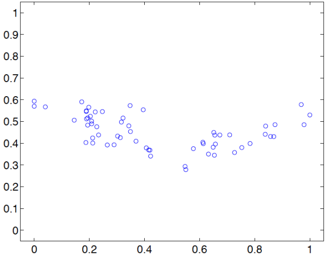
New trend: Deep Learning

1. Neural Networks with many hidden layers (*deep networks*)
    1. Insufficient depth can hurt
    2. The brain has a deep architecture
    3. Cognitive processes seem deep
2. Networks with probabilistic interpretation (Boltzmann Machines)
3. Use of unsupervised learning (autoencoders) for incremental training (one layer at a time)
4. Brute force training using GPUs

Some practical issues

Underfitting/Overfitting and learning parameters

- Suppose we have some data (60 points) that we want to fit a curve to



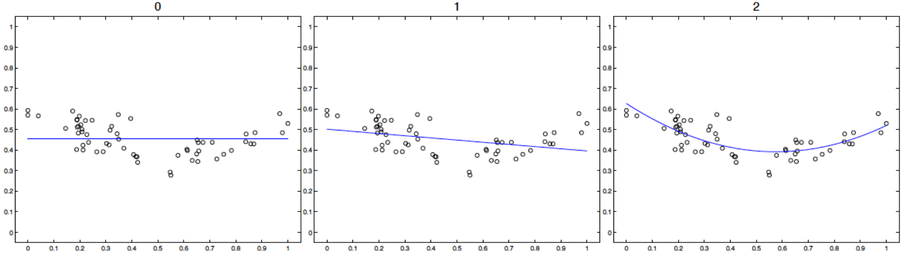
- Let fit a polynomial, of the form

$$y = w_0 + w_1 x + w_2 x^2 + \dots + w_p x^p$$

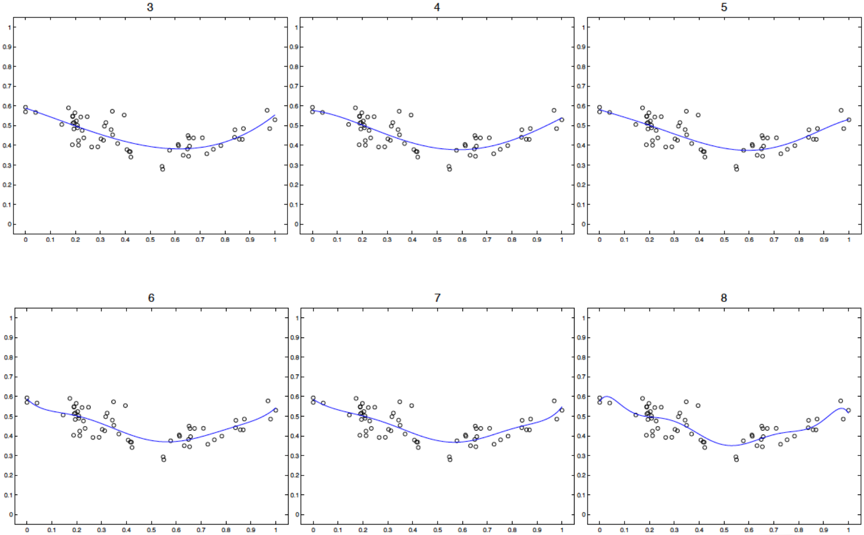
Some practical issues

Underfitting/Overfitting and learning parameters

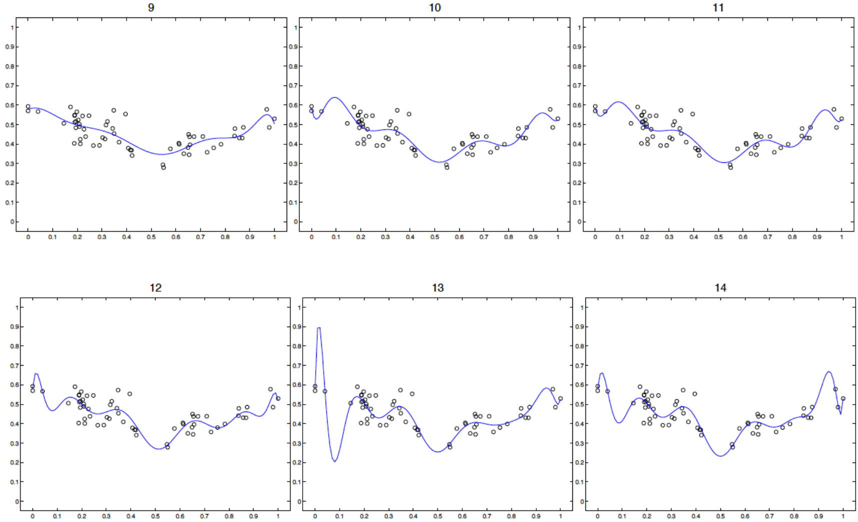
- How to choose p ? (Hypothesis Space)
- For various p, we can find and plot the best polynomial, in terms of minimizing the Empirical Error
- Here are the solutions for different values of p



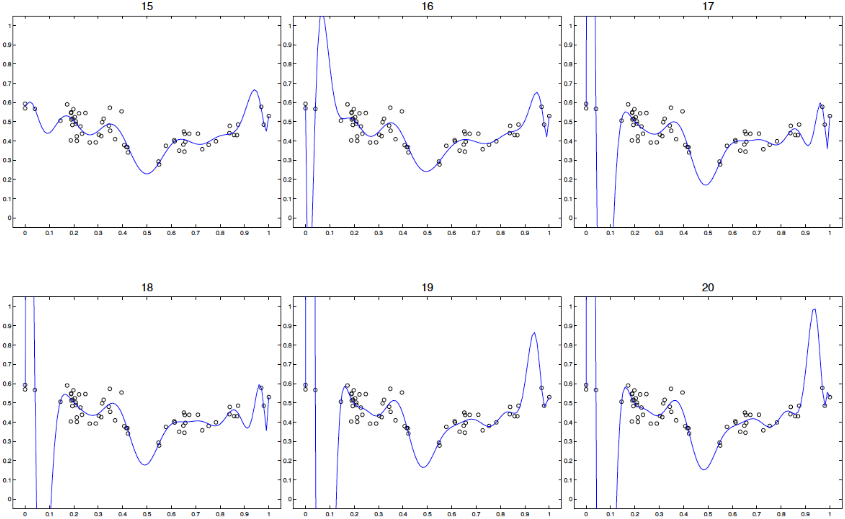
Some practical issues



Some practical issues

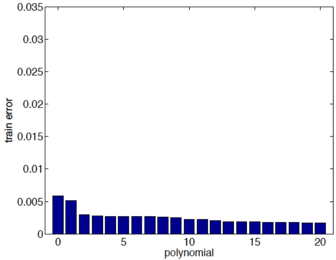


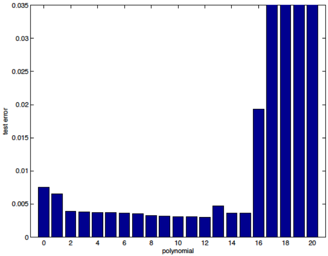
Some practical issues



Underfitting/Overfitting and learning parameters

- Here is a summary of the Empirical Error ... and the Empirical Error over some new TEST data (100,000 extra points) from the same distribution, as a function of p:





Some practical issues

Underfitting/Overfitting and learning parameters

- For very low p, the model is very simple, and so cannot capture the full complexities of the data (Underfitting! also called **bias**)
- For very high p, the model is complex, and so tends to overfit to spurious properties of the data (Overfitting! also called **variance**)

Unfortunately we cannot use the test set to pick up the right value of p!

PRACTICAL PROBLEM: how can we use the training set to set p ?

Model Selection and Hold-out

We can hold out some of our original training data

Hold-out procedure

1. A small subset of Tr, called the validation set (or hold-out set), denoted Va, is identified
2. A classifier/regressor is learnt using examples in $Tr - Va$
3. Step 2 is performed with different values of the parameter(s) (in our example, p), and tested against the hold-out sample

In an operational setting, after parameter optimization, one typically re-trains the classifier on the entire training corpus, in order to boost effectiveness (debatable step!)

It is possible to show that the evaluation performed in Step 2 gives an unbiased estimate of the error performed by a classifier learnt with the same parameter(s) and with training set of cardinality $|Tr| - |Va| < |Tr|$

K-fold Cross Validation

An alternative approach to model selection (and evaluation) is the K-fold cross-validation method

K-fold CV procedure

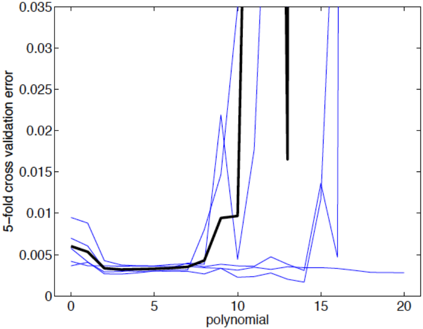
1. K different classifiers/regressors $h_1, h_2, \ldots, h_k$ are built by partitioning the initial corpus Tr into k disjoint sets $Va_1, \ldots, Va_k$ and then iteratively applying the Hold-out approach on the k-pairs $(Tr_i = Tr - Va_i, Va_i)$
2. Final error is obtained by individually computing the errors of $h_1, \ldots, h_k$, and then averaging the individual results

The above procedure is repeated for different values of the parameter(s) and the setting (model) with smaller final error is selected

The special case $k = |Tr|$ of k-fold cross-validation is called **leave-one-out** cross-validation

Back to our example

- Let's apply 5-fold CV



- Minimum error reached for $p = 3$, rather than the optimal $p = 12$
- Clearly, cross validation is no substitute for a large test set. However, if we only have a limited training set, it is often the best option available.

Back to our example

Why cross-validation selected a simpler model than optimal ?

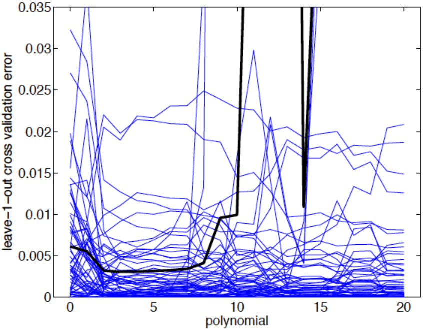
- Notice that with 60 points, 5-fold cross validation effectively tries to pick the polynomial that makes the best bias-variance tradeoff for 48 ($60 * \frac{4}{5}$) points
- With 10-fold cross validation, it would instead try to pick the best polynomial for 54 ($60 * \frac{9}{10}$) points

Thus, cross validation biases towards simpler models

- **leave-one-out** cross-validation reduces this tendency to the minimum possible by doing 60-fold cross validation

Back to our example

- So let's try **leave-one-out** cross-validation



- We still get $p = 3$!
- Cross validation is a good technique, but it doesn't work miracles: there is only so much information in a small dataset.

In general...

Almost all learning algorithms have (hyper)parameters!

- Support Vector Machines: C, type of kernel (polynomial, RBF, etc.), kernel parameter (degree of polynomial, width of RFB, etc.)
- Neural Networks: nonlinear/linear neurons, number of hidden units, η, other learning parameters we have not discussed (e.g., momentum μ)

Hold-out or Cross-Validation can be used to select the "optimal" values for the (hyper)parameters (i.e., select the "optimal" model).

After model selection, the training set is used to evaluate the goodness of the selected model

Many other algorithms/approaches and error measures

Please, remember that we have only presented **some** of the proposed learning algorithms/approaches, as well as possibile learning tasks and related error functions (loss functions)!
Just to name a few popular learning approaches

- Probabilistic approaches
- Decision trees
- Boosting
- Ensembles/Committees
- Genetic algorithms
- Prototype methods

Unsupervised Approaches!! Other model selection criteria: BIC, MDL, Bootstrap,...

Some tools

- Kernel Machines:
  - **Gaussian Processes**, **Mathematical Programming**, **Support Vectors**: http://www.kernel-machines.org/software
- Deep Learning:
  - **Caffe**: http://caffe.berkeleyvision.org
  - **Cuda-convnet**: https://code.google.com/p/cuda-convnet/
  - **Theano**: http://deeplearning.net/software/theano/
- General Machine Learning Tools:
  - in Java ⇒ **Weka**: http://www.cs.waikato.ac.nz/ml/weka/
  - in Python ⇒ **Scikit-Learn**: http://scikit-learn.org/stable/
- Data Stream Mining:
  - **MOA**: http://moa.cs.waikato.ac.nz

Some examples of applications

- **Spam Detection:** Given email in an inbox, identify those email messages that are spam and those that are not. Having a model of this problem would allow a program to leave non-spam emails in the inbox and move spam emails to a spam folder.

- **Credit Card Fraud Detection:** Given credit card transactions for a customer in a month, identify those transactions that were made by the customer and those that were not. A program with a model of this decision could refund those transactions that were fraudulent.

Some examples of applications

- **Face Detection:** Given a digital photo album of many hundreds of digital photographs, identify those photos that include a given person. A model of this decision process would allow a program to organize photos by person. Some cameras and software like iPhoto has this capability.

- **Digit Recognition:** Given a zip codes hand written on envelops, identify the digit for each hand written character. A model of this problem would allow a computer program to read and understand handwritten zip codes and sort envelops by geographic region.

Some examples of applications

- **Speech Understanding:** Given an utterance from a user, identify the specific request made by the user. A model of this problem would allow a program to understand and make an attempt to fulfil that request. The iPhone with Siri has this capability.

- **Stock Trading:** Given the current and past price movements for a stock, determine whether the stock should be bought, held or sold. A model of this decision problem could provide decision support to financial analysts.

Some examples of applications

- **Medical Diagnosis:** Given the symptoms exhibited in a patient and a database of anonymized patient records, predict whether the patient is likely to have an illness. A model of this decision problem could be used by a program to provide decision support to medical professionals.

- **Product Recommendation:** Given a purchase history for a customer and a large inventory of products, identify those products in which that customer will be interested and likely to purchase. A model of this decision process would allow a program to make recommendations to a customer and motivate product purchases. Amazon has this capability. Also think of Facebook, GooglePlus and Facebook that recommend users to connect with you after you sign-up.

Some examples of applications

- **Shape Detection:** Given a user hand drawing a shape on a touch screen and a database of known shapes, determine which shape the user was trying to draw. A model of this decision would allow a program to show the platonic version of that shape the user drew to make crisp diagrams. The Instaviz iPhone app does this.

- **Customer Segmentation:** Given the pattern of behaviour by a user during a trial period and the past behaviours of all users, identify those users that will convert to the paid version of the product and those that will not. A model of this decision problem would allow a program to trigger customer interventions to persuade the customer to covert early or better engage in the trial.

Some examples of applications

Let's now give...voice to experts in some popular applications!