

Lecture 16

Generative classifiers (continued)

COURSE MATH 637: Mathematical Techniques in Data Science

WEBSITE <https://jacobcalvert.com/teaching/math637fall2026/>

INSTRUCTOR Jacob Calvert (calvert@udel.edu)

OFFICE HOURS MW 2:45pm–3:45pm, Ewing Hall 509

Plan for today

- **Recap: Quadratic discriminant analysis** 4 min
- Linear discriminant analysis 12 min
- Fisher's discriminant and PCA 12 min
- Too many parameters: naive Bayes 15 min
- Which assumption fails? 6 min

Recap: The idea of *generative* classifiers

Key idea: Instead of $\eta_k(x) = \mathbb{P}(Y = k \mid X = x)$, model $\mathbb{P}(X = x \mid Y = k)$, i.e., how data vary within each class—like a mixture model!

Recall: Then, to get $\eta_k(x)$, use Bayes' rule:

$$\eta_k(x) = \frac{\pi_k f_k(x)}{\sum_j \pi_j f_j(x)}, \quad \pi_k = \mathbb{P}(Y = k), \quad f_k = \text{density of } x \text{ in class } k.$$

- **QDA** = each f_k is Gaussian. Since labels known, **one** M step gives $\hat{\pi}_k, \hat{\mu}_k$, and $\hat{\Sigma}_k$ from class k 's examples

Recap: The idea of *generative* classifiers

Key idea: Instead of $\eta_k(x) = \mathbb{P}(Y = k \mid X = x)$, model $\mathbb{P}(X = x \mid Y = k)$, i.e., how data vary within each class—like a mixture model!

Recall: Then, to get $\eta_k(x)$, use Bayes' rule:

$$\eta_k(x) = \frac{\pi_k f_k(x)}{\sum_j \pi_j f_j(x)}, \quad \pi_k = \mathbb{P}(Y = k), \quad f_k = \text{density of } x \text{ in class } k.$$

- **QDA** = each f_k is Gaussian. Since labels known, **one** M step gives $\hat{\pi}_k, \hat{\mu}_k$, and $\hat{\Sigma}_k$ from class k 's examples
- then $\hat{\eta}_k(x) \propto \hat{\pi}_k \mathcal{N}(x \mid \hat{\mu}_k, \hat{\Sigma}_k)$ for any new x !

Recap: The idea of *generative* classifiers

Key idea: Instead of $\eta_k(x) = \mathbb{P}(Y = k \mid X = x)$, model $\mathbb{P}(X = x \mid Y = k)$, i.e., how data vary within each class—like a mixture model!

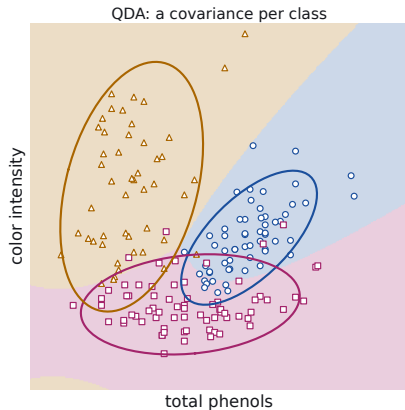
Recall: Then, to get $\eta_k(x)$, use Bayes' rule:

$$\eta_k(x) = \frac{\pi_k f_k(x)}{\sum_j \pi_j f_j(x)}, \quad \pi_k = \mathbb{P}(Y = k), \quad f_k = \text{density of } x \text{ in class } k.$$

- **QDA** = each f_k is Gaussian. Since labels known, **one** M step gives $\hat{\pi}_k, \hat{\mu}_k$, and $\hat{\Sigma}_k$ from class k 's examples
- then $\hat{\eta}_k(x) \propto \hat{\pi}_k \mathcal{N}(x \mid \hat{\mu}_k, \hat{\Sigma}_k)$ for any new x !
- with equal costs, predict $\arg \max_k \hat{\eta}_k(x)$

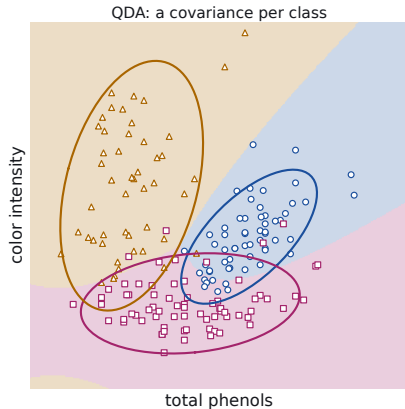
Recap: QDA's boundaries are quadratic

- decision boundary consists of x where $\arg \max_k \hat{\eta}_k(x)$ is not unique



Recap: QDA's boundaries are quadratic

- decision boundary consists of x where $\arg \max_k \hat{\eta}_k(x)$ is not unique
- last slide: $\hat{\eta}_k(x) \propto \hat{\pi}_k \mathcal{N}(x \mid \hat{\mu}_k, \hat{\Sigma}_k)$

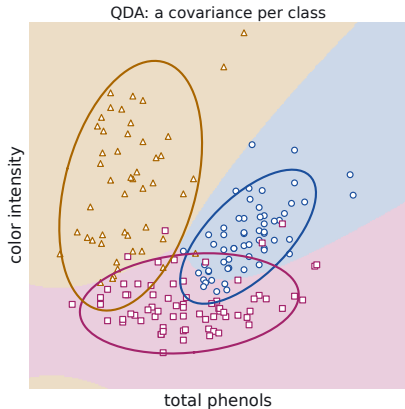


Recap: QDA's boundaries are quadratic

- decision boundary consists of x where $\arg \max_k \hat{\eta}_k(x)$ is not unique
- last slide: $\hat{\eta}_k(x) \propto \hat{\pi}_k \mathcal{N}(x \mid \hat{\mu}_k, \hat{\Sigma}_k)$
- so $\hat{\eta}_k(x)$ has same argmax as

$$\begin{aligned} \hat{\delta}_k(x) = & \log \hat{\pi}_k - \frac{1}{2} \log \det \hat{\Sigma}_k \\ & - \frac{1}{2} (x - \hat{\mu}_k)^T \hat{\Sigma}_k^{-1} (x - \hat{\mu}_k), \end{aligned}$$

the *discriminant* of class k



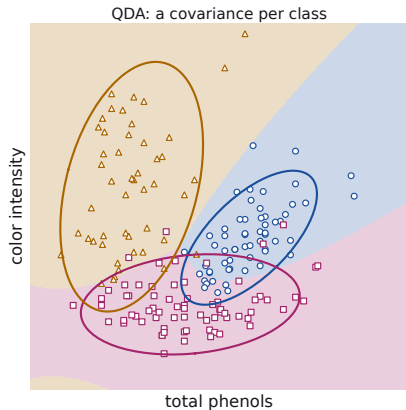
Recap: QDA's boundaries are quadratic

- decision boundary consists of x where $\arg \max_k \hat{\eta}_k(x)$ is not unique
- last slide: $\hat{\eta}_k(x) \propto \hat{\pi}_k \mathcal{N}(x \mid \hat{\mu}_k, \hat{\Sigma}_k)$
- so $\hat{\eta}_k(x)$ has same argmax as

$$\begin{aligned} \hat{\delta}_k(x) = & \log \hat{\pi}_k - \frac{1}{2} \log \det \hat{\Sigma}_k \\ & - \frac{1}{2} (x - \hat{\mu}_k)^T \hat{\Sigma}_k^{-1} (x - \hat{\mu}_k), \end{aligned}$$

the *discriminant* of class k

- $\hat{\delta}_k(x) - \hat{\delta}_l(x)$ is a polynomial in x of degree ≤ 2



Plan for today

- Recap: Quadratic discriminant analysis 4 min
- **Linear discriminant analysis** 12 min
- Fisher's discriminant and PCA 12 min
- Too many parameters: naive Bayes 15 min
- Which assumption fails? 6 min

Definition: LDA

Question

What if every class has the **same** covariance $\Sigma_k = \Sigma$, i.e., all classes have the same shape?

Definition: LDA

Question

What if every class has the **same** covariance $\Sigma_k = \Sigma$, i.e., all classes have the same shape?

Definition (Linear discriminant analysis)

QDA with one shared covariance $\hat{\Sigma}_k = \hat{\Sigma}$, estimated by pooling the classes:

$$\hat{\Sigma} = \frac{1}{n} \sum_k \sum_{y_i=k} (x_i - \hat{\mu}_k)(x_i - \hat{\mu}_k)^T.$$

Definition: LDA

Question

What if every class has the **same** covariance $\Sigma_k = \Sigma$, i.e., all classes have the same shape?

Definition (Linear discriminant analysis)

QDA with one shared covariance $\hat{\Sigma}_k = \hat{\Sigma}$, estimated by pooling the classes:

$$\hat{\Sigma} = \frac{1}{n} \sum_k \sum_{y_i=k} (x_i - \hat{\mu}_k)(x_i - \hat{\mu}_k)^T.$$

Idea: The new discriminants $\hat{\delta}_k(x)$ are *linear* in x , instead of quadratic.

However: LDA's probabilities are **not** linear

The simplification of $\hat{\Sigma}_k = \hat{\Sigma}$ makes the discriminant of class k equal to

$$\hat{\delta}_k(x) = \log \hat{\pi}_k - \frac{1}{2} \hat{\mu}_k^T \hat{\Sigma}^{-1} \hat{\mu}_k + x^T \hat{\Sigma}^{-1} \hat{\mu}_k.$$

- linear in x , so the boundaries $\{x : \hat{\delta}_k(x) = \hat{\delta}_l(x)\}$ are hyperplanes

However: LDA's probabilities are **not** linear

The simplification of $\hat{\Sigma}_k = \hat{\Sigma}$ makes the discriminant of class k equal to

$$\hat{\delta}_k(x) = \log \hat{\pi}_k - \frac{1}{2} \hat{\mu}_k^T \hat{\Sigma}^{-1} \hat{\mu}_k + x^T \hat{\Sigma}^{-1} \hat{\mu}_k.$$

- linear in x , so the boundaries $\{x : \hat{\delta}_k(x) = \hat{\delta}_l(x)\}$ are hyperplanes
- by definition, the probabilities $\hat{\eta}_k(x)$ are the **softmax** of the $\hat{\delta}_k(x)$:

$$\hat{\eta}_k(x) = \frac{e^{\hat{\delta}_k(x)}}{\sum_j e^{\hat{\delta}_j(x)}}$$

However: LDA's probabilities are **not** linear

The simplification of $\hat{\Sigma}_k = \hat{\Sigma}$ makes the discriminant of class k equal to

$$\hat{\delta}_k(x) = \log \hat{\pi}_k - \frac{1}{2} \hat{\mu}_k^T \hat{\Sigma}^{-1} \hat{\mu}_k + x^T \hat{\Sigma}^{-1} \hat{\mu}_k.$$

- linear in x , so the boundaries $\{x : \hat{\delta}_k(x) = \hat{\delta}_l(x)\}$ are hyperplanes
- by definition, the probabilities $\hat{\eta}_k(x)$ are the **softmax** of the $\hat{\delta}_k(x)$:

$$\hat{\eta}_k(x) = \frac{e^{\hat{\delta}_k(x)}}{\sum_j e^{\hat{\delta}_j(x)}}$$

However: LDA's probabilities are **not** linear

The simplification of $\hat{\Sigma}_k = \hat{\Sigma}$ makes the discriminant of class k equal to

$$\hat{\delta}_k(x) = \log \hat{\pi}_k - \frac{1}{2} \hat{\mu}_k^T \hat{\Sigma}^{-1} \hat{\mu}_k + x^T \hat{\Sigma}^{-1} \hat{\mu}_k.$$

- linear in x , so the boundaries $\{x : \hat{\delta}_k(x) = \hat{\delta}_l(x)\}$ are hyperplanes
- by definition, the probabilities $\hat{\eta}_k(x)$ are the **softmax** of the $\hat{\delta}_k(x)$:

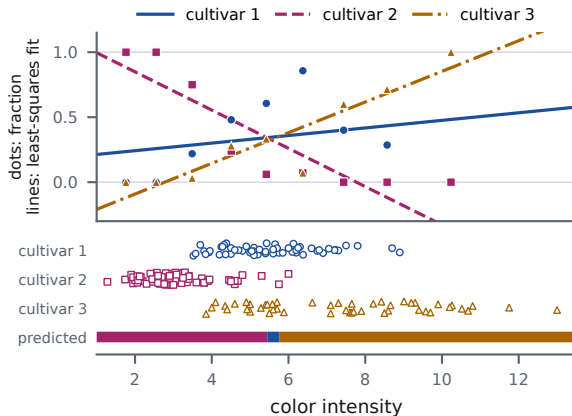
$$\hat{\eta}_k(x) = \frac{e^{\hat{\delta}_k(x)}}{\sum_j e^{\hat{\delta}_j(x)}}$$

Idea: LDA's decision boundaries are linear, but its estimates of the class probabilities are not! Contrast with linear regression on indicators.

Recall: Masking

Example. Wine cultivar from color intensity x alone.

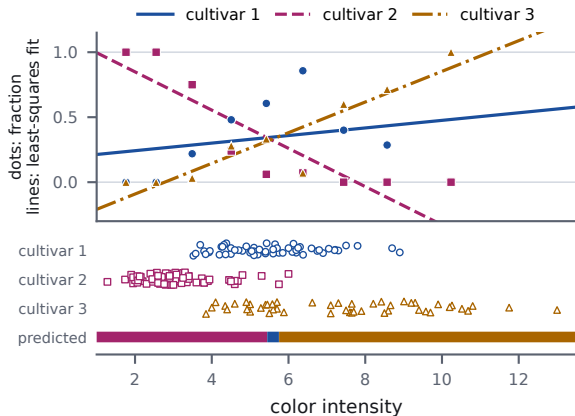
- Least squares on each cultivar's indicator



Recall: Masking

Example. Wine cultivar from color intensity x alone.

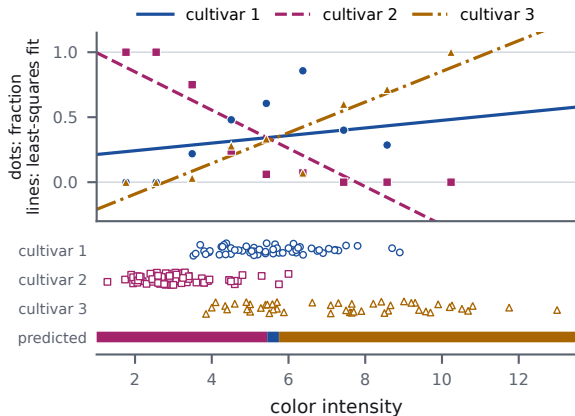
- Least squares on each cultivar's indicator
- Predict the most probable cultivar



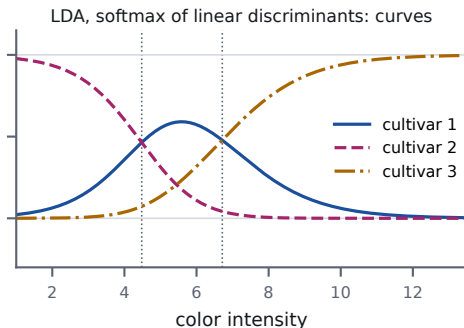
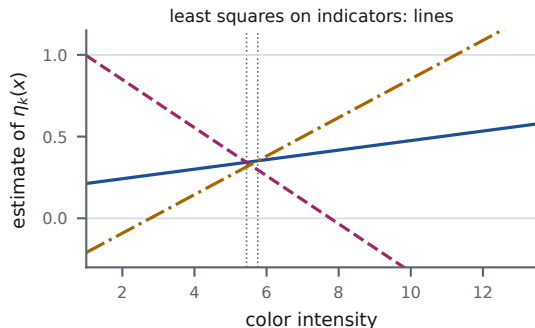
Recall: Masking

Example. Wine cultivar from color intensity x alone.

- Least squares on each cultivar's indicator
- Predict the most probable cultivar
- Only 7/59 cultivar 1 wines classified correctly

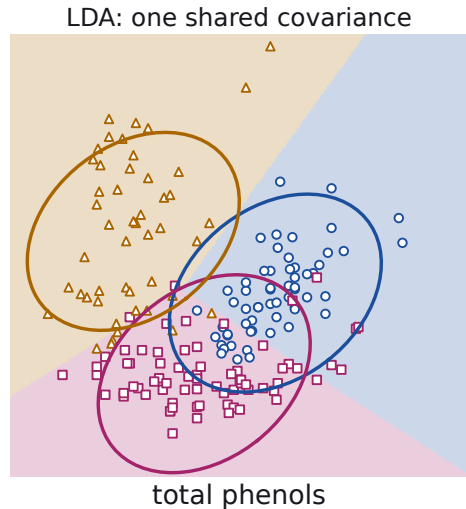
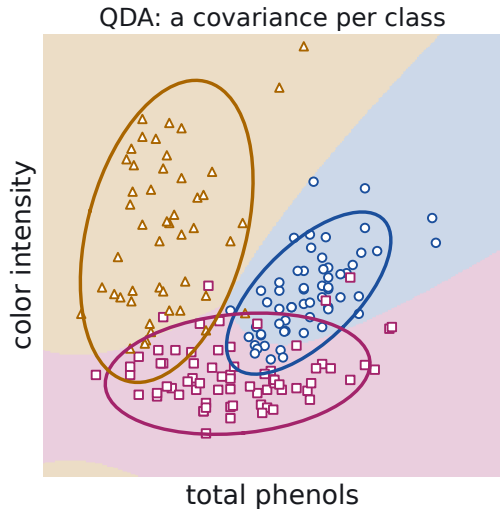


Example: Two estimates of η_k



- 35 of 59 cultivar 1 wines now correctly classified

Example: QDA and LDA on the wine data



Aside: LDA is a **nearest-centroid** rule

Definition (Mahalanobis distance)

$$d_{\Sigma}(x, \mu)^2 = (x - \mu)^{\top} \Sigma^{-1} (x - \mu)$$

Note: It is the exponent of the Gaussian density: $\mathcal{N}(x \mid \mu, \Sigma) = (2\pi)^{-p/2} (\det \Sigma)^{-1/2} e^{-\frac{1}{2} d_{\Sigma}(x, \mu)^2}$.

- so LDA's discriminant is $\delta_k(x) = \log \pi_k - \frac{1}{2} d_{\Sigma}(x, \mu_k)^2$ plus a term that is the same for every class

Aside: LDA is a **nearest-centroid** rule

Definition (Mahalanobis distance)

$$d_{\Sigma}(x, \mu)^2 = (x - \mu)^{\top} \Sigma^{-1} (x - \mu)$$

Note: It is the exponent of the Gaussian density: $\mathcal{N}(x \mid \mu, \Sigma) = (2\pi)^{-p/2} (\det \Sigma)^{-1/2} e^{-\frac{1}{2} d_{\Sigma}(x, \mu)^2}$.

- so LDA's discriminant is $\delta_k(x) = \log \pi_k - \frac{1}{2} d_{\Sigma}(x, \mu_k)^2$ plus a term that is the same for every class
- LDA predicts the class whose mean is nearest in Mahalanobis distance; a direction in which the classes vary little counts for more

Aside: LDA is a **nearest-centroid** rule

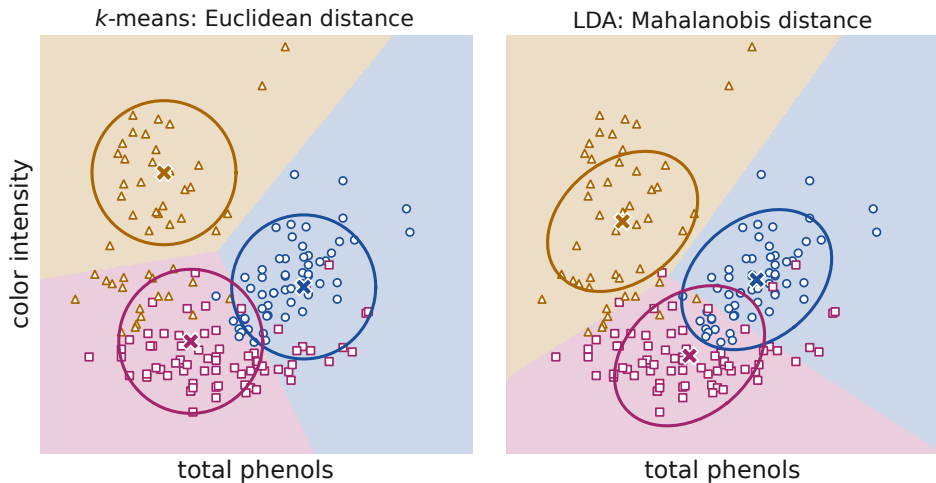
Definition (Mahalanobis distance)

$$d_{\Sigma}(x, \mu)^2 = (x - \mu)^{\top} \Sigma^{-1} (x - \mu)$$

Note: It is the exponent of the Gaussian density: $\mathcal{N}(x \mid \mu, \Sigma) = (2\pi)^{-p/2} (\det \Sigma)^{-1/2} e^{-\frac{1}{2} d_{\Sigma}(x, \mu)^2}$.

- so LDA's discriminant is $\delta_k(x) = \log \pi_k - \frac{1}{2} d_{\Sigma}(x, \mu_k)^2$ plus a term that is the same for every class
- LDA predicts the class whose mean is nearest in Mahalanobis distance; **a direction in which the classes vary little counts for more**
- k -means also assigns to the nearest centroid, but in Euclidean distance, with centroids found **without labels**

Aside: Two nearest-centroid rules



Python: QDA and LDA

Recall: Fitting = one M step of GMM with $r_{ik} = \mathbf{1}\{y_i = k\}$. Predicting evaluates the E-step formula at a new x .

```
from sklearn.discriminant_analysis import (  
    LinearDiscriminantAnalysis as LDA,  
    QuadraticDiscriminantAnalysis as QDA)  
qda = QDA().fit(X_train, y_train)  
lda = LDA().fit(X_train, y_train)  
eta_hat = lda.predict_proba(X_test)  # E-step formula  
y_hat = lda.predict(X_test)  # argmax: equal costs  
qda2 = QDA(reg_param=0.1)  # shrink each Sigma_k
```

Plan for today

- Recap: Quadratic discriminant analysis 4 min
- Linear discriminant analysis 12 min
- Fisher's discriminant and PCA 12 min
- Too many parameters: naive Bayes 15 min
- Which assumption fails? 6 min

Two classes: **subtract** the discriminants

Recall: $\hat{\delta}_k(x) = x^\top \hat{\Sigma}^{-1} \hat{\mu}_k - \frac{1}{2} \hat{\mu}_k^\top \hat{\Sigma}^{-1} \hat{\mu}_k + \log \hat{\pi}_k.$

Predict class 2 when $\hat{\delta}_2(x) - \hat{\delta}_1(x) > 0$, where

$$\hat{\delta}_2(x) - \hat{\delta}_1(x) = \underbrace{x^\top \hat{\Sigma}^{-1} \hat{\mu}_2 - x^\top \hat{\Sigma}^{-1} \hat{\mu}_1}_{\text{depends on } x} \underbrace{- \frac{1}{2} \hat{\mu}_2^\top \hat{\Sigma}^{-1} \hat{\mu}_2 + \frac{1}{2} \hat{\mu}_1^\top \hat{\Sigma}^{-1} \hat{\mu}_1 + \log \frac{\hat{\pi}_2}{\hat{\pi}_1}}_{\text{a number that we'll call } -c}.$$

Two classes: **subtract** the discriminants

Recall: $\hat{\delta}_k(x) = x^\top \hat{\Sigma}^{-1} \hat{\mu}_k - \frac{1}{2} \hat{\mu}_k^\top \hat{\Sigma}^{-1} \hat{\mu}_k + \log \hat{\pi}_k$.

Predict class 2 when $\hat{\delta}_2(x) - \hat{\delta}_1(x) > 0$, where

$$\hat{\delta}_2(x) - \hat{\delta}_1(x) = \underbrace{x^\top \hat{\Sigma}^{-1} \hat{\mu}_2 - x^\top \hat{\Sigma}^{-1} \hat{\mu}_1}_{\text{depends on } x} \underbrace{- \frac{1}{2} \hat{\mu}_2^\top \hat{\Sigma}^{-1} \hat{\mu}_2 + \frac{1}{2} \hat{\mu}_1^\top \hat{\Sigma}^{-1} \hat{\mu}_1 + \log \frac{\hat{\pi}_2}{\hat{\pi}_1}}_{\text{a number that we'll call } -c}.$$

- collect the terms with x :

$$x^\top \hat{\Sigma}^{-1} \hat{\mu}_2 - x^\top \hat{\Sigma}^{-1} \hat{\mu}_1 = x^\top \hat{\Sigma}^{-1} (\hat{\mu}_2 - \hat{\mu}_1)$$

Two classes: **subtract** the discriminants

Recall: $\hat{\delta}_k(x) = x^\top \hat{\Sigma}^{-1} \hat{\mu}_k - \frac{1}{2} \hat{\mu}_k^\top \hat{\Sigma}^{-1} \hat{\mu}_k + \log \hat{\pi}_k.$

Predict class 2 when $\hat{\delta}_2(x) - \hat{\delta}_1(x) > 0$, where

$$\hat{\delta}_2(x) - \hat{\delta}_1(x) = \underbrace{x^\top \hat{\Sigma}^{-1} \hat{\mu}_2 - x^\top \hat{\Sigma}^{-1} \hat{\mu}_1}_{\text{depends on } x} \underbrace{- \frac{1}{2} \hat{\mu}_2^\top \hat{\Sigma}^{-1} \hat{\mu}_2 + \frac{1}{2} \hat{\mu}_1^\top \hat{\Sigma}^{-1} \hat{\mu}_1 + \log \frac{\hat{\pi}_2}{\hat{\pi}_1}}_{\text{a number that we'll call } -c}.$$

- collect the terms with x :

$$x^\top \hat{\Sigma}^{-1} \hat{\mu}_2 - x^\top \hat{\Sigma}^{-1} \hat{\mu}_1 = x^\top \hat{\Sigma}^{-1} (\hat{\mu}_2 - \hat{\mu}_1)$$

- $\hat{\Sigma}^{-1}$ is symmetric, so this is

$$(\hat{\Sigma}^{-1} (\hat{\mu}_2 - \hat{\mu}_1))^\top x = \hat{w}^\top x \quad \text{where} \quad \hat{w} = \hat{\Sigma}^{-1} (\hat{\mu}_2 - \hat{\mu}_1).$$

LDA with two classes: **project**, then cutoff

- from last slide: $\hat{\delta}_2(x) - \hat{\delta}_1(x) = \hat{w}^\top x - c$

LDA with two classes: **project**, then cutoff

- from last slide: $\hat{\delta}_2(x) - \hat{\delta}_1(x) = \hat{w}^\top x - c$
- predict class 2 $\iff \hat{w}^\top x > c$ where $\hat{w} = \hat{\Sigma}^{-1}(\hat{\mu}_2 - \hat{\mu}_1)$

LDA with two classes: **project**, then cutoff

- from last slide: $\hat{\delta}_2(x) - \hat{\delta}_1(x) = \hat{w}^\top x - c$
- predict class 2 $\iff \hat{w}^\top x > c$ where $\hat{w} = \hat{\Sigma}^{-1}(\hat{\mu}_2 - \hat{\mu}_1)$
- **project**: LDA reduces $x \in \mathbb{R}^p$ to one number, $\hat{w}^\top x$, its coordinate along $\hat{w}$

LDA with two classes: **project**, then cutoff

- from last slide: $\hat{\delta}_2(x) - \hat{\delta}_1(x) = \hat{w}^\top x - c$
- predict class 2 $\iff \hat{w}^\top x > c$ where $\hat{w} = \hat{\Sigma}^{-1}(\hat{\mu}_2 - \hat{\mu}_1)$
- **project**: LDA reduces $x \in \mathbb{R}^p$ to one number, $\hat{w}^\top x$, its coordinate along $\hat{w}$
- **cutoff**: at c , which is the projected midpoint of the means

LDA with two classes: **project**, then cutoff

- from last slide: $\hat{\delta}_2(x) - \hat{\delta}_1(x) = \hat{w}^\top x - c$
- predict class 2 $\iff \hat{w}^\top x > c$ where $\hat{w} = \hat{\Sigma}^{-1}(\hat{\mu}_2 - \hat{\mu}_1)$
- **project**: LDA reduces $x \in \mathbb{R}^p$ to one number, $\hat{w}^\top x$, its coordinate along $\hat{w}$
- **cutoff**: at c , which is the projected midpoint of the means
- the boundary $\{x : \hat{w}^\top x = c\}$ is a hyperplane perpendicular to $\hat{w}$, and $\hat{\eta}_2(x) = 1/(1 + e^{-(\hat{w}^\top x - c)})$ depends on x only through $\hat{w}^\top x$

LDA with two classes: **project**, then cutoff

- from last slide: $\hat{\delta}_2(x) - \hat{\delta}_1(x) = \hat{w}^\top x - c$
- predict class 2 $\iff \hat{w}^\top x > c$ where $\hat{w} = \hat{\Sigma}^{-1}(\hat{\mu}_2 - \hat{\mu}_1)$
- **project**: LDA reduces $x \in \mathbb{R}^p$ to one number, $\hat{w}^\top x$, its coordinate along $\hat{w}$
- **cutoff**: at c , which is the projected midpoint of the means
- the boundary $\{x : \hat{w}^\top x = c\}$ is a hyperplane perpendicular to $\hat{w}$, and $\hat{\eta}_2(x) = 1/(1 + e^{-(\hat{w}^\top x - c)})$ depends on x only through $\hat{w}^\top x$

LDA with two classes: **project**, then cutoff

- from last slide: $\hat{\delta}_2(x) - \hat{\delta}_1(x) = \hat{w}^\top x - c$
- predict class 2 $\iff \hat{w}^\top x > c$ where $\hat{w} = \hat{\Sigma}^{-1}(\hat{\mu}_2 - \hat{\mu}_1)$
- **project**: LDA reduces $x \in \mathbb{R}^p$ to one number, $\hat{w}^\top x$, its coordinate along $\hat{w}$
- **cutoff**: at c , which is the projected midpoint of the means
- the boundary $\{x : \hat{w}^\top x = c\}$ is a hyperplane perpendicular to $\hat{w}$, and $\hat{\eta}_2(x) = 1/(1 + e^{-(\hat{w}^\top x - c)})$ depends on x only through $\hat{w}^\top x$

Question

PCA projects on the direction of largest variance. Which direction separates two classes best?

Recall: PCA

Recall: For centered data $X \in \mathbb{R}^{n \times p}$, PCA's first direction maximizes

$$\frac{w^\top X^\top X w}{w^\top w}, \quad w^\top X^\top X w = \sum_{i=1}^n (w^\top (x_i - \bar{x}))^2 :$$

the top eigenvector of $X^\top X$.

- $w^\top X^\top X w$ is the variation of the data along w

Recall: PCA

Recall: For centered data $X \in \mathbb{R}^{n \times p}$, PCA's first direction maximizes

$$\frac{w^\top X^\top X w}{w^\top w}, \quad w^\top X^\top X w = \sum_{i=1}^n (w^\top (x_i - \bar{x}))^2 :$$

the top eigenvector of $X^\top X$.

- $w^\top X^\top X w$ is the variation of the data along w
- PCA never looks at the labels

Recall: PCA

Recall: For centered data $X \in \mathbb{R}^{n \times p}$, PCA's first direction maximizes

$$\frac{w^\top X^\top X w}{w^\top w}, \quad w^\top X^\top X w = \sum_{i=1}^n (w^\top (x_i - \bar{x}))^2 :$$

the top eigenvector of $X^\top X$.

- $w^\top X^\top X w$ is the variation of the data along w
- PCA never looks at the labels

Recall: PCA

Recall: For centered data $X \in \mathbb{R}^{n \times p}$, PCA's first direction maximizes

$$\frac{w^\top X^\top X w}{w^\top w}, \quad w^\top X^\top X w = \sum_{i=1}^n (w^\top (x_i - \bar{x}))^2 :$$

the top eigenvector of $X^\top X$.

- $w^\top X^\top X w$ is the variation of the data along w
- PCA never looks at the labels

Watch out: The direction of largest variation need not separate the classes well.

New: with labels, the variation **splits** in two

Write each centered example as its deviation from its class mean, plus the deviation of its class mean:

$$x_i - \bar{x} = (x_i - \hat{\mu}_{y_i}) + (\hat{\mu}_{y_i} - \bar{x}).$$

It turns out that

$$X^T X = \underbrace{\sum_k \sum_{y_i=k} (x_i - \hat{\mu}_k)(x_i - \hat{\mu}_k)^T}_{W: \text{ within classes}} + \underbrace{\sum_k n_k (\hat{\mu}_k - \bar{x})(\hat{\mu}_k - \bar{x})^T}_{B: \text{ between classes}}.$$

New: with labels, the variation **splits** in two

Write each centered example as its deviation from its class mean, plus the deviation of its class mean:

$$x_i - \bar{x} = (x_i - \hat{\mu}_{y_i}) + (\hat{\mu}_{y_i} - \bar{x}).$$

It turns out that

$$X^T X = \underbrace{\sum_k \sum_{y_i=k} (x_i - \hat{\mu}_k)(x_i - \hat{\mu}_k)^T}_{W: \text{ within classes}} + \underbrace{\sum_k n_k (\hat{\mu}_k - \bar{x})(\hat{\mu}_k - \bar{x})^T}_{B: \text{ between classes}}.$$

- along w : $w^T X^T X w = w^T W w + w^T B w$, the spread within the classes plus the spread of the class means

New: with labels, the variation **splits** in two

Write each centered example as its deviation from its class mean, plus the deviation of its class mean:

$$x_i - \bar{x} = (x_i - \hat{\mu}_{y_i}) + (\hat{\mu}_{y_i} - \bar{x}).$$

It turns out that

$$X^T X = \underbrace{\sum_k \sum_{y_i=k} (x_i - \hat{\mu}_k)(x_i - \hat{\mu}_k)^T}_{W: \text{ within classes}} + \underbrace{\sum_k n_k (\hat{\mu}_k - \bar{x})(\hat{\mu}_k - \bar{x})^T}_{B: \text{ between classes}}.$$

- along w : $w^T X^T X w = w^T W w + w^T B w$, the spread within the classes plus the spread of the class means
- To separate the classes, we want $w^T B w$ large **relative to** $w^T W w$!

Definition: Fisher's discriminant

Definition (Fisher, 1936)

Choose the direction w that maximizes the between-class variation relative to the within-class variation:

$$\frac{w^T B w}{w^T W w}.$$

- no Gaussian model: only a question about separation

Definition: Fisher's discriminant

Definition (Fisher, 1936)

Choose the direction w that maximizes the between-class variation relative to the within-class variation:

$$\frac{w^T B w}{w^T W w}.$$

- no Gaussian model: only a question about separation
- PCA is unsupervised dimension reduction; Fisher's is **supervised**: the labels choose the projection

Fisher's answer for two classes

Recall: LDA with two classes compares $\hat{w}^\top x$ with a cutoff, where $\hat{w} = \hat{\Sigma}^{-1}(\hat{\mu}_2 - \hat{\mu}_1)$.

Theorem

For two classes, with W invertible, the ratio is largest at $w \propto W^{-1}(\hat{\mu}_2 - \hat{\mu}_1)$, which is **LDA's direction** $\hat{w}$, since $\hat{\Sigma} = W/n$.

Fisher's answer for two classes

Recall: LDA with two classes compares $\hat{w}^\top x$ with a cutoff, where $\hat{w} = \hat{\Sigma}^{-1}(\hat{\mu}_2 - \hat{\mu}_1)$.

Theorem

For two classes, with W invertible, the ratio is largest at $w \propto W^{-1}(\hat{\mu}_2 - \hat{\mu}_1)$, which is **LDA's direction** $\hat{w}$, since $\hat{\Sigma} = W/n$.

- why: B sees only $\hat{\mu}_2 - \hat{\mu}_1$, so it pulls w toward that difference; W^{-1} discounts directions in which the classes are spread out

Fisher's answer for two classes

Recall: LDA with two classes compares $\hat{w}^\top x$ with a cutoff, where $\hat{w} = \hat{\Sigma}^{-1}(\hat{\mu}_2 - \hat{\mu}_1)$.

Theorem

For two classes, with W invertible, the ratio is largest at $w \propto W^{-1}(\hat{\mu}_2 - \hat{\mu}_1)$, which is **LDA's direction** $\hat{w}$, since $\hat{\Sigma} = W/n$.

- why: B sees only $\hat{\mu}_2 - \hat{\mu}_1$, so it pulls w toward that difference; W^{-1} discounts directions in which the classes are spread out

Fisher's answer for two classes

Recall: LDA with two classes compares $\hat{w}^\top x$ with a cutoff, where $\hat{w} = \hat{\Sigma}^{-1}(\hat{\mu}_2 - \hat{\mu}_1)$.

Theorem

For two classes, with W invertible, the ratio is largest at $w \propto W^{-1}(\hat{\mu}_2 - \hat{\mu}_1)$, which is **LDA's direction** $\hat{w}$, since $\hat{\Sigma} = W/n$.

- why: B sees only $\hat{\mu}_2 - \hat{\mu}_1$, so it pulls w toward that difference; W^{-1} discounts directions in which the classes are spread out

Key idea: Fisher (no model) and LDA (Gaussian classes) project onto the same direction. LDA's model also supplies the cutoff.

Aside: LDA needs only the **Fisher coordinates**

Theorem

If x and x' have the same Fisher coordinates, then LDA gives them the same $\hat{\eta}_k$ for every k , and so the same prediction.

Aside: LDA needs only the Fisher coordinates

Theorem

If x and x' have the same Fisher coordinates, then LDA gives them the same $\hat{\eta}_k$ for every k , and so the same prediction.

- why: LDA predicts the nearest class mean in Mahalanobis distance (with $\log \hat{\pi}_k$); moving x without changing z changes its squared distance to every mean equally

Aside: LDA needs only the Fisher coordinates

Theorem

If x and x' have the same Fisher coordinates, then LDA gives them the same $\hat{\eta}_k$ for every k , and so the same prediction.

- why: LDA predicts the nearest class mean in Mahalanobis distance (with $\log \hat{\pi}_k$); moving x without changing z changes its squared distance to every mean equally
- so LDA on $x \in \mathbb{R}^p$ is LDA on $z \in \mathbb{R}^m$: $13 \rightarrow 2$ numbers for a wine, $784 \rightarrow 9$ for a digit

Aside: LDA needs only the Fisher coordinates

Theorem

If x and x' have the same Fisher coordinates, then LDA gives them the same $\hat{\eta}_k$ for every k , and so the same prediction.

- why: LDA predicts the nearest class mean in Mahalanobis distance (with $\log \hat{\pi}_k$); moving x without changing z changes its squared distance to every mean equally
- so LDA on $x \in \mathbb{R}^p$ is LDA on $z \in \mathbb{R}^m$: $13 \rightarrow 2$ numbers for a wine, $784 \rightarrow 9$ for a digit

Aside: LDA needs only the **Fisher coordinates**

Theorem

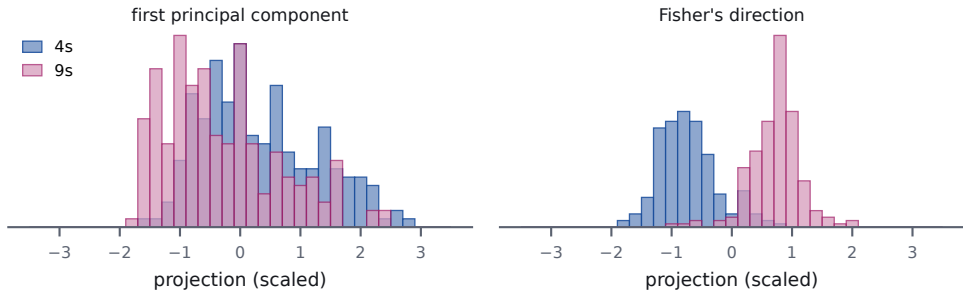
If x and x' have the same Fisher coordinates, then LDA gives them the same $\hat{\eta}_k$ for every k , and so the same prediction.

- why: LDA predicts the nearest class mean in Mahalanobis distance (with $\log \hat{\pi}_k$); moving x without changing z changes its squared distance to every mean equally
- so LDA on $x \in \mathbb{R}^p$ is LDA on $z \in \mathbb{R}^m$: $13 \rightarrow 2$ numbers for a wine, $784 \rightarrow 9$ for a digit

Key idea: With K classes, LDA reduces x to at most $K-1$ numbers, chosen using the labels.

Example: Variance is not separation

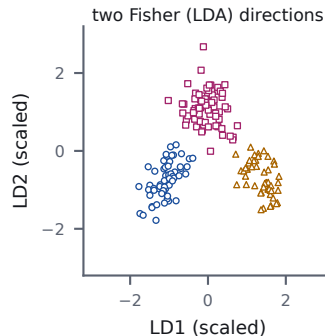
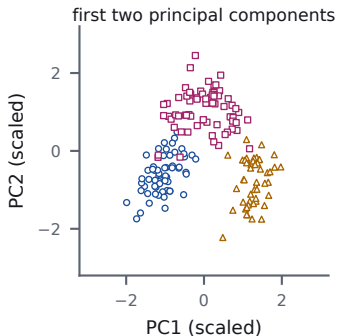
Test images of 4s and 9s, projected on one direction fitted to the training images. W is singular here, so Fisher's direction uses its pseudoinverse.



- LDA on the first PC: 62%; on the first two PCs: 73%; on Fisher's direction: 93%

Example: All 13 wine measurements

Standardized; each coordinate rescaled to unit variance.



- between-class over within-class variance: 4.0 along PC1, 9.1 along the first Fisher direction

Plan for today

- Recap: Quadratic discriminant analysis 4 min
- Linear discriminant analysis 12 min
- Fisher's discriminant and PCA 12 min
- Too many parameters: naïve Bayes 15 min
- Which assumption fails? 6 min

Why count **parameters**?

- QDA estimates $\hat{\pi}_k$, $\hat{\mu}_k$ and $\hat{\Sigma}_k$ for every class, each from that class's examples

Why count **parameters**?

- QDA estimates $\hat{\pi}_k$, $\hat{\mu}_k$ and $\hat{\Sigma}_k$ for every class, each from that class's examples
- more parameters from the same data: lower bias but higher variance, as with polynomial features on the Advertising data

Why count **parameters**?

- QDA estimates $\hat{\pi}_k$, $\hat{\mu}_k$ and $\hat{\Sigma}_k$ for every class, each from that class's examples
- more parameters from the same data: lower bias but higher variance, as with polynomial features on the Advertising data
- $\hat{\Sigma}_k$ must also be inverted, which needs more than p examples in class k

Why count **parameters**?

- QDA estimates $\hat{\pi}_k$, $\hat{\mu}_k$ and $\hat{\Sigma}_k$ for every class, each from that class's examples
- more parameters from the same data: lower bias but higher variance, as with polynomial features on the Advertising data
- $\hat{\Sigma}_k$ must also be inverted, which needs more than p examples in class k

Why count **parameters**?

- QDA estimates $\hat{\pi}_k$, $\hat{\mu}_k$ and $\hat{\Sigma}_k$ for every class, each from that class's examples
- more parameters from the same data: lower bias but higher variance, as with polynomial features on the Advertising data
- $\hat{\Sigma}_k$ must also be inverted, which needs more than p examples in class k

Question

How many numbers does QDA estimate? LDA?

How many parameters?

Recall: For BIC we counted a Gaussian mixture's parameters: $(k - 1) + kp + k \frac{p(p+1)}{2}$. QDA fits one Gaussian per class, so it has the same count, with $k = K$.

Example (Wine data)

$p = 13$ features and $K = 3$ classes. Count the parameters of QDA and LDA.

How many parameters?

Recall: For BIC we counted a Gaussian mixture's parameters: $(k - 1) + kp + k \frac{p(p+1)}{2}$. QDA fits one Gaussian per class, so it has the same count, with $k = K$.

Example (Wine data)

$p = 13$ features and $K = 3$ classes. Count the parameters of QDA and LDA.

	means	covariances	priors	total
QDA	$Kp = 39$	$K \frac{p(p+1)}{2} = 273$	2	314
LDA	39	$\frac{p(p+1)}{2} = 91$	2	132
diagonal Σ_k	39	$Kp = 39$	2	80

When QDA **cannot** be fit

- $\hat{\Sigma}_k$ averages n_k outer products $(x_i - \hat{\mu}_k)(x_i - \hat{\mu}_k)^\top$ whose vectors sum to 0, so $\text{rank } \hat{\Sigma}_k \leq n_k - 1$

When QDA **cannot** be fit

- $\hat{\Sigma}_k$ averages n_k outer products $(x_i - \hat{\mu}_k)(x_i - \hat{\mu}_k)^\top$ whose vectors sum to 0, so $\text{rank } \hat{\Sigma}_k \leq n_k - 1$
- if $n_k \leq p$, $\hat{\Sigma}_k$ is singular: class k 's examples lie in a flat of dimension less than p , and the fitted Gaussian has zero variance across it

When QDA **cannot** be fit

- $\hat{\Sigma}_k$ averages n_k outer products $(x_i - \hat{\mu}_k)(x_i - \hat{\mu}_k)^\top$ whose vectors sum to 0, so $\text{rank } \hat{\Sigma}_k \leq n_k - 1$
- if $n_k \leq p$, $\hat{\Sigma}_k$ is singular: class k 's examples lie in a flat of dimension less than p , and the fitted Gaussian has zero variance across it
- then $\hat{\Sigma}_k^{-1}$ and $\log \det \hat{\Sigma}_k$ do not exist, and neither does $\hat{\delta}_k(x)$

When QDA **cannot** be fit

- $\hat{\Sigma}_k$ averages n_k outer products $(x_i - \hat{\mu}_k)(x_i - \hat{\mu}_k)^\top$ whose vectors sum to 0, so $\text{rank } \hat{\Sigma}_k \leq n_k - 1$
- if $n_k \leq p$, $\hat{\Sigma}_k$ is singular: class k 's examples lie in a flat of dimension less than p , and the fitted Gaussian has zero variance across it
- then $\hat{\Sigma}_k^{-1}$ and $\log \det \hat{\Sigma}_k$ do not exist, and neither does $\hat{\delta}_k(x)$

When QDA **cannot** be fit

- $\hat{\Sigma}_k$ averages n_k outer products $(x_i - \hat{\mu}_k)(x_i - \hat{\mu}_k)^\top$ whose vectors sum to 0, so $\text{rank } \hat{\Sigma}_k \leq n_k - 1$
- if $n_k \leq p$, $\hat{\Sigma}_k$ is singular: class k 's examples lie in a flat of dimension less than p , and the fitted Gaussian has zero variance across it
- then $\hat{\Sigma}_k^{-1}$ and $\log \det \hat{\Sigma}_k$ do not exist, and neither does $\hat{\delta}_k(x)$

Recall: This is related to the issue of variance collapse in MLE for fitting Gaussians.

When QDA **cannot** be fit

- $\hat{\Sigma}_k$ averages n_k outer products $(x_i - \hat{\mu}_k)(x_i - \hat{\mu}_k)^\top$ whose vectors sum to 0, so $\text{rank } \hat{\Sigma}_k \leq n_k - 1$
- if $n_k \leq p$, $\hat{\Sigma}_k$ is singular: class k 's examples lie in a flat of dimension less than p , and the fitted Gaussian has zero variance across it
- then $\hat{\Sigma}_k^{-1}$ and $\log \det \hat{\Sigma}_k$ do not exist, and neither does $\hat{\delta}_k(x)$

Recall: This is related to the issue of variance collapse in MLE for fitting Gaussians.

Key idea: QDA needs at least $p + 1$ examples in *every* class: 14 per cultivar, 785 per digit.

Example: Gaussian classes for digits?

Digits: 10,000 training images, 1,000 of each digit, and $p = 784$ pixels.

Watch out: Each class covariance has 307,720 distinct entries. And although $1,000 > 784$, each is singular: some pixels are constant within a digit, and many are 0 in almost every image of it.

Example: Gaussian classes for digits?

Digits: 10,000 training images, 1,000 of each digit, and $p = 784$ pixels.

Watch out: Each class covariance has 307,720 distinct entries. And although $1,000 > 784$, each is singular: some pixels are constant within a digit, and many are 0 in almost every image of it.

Three ways around it:

- **shrink:** $(1 - \alpha)\hat{\Sigma}_k + \alpha I$ is invertible for every $\alpha > 0$

Example: Gaussian classes for digits?

Digits: 10,000 training images, 1,000 of each digit, and $p = 784$ pixels.

Watch out: Each class covariance has 307,720 distinct entries. And although $1,000 > 784$, each is singular: some pixels are constant within a digit, and many are 0 in almost every image of it.

Three ways around it:

- **shrink:** $(1 - \alpha)\hat{\Sigma}_k + \alpha I$ is invertible for every $\alpha > 0$
- **reduce p first:** on 30 PCs, each Σ_k has 465 entries, estimated from 1,000 images

Example: Gaussian classes for digits?

Digits: 10,000 training images, 1,000 of each digit, and $p = 784$ pixels.

Watch out: Each class covariance has 307,720 distinct entries. And although $1,000 > 784$, each is singular: some pixels are constant within a digit, and many are 0 in almost every image of it.

Three ways around it:

- **shrink:** $(1 - \alpha)\hat{\Sigma}_k + \alpha I$ is invertible for every $\alpha > 0$
- **reduce** p first: on 30 PCs, each Σ_k has 465 entries, estimated from 1,000 images
- **restrict** the covariances: one shared Σ (LDA), or diagonal Σ_k (next)

Example: Gaussian classifiers for digits

On 2,000 test images, k -NN reached 96.7% (all pixels) and 97.5% (50 PCs). Five fixed Gaussian models, test accuracy (exploratory: none was chosen by validation):

QDA on all pixels	cannot be fit
QDA, $\hat{\Sigma}_k$ shrunk toward I	96.7%
LDA on all pixels	89.4%
LDA, $\hat{\Sigma}$ shrunk (Ledoit-Wolf)	90.3%
PCA to 30 dimensions, then QDA	97.6%

- shrinking a covariance is the same idea as the ridge penalty

Example: Gaussian classifiers for digits

On 2,000 test images, k -NN reached 96.7% (all pixels) and 97.5% (50 PCs). Five fixed Gaussian models, test accuracy (exploratory: none was chosen by validation):

QDA on all pixels	cannot be fit
QDA, $\hat{\Sigma}_k$ shrunk toward I	96.7%
LDA on all pixels	89.4%
LDA, $\hat{\Sigma}$ shrunk (Ledoit-Wolf)	90.3%
PCA to 30 dimensions, then QDA	97.6%

- shrinking a covariance is the same idea as the ridge penalty
- the highest test accuracy we have seen, but only 3 images of 2,000 above 3-NN on 50 PCs

Another **special case** of QDA

Recall: LDA is QDA with every Σ_k equal: one covariance instead of K .

Question

What if, instead, every Σ_k is **diagonal**?

Another **special case** of QDA

Recall: LDA is QDA with every Σ_k equal: one covariance instead of K .

Question

What if, instead, every Σ_k is **diagonal**?

- the features are uncorrelated within each class; for Gaussians, independent: $\mathcal{N}(x \mid \mu_k, \Sigma_k) = \prod_j \mathcal{N}(x_j \mid \mu_{kj}, \sigma_{kj}^2)$

Another **special case** of QDA

Recall: LDA is QDA with every Σ_k equal: one covariance instead of K .

Question

What if, instead, every Σ_k is **diagonal**?

- the features are uncorrelated within each class; for Gaussians, independent: $\mathcal{N}(x \mid \mu_k, \Sigma_k) = \prod_j \mathcal{N}(x_j \mid \mu_{kj}, \sigma_{kj}^2)$
- $2Kp + K - 1$ parameters: 80 for wine

Another **special case** of QDA

Recall: LDA is QDA with every Σ_k equal: one covariance instead of K .

Question

What if, instead, every Σ_k is **diagonal**?

- the features are uncorrelated within each class; for Gaussians, independent: $\mathcal{N}(x \mid \mu_k, \Sigma_k) = \prod_j \mathcal{N}(x_j \mid \mu_{kj}, \sigma_{kj}^2)$
- $2Kp + K - 1$ parameters: 80 for wine

Another **special case** of QDA

Recall: LDA is QDA with every Σ_k equal: one covariance instead of K .

Question

What if, instead, every Σ_k is **diagonal**?

- the features are uncorrelated within each class; for Gaussians, independent: $\mathcal{N}(x \mid \mu_k, \Sigma_k) = \prod_j \mathcal{N}(x_j \mid \mu_{kj}, \sigma_{kj}^2)$
- $2Kp + K - 1$ parameters: 80 for wine

Idea: Keep the product, with any one-dimensional model for each factor: **naïve Bayes**.

Definition: Naive Bayes

Definition (Naive Bayes)

Model: priors π_k , and features independent within each class: $f_k(x) = f_{k1}(x_1) \cdots f_{kp}(x_p)$.

Fit: $\hat{\pi}_k = n_k/n$; each $\hat{f}_{kj}$ from the values of feature j in class k alone.

Predict: $\hat{\eta}_k(x) \propto \hat{\pi}_k \hat{f}_{k1}(x_1) \cdots \hat{f}_{kp}(x_p)$.

- each f_{kj} : Gaussian for a measurement; frequencies for a category or yes/no

Definition: Naive Bayes

Definition (Naive Bayes)

Model: priors π_k , and features independent within each class: $f_k(x) = f_{k1}(x_1) \cdots f_{kp}(x_p)$.

Fit: $\hat{\pi}_k = n_k/n$; each $\hat{f}_{kj}$ from the values of feature j in class k alone.

Predict: $\hat{\eta}_k(x) \propto \hat{\pi}_k \hat{f}_{k1}(x_1) \cdots \hat{f}_{kp}(x_p)$.

- each f_{kj} : Gaussian for a measurement; frequencies for a category or yes/no

Definition: Naive Bayes

Definition (Naive Bayes)

Model: priors π_k , and features independent within each class: $f_k(x) = f_{k1}(x_1) \cdots f_{kp}(x_p)$.

Fit: $\hat{\pi}_k = n_k/n$; each $\hat{f}_{kj}$ from the values of feature j in class k alone.

Predict: $\hat{\eta}_k(x) \propto \hat{\pi}_k \hat{f}_{k1}(x_1) \cdots \hat{f}_{kp}(x_p)$.

- each f_{kj} : Gaussian for a measurement; frequencies for a category or yes/no

Key idea: With Gaussian f_{kj} , naive Bayes is QDA with every Σ_k diagonal; with other f_{kj} it goes beyond Gaussian classes.

Naive Bayes versus QDA

QDA

- models the correlations within each class
- $Kp + K\frac{p(p+1)}{2} + K - 1$ parameters: 314 for wine
- needs at least $p + 1$ examples in every class
- Gaussian features

naive Bayes

- ignores the correlations within each class
- $2Kp + K - 1$ parameters: 80 for wine
- needs 2 examples in every class
- any one-dimensional model per feature

Naive Bayes versus QDA

QDA

- models the correlations within each class
- $Kp + K\frac{p(p+1)}{2} + K - 1$ parameters: 314 for wine
- needs at least $p + 1$ examples in every class
- Gaussian features

naive Bayes

- ignores the correlations within each class
- $2Kp + K - 1$ parameters: 80 for wine
- needs 2 examples in every class
- any one-dimensional model per feature

Idea: LDA (shared Σ) and naive Bayes (diagonal Σ_k) restrict QDA in different ways; neither contains the other. Which restriction helps depends on the data.

Example: Fewer training wines

All 13 measurements; test accuracy averaged over 20 random splits:

training wines (per cultivar)	QDA	LDA	naive Bayes
90 (30, 36, 24)	96.6%	97.7%	97.2%
45 (15, 18, 12)	cannot be fit	95.8%	95.5%
27 (9, 11, 7)	cannot be fit	91.1%	92.9%

- 12 wines of cultivar 3, 13 measurements: $\hat{\Sigma}_3$ has rank at most 11

Example: Fewer training wines

All 13 measurements; test accuracy averaged over 20 random splits:

training wines (per cultivar)	QDA	LDA	naive Bayes
90 (30, 36, 24)	96.6%	97.7%	97.2%
45 (15, 18, 12)	cannot be fit	95.8%	95.5%
27 (9, 11, 7)	cannot be fit	91.1%	92.9%

- 12 wines of cultivar 3, 13 measurements: $\hat{\Sigma}_3$ has rank at most 11
- here, with few examples per class, the restricted models do better

Plan for today

- Recap: Quadratic discriminant analysis 4 min
- Linear discriminant analysis 12 min
- Fisher's discriminant and PCA 12 min
- Too many parameters: naive Bayes 15 min
- Which assumption fails? 6 min

Example: Wine, 20 random splits

Two features; 90 training and 88 test wines in each split.

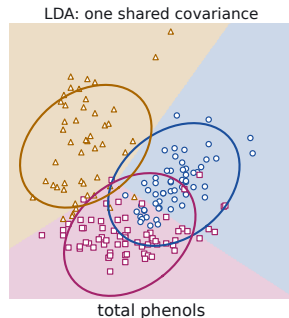
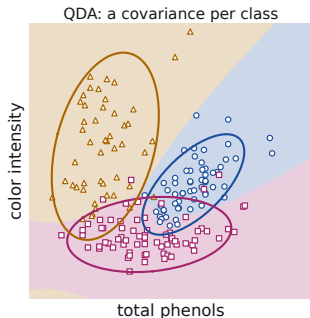
	mean correct (of 88)	range
QDA	78.9	76–82
naïve Bayes	76.1	71–82
LDA	73.5	66–80

- QDA beats LDA on all 20 splits. Why?

The shared-covariance assumption **fails**

	sd(color)	corr.
cultivar 1	1.24	0.65
cultivar 2	0.92	0.17
cultivar 3	2.31	0.34

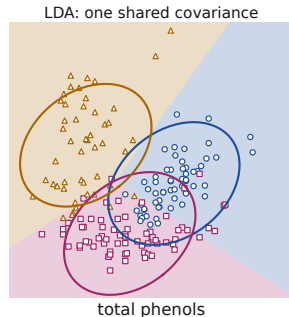
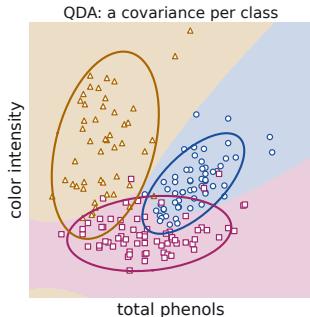
- one ellipse cannot fit all three



The shared-covariance assumption **fails**

	sd(color)	corr.
cultivar 1	1.24	0.65
cultivar 2	0.92	0.17
cultivar 3	2.31	0.34

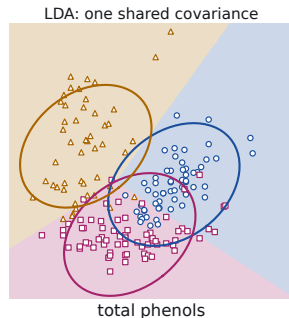
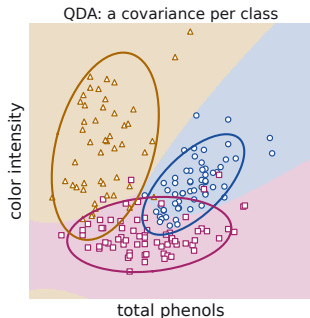
- one ellipse cannot fit all three
- QDA drops the assumption



The shared-covariance assumption **fails**

	sd(color)	corr.
cultivar 1	1.24	0.65
cultivar 2	0.92	0.17
cultivar 3	2.31	0.34

- one ellipse cannot fit all three
- QDA drops the assumption
- naive Bayes: a variance per class, no correlation



Summary

- a generative classifier models each class's x and applies Bayes' rule; with Gaussian classes it is the mixture model with known labels
- QDA: a covariance per class; LDA: one shared covariance, so linear boundaries and nearest centroid in Mahalanobis distance
- LDA's discriminants are linear, so its boundaries are; its probabilities, a softmax, are not, and on wine they reduced masking
- LDA projects on the direction that best separates the classes (Fisher), not the one of largest variance
- Gaussian naive Bayes is QDA with diagonal covariances: a different restriction from LDA's, with fewer parameters