ECON484 Machine Learning

4. Linear Regression, Logistic Regression, Ridge and Lasso

Lecturer:
Bora GÜNGÖREN
bora.gungoren@atilim.edu.tr

Linear Regression

- Given a set of observations $\{x_n\}$ with corresponding target values $\{t_n\}$ the goal is to predict the value of t for a new x.
- We conceptualize this as finding the parameters $\{w_D\}$ of a D-dimensional function y(x,w).
  - For any given x, this is a function of $\{w_D\}$ and for a given $\{w_D\}$ this is a function of x.
  - So when we choose a particular $\{w_D\}$, we will have an estimator in the form t = y(x).

Linear Regression

- We conceptualize this as finding the parameters $\{w_D\}$ of a D-dimensional function y(x,w).
  - The form of y(x,w) need not be linear.
  - The initial and fixed assumption we make about the form of y(x,w) is like this.
  - The functions $\varnothing_j(x)$ are called the basis functions.
  - It is also common to define a dummy basis function $\varnothing_0(x)=1$.

$$y(\mathbf{x}, \mathbf{w}) = w_0 + \sum_{j=1}^{M-1} w_j \phi_j(\mathbf{x})$$

Linear Regression

- By using non-linear basis functions, we allow the function y(x,w) to be **non-linear of x, but it will still be linear of w**.
  - This is an important aspect. Because in linear regression, we try to find a particular $\{w_D\}$, and the fact that y(x,w) is linear of w, helps in the assumptions.
  - Once we find a particular $\{w_D\}$, calculating y for a given x value is simple computation, however complex and non-linear the basis functions are.
    - Polynomial regression is a special case where x is one dimensional, and the basis function is in the form of $\varnothing_j(x)=x^j$.

Linear Regression

- By using different basis functions, one can achieve a wide variety of properties in the function y(x,w).
  - A neat trick is to define a basis function that would have different dominant aspects in different parts of the number-space.
  - This can be achieved with **a partially defined function** or with **multiplication with a decaying exponential** or any other way you can think of.
  - This trick is called a **spline** and usually involves an additional variable which determines if a basis function is dominant (ie. governs) in a particular part of number-space.

Linear Regression

- The term spline comes from the flexible **spline devices** (also known as a French curve) used by shipbuilders and draftsmen to draw smooth shapes.
  - It is very interesting to observe how the hand movements in using a French curve is similar to how the mathematics work in splines.
  - Please watch – https://youtu.be/wgATTmC0JSQ
  - The **curvature changes** in each section of the device. So when you switch from one part to the other, another function that gives that particular curvature dominates.
    - But you need **an initial set of points to cover**.

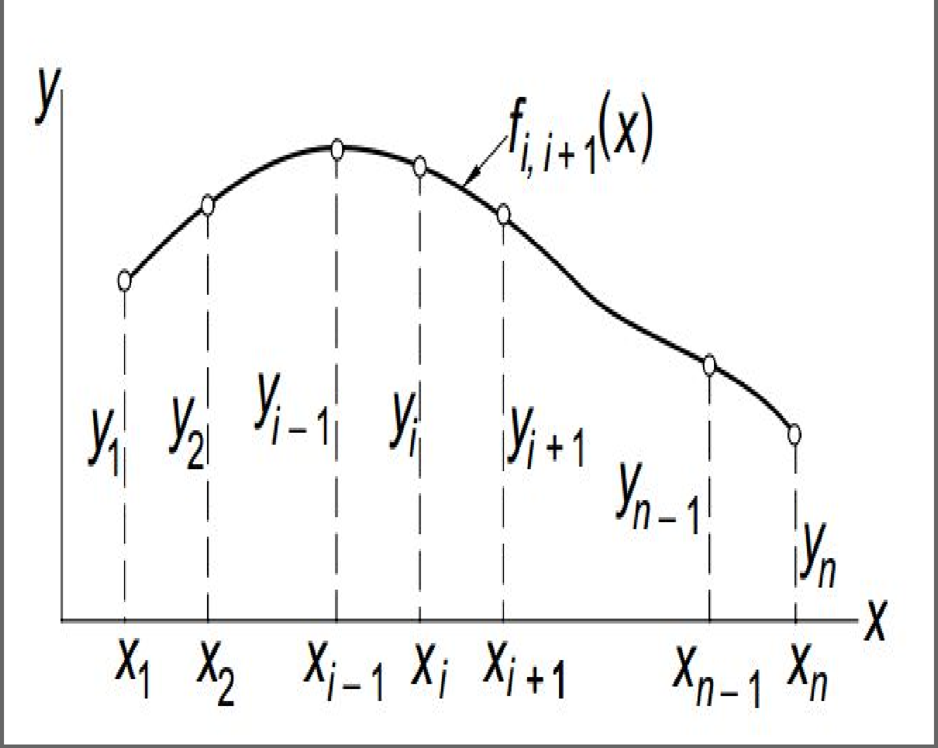
Linear Regression

- Some well known spline functions exist. We will cover the most basic one here.
- “Natural” cubic-spline.
  - The word "natural" means that the second derivatives of the spline polynomials are set equal to zero at the endpoints of the interval of interpolation.
  - This falls into the category of piece-wise **interpolation**.
  - Interpolation means the function **must** pass through the data points (also known as **knots**).
  - Interpolation with a large number of knots results in a high-degree polynomial (ie. complexity of model)
  - Piece-wise interpolation allows the use of multiple lower-degree polynomials to be dominant at regions.

$$S_j(x) = a_j + b_j(x - x_j) + c_j(x - x_j)^2 + d_j(x - x_j)^3$$

Linear Regression

- "Natural" cubic-spline.
  - To have a third degree polynomial, you need 3+1=4 points.
  - So for the interval (x_i, x_{i+1}) we choose a polynomial of degree <= 3 such that itself, and its first two derivatives are continuous.
  - The value of the second derivative at x_i is called k_i, ie. $k_i=f''_{i,i+1}(x_i)$. We use k for knots.
  - When we choose the consecutive polynomial functions, we just let k_i values to match at connection points. This is called **slope continuity**.
  - For the whole derivation of the method please visit – https://bit.ly/3u2TI8e



Linear Regression

- Theoretical discussion about linear regression and use of splines is **mostly independent of the particular choice of basis function set**.
  - For sake of simplicity, examples in sources are usually given with the simplest mathematical form possible.
  - However, in applied work the functions used may be very complex.

Linear Regression

- Why have mathematicians invented the spline trick?
  - A practical problem with linear regression is the **size** of the problem involved with the number of knots.
  - At some point you need to calculate the inverse of a matrix (or do something equivalent to it). And as the dimension of this matrix increases, it becomes problematic.
  - If you are **memory constrained** and/or **time constrained**, you simply can not do this.
  - Splines help us reduce one large problem into a series of smaller problems, **very much easing the memory-constraint**.

Linear Regression

- For programming, there are many libraries to use.
  - A really nice demonstration in R - https://bit.ly/3iVc1Wm
  - A short introduction to use of both linear regression and splines in Python - https://bit.ly/3u0JvsF
  - A review paper that has an attached table of popular R packages regarding splines - https://bit.ly/3DvMYTo
- Please try to observe memory use when following these demonstrations.

Linear Regression

- Another solution to the size problem is **sequential learning** (also known as **on-line algorithms**)
  - The data points are introduced one at a time, and the the model improves just a little after each introduction.
  - This is particularly useful when **you are required to make a prediction before you can review and process all data points**.
  - The easiest method of implementing sequential algorithms is using a **gradient-descent** approach.
    - After each iteration the parameters $\{w_N\}$ are **updated by a fraction of the computed error**.
    - You make decisions on how to calculate error (**update function**), and what fraction to use (**learning rate parameter)**.
    - You also need to have a good **initialization method**.

Linear Regression

- Another solution to the size problem is **sequential learning** (also known as **on-line algorithms**)
  - We want the updated parameters $\{w_N\}$ so that
    - We have **less error** (smaller loss) on the current sample.
    - The parameters **do not fluctuate** (ie. Stay close to the previous parameters)
  - **Widrow-Hoff Algorithm** (1960) is the most famous gradient-descent approach.
    - This algorithm is also known as **one of the first** single layer neural network implementations.
    - An implementation example – https://bit.ly/3u0NIwH
    - A nice discussion (also comparing to the linear perceptron model) - https://bit.ly/3qWbV5a

Linear Regression

- Note that on-line algorithms can also be used for classification problems.
  - Randomized weighted majority
  - Winnow algorithm

Linear Regression

- The assumption that the basis functions are fixed before observing the training data is problematic in itself.
  - With larger datasets, this assumption requires the number of basis functions to increase, and makes the procedure difficult (if not impossible) to continue.
- We are lucky that **most real world data is actually very suitable for the use of splines**.
  - There always appear many strong **localized** correlations among variables.
  - In multi-variate problems (multiple x’es, not w’s) the direction on which you approach may also be important. There may be **strong correlations along a particular path**.
  - More advanced techniques based on these easy to express facts are **support vector machines** and **relevance vector machines**.

Logistic Regression

- Logistic regression is a process of modeling the probability of a discrete outcome given an input variable.
  - The most common logistic regression models a binary outcome, ie. true/false, yes/no.
- The primary difference between linear regression and logistic regression is that logistic regression's range is bounded between 0 and 1.
- In addition, as opposed to linear regression, **logistic regression does not require a linear relationship between inputs and output variables**.

Logistic Regression

- The most common example used in teaching logistic regression is the **hours studied vs pass/fail** example.
    - 20 students study for the same exam. They study between 0 to 6 hours (data actually including zero hours).
    - Some of them pass, some of them fail.
    - Studying the data set, how does the amount of study (hours spent) affect the probability of passing?
        - Hours spent – Explanatory variable (x)
        - Pass/fail – Categorical variable (y)
    - The logistic function (of x) is the in the form presented right (above), where s **is used to scale the values** and **μ is a midpoint value of all x-values. in the sample (but not the mean of x)**.
    - So the term in the logistic function could be expressed also as presented right (below).

$$p(x) = \frac{1}{1+e^{-(x-\mu)/s}}$$

$$p(x) = \frac{1}{1+e^{-(\beta_0+\beta_1 x)}}$$

Logistic Regression

- We try to find the best fit parameters {β} so that the likelihood function L that measures **the probability of each prediction being correct** is maximized.
  - Because these are probabilities the likelihood function is a multiplication of individual probability functions.
  - The underlying mathematics uses typical calculus techniques, Newton's method, Lagrange multipliers, etc. and **needs to converge**.
- In some cases logistic regression **will not converge**.
  - This usually indicates the data set is not **meaningful**.
  - Another explanation is that of **fabricated data which is too perfect to predict**.
    - In this case logistic regression model will have infinity values and thus will not complete its computation on a computer.
    - This may appear in non-fabricated data as well. The problem is then termed **complete or quasi-complete separation**.
- Logistic regression **does not work well with missing values**. Removing data points with missing values in categorical models always has the risk of omitting a whole category.

Logistic Regression

- There are many libraries covering logistic regression:
  - For R examples - https://bit.ly/3DFikXF
  - For Python examples - https://bit.ly/3IVClu4
  - Regarding complete or quasi-complete separation, read this – https://bit.ly/3J3cuAy and this - https://bit.ly/3iZ4oyg

Ridge and Lasso

- Recall that when discussing bias-variance trade-off, a situation of **low bias and high variance is termed as over-fitting**.
- One family of techniques used to avoid over-fitting is regularization.
  - Ridge and Lasso are regularization techniques.
  - They are used commonly in regression.

Ridge and Lasso

- Ridge regression is a method of estimating the coefficients of multiple-regression models **in scenarios where linearly independent variables are highly correlated**.
  - It is one of the most popular short-cuts when handling large datasets.
  - Introduced by Hoerl and Kennard in 1970, after a decade of research.
  - This method performs L2 regularization, by adding adding a squared magnitude of coefficient as **penalty term** to the loss function.

Ridge and Lasso

- Lasso Regression (Least Absolute Shrinkage and Selection Operator) is a popular L1 regularization technique.
  - Lasso adds absolute value of magnitude of coefficient as **penalty term** to the loss function.
- The key difference between Ridge and Lasso is that **Lasso shrinks the less important feature’s coefficient to zero thus, removing some feature altogether**.
  - This works well for feature selection in case we have a huge number of features.
  - Once the number of features are reduced this way, typical techniques such as k-fold cross-validation are now practically applicable.

Ridge and Lasso

- Ridge

$$\sum_{i=1}^{n}(y_i - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} \beta_j^2$$

- Lasso

$$\sum_{i=1}^{n}(Y_i - \sum_{j=1}^{p} X_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j|$$

Ridge and Lasso

- Because they are common, Ridge and Lasso are available in typical libraries
  - Python example - https://bit.ly/3iVBPS6
  - A really nice tutorial in R with an interesting data set - https://bit.ly/3K9PCRm

Questions?

CONTACT:
bora.gungoren@atilim.edu.tr