Introduction to Machine learning

Sylvain Le Corff

Summary

Supervised Learning

Supervised Learning Framework

- Input measurement $\mathbf{X} \in \mathcal{X}$ (often $\mathcal{X} \subset \mathbb{R}^d$).
- Output measurement $Y \in \mathcal{Y}$.
- The joint distribution of $(\mathbf{X}, Y)$ is unknown.
- $Y \in \{1, \ldots, M\}$ (classification) or $Y \in \mathbb{R}^m$ (regression).
- A predictor is a measurable function $f : \mathcal{X} \to \mathcal{Y}$.

Training data

- i.i.d. with the same distribution as $(\mathbf{X}, Y)$:

$$\mathcal{D}_n = \{(\mathbf{X}_1, Y_1), \ldots, (\mathbf{X}_n, Y_n)\} .$$

Goal

- Construct a good predictor $\widehat{f}_n$ from the training data.
- Need to specify the meaning of good.

Loss function

→ $\ell(Y, f(\mathbf{X}))$: the goodness of the prediction of Y by $f(\mathbf{X})$.
→ Prediction loss: $\ell(Y, f(\mathbf{X})) = \mathbf{1}_{Y \neq f(\mathbf{X})}$.
→ Quadratic loss: $\ell(Y, \mathbf{X}) = \|Y - f(\mathbf{X})\|_2^2$.

Risk function

→ Risk measured as the average loss:

$$\mathcal{R}(f) = \mathbb{E}[\ell(Y, f(\mathbf{X}))] .$$

→ Prediction loss: $\mathbb{E}[\ell(Y, f(\mathbf{X}))] = \mathbb{P}(Y \neq f(\mathbf{X}))$.
→ Quadratic loss: $\mathbb{E}[\ell(Y, f(\mathbf{X}))] = \mathbb{E}[\|Y - f(\mathbf{X})\|_2^2]$.

→ **Beware:** As $\widehat{f}_n$ depends on $\mathcal{D}_n$, $\mathcal{R}(\widehat{f}_n)$ is a random variable!

Summary

Recall: multivariate Gaussian distribution

Definition

Let $\mu \in \mathbb{R}^d$, Σ be a positive definite matrix. We write $X \sim \mathcal{N}(\mu, \Sigma)$ when the Lebesgue density of X is

$$\begin{aligned} x \in \mathbb{R}^d \mapsto & |2\pi\Sigma|^{-\frac{1}{2}} e^{-\frac{1}{2}(x-\mu)^\top \Sigma^{-1}(x-\mu)} \\ &= \frac{1}{(2\pi)^{\frac{d}{2}} |\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}(x-\mu)^\top \Sigma^{-1}(x-\mu)}, \end{aligned}$$

where $|\Sigma|$ is the determinant of Σ. In addition, we have

$$\mathbb{E}[X] = \mu, \quad \mathbb{V}[X] = \Sigma,$$

where $\mathbb{V}[X]$ is the covariance matrix of X.

Maximum likelihood estimation

Proposition

Let $\mu^\star \in \mathbb{R}^d$, $\Sigma^\star$ *be a positive definite matrix and* $\{X_1, \ldots, X_n\}$ *be a sample i.i.d. according to* $\mathcal{N}(\mu^\star, \Sigma^\star)$.
Then

$$\hat{\mu}_n = \frac{1}{n} \sum_{i=1}^{n} X_i$$

and

$$\hat{\Sigma}_n = \frac{1}{n} \sum_{i=1}^{n} (X_i - \hat{\mu})(X_i - \hat{\mu})^\top$$

are maximum likelihood estimators respectively of $\mu^\star$ *and* $\Sigma^\star$.

Proof on blackboard

Binary classification

Bayes classifier

The Bayes classifier $g^\star$ is defined as:

$$g^\star(X) = \begin{cases} 1 & \text{if} \quad \mathbb{P}\left(Y = 1|X\right) > \mathbb{P}\left(Y = 0|X\right), \\ 0 & \text{otherwise.} \end{cases}$$

Equivalently,

$$g^\star(X) = \begin{cases} 1 & \text{if} \quad \mathbb{P}\left(Y = 1|X\right) > 1/2, \\ 0 & \text{otherwise,} \end{cases}$$

Lemma

For any classification rule $g : \mathbb{R}^d \to \{0, 1\}$, *one has*

$$\mathcal{R}(g^\star) \leqslant \mathcal{R}(g).$$

Multiclass problem

In the case where $Y \in \{1, \ldots, M\}$ for $M > 1$.

Bayes classifier

$$g^\star(X) \in \mathrm{argmax}_{i \in \{1,\ldots,M\}} \mathbb{P}(Y = i | X).$$

How to model the conditional law of Y?

In practice we do not know the conditional law of Y given X.
Several solutions to overcome this issue.

Fully parametric modeling.
Estimate the law of (X, Y) and use the **Bayes formula** to deduce an estimate of the conditional law of Y: *LDA/QDA, Naive Bayes...*

Parametric conditional modeling.
Estimate the conditional law of Y by a **parametric** law: *linear regression, logistic regression, Feed Forward Neural Networks...*

Nonparametric conditional modeling.
Estimate the conditional law of Y by a **non parametric** estimate: *kernel methods, nearest neighbors...*

Discriminant analysis

- $(X, Y) \in \mathbb{R}^d \times \{1, \ldots M\}$ be a pair of r.v.
- Y is a label characterizing the class of X.
- **Goal:** computing the Bayes classifier when for all $i \in \{1, \ldots, M\}$ the conditional distribution of X given $\{Y = i\}$ is Gaussian with positive definite matrix Σ_i and mean $\mu_i \in \mathbb{R}^d$.

Recall: a Bayes classifier

For multiclasses

$$g^\star(X) \in \mathrm{argmax}_{i \in \{1, \ldots, M\}} \mathbb{P}(Y = i | X).$$

Bayes classifier for normal distributions

Assume that for all $1 \in \{1, \ldots, M\}$, $\mathbb{P}(Y = i) = \pi_i$, where $\pi_i \in (0, 1)$.

Proposition

A Bayes classifier $g^\star$ is defined, for all $x \in \mathbb{R}^d$, by

$$g^\star(x) \in \text{argmax}_{i \in \{1,\ldots,M\}} \log(\pi_i) - \frac{1}{2} \log |\Sigma_i| - \frac{1}{2}(x - \mu_i)^\top \Sigma_i^{-1}(x - \mu_i).$$

Proof on blackboard

The case of two classes

- Only two classes ($M = 2$)
- In this case, a Bayes classifier satisfies

$$g^\star \colon X \mapsto \begin{cases} 1 & \text{if } \mathbb{P}(Y = 1|X) > \mathbb{P}(Y = 2|X) \\ 2 & \text{otherwise.} \end{cases}$$

Linear discriminant analysis (LDA)

We assume that the covariance is the same in each class.

$$\Sigma_1 = \Sigma_2 = \Sigma.$$

Proposition

Define

$$h \colon x \in \mathbb{R}^d \mapsto (\mu_1 - \mu_2)^\top \Sigma^{-1} x$$
$$b = \frac{1}{2}(\mu_2^\top \Sigma^{-1} \mu_2 - \mu_1^\top \Sigma^{-1} \mu_1) + \log\left(\frac{\pi_1}{\pi_2}\right).$$

Then, a Bayes classifier is

$$g^\star \colon x \in \mathbb{R}^d \mapsto \begin{cases} 1 & \text{if } h(x) + b > 0 \\ 2 & \text{otherwise.} \end{cases}$$

Proof on blackboard, see also `https://sylvainlc.github.io/`

LDA

- Note that the function $h(x) + b$ is linear in x.
- This is a linear classifier!

What happens when $\pi_1 = \pi_2$

- if $\pi_1 = \pi_2$, we have:

$$g^\star(x) = 1 \iff (x - \mu_1)^\top \Sigma^{-1} (x - \mu_1) < (x - \mu_2)^\top \Sigma^{-1} (x - \mu_2),$$

- $\pi_1 = \pi_2$ if and only if x is closer to μ_1 than μ_2 with respect to the Mahalanobis distance ruled by Σ.

Quadratic discriminant analysis (QDA)

- Each class is normally distributed
- But with different covariances

Proposition

Define

$$h\colon x \in \mathbb{R}^d \mapsto \frac{1}{2} x^\top (\Sigma_2^{-1} - \Sigma_1^{-1}) x + (\mu_1^\top \Sigma_1^{-1} - \mu_2^\top \Sigma_2^{-1}) x$$

$$b = \frac{1}{2}(\mu_2^\top \Sigma_2^{-1} \mu_2 - \mu_1^\top \Sigma_1^{-1} \mu_1) - \frac{1}{2} \log\left(\frac{|\Sigma_1|}{|\Sigma_2|}\right) + \log\left(\frac{\pi_1}{\pi_2}\right).$$

Then, a Bayes classifier is

$$g^\star\colon x \in \mathbb{R}^d \mapsto \begin{cases} 1 & \text{if } h(x) + b > 0 \\ 2 & \text{otherwise.} \end{cases}$$

Proof on blackboard, see also `https://sylvainlc.github.io/`

Comparison of LDA and QDA

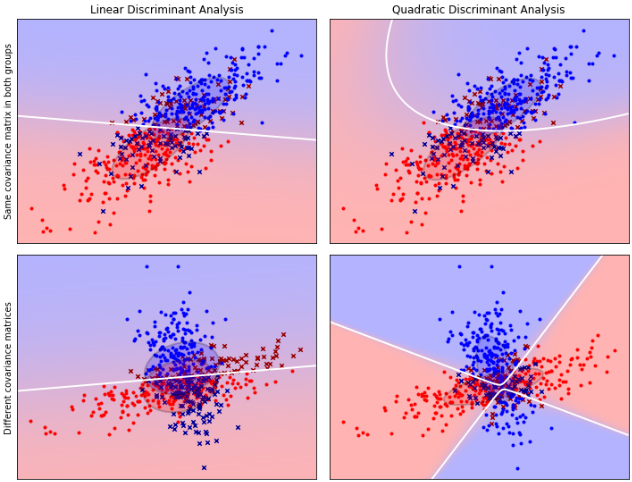

Figure: (Top) Data are generated with the same covrariance matrix in each group. (Bottom) Data are generated with different covrariance matrices in the two groups.

Comparison of LDA and QDA

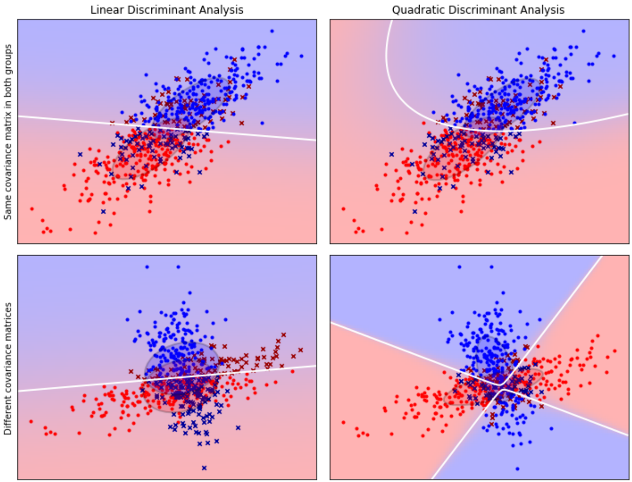

Figure: (Left) Classification boundary obtained with LDA, assuming the covariance matrix is the same in each group. (Right) Classification obtained with QDA, assuming the covariance matrices are different.

Comparison of LDA and QDA

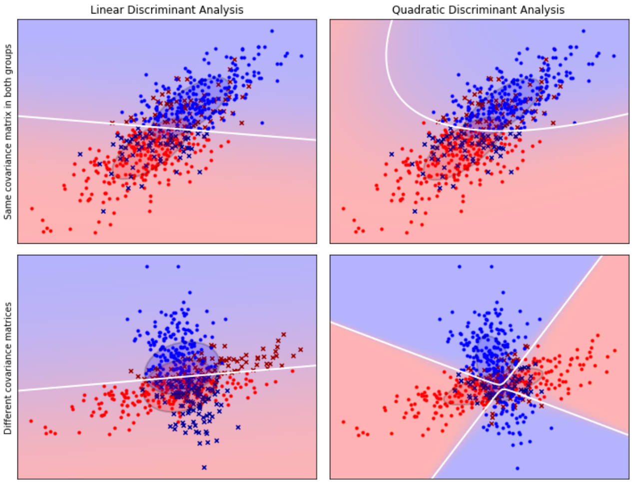

Figure: Crosses are all false positives i.e. all data wrongly classified by the discriminant analysis. Simulations are inspired by [Discriminant analysis with scikit-learn] and can be found here [?].

A very simple case - Naive Bayes

Classical algorithm using a crude modeling for the conditional law of X given Y

$\rightharpoondown$ Feature independence assumption: all components of X are independent given Y.

$\rightharpoondown$ Simple featurewise model: binomial if binary, multinomial if finite and Gaussian if continuous.

If all features are continuous, the law of X given Y is Gaussian with a diagonal covariance matrix!

Very simple learning even in very high dimension!

A very simple case - Gaussian Naive Bayes

$\rightharpoondown$ Feature independence assumption.

For $k \in \{1, 2\}$, $\mathbb{P}(Y = k) = \pi_k$ and the conditional density of $X^{(j)}$ given $\{Y = k\}$ is

$$g_k(x^{(j)}) = (2\pi\sigma_{j,k}^2)^{-1/2} \exp\left\{-(x^{(j)} - \mu_{j,k})^2/(2\sigma_{j,k}^2)\right\} .$$

The conditional distribution of X given $\{Y = k\}$ is then

$$g_k(x) = (\det(2\pi\Sigma_k))^{-1/2} \exp\left\{-(x - \mu_k)^T \Sigma_k^{-1} (x - \mu_k)/2\right\} ,$$

where $\Sigma_k = \mathrm{diag}(\sigma_{1,k}^2, \ldots, \sigma_{d,k}^2)$ and $\mu_k = (\mu_{1,k}, \ldots, \mu_{d,k})^\top$.

A very simple case - Gaussian Naive Bayes

In a two-classes problem, the optimal classifier is (see linear discriminant analysis):

$$f^*(X) = 2\mathbb{1}_{\mathbb{P}(Y=1|X)>\mathbb{P}(Y=-1|X)} - 1\,.$$

$\rightharpoondown$ When the parameters are unknown, they may be replaced by their maximum likelihood estimates. This yields, for $k \in \{1,2\}$,

$$\widehat{\pi}_{\mathbf{k}}^{\mathbf{n}} = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}_{Y_i=k}\,,$$

$$\widehat{\mu}_{\mathbf{k}}^{\mathbf{n}} = \frac{1}{\sum_{i=1}^{n}\mathbb{1}_{Y_i=k}}\sum_{i=1}^{n} \mathbb{1}_{Y_i=k}\,X_i\,,$$

$$\widehat{\Sigma}_k^n = \mathrm{diag}\left(\frac{1}{n}\sum_{i=1}^{n}(X_i - \widehat{\mu}_k^n)(X_i - \widehat{\mu}_k^n)^T\,\mathbb{1}_{Y_i=k}\right)\,.$$

Linear discriminant analysis in practice

The loglikelihood of the observations is given by

$$\ell_n(\theta) = -\frac{nd}{2}\log(2\pi) - \frac{n}{2}\log\det(\Sigma) + n_1 \log\pi_1 + (n - n_2)\log(1-\pi_1)$$
$$- \frac{1}{2}\sum_{i=1}^{n} \mathbb{1}_{Y_i=1} (X_i - \mu_1)^\top \Sigma^{-1} (X_i - \mu_1)$$
$$- \frac{1}{2}\sum_{i=1}^{n} \mathbb{1}_{Y_i=-1} (X_i - \mu_{-1})^\top \Sigma^{-1} (X_i - \mu_{-1})\,,$$

where $n_1 = \sum_{i=1}^n \mathbb{1}_{Y_i=1}$. This yields, for $k \in \{1,2\}$, the following MLE estimates:

$$\widehat{\pi}_{\mathbf{k}} = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}_{Y_i=k}\,, \quad \widehat{\mu}_{\mathbf{k}} = \frac{1}{\sum_{i=1}^n \mathbb{1}_{Y_i=k}} \sum_{i=1}^{n} \mathbb{1}_{Y_i=k} X_i\,,$$
$$\widehat{\Sigma} = \frac{1}{n}\sum_{i=1}^{n} (X_i - \widehat{\mu}_{Y_i})(X_i - \widehat{\mu}_{Y_i})^\top\,.$$

Linear discriminant analysis in practice

We assume that the covariance is the same in each class.

$$\Sigma_1 = \Sigma_2 = \Sigma.$$

Proposition

Define

$$\widehat{h} \colon x \in \mathbb{R}^d \mapsto (\hat{\mu}_1 - \hat{\mu}_2)^\top \hat{\Sigma}^{-1} x$$

$$\widehat{b} = \frac{1}{2}(\hat{\mu_2}^\top \hat{\Sigma}^{-1} \hat{\mu}_2 - \hat{\mu}_1^\top \hat{\Sigma}^{-1} \hat{\mu}_1) + \log\left(\frac{\hat{\pi}_1}{\hat{\pi}_2}\right).$$

Then, a "Plug-in" Bayes classifier is

$$\widehat{g} \colon x \in \mathbb{R}^d \mapsto \begin{cases} 1 & \text{if } \widehat{h}(x) + \widehat{b} > 0 \\ 2 & \text{otherwise.} \end{cases}$$

Proof on blackboard, see also `https://sylvainlc.github.io/`