Classification

Lecture 7 of “Mathematics and AI”

Outline

1. Classification

2. The bias-variance tradeoff

3. Discriminative models

   Logistic regression, K Nearest Neighbors

4. Generative models

   Linear discriminant analysis, quadratic discriminatn analysis, naïve Bayes

Classification

Classification

Query: ~~***How much***~~ ***What*** is this?

Binary classification: K=2

Possible answers: [‘Cat’, ‘Dog’]

Multinomial classification: K>2

Possible answers: [‘Cat’, ‘Dog’, ‘Bird’,…]

Image source: Photo by Dr. Schwarze of Dr. Schwarze’s cat Emmy

Quality of fit for classification problems

Mean-squared error $\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{f}(x_i)\right)^2$ same as for regression

Error rate $\mathrm{ER} = \frac{1}{n}\sum_{i=1}^{n} I\left(y_i \neq \hat{f}(x_i)\right)$ for classification specifically

True-positive rate $\mathrm{TPR} = \frac{1}{n}\frac{\sum_{i:y_i=1}^{n} I\left(y_i \neq \hat{f}(x_i)\right)}{\sum_{i:y_i=1}^{n} 1}$ for binary classification

False-positive rate $\mathrm{FPR} = \frac{1}{n}\frac{\sum_{i:y_i=0}^{n} I\left(y_i \neq \hat{f}(x_i)\right)}{\sum_{i:y_i=0}^{n} 1}$

Bias-variance tradeoff

Bias-variance tradeoff

How sensitive should our model be to our training data?

Expected mean squared error

$$E[\text{MSE}] = E\left[\left(y_0 - \hat{f}(x_0)\right)^2\right] = \text{Var}[\hat{f}(x_0)] + \left[\text{Bias}[\hat{f}(x_0)]\right]^2 + \text{Var}[\varepsilon]$$

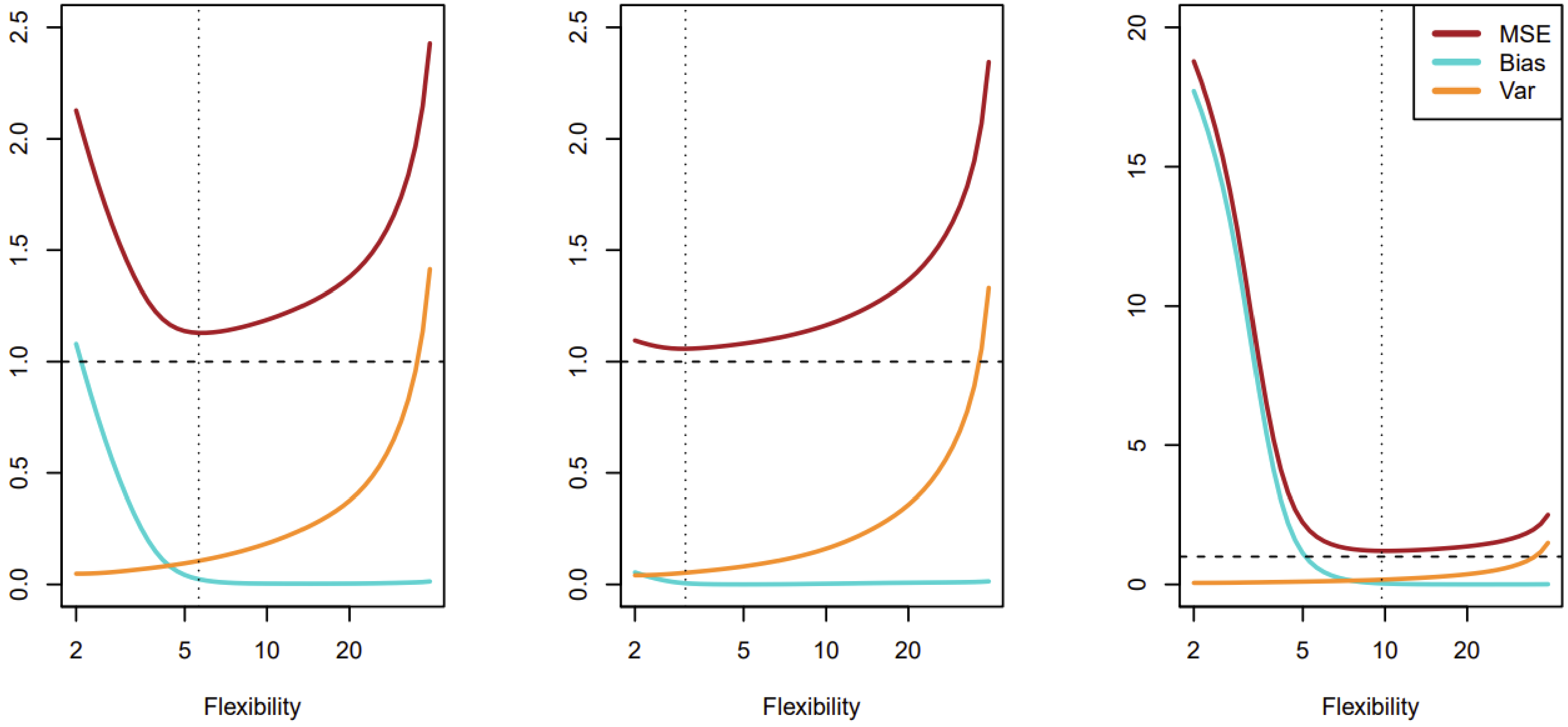


FIGURE 2.12. *Squared bias (blue curve), variance (orange curve), Var(ϵ) (dashed line), and test MSE (red curve) for the three data sets in Figures 2.9–2.11. The vertical dotted line indicates the flexibility level corresponding to the smallest test MSE.*

Image source: Introduction to Statistical Learning in Python

Discriminative models

Discriminative models

Estimate $p(Y = k | X = x)$ (or a related quantity) from data

Examples: K nearest neighbors, logistic regression

Logistic regression: The model

Why not linear regression?

Binary classification via logistic regression

- $p(Y = 1|X = x)$ should grow as $\exp(\beta_0 + \beta_1 X)$ with X for small probabilities
- $p(Y = 0|X = x)$ should grow as 1 with X for small probabilities
- $p(Y = 1|X = x)$ is a logistic function of X:

$$p(Y = 1|X = x) = \frac{\exp(\beta_0 + \beta_1 x)}{1 + \exp(\beta_0 + \beta_1 x)}$$

Logistic regression: The model

- $p(Y = 1|X = x)$ is a logistic function of X:

$$p(Y = 1|X = x) = \frac{\exp(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p)}{1 + \exp(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p)}$$

Logistic regression: The interpretation

Logistic model $p(Y = 1|X = x) = \frac{\exp(\beta_0 + \beta_1 x)}{1 + \exp(\beta_0 + \beta_1 x)}$

has $log\left(\frac{p(Y = 1|X = x)}{1 - p(Y = 1|X = x)}\right) = \beta_0 + \beta_1 x$

Where the left-hand side are the log-odds for a positive result

$$log\left(\frac{p(Y = 1|X = x)}{p(Y = 0|X = x)}\right) = log\left(\frac{p_{True}}{p_{False}}\right)$$

Logistic model assumes linear increase of log-odds with independent variable!

Logistic regression: The fit

Likelihood function

$$L(\beta_0, \beta_1) = \prod_{i:y_i=1} p(Y = y_i|X = x_i) \prod_{i:y_i=0} [1 - p(Y = y_i|X = x_i)]$$

Log-likelihood function

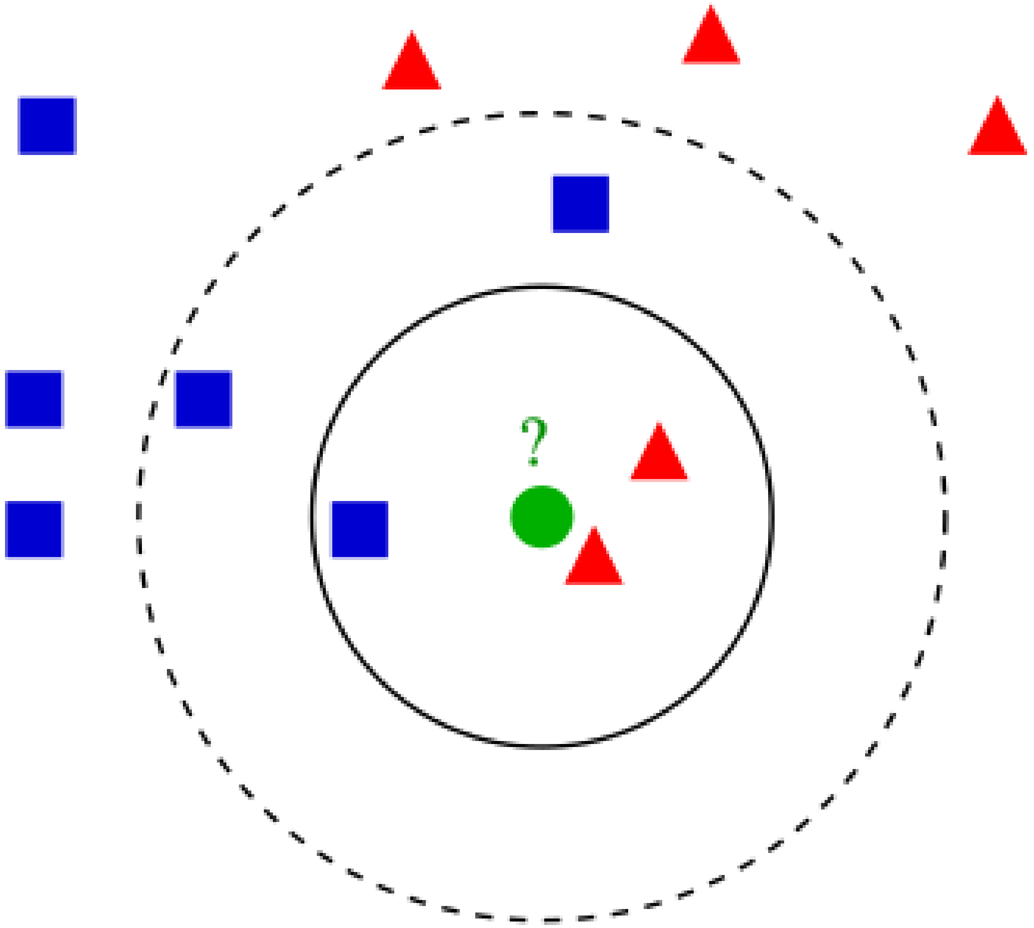
$$\log(L(\beta_0, \beta_1)) = \sum_{\substack{i:\\ y_i=1}} \log(p(Y = y_i|X = x_i)) + \sum_{\substack{i:\\ y_i=0}} \log([1 - p(Y = y_i|X = x_i)])$$

Obtain parameter estimates for β_0, β_1 via (log-)likelihood maximization.

K nearest neighbors

Interpolate $p(Y = y_i | X = x_i)$ from the k nearest data points in the training set

- Non-parametric method
- Benefits from large training sets



Exercise

Generative models

Generative models

Estimate $p(X = x | Y = k)$ from data and use Bayes theorem

Examples: Linear discriminant analysis (LDA), quadratic discriminant analysis (QDA), naïve Bayes

Bayes classifier

For each query x_i assign response $\hat{f}(x_i) = k$ that has the largest conditional probability $p(Y = k | X = x_i)$

Normal distribution

$$f(x) = \frac{\exp\left(-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)\right)}{(2\pi)^{p/2}|\Sigma|^{1/2}}$$

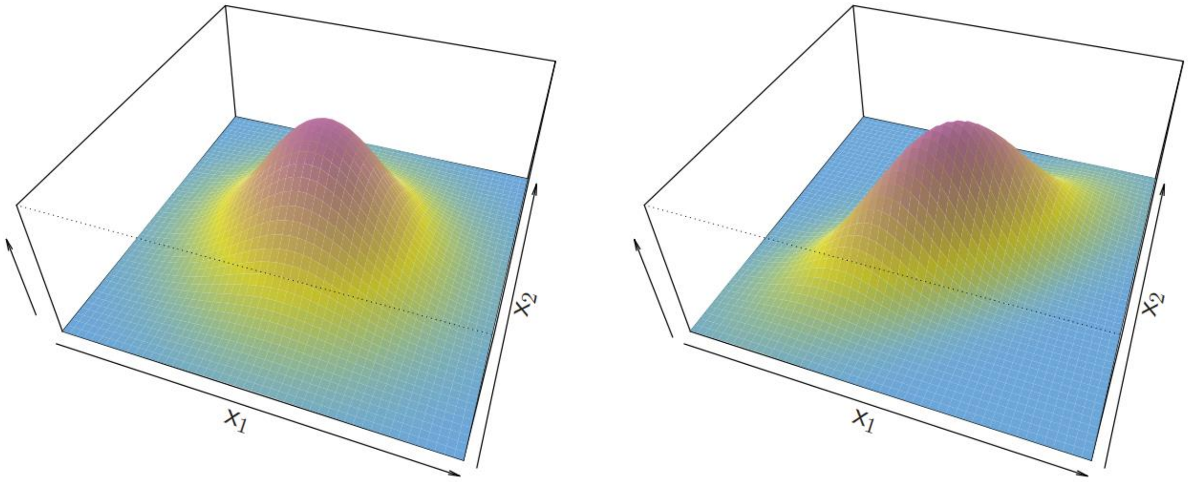


Image source: Introduction to Statistical Learning in Python

Generative models

Estimate $p(X = x|Y = k)$ from data and use Bayes theorem:

$$p(Y = k|X = x) = \frac{p(Y = k)p(X = x|Y = k)}{p(X = x)}$$

$$p(Y = k|X = x) = \frac{\pi_k f_k(x)}{\sum_{l=1}^{K} \pi_l f_l(x)}$$

How do we estimate $f_k(x)$?

Approach 1: Assume all $f_k(x)$ are Gaussian distributions with the same variance/ covariance matrix for each class (LDA)

Approach 2: Assume all $f_k(x)$ are Gaussian distributions with different variances/ covariance matrices for each class (QDA)

Approach 3: Assume $f_k(x)$ factorizes within each response class (naïve Bayes)

DARTMOUTH

Linear discriminant analysis

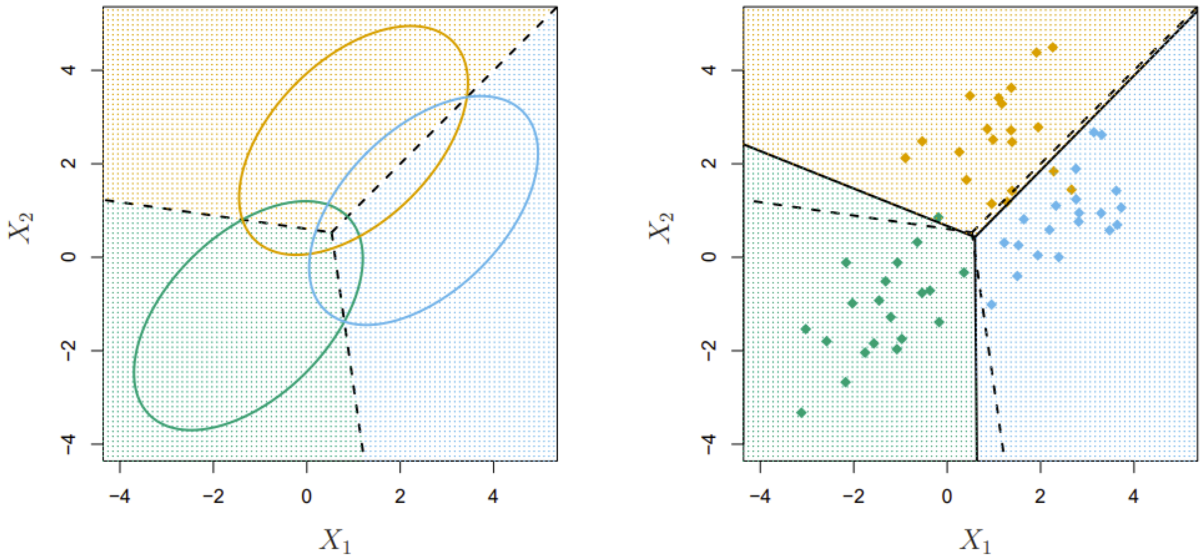


FIGURE 4.6. *An example with three classes. The observations from each class are drawn from a multivariate Gaussian distribution with $p = 2$, with a class-specific mean vector and a common covariance matrix.* Left: *Ellipses that contain 95 % of the probability for each of the three classes are shown. The dashed lines are the Bayes decision boundaries.* Right: 20 *observations were generated from each class, and the corresponding LDA decision boundaries are indicated using solid black lines. The Bayes decision boundaries are once again shown as dashed lines.*

Image source: Introduction to Statistical Learning in Python

Quadratic discriminant analysis

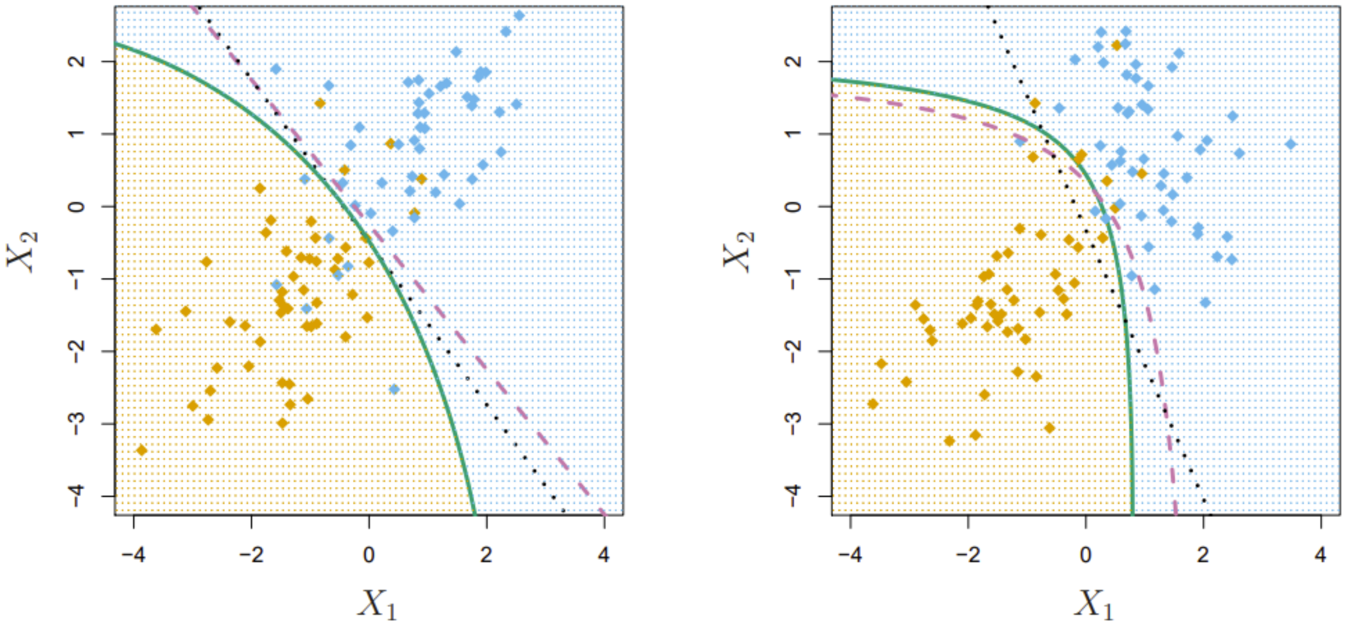


FIGURE 4.9. Left: *The Bayes (purple dashed), LDA (black dotted), and QDA (green solid) decision boundaries for a two-class problem with* $\mathbf{\Sigma}_1 = \mathbf{\Sigma}_2$*. The shading indicates the QDA decision rule. Since the Bayes decision boundary is linear, it is more accurately approximated by LDA than by QDA.* Right: *Details are as given in the left-hand panel, except that* $\mathbf{\Sigma}_1 \neq \mathbf{\Sigma}_2$*. Since the Bayes decision boundary is non-linear, it is more accurately approximated by QDA than by LDA.*

Image source: Introduction to Statistical Learning in Python

Exercise