1 19th of September 2018 — A. Frangioni

This course will deal with the optimization and numerical analysis of machine learning problems. We are not going to solve difficult problems (e.g. NP-hard problems), besides we try to find an efficient solution for simple ones (often **convex** ones), since we are dealing with huge amount of data.

Let us start with a warm up on machine learning problems.

1.1 Introduction to machine learning problems

Machine learning techniques are not as "young" as it might seem, the intuition has been there for ages, but we did not have enough calculus power. Machine learning algorithms are starting working well nowadays, thanks to the many improvements in computer performances; for this reason, it is becoming a more and more popular subject to study.

The main idea behind machine learning is to take a huge amount of data (e.g. frames of a video for object-recognition) and squeeze them, in order to process them. This intuitive concept is translated in mathematical terms as "building a **model**" that fits our data. As in practical engineering problems, people want to construct a model (a small sized representation of the large thing we want to produce in the end) and try to understand its behaviour, before actually build the thing. Take as an example the problem of designing a jet. It is not clever to start building the plane before designing a cheap prototype to better study its behaviour in the athmosphere.

The kind of models we want to build are cheap to construct and as close as possible to the real problem we are studying. In phisics, people try to find the best mathematical model to describe a real world phenomenon. The main issue is computation, since the more accurate the model, the more costly the prediction phase. Hence, a model is a good when it is a good tradeoff between accuracy and simplicity, namely it provides good prediction without incurring in slow computations.

The model, though, has to be parametric: we do not have only one model, we have a "shape" of a model, which is fit to our problem through the tuning of some parameters.

Example 1.1. *As an example, we are given three couples:* $f(x_1) = y_1$, $f(x_2) = y_2$, $f(x_3) = y_3$, *as shown in* **??**.

We need to make some choices: first, we need to decide the kind of model we believe is a good approximation of the objective function, say a linear model $f(x) = ax + b$. *After doing that, we are left with choosing its parameters (in order to pick a line among the whole family of functions), namely* a *and* b.

The aim of machine learning is to build a model that fits the data we are given and then to tune parameters in order to achieve a good "predicting power" on unseen inputs (in machine learning the technical term is "not **overfitting**", see **??**).

Another important characteristic of a good model is that it should not take too long to be built.

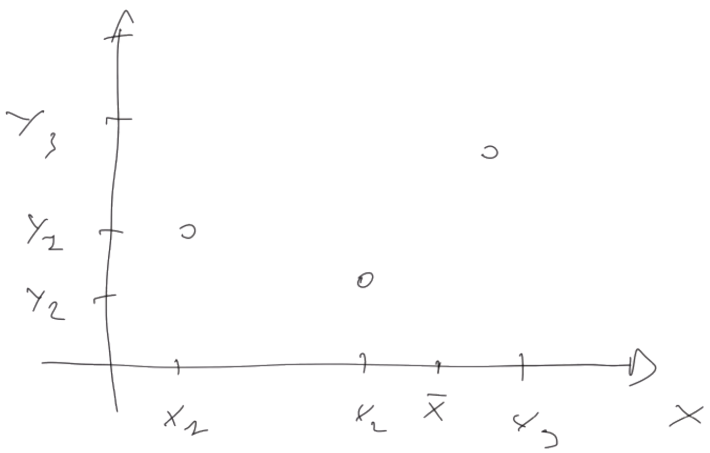

FIGURE 1.1: Geometic representation of the input. We are interested in finding a model that fits the input data and allows to predict $\bar{y}$ out of $\bar{x}$.

In this course we do not concentrate on the problem of finding the model that best fits our data, but we are already given a problem and a model and we only study its behaviour through its parameters.

1.2 Optimization

In the rest of this lecture we are going to understand what an optimization problem is, through some intuitive real world examples.

1.2.1 Linear estimation

Definition 1.1 (Linear model)**.** *Let $f : \mathbb{R}^n \to \mathbb{R}^n$ be the objective function. We call $\widetilde{f}(x) = \sum_{i=1}^{n} w_i x_i + w_0 = wx + w_0$ the **linear model** of f for a given set of parameters, which is a vector $w = (w_0, w_1, \ldots, w_n) \in \mathbb{R}^{n+1}$.*

How can we evaluate the "similarity" between our model and the objective function? Through computing the "error" or difference between the objective function and the model on each input. Under this assumption, the error function may be used to find the best parameters for our model, through a minimum problem:

Definition 1.2 (Least squares problem)**.** *Let $f : \mathbb{R}^n \to \mathbb{R}^n$ be the objective function, such that $f(x) = y$ and let Xw be our linear model. Then we can find the best values for vector $w \in \mathbb{R}^{n+1}$ by computing*

$$\min_{w} \|y - Xw\|$$

If the matrix X is invertible then the simple solution is $w = X^{-1}y$. The point is that this operation is very costly when dealing with a huge number of entries.

1.2.2 Low-rank approximation

We may want to approximate a matrix $M \in M(n, m, \mathbb{R})$ as the product between two smaller matrices: $A \in M(n, k, \mathbb{R})$ and a "fat and large" $B \in M(k, m, \mathbb{R})$ ($k \ll n, m$).

$$\boxed{M} \approx \boxed{A} \cdot \boxed{B}$$

This problem can be translated into a numerical analysis problem of the following shape

$$\min_{A,B} \|M - AB\|$$

1.2.3 Support vector machines

Let us take a decision problem: given a set of values of many parameters (aka variables) "label" a person as ill or healthy.

The geometric intuition in two dimensions is given by **??**. We would like to find the line that better splits the plane into two regions. The rationale here is to maximize the space between the line and the nearest points (called **margin**), in order to have a better accuracy.

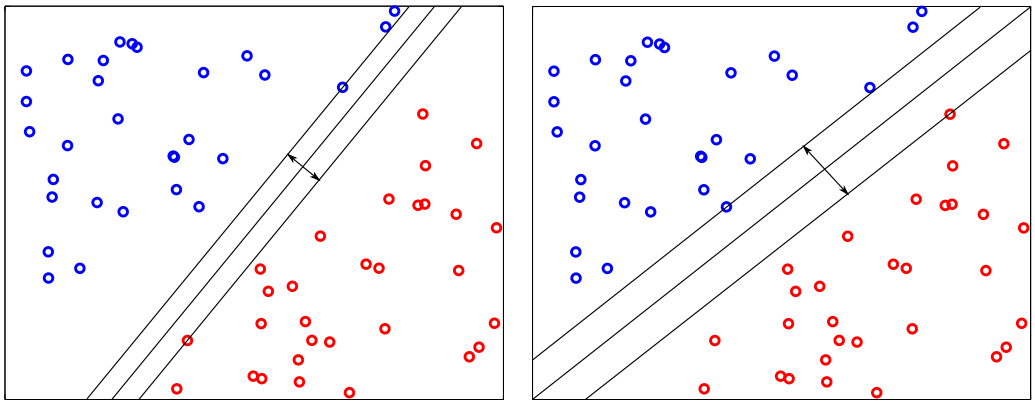

FIGURE 1.2: There are many possile boundaries that can be chosen as a model using many angular coefficients. Our best guess is the one that maximizes the distance between the line and the nearest points.

The maximum-margin separating hyperplane is the solution of

$$\min_{w} \{\|w_+\|^2 \ : \ y^i(w_+ x^i + w_0) \geq 1, \ i = 1, \ldots, m\}$$

where the margin is $\frac{2}{\|w_+\|}$, assuming any exists.

In fact, what happens most of the times is that there is no such line. To overcome this issue we introduce the concept of "penalty" that accounts for the number of points that are misclassified.

Definition 1.3 (Multi-objective optimization problem)**.**

$$\min_{w,\xi} \|w_+\|^2 + C \sum_{i=1}^{m} \xi_i$$

where $y^i(w_+x^i + w_0) \geq 1 - \xi_i,\ \forall i = 1, \ldots, m$ *and* $xi_i \geq 0,\ \forall i = 1, \ldots, m$*, where* C *is called* ***hyperparameter****.*

This formula formalizes the intuition that the approximated function may have a greater norm and lead to a very small misclassification error, or it could be the other way round. Both these solutions are acceptable and their performances depend only on the problem.

This whole course has the aim of presenting some techniques for solving efficiently **convex quadratic problems**, as the ones presented above.

Whenever we are able to solve the multiobjective optimization problem we are also able to solve what is called the **dual problem**, which is formally defined in **??** and has the following shape in our case:

$$\max_{\alpha} \left\{ \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i < x^i, x^j > \alpha_j \right\}$$

where $\sum_{i=1}^{m} y^i \alpha_i = 0$ and $0 \leq \alpha_i \leq C,\ \forall i = 1, \ldots, m$.