Mathematical techniques in data science

Vu Dinh

Lecture 6: Classification – Linear Discriminant Analysis

February 27th, 2019

Schedule

Week	Chapter
1	Chapter 2: Intro to statistical learning
3	Chapter 4: Classification
4	Chapter 9: Support vector machine and kernels
5, 6	Chapter 3: Linear regression
7	Chapter 8: Tree-based methods + Random forest
8	
9	Neural network
10	Bootstrap and CV + Bayesian methods + UQ
11	Clustering: K-means → Spectral Clustering
12	PCA → Manifold learning
13	Reinforcement learning/Online learning/Active learning
14	Project presentation

Classification

- Logistic regression
- Linear Discriminant Analysis
- Support Vector Machines
- Nearest neighbours

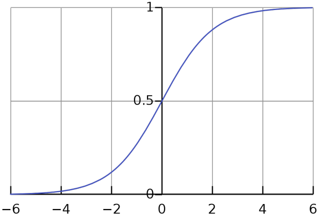
Classification: Logistic regression

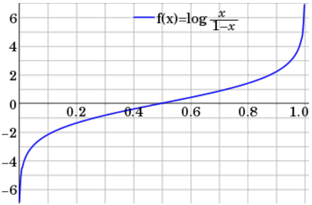
Logistic regression

- Suppose we work with binary outputs $\mathcal{Y} = \{0, 1\}$, and $\mathcal{X}$ is a subset of $\mathbb{R}^d$
- Goal: Given input X, we want to model the probability that $Y = 1$

$$P[Y = 1|X = x]$$

Transformation between $(-\infty, \infty)$ and $[0, 1]$





$$f(x) = \frac{e^x}{1 + e^x}$$

$$logit(p) = \log \frac{p}{1 - p}$$

Logistic regression: Assumptions

Assumption

Given $X = x$, Y is a Bernoulli random variable with parameter $p(x) = P[Y = 1|X = x]$ and

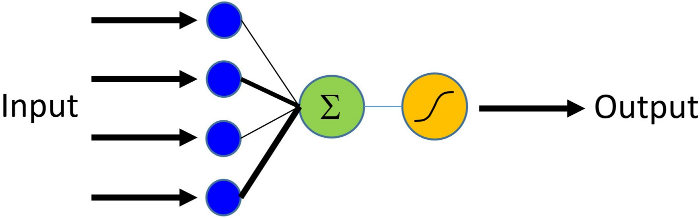
$$logit(p(x)) = \log \frac{p(x)}{1 - p(x)} = \log \frac{P[Y = 1|X = x]}{P[Y = 0|X = x]} = x^T \beta$$

for some vector $\beta \in \mathbb{R}^{d+1}$.

Note: Here we denote

$$x^T \beta = \beta_0 + \beta_1 x_1 + \ldots + \beta_d x_d$$

Logistic regression: Assumptions



Parameter estimation: maximum likelihood

- Remember that for Bernoulli r.v. with parameter p

$$P[Y = y] = p^y (1-p)^{1-y}, \quad y \in \{0, 1\}$$

- Given samples $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$, we have

$$L(\beta) = \prod_{i=1}^{n} p(x_i, \beta)^{y_i} (1 - p(x_i, \beta))^{1-y_i}$$

- Maximum likelihood (ML): maximize this likelihood function

- The optimization needs to be performed by a numerical optimization method
- Penalties can be added to regularize the problem to avoid overfitting

$$\min_{\beta} -\ell(\beta) + \alpha\|\beta\|_1$$

or

$$\min_{\beta} -\ell(\beta) + \alpha\|\beta\|_2$$

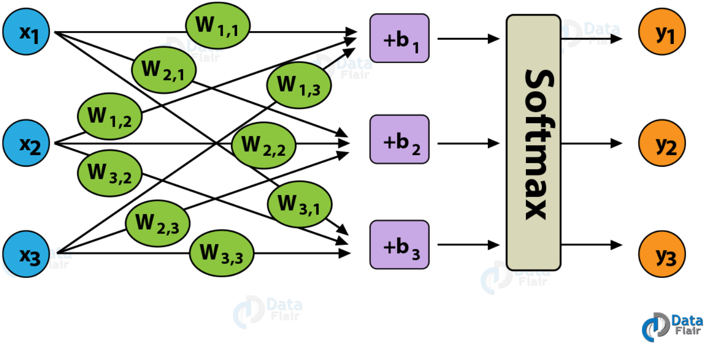
Logistic regression with more than 2 classes

- Suppose now the response can take any of $\{1, \ldots, K\}$ values
- We use the categorical distribution instead of the Bernoulli distribution

$$P[Y = k | X = x] = p_k(x), \quad \sum_{k=1}^{K} p_k(x) = 1.$$

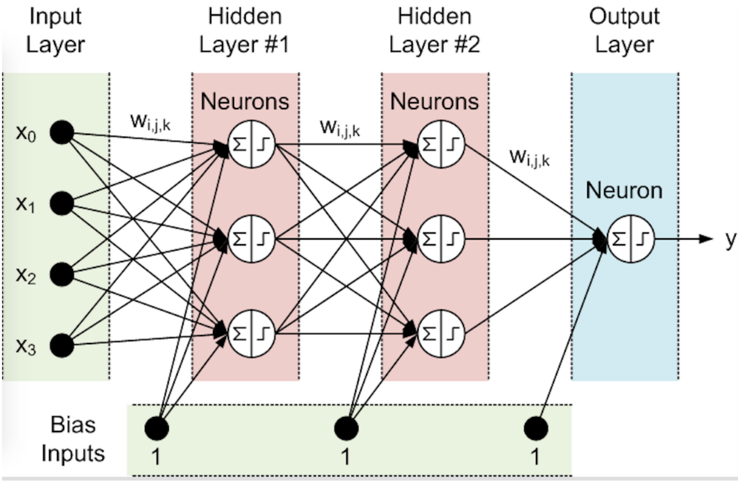
- Model

$$p_k(x) = \frac{e^{x^T \beta^{(k)}}}{\sum_{k=1}^{K} e^{x^T \beta^{(k)}}}$$



x1
x2
x3
W 1,1
W 2,1
W 1,2
W 1,3
W 2,2
W 3,2
W 3,1
W 2,3
W 3,3
+b 1
+b 2
+b 3
Softmax
y1
y2
y3
Data Flair

Feed-forward neural network



Classification: Linear discriminant analysis

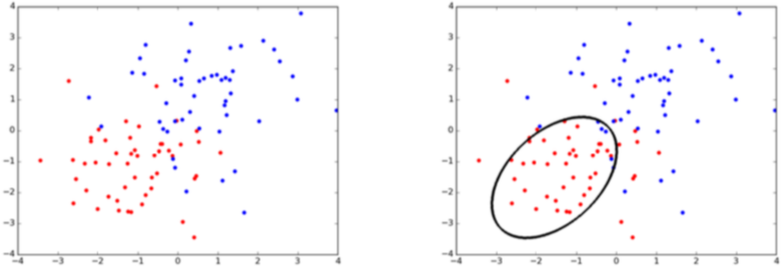
Classification

- Suppose we work with outputs $\mathcal{Y} = \{1, 2, \ldots, K\}$, and $\mathcal{X}$ is a subset of $\mathbb{R}^d$
- Goal: Given input X, we want to model the probability that Y condition on X

$$P[Y = i|X = x], \quad i \in \mathcal{Y}$$

- But some time, $P[X = x|Y = i]$ is easier to model!

Bayes' formula



$$P(X = x | Y = \text{red}).$$

Bayes' formula

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Linear discriminant analysis

Suppose

- $Y \in \{1, 2, \ldots, K\}$
- $P(Y = i) = \pi_i, \quad i = 1, 2, \ldots, K.$
- $P(X = x|Y = i) \sim f_i(x)$

Then

$$P(Y = i|X = x) = \frac{P(X = x|Y = i)P(Y = i)}{\sum_{j=1}^{K} P(X = x|Y = j)P(Y = j)}$$
$$= \frac{f_i(x)\pi_i}{\sum_{j=1}^{K} f_j(x)\pi_j}$$

Model $P(X = x|Y = i)$ using Gaussian distributions

The natural model for $f_i(x)$ is the multivariate Gaussian distribution

$$f_i(x) = \frac{1}{\sqrt{(2\pi)^p \det(\Sigma_i)}} e^{-\frac{1}{2}(x-\mu_i)^T \Sigma_i^{-1}(x-\mu_i)}, \quad x \in \mathbb{R}^p$$

- μ: mean vector
- Σ: covariance matrix

$$\Sigma = E[(X-\mu)^T(X-\mu)]$$

Model $P(X = x|Y = i)$ using Gaussian distributions

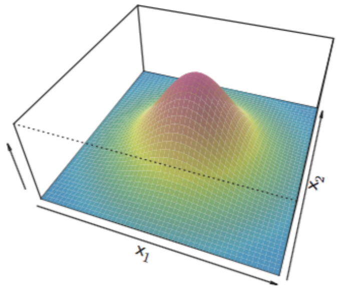


FIGURE 4.5. *Two multivariate Gaussian density functions are shown, with* $p = 2$. Left: *The two predictors are uncorrelated.* Right: *The two variables have a correlation of* 0.7.

LDA and QDA

The natural model for $f_i(x)$ is the multivariate Gaussian distribution

$$f_i(x) = \frac{1}{\sqrt{(2\pi)^p \det(\Sigma_i)}} e^{-\frac{1}{2}(x-\mu_i)^T \Sigma_i^{-1}(x-\mu_i)}, \quad x \in \mathbb{R}^p$$

- Linear discriminant analysis (LDA): We assume

$$\Sigma_1 = \Sigma_2 = \ldots = \Sigma_K$$

- Quadratic discriminant analysis (QDA): general cases

Parameter estimation

We need to estimate

- An estimate of the class probabilities π_i
- Estimate the mean vectors $\mu_1, \ldots, \mu_K$
- Estimate the covariance matrices $\Sigma_1, \ldots, \Sigma_K$ (or Σ for LDA)

Parameter estimation: LDA

Suppose we have dataset $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$ where n_i observations have label i.

- An estimate of the class probabilities π_i

$$\hat{\pi}_i = \frac{n_i}{n}$$

- Estimate the mean vectors

$$\hat{\mu}_i = \frac{1}{n_i} \sum_{y_j = i} x_j$$

- Estimate the covariance matrix Σ

$$\hat{\Sigma} = \frac{1}{N - K} \sum_{i=1}^{K} \sum_{y_j = i} (x_j - \hat{\mu}_i)(x_j - \hat{\mu}_i)^T$$

Decision rule

Suppose we have dataset $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$ where n_i observations have label i.

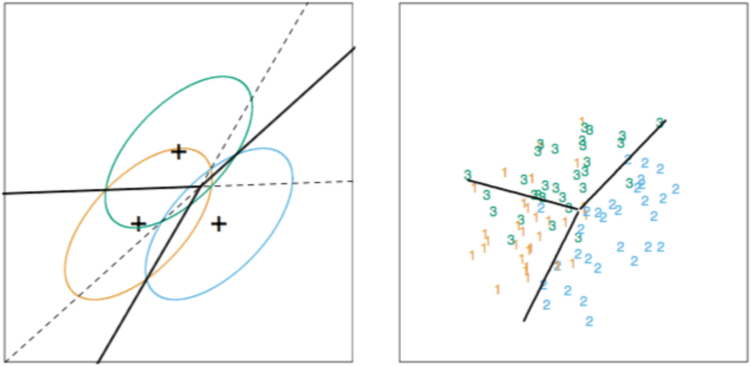
A new instance x arrives, how to predict label of x?

- Compute $\hat{\pi}_i$, $\hat{\mu}_i$ and $\hat{\Sigma}$
- Compute

$$P(Y = i|X = x) \approx p_k(x) = \frac{f_i(x, \hat{\mu}_i, \hat{\Sigma})\hat{\pi}_i}{\sum_{j=1}^{K} f_j(x, \hat{\mu}_j, \hat{\Sigma})\hat{\pi}_j}$$

- Bayes classifier: assign an observation to the class for which the posterior probability $p_k(x)$ is greatest.

LDA



LDA: linearity of the decision boundary

Note that

$$p_i(x) = \frac{f_i(x, \hat{\mu}_i, \hat{\Sigma})\hat{\pi}_i}{\sum_{j=1}^K f_j(x, \hat{\mu}_j, \hat{\Sigma})\hat{\pi}_j}$$

and

$$f_i(x) = \frac{1}{\sqrt{(2\pi)^p \det(\hat{\Sigma})}} e^{-\frac{1}{2}(x-\hat{\mu}_i)^T \Sigma^{-1}(x-\hat{\mu}_i)}, \quad x \in \mathbb{R}^p$$

Thus

$$\log \frac{p_i(x)}{p_k(x)} = \log \frac{\hat{\pi}_i}{\hat{\pi}_k} - \frac{1}{2}(\hat{\mu}_i + \hat{\mu}_k)^T \hat{\Sigma}^{-1}(\hat{\mu}_i - \hat{\mu}_k) + x^T \hat{\Sigma}^{-1}(\hat{\mu}_i - \hat{\mu}_k)$$
$$= \beta_0 + x^T \beta$$

How is that different from logistic regression?

Recall that for logistic regression

$$\log \frac{P[Y = 1|X = x]}{P[Y = 0|X = x]} = \beta_0 + x^T \beta$$

- Both methods use linear decision boundary
- Both are simple, and often perform very well.

  However
- The probability models are different
- The estimations are different

Practical problem when $n \ll p$

- Estimating covariance matrices when $n \ll p$ is challenging
- The sample covariance $\hat{\Sigma}$ is singular when $n \ll p$
- Need regularization

LDA

```
>>> import numpy as np
>>> from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
>>> X = np.array([[-1, -1], [-2, -1], [-3, -2], [1, 1], [2, 1], [3, 2]])
>>> y = np.array([1, 1, 1, 2, 2, 2])
>>> clf = LinearDiscriminantAnalysis()
>>> clf.fit(X, y)
LinearDiscriminantAnalysis(n_components=None, priors=None, shrinkage=None,
              solver='svd', store_covariance=False, tol=0.0001)
>>> print(clf.predict([[-0.8, -1]]))
[1]
```