Machine learning 1

Introduction to machine learning

Piotr Wójcik
pwojcik@wne.uw.edu.pl

academic year 2017/2018

Uniwersytet Warszawski
Uniwersytet Warszawski Wydział Nauk Ekonomicznych

Organizational matters

Introduction to machine learning

Prediction

Inference

Methods of estimating f

Supervised vs. unsupervised learning

Regression versus classification

Organizational matters

Aims of the course

- give **theoretical background** and **intuitive explanation** for different machine learning algorithms
- learn to **select**, **implement**, **assess** and **compare** predictive models for **regression** and **classification** tasks
- learn to apply machine learning tools in R or Python on **real data**,
- prior knowledge of R or Python is **expected**.

Contents of the course

0. Organizational matters, introduction to machine learning
1. Initial data analysis, data preparation
2. Sample parametric methods: linear regression, logistic regression, linear discriminant analysis
3. Cost function, algorithm evaluation metrics for regression and classification
4. Train and test datasets, cross-validation, repeated cross-validation, bootstrap validation
5. Bayesian methods, naive bayes
6. Support Vector Machines
7. Variables transformation methods for inputs and target

Contents of the course – cont'd

8. Variables selection methods, variable importance measurement
9. Regularization methods
10. Lasso
11. Different optimization methods
12. Up-sampling and down-sampling
13. Workshops
14. Students' presentations – project on a **small dataset**
15. Students' presentations – project on a **large dataset**

Literature (interactive links)

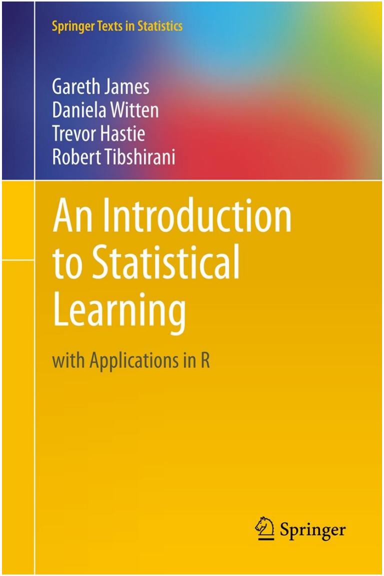
Basic handbooks:

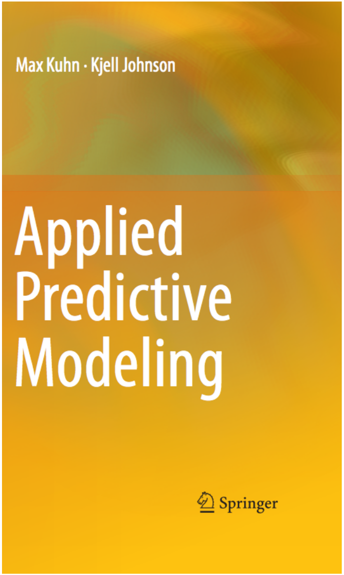
- Gareth James, Daniela Witten, Trevor Hastie and Robert Tibshirani (2017), "Introduction to statistical learning. With Applications in R", Springer-Verlag
- Kuhn Max, Johnson Kjell (2013), "Applied predictive modelling", Springer-Verlag

Additional handbook:

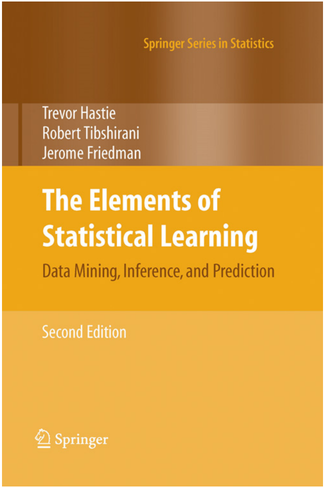
- Hastie Trevor, Robert Tibshirani and Jerome Friedman (2009), "Elements of statistical learning", Springer-Verlag

Basic handbooks





Additional handbook



Recommended Data-Camp courses (interactive links)

For R users:

- Introduction to machine learning with R
- Machine learning toolbox
- Supervised learnings in R: regression
- Supervised learning in R: classification

For Python users:

- Introduction to Python & Machine Learning
- Kaggle Python Tutorial on Machine Learning
- Machine learning with Python
- Supervised Learning with scikit-learn

Course assessment

- two practical machine learning projects prepared in groups of **at most 2 students** – one for **regression** and one for **classification**
- each project on a different dataset selected by students – **accepted by teachers**, for example from kaggle or USI Machine learning repository or any other source – see eg. here, here or here
  - one reasonably **small dataset** – # rows measured in hundreths or thousands
  - one **large dataset** – (# columns $\times$ # rows) **above** 1.000.000 and **at least** 20 variables of different types (continuous, categorical)

Course assessment – cont'd

- each project should include:
  - clear description of data and problem analyzed
  - initial descriptive analyses of the data
  - variable transformations
  - variable selection methods
  - training/test data division, resampling (e.g. cross-validation, down-sampling, up-sampling)
  - comparison of prediction accuracy of **at least 3** different machine learning methods on test data
  - summary and conclusions
- 50 points to be collected for **each project**:
  - **presentation** in class (15 pts)
  - **written report** in **RMarkdown** or **Python notebook** (35 pts)

Written report assessment criteria

- the report should be submitted as **PDF/html file** together with a **source R Markdown or Python Notebook file** with R/Python code chunks that allow the teachers to **fully reproduce** the applied analysis and generate a submitted PDF/html file
- assessment criteria:
  - clear **description** of the data and problem analysed
  - correctness of **selection**, **application** and **interpretation** of results
  - clear summary of **conclusions**
  - **form** (structure, language, tables, links, etc.)
  - **correctness of the R/Python codes**
- **further details related to project will be announced in March/April**

R or Python used in labs

- Python is a general purpose programming language whereas R specializes in a smaller subset of statistically-oriented tasks
- it is easier to learn basics of machine learning in R
- Python gives a more consistent interface once you have moved beyond the basics
- In R, switching between different models usually means learning a new package written by a different author
- **caret** is an excellent R package that attempts to provide a consistent interface for machine learning models in R
- **scikit-learn** is a very elegant Python equivalent for machine learning applications
- in case of using Python we suggest to use the Anaconda distribution – it includes nearly every Python package needed on the course and has a package management system similar to CRAN in R

Python or R – neverending discussion

- Python or R for data science?
- R vs Python data science
- The great "R versus Python" for data science debate
- Python vs or R artificial intelligence, ai, machine learning, data science, which use
- R vs Python for data science, big data, artificial intelligence, ml
- Which is better for data analysis R or Python
- Python vs R for machine learning
- Python vs R- he battle for data scientist mind share
- R vs Python which programming language should I learn
- R vs Python for data models data science
- R vs Python for data science
- Python vs R data science programming language

www & email

Maciej Wilamowski

homepage: www.wne.uw.edu.pl/mwilamowski

email: mwilamowski@wne.uw.edu.pl

Office hours: to be announced

Piotr Wójcik

homepage: www.wne.uw.edu.pl/pwojcik

email: pwojcik@wne.uw.edu.pl

Office hours: to be announced

Questions?

Introduction to machine learning

What is machine learning?

- The term **machine learning** is often used interchangeably with **predictive modelling**, **statistical learning**, **pattern recognition** and refers to a vast set of tools for understanding data
- these tools are usually used to build a model whose **main objective** is to provide accurate forecasts on **test data**
- the tools can be classified as **supervised** or **unsupervised**
- **supervised learning** involves building a statistical model for predicting, or estimating, an output based on one or more inputs.
- With **unsupervised statistical** learning, there are inputs but **no supervising output**

Brief history of statistical methods

- many of the concepts that underlie machine learning were developed long ago
- at the beginning of the XIX century the least squares method, now known as **linear regression** was introduced by Legendre and Gauss.
- **linear regression** is used for predicting **quantitative** values, such as revenue on a client, sales, etc.
- **qualitative** values, such as credit default, client churn, client preference over a basket of products/brands, etc. can be predicted by **linear discriminant analysis** proposed by Fisher in 1936.
- the alternative approach for **qualitative** variables prediction – **logistic regression** is known since 1940s.
- **generalized linear models**, the entire group of methods that include both linear and logistic regression as special cases appeared in the early 1970s.

Brief history of statistical methods – cont'd

- by the end of the 1970s, many more types of predictive models were available
- to great extent they were **linear methods** – fitting non-linear models was to costly from computational point of view at that time
- in mid 1980s classification and regression trees were introduced, together with cross-validation for model selection
- a class of non-linear extensions to generalized linear models – **generalized additive models** were presented in 1986 by Hastie and Tibshirani
- Since that time machine learning has emerged as a new subfield in statistics
- recent progress in statistical learning has been marked by the increasing availability of powerful and relatively user-friendly software

Increasing availability = wider audience

- that is why the field of statistical learning has also expanded its audience.
- user-friendly software generated interest in the field from non-statisticians, eager to use modern statistical tools to analyze their data
- highly technical nature of statistics restricted its practical use to experts in statistics, computer science, and related fields
- in recent years, new and improved software have significantly eased the implementation burden for many statistical learning methods
- at the same time, there has been growing recognition across a number of industries that statistical learning is a powerful tool with important practical applications
- as a result, the field has moved from one of primarily academic interest to a mainstream discipline, with an enormous potential audience

The purpose of Machine Learning courses (ML1 and ML2)

- we will **NOT** discuss all technical details behind machine learning methods, such as optimization algorithms and theoretical properties
- most users do not need a deep understanding of these aspects to become **informed users** of the various methodologies
- the aim is to focus on **intuitions**, and **strengths and weaknesses** of the various methods
- and present methods which are **most widely** used in **practical applications**
- describe **basic assumptions** and **intuition** together with **trade-offs** behind each of the approaches
- assumption: student **is comfortable with basic mathematical concepts**
- examples will show applications of machine learning methods on **real data**

General notation

- **input variables** are typically denoted using the symbol X, with a subscript to distinguish them
- **inputs** might be alternatively called: **predictors**, **independent variables**, **features** or sometimes just **variables**
- **output variable** is often called the **response** or **dependent variable** and is typically denoted using the symbol Y.

Relationship between output and inputs

- More generally, suppose that we observe a quantitative response Y and p different predictors, $X_1, X_2, \ldots, X_p$.
- We assume that there is **some relationship** between Y and $X = (X_1, X_2, \ldots, X_p)$, which can be written in the very general form as:

$$Y = f(X) + \epsilon$$

- f is some **fixed but unknown function** of $X_1, X_2, \ldots, X_p$, and ϵ is a **random error** term, which is independent of X and has mean zero.
- in other words f represents the **systematic information** that X provides about Y

The essence of machine/statistical learning

- statistical/machine learning refers to a set of approaches for estimating f
- there are two main reasons to estimate f: **prediction** and **inference**
- the distinction between **statistical learning** and **machine learning** is fuzzy
- **machine learning** is concerned primarily with **predictive accuracy** over **model interpretability**
- **statistical learning** places a greater priority on **interpretability and statistical inference**

Prediction

Prediction

- In many situations, a set of inputs X are readily available, but the output Y cannot be easily obtained.
- In this setting, since the error term averages to zero, we can predict Y using:

$$\hat{Y} = \hat{f}(X)$$

where $\hat{f}$ represents our **estimate** for f, and $\hat{Y}$ represents the resulting **prediction** for Y.

- In this setting, $\hat{f}$ is often treated as a **black box**, in the sense that one is not typically concerned with the **exact form** of $\hat{f}$, provided that it yields **accurate predictions** for Y

Accuracy of prediction

- In general, $\hat{f}$ will **not** be a perfect estimate for f, and this inaccuracy will introduce some **error**.
- The accuracy of $\hat{Y}$ as a prediction for Y depends on two quantities
  - **reducible error** – we can potentially improve the accuracy of $\hat{f}$ by using the most appropriate statistical method to estimate f
  - **irreducible error** – our prediction would still have some error, because Y is also a function of ϵ, which cannot be predicted using X
- variability associated with ϵ also affects the accuracy of our predictions
- **no matter how well we estimate f, we cannot reduce the error introduced by ϵ**

Irreducible error

Why is the irreducible error larger than zero?

- some factors influencing Y are **NOT** directly measured (e.g. sentiment of clients) – their impact will be included in ϵ
- since we don't measure them, f cannot use them for its prediction.
- The quantity ϵ may also contain unmeasurable variation – eg. !!!!!!!!!!!!!

Formal

- Consider a given estimate $\hat{f}$ and a set of predictors X, which yields the prediction $\hat{Y} = \hat{f}(X)$
- Assume for a moment that both $\hat{f}$ and X are fixed
- one can show that

$$
\begin{aligned}
E(Y - \hat{Y})^2 &= E(f(X) + \epsilon - \hat{f}(X))^2 = \\
&= \underbrace{(f(X) - \hat{f}(X))^2}_{reducible} + \underbrace{Var(\epsilon)}_{irreducible}
\end{aligned}
$$

- The focus of this course is on techniques for estimating f with the aim of **minimizing the reducible error**.
- the irreducible error will always provide an upper bound on the accuracy of our prediction for Y
- **This bound is almost always unknown in practice**.

Inference

Inference

- We are often interested in **understanding the way** that Y is affected as $X_1, \ldots, X_p$ change.
- In this situation we wish to estimate f, but our goal is **not necessarily to make predictions** for Y.
- We instead want to **understand the relationship** between X and Y, or more specifically, to understand how Y changes as a function of $X_1, \ldots, X_p$.
- Now $\hat{f}$ **cannot** be treated as a black box, because we need to know its exact form.

Inference – cont'd

In this setting, one may be interested in answering the following questions:

- **Which predictors** are associated with the response? Identifying the **few important** predictors among a large set of possible variables
- **What is the relationship between** the response and each predictor? **positive** or **negative**; the relationship between the response and predictor may depend on values of other predictors – **interactions**
- Can the relationship between Y and each predictor be adequately summarized using a **linear equation or** is the relationship **more complicated**?
- often the true relationship is more complex and a linear model may not provide an accurate representation of the relationship

When prediction is more important than inference?

- predicting risk of credit default
- spam detection
- obtaining high response rate in direct-marketing campaign
- predicting future price of a stock

When inference is more important than prediction?

- marketing campaign: which media contribute most to sales? how expenses on marketing influence sales?
- churn prediction: which actions most efficiently decrease the risk of churn?
- revenue on user/client: which products most effectively boost average revenues of clients?
- brand of of a product: which are selected depending on price, store location, discount levels, competition price – efficient storage management, modelling demand elasticitt, etc.

Model selection depends on the aim

- Depending on whether our ultimate goal is prediction, inference, or a combination of the two, different methods for estimating f may be appropriate
- linear models allow for relatively simple and interpretable inference, but may not yield as accurate predictions as some other approaches
- In contrast, some of the highly non-linear approaches can potentially provide **more accurate predictions** for Y, but at the expense of a **less interpretable model** for which inference is more challenging

Methods of estimating f

General notation

- by n we will denote the number of data points called **observations**
- x_{ij} represents the value of the j-th **predictor**, or input, for observation i, where $i = 1, 2, \ldots, n$ and $j = 1, 2, \ldots, p$
- y_i represents the **response variable** for the i-th observation
- then the data consist of $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$ where $x_i = (x_{i1}, x_{i2}, \ldots, x_{ip})^T$
- The goal is to apply a statistical learning method to the data in order to **estimate** the unknown function f
- in other words, we want to find a function $\hat{f}$ such that $Y \approx \hat{f}(X)$ for any observation (X, Y).
- broadly speaking, most statistical methods for this task can be characterized as either **parametric** or **non-parametric**.

Parametric methods

Parametric methods involve a two-step model-based approach:

- First, we make an **assumption about the functional form**, or shape, of f.
- For example, a very simple assumption is that f is linear in X:

$$f(X) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p Xp$$

- After a model has been selected, we need a procedure that uses the data to **fit** or **train** the model
- in case of the linear model, we need to **estimate the parameters** $\beta_0, \beta_1, \ldots, \beta_p$, i.e. we want to find values of these parameters such that:

$$Y \approx \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p Xp$$

Parametric approach – (dis)advantages

- parametric (or model-based) approach **reduces the problem of estimating** f to estimating a set of parameters
- it is generally **much easier to estimate** a set of parameters than it is to fit an entirely arbitrary function f
- the disadvantage of a parametric approach is that the model we choose usually **does not match the true unknown form** of f
- if the chosen model is too far from the true f, then our estimate will be **poor**
- one can try to address this problem by choosing **flexible models** that can fit many different possible functional forms for f
- But in general, fitting a more flexible model **requires estimating a greater number of parameters**
- these more complex models can lead to a phenomenon known as **overfitting** the data, which essentially means they follow the errors, or noise, too closely.

Non-parametric methods

- non-parametric methods **do not make explicit assumptions** about the functional form of f
- instead they seek an estimate of f that gets **as close to the data points as possible**
- such approaches can have a **major advantage over parametric approaches**: by avoiding the assumption of a particular functional form for f, they have the potential to **accurately fit** a wider range of possible shapes for f
- but non-parametric approaches do suffer from an **important disadvantage**: since they do not reduce the problem of estimating f to a small number of parameters, a **very large number of observations is required** in order to obtain an accurate estimate for f

Trade-off between accuracy and interpretability

- if we are mainly interested in inference, then restrictive, i.e. **parametric models are much more interpretable**
- flexible non-parametric models can lead to such complicated estimates of f that it is **difficult to understand** how any individual predictor is associated with the response
- however, sometimes we are only interested in prediction, and the interpretability of the predictive model is simply not of interest
- we might expect that it will be best to use the most flexible model available
- **surprisingly, this is not always the case!**
- one can often obtain **better predictions using a less flexible method**
- this counterintuitive phenomenon has to do with the **potential for overfitting** in highly flexible methods

Supervised vs. unsupervised learning

Supervised vs. unsupervised learning

- most statistical learning problems can be classified as **supervised** or **unsupervised**
- **supervised** – for each observation of the predictor measurement(s) $x_i, i = 1, \ldots, n$ there is an associated response measurement y_i
- we wish to fit a model that relates the response to the predictors, with the aim of accurately predicting the outcome or better understanding the relationship
- **unsupervised** – for every observation $i = 1, \ldots, n$, we observe a vector of measurements x_i but **no associated response** y_i
- this is referred to as unsupervised because we lack a response variable that can **supervise** our analysis
- one can seek to understand (discover) the relationships between variables or observations – e.g. **cluster** variables of observations into homogeneous/correlated groups.

Regression versus classification

Regression versus classification

- variables can be either **quantitative** or **qualitative**
- **quantitative** (continuous) variables take on numerical values
- **qualitative** (categorical) variables take on values in one of K different classes, or categories
- We tend to refer to problems with a **quantitative response** as **regression problems**, while those involving a **qualitative response** are often referred to as **classification problems**

Regression versus classification – cont'd

- the distinction is not always that sharp
- for example **logistic regression** is used with a qualitative (two-class, or binary) response, so it is used to solve **classification** problems
- many methods can be used in **both cases** – for either quantitative or qualitative responses
- appropriate statistical method is usually selected based on the basis of type of **response** (quantitative or qualitative)
- it is less important whether the **predictors** are qualitative or quantitative
- most of the models can be applied regardless of thetype of predictors, provided that qualitative predictors are properly (re)coded

Thank you for your attention