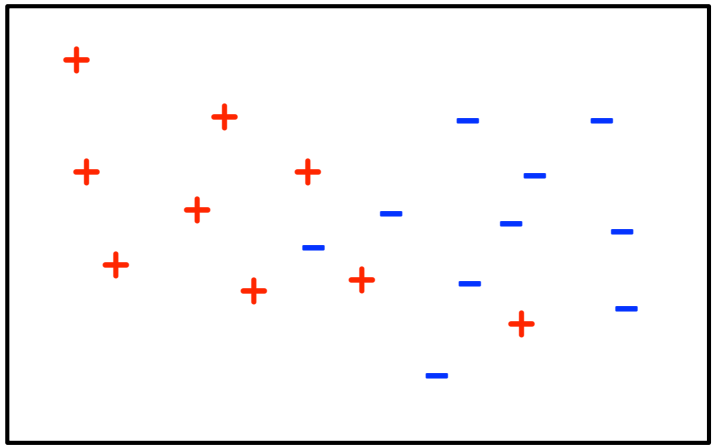
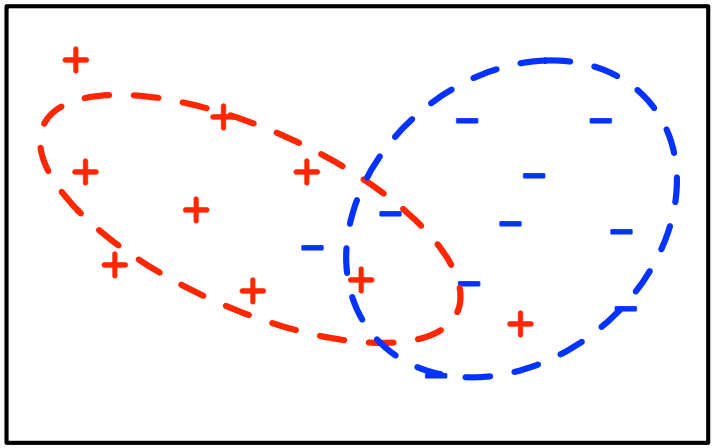
Introduzione al Machine Learning: Classificazione generativa

Vincenzo Bonifaci

Approccio generativo alla classificazione





Durante l'apprendimento:

- Fai il fit di una distribuzione di probabilità per ciascuna classe

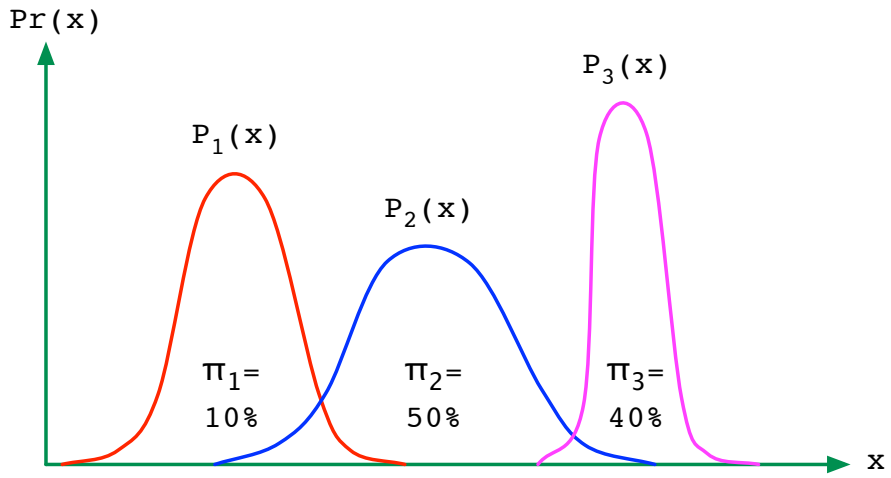
Per classificare un nuovo punto:

- Determina da quale distribuzione di probabilità è più verosimile che il punto sia stato generato

Modelli generativi

Esempio:

- Spazio di input $\mathcal{X} = \mathbb{R}$
- Spazio di output $\mathcal{Y} = \{1, 2, 3\}$



Per ciascuna classe j, stimeremo:

- la probabilità a priori di quella classe, $\pi_j \stackrel{\text{def}}{=} \Pr(y = j)$
- la distribuzione di x in quella classe, $P_j(x) \stackrel{\text{def}}{=} \Pr(x|y = j)$

Per classificare x: scegli l'etichetta y che massimizza $\Pr(y|x)$

Regola di Bayes

Per due eventi A e B,

$$\Pr(A|B) = \frac{\Pr(A) \cdot \Pr(B|A)}{\Pr(B)}$$

Approccio *generativo* perché cerca di apprendere la *distribuzione congiunta* che genera i dati: $\Pr(x, y) = \Pr(y)\Pr(x|y) = \pi_j P_j(x)$ se $y = j$

Giustificazione del criterio Bayesiano

Ricordiamo che la funzione costo 0-1 è:

$$\ell(h,(x,y)) \stackrel{\text{def}}{=} \begin{cases} 0 & \text{se } h(x) = y \\ 1 & \text{se } h(x) \neq y \end{cases}$$

Il rischio atteso *condizionato* all'osservazione di x è

$$\mathbb{E}[\ell|x] = \Pr[h(x) \neq y|x] = 1 - \Pr[h(x) = y|x]$$

Minimizzarlo significa scegliere y che massimizza $\Pr(y|x)$

Classificatore Bayesiano

$$h(x) = \operatorname{argmax}_{y \in \mathcal{Y}} \Pr(y|x)$$

Analisi del discriminante

Per ogni $x \in \mathcal{X}$ e ogni etichetta $j \in \mathcal{Y}$,

$$\Pr(y = j|x) = \frac{\Pr(y = j) \cdot \Pr(x|y = j)}{\Pr(x)} = \frac{\pi_j P_j(x)}{\Pr(x)}$$

Il termine $\Pr(x)$ non dipende da j

Dato x, l'etichetta j più verosimile è quella che massimizza $\pi_j P_j(x)$

La quantità $\delta_j(x) \stackrel{\text{def}}{=} \log(\pi_j P_j(x))$ è chiamata *discriminante*

Dato x, l'etichetta j più verosimile è quella che massimizza $\delta_j(x)$

Fit di un modello generativo

Esempio: Classificazione di bottiglie di vino in base alla cantina di provenienza (dataset `wine`)

Training set: 130 bottiglie

- Cantina 1: 43 bottiglie; Cantina 2: 54 bottiglie; Cantina 3: 33 bottiglie
- Per ogni bottiglia, 13 feature: `Alcool`, `Acido malico`, `Ceneri`, `Alcalinità delle ceneri`, `Magnesio`, `Fenoli totali`, `Flavonoidi`, `Fenoli non flavonoidi`, `Proantocianina`, `Intensità di colore`, `Tonalità`, `OD280/OD315`, `Prolina`

Test set: 48 bottiglie

Pesi delle classi:

$\pi_1 = 43/130 \approx 0.33 \quad \pi_2 = 54/130 \approx 0.41 \quad \pi_3 = 33/130 \approx 0.26$

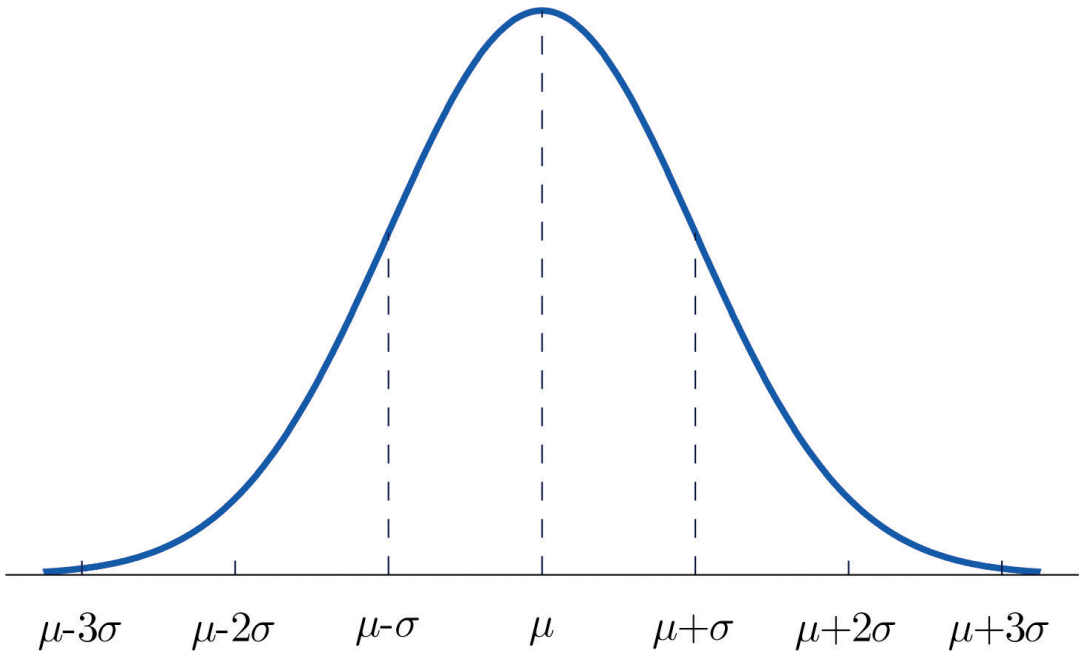
Vogliamo stimare le distribuzioni P_1, P_2, P_3

Supponiamole *gaussiane* e (per iniziare) dipendenti da un'unica feature: `Alcool`

La Gaussiana univariata

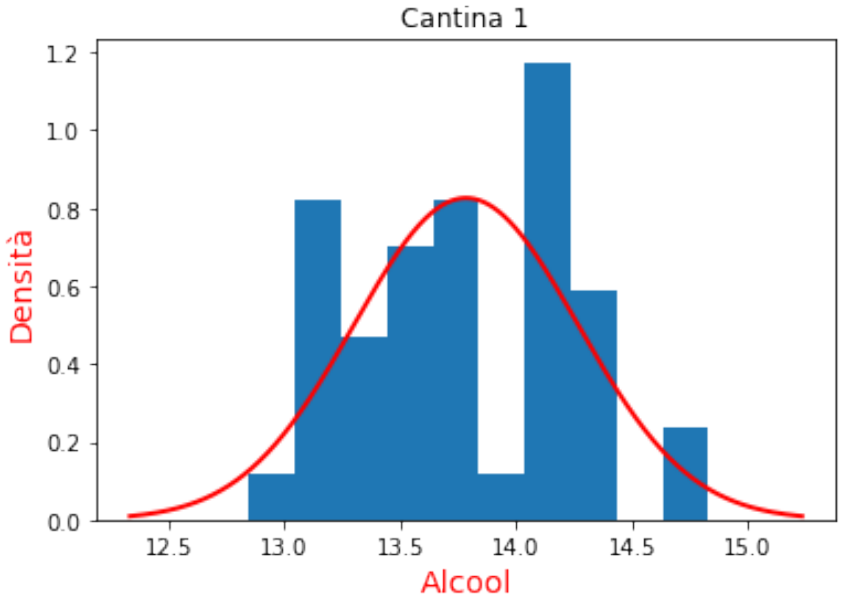
La Gaussiana $N(\mu, \sigma^2)$ ha media μ, varianza σ^2, e densità di probabilità

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



Distribuzione per la Cantina 1

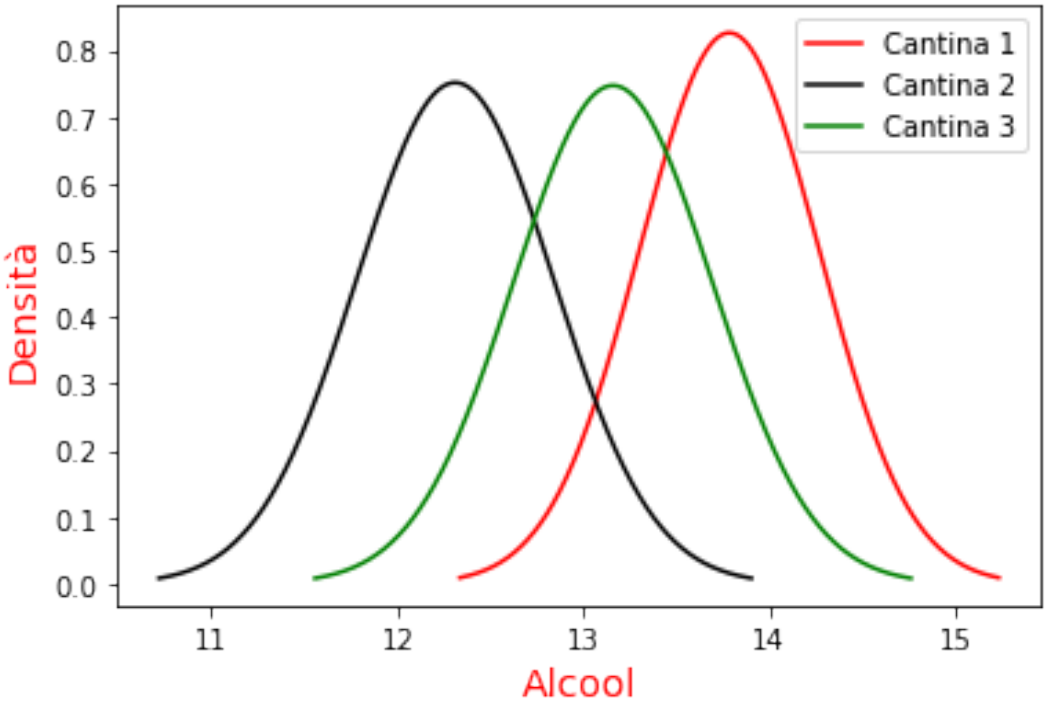
Unica feature che utilizziamo: `Alcool`



Media	$\mathbb{E}\,x$	Media stimata	$(1/m)\sum_i x^{(i)}$
Varianza	$\mathbb{E}\,(x-\mu)^2$	Var. stimata	$(1/m)\sum_i (x^{(i)}-\mu)^2$

Nell'esempio: media stimata $\mu \approx 13.78$, varianza stimata $\sigma^2 \approx 0.23$

Analisi del discriminante unidimensionale



$\pi_1 = 0.33,\ P_1 = N(13.78, 0.23)$

$\pi_2 = 0.41,\ P_2 = N(12.31, 0.28)$

$\pi_3 = 0.26,\ P_3 = N(13.15, 0.28)$

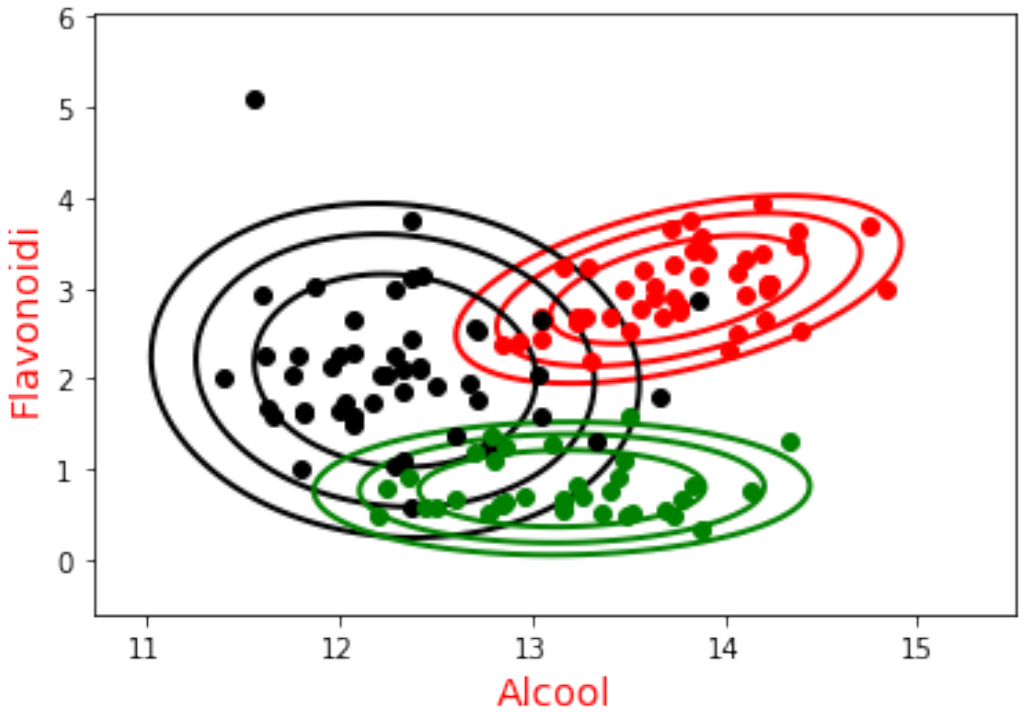
Per classificare x: determina l'etichetta j che massimizza $\pi_j P_j(x)$

Errore di test: $17/48 \approx 35\%$

Aggiunta di feature

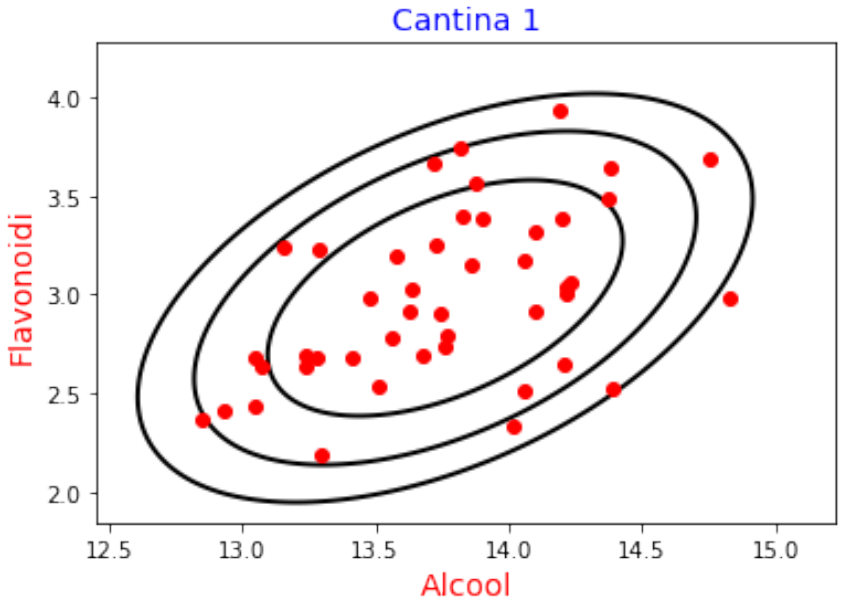
Più feature permettono una maggiore separazione tra le classi

Aggiungiamo la variabile `Flavonoidi`



Errore di test diventa $3/48 \approx 6\%$

La Gaussiana bivariata



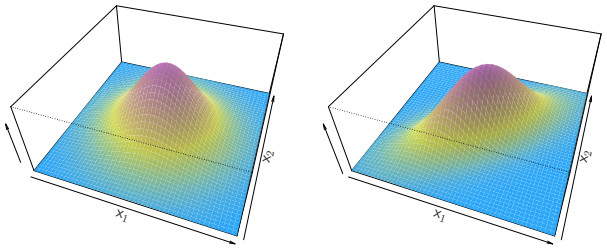
Modelliamo la classe 1 con una Gaussiana bivariata:

media $\mu = \begin{pmatrix} 13.7 \\ 2.98 \end{pmatrix}$ matrice di covarianza $\Sigma = \begin{pmatrix} 0.22 & 0.09 \\ 0.09 & 0.17 \end{pmatrix}$

$$\mu_i = \mathbb{E}\, x_i$$

$$\Sigma_{ij} = \mathrm{Cov}(x_i, x_j) = \mathbb{E}[(x_i - \mu_i)(x_j - \mu_j)]$$

Densità della Gaussiana bivariata



- Media $\mu = (\mu_1, \mu_2) \in \mathbb{R}^2$
- Matrice di covarianza $\Sigma = \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix} \in \mathbb{R}^{2\times 2}$

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

- $|\Sigma|$ qui indica il *determinante* di Σ

La Gaussiana multivariata

$N(\mu, \Sigma)$: Gaussiana in $\mathbb{R}^d$

- media: $\mu \in \mathbb{R}^d$
- covarianza: $\Sigma \in \mathbb{R}^{d \times d}$
- μ è il vettore delle medie:

$$\mu_1 = \mathbb{E}\, x_1, \mu_2 = \mathbb{E}\, x_2, \ldots, \mu_d = \mathbb{E}\, x_d$$

- Σ è la matrice di covarianza:

$$\Sigma_{ij} = \mathrm{Cov}(x_i, x_j)$$

Densità:

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

Analisi del discriminante quadratica (QDA)

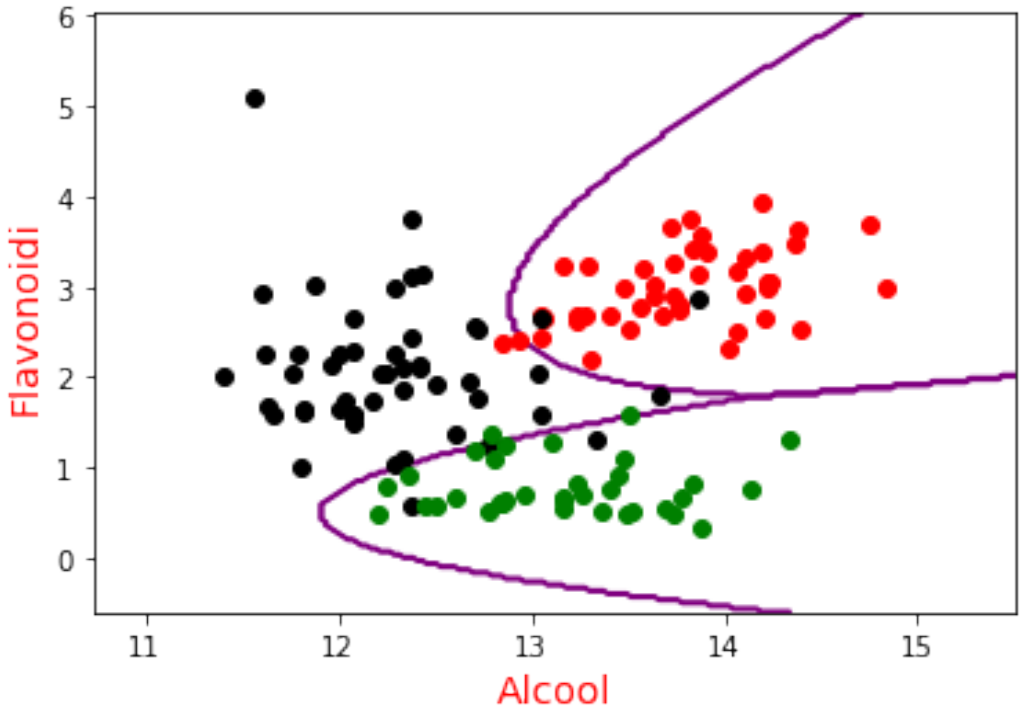
1. Calcola le probabilità a priori π_j per ogni classe j
2. Fai il fit di una gaussiana multivariata $P_j(x)$ per ogni classe j:
   - Calcola il vettore di media empirica $\mu^{(j)}$
   - Calcola la matrice di covarianza empirica $\Sigma^{(j)}$
3. Dato x, restituisci j che massimizza $\pi_j P_j(x)$
   (equivalentemente: che massimizza $\delta_j(x)$)

Analisi discriminante quadratica (QDA)

Si può dimostrare che il discriminante di ogni classe j è una funzione quadratica di x

Le *frontiere di decisione* sono determinate da equazioni quadratiche in x

QDA per il dataset `wine`



Considerando tutte e 13 le feature, l'errore di test diventa zero

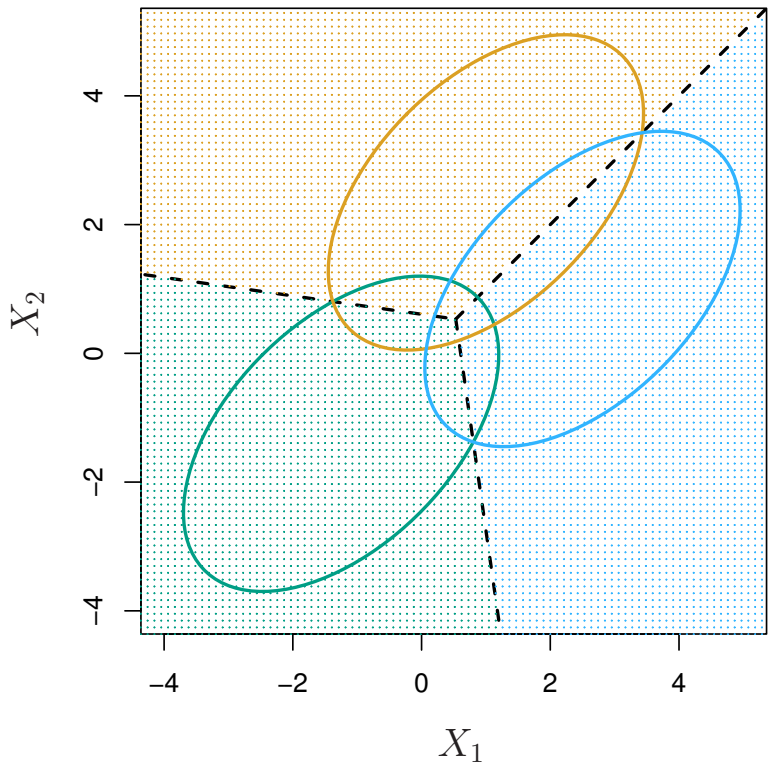
Analisi discriminante lineare (LDA)

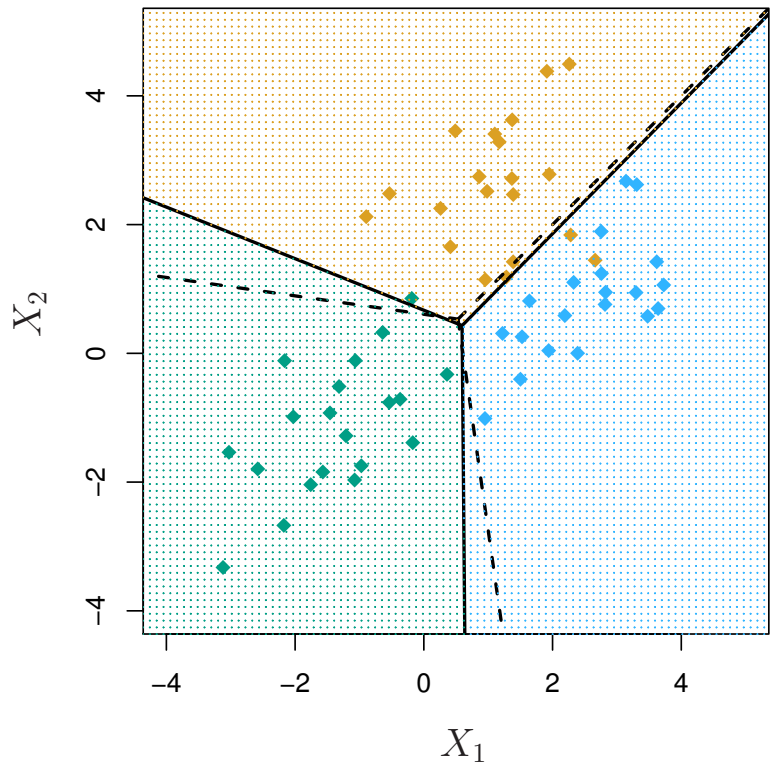
L'*analisi discriminante lineare* procede come la QDA ma assume che la matrice di covarianza Σ sia comune a tutte le classi (anche se empiricamente si osservano matrici di covarianza distinte)

Le frontiere di decisione sono determinate da equazioni lineari in x

Per stimare Σ si utilizza la formula $\sum_j \pi_j \Sigma^{(j)}$

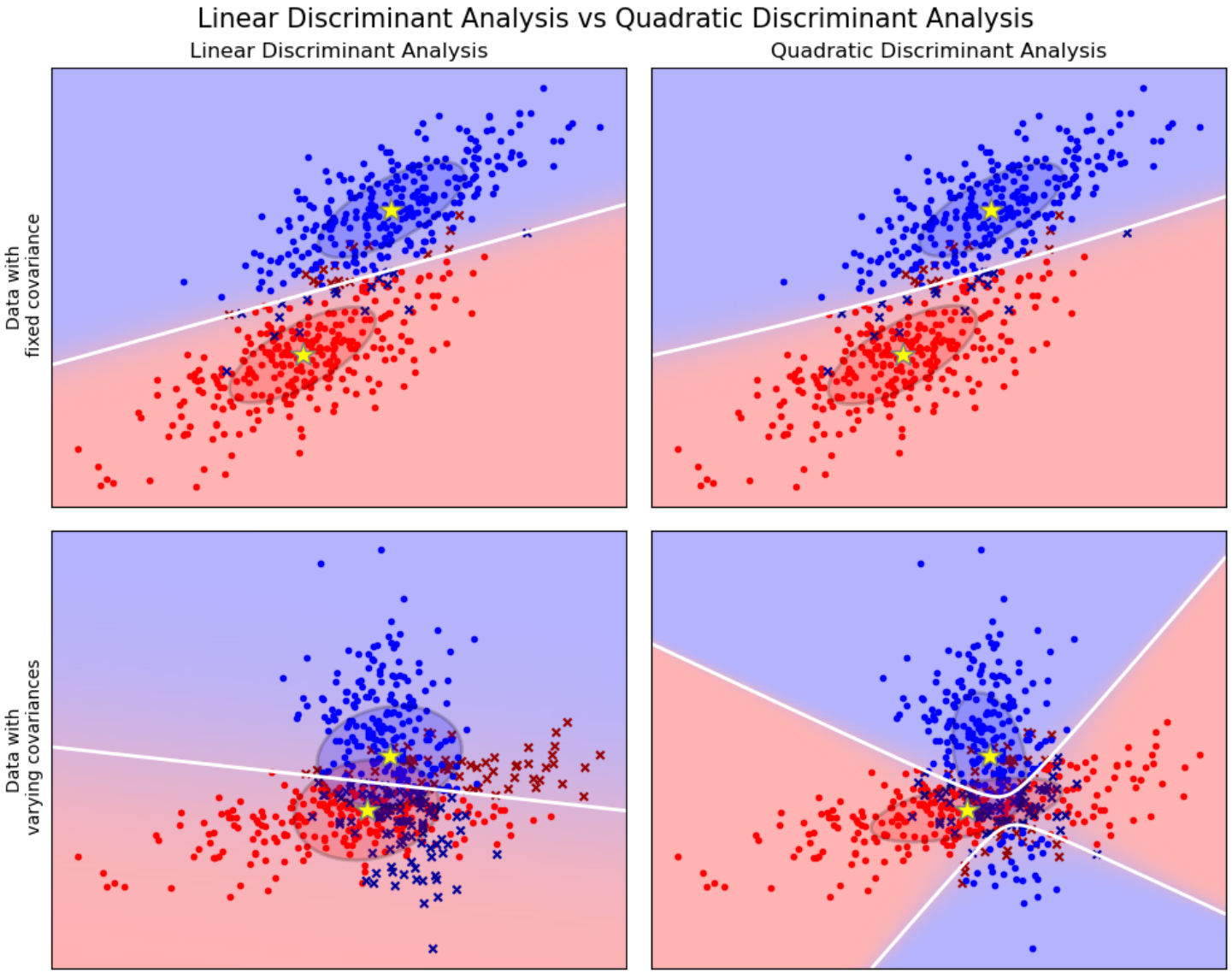
LDA: Esempio





- Sinistra: ellissi contenenti il 95% di probabilità per ciascuna delle tre classi
- Destra: le frontiere di decisione determinate da 20 osservazioni

LDA vs. QDA


Linear Discriminant Analysis vs Quadratic Discriminant Analysis
Linear Discriminant Analysis
Quadratic Discriminant Analysis
Data with fixed covariance
Data with varying covariances

Modellazione generativa con altre distribuzioni

La modellazione generativa non è ristretta all'uso di distribuzioni gaussiane

Possibilità (tutti esempi di *famiglie esponenziali*):

- Distribuzione Gamma (valori reali non negativi)
- Distribuzione di Poisson (valori interi non negativi)
- Distribuzione categorica (valori in un insieme finito)
- Distribuzione gaussiana (valori reali)

Tutte le distribuzioni di famiglie esponenziali possono essere stimate con relativa facilità (utilizzando il cosiddetto principio di massima verosimiglianza)

Naive Bayes

Se il numero di variabili d è molto alto, l'elaborazione delle matrici di covarianza (matrici $d \times d$) diventa impraticabile

Il metodo *Naive Bayes* offre una alternativa più rozza ma efficiente

Naive Bayes

1. Stima una distribuzione condizionata $\Pr_i$ per ciascuna variabile x_i, separatamente e indipendentemente
2. Assumi $\Pr(x|y) = \Pr_1(x_1|y) \cdot \Pr_2(x_2|y) \ldots \cdot \Pr_d(x_d|y)$
3. Dato x, restituisci j che massimizza $\pi_j \Pr(x|y = j)$

Attenzione: L'assunzione di indipendenza porta tipicamente ad una stima inaccurata delle *probabilità*

Ciononostante, la qualità della *classificazione* può essere adeguata e il risparmio computazionale è notevole

Esempio: Classificazione di messaggi (spam/no spam)

Dizionario: $D = \{$ `a`, `aardvark`, `...`, `buy`, `...`, `zygmurgy` $\}$
Dimensione: $d = |D| = 5000$

Rappresentiamo un messaggio con un vettore $x \in \{0,1\}^d$:

$$x = \begin{bmatrix} 1 \\ 0 \\ \dots \\ 1 \\ \dots \\ 0 \end{bmatrix}$$

dove $x_k = 1 \Leftrightarrow$ il messaggio contiene la k-esima parola di D

$$y \in \{1\,(\text{spam}), 0\,(\text{non spam})\}$$

Un modello generativo per una $\Pr(x|y)$ categorica richiederebbe $2^d - 1$ parametri!

Esempio: Classificazione di messaggi (spam/no spam)

Usando l'assunzione Naive Bayes:

$$\Pr(x_1, \ldots, x_d | y) = \prod_{k=1}^{d} \Pr_k(x_k | y)$$

dove ciascuna $\Pr_k$ è specificata dai due parametri

$$\phi_{k|y=1} = \Pr(x_k = 1 | y = 1), \qquad \phi_{k|y=0} = \Pr(x_k = 1 | y = 0)$$

Inoltre modelliamo le probabilità a priori delle classi:

$$\pi_1 = \Pr(y = 1), \qquad \pi_0 = \Pr(y = 0) = 1 - \pi_1$$

In questo esempio i parametri scendono quindi a $2d + 1$

Esempio: Classificazione di messaggi (spam/no spam)

Il principio di massima verosimiglianza fornisce le seguenti stime:

$$\phi_{k|y=1} \approx \frac{\text{n. di messaggi spam con la parola } k}{\text{n. di messaggi spam}}$$

$$\phi_{k|y=0} \approx \frac{\text{n. di messaggi non spam con la parola } k}{\text{n. di messaggi non spam}}$$

$$\pi_1 \approx \frac{\text{n. di messaggi spam}}{\text{n. di messaggi}}$$

Si possono calcolare con un'unica passata sul dataset

Per classificare x, restituiamo come al solito

$$\operatorname*{argmax}_j \Pr(y = j|x) = \operatorname*{argmax}_j [\pi_j \Pr(x|y = j)]$$

Esempio: Classificazione di messaggi (spam/no spam)

Si può utilizzare anche una variante bayesiana delle stime, detta *Laplace smoothing*:

$$\phi_{k|y=1} \approx \frac{1 + \text{n. di messaggi spam con la parola } k}{2 + \text{n. di messaggi spam}}$$

$$\phi_{k|y=0} \approx \frac{1 + \text{n. di messaggi non spam con la parola } k}{2 + \text{n. di messaggi non spam}}$$

$$\pi_1 \approx \frac{1 + \text{n. di messaggi spam}}{2 + \text{n. di messaggi}}$$

Lo smoothing può attenuare il problema dei "*cigni neri*" (parole del dizionario mai osservate nei messaggi di training)