FTML PROJECT REPORT

Luu Hoang Long Vo - Phu Hien Le - Yassin Bouhassoun - Youssef Bouarfa Dinia

Contents

1 Exercise 1

1.1 Question 1

Condsider the following joint random variable (X,Y).

$$\begin{aligned} &X = \{0, 1, 2, 3, 4, 5\} \\ &y = \{-1, 1\} \\ &Y = \begin{cases} B(\frac{1}{4}) \ if \ X = 0 \\ B(\frac{2}{5}) \ if \ X = 1 \\ B(\frac{1}{3}) \ if \ X = 2 \\ B(\frac{3}{4}) \ if \ X = 3 \\ B(\frac{5}{6}) \ if \ X = 4 \\ B(\frac{7}{8}) \ if \ X = 5 \end{cases} \end{aligned} \tag{1}$$

With $B(p)$ a Bernouilli law with parameter p.

1.1.1 Bayes predictor, general case

We prove again the general result on the Bayes predictor in the case of binary classification. We have seen that the Bayes predictor is defined by:

$$f^*(x) = arg \ min \ E[l(y, z)|X = x] \tag{2}$$

Hence with $z \in y$:

$$\begin{aligned} f^*(x) &= arg \ min \ E[l(y, z)|X = x] \\ &= arg \ min \ P(Y \neq z|X = x) \\ &= arg \ min \ 1 - P(Y = z|X = x) \\ &= arg \ max \ P(Y = z|X = x) \end{aligned} \tag{3}$$

The optimal classifier selects the most probable output given X = x.

1.1.2 Bayes Risk, general case

We have also seen that using the law of total expectation, with the "0-1" loss,

$$\begin{aligned} R^* &= E[l(Y, f^*(x))] \\ &= E_X[E_Y(l|Y \neq f^*(X)|X)] \\ &= E_X[P(Y \neq f^*(X)|X)] \end{aligned} \tag{4}$$

But we have

$$P(Y \neq f^*(X)|X = x)) = P(Y \neq f^*(x)) \tag{5}$$

We note $\eta(x) = P(Y = 1|X = x)$

If $\eta(x) > \frac{1}{2}$:

$$f^*(x) = 1 \ \& \ P(Y \neq f^*(x)) = P(Y = 0) = 1 - \eta(x) \tag{6}$$

If $\eta(x) < \frac{1}{2}$:

$$f^*(x) = 0 \ \& \ P(Y \neq f^*(x)) = P(Y = 1) = \eta(x) \tag{7}$$

In both cases,

$$P(Y \neq f^*(x)) = min(\eta(x), 1 - \eta(x)). \tag{8}$$

We conclude that

$$R^* = E_X[min(\eta(X), 1 - \eta(X))] \tag{9}$$

1.1.3 Application

$$f^*(0) = -1 \tag{10}$$

$$f^*(1) = -1 \tag{11}$$

$$f^*(2) = -1 \tag{12}$$

$$f^*(3) = 1 \tag{13}$$

$$f^*(4) = 1 \tag{14}$$

$$f^*(5) = 1 \tag{15}$$

$$\begin{aligned} R^* &= E_X[min(n(X), 1 - n(X))] \\ &= \frac{1}{6} * \frac{1}{4} + \frac{1}{6} * \frac{2}{5} + \frac{1}{6} * \frac{1}{3} + \frac{1}{6} * \frac{1}{4} + \frac{1}{6} * \frac{1}{6} + \frac{1}{6} * \frac{1}{8} \\ &= \frac{1}{6}(\frac{1}{4} + \frac{2}{5} + \frac{1}{3} + \frac{1}{4} + \frac{1}{6} + \frac{1}{8}) \\ &= 0.2541 \end{aligned} \tag{16}$$

1.2 Question 2

Can be found in the exercice1.ipynb

2 Exercise 2

2.1 Question 1

If $\forall\, x \in X$,

$$\begin{cases} P(Y=0|X=x) = 2/3 \\ P(Y=1|X=x) = 1/3 \end{cases} \tag{17}$$

The Minimizer of the mean square error is:

$$\begin{aligned} f^*(x) &= E[Y|X=x] \\ &= 0 * \frac{2}{3} + 1 * \frac{1}{3} \\ &= \frac{1}{3} \end{aligned} \tag{18}$$

For the Mean absolute error:

$$\begin{aligned} E[|f^*(x) - Y|] &= E[|\frac{1}{3} - Y|] \\ &= |\frac{1}{3} - 0| * \frac{2}{3} + |\frac{1}{3} - 1| * \frac{1}{3} \\ &= \frac{1}{3} * \frac{1}{3} + \frac{2}{3} * \frac{1}{3} \\ &= \frac{4}{9} \end{aligned} \tag{19}$$

Whereas $\hat{f}(\mathrm{x}) := 0$ give:

$$\begin{aligned} E[|\hat{f}(x) - Y|] &= E[|0 - Y|] \\ &= |0 - 0| * \frac{2}{3} + |0 - 1| * \frac{1}{3} \\ &= \frac{1}{3} \end{aligned} \tag{20}$$

So we can conclude that $\hat{f}$ is better than f^*.

2.2 Question 2

Minimizing of $g(z) = \int |y - z| p(y) dy$:

$$\begin{aligned}
\frac{d}{dz} g(z) &= \frac{d}{dz} g(z) \\
&= \frac{d}{dz} \left(\int_{-\infty}^{z} (z - y) p(y) dy + \int_{z}^{\infty} (y - z) p(y) dy \right) \\
&= \left((z - y) p(y) \right)_{|y=z} + \int_{-\infty}^{z} p(y) dy - \left((y - z) p(y) \right)_{|y=z} + \int_{z}^{\infty} -p(y) dy \\
&= - \int_{z}^{\infty} p(y) dy + \int_{-\infty}^{z} p(y) dy \\
&= \int_{-\infty}^{z} p(y) dy - 1 + \int_{-\infty}^{z} p(y) dy \\
&= 2 \int_{-\infty}^{z} p(y) dy - 1
\end{aligned} \tag{21}$$

Let $P(z) := \int_{-\infty}^{z} p(y) dy$ be the cumulative distribution function, then:

$$\frac{d}{dz} g(z) = 2P(z) - 1 \tag{22}$$

The median is the solution of:

$$P(m) = \frac{1}{2} \quad (m := p^{-1}(\frac{1}{2})) \tag{23}$$

So if $z \leq m$ then:

$$\begin{aligned}
\frac{d}{dz} g(z) &= 2P(z) - 1 \\
&\leq 2P(m) - 1 \\
&= 2 * \frac{1}{2} - 1 = 0
\end{aligned} \tag{24}$$

If $z \geq m$ then :

$$\begin{aligned}
\frac{d}{dz} g(z) &= 2P(z) - 1 \\
&\geq 2P(m) - 1 = 0
\end{aligned} \tag{25}$$

Hence:

z	$-\infty$	m	$+\infty$
$g(z)$	a ↘	b	↗ a

(26)

The minimizer is m (the median of Y under X=x).

3 Exercise 3

3.1 Step 1

We have:

$$Y = x\theta^* + \epsilon \tag{27}$$

$$\hat{\theta} = (XX^T)^{-1}XY \tag{28}$$

From that, we can deduce:

$$\begin{aligned}
R_X(\hat{\theta}) &= E[R_n(\hat{\theta})] \\
&= E[\frac{1}{n}||Y - X\hat{\theta}||^2] \\
&= E[\frac{1}{n}||Y - X(X^TX)^{-1}X^TY)||^2] \\
&= E[\frac{1}{n}||X\theta^* + \epsilon - X(X^TX)^{-1}X^T(X\theta^* - \epsilon)||^2] \\
&= E[\frac{1}{n}||X\theta^* + \epsilon - X\theta^* + X(X^TX)^{-1}X^T\epsilon] \\
&= E[\frac{1}{n}||\epsilon + X(X^TX)^{-1}X^T\epsilon||^2] \\
&= E[\frac{1}{n}||(I_n + X(X^TX)^{-1}X^T)\epsilon||^2]
\end{aligned} \tag{29}$$

3.2 Step 2

$$(A^T)_{i,j} = A_{j,i} \tag{30}$$

$$\begin{aligned}
(A^TA)_{i,j} &= \sum_k (A^T)_{i,k}A_{k,j} \\
&= \sum_k A_{k,i}A_{k,j}
\end{aligned} \tag{31}$$

$$\begin{aligned}
tr(A^TA) &= \sum_i (A^TA)_{i,i} \\
&= \sum_i \sum_k A_{k,i}A_{k,i} \\
&= \sum_i \sum_k (A_{k,i})^2 \\
&= \sum_i \sum_j (A_{i,j})^2
\end{aligned} \tag{32}$$

3.3 Step 3

$$E_\epsilon[\frac{1}{n}||A\epsilon||^2] \tag{33}$$

$$\begin{aligned} ||A\epsilon||^2 &= (A\epsilon)^T A\epsilon = \epsilon^T A^T A\epsilon \\ &= tr(\epsilon^T A^T A\epsilon) = tr(A^T A\epsilon\epsilon^T) \end{aligned} \tag{34}$$

$$\begin{aligned} E_\epsilon[\frac{1}{n}||A\epsilon||^2] &= E_\epsilon[\frac{1}{n}tr(A^T A\epsilon\epsilon^T)] \\ &= \frac{1}{n}tr(A^T AE_\epsilon[\epsilon\epsilon^T]) \\ &= \frac{1}{n}tr(A^T A\sigma^2 I_n) \\ &= \frac{\sigma^2}{n}tr(A^T A) \end{aligned} \tag{35}$$

3.4 Step 4

$$\begin{aligned} A^T A &= (I_n - X(X^T X)^{-1}X^T)^T(I_n - X(X^T X)^{-1}X^T) \\ &= (I_n - X(X^T X)^{-1}X^T)(I_n - X(X^T X)^{-1}X^T) \\ &= I_n - 2X(X^T X)^{-1}X^T + X(X^T X)^{-1}X^T X(X^T X)^{-1}X^T \\ &= I_n - 2X(X^T X)^{-1}X^T + X(X^T X)^{-1}X^T \\ &= I_n - X(X^T X)^{-1}X^T \\ &= A \end{aligned} \tag{36}$$

3.5 Step 5

$$\begin{aligned}
E[R_n(\hat{\theta})] &= E[\frac{1}{n}||A\epsilon||^2] \\
&= \frac{\sigma^2}{n} tr(A^T A) \\
&= \frac{\sigma^2}{n} tr(A) \\
&= \frac{\sigma^2}{n} tr(I_n - X(X^T X)^{-1} X^T) \\
&= \frac{\sigma^2}{n} (tr(I_n) - tr(X(X^T X)^{-1} X^T) \\
&= \frac{\sigma^2}{n} (tr(I_n) - tr((X^T X)^{-1} X^T X)) \\
&= \frac{\sigma^2}{n} (tr(I_n) - tr(Id) \\
&= \frac{\sigma^2}{n} (n - d) \\
&= \frac{\sigma^2 (n-d)}{n}
\end{aligned} \tag{37}$$

3.6 Step 6

By definition:

$$R_n(\hat{\theta}) = \frac{1}{n} ||y - X\hat{\theta}||_2^2 \tag{38}$$

So,

$$E[\frac{1}{n} ||y - X\hat{\theta}||_2^2] = \frac{\sigma^2 (n-d)}{n} \tag{39}$$

And,

$$E[\frac{||y - X\hat{\theta}||_2^2}{n-d}] = \sigma^2 \tag{40}$$

3.7 Step 7

Can be found in the exercice3.ipynb

4 Exercise 4

exercice4.ipynb

4.1 Choice of regression methods

We performed regression on the data set using 3 different regression methods:

- Logistic Regression. Easy to implement, interpret and very efficient to train. Logistic regression is less inclined to over-fitting but it can over-fit in high dimensional data sets.
- Ridge Regression (alpha=0.1). Ridge Regression is very useful when there is multicollinearity in data (dependencies between the variables in the model). Ridge Regression also prevents over fitting
- Lasso Regression (alpha=0.1, fit_intercept=False, tol=0.001, max_iter=1000, positive=True). An advantage of Lasso is that it selects features. by shrinking co-efficient towards zero, and it avoids over fitting.

4.2 Performance

Algorithm	R2
Logistic Regression	0.9125
Ridge Regression	0.9129
Lasso Regression	0.911

4.3 Optimization method (Hyperparameter Tuning)

4.3.1 GridSearchCV

Here we can have a grid of parameter that will be passed as an argument to the estimator along side the estimator of choice. The function will then create numerous models and cross validate it to determine which set of parameters perform best for the given test and will give back the best estimator along side its parameters.

4.3.2 Builtin Cross Validate Estimator

With the same principle as the previous method, how ever the tuning here is done without the user needing to set a grid of parameters. The method will automatically find the best performing set parameters and give back to the user.

5 Exercise 5

exercice5.ipynb

5.1 Choice of classifiers

We performed classification on the dataset using 4 different classifiers

- Logistic Regression (random_state=0). Logistic Regression is easy to implement, interpret and very efficient to train. It has good accuracy for many simple data sets and it performs well when the data set is linearly separable.
- LinearSVC (random_state=0, tol=1e-5). LinearSVC is similar to SVC, however LinearSVC tends to be faster to converge the larger the number of samples is. Linear is based on the library 'liblinear' which offers more penalties and loss functions in order to scale better with large number of samples.
- SVC (gamma='auto'). Support Vector Machine (SVM) is more effective in high dimensional spaces and is relatively memory efficient. SVM is also effective in cases where the number of dimensions is greater than the number samples. However SVM is not suitable for large data sets and when the data sets has more noise.
- NuSVC (nu=0.3). NuSVC and SVC are mathemathically equivalent with both methods based on the library 'libsvm'. The main difference is NuSVC uses parameter 'v' (instead of 'C' in SVC) which controls the number of support vectors and the margin errors. Parameter 'v' is an upper bound on the fraction of margin errors and a lower of the fraction of support vectors. A margin error corresponds to a sample that lies on the wrong side of its margin boundary: it is either misclassified, or it is correctly classified but does not lie beyond the margin.

5.2 Performance

Algorithm	Accuracy
Logistic Regression	0.91
LinearSVC	0.905
SVC	0.905
NuSVC	0.89

5.3 Optimization method

With the same method as GridSearch we use a package that is called skopt in order to create a search space with numerous parameters set in a range. The method will create multiple estimators and return the set of parameters that performs best given the input and output data.

```
from sklearn.model_selection import RepeatedStratifiedKFold
from skopt.utils import use_named_args

# define the function used to evaluate a given configuration
@use_named_args(search_space)
def evaluate_model(**params):
        # configure the model with specific hyperparameters
        model = SVC()
        model.set_params(**params)
        # define test harness
        cv = RepeatedStratifiedKFold(n_splits=10,
        n_repeats=3, random_state=1)
        # calculate 5-fold cross validation
        result = cross_val_score(model, inputs, np.ravel(labels),
        cv=cv, n_jobs=-1, scoring='accuracy')
        # calculate the mean of the scores
        estimate = np.mean(result)
        # convert from a maximizing score to a minimizing score
        return 1.0 - estimate
```

```
from skopt import gp_minimize
# perform optimization
result = gp_minimize(evaluate_model, search_space)
# summarizing finding:
print('Best Accuracy: %.3f' % (1.0 - result.fun))
print('Best Parameters: %s' % (result.x))
```