Machine Learning

Winter 2011/12

Roland Memisevic

Lecture 1, Oct. 17, 2011

Machine Learning WS2011/12

- Instructor:

  Roland Memisevic
  Machine Learning, Robert-Mayer-Str. 10 - 201
- Lectures: Mondays 2pm - 4pm
- Tutorials: Wednesdays 2pm - 4pm
- All course related emails to:

  ml@vsi.cs.uni-frankfurt.de
- Course website:

  http://www.ml.cs.uni-frankfurt.de/teaching/ml2011/

Tutorials

- Mix of theory and programming exercises.
- Pre-conditions for taking part in the tutorials: Prepare and hand in your solutions.
- Doing the exercises is not mandatory, but you can improve your grade by doing them.
- For programming exercises we will make use of

  Python + (numpy, matplotlib, scipy) = "pylab"

Related courses

- Adaptive Systeme (Brause)
- Mustererkennung (Mester)
- Computational Learning Theory (Schnitger)

Readings

- Textbook:

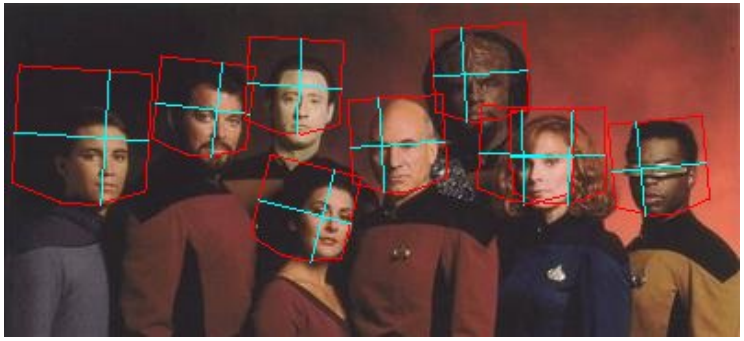
  Christopher Bishop: "Pattern Recognition and Machine Learning"

- Other useful books:
- D. MacKay: "Information Theory, Inference and Learning Algorithms"
- R. Duda and P Hart: "Pattern Classification"
- T. Hastie, R. Tibshirani and J. Friedman: "Elements of Statistical Learning"

What is Machine Learning?

- Some tasks are extremely hard or tedious to program by hand.
- For example, face recognition:

What makes a face?



- The output of a face-detector.
- Try to *program* this!

What makes a "2"?

- What are the rules that define a 2?

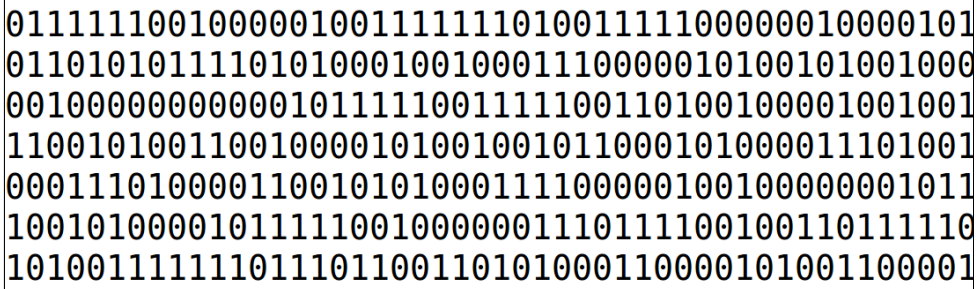
What makes *French*?

"Wikipedia.fr est un site de l'association Wikimedia France. Les resultats du moteur de recherche proviennent de Wikiwix"

- How to recognize the language, given a stream of characters?

Data analysis

- In many areas, including most sciences, we are facing a gigantic increase in available data, due to the Internet, cheap computers, cheap storage, etc.
- Machine learning can help make sense of, and understand, all this data.



More examples

- Email/Spam classification
- Facial identity recognition (eg., on facebook)
- Smile recognition (in your digital camera)
- Building autonomuous robots
- Surveillance
- Music recognition (on your cell-phone)
- Self-parking (or driving) car
- Stock-price prediction
- Speech recognition
- Action understanding (eg., in your Xbox/Kinect)
- Network anomaly detection
- Recommendation systems
- "Mind reading"
- Computational biology
- Machine translation
- etc.

Relation to neuroscience

- The best data-analysis and machine learning system is the human brain.
- Many machine learning techniques are therefore inspired by what we know about information processing in the brain.
- In turn, our understanding of the brain grows with our studying of machine learning techniques, their limits, their implementations, etc.
- A recent example: "The Bayesian Brain"

Machine learning and statistics

- The original science of data is statistics.
- Like statistics, machine learning makes heavy use of probability theory.
- An attempt at distinguishing the two:
- In contrast to statistics, the focus in machine learning is not on building tools that allow humans to analyze data, but on building systems that understand data by themselves.

Course outline (approximate)

- Overview
- Linear regression
- Linear classification
- Neural networks
- Kernel machines
- Bayesian reasoning
- Clustering, mixture models, the EM algorithm
- Sequences and Hidden Markov Models
- Graphical models, approximate inference, sampling
- Structured prediction
- Deep learning and feature learning

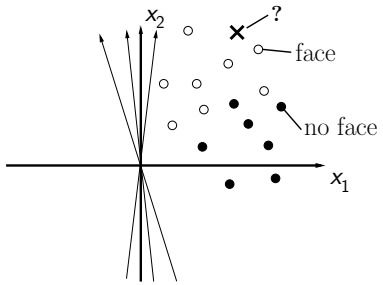
Different types of learning

- **Supervised learning:** Given data pairs $(\mathbf{x}, \mathbf{t})$, learn a function $\mathbf{y}(\mathbf{x})$ that *predicts* $\mathbf{t}$ on future points $\mathbf{x}$.
- **Unsupervised learning:** Given unlabelled data $\mathbf{x}$ (for example, images), learn to re-represent the data. This can include finding groups (clustering), extract features, compression, etc.
- **Reinforcement learning:** Learn to take a series of actions that maximize reward (for example, in games). Not covered in detail in this course.

Representing data using variables

- We use variables to represent everything.
- Both $\mathbf{x}$ and $\mathbf{t}$ can be
  - Scalar or vectors
  - We will denote scalars also in non-bold: x and t
- Both $\mathbf{x}$ and $\mathbf{t}$ can be
  - Continuous: Eg. $x = 1.73457$, or $\mathbf{x} \in \mathbb{R}^D$
  - Discrete: Eg. $x \in \{0, 1\}$, or $\mathbf{x} \in \{\text{`}a\text{'}, \text{`}b\text{'}, \text{`}c\text{'}, \ldots, \text{`}z\text{'}\}^D$
- Learning almost always amounts to adjusting parameters. We will usually stack parameters in a vector $\mathbf{w}$.

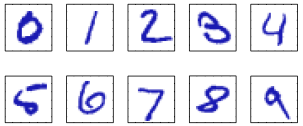
Points in space



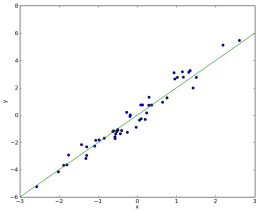
- Data cases **x** will typically be high-dimensional.
- But we can imagine only up to three dimensions.
- Our 3 dimensional intuitions often work, but sometimes they fail!
- High-dimensional spaces have certain peculiar properties, such as being "fairly empty". These properties are sometimes referred to as the **curse of dimensionality**.

Supervised learning

- **Classification:** Predict outputs that are discrete.

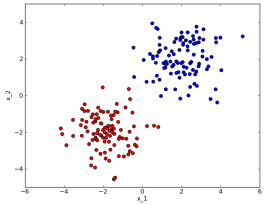


- **Regression:** Predict outputs that are continuous.



Unsupervised learning

- **Clustering:** Find groups in data. This is like finding a new, discrete representation for data.



- **Dimensionality reduction:** Find a concise, continuous description of data.
- Unsupervised learning is *lossy compression*: Data is represented using fewer bits.

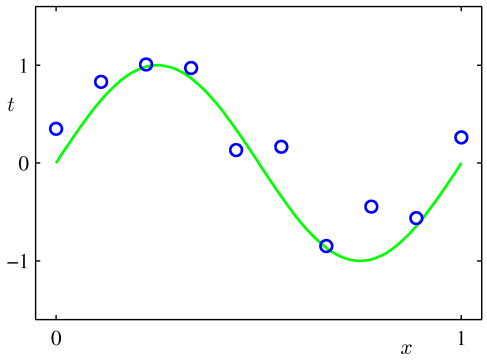
Generalization

- Both, supervised learning and unsupervised learning adapt model parameters based on **training data**.
- Training is like search in a **hypothesis space** containing functions.
- A central question is how well the model will perform on future data, which is known as **generalization**.
- How well the model performs on future data is a function of (a) how well it performs on the training data *and* (b) the size of the hypothesis space.

Example

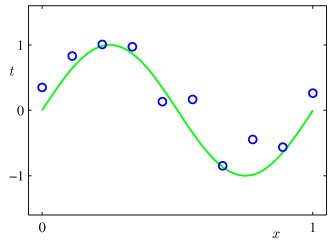
An easy way to solve digit classification is to *memorize each training example*. But this will be useless on new data!

Example: Curve fitting



- Training data
- Green line represents the true relationship between x and t
- Observations are noisy.

Example: Curve fitting



- Fit this with a polynomial:

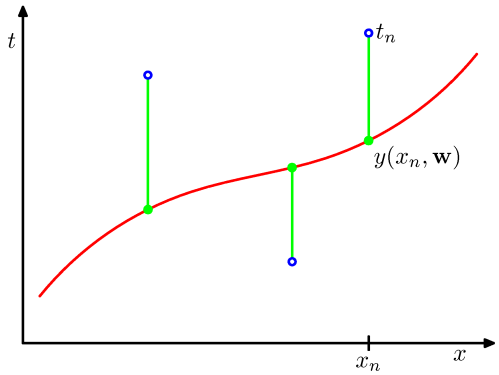
$$y(x, \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + \ldots + w_M x^M$$

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

- To set parameters $\mathbf{w}$, minimize the squared error:

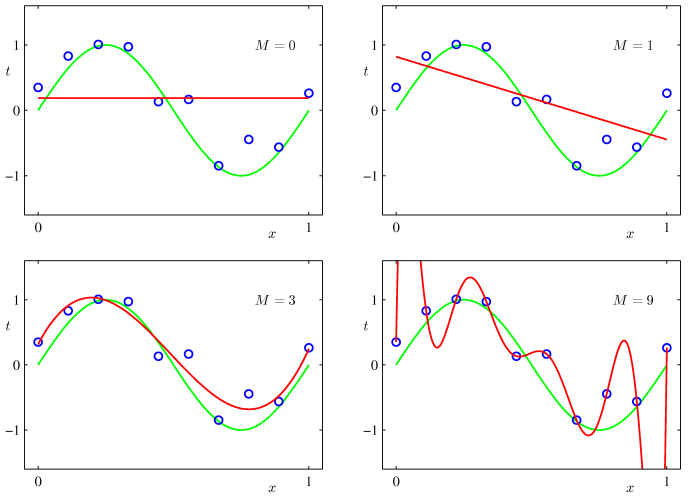
$$E(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^{N} \left(y(x_n, \mathbf{w}) - t_n \right)^2$$

Example: Curve fitting

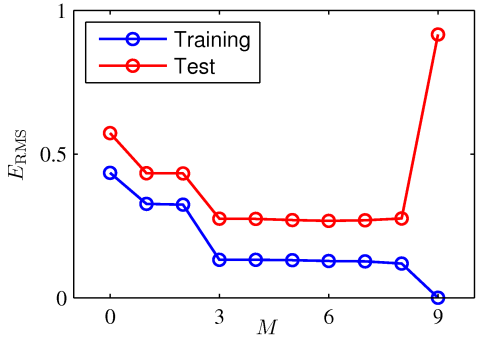


- Three training points and some learned function $y(x, \mathbf{w})$

Example: Curve fitting

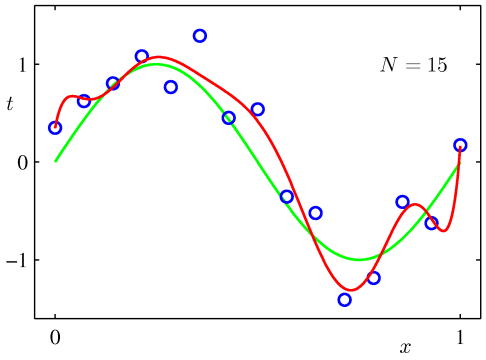


Example: Curve fitting



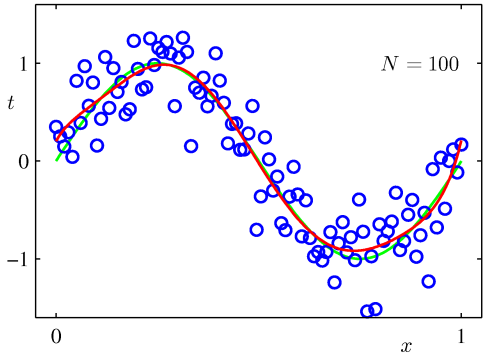
- Doing better on training data does not imply that you do better on test data.
- This is called **overfitting**.
- Thus reducing capacity helps generalize.

Example: Curve fitting



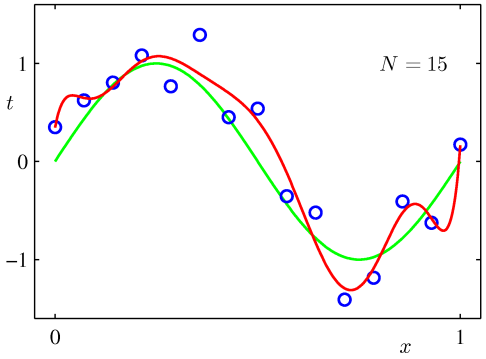
- More data does, too:

Example: Curve fitting



- More data does, too.

Example: Curve fitting



- So what can we do against overfitting, when the amount of data is limited?

Preventing overfitting

- **Model selection:** Use the right model class.
- **Regularization:** Penalize large coefficients:

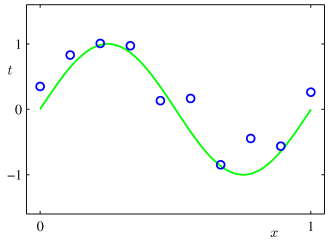
$$\tilde{E}(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\left(y(x_n,\mathbf{w}) - t_n\right)^2 + \frac{\lambda}{2}\|\mathbf{w}\|^2$$

- **Bayesian modeling:** Do not try to fit a model at all! Compute a conditional probability distribution $p(\text{model}|\text{data})$ over models given the training data. This is significantly harder to do in practice, but it is a natural way to prevent overfitting, and it works very well.
- These approaches are closely interlinked and connected. For example, Bayesian reasoning can provide us with a way to do model selection; regularization can be thought of as a (very) poor man's Bayes.

Training data, validation data, test data

- To perform model selection, it is common to split your training set into a **training set** and a **validation set**.
- Now fit several different models (for example, with different M, or λ) to the training set.
- Then check, how well each one does on the validation data!
- One can exchange the roles of training and validation subsets to get a more stable estimate. This is known as **cross-validation**.
- Extreme case: **"Leave-one-out" cross validation:** Fit models on all subsets of $N-1$ cases, evaluate each on the remaining case.

No free lunch



- When all you have are the training points, there is *no* way you can generalize.
- **Inductive bias**: To learn something, we *must* make assumptions.
- This has been formalized in a variety of so-called "No free lunch theorems".
- The most common assumption made, which often works surpisingly well: *Smoothness* of the underlying function.

Many open research questions

- Improve speed, accuracy, generality of methods.
- Find the right inductive biases for real-world tasks.
- End-to-end learning of complex models.
- Application specific problems.
- Etc.

Conferences and Journals

- NIPS: Neural Information Processing Systems, ICML, UAI, AISTATS
- PAMI: Pattern Analysis and Machine Intelligence, Journal of Machine Learning Research, Journal Machine Learning, Neural Computation