T-61.5040
Learning Models and Methods, (5 cr) L

Spring 2007

Lecturer: **Petteri Pajunen**

Course Assistant: **Ville Viitaniemi**

LECTURE 1: 17.1.2007

GENERAL MATTERS
INTRODUCTION
CAN WE LEARN FROM DATA?

Times, Dates, Locations

- lectures: wed 12-14 in T4, from 17.1.2007 given by Petteri Pajunen
- exercises: fri 10-12 in T3, from 19.1.2007 given by Ville Viitaniemi
- contact: t615040@mail.cis.hut.fi (lecturer and assistant)
- office hour: lecturer on wed 14-15, B309

Course Material

- no textbook: lecture slides and possibly some other material will appear on course homepage as the course proceeds: `http://www.cis.hut.fi/Opinnot/T-61.5040/`
- exercise problems will be available on the website before each exercise session
- solutions will be posted on the website as the course proceeds

- a major topic of the course is *Bayesian Inference*
- for more in-depth information on Bayesian Inference, one of the following books may be useful:
  1. Gelman et al: Bayesian Data Analysis, CRC Press, 1996 (or 2nd ed Chapman & Hall/CRC 2003)
  2. Bernardo and Smith: Bayesian Theory, Wiley, 2000
- these are *not required reading* and do not contain even half of the topics on the course

Prerequisites

- prerequisites are basic math courses, especially probability, matrices and calculus
- some course in modeling data nonlinearly (for example neural networks or pattern recognition) would be helpful
- some programming skills are needed for the computer assignment

Passing the Course

- final exam+compulsory assignment
- first final exam in May 2007, everyone may take this
- later exams can be taken only after your computer assignment has been accepted

Computer Assignment

- instructions will appear on the website by the end of March
- datasets and “technical support” are available in MATLAB and R (`www.r-project.org`)
- but you can use other software/language if you wish
- deadline will be around May/June (announced later)

Exercises

- some (or many) exercise problems are difficult and/or not meant to be solved based on what has been presented at the lectures
- these are intended to illustrate some important ideas in more detail than is possible in the lectures
- these are marked “demo” in the problem sheet
- other problems are more or less possible to be solved by yourself, though they may also be difficult...

Introduction

- emphasis is on presenting fundamental ideas through simple examples and counterexamples
- no overly complicated derivations of theoretical results
- however, somewhat complicated calculations are unavoidable when applying Bayesian Inference
- most skipped details can be found in the two books mentioned before, or from references that will be posted on the website

- don’t take the course if you want
  - lots of algorithms to stick data into without thinking
  - rigorous derivations of theoretical results
- do take the course if you want
  - to understand the learning problem (have data, what to do with it?)
  - to understand Bayesian methods at a general level

- *learning*: for our purposes, using a given set of observed data to get information about unobserved quantities
- for example, predicting something or making an optimal decision
- this definition of learning covers most situations where one would like to learn from data in practice
- note that it does not contain algorithms that represent the data in another way (Fourier transform, PCA etc..), although these algorithms can be useful as a part of a learning method

- example: where to sit in a lecture hall?
  - first you try the front row and find that the temperature is too cold
  - next week you try the last row: now the temperature is too high
  - where do you try to sit next week?
  - you have data, and are trying to make an optimal decision using it

- *model*: represents the information we have about the learning problem
- in the lecture hall example, most people have a model which might be described as "temperature changes slowly as a function of position"
- without a model, you might as well choose the next seat randomly
- we will see that learning from a given set of data without a model (i.e. without assuming anything) is not possible in any meaningful way

- modeling data has been done in statistics for a long time
- but traditionally the lack of computing power has resulted in:
  - realistic but unsolvable models, or
  - computable, simple standard models (usually unrealistic)
- one can run into complicated models quite easily:
  - missing data (parts of data are missing in various ways)
  - mixture models (discussed in a later lecture)

- emerging trend: increasingly complex models for which approximate solutions can be computed (neural networks, data mining etc...)
- often these models are "generic" (not constructed for a specific problem and are often used to solve a wide variety of problems)
- this approach represents a compromise between a realistic unsolvable model and a solvable but too simple model
- reasons for not using the problem-specific correct model may include computational constraints, not enough data, model too difficult to solve etc.

- some wrong intuitions:
  - a complex model and lots of data solves all problems (not so)
  - a model and data solves decision problems (not so)
  - results of learning from data are objective (not so)
  - some learning methods are better on average than others, assuming nothing about the problems they are applied to (not so, in a sense to be made clear later)

- in this course, we deal with the problem of learning from a *given set of data*
- in practice, other steps are needed that are often problem specific
- e.g. what data to collect, how to preprocess it etc..
- sometimes the outcome of system is not unique: for example, fault detection requires deciding how the different faults are defined
- most of these extra steps in learning require decision theory

- we end up using *probabilistic modeling*
- suppose we look at a *deterministic* system that gives always the output y_i when given the input x_i
- even in this case there are good reasons to use probabilities
- first, the system may contain variables that we do not observe

- even if the world works in a deterministic way, one can only model a tiny part of it relevant to the problem at hand
- examples:
  - flipping a coin: measuring the initial speed, direction, and rotation of the coin, and predicting where it will land is too difficult
  - the change in direction of a moving car: speed and steering wheel position explain it well, but is also affected by changes in speed, surface friction and unevenness etc.
- therefore we will end up using probabilistic models

Causality

- dependence between inputs and outputs can be one of the following type:
  - output caused by input (steering wheel position $\rightarrow$ direction of car)
  - input caused by output (symptoms $\rightarrow$ illness; the arrow denotes the system, not causality)
  - input and output caused by something else (consumption of soft drinks, number of people drowning; both are probably caused by warm weather)

  - noncausal dependence
- cause of dependence, or anything else that is unobserved, cannot be learnt from given data unless one is willing to use a model containing the causality as a parameter
- without such a model, learning methods can be used to predict unknown outputs. This does not require causality.

Summary

- no long derivations of theoretical results, no black box tools
- demonstrate the impossibility of learning from data without assumptions
- can we learn from data by assuming very little?
- develop probabilistic approach to learning (Bayesian Inference)
- examine basic properties of Bayesian Inference
- does Statistical Learning Theory say that one can

  learn without assumptions?
- apply Bayesian Inference in various situations
- how to compute approximate solutions
- missing data
- latent variable models
- Gaussian processes
- making decisions

- goals of the course: to understand
  - the impossibility of learning without assumptions
  - the difficulties of high dimensional problems
  - in what sense probabilistic learning is correct
  - how to think about problems using probabilities
  - how to do BI in practice
  - the need to approximate BI, and some methods to implement this
  - how to make decisions and what this has to do with BI

Cannot Learn From Data

- if one wants to learn from data, there must be something to be learned
- define this as θ, *state of nature*, which is *unknown*
- set of all possible states of nature is Θ
- θ can also be a *parameter* of some function or distribution

- for practical reasons, lets assume that θ has a numerical value (can be a vector or a matrix)
- the learning problem is to get information about the value of θ using data (denote here by D)
- can we say something useful about θ *without using any information except data* D?
- when stated this way, it seems obvious that we can't
- but putting data into an algorithm which is unrelated to the problem, or is not known to be related, is just trying to achieve this

- example: predict output y from a given input x (regression)
- $D = \{(x_1, y_1), \ldots, (x_n, y_n)\}$, $\theta = \tilde{y}$ corresponding to $\tilde{x} \notin \{x_1, \ldots, x_n\}$?
- no-brain solution: pick some learning method A, train it on data D and use it on $\tilde{x}$ to get a predicted value $\tilde{y}$

- you might as well have chosen a different learning method B
- suppose B gives a different prediction $\tilde{y}$ than method A
- this is not learning from data: we are deciding the prediction result as we choose the learning method A or B (or C,D,E,...)
- any learning method contains implicit information that is combined with data

- how about using data to select the learning method?
- this sounds reasonable and is often done (e.g. model selection)
- construct method C so it is identical to A within the training set and identical to B outside of it
- training data cannot distinguish between A and C, so we must pick one and decide the prediction again

- last straw: A is better than B or C in general, so we just use A
- sure, we used information “A is best” in addition to data but this is OK since we can now use A in every problem and do not need additional information
- this sounds good: just find the “best” learning method and be done with it
- researchers have certainly looked for it (cybernetics, neural networks, fuzzy logic, genetic algorithms, kernel methods etc..)

- are some learning methods better than others on average?
- such a method should have a better average performance than, for example, guessing the predicted values
- it sounds easy to come up with methods that beat guessing
- but it is even easier to prove this is impossible, either rigorously or by simple counterexamples

No Free Lunch Theorems

- it can be shown (D. Wolpert, No Free Lunch Theorems) that no information except training data leads to equal average performance over all learning methods
- for proofs, see original article(s) (reference on webpage)
- proofs are somewhat technical so we settle for examples restricted to predicting bits in the exercise problems

- NFL theorems are based on the idea that observed data without any other information does not restrict the unobserved data in any way
- no information means "all solutions equally possible"
- easiest to demonstrate in discrete, finite problems
- consider $\Theta = \{\text{every 4-bit sequence}\}$

- no information $\implies$ every 4-bit sequence equally possible
- observe first three bits, 010, and predict the last bit
- obviously the last bit is equally likely either 0 or 1
- easy to generalize to n-bit outputs, which essentially is all that is needed (unless you have an analog computer or enjoy making decisions with infinite number of choices)

- no information was interpreted as having a uniform distribution over Θ (e.g. over 4-bit sequences)
- NFL theorem $\implies$ performance of any learning method averaged over all problems is constant
- applies even for guessing and other apparently useless methods!
- intuitively clear, because regardless of training data, the distribution of non-training data is uniform

- performance = average over all $\theta \in \Theta$
- i.e. if method A predicts 1 for the input 010, we count this as 0 (no error) for $\theta = 0101$ and 1 (error) for $\theta = 0100$
- average performance is computed using a uniform distribution over Θ
- one might object that 'real-world' problems have 'structure', they are not uniformly distributed as assumed above

- another NFL theorem answers the non-uniform critique
- assume that the distribution over Θ can be non-uniform
- no information = we don't know this non-uniform distribution
- averaging uniformly over *all distributions over* $\Theta \implies$ same performance for all learning methods
- interpretation: to learn from data, you have to know something about the structure of the problem, not just that it has some

- NFL-idea seems almost trivial
- it makes clear that no information leads to no learning, regardless how much data one has
- yet much research seems to rely on this possibility
- heuristics such as cross-validation, Occam's razor, Minimum Description Length etc... don't change the conclusion

- in practice one expects to beat guessing
- if the learning method makes use of strong and correct assumptions, this is possible
- but in some areas such as neurocomputing, implicit assumptions seem very weak, such as simply assuming some vague regularity
- weak assumptions might be expected to hold for a large set of learning problems

- does a method A exists which beats guessing (B) on a large set of problems?
- it can be shown that the set of problems where A and B have even a small difference in performance, is *very small* (demo exercise problem)
- all methods are almost as bad as guessing on almost all problems
- expecting a method to work well, one's problem must come from the very small set of problems (a strong assumption)

- summary:
  - NFL thms $\implies$ no information, no learning
  - little information $\implies$ little benefit from training data
  - good expected performance $\implies$ strong, correct assumption (either implicit or explicit)
  - since one must bring a lot of information to the learning problem, it is preferable to know what is being assumed
  - this motivates the study of probabilistic learning methods where the assumptions are made explicitly

- does this bet sound appealing? It should for those who believe that information can be extracted from lots of data
  - I generate 10^6 bits of training data (given to you), and 100 bits of test data (not given)
  - you win 1000 EUR if you predict more than 60 bits of test data correctly
  - otherwise you lose 1000 EUR
  - this bet should be appealing if you believe your learning method has better performance than 0.6
  - hint: $\sum_{k=0}^{60} Bin(k|100, 0.5) \approx 0.982$