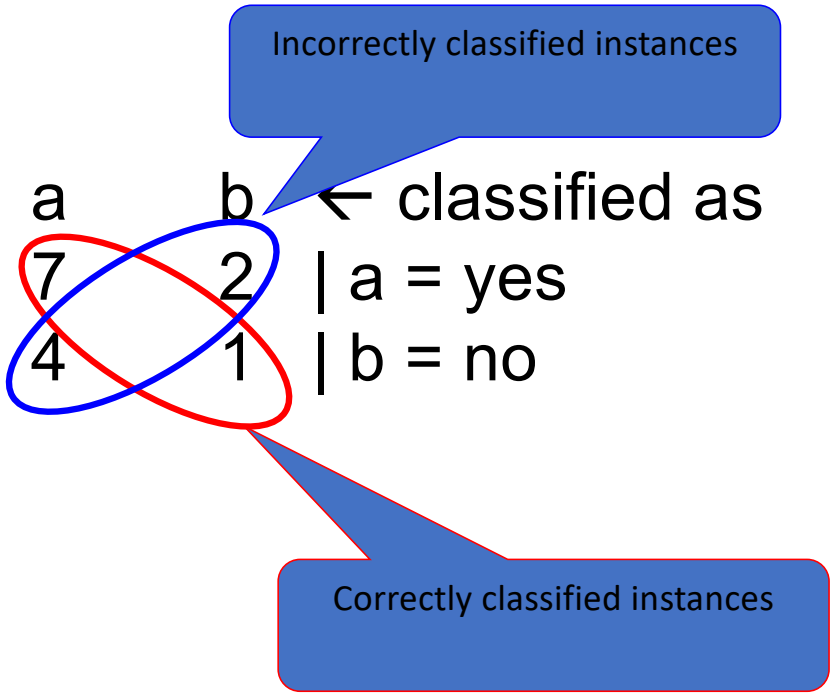


Table 8.1 Confusion matrix for two classes C_1 and *not* C_1. Out of 14 examples $8 = 7 + 1$ were classified correctly and $6 = 4 + 2$ wrong.

	predicted C_1	predicted *not* C_1
actual C_1	7	2
actual *not* C_1	4	1

- Using a confusion matrix for a binary classifier we can define precision and recall with the renaming of the two classes as C1=positive and not C1=negative

Table 8.2 Confusion matrix for for classes positive and negative.

	predicted positive	predicted negative
actual positive	true positive	false negative
actual negative	false positive	true negative

$$Precision = \frac{true\ positive}{true\ positive + false\ positive}$$

$$Recall = \frac{true\ positive}{true\ positive + false\ negative}.$$

- A high recall value without a high precision does not give us any confidence about the quality of the binary classifier.
- We can obtain a high recall value by classifying all patterns as positive (the recall value will be one); however, the precision value will be very low.
- Conversely, by classifying only one pattern correctly as positive, we obtain the maximal precision value of one but a low recall value. Both values have to be simultaneously interpreted.

- To that end, we can combine both values with the harmonic mean

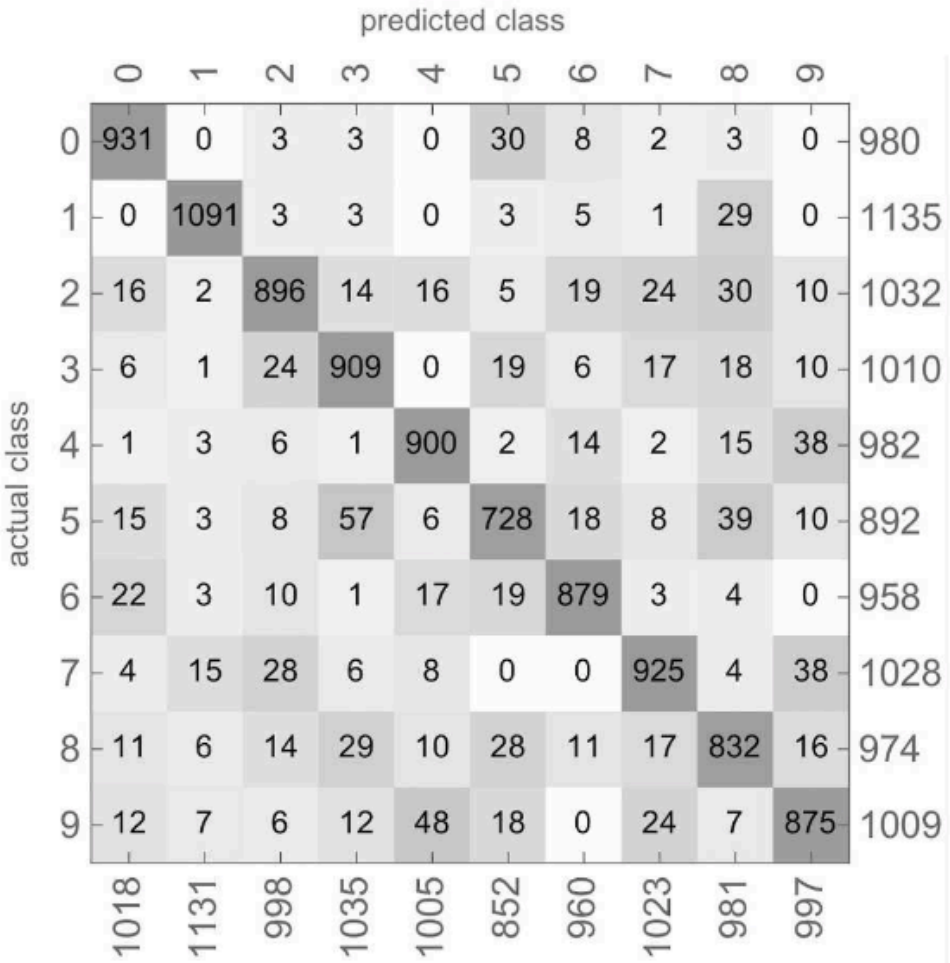
$$F = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$$

in which both values are evenly weighted. This measure is also called the

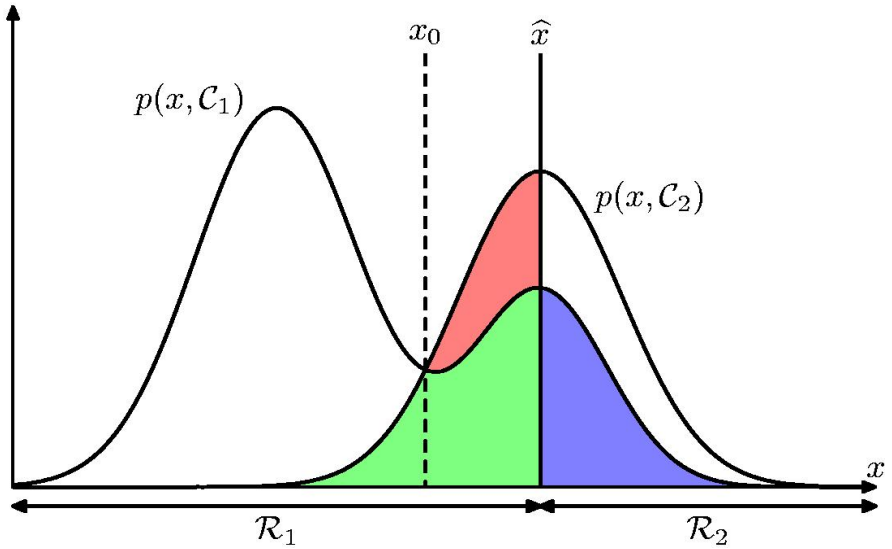
- balanced measure.

Fig. 8.2 Example of MNIST digits represented by gray images of size 28×28.

Error/Confusion Matrix for 10 Classes



Minimum Misclassification Rate



$$
\begin{aligned}
p(\text{mistake}) &= p(\mathbf{x} \in \mathcal{R}_1, \mathcal{C}_2) + p(\mathbf{x} \in \mathcal{R}_2, \mathcal{C}_1) \\
&= \int_{\mathcal{R}_1} p(\mathbf{x}, \mathcal{C}_2)\, \mathrm{d}\mathbf{x} + \int_{\mathcal{R}_2} p(\mathbf{x}, \mathcal{C}_1)\, \mathrm{d}\mathbf{x}.
\end{aligned}
$$

Cross-Validation

- Estimate the accuracy of a hypothesis induced by a supervised learning algorithm
- Predict the accuracy of a hypothesis over future unseen instances
- Select the optimal hypothesis from a given set of alternative hypotheses
  - Model selection
  - Feature selection
- Combining multiple classifiers (boosting)

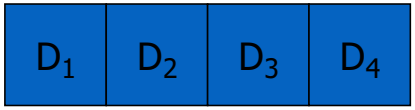
Holdout Method

- Partition data set $D = \{(v_1,y_1),\ldots,(v_n,y_n)\}$ into *training* D_t and *validation* set $D_h = D \setminus D_t$

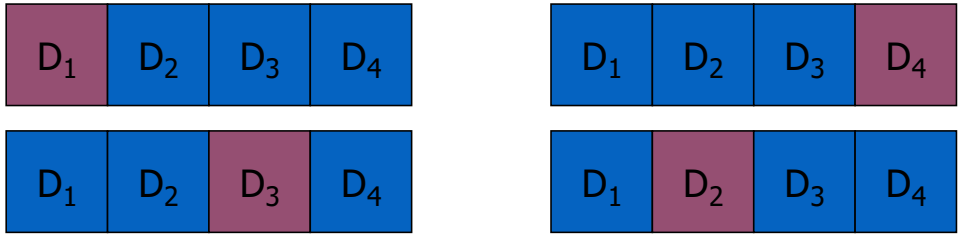
Problems:
- makes insufficient use of data
- training and validation set are correlated

Cross-Validation

- k-fold cross-validation splits the data set D into k mutually exclusive subsets $D_1, D_2, \ldots, D_k$

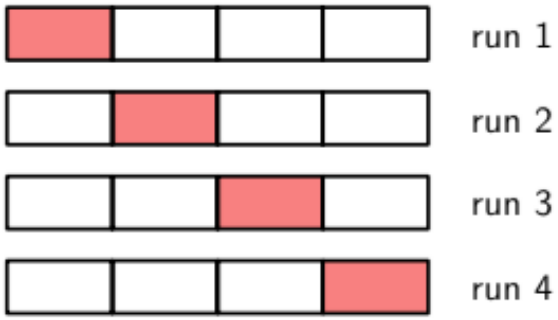


- Train and test the learning algorithm k times, each time it is trained on $D \backslash D_i$ and tested on D_i



Cross-Validation

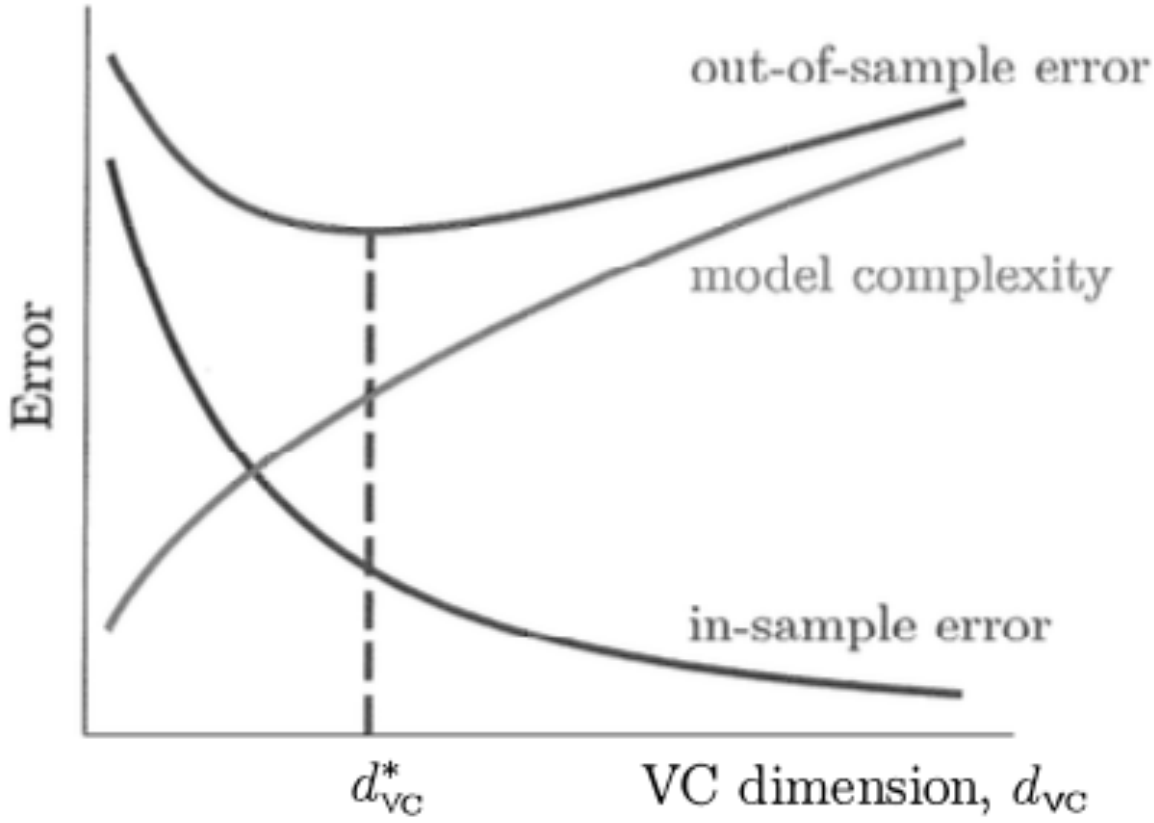
- Uses all the data for training and testing
- Complete *k*-fold cross-validation splits the dataset of size m in all (m over *m/k*) possible ways (choosing *m/k* instances out of *m*)
- Leave *n*-out cross-validation sets *n* instances aside for testing and uses the remaining ones for training (leave one-out is equivalent to n-fold cross-validation)
- In stratified cross-validation, the folds are stratified so that they contain approximately the same proportion of labels as the original data set



- One major drawback of cross-validation is that the number of training runs that must be performed is increased by a factor of *k*
- Cross-validation that use separate data to assess performance is that we might have **multiple** complexity **parameters** for a single model (for instance, there might be several regularization parameters).

Model Selection

- Using the Bayesian approach or the regularised method of least squares
- The estimation is statistical, we need an *appropriate measure* of the fit between the model and the observed data
- We refer to this problem as that of model selection
- For example, we may want to estimate the number of degrees of freedom (i.e., adjustable parameters) of the model, or even the general structure of the model
  - Nonlinear regression, number of basis functions. Example polynomials.


out-of-sample error
model complexity
in-sample error
Error
d^*_{VC}
VC dimension, d_{VC}

Occam's razor

- Occam Razor was first articulated by the medieval logician William of Occam in 1324
  - born in the village of Ockham in Surrey (England) about 1285, believed that he died in a convent in Munich in 1349, a victim of the Black Death
  - It is vain do with more what can be done with less..
- We should always accept the simplest answer that correctly fits our data

- Occam's razor: **All other things being equal, the simplest model is the best**
  - A good principle for life as well

William of Ockham

Minimum-Description-Length (MDL)

- Minimum-Description-Length (MDL) principle
- Inspiration for the development of the MDL principle is traced back to Kolmogorov complexity theory
- Find the shortest program that produces the data (uncomputable).

Minimum-Description-Length (MDL)

- The algorithmic (descriptive) complexity of a data sequence is the length of the shortest binary computer program that prints out the sequence and then halts
- Definition of complexity that looks to the computer, the most general form of data compressor, rather than the notion of probability distribution for its basis
- The goal of which is to find regularity in a given data sequence

Learning as Data Compression

- The idea of viewing learning as trying to find regularity provided the first insight that was used by Rissanen in formulating the MDL principle.
- The second insight used by Rissanen is that regularity itself may be identified with the ability to compress

- View the process of l**earning** as data **compression**

- Jorma J. Rissanen (born 20 October 1932) is an information theorist, known for inventing the minimum description length principle and practical approaches to arithmetic coding for lossless data compression

MDL and Machine Learning

- Why does the shorter encoding make sense?

- Shorter encoding implies regularities in the data
- Regularities in the data imply patterns
- Patterns are interesting

Fig. 8.5 Andrey Nikolaevich Kolmogorov in Paris 1958.

- Inspiration for the development of the Minimum-Description-Length (MDL) principle is traced back to Kolmogorov complexity theory
- The basic idea is to try to find the shortest program that produces some data.

- The algorithmic (descriptive) complexity of a data sequence is the length of the shortest binary computer program that prints out the sequence and then halts.
- This definition of complexity uses the computer that is the most general form of data compressor, rather than being based on the notion of probability distribution.
- When there are regularities in the data sequence it can be produced by a simpler program.

Example: Regularity

000010000100001000010000100001000010000100010000100001

Short description length, just repeat 12 times 00001

010011100101001101101010000111010111101101101011100100111 00

Random sequence, no patterns, no compression

- Shorter encoding implies regularities in the data.
- Regularities in the data imply patterns.
- Patterns are interesting.

0100111001010011011010100001110101111011011010101110010011100.

- This string is basically a random sequence and since no patterns can be found no compression can be achieved.

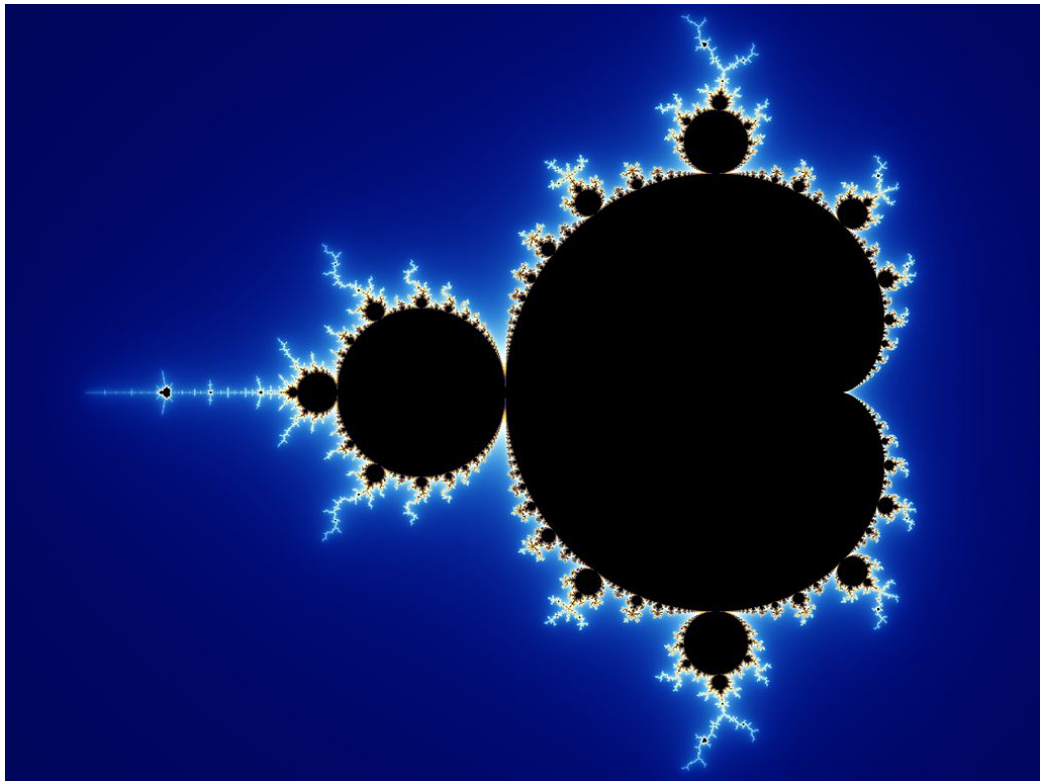
- We claimed that any regularity detected in the data can be used to compress the data
- Describe it in a short manner

- Example:
- Pi, e,..
- The Mandelbrot set is the set of complex numbers for which the function

$$f_c(z) = z^2 + c$$

does not diverge when iterated from *z=0*

Mandelbrot set



Two-part code MDL principle

- The simplistic two-part code MDL principle for probabilistic modelling is simplistic: the codelengths under consideration are not determined in an optimal fashion
- Suppose that we are given a candidate model or model class M
- With all the elements of M being probabilistic sources, we henceforth refer to a point hypothesis as p rather than H
- In particular, we look for the probability density function $p \in M$ that best explains a given data sequence d

Two-part code MDL principle

- The two-part code MDL principle then tells us to look for the (point) hypothesis $p \in M$ that minimizes the description length of p, which we denote by $L_1(p)$, and the description length of the data sequence d when it is encoded with the help of p, which we denote as $L_2(d|p)$.

$$L_{1,2} = L_1(p) + L_2(d|p)$$

Chose that hypothesis $p \in M$ that minimizes $L_{1,2}$

It is crucial that p itself be encoded as well here

- In finding the hypothesis that compresses the data sequence *d* the most, we must encode (describe or compress) the data in such a way that a decoder can retrieve the data even without knowing the hypothesis in advance

The model-order selection problem

Let $M_1, M_2, \cdots, M_k, \cdots$ denote a family of linear regression models that are associated with the parameter vector $\mathbf{w}^{(k)} \in W_k$

The model order $k = 1, 2, \cdots$ weight spaces $W_1, W_2, \cdots W_k, \cdots$ is of increasing dimensionality

Identify the model that best explains an unknown environment that is responsible for generating the training sample $(\mathbf{x}_\eta, t_\eta)$ for $\eta \in (1, 2, \cdots, N)$

In working through the statistical characterization of the composite length $L_{12}(p, d)$, the two-part code MDL principle tells us to pick the kth model that is the mimimizer

$$\min_k \left\{ \left(\frac{2}{k}(N) + O(k) \right) + \left(-\log(p(t_\eta | \mathbf{w}^{(k)}) \cdot p(\mathbf{w}^{(k)}) \right) \right\}$$

$$\min_k \left\{ Complexity\ term(N, k) + Error\ term^{(k)} \right\}$$

For large sample size N, $O(k)$ gets overwhelmed by $\frac{2}{k}(N)$

$$\min_{k} \left\{ \left(\frac{2}{k}(N) + O(k) \right) + \left(-\log(p(t_\eta|\mathbf{w}^{(k)}) \cdot p(\mathbf{w}^{(k)}) \right) \right\}$$

the error term, denoted by $-\log(p(t_\eta|\mathbf{w}^{(k)}) \cdot p(\mathbf{w}^{(k)})$ which relates to the model and the data;

the hypothesis complexity term, denoted by $\frac{2}{k}(N) + O(k)$, which relates to the model alone.

In practice, the $O(k)$ term is often ignored to simplify matters

with

$$y(\mathbf{x}, \mathbf{w}) = \sum_{j=0}^{M-1} w_j \cdot \phi_j(\mathbf{x}) = \langle \mathbf{w} | \Phi(\mathbf{x}) \rangle = \mathbf{w}^T \Phi(\mathbf{x})$$

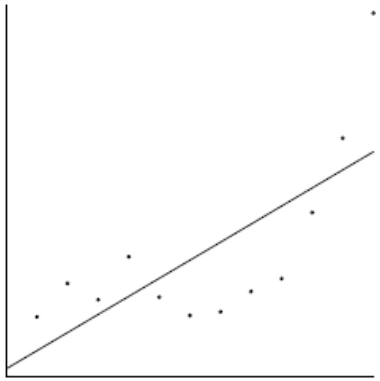
we get

$$L_{1,2} = \arg\min_{\Phi} \left\{ L(\Phi) + \sum_{\eta=1}^{N} (t_\eta - \mathbf{w}^T \cdot \Phi(\mathbf{x}_\eta))^2 + \frac{\lambda}{2} \|\mathbf{w}\|_2^2 \right\}$$

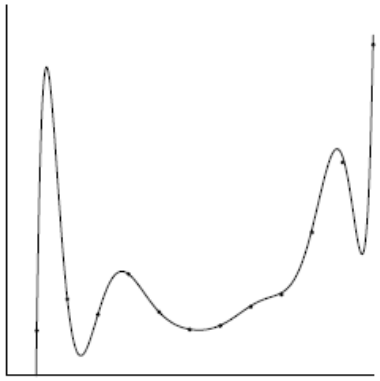
$$L(\Phi) = M \cdot B \ bits$$

Example

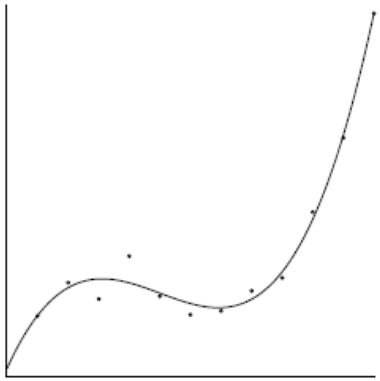
- Regression: find the polynomial for describing the data
- Complexity of the model vs. Goodness of fit



Low model cost
High data cost



High model cost
Low data cost



Low model cost
Low data cost

Attributes of the MDL Principle

- When we have two models that fit a given data sequence equally well, the MDL principle will pick the one that is the simplest in the sense that it allows the use of a shorter description of the data
  - MDL principle implements a precise form of Occam's razor, which states a preference for simple theories

- The MDL principle is a consistent model selection estimator in the sense that it converges to the true model order as the sample size increases

MDL and Regularization

If we use just likelihood

$$\min_k \left\{ \left(\frac{2}{k}(N) + O(k) \right) + \left(-\log(p(t_\eta | \mathbf{w}^{(k)}) \right) \right\}$$

with

$$y(\mathbf{x}, \mathbf{w}) = \sum_{j=0}^{M-1} w_j \cdot \phi_j(\mathbf{x}) = \langle \mathbf{w} | \Phi(\mathbf{x}) \rangle = \mathbf{w}^T \Phi(\mathbf{x})$$

we get

$$L_{1,2} = \arg\min_{\Phi} \left\{ L(\Phi) + \sum_{\eta=1}^{N} (t_\eta - \mathbf{w}^T \cdot \Phi(\mathbf{x}_\eta))^2 \right\}$$

$$L(\Phi) = M \cdot B \; bits$$

This representation discards the true complexity

- Two polynomials with the same number of weights can be described by the code having different length

- Smooth functions are more easy to predict, which suggests adapting description lengths reflecting this property

For simplicity we assume single dimensional input $D = 1$ with

$$\phi(x) = \sum_{j=0}^{M-1} w_j \cdot x^j$$

$$\phi'(x) = \sum_{j=0}^{M-1} j \cdot w_j \cdot x^{j-1}$$

$$L(\Phi) = \|\Phi\|^2 = \int \left(\Phi'(x)\right)^2 dx = \sum_{j=0}^{M-1} \sum_{i=0}^{M-1} j \cdot i \cdot w_j w_i \int \cdot x^{j-1} \cdot x^{i-1} dx$$

which is a seminorm that is small for smooth functions.
A seminorm, on the other hand, is allowed to assign zero length to some non-zero vectors with $|\Phi\|^2 = 0$ for a constant function

With

$$p_{ji} := j \cdot i \cdot \int \cdot x^{j-1} \cdot x^{i-1} dx$$

$$L(\Phi) = \|\Phi\|^2 = \sum_{j=0}^{M-1} \sum_{i=0}^{M-1} p_{ji} w_j w_i = \|\mathbf{w}\|_p^2$$

We can generalise to

$$L(\Phi) = \|\mathbf{w}\|^p = \left(\sum_{i=0}^{M-1} |w_i|^p \right)^{1/p}$$

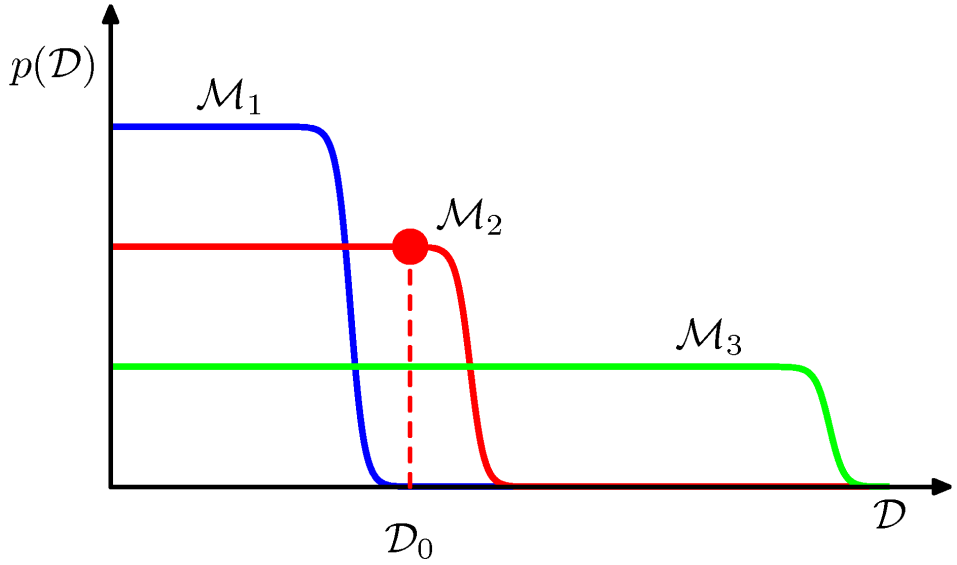
For $p = 2$ we get the l_2 regularisation as before and for $p = 1$ l_1 regularisation

For l_1 regularisation we get sparse representation.

See Section 2.4 Marco Gori, Machine Learning: A Constraint-Based Approach, Morgan Kaufmann, 2017

Bayesian Model Comparison

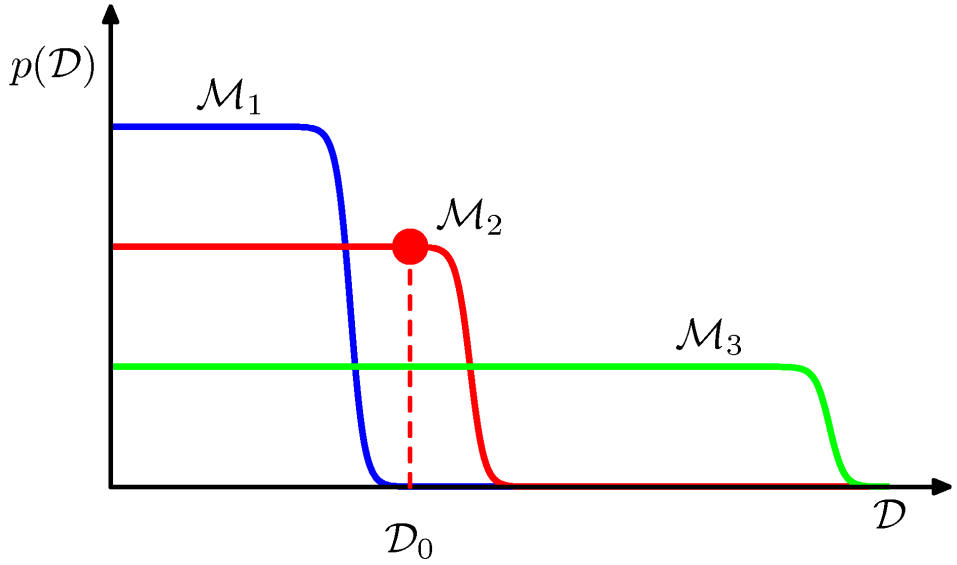
- Matching data and model complexity



- The horizontal axis is a one-dimensional representation of the space of possible data sets, so that each point on this axis corresponds to a specific data set
- A simple model preforms better on simple task then a complex model
- By contrast, a complex model (such as a ninth order polynomial) can generate a great variety of different data sets but preforms worse on simple task
- We now consider three models M_1, M_2 and M_3 of successively increasing complexity

Bayesian Model Comparison

- Matching data and model complexity



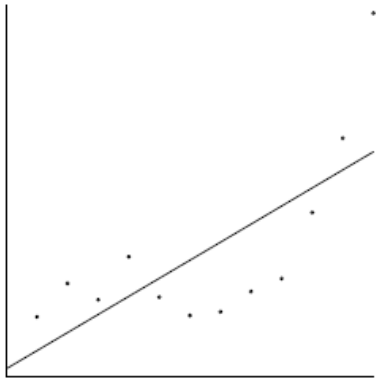
Did we learn something new?

Fig. 8.8 (a) Socrates (470 - 399 BC), after a marble, after roman artwork (1st century), perhaps a copy of a lost bronze statue made by Lysippos. (b) Aristotle (384 - 322 B.C), after roman artwork after a statue made by Lysippos (Lysippos was a Greek sculptor of the 4th century BC).

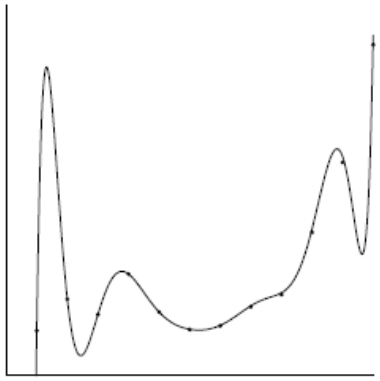
- In ancient Greek philosophy, especially that of *Aristotle*, **the golden mean or golden middle way is the desirable middle between two extremes**, one of excess and the other of deficiency
- *Socrates* teaches that a man must know how **to choose the mean and avoid the extremes on either side**, as far as possible

Example

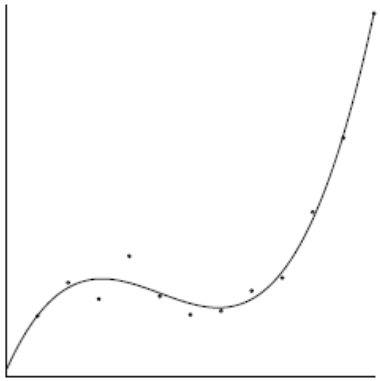
- Regression: find the polynomial for describing the data
- Complexity of the model vs. Goodness of fit



Low model cost
High data cost



High model cost
Low data cost



Low model cost
Low data cost

Paradox of Deep Learning Complexity

- As we have stated before and will see in the next chapters, in deep learning we increase the complexity of the model by using many hidden layers with thousands (or even millions) of parameters
- However, it would seem that by using so many parameters, we would be increasing a lot the model complexity because we need to code all of these parameter values.
- If that is the case we are in direct contradiction of the principle of Parsimony

- Assuming that the size of the numbers contained in parameter wj is correlated to the coding costs

$$w_j \approx w_j \; bits,$$

- w = 0 would require zero bits and a small number would be coded with less bits than a big number.
- Then, the hypothesis complexity term corresponds to the l1 regularization

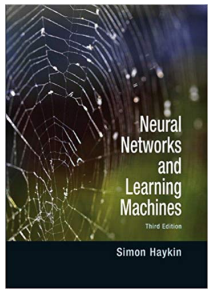
$$L(\mathbf{w}) = |\mathbf{w}| \cdot B \; bits = \|\mathbf{w}\|_1 \cdot B \; bits.$$

- Similarly, for l2 regularization we assume

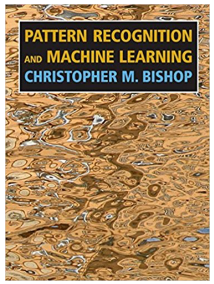
$$L(\mathbf{w})| = \|\mathbf{w}\|_2^2 \cdot B \; bits.$$

- According to this reasoning, the relation between the hypothesis complexity before learning and after learning changes.
- The complexity of a deep learning **model before learning is tremendous**, yet, **after learning it is reduced considerably** due to the use of regularization.
- Depending on the type of regularization (l2 or l1) most weights will have small values or become zero.

Literature

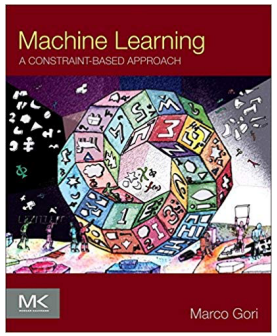


- Simon O. Haykin, Neural Networks and Learning Machine, (3rd Edition), Pearson 2008
  - Chapter 2, Section 2.6

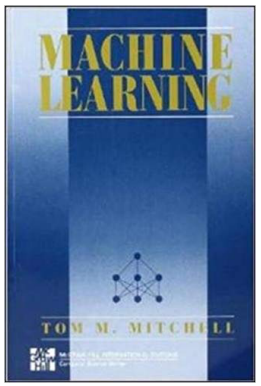


- Christopher M. Bishop, Pattern Recognition and Machine Learning (Information Science and Statistics), Springer 2006
  - Chapter 1 Section 1.3, Chapter 3, Section 3.4

Literature (Additional)

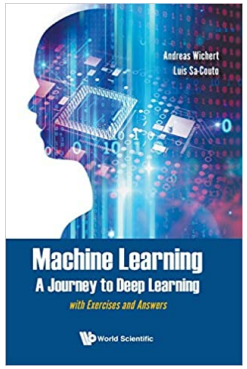


- Marco Gori, Machine Learning: A Constraint-Based Approach, Morgan Kaufmann, 2017
  - Section 2.4



- Tom M. Mitchell, Machine Learning, McGraw-Hill; 1st edition (October 1, 1997)
  - Section 3.7

Literature



- Machine Learning - A Journey to Deep Learning, A. Wichert, Luis Sa-Couto, World Scientific, 2021
  - Chapter 8