Statistical Learning

Notes (and extras) from Pierpaolo Brutti's course.

Sapienza - Universit? di Roma

February 2017 - June 2017.

Contents

Chapter 1

Non-Parametric Regression

1.1 Setup

$$(y, \underline{x}) \sim p(y, \underline{x})$$
$$\mathcal{R}(g) = \mathbb{E}[y - g(\underline{x})]$$
$$g^*(\underline{x}) = \underset{g}{\operatorname{argmin}}\, \mathcal{R}(g) = \mathbb{E}[y|\underline{x} = x]$$

The target in this case is the regression function.

In general, our target $g^*(.)$ is a complex function of $\underline{x}$.

The parameter we're trying to estimate live in a complicated parameter space $\mathcal{F}$. That's just some sort of function's space (plus some smoothness).

To get good predictions, we need to estimate well $g^*(.)$ that is a point in some well-behaved function space.

1.2 Methods

1. Orthogonal expansions.
    - Linear regression in high-dim.
2. Local average.
    - Kernel methods. Local Poly. K-nn.
3. Kernel methods (The other kernel)
    - Ridge regression.

1.3 Orthogonal expansion in hilbert spaces

1.3.1 Vector spaces

Chapter 2

Splines

We need to represent a function $g : [a, b] \to \mathbb{R}$.

2.1 Informally

A piecewise polynomial plus smoothness constraints on the joints.

Image 1

2.2 Definition: k-order spline

A function $f : \mathbb{R} \to \mathbb{R}$ is a **k-order** spline relative to the knot set $\{t_1, t_2, ..., t_n\}$ iff $f(.)$ is a polynomial of order k on each interval $(-\infty, t_1], (t_1, t_2], ..., (t_{m-1}, t_m], (t_m, \infty)$ and $f^{(j)}(.)$[1] is continous at the knots for each $j \in \{0, 1, ..., k-1\}$.

1. Comment

   Linear an cubic splines are the famous ones.

Imagine that you pick the space of k-order splines on a arbitrary set on knots as parameter space where to search for an estimator of our regression function $\Rightarrow$ we need a set of function (basis) to generate this space.

$\{\phi_1(x), \phi_2(x), \dots\}$ such that each time we take linear combinations of these functions we get back a k-th order spline over some set of knots.

2.3 Options

1. A simple one: Truncated power basis.

   $\phi_1(x) = 1\ \phi_2(x) = x\ \phi_3(x) = x^2\ \dots\ \phi_{k+1}(x) = x^k$ therefore $\phi_{k+1+j}(x) = (x - t_j)_+^k$ for $j \in \{1, \dots, m\}$[2]

   - Any k-order spline on $\{t_1, \dots, t_m\}$ can be represented as a linear combination appropriate Truncated Power Basis.
   - This representation is usually unstable... we can do better: **Natural Spline**.

[1] j-th derivative

[2] Remember $(x)_+ = \max(0, x)$

Represent our unknown function $g(.)$ in TPB.

$$g(x) = \sum_{j=1}^{k+1+m} \alpha_j \phi_j(x)$$

[3].

Now we have data $\{(y_i, x_i)\}_{i=1}^n$. So let, $\Phi = [\phi_j(x_i)]_{ij}$ of size n x $k+1+m$ not orthogonal design matrix. $\Rightarrow$ for k and $\{t_1, \ldots, t_m\}$ **fixed** (but actually we have to choose them base on data) we are back to linear regression. Meaning that we can estimate $\alpha \in \mathbb{R}^{k+1+m}$ as

$$\hat{\alpha} = argmin_{\alpha \in \mathbb{R}^{k+1+m}} ||y - \Phi_m \alpha||$$

Every alpha and y with piso

Therefore via OLS, $\hat{\alpha} = \left(\Phi^T \Phi\right)^{-1} \Phi^T y$

Perfectly fine problem if we choose the number of knots m their position and the spline order k appropriatly.
Tuning parameters in regression splines are:

- m: # knots.
- $t_1, \ldots, t_m$: Knots position.
- k: spline order.

This is called REDUCED RANK REGRESSION.

For example,

- The higher k the more flexible the spline the higher its variance as an estimator.
- Same for m
- Knots position: You'd like to habe more knots... where the true function is more complex/wiggly.

image 2

Typical choices

- $k = 3$
- $m \rightarrow$ GCV.
- The knots are placed on suitable quantities of the x's.

Other choices

- Adaptive or free knots splines $\rightarrow$ try to optimize the position and numbers of the knots. (k fixed$). Negative side: time consuming to be solved by optimization or bayesian reasoning (MCMC)

[3] $\phi_j(x)$ TBT not ortho base

Chapter 3

Support Vector Machines

SVM are in general linear classifiers. We obtain them as a penalized hinged-loss problem.

3.1 The End

$y_i \in \{-1, 1\}$

We want to find a linear classifier $h(\underline{x}) = sign\left(H_\beta(\underline{x})\right)$ then $H_\beta(\underline{x}) = \underline{x}^T\beta$ [1]

$$\hat{\beta} = argmin_{all\ \beta} \sum_{i=1}^{n} [1 - y_i H_\beta(\underline{x})]_+ + \lambda||\beta||_2^2$$

3.2 The beginning

3.2.1 Geometrical construction/motivation

With $k = 2$.

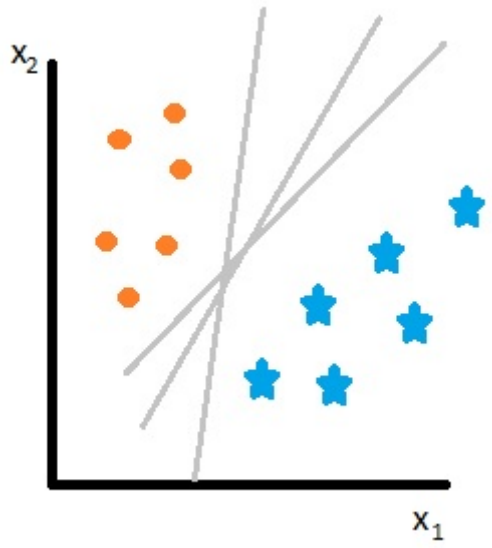

Figure 3.1: Linear separated

[1] poner underline en beta y x

Exist at least one hyperplane that perfectly separates our two classes.

We need to pick the optimal one and we gonna do that taking the one that brings the LARGEST MARGIN between the two classes

Image 2

Signed from the boundary of each data point.

Decision boundary

$\left\{\underline{x} \in \mathbb{R}^k / H_\beta(\underline{x}) = 0\right\}$

3.2.2 Geometry 101

$H_\beta(\underline{x}) = \beta_0 + \underline{x}^T\beta$

$\underline{x}, \underline{x}' \in P \rightarrow \underline{x} - \underline{x}' \in P$

We want $< n, \underline{x} - \underline{x}' >= 0 \rightarrow \underline{x}^T n = c \rightarrow n = \beta \rightarrow n = \frac{\beta}{||\beta||_2}$

$w = \underline{x}^0 - \underline{x}'$ then $cos\,(\theta) = \dfrac{d}{||w||_2}$

$\frac{||n||_2}{||n||_2}||w||_2 cos\,(\theta) = d$

$cos\,(\theta) = \frac{<n,w>}{||n||_2||w||_2}$

Finally,

$d = \frac{<n,w>}{||n||_2} \rightarrow d = \frac{H_\beta(\underline{x}_0)}{||\beta||_2}$

The signed distance of any point $\underline{x}_0$ from the plane $H_\beta(\underline{x})$ is equal to:

$$\frac{H_\beta(\underline{x}_0)}{||\beta||_0}$$

Assume that $H_\beta(\underline{x})$ is any separating hyperplane the y_i and $H_\beta(\underline{x}_i)$ have the same sign so the distance of any datapoint $\underline{x}_i$ from the boundary then is equal to $\frac{y_i H_\beta(\underline{x}_i)}{||\beta||_2}$

This leads to the following optimization problem to choose the largest margin.

Cases

$\max\limits_{\beta_0,\beta} M$ s.t $\frac{y_i H_\beta(\underline{x}_i)}{||\beta||_2} \geq M, \forall i \in \{1, \ldots, n\}$

By a scaling argument

$\min\limits_{\beta_0,\beta} ||\beta||_2$ s.t $\frac{y_i H_\beta(\underline{x}_i)}{||\beta||_2} \geq 1, \forall i \in \{1, \ldots, n\}$

Remarks

1. Quadratic optimization problem (Quadratic object function with linear constraints).
2. Standard techniques from convex optimization to solve it.

Once we have a constrained optimization problem, usually we go “lagrangian”.

3.2.3 Primal Langrangian

$$\min_{\beta_0,\beta} \frac{1}{2}||\beta||_2^2 + \sum_{i=1}^{n} \gamma_i \left[1 - y_i H_\beta\left(x_i\right)\right] \text{ s.t } \gamma_i \geq 0, i \in \{1, \ldots, n\}$$

Differentiate the PL to get the KKT optimality condition over β and γ.

We have that:

1. $\beta = \sum_{i=1}^{n} \gamma_i y_i x_i = \sum_{i=1}^{n} \alpha_i x_i$
2. $\sum_{i=1}^{n} \gamma_i y_i = 0$ we name $\alpha_i = \gamma_i y_i$

Remember $\gamma_i \geq 0$ and $y_i \in \{-1, 1\}$ then for the sum to be equal to zero... some of the $\alpha_i = 0$ and if y is the set of active vector or support vector having $\alpha_i \neq 0$ then $\beta = \sum_{i \in y} \alpha_i x_i$

For computational reasons it is usually better (more efficient) to solve the dual lagrangangian. You get it inserting the optimal solution I wrote before into the primal.

$$\beta = \sum_i \gamma_i y_i x_i$$

Then we want to: $\max_{\gamma_1, \ldots, \gamma_n} \sum_{i=1}^{n} \gamma_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \gamma_i \gamma_j \left(y_i y_j\right) \left(x_i^T x_i\right)$ with $\gamma_i \geq 0, \forall i \in \{1, \ldots, n\}, \sum_{i=1}^{n} \gamma_i y_i$

1. $n < k$
2. It depends only on $< \underline{x}_i, \underline{x}_j >_{\mathbb{R}^k}$

3.2.4 Moving from linearly separable to non linearly separable case

Soft-margin:

$\min_{\beta_0, \beta} ||\beta||_2$ s.t $\frac{y_i H_\beta(\underline{x}_i)}{||\beta||_2} \geq 1, \forall i \in \{1, \ldots, n\}$

We want to allow some violations. In optimization you just augment the original problem with as many slacks (variables) as needed to soften the constraints.

$\min_{\beta_0, \beta} ||\beta||_2$ s.t $\frac{y_i H_\beta(\underline{x}_i)}{||\beta||_2} \geq 1 - \epsilon_i, \forall i \in \{1, \ldots, n\}, \epsilon_i \geq 0 \forall \in \{1, \ldots, n\}, \sum_{i=1}^{n} \epsilon_i \leq B$

1. B (overal budget to account for violations) becomes a tuning parameter.
2. The solution to this relaxed problem has the same structure as before $\beta = \sum_{i \in y} \alpha_i x_i$ but now y is the support set $= supportvectors \cup violators$
3. If B is larger we get more stable solutions (involving a larger support set) and lower variance.
4. We need $\epsilon_i > 0$ when our margin is less than 1 an we change by an amount equal to the sum of the slacks
$$\sum_{i=1}^{n} \left[1 - y_i H_\beta\left(x_i\right)\right]_+ \leq B$$

$\min_{\beta_0, \beta} ||\beta||_2$ s.t $\sum_{i=1}^{n} \left[1 - y_i H_\beta\left(x_i\right)\right]_+ \leq B$

Finally,

$$\min_{\beta} ||\beta||_2^2 + \gamma \sum \left[derivar\right]_+, \lambda = \frac{1}{\gamma}$$

Appendix A

Hilbert spaces

Appendix B

Convex and concave funcions

Appendix C

Norms

Appendix D

Gradients

Appendix E

Mathematical programming