Most common Machine Learning Algorithms

Hamza Tazi Bouardi

April 10, 2018

1 Linear Regression

1.1 Classic Least Square Estimator

Let's consider integers $p, n \geq 1$ (n being the number of observations, p the number of regressors or features) and a vector $\beta \in \mathbb{R}^p$ and a matrix of features $X \in \mathbb{R}^{n\times p}$. We consider the multilinear regression problem :

$$Y = X\beta + \varepsilon \tag{1}$$

Where ε is a noise (residuals) following a centered normal distribution *i.e.* $\varepsilon \sim \mathcal{N}(0, \sigma^2 I_n)$ with σ independent from our features (constant) and those residuals supposed uncorrelated, and $Y \in \mathbb{R}^n$ is the dependent variable we want to explain.
Considering the *Residual Sum of Squares (RSS)* defined by:

$$\boxed{RSS = ||Y - X\beta||_2^2 = \sum_{i=1}^{n}(Y_i - X_i\beta)^2} \tag{2}$$

Where $X_i = (X_{i1}, ..., X_{ip})$, our goal is to minimize this RSS. The solution to this problem is defined as:

$$\widehat{\beta} = \underset{\beta}{\operatorname{argmin}}\ ||Y - X\beta||_2^2 \tag{3}$$

If the matrix $X^T X$ isn't singular (*i.e.* if it is injective, *i.e.* invertible) then the solution to this optimization problem is $\boxed{\widehat{\beta} = (X^T X)^{-1} X^T Y}$
Proof.

$$\begin{aligned}
&\frac{\partial RSS}{\partial \beta} = 0 \\
&\Leftrightarrow \frac{\partial \langle Y - X\beta, Y - X\beta\rangle}{\partial \beta} = 0 \\
&\Leftrightarrow \langle -X, Y - X\beta\rangle + \langle Y - X\beta, -X\rangle = 0 \\
&\Leftrightarrow X^T(Y - X\beta) = 0 \\
&\Leftrightarrow \beta = (X^T X)^{-1} X^T Y \ \ if\ (X^T X)\ invertible.
\end{aligned}$$

□

The properties of this $\widehat{\beta}$ are very useful (confidence intervals, Student tests etc.)

1. $\widehat{\beta}$ is not biased *i.e.* $\mathbb{E}[\widehat{\beta}] = \beta$ and $\mathbb{V}[\widehat{\beta}] = \sigma^2(X^T X)^{-1}$ *because* $\mathbb{V}[CZ] = C\mathbb{V}(Z)C^T$, C being a matrix, *i.e.* $\widehat{\beta} \sim \mathcal{N}(\beta, \sigma^2(X^T X)^{-1})$.

2. $\frac{n\widehat{s}^2}{\sigma^2} \sim \chi^2(n-p)$ where $\widehat{s}^2 = \frac{n-p}{n}\widehat{\sigma}^2 = \frac{1}{n}||y - X\widehat{\beta}||_2^2$.

1.2 Ridge Regression

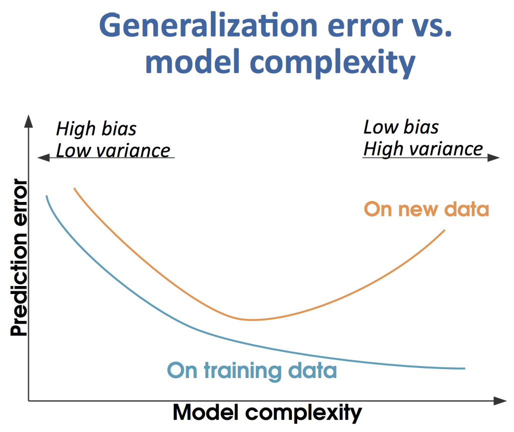
The general problem is as follows: given the properties of our former classic Least Squared estimator, even if this very estimator does great on training sets, nothing tells us it will do great on new data. The whole point is to make a compromise between ***Bias*** (difference between the expected value of the estimator and the true value being estimated) and ***Variance*** (deviation from the expected value of the estimates) and we have the following decomposition:

$$\boxed{MSE(f(x)) = \mathbb{E}[(f(x) - y)^2] = Bias^2(f(x)) + \mathbb{V}[f(x)])}$$

So adding a little bias to our estimator might significantly decrease the variance and thus make it cope well with new data. This is called the **Bias-Variance Tradeoff**, keeping in mind that:

- **High bias** can cause *Underfitting*: not fitting to the training data (not capturing the good amount of correlations), i.e. failing to capture relevant patterns, again leading to less accurate predictions.

- **High variance** can cause *Overfitting*: fitting perfectly to the training data but doing poorly on test data or new data. Capturing spurious patterns that won't recur in the future, leading to less accurate predictions.



If the elements of $\widehat{\beta}$ (*i.e.* $(\widehat{\beta}_j)_{1\leq j\leq p}$) are unconstrained, they can explode (especially if X^TX if close to a non-invertible matrix, which is the case if the $rg(X) < p$, p being the number of columns of X) and are therefore susceptible to very high variance, so in order to control this variance we ***regularize*** the coefficients and allows us to avoid the singularity problem of X. In the Ridge method, we impose the so-called **Ridge constraint**, and our problem thus becomes (we should be using the Lagrangian and KKT method but it is equivalent).

$$\begin{aligned} \widehat{\beta} &= \underset{\beta}{\text{argmin}}\ ||Y - X\beta||_2^2 \ \ s.t.\ ||\beta||_2^2 \leq t,\ t > 0 \\ &= \underset{\beta}{\text{argmin}}\ ||Y - X\beta||_2^2 + \lambda||\beta||_2^2,\ \lambda \geq 0 \end{aligned} \tag{4}$$

This last element is called **Penalized Residual Sum of Squares** (PRSS), and we obtain as a result (note that *s.t.* means *"subject to"*):

$$\boxed{\widehat{\beta}_\lambda^{ridge} = (X^TX + \lambda I_p)^{-1}X^TY} \tag{5}$$

This estimator here always exists because **the matrix $M = (X^TX + \lambda I_p)$ is always invertible.** Indeed, let's remember that $Sp(X^TX) \subset \mathbb{R}_+$ and that if there is a problem with the matrix's rank, the eigenvalues of X^TX start to be very small from a certain point. M and X^TX have the same eigenvectors, but $Sp(M) \subset \mathbb{R}_+^*$ because we're adding the $\lambda > 0$ and so M is always invertible (since the determinant is the product of the eigenvalues to the power of their multiplicity).

Properties of $\widehat{\beta}_\lambda^{ridge}$:

1. $\mathbb{E}[\widehat{\beta}_\lambda^{ridge}] = -\lambda(X^TX + \lambda I_p)^{-1}\beta$
2. $\mathbb{V}[\widehat{\beta}_\lambda^{ridge}] = \mathbb{V}[(X^TX + \lambda I_p)^{-1}X^T(X\beta + \varepsilon] = \sigma^2(X^TX + \lambda I_p)^{-1}(X^TX)(X^TX + \lambda I_p)^{-1}$

Since the eigenvalues of $M = X^TX + \lambda I_p$ are greater than those of X^TX, the variance of $\widehat{\beta}_\lambda^{ridge}$ is smaller (since it implies the inverse of M) than the one of the classic Least Square estimator (which implies the inverse of X^TX). As a conclusion, we've added a little of bias in order to lose some variance, which is exactly what we wanted. More specifically, the **Ridge Regression shrinks the coefficients of low-variance components** and it also **gives the same weights to similar variables**.

Concerning the choice of λ, we could either calculate it (depends on the constraint we fix and thus depends on t), but standard practice consists in using cross-validation between the different λ's that each give a different $\widehat{\beta}_\lambda^{ridge}$ and **choose the λ for which we have the lowest Cross-Validation error**.

In practice, we also usually standardize Y as well as X $following$ $\widetilde{X_j} = \frac{X_j - \overline{X_j}\mathbf{1}_n}{\widehat{\sigma_j}}$ where $\widehat{\sigma_j} = \frac{1}{n}\sum_i (X_{ij} - \overline{X_j})^2$ (*N.B*: X_j corresponds to the line j of X)

1.3 LASSO Regression

The LASSO Regression (*Least Absolute Shrinkage and Selection Operator*) consists in a regularization of the Residual Sum of Squares but with the ℓ_1 norm instead of the ℓ_2 norm. The optimization problem is thus:

$$\begin{aligned}\widehat{\beta}_\lambda^{lasso} &= \underset{\beta}{\operatorname{argmin}}\ ||Y - X\beta||_1 \ \ s.t.\ ||\beta||_1 \le t,\ t > 0 \\ &= \underset{\beta}{\operatorname{argmin}}\ ||Y - X\beta||_2^2 + \lambda||\beta||_1,\ \lambda \ge 0\ and\ ||\beta||_1 = \sum_{i=1}^{p}|\beta_i|\end{aligned} \tag{6}$$

Often, we believe that many of the β_j's should be 0 and we hence seek a **sparse solution**. Large enough λ (or small enough t) will set some coefficients of β **exactly equal to** 0 so the **LASSO will perform a model selection for us**.

Note that model selection can be done in the frame of Least Square classic estimator with analytical tools, such as *AIC, BIC or Mallow's C_p* criteria (we penalize a certain error, either the *Log-Likelihood* for the two former of the *Residual Sum of Squares* for the latter, and we minimize the whole thing) and we can perform a stepwise selection to do so (or a backward, going from whole model to the one minimizing the criterion).

1.4 Elastic Net Regression

There are a few limitations to the variable selection of LASSO:

- If $p > n$, the lasso selects at most n variables, which means the selection is bounded by the number of samples (problematic for gene selection for example).
- The LASSO fails to do grouped selection: it tends to select one variable from a group of variables which are highly correlated pairwise and ignore the others (while Ridge manages to give similar weights to the members of those groups).
- Ridge outperforms LASSO in the usual $n > p$ scenario.

Therefore the Elastic Net regularization consists in mixing both ℓ_1 and ℓ_2 penalizations in order to get the following optimization problem:

$$\begin{aligned}\widehat{\beta}^{E-N}_{\lambda_1,\lambda_2} &= \underset{\beta}{\operatorname{argmin}}\ ||Y - X\beta||_2^2 \ s.t.\ \alpha||\beta||_1 + (1-\alpha)||\beta||_2^2 \leq t \ for\ some\ t,\ \ \alpha \in [0,1] \\ &= \underset{\beta}{\operatorname{argmin}}\ ||Y - X\beta||_2^2 + \lambda_1||\beta||_1 + \lambda_2||\beta||_2^2,\ \ \lambda_1,\lambda_2 \geq 0 \quad and \quad \alpha = \frac{\lambda_1}{\lambda_1+\lambda_2}\end{aligned} \tag{7}$$

The ℓ_1 part of the penalty generates a sparse model while the ℓ_2 (quadratic) part of it:

- Removes the limitation on the number of selected variables.
- Encourages grouping effect (we can shrink together coefficients of correlated variables).
- Stabilizes the ℓ_1 regularization path (we get more non-zero coefficients than LASSO but with smaller amplitudes).

2 Classification

2.1 K-Nearest Neighbors

The $K-NNAlgorithm$ is an Instance-based learning algorithm (it computes the label for a new instance based on its similarity with the stored instances) which goal is to predict the class of the most frequent label among the K neighbors of a point.

2.1.1 Distances and Similarities

To this end we need to compute distances and similarities between different instances ($e.g$ observations, data points). As a reminder, a distance is an function that satisfies 3 properties: symmetry, triangular inequality and separability. The 3 most common distances are the following:

- **Euclidian Distance**:

$$(x^1, x^2) = \|x^1 - x^2\|_2 = \sqrt{\sum_{i=1}^{p} (x_i^1 - x_i^2)^2}$$

- **Manhattan Distance**:

$$d(x^1, x^2) = \|x^1 - x^2\|_1 = \sqrt{\sum_{i=1}^{p} |x_i^1 - x_i^2|}$$

- **Minkowski Distance**:

$$d(x^1, x^2) = \|x^1 - x^2\|_q = \left(\sum_{i=1}^{p} (x_i^1 - x_i^2)^q\right)^{1/q}$$

The **Similarity** between instances is defined as $s = \frac{1}{1+d}$, and we also define **Pearson's Correlation** as follows:

$$\begin{aligned}
\rho(x, z) &= \frac{\sum_{i=1}^{p}(x_j - \overline{x})(z_j - \overline{z})}{\sqrt{\sum_{i=1}^{p}(x_j - \overline{x})^2}\sqrt{\sum_{i=1}^{p}(z_j - \overline{z})^2}} \\
&= \frac{\sum_{i=1}^{p} x_j z_j}{\sqrt{\sum_{i=1}^{p} x_j^2}\sqrt{\sum_{i=1}^{p} z_j^2}} \text{ assuming the data is centered} \\
&= \frac{\langle x, z\rangle}{||x|| \cdot ||z||} = \cos\theta, \ \theta \text{ being the angle between x and z}
\end{aligned}$$

2.1.2 Decision boundary of K-NN

A Decision boundary is the line separating the positive from the negative regions (*e.g.* to the classification test). We usually use the **Voronoi tesselation**. The **Voronoi cell** of x is the set of all points of the space closer to x than any other point of the training set and is represented by a polyhedron. The **Voronoid tesselation** of the space is the union of Voronoi cells and it defines the decision boundary of the 1-NN, and the K-NN also partitions the space but in a more complex way.

2.1.3 Choice of K

The choice of K is a little empirical: K shouldn't be too small (because would be too noisy) and not too large (because would be computationally too intensive). In practice, K is often set by cross-validation, and a heuristic method would be to start by setting $K = \lfloor \sqrt{n} \rfloor$ where n is the number of observations.

2.1.4 Advantages and Drawbacks

There are many advantages to K-NN: training is very fast and it keeps the training data, it's rather robust to noisy data (averages K votes) and can learn complex functions.

Still, it requires quite a lot of memory and prediction can be slow (Complexity of labeling 1 new data point is in $\mathcal{O}(np + n \log K)$) and $K - NN$ are easily fooled by irrelevant attributes (which make distances meaningless when only 10 out of $p = 1000$ features are relevant).

2.2 Decision Trees

2.3 Random Forest

3 Clustering & Unsupervised Learning