Statistical Methods: Portfolio 3

By Henry Bourne

1 Linear Classifiers

In the portfolio 1 we saw how to conduct binary classification, now we will look at how to conduct **multi-class classification**. This is where we have an input $\boldsymbol{x} \in \mathbb{R}^d$ and an output $y \in \{1, ..., K\}$.

The geometry of the problem in this case is more complicated than we had with binary classification, we can no longer simply check the sign of a single $f(\boldsymbol{x})$ to classify. We could try introducing multiple functions. Let's say we have 3 classes, we could try introducing another function so now we have f_1 and f_2, we could then perform classification by checking the signs of both f_1 and f_2 and have this dictate the class. However, this can get confusing as for example in this case we have 4 possible outcomes, $\{+, -\}^2$, for the signs of our f's, however, we only have 3 classes. What about if we introduce pairwise binary classifiers, ie. we have $f_{i,j}$ classifies a point as either i or j and we have such a function for any pair of classes, then we classify a point as the majority vote given by the binary classifiers. The problem here is that all the classifiers may disagree in which case there is no majority vote.

Rather than relying on the sign of a function f to make predictions lets instead fit a vector valued function $\boldsymbol{f} : \mathbb{R}^d \to \mathbb{R}^K$, where K is the number of classes. Given an input $\boldsymbol{x}$ our prediction is $\hat{k} = \text{argmax}_k \, f^{(k)}(\boldsymbol{x})$. Note that this no longer has a simple geometric interpretation anymore.

1.1 Least Squares Classifier

We will first define the least squares classifier in the binary classification case and after extend it to the multi-class case:

Definition 1.1 ***Least Squares Binary Classifier:***
We first perform LS on the data, ie. find:

$$\boldsymbol{w}_{LS} := \underset{\boldsymbol{w}}{\text{argmin}} \sum_{i \in D} [y_i - f(\boldsymbol{x}_i; \boldsymbol{w})]^2 \tag{1}$$

Then we can find the predicted label $\hat{y} := sign(f(\boldsymbol{x}_i; \boldsymbol{w}_{LS}))$

Note that we can also use a feature transform for f as well, which would allow us to fit more complex classifiers, as data not separable in the original space may be separable in the feature space. In the multi-class case:

Definition 1.2 ***Multi-class LS classification:***
We use a ***one-hot encoding*** *which is where we replace* $y_i = k$ *in our data with* $\boldsymbol{t}_i \in \{0, 1\}^K$ *where all entries are 0 bar* $t_i^{(k)} = 1$. *Then we have:*

$$\boldsymbol{W}_{LS} := \underset{\boldsymbol{W}}{\text{argmin}} \sum_{i \in D} ||\boldsymbol{t}_i - f(\boldsymbol{x}_i; \boldsymbol{W})||^2 \tag{2}$$

where $\boldsymbol{W} \in \mathbb{R}^{(d+1) \times K}$, $f(\boldsymbol{x}; \boldsymbol{W}) = \boldsymbol{W}^T \tilde{\boldsymbol{x}}$ *and* $\tilde{\boldsymbol{x}} := [\boldsymbol{x}^T, 1]^T$.
Then our prediction $\hat{y} = \text{argmax}_k \, f(\boldsymbol{x}; \boldsymbol{W})^{(k)} = \text{argmax}_k (\boldsymbol{w}_{LS}^{(k)})^T \tilde{\boldsymbol{x}}$, *where* $\boldsymbol{w}^{(k)}$ *is the k-th column of* $\boldsymbol{W}$.

Although this method can work the square loss tends not to make sense in a classification task, as a point far away from the boundary (fit without that point) can dramatically affect our decision boundary, even if it is correctly classified by the decision boundary (fit without that point). Also, unlike with LS regression, LS classification lacks a probabilistic interpretation.

1.2 Fisher Discriminant Analysis (FDA)

Note that taking the inner product $\langle \boldsymbol{w}, \boldsymbol{x} \rangle$ embeds $\boldsymbol{x}$ onto a one-dimensional line along the $\boldsymbol{w}$ direction. We can say $\boldsymbol{w}$ gives a good embedding if the $\boldsymbol{x}$ it embeds are close together in the embedding if they are from the same class but far apart if they are from different classes. We can define these two properties we want from our embedding as:

Definition 1.3 ***Within-class Scatterness:***
For embedding $\boldsymbol{w}^T\boldsymbol{x}$, the ***embedding centre*** *for class k is:*

$$\hat{\mu}_k = \frac{1}{n_k} \sum_{i, y_i = k} \boldsymbol{w}^T \boldsymbol{x}_i \tag{3}$$

then the within-class scatterness of class k is:

$$s_{\boldsymbol{w},k} := \sum_{i, y_i = k} (\boldsymbol{w}^T \boldsymbol{x}_i - \hat{\mu}_k)^2 \tag{4}$$

Definition 1.4 ***Between-class Scatterness:***
The ***embedded dataset centre*** *is:*

$$\hat{\mu} = \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{w}^T \boldsymbol{x}_i \tag{5}$$

then teh between-class scatterness is:

$$s_{b,k} = n_k (\hat{\mu}_k - \hat{\mu})^2 \tag{6}$$

note the n_k is needed to make $s_{b,k}$ the same scale as $s_{w,k}$.

Then ideally we would like to maximize the between-class scatterness and minimize the within-class scatterness for all the classes, which is what the following does:

Definition 1.5 ***Fisher Discriminant Analysis (FDA):***

$$\max_{\boldsymbol{w}} [\sum_k s_{b,k} / \sum_k s_{\boldsymbol{w},k}] \tag{7}$$

if $K = 2$ then this has a simple solution: $\boldsymbol{w} := \boldsymbol{S}_{\boldsymbol{w}}^{-1}(\boldsymbol{\mu}_+ - \boldsymbol{\mu}_-)$, where $\boldsymbol{S}_{\boldsymbol{w}} := \sum_{k=1}^{K} \boldsymbol{S}_k$ and $\boldsymbol{S}_k$ is the sample covariance matrix of class k times n_k. However, note that the FDA does not learn a decision function f, the $\boldsymbol{w}_{FDA}$ obtained cannot be used directly by the prediction function to make a prediction. This is because (eg. in the binary case) $f(\boldsymbol{x}; \boldsymbol{w}_{FDA}) > 0$ does not mean that $\boldsymbol{x}$ is predicted as the positive class. The FDA also doesn't care about classification accuracy, ie. minimizing the FP or FN rate.

1.3 Probabilistic (Generative) Classifiers

Appendices

A Proofs

A.1

B Homeworks

B.1 For Section 1

Question B.1.1