Machine Learning
Regression

Charlotte Pelletier
Univ. Bretagne Sud – IRISA Vannes
Based on C. Friguet's lecture.

07 October 2021

Content

Recall

- **Supervised learning**: in the observed data, we know the ”true” value of the predictor variable, and we look for understanding/predicting the (presumed) relationship between explanatory variables and the target variable
- **What is the type of the explanatory variable (Y) ?**
  - **quantitative**: **regression**
  - qualitative (2 or > 2 modes): classification (binary / multiclasses)
- **What is the type and the number of explanatory variables (X) ?**
  - type: **qualitative** and/or **quantitative**
  - **One** variable
    - Not frequent in practice, but it is useful to understand how works the method $\Rightarrow$ visualisation
  - **Several** variables
    - Several = from a dozen to thousands $\Rightarrow$ variable selection

Linear and logistic models (1/4)

Analysing the relationship between $\mathbf{Y}$ and all of the explanatory variables $[\mathbf{X}_1, \mathbf{X}_2, \cdots, \mathbf{X}_d]$:

- Linear regression: $\mathbf{Y}$ quantitative

$$y_i \approx f_{\boldsymbol{\beta}}(\mathbf{x}_i) = f_{\boldsymbol{\beta}}(x_{i1}, x_{i2}, \cdots, x_{id}) = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_d x_{id}$$

- Logistic regression: $\mathbf{Y}$ binary (0/1)

$$f_{\boldsymbol{\beta}}(\mathbf{x}_i) = \frac{e^{\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_d x_{id}}}{1 + e^{\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_d x_{id}}} = \mathbb{P}(Y = 1 | X = \mathbf{x}_i)$$

We want to find $\boldsymbol{\beta}$ such than $f_{\boldsymbol{\beta}}(\mathbf{x}_i)$ is as close as possible of y_i for all the training instances $\{\mathbf{x}_i, y_i\}_{i=1}^m$ where $\mathbf{x}_i \in \mathbb{R}^d$

- Matrix notation:

$$f_{\boldsymbol{\beta}}(\mathbf{X}) \approx \tilde{\mathbf{X}}\boldsymbol{\beta}$$

$$\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ \\ y_m \end{pmatrix} \approx \begin{pmatrix} 1 & x_{11} & x_{12} & \dots & x_{1d} \\ 1 & x_{21} & x_{22} & \dots & x_{2d} \\ \vdots & \vdots & \vdots & \vdots & \\ \\ 1 & x_{m1} & x_{m2} & \dots & x_{md} \end{pmatrix} \quad \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \\ \beta_d \end{pmatrix}$$

where $\mathbf{X} \in \mathbb{R}^{m \times d}$, $\tilde{\mathbf{X}} \in \mathbb{R}^{m \times (d+1)}$ et $\boldsymbol{\beta} \in \mathbb{R}^{d+1}$

Linear and logistic models (3/4)

Global cost

- Linear regression:

$$\sum_{i=1}^{m}\Big(f_{\boldsymbol{\beta}}(\mathbf{x}_i) - y_i\Big)^2$$

- Logistic regression:

$$\sum_{i=1}^{m}\Big[y_i log\Big(f_{\boldsymbol{\beta}}(\mathbf{x}_i)\Big) + (1 - y_i) log\Big(1 - f_{\boldsymbol{\beta}}(\mathbf{x}_i)\Big)\Big]$$

Linear and logistic models (4/4)

Objective: minimise the cost:

$$\boldsymbol{\beta}^* = \underset{\boldsymbol{\beta}}{\operatorname{argmin}}\Big(\mathbf{J}(\boldsymbol{\beta})\Big)$$

- Linear regression: explicit solution (Least Squares) - if $S = (\mathbf{X}^T\mathbf{X})$ can be inverted

$$\underset{\boldsymbol{\beta}}{\operatorname{argmin}}\Big(\mathbf{J}(\boldsymbol{\beta})\Big) \quad = \quad \underset{\boldsymbol{\beta}}{\operatorname{argmin}}\,||\mathbf{Y} - \tilde{\mathbf{X}\boldsymbol{\beta}}|| = (\tilde{\mathbf{X}}^T\mathbf{X})^{-1}\tilde{\mathbf{X}}^T\mathbf{Y}$$

- Logistic regression: no explicit solution. It requires the use of iterative optimisation algorithms such as the descent gradient algorithm (and its variants).

Iteration k of the gradient descent algorithm for the linear/logistic regression

$$\beta_j^{(k)} \quad := \quad \beta_j^{(k-1)} - \alpha\frac{1}{m}\sum_{i=1}^{m}\Big(f_{\boldsymbol{\beta}^{(k-1)}}(\mathbf{x}_i) - y_i\Big)x_{ij}$$

Note: $\forall i, x_{i0} = 1$

Content

Goodness of fit

How to check the quality of a regression model?

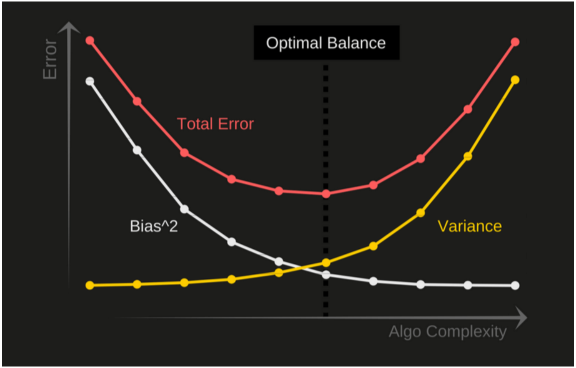
- visually
- as a function of the prediction quality (MAE, RMSE, R^2) $\Rightarrow$ the goodness of fit must be measure on a positive (or negative) scale
- as a function of the consequences of the actions (predictions can be seen as a special case of actions)

- **Generalisation**: important propriety of a learning process
  - The model generalisation presents the model capacity to be able to use robust predictions on new data.
- **Overfitting/underfitting** = model that does not generalize well from the training data

➜ A tradeoff between the bias (underfitting) and the variance (overfitting) is required

Bias-variance decomposition

Total error = Bias^2 + Variance + error



Source: `https://elitedatascience.com/bias-variance-tradeoff`

The example of the least square error

- Let $\mathbf{x}$ be independent variables and y the dependent answer variable.
- We assume f model the "true" relationship between $\mathbf{x}$ and y :

$$y = f(\mathbf{x}) + \epsilon$$

  where ϵ is a random variable, which models the inherent data ($\mathbb{E}[\epsilon] = 0$ and $var(\epsilon) = \mathbb{E}[\epsilon^2] = \sigma_\epsilon^2$).
- In practice, we do not know f but we look for an approximation $\hat{f}$, which is the model.
  For a test observation $\mathbf{x}$ (that does not belong to the training data), we search for $\hat{f}$ such as $y \approx \hat{f}(x)$.

It is possible to decompose the mean square error (MSE) – $MSE = \mathbb{E}_{\hat{f}}[(y - \hat{f}(x))^2]$ – for a set of test examples as follows:

$$\mathbb{E}_{\mathbf{x}}\mathbb{E}_{\hat{f}}[(y - \hat{f}(x))^2] = \mathbb{E}_{\mathbf{x}}[biais[\hat{f}(x)]^2] + \mathbb{E}_{\mathbf{x}}[var(\hat{f}(x))] + \epsilon^2$$

- How to correctly select a model?
    - Complex model (high variance) $\Rightarrow$ the underlying phenomenon is poorly represented, the model is too dependent on the training data and noise (random fluctuations that are not representative of the phenomenon)
    - Simple model (high bias) $\Rightarrow$ the complexity of the phenomenon is not captured, the model is not enough specialized to provide accurate predictions

    ➜ We look for a tradeoff!

- How to correctly select a model?
  - Complex model (high variance) $\Rightarrow$ the underlying phenomenon is poorly represented, the model is too dependent on the training data and noise (random fluctuations that are not representative of the phenomenon)
  - Simple model (high bias) $\Rightarrow$ the complexity of the phenomenon is not captured, the model is not enough specialized to provide accurate predictions
  - ➜ We look for a tradeoff!

- How to correctly select a model?
  - training data: to build the model
  - validation data: to tune the model hyperparameters
  - testing data: to evaluate the model performance on new data (not seen during the learning process)

Content

Regularisation

- Goal: add some information to avoid an overfitting situation by penalizing a model too complex

Regularisation

- Goal: add some information to avoid an overfitting situation by penalizing a model too complex
- Solution: all the explanatory variables are kept in the model, but a norm is added to the model parameters in the cost function
  - $\mathcal{L}_1$-norm: $||\boldsymbol{\beta}||_1 = \sum_j |\beta_j|$
  - $\mathcal{L}_2$-norm: $||\boldsymbol{\beta}||_2^2 = \sum_j \beta_j^2$

Regularisation

- Goal: add some information to avoid an overfitting situation by penalizing a model too complex
- Solution: all the explanatory variables are kept in the model, but a norm is added to the model parameters in the cost function
  - $\mathcal{L}_1$-norm: $||\boldsymbol{\beta}||_1 = \sum_j |\beta_j|$
  - $\mathcal{L}_2$-norm: $||\boldsymbol{\beta}||_2^2 = \sum_j \beta_j^2$
- Consequences:
  - we control values of some parameters, the model is thus more simple and can generalize more easily
  - the model should be more efficient as the (expected value of the) prediction error decreases

In practice

We slightly modify the optimisation problem by adding a penalty term: maximisation of the data likelihood while having satisfactory values for the penalty term

$$\boldsymbol{\beta}^* = \underset{\boldsymbol{\beta}}{\operatorname{argmin}}\Big(\mathbf{J}(\boldsymbol{\beta}) + \lambda\mathcal{R}(\boldsymbol{\beta})\Big)$$

- $\mathcal{R}(\boldsymbol{\beta})$: penalty term (positive function of $\boldsymbol{\beta}$)
- $\lambda > 0$: regularisation hyperparameter to be defined by the user. It controls the importance of the regularization term.

Ridge regression

Ridge regression (*shrinkage*) = we oblige the parameters to take small values
$\Rightarrow \mathcal{L}_2$-regularizer

- Linear regression:

$$\mathbf{J}(\boldsymbol{\beta}, \lambda) = \frac{1}{2m} \sum_{i=1}^{m} \left(f_{\boldsymbol{\beta}}(\mathbf{x}_i) - y_i \right)^2 + \frac{\lambda}{2m} \sum_{j=1}^{d} \beta_j^2$$

Explicit solution : $\boldsymbol{\beta}^* = \left[(\mathbf{X}^T\mathbf{X}) + \lambda \mathbb{I} \right]^{-1} \mathbf{X}^T\mathbf{Y}$

- Logistic regression:

$$\mathbf{J}(\boldsymbol{\beta}, \lambda) = -\frac{1}{m} \sum_{i=1}^{m} \left[y_i log\left(f_{\boldsymbol{\beta}}(\mathbf{x}_i) \right) + (1 - y_i) log\left(1 - f_{\boldsymbol{\beta}}(\mathbf{x}_i) \right) \right] + \frac{\lambda}{2m} \sum_{j=1}^{d} \beta_j^2$$

$\sim$ *weight-decay* (applied to the stochastic gradient descent algorithm)

Ridge regression

Iteration k of the gradient descent algorithm with regularisation

$$
\begin{aligned}
\beta_0^{(k)} &:= \beta_0^{(k-1)} - \frac{\alpha}{m}\sum_{i=1}^{m}\Big(f_{\boldsymbol{\beta}^{(k-1)}}(\mathbf{x}_i) - y_i\Big) \\
\beta_j^{(k)} &:= \beta_j^{(k-1)} - \alpha\left[\frac{1}{m}\sum_{i=1}^{m}\Big(f_{\boldsymbol{\beta}^{(k-1)}}(\mathbf{x}_i) - y_i\Big)x_{ij} + \frac{\lambda}{m}\beta_j^{(k-1)}\right] \\
&= \beta_j^{(k-1)}\left(1 - \frac{\alpha\lambda}{m}\right) - \frac{\alpha}{m}\sum_{i=1}^{m}\Big(f_{\boldsymbol{\beta}^{(k-1)}}(\mathbf{x}_i) - y_i\Big)x_{ij}
\end{aligned}
$$

LASSO (*Least Absolute Shrinkage and Selection Operation*) = we oblige the coefficients to take values close from zero$\Rightarrow$ $\mathcal{L}_1$-regularizer

- Linear regression:

$$\mathbf{J}(\boldsymbol{\beta}, \lambda) = \frac{1}{2m}\sum_{i=1}^{m}\Big(f_{\boldsymbol{\beta}}(\mathbf{x}_i) - y_i\Big)^2 + \frac{\lambda}{2m}\sum_{j=1}^{d}|\beta_j|$$

- Logistic regression:

$$\mathbf{J}(\boldsymbol{\beta}, \lambda) = -\frac{1}{m}\sum_{i=1}^{m}\Big[y_i log\Big(f_{\boldsymbol{\beta}}(\mathbf{x}_i)\Big) + (1-y_i)log\Big(1 - f_{\boldsymbol{\beta}}(\mathbf{x}_i)\Big)\Big] + \frac{\lambda}{2m}\sum_{j=1}^{d}|\beta_j|$$

- What about the parameter β_0?
  - we do not regularize β_0 (known as the intercept or the bias term)
- Explanatory variables $\mathbf{X}$ must be mean-centring in order to limit the influence of the variables with a high variance (while keeping $\forall i, x_{i0} = 1$)

Notes

Notes on LASSO (only)

- No algorithm to compute directly the parameters $\Rightarrow$ use of iterative approaches with an initialisation $\forall j,\ \beta_j = 0$
- LASSO effect
  - some parameters are set to 0 $\Rightarrow$ some explanatory variables are ”excluded” from the model
  - similar to a variable selection procedure (for example, selection of one variable among correlated variables)
- LASSO allows to have at most m non-zeros parameters

Regularisation hyperparameter

Notes

- What is the role of λ?
  - $\lambda \mapsto +\infty$: all the parameters $\beta \mapsto 0$
  - $\lambda = 0$: no regularisation

- How to select the value of λ?
  - for example by cross-validation (minimisation of the prediction error)

Elasticnet

Combining *ridge* and *LASSO*

- Linear regression:

$$\mathbf{J}(\boldsymbol{\beta}, \lambda_1, \lambda_2) = \frac{1}{2m}\sum_{i=1}^{m}\Big(f_{\boldsymbol{\beta}}(\mathbf{x}_i) - y_i\Big)^2 + \frac{\lambda_1}{2m}\sum_{j=1}^{d}|\beta_j| + \frac{\lambda_2}{2m}\sum_{j=1}^{d}\beta_j^2$$

- Logistic regression:

$$\begin{aligned}\mathbf{J}(\boldsymbol{\beta}, \lambda_1, \lambda_2) \quad = \quad & -\frac{1}{m}\sum_{i=1}^{m}\Big[y_i log\Big(f_{\boldsymbol{\beta}}(\mathbf{x}_i)\Big) + (1 - y_i) log\Big(1 - f_{\boldsymbol{\beta}}(\mathbf{x}_i)\Big)\Big] \\ & + \frac{\lambda_1}{2m}\sum_{j=1}^{d}|\beta_j| + \frac{\lambda_2}{2m}\sum_{j=1}^{d}\beta_j^2\end{aligned}$$

Other possible parametrisation

- Linear regression:

$$\mathbf{J}(\boldsymbol{\beta}, \lambda, \alpha) = \frac{1}{2m} \sum_{i=1}^{m} \Big(f_{\boldsymbol{\beta}}(\mathbf{x}_i) - y_i \Big)^2 + \lambda \Big[\frac{\gamma}{2m} \sum_{j=1}^{d} |\beta_j| + \frac{1-\gamma}{2m} \sum_{j=1}^{d} \beta_j^2 \Big]$$

- Logistic regression:

$$\begin{aligned} \mathbf{J}(\beta, \lambda, \alpha) \quad &= \quad -\frac{1}{m} \sum_{i=1}^{m} [y_i log\Big(f_{\boldsymbol{\beta}}(\mathbf{x}_i) \Big) + (1 - y_i) log\Big(1 - f_{\boldsymbol{\beta}}(\mathbf{x}_i)\Big)] \\ &\quad + \gamma \Big[\frac{\alpha}{2m} \sum_{j=1}^{d} |\beta_j| + \frac{1-\gamma}{2m} \sum_{j=1}^{d} \beta_j^2 \Big] \end{aligned}$$

Notes

- Variable selection (parameter = 0) – as LASSO
- Grouping correlated variables: sharing the weights – as Ridge
- Estimation of the parameters with optimisation techniques (*coordinate descent algorithm*)
- How to select λ_1 and λ_2? With a two-step procedure.

Content

Instructions (1/2)

The goal of this assignment is to implement the regularisation techniques presented in this lecture, and to test them.

1. Ridge regression
   - Implement ridge regression by changing the functions that you have implemented during the second lab session.
   - Study the sensitivity of ridge regression to the regularisation hyperparameter λ. Comment the results.
2. Comparison of regularization techniques
   - Use Python methods to compare the results of the three regularization techniques: ridge regression, LASSO, and ElasticNet
   - Tune the lambda hyperparameter value by using a cross-validation procedure.
   - Evaluate the performance of the linear regression algorithm on test data with and without regularization. Comment the results.