machine learning 1
week 1: introduction and prerequisites

justin bayer (TUM) *1983
claudio castellini (DLR) *1972
christian osendorfer (TUM) *1977
patrick van der smagt (DLR & TUM) *1966

what is machine learning?

When computers are applied to solve a practical problem it is usually the case that the method of deriving the required output from a set of inputs can be described explicitly. [. . .] As computers are applied to solve more and more complex tasks, however, situations can arise when there is no known method for computing the desired output from a set of inputs, or where that computation may be very expensive.

Cristianini & Shawe-Taylor
An introduction to Support Vector Machines, 2000

In general, we will be looking for the function $\mathcal{F} : \boldsymbol{x} \rightarrow \boldsymbol{z}$

machine “learning”

> *Learning denotes changes in the system that are adaptive in the sense that they enable the system to do the task or tasks drawn from the same population more efficiently and more effectively the next time.*
>
> Herbert Simon

No computer needs “learning” to do algebraic computation.

Machine Learning is programming computers to **optimise** a performance criterion using **example data** or **past experience**.

why not machine learning?

knowing what thou knowest not

is in a sense omniscience

Piet Hein, Grooks

a word for the wise: don’t learn what you already know!

If you need to control a robot, buy an engineer. He can solve that!

ML is useful when we *need* it

the real world: a phenomenon $\mathcal{F} : \boldsymbol{x} \rightarrow \boldsymbol{z}$

the engineer's world: we can *model* it since we know the machine

$\mathcal{M}$ is meticulously obtained by careful engineering

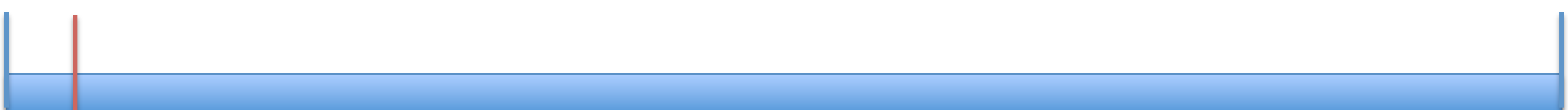
$$\mathcal{F} \equiv \mathcal{M} + \eta$$

the real world: we can *measure* it

$\mathcal{M}$ is an *educated guess* with parameters $\boldsymbol{w}$

$$\mathcal{F} \equiv \mathcal{M}\,(\boldsymbol{w}) + \eta$$

and we can obtain data $(x_i,\ z_i)$



once again

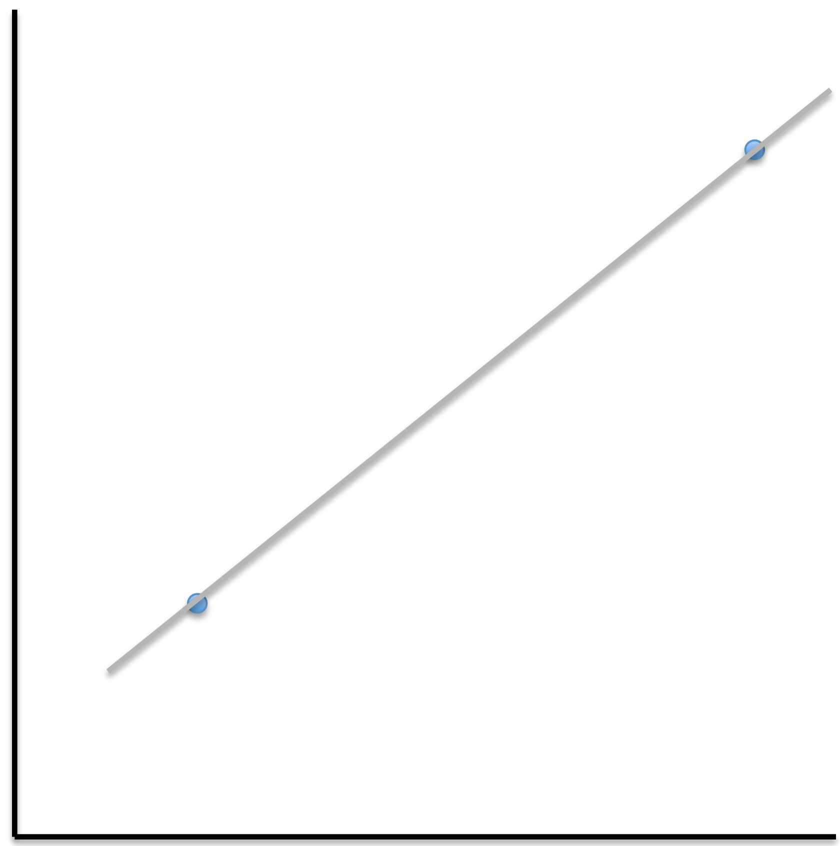
1. *models* are good when you can get them

2. *ML* is good to extend partial models

3. *BUT:* do not try to solve ill-posed problems with ML

4. LOOK AT YOUR DATA!

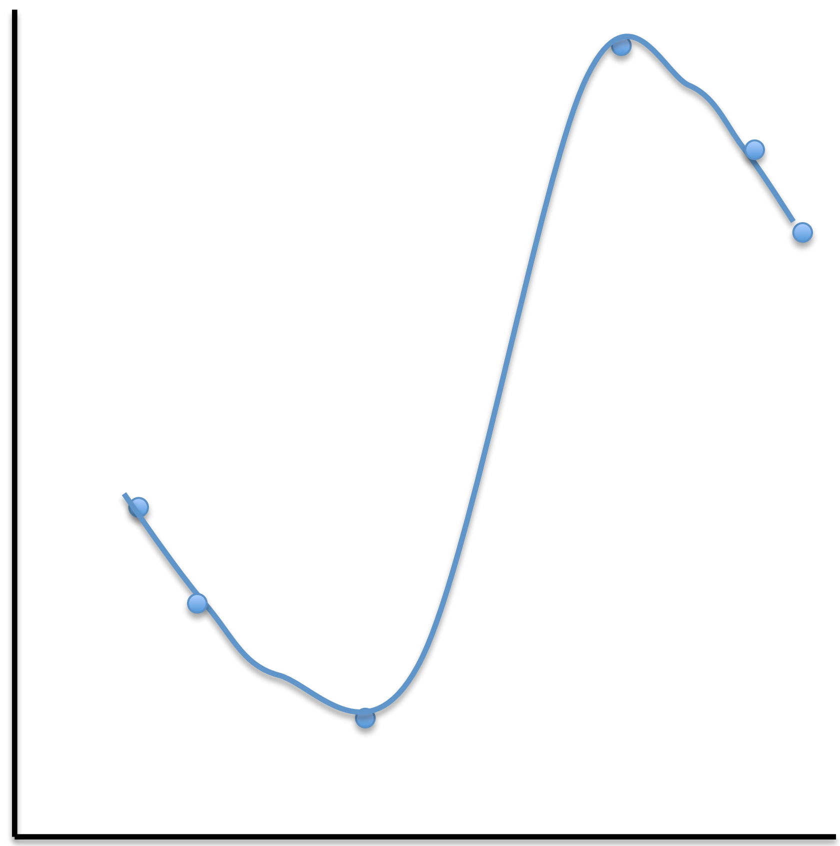
how can we best fit the underlying phenomenon?
how can we best fit the data?



knowing nothing of where the data came from, I am assuming a simple model
This simple model is obtained by **looking at the data**

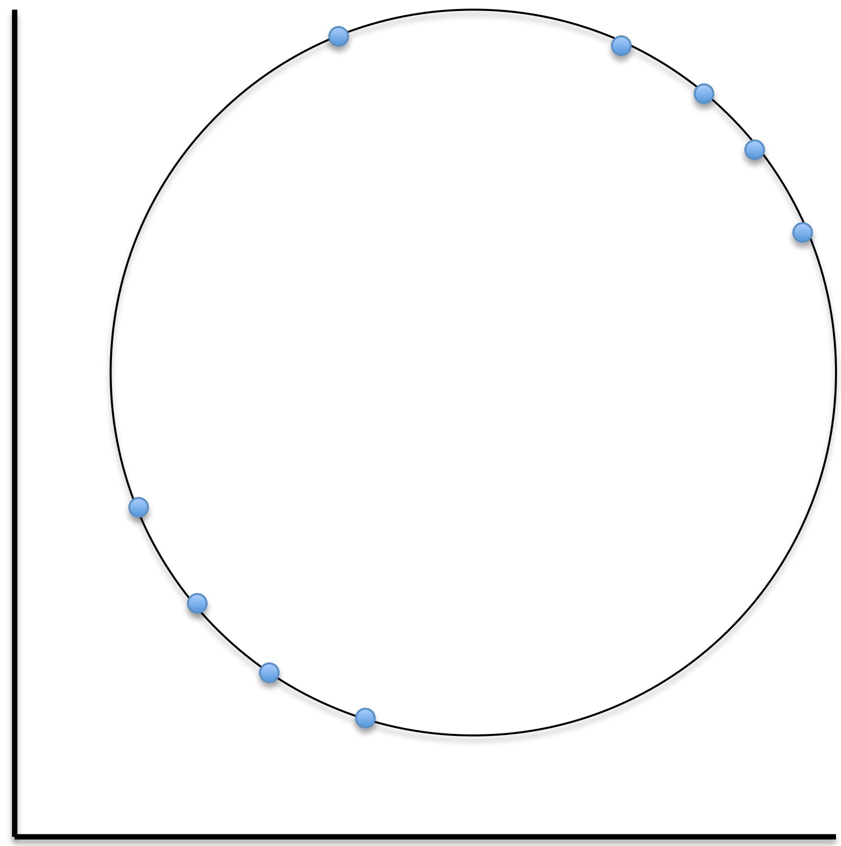
how can we best fit the underlying phenomenon?

how can we best fit the data?



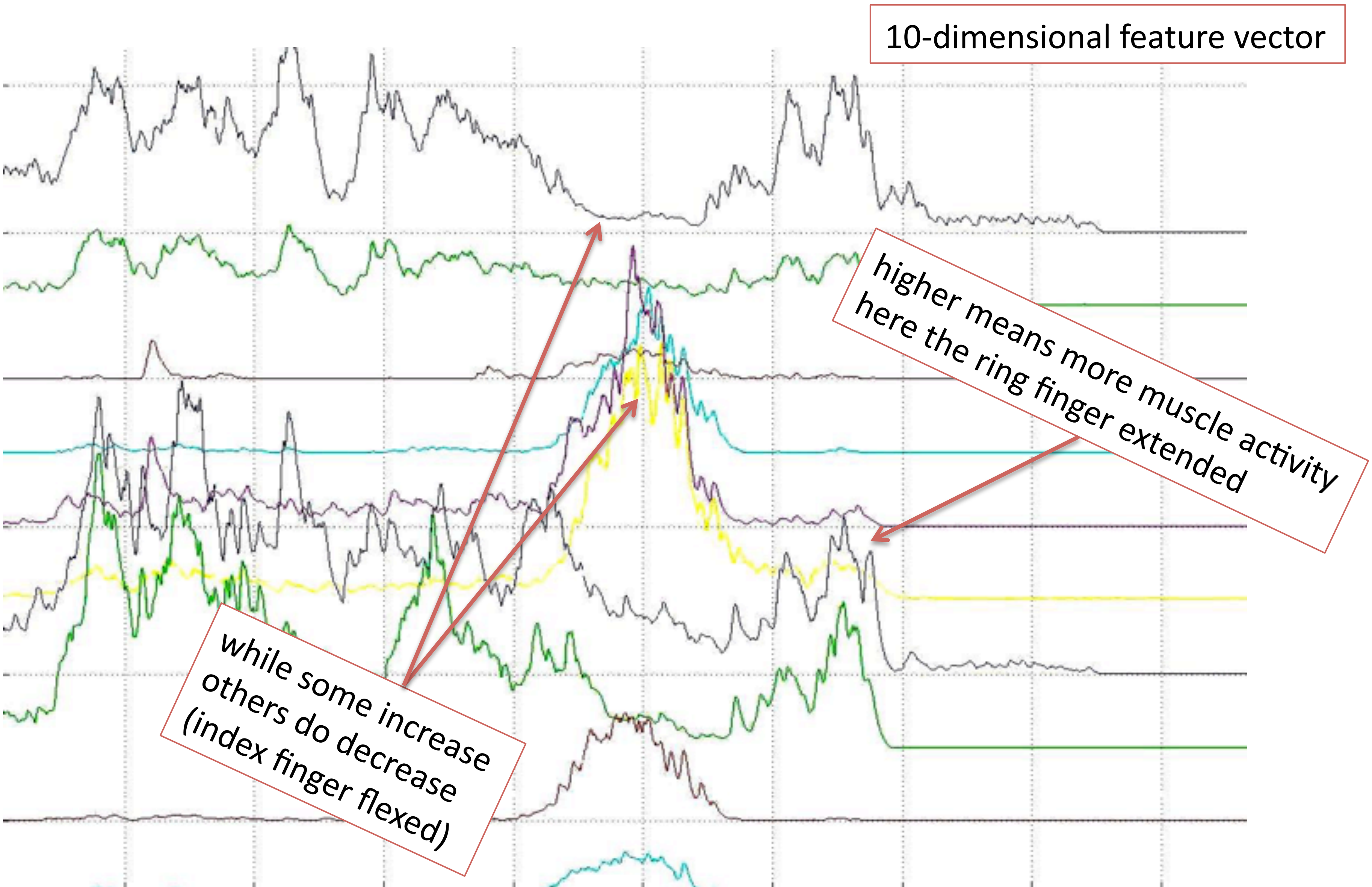
how can we best fit the underlying phenomenon?

how can we best fit the data?



looking at the data is easy to do in 2D… but what do you do in the real world?
This is where the statistics of ML comes in.

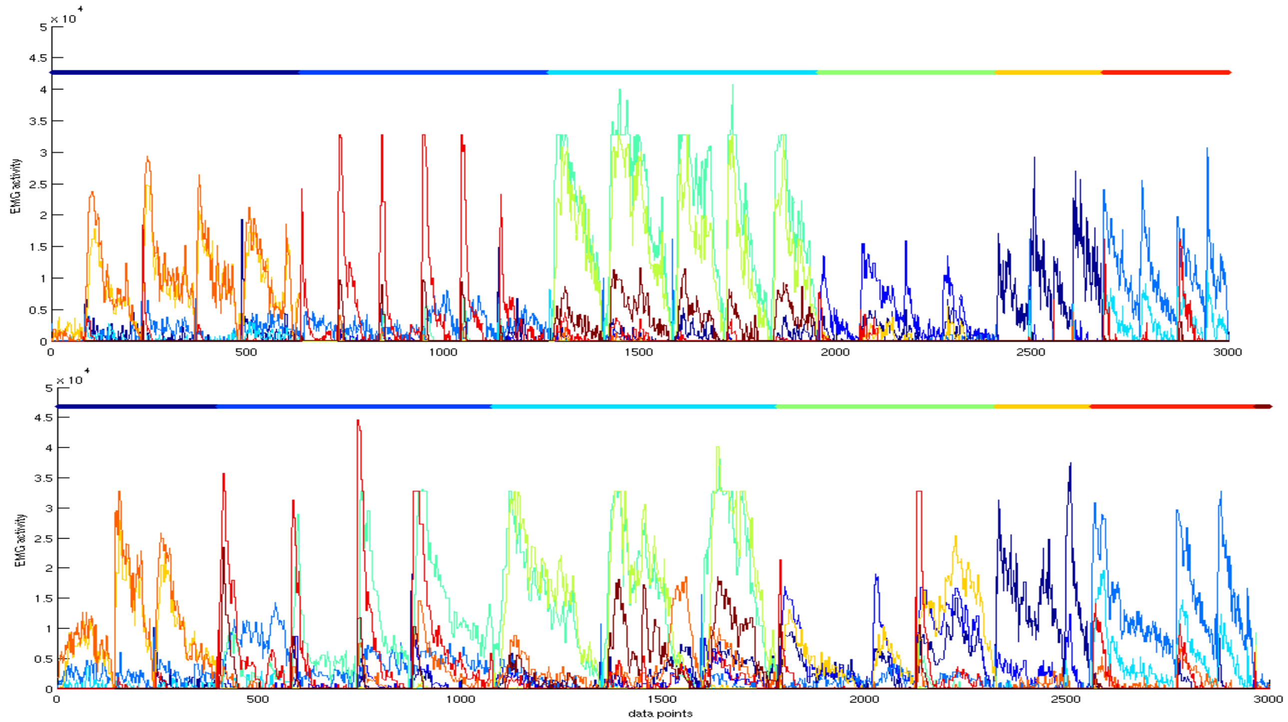
a real-world example: surface ElectroMyoGraphy



looking at EMG data

so the data seems sensible, but too complex to analyse “by hand”

furthermore, remember the context dependency which adds another dimension to all data vectors



11

why is this different?

because we can measure

- joint positions
- joint torques
- Cartesian object coordinates
- object forces
- ...

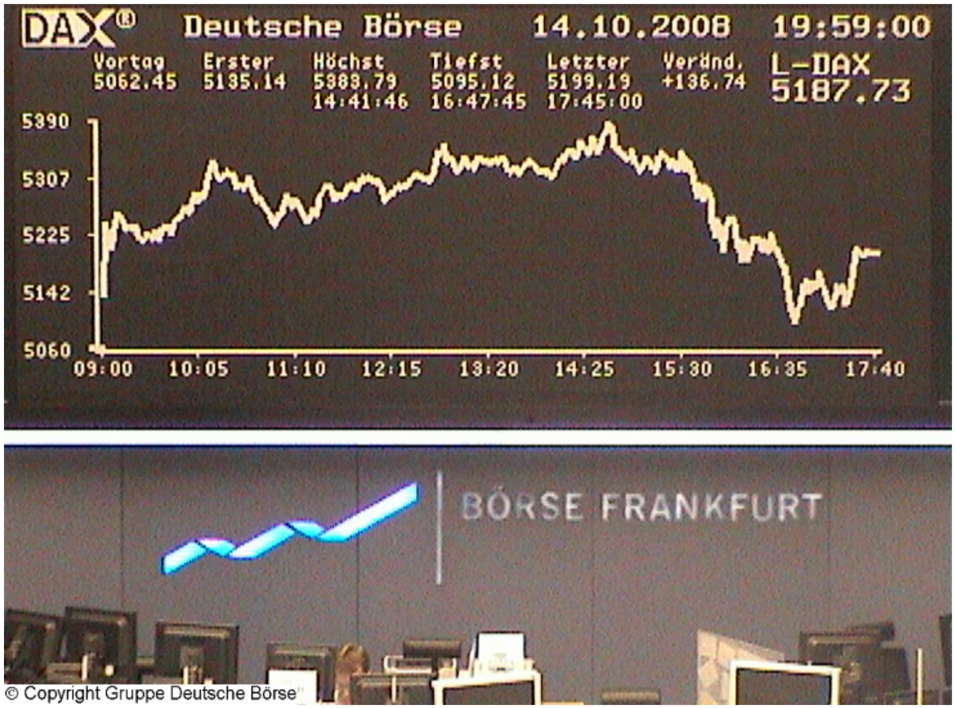
how does that work?

1. images I are recorded by the cameras
2. known object properties allows the object to be located in I
3. this can be converted to a Cartesian object position $\boldsymbol{p}$
4. we also know the joint positions $\boldsymbol{q}$
5. computing $\boldsymbol{q}'$ with which position $\boldsymbol{p}$ can be reached is algebra
6. we can compute the series of motor currents $\boldsymbol{i}(t)$ with which we can move the robot from $\boldsymbol{q}$ to $\boldsymbol{q}'$
7. and something similar for exerting forces in the contact case

… because everything is *modelled*

you don’t always know whether your data is sufficient or sensible

$\mathcal{F}($



$) \stackrel{?}{=} €€€$

but we *do* know how to make the best of it!

what we would like to teach you (tentative!)

1. introduction; refresher linear algebra; refresher probability theory
2. Gaussians
3. from machine learning to Bayesian: the coin
4. Maximum Likelihood Estimation (MLE) and
   Maximum A-Posteriori (MAP) linear regression
5. Bayesian linear regression
6. linear classification
7. generative modelling of EMG
8. kernels
9. support vector machines (SVM)
10. neural networks
11. optimisation
12. expectation maximisation
13. continuous latent variables

recommended reading

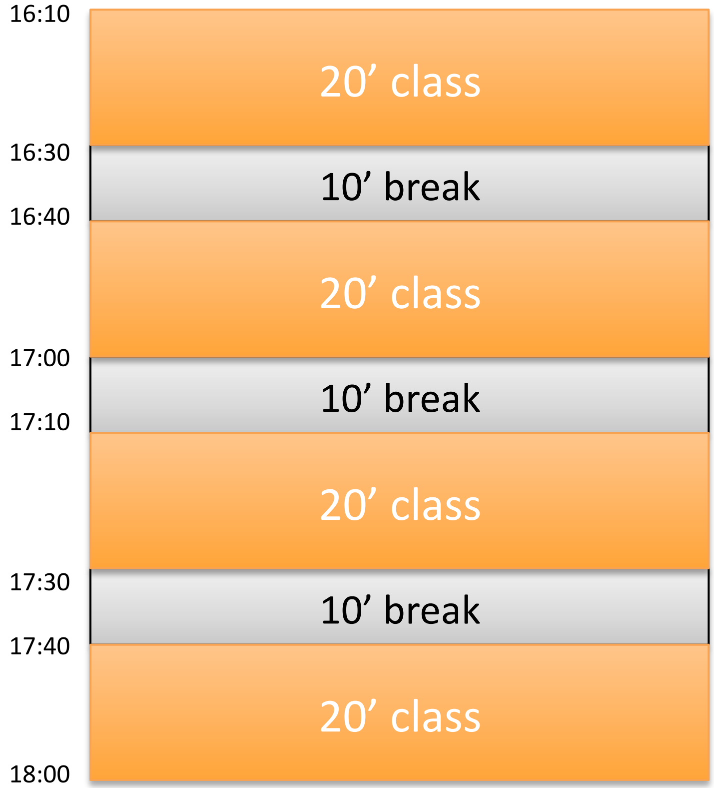
Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, Berlin, New York, 2006.

David J. C. MacKay. *Information theory, inference, and learning algorithms*. Cambridge Univ. Press, 2008 (free, online book!)

Kevin Murphy. *Machine Learning: A probabilistic perspective*. MIT Press, 2012 (selected chapters available which you can xerox, room tba).

block structure

Mo and Wed from 4pm till 6pm are divided into 4 blocks



things that matter

- all slides will be distributed online on the day of the class

- *not* coming to the class may make it tough to get the exam done.

- **Ask questions in class**. If something isn't clear to you, it probably isn't clear to others either. Most questions arise because the instructor hasn't made a connection clear or has inadvertently left out an important point. Your question gives the instructor a chance to explain more clearly.

exercises

we do have exercises handed out regularly. Do them and hand in the result first class of the following week (Mondays).

Some exercises require programming. We suggest you do this with python+numpy (or Matlab/Octave). For python+numpy see http://scipy-lectures.github.com/
Don't hand in the code; hand in the result only.

Solutions must be handed in as a typeset "manuscript" printed on recycled trees (typically 1 page). No handwritten results please. Write your name and IMAT# on each page.
Groups of 1 or 2 people are acceptable.

If you successfully solve an exercise in front of the class at three different occasions, you will be awarded a 0.3pt grade improvement (once only).

Access to the final exam is only granted if you successfully completed at least 2/3 of all exercises that you handed in.

final project

a final project will count as **1/4 of your credit** (the exam 3/4) [TBC]. The final project should allow you to work more in depth on a specific aspect of machine learning that you are interested in. In general, you can come up with any machine learning related idea as your project. The most straightforward ideas probably revolve around applying a machine learning technique to several interesting data sets. Or maybe you have an idea for improving a specific algorithm? Or want to analyse some theoretical aspect in more detail? Maybe you are more into visualisation and have an interesting idea how to connect that to Machine Learning.

Some examples:

- if you found some interesting data that you want to analyse and you are looking for a suitable algorithm, check out recent publications in the community (from about 2005 on, see NIPS, ICML, AISTATS, CVPR, UAI, ...)
- a specific, but unusual idea: Implement neural networks in Lua, using LuaJIT FFI to a BLAS (preferable GotoBLAS).
- Analyse tweets using the recent paper Learning Continuous Phrase Representations and Syntactic Parsing with Recursive Neural Networks
- Implement an SVM on a GPU using cudamat
- use a differentiable unsupervised preprocessing technique, combining it with any supervised learning approach that can be trained with gradient descent

but don't bother us with an implementation of backprop in C (which you can download from several thousands sources on the web)!

Proposals for final projects must be handed in by **December 16, 2011, 23:59:59.00 CET**.

You can decide to do your final project by yourself or in a group of 2.

Hand in by **February 26, 2012**: summary (3-6 pages) and a poster, you have to submit all used code and used datasets. You are free to give a short presentation (about 10-15 minutes).

a summary on grades

access to the final exam is only granted if you successfully completed at least 2/3 of all exercises that you handed in.

final project is ¼, written exam is ¾ of the grade (to be confirmed)

no oral exam possible; only one written exam will be held.

one-time grade improvement of 0.3pt possible if you solve an exercise in front of the class on 3 different occasions

one-time grade improvement of 0.3pt possible if you make *good* notes during the class, typeset them, and hand over the files to us for distribution via the website. Give us a sign before class when you opt for this!

some more background information

you are well advised to join our mailing list, go to

http://lists.brml.de/cgi-bin/mailman/listinfo/mlstudents

to subscribe. Here we will announce all course-related info.

the webpage of the class is at

http://www.brml.de/class/

organisational questions w.r.t. the class: ask smagt@tum.de

#ECTS = 6

the symbols that we will (try to) use

d dimensionality (of data)
n number of samples
p probability
t time
$\boldsymbol{x}$ input
$\boldsymbol{y}$ output of model/prediction
$\boldsymbol{z}$ target
$\mathcal{D}$ dataset = $\{(x_i, z_i)\}$, i = 1,...
$\mathcal{F}$ function
$\mathcal{M}$ model
$\mathcal{N}$ normal distribution
ϕ for basis functions, transfer functions, feature extractors
λ constant factor for regularization term
μ for mean
θ, w are parameters
σ is standard deviation
$\mathbb{R}$ *et al.* for real numbers etc.

vectors are bold
matrices are capital