

Lecture 15

Generative classifiers

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

- | | |
|----------------------------------|--------|
| • The mixture model, with labels | 8 min |
| • QDA and LDA | 10 min |
| • Back to the indicator fits | 5 min |
| • Fisher's discriminant and PCA | 10 min |
| • Parameters and naive Bayes | 11 min |
| • Which assumption fails? | 6 min |

Today's question

Problem

We want $\eta_k(x) = \mathbb{P}(Y = k \mid X = x)$. What if it is easier to model how x varies within each class?

Today's question

Problem

We want $\eta_k(x) = \mathbb{P}(Y = k \mid X = x)$. What if it is easier to model how x varies within each class?

Bayes' rule, with $\pi_k = \mathbb{P}(Y = k)$ and f_k the density of x in class k :

$$\eta_k(x) = \frac{\pi_k f_k(x)}{\sum_j \pi_j f_j(x)}.$$

Today's question

Problem

We want $\eta_k(x) = \mathbb{P}(Y = k \mid X = x)$. What if it is easier to model how x varies within each class?

Bayes' rule, with $\pi_k = \mathbb{P}(Y = k)$ and f_k the density of x in class k :

$$\eta_k(x) = \frac{\pi_k f_k(x)}{\sum_j \pi_j f_j(x)}.$$

- today: estimate π_k and f_k from the labeled examples, then use this formula

Today's question

Problem

We want $\eta_k(x) = \mathbb{P}(Y = k \mid X = x)$. What if it is easier to model how x varies within each class?

Bayes' rule, with $\pi_k = \mathbb{P}(Y = k)$ and f_k the density of x in class k :

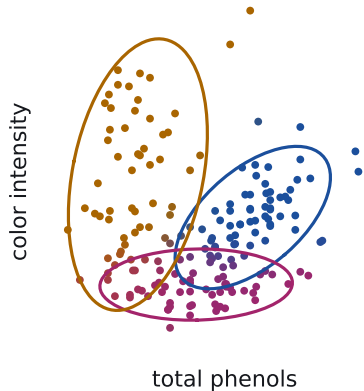
$$\eta_k(x) = \frac{\pi_k f_k(x)}{\sum_j \pi_j f_j(x)}.$$

- today: estimate π_k and f_k from the labeled examples, then use this formula
- the main assumption: each f_k is a Gaussian

Recall: Gaussian mixture model for wine

$$f(x) = \sum_{k=1}^3 \pi_k \mathcal{N}(x \mid \mu_k, \Sigma_k)$$

- fitted by EM, without the cultivars

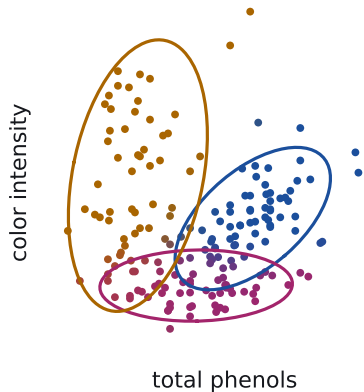


Recall: Gaussian mixture model for wine

$$f(x) = \sum_{k=1}^3 \pi_k \mathcal{N}(x \mid \mu_k, \Sigma_k)$$

- fitted by EM, without the cultivars
- responsibility of component k :

$$r_k(x) = \frac{\pi_k \mathcal{N}(x \mid \mu_k, \Sigma_k)}{\sum_j \pi_j \mathcal{N}(x \mid \mu_j, \Sigma_j)}$$



Recall: Gaussian mixture model for wine

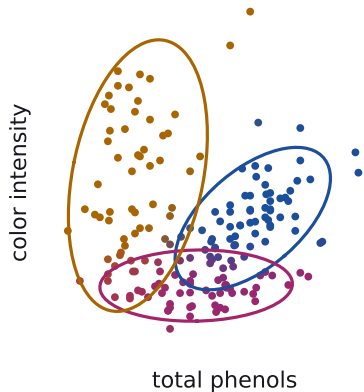
$$f(x) = \sum_{k=1}^3 \pi_k \mathcal{N}(x \mid \mu_k, \Sigma_k)$$

- fitted by EM, without the cultivars
- responsibility of component k :

$$r_k(x) = \frac{\pi_k \mathcal{N}(x \mid \mu_k, \Sigma_k)}{\sum_j \pi_j \mathcal{N}(x \mid \mu_j, \Sigma_j)}$$

- this is just Bayes' rule for

$$\mathbb{P}(Z = k \mid X = x)$$

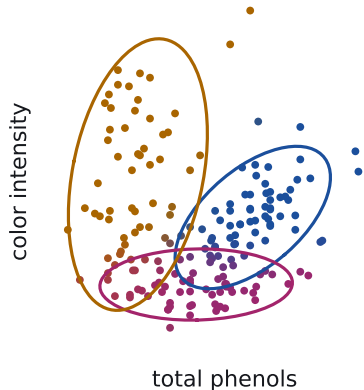


Recall: When the labels are **unknown**

Problem

Fit π_k, μ_k, Σ_k when the component of each point is **unknown**.

- E step: Given params, update r_{ik}
- M step: Given r_{ik} , update params
- Repeat until log-likelihood stops changing



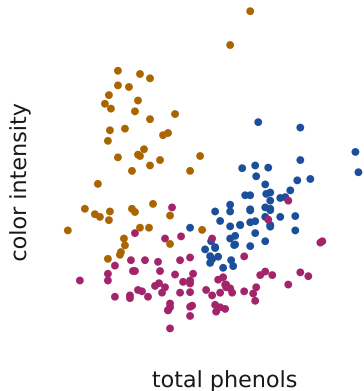
New: When the labels are **known**

Note: The homework 2 problem 1(e) covered this very scenario!

Problem

Fit π_k, μ_k, Σ_k when the component of each point is **known**: y_i .

- No E step, just set $r_{ik} = \mathbf{1}\{y_i = k\}$



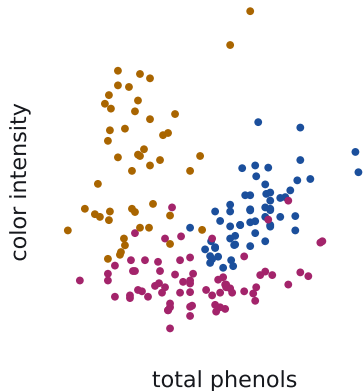
New: When the labels are **known**

Note: The homework 2 problem 1(e) covered this very scenario!

Problem

Fit π_k, μ_k, Σ_k when the component of each point is **known**: y_i .

- No E step, just set $r_{ik} = 1\{y_i = k\}$
- Do one M step to obtain params, then stop!



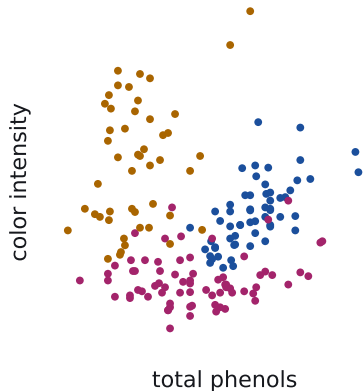
New: When the labels are **known**

Note: The homework 2 problem 1(e) covered this very scenario!

Problem

Fit π_k, μ_k, Σ_k when the component of each point is **known**: y_i .

- No E step, just set $r_{ik} = 1\{y_i = k\}$
- Do one M step to obtain params, then stop!



New: When the labels are **known**

Note: The homework 2 problem 1(e) covered this very scenario!

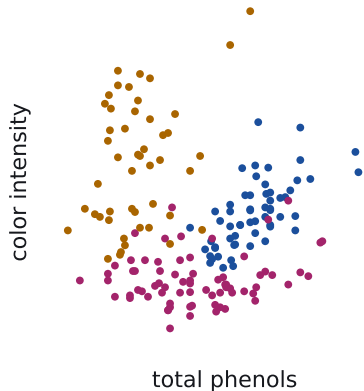
Problem

Fit π_k, μ_k, Σ_k when the component of each point is **known**: y_i .

- No E step, just set $r_{ik} = \mathbf{1}\{y_i = k\}$
- Do one M step to obtain params, then stop!

Check

Write the M step with $r_{ik} = \mathbf{1}\{y_i = k\}$.



Definition: QDA

Definition (Quadratic discriminant analysis)

Fit each class's weight, mean, and covariance to its own examples:

$$\hat{\pi}_k = \frac{n_k}{n}, \quad \hat{\mu}_k = \frac{1}{n_k} \sum_{y_i=k} x_i, \quad \hat{\Sigma}_k = \frac{1}{n_k} \sum_{y_i=k} (x_i - \hat{\mu}_k)(x_i - \hat{\mu}_k)^\top.$$

Then set $\hat{\eta}_k(x) \propto \hat{\pi}_k \mathcal{N}(x \mid \hat{\mu}_k, \hat{\Sigma}_k)$ and (usually) predict $\arg \max_k \hat{\eta}_k(x)$.

Definition: QDA

Definition (Quadratic discriminant analysis)

Fit each class's weight, mean, and covariance to its own examples:

$$\hat{\pi}_k = \frac{n_k}{n}, \quad \hat{\mu}_k = \frac{1}{n_k} \sum_{y_i=k} x_i, \quad \hat{\Sigma}_k = \frac{1}{n_k} \sum_{y_i=k} (x_i - \hat{\mu}_k)(x_i - \hat{\mu}_k)^\top.$$

Then set $\hat{\eta}_k(x) \propto \hat{\pi}_k \mathcal{N}(x \mid \hat{\mu}_k, \hat{\Sigma}_k)$ and (usually) predict $\arg \max_k \hat{\eta}_k(x)$.

Check

What is the point of introducing $\hat{\eta}_k(x)$ if we already know the responsibilities $r_{ik} = \mathbf{1}\{y_i = k\}$?

Plan for today

- | | |
|----------------------------------|--------|
| • The mixture model, with labels | 8 min |
| • QDA and LDA | 10 min |
| • Back to the indicator fits | 5 min |
| • Fisher's discriminant and PCA | 10 min |
| • Parameters and naive Bayes | 11 min |
| • Which assumption fails? | 6 min |

What shape are QDA's decision boundaries?

Note: For the moment, assume we know the params π_k, μ_k, Σ_k .

- Decision boundary consists of x where argmax of $\eta_k(x)$ isn't unique

What shape are QDA's decision boundaries?

Note: For the moment, assume we know the params π_k, μ_k, Σ_k .

- Decision boundary consists of x where argmax of $\eta_k(x)$ isn't unique
- Since $\eta_k(x) \propto \pi_k \mathcal{N}(x \mid \mu_k, \Sigma_k)$, its argmax is same as that of

$$\log(\pi_k \mathcal{N}(x \mid \mu_k, \Sigma_k)) .$$

What shape are QDA's decision boundaries?

Note: For the moment, assume we know the params π_k, μ_k, Σ_k .

- Decision boundary consists of x where argmax of $\eta_k(x)$ isn't unique
- Since $\eta_k(x) \propto \pi_k \mathcal{N}(x \mid \mu_k, \Sigma_k)$, its argmax is same as that of

$$\log(\pi_k \mathcal{N}(x \mid \mu_k, \Sigma_k)).$$

- Ignoring terms that don't depend on class gives the *discriminant*

$$\delta_k(x) = \log \pi_k - \frac{1}{2} \log \det \Sigma_k - \frac{1}{2} (x - \mu_k)^\top \Sigma_k^{-1} (x - \mu_k).$$

What shape are QDA's decision boundaries?

Note: For the moment, assume we know the params π_k, μ_k, Σ_k .

- Decision boundary consists of x where argmax of $\eta_k(x)$ isn't unique
- Since $\eta_k(x) \propto \pi_k \mathcal{N}(x \mid \mu_k, \Sigma_k)$, its argmax is same as that of

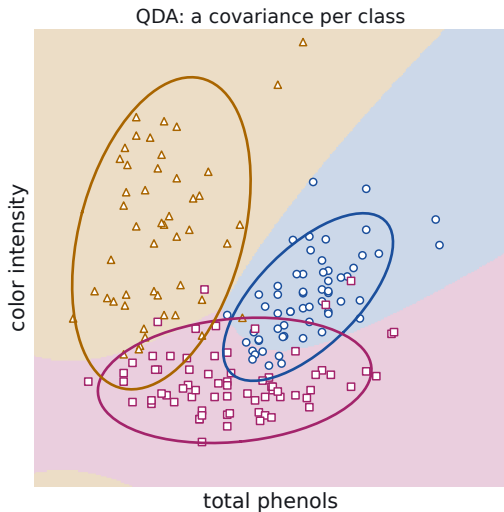
$$\log(\pi_k \mathcal{N}(x \mid \mu_k, \Sigma_k)) .$$

- Ignoring terms that don't depend on class gives the *discriminant*

$$\delta_k(x) = \log \pi_k - \frac{1}{2} \log \det \Sigma_k - \frac{1}{2} (x - \mu_k)^\top \Sigma_k^{-1} (x - \mu_k) .$$

- $\delta_k(x) - \delta_l(x)$ is a polynomial in x of degree ≤ 2 .

Example: QDA on the wine data



Definition: LDA

Question

What if every class has the **same** covariance Σ ?

Definition: LDA

Question

What if every class has the **same** covariance Σ ?

Definition (Linear discriminant analysis)

QDA with one shared covariance, 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 Σ ?

Definition (Linear discriminant analysis)

QDA with one shared covariance, estimated by pooling the classes:

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

With $\Sigma_k = \Sigma$, the terms $x^{\top} \Sigma^{-1} x$ and $\log \det \Sigma$ are the same for every class, so

$$\delta_k(x) - \delta_l(x) = (\mu_k - \mu_l)^{\top} \Sigma^{-1} \mathbf{x} + \text{constant}.$$

What is **linear** in LDA?

LDA's scores, and its probabilities:

$$a_k(x) = x^\top \hat{\Sigma}^{-1} \hat{\mu}_k - \frac{1}{2} \hat{\mu}_k^\top \hat{\Sigma}^{-1} \hat{\mu}_k + \log \hat{\pi}_k, \quad \hat{\eta}_k(x) = \frac{e^{a_k(x)}}{\sum_j e^{a_j(x)}}.$$

- the scores $a_k(x)$ are linear in x , and so is each log-odds $a_k - a_l$

What is **linear** in LDA?

LDA's scores, and its probabilities:

$$a_k(x) = x^\top \hat{\Sigma}^{-1} \hat{\mu}_k - \frac{1}{2} \hat{\mu}_k^\top \hat{\Sigma}^{-1} \hat{\mu}_k + \log \hat{\pi}_k, \quad \hat{\eta}_k(x) = \frac{e^{a_k(x)}}{\sum_j e^{a_j(x)}}.$$

- the scores $a_k(x)$ are linear in x , and so is each log-odds $a_k - a_l$
- the probabilities $\hat{\eta}_k(x)$ are not: a middle class's probability can rise and then fall

What is **linear** in LDA?

LDA's scores, and its probabilities:

$$a_k(x) = x^\top \hat{\Sigma}^{-1} \hat{\mu}_k - \frac{1}{2} \hat{\mu}_k^\top \hat{\Sigma}^{-1} \hat{\mu}_k + \log \hat{\pi}_k, \quad \hat{\eta}_k(x) = \frac{e^{a_k(x)}}{\sum_j e^{a_j(x)}}.$$

- the scores $a_k(x)$ are linear in x , and so is each log-odds $a_k - a_l$
- the probabilities $\hat{\eta}_k(x)$ are not: a middle class's probability can rise and then fall

What is **linear** in LDA?

LDA's scores, and its probabilities:

$$a_k(x) = x^T \hat{\Sigma}^{-1} \hat{\mu}_k - \frac{1}{2} \hat{\mu}_k^T \hat{\Sigma}^{-1} \hat{\mu}_k + \log \hat{\pi}_k, \quad \hat{\eta}_k(x) = \frac{e^{a_k(x)}}{\sum_j e^{a_j(x)}}.$$

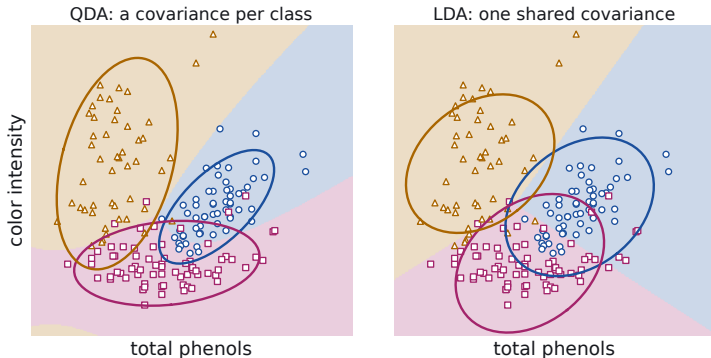
- the scores $a_k(x)$ are linear in x , and so is each log-odds $a_k - a_l$
- the probabilities $\hat{\eta}_k(x)$ are not: a middle class's probability can rise and then fall

Question

Can LDA follow cultivar 1's rise and fall, which no line could?

Example: QDA and LDA on the wine data

Ellipses: the fitted covariances (2 standard deviations). Shading: the predicted cultivar.



LDA is a **nearest-centroid** rule

Definition (Mahalanobis distance)

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

- 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 is a **nearest-centroid** rule

Definition (Mahalanobis distance)

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

- 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
- equal priors: LDA predicts the class whose mean is nearest in Mahalanobis distance; a direction in which the classes vary little counts for more

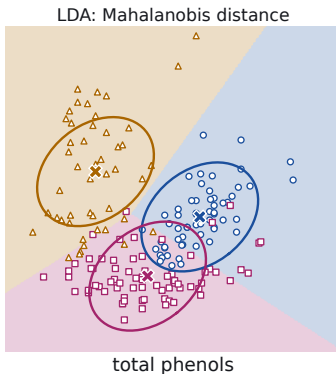
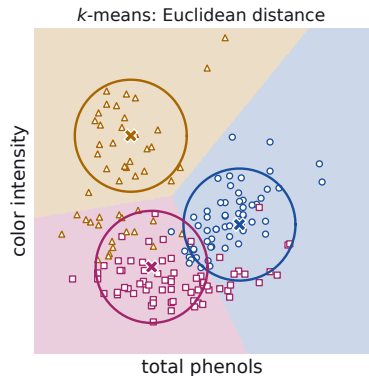
LDA is a **nearest-centroid** rule

Definition (Mahalanobis distance)

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

- 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
- equal priors: 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

Example: Two nearest-centroid rules

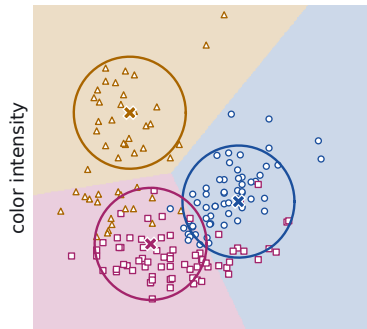


circles: equal
Euclidean distance;
ellipses: equal
Mahalanobis distance

- match with the
cultivars:
k-means 75%, LDA
85%

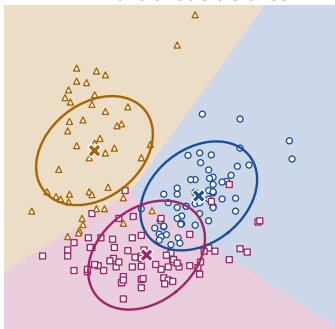
Example: Two nearest-centroid rules

k-means: Euclidean distance



total phenols

LDA: Mahalanobis distance



total phenols

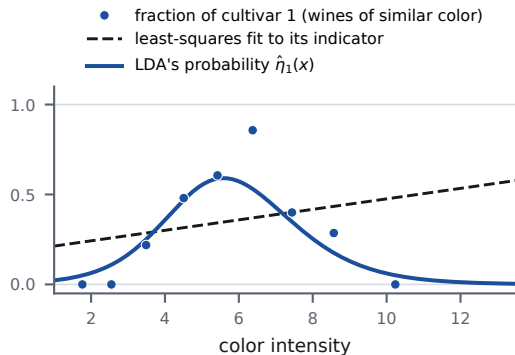
circles: equal
Euclidean distance;
ellipses: equal
Mahalanobis distance

- match with the cultivars:
k-means 75%, LDA 85%
- Euclidean distance to the class means: 83%

Plan for today

- | | |
|----------------------------------|--------|
| • The mixture model, with labels | 8 min |
| • QDA and LDA | 10 min |
| • Back to the indicator fits | 5 min |
| • Fisher's discriminant and PCA | 10 min |
| • Parameters and naive Bayes | 11 min |
| • Which assumption fails? | 6 min |

Example: LDA's probability for cultivar 1



x is color intensity; all 178 wines.

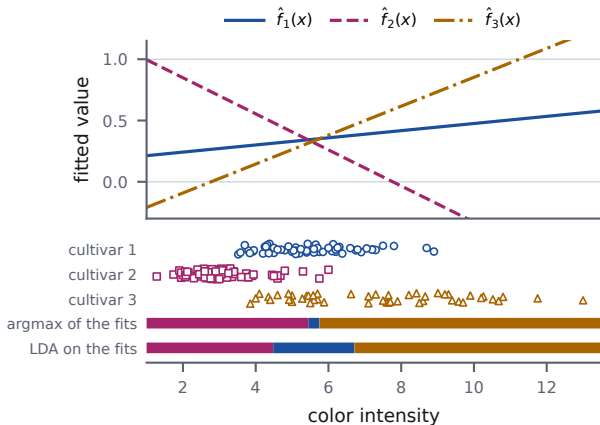
	lines	LDA
correct	59.6%	70.8%
cultivar 1 correct	7 of 59	35 of 59
predicted cultivar 1	14	58
where	[5.4, 5.8]	[4.5, 6.7]

- LDA's probability rises and falls

The **same** fits, a different rule

x is color intensity; the lines are least squares on the 3 indicators.

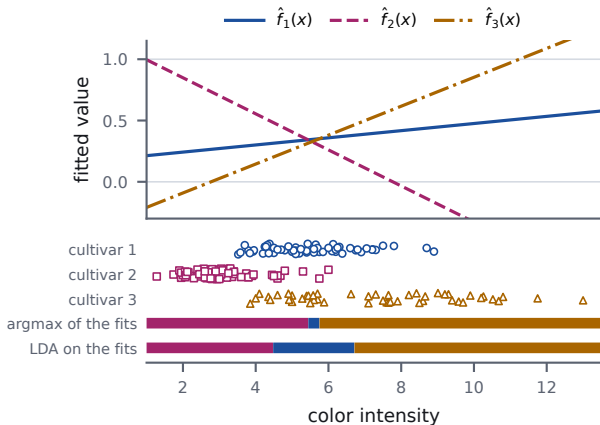
- **argmax**: the largest $\hat{f}_k(x)$; 7 of 59 **cultivar 1** wines



The **same** fits, a different rule

x is color intensity; the lines are least squares on the 3 indicators.

- **argmax**: the largest $\hat{f}_k(x)$; 7 of 59 **cultivar 1** wines
- **LDA on the fits**: input $\hat{y}(x) = (\hat{f}_1(x), \hat{f}_2(x), \hat{f}_3(x))$, nearest class mean of $\hat{y}$; 35 of 59, exactly as LDA on x



The fits keep **LDA's** information

Theorem

Least squares with an intercept, unpenalized, on the K indicators; the pooled covariance of x invertible. Then LDA on the fitted vectors $\hat{y}(x_i)$, in the subspace they span, with the same priors, classifies every x as LDA on x does.

- the $\hat{y}(x_i)$ sum to 1, so their covariance is singular (**why?**)

The fits keep **LDA's** information

Theorem

Least squares with an intercept, unpenalized, on the K indicators; the pooled covariance of x invertible. Then LDA on the fitted vectors $\hat{y}(x_i)$, in the subspace they span, with the same priors, classifies every x as LDA on x does.

- the $\hat{y}(x_i)$ sum to 1, so their covariance is singular (**why?**)
- 13 measurements: LDA on the 3 fits agrees with LDA on all 13, for every wine

The fits keep LDA's information

Theorem

Least squares with an intercept, unpenalized, on the K indicators; the pooled covariance of x invertible. Then LDA on the fitted vectors $\hat{y}(x_i)$, in the subspace they span, with the same priors, classifies every x as LDA on x does.

- the $\hat{y}(x_i)$ sum to 1, so their covariance is singular (why?)
- 13 measurements: LDA on the 3 fits agrees with LDA on all 13, for every wine

The fits keep LDA's information

Theorem

Least squares with an intercept, unpenalized, on the K indicators; the pooled covariance of x invertible. Then LDA on the fitted vectors $\hat{y}(x_i)$, in the subspace they span, with the same priors, classifies every x as LDA on x does.

- the $\hat{y}(x_i)$ sum to 1, so their covariance is singular (why?)
- 13 measurements: LDA on the 3 fits agrees with LDA on all 13, for every wine

Idea: The fits are poor probability estimates but keep LDA's information; the Gaussian model reads it better on this example.

Plan for today

- The mixture model, with labels 8 min
- QDA and LDA 10 min
- Back to the indicator fits 5 min
- Fisher's discriminant and PCA 10 min
- Parameters and naive Bayes 11 min
- Which assumption fails? 6 min

Definition: Fisher's discriminant

Recall: PCA's first direction maximizes $w^T T w / w^T w$, where $T = W + B$ is the total scatter: the top eigenvector of T .

Definition (Fisher, 1936)

Choose w to maximize between-class variance relative to within-class variance:

$$\frac{w^T B w}{w^T W w}, \quad B = \sum_k n_k (\hat{\mu}_k - \bar{x})(\hat{\mu}_k - \bar{x})^T, \quad W = \sum_k \sum_{y_i=k} (x_i - \hat{\mu}_k)(x_i - \hat{\mu}_k)^T.$$

- no Gaussian model needed: only a question about separation

Fisher's direction is LDA's direction

- with W positive definite, the maximizers are top eigenvectors of $W^{-1}B$, whose eigenvalues are those of the symmetric $W^{-1/2}BW^{-1/2}$; at most $\min(p, K - 1)$ are nonzero

Fisher's direction is **LDA's** direction

- with W positive definite, the maximizers are top eigenvectors of $W^{-1}B$, whose eigenvalues are those of the symmetric $W^{-1/2}BW^{-1/2}$; at most $\min(p, K - 1)$ are nonzero
- two classes: $B = \frac{n_1 n_2}{n} (\hat{\mu}_2 - \hat{\mu}_1)(\hat{\mu}_2 - \hat{\mu}_1)^\top$ has rank one, so $w \propto W^{-1}(\hat{\mu}_2 - \hat{\mu}_1)$ (**Check this.**)

Fisher's direction is **LDA's** direction

- with W positive definite, the maximizers are top eigenvectors of $W^{-1}B$, whose eigenvalues are those of the symmetric $W^{-1/2}BW^{-1/2}$; at most $\min(p, K - 1)$ are nonzero
- two classes: $B = \frac{n_1 n_2}{n} (\hat{\mu}_2 - \hat{\mu}_1)(\hat{\mu}_2 - \hat{\mu}_1)^\top$ has rank one, so $w \propto W^{-1}(\hat{\mu}_2 - \hat{\mu}_1)$ (**Check this.**)
- LDA's log-odds is $(\hat{\mu}_2 - \hat{\mu}_1)^\top \hat{\Sigma}^{-1}x + \text{constant}$, and $\hat{\Sigma} = W/n$

Fisher's direction is LDA's direction

- with W positive definite, the maximizers are top eigenvectors of $W^{-1}B$, whose eigenvalues are those of the symmetric $W^{-1/2}BW^{-1/2}$; at most $\min(p, K - 1)$ are nonzero
- two classes: $B = \frac{n_1 n_2}{n} (\hat{\mu}_2 - \hat{\mu}_1)(\hat{\mu}_2 - \hat{\mu}_1)^\top$ has rank one, so $w \propto W^{-1}(\hat{\mu}_2 - \hat{\mu}_1)$ (Check this.)
- LDA's log-odds is $(\hat{\mu}_2 - \hat{\mu}_1)^\top \hat{\Sigma}^{-1}x + \text{constant}$, and $\hat{\Sigma} = W/n$
- so LDA projects x on Fisher's direction and compares with a cutoff

Fisher's direction is LDA's direction

- with W positive definite, the maximizers are top eigenvectors of $W^{-1}B$, whose eigenvalues are those of the symmetric $W^{-1/2}BW^{-1/2}$; at most $\min(p, K - 1)$ are nonzero
- two classes: $B = \frac{n_1 n_2}{n}(\hat{\mu}_2 - \hat{\mu}_1)(\hat{\mu}_2 - \hat{\mu}_1)^\top$ has rank one, so $w \propto W^{-1}(\hat{\mu}_2 - \hat{\mu}_1)$ (Check this.)
- LDA's log-odds is $(\hat{\mu}_2 - \hat{\mu}_1)^\top \hat{\Sigma}^{-1}x + \text{constant}$, and $\hat{\Sigma} = W/n$
- so LDA projects x on Fisher's direction and compares with a cutoff

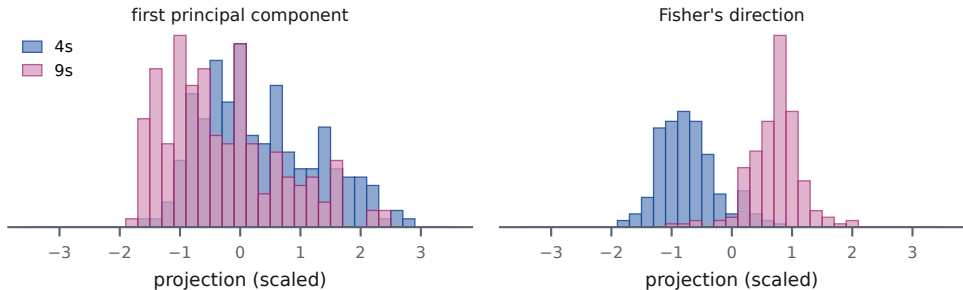
Fisher's direction is LDA's direction

- with W positive definite, the maximizers are top eigenvectors of $W^{-1}B$, whose eigenvalues are those of the symmetric $W^{-1/2}BW^{-1/2}$; at most $\min(p, K - 1)$ are nonzero
- two classes: $B = \frac{n_1 n_2}{n} (\hat{\mu}_2 - \hat{\mu}_1)(\hat{\mu}_2 - \hat{\mu}_1)^\top$ has rank one, so $w \propto W^{-1}(\hat{\mu}_2 - \hat{\mu}_1)$ (Check this.)
- LDA's log-odds is $(\hat{\mu}_2 - \hat{\mu}_1)^\top \hat{\Sigma}^{-1}x + \text{constant}$, and $\hat{\Sigma} = W/n$
- so LDA projects x on Fisher's direction and compares with a cutoff

Key idea: With K classes, LDA's prediction depends on x only through at most $\min(p, K - 1)$ Fisher coordinates: 2 for the three cultivars, 9 for the ten digits.

Example: Variance is not separation

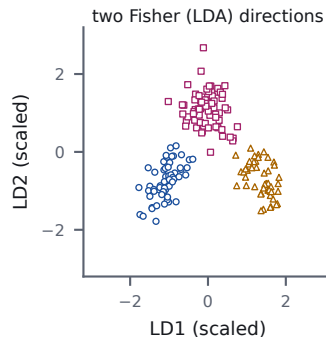
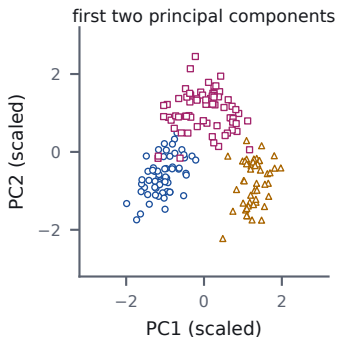
Test images of 4s and 9s, projected on one direction fitted to the training images. W is singular here, so Fisher's direction uses its pseudoinverse.



- 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

- | | |
|----------------------------------|--------|
| • The mixture model, with labels | 8 min |
| • QDA and LDA | 10 min |
| • Back to the indicator fits | 5 min |
| • Fisher's discriminant and PCA | 10 min |
| • Parameters and naïve Bayes | 11 min |
| • Which assumption fails? | 6 min |

How many parameters?

Problem

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

How many parameters?

Problem

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

	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

How many parameters?

Problem

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

	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

Watch out: Digits: each class covariance has 307,720 distinct entries, so estimating it is demanding. Here it is also singular: some pixels are constant within each class.

Example: Gaussian classifiers for digits

Test accuracy of five fixed models (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

Test accuracy of five fixed models (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

Definition: Naive Bayes

Definition (Naive Bayes)

Assume the features are independent given the class:

$$f_k(x) = f_{k1}(x_1) f_{k2}(x_2) \cdots f_{kp}(x_p).$$

Definition: Naive Bayes

Definition (Naive Bayes)

Assume the features are independent given the class:

$$f_k(x) = f_{k1}(x_1) f_{k2}(x_2) \cdots f_{kp}(x_p).$$

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

- each f_{kj} is any one-dimensional model: counts, categories, mixed types

Definition: Naive Bayes

Definition (Naive Bayes)

Assume the features are independent given the class:

$$f_k(x) = f_{k1}(x_1) f_{k2}(x_2) \cdots f_{kp}(x_p).$$

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

- each f_{kj} is any one-dimensional model: counts, categories, mixed types
- digits: 69.3% on raw pixels; 90.4% on the first 30 PCs, which are uncorrelated overall but not independent within each class

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 more than p 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 more than p 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 and 13 measurements: $\hat{\Sigma}_3$ is singular

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 and 13 measurements: $\hat{\Sigma}_3$ is singular
- here, with few examples per class, the restricted models do better

Example: Filtering spam

Problem

Feature: does the email contain “free”? It occurs in 0 of 40 legitimate training emails. With $\hat{p} = 0/40$, what does naive Bayes conclude about a legitimate-looking email that contains “free”?

Example: Filtering spam

Problem

Feature: does the email contain “free”? It occurs in 0 of 40 legitimate training emails. With $\hat{p} = 0/40$, what does naive Bayes conclude about a legitimate-looking email that contains “free”?

- the factor $\hat{p} = 0$ makes $\mathbb{P}(\text{legitimate} \mid x) = 0$, whatever else the email contains

Example: Filtering spam

Problem

Feature: does the email contain “free”? It occurs in 0 of 40 legitimate training emails. With $\hat{p} = 0/40$, what does naive Bayes conclude about a legitimate-looking email that contains “free”?

- the factor $\hat{p} = 0$ makes $\mathbb{P}(\text{legitimate} \mid x) = 0$, whatever else the email contains
- uniform prior $\text{Beta}(1, 1)$ on p : posterior $\text{Beta}(1, 41)$, with mean $\frac{1}{42}$

Example: Filtering spam

Problem

Feature: does the email contain “free”? It occurs in 0 of 40 legitimate training emails. With $\hat{p} = 0/40$, what does naive Bayes conclude about a legitimate-looking email that contains “free”?

- the factor $\hat{p} = 0$ makes $\mathbb{P}(\text{legitimate} \mid x) = 0$, whatever else the email contains
- uniform prior $\text{Beta}(1, 1)$ on p : posterior $\text{Beta}(1, 41)$, with mean $\frac{1}{42}$
- in general $\frac{c+1}{n+2}$: add one to each count (**Laplace smoothing**)

Python: Generative classifiers

```
from sklearn.discriminant_analysis import (  
    LinearDiscriminantAnalysis as LDA,  
    QuadraticDiscriminantAnalysis as QDA)  
from sklearn.naive_bayes import GaussianNB  
lda = LDA().fit(X_train, y_train)  
Z = lda.transform(X_test)  # at most  $K - 1$  coords  
lda2 = LDA(priors=[0.2, 0.3, 0.5])  # set the priors  
qda = QDA(reg_param=0.1).fit(X_train, y_train)  
nb = GaussianNB().fit(X_train, y_train)  
p_hat = qda.predict_proba(X_test)
```

- unregularized LDA and QDA do not change when features are rescaled; with `reg_param`, the scaling and the shrinkage target are part of the model

Plan for today

- | | |
|----------------------------------|--------|
| • The mixture model, with labels | 8 min |
| • QDA and LDA | 10 min |
| • Back to the indicator fits | 5 min |
| • Fisher's discriminant and PCA | 10 min |
| • Parameters and naive Bayes | 11 min |
| • Which assumption fails? | 6 min |

Example: Wine, 20 random splits

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

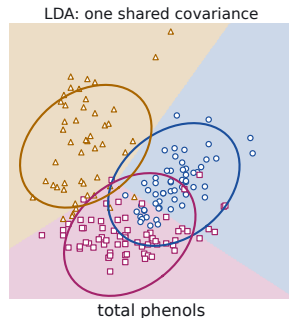
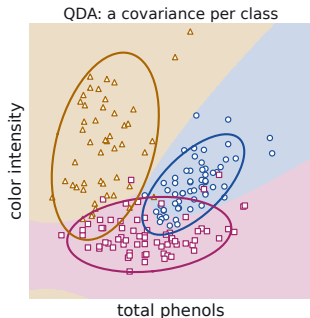
	mean correct (of 88)	range
QDA	78.9	76–82
naïve Bayes	76.1	71–82
LDA	73.5	66–80

- 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

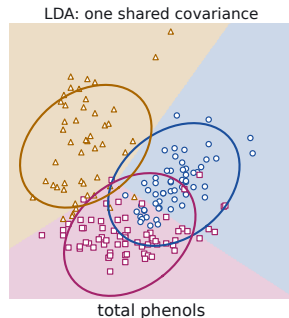
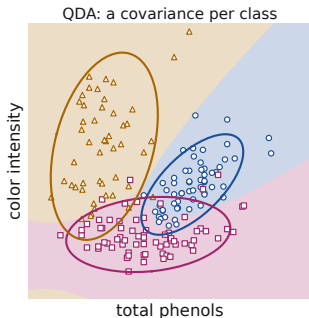
- one ellipse cannot fit all three



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

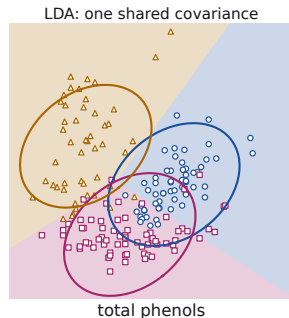
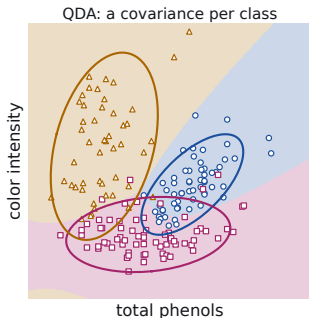
- one ellipse cannot fit all three
- QDA drops the assumption



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 scores are linear but its probabilities are not; the indicator fits are poor probabilities but keep LDA's information
- Fisher's directions maximize separation, not variance; they are LDA's directions, at most $\min(p, K - 1)$ of them
- Gaussian naive Bayes is QDA with diagonal covariances: a different restriction from LDA's, with fewer parameters