EECS 553 Lecture 02
Machine Learning (ECE)
[class version]

Jeff Fessler

University of Michigan
©Laura Balzano and Jeff Fessler, UM, September 2, 2026, 22:57

F26

Logistics

© 2026 Fessler

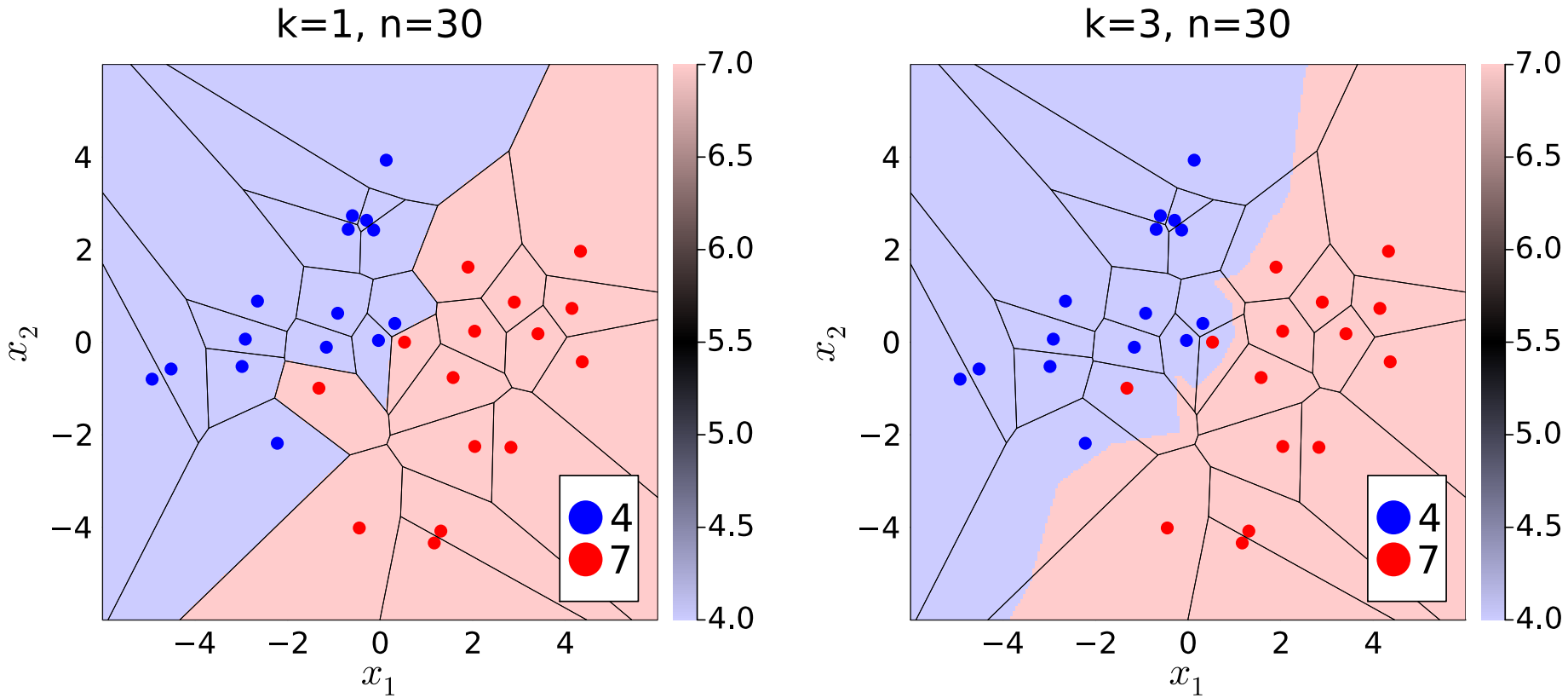
- iClicker starts counting today (drop lowest 4 days)
- HW1 due Wed. Sep. 9, 11PM
- Sit near (at least) one other classmate!

Recap / Outline

© 2026 Fessler

- Recap
  - class intro
  - k-NN (nearest neighbor) classification
- Today
  - k-NN recap
  - Probabilistic methods for classification
    - Loss and risk
    - 0-1 loss and error probability
  - Bayes classifier
  - Plug-in classifiers
    - Linear discriminant analysis (LDA)
    - Generative vs discriminative
    - Parametric vs nonparametric
    - Quadratic discriminant analysis (QDA)

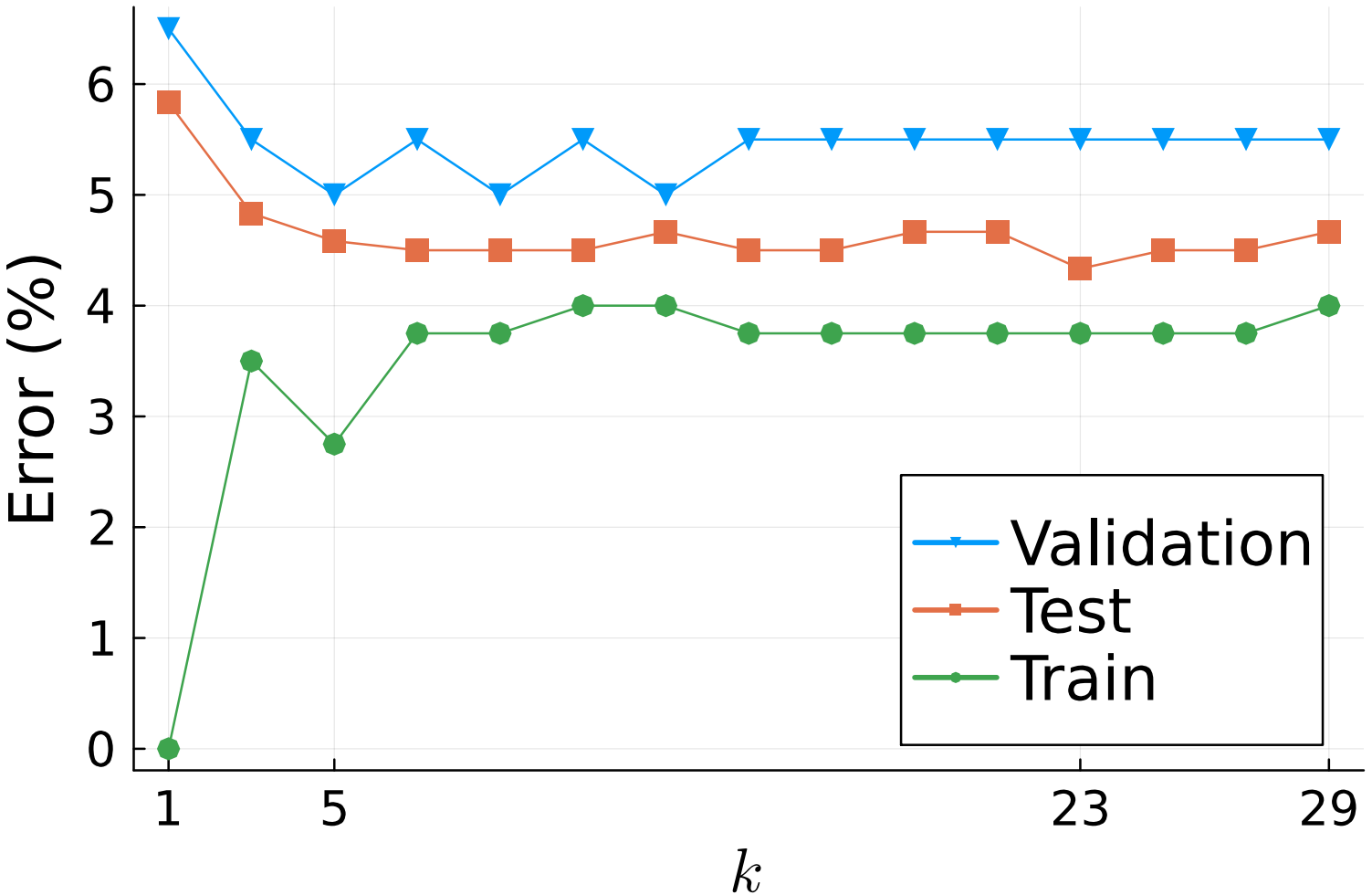
Relating nearest-neighbor (NN) classifier and **Voronoi diagrams**.
Shown here for $n = 30$ for easier visualization.



- k-NN decision boundary unrelated to Voronoi cells for $k > 1$.

k-NN error plot revisited

© 2026 Fessler

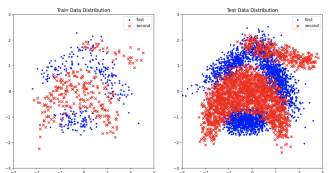


Now for odd k values only!

Goal: work towards designing *optimal* classifiers.
Optimizing requires some modeling assumptions.
ML literature abounds with probabilistic models.

- Feature vector $\boldsymbol{X} \in \mathbb{R}^d$
- Label $Y \in \mathcal{Y}$, *e.g.*, $\mathcal{Y} = \{1, \ldots, K\}$ or $\mathcal{Y} = \{+1, -1\}$
- Assume each pair $(\boldsymbol{X}, Y)$ is a

- Design classifier $f : \mathbb{R}^d \to \{1, \ldots, K\}$ in principled (optimal?) way using $\mathrm{p}(\boldsymbol{x}, y)$.
- Example: colab demo used synthetic data from some $\mathrm{p}(\boldsymbol{x}, y)$.



CQ: Probability

© 2026 Fessler

Q02.1 The three components of a **probability space** are:

A: The sample space, event space, and probability measure.

B: The sampling procedure, sigma algebra, and joint probability distribution.

C: The sample space, a partition of the real line into countably infinite disjoint sets, and the measure of those sets.

D: none of these

Probabilistic framework for classification

© 2026 Fessler

- We wish to learn/model/exploit the joint distribution of $(\boldsymbol{X}, Y)$.
  - Often (but not always) we will use "ECE 501 notation" $(\boldsymbol{X}, Y)$ for random vectors / random variables.
  - Here $\boldsymbol{X} \in \mathbb{R}^d$ is a random vector (not a matrix)
  - Later we often use $\boldsymbol{A}$ and $\boldsymbol{X}$ with bold **serif** fonts for **matrices**.
- Notation:
  - $\pi_k \triangleq \mathsf{P}\{Y = k\}$ class probabilities
  - $g_k(\boldsymbol{x}) \triangleq$ pdf/pmf of $\boldsymbol{X}$ given $Y = k$, *e.g.*, $g_k(\boldsymbol{x}) =$
    Why "g_k"?
- Learning / modeling:
  - $\{\pi_k\}$ rarely known; typically need to be learned
  - $\{g_k(\cdot)\}$ rarely known; typically need to be learned (via some parametric or nonparametric model)

Theorem (**Law of total probability**)

*Consider a probability space with a sample space, event space, and a probability measure. If $\{B_k\}$ is a finite or countably infinite **partition** of the sample space, then for any event A on the same probability space:*

$$P(A) =$$

$$= $$

For binary classification, use **partition** $B_1 = \{Y = 1\}$ and $B_2 = \{Y = 2\}$.
So the (total) distribution (pdf or pmf) $g(\boldsymbol{x})$ is a **mixture distribution**:

$$g(\boldsymbol{x}) =$$

- We wish to learn/model/exploit the joint distribution of $(\boldsymbol{X}, Y)$.
- Notation:
  - $\pi_k \triangleq \mathsf{P}\{Y = k\}$
  - $g_k(\boldsymbol{x}) \triangleq$ pdf/pmf of $\boldsymbol{X}$ given $Y = k$
  - $\eta_k(\boldsymbol{x}) \triangleq$
    Why η?

Q02.2 The notation $\pi_k \triangleq \mathsf{P}\{Y = k\}$ represents the $\boxed{?}$ class probability and the notation $\eta_k(\boldsymbol{x}) \triangleq \mathsf{P}\{Y = k \mid \boldsymbol{X} = \boldsymbol{x}\}$ represents the $\boxed{?}$ class probability.

A: kth, $\boldsymbol{x}$ given k

B: conditional, marginal

C: prior, posterior

D: binary, multi-class

E: none of these

Foreshadowing the Bayes' classifier

© 2026 Fessler

- A **classifier** is a function $f : \mathbb{R}^d \mapsto \{1, \ldots, K\}$.
- Given the joint distribution of $(\boldsymbol{X}, Y)$, what is the "best" possible classifier?
- The best classifier depends on a chosen **performance measure**.
- The most common performance measure for **classification** problems is the **probability of error**, defined by:

$$R(f) \triangleq$$

i.e., the probability of the event

$$\{(\boldsymbol{x}, y) \in \mathbb{R}^d \times \{1, \ldots, K\} \, : \, f(\boldsymbol{x}) \neq y\}.$$

- It treats all errors *equally* (both hard and easy cases).
- Given $\mathrm{p}(\boldsymbol{x}, y)$, a "best" classifier is $f_* \triangleq$ an **optimization problem**.

Loss and risk in general ML problems

© 2026 Fessler

- Consider a supervised learning problem with feature vectors / labels $(\boldsymbol{X}, Y)$ having joint distribution $\mathrm{p}(\boldsymbol{x}, y)$.
- Let $\mathcal{Y}$ denote the output space
  - Classification: $\mathcal{Y} =$
  - Regression: $\mathcal{Y} =$
- A **loss function** is a function $L : \mathcal{Y} \times \mathcal{Y} \mapsto [0, \infty)$:

$$L(y, \hat{y}) =$$

- Let $f : \mathbb{R}^d \mapsto \mathcal{Y}$ denote a **classifier** or **regressor**, aka **predictor**. The (statistical) **risk** of f is defined to be

$$R(f) = R_L(f) \triangleq$$

Loss and risk for binary classification

© 2026 Fessler

- For binary classification problems, often $\mathcal{Y} = \{1, -1\}$, and f is called a **decision function** or **decision rule** or **discriminant function**. The predicted label is

$$\hat{y} = f(\boldsymbol{x}) =$$

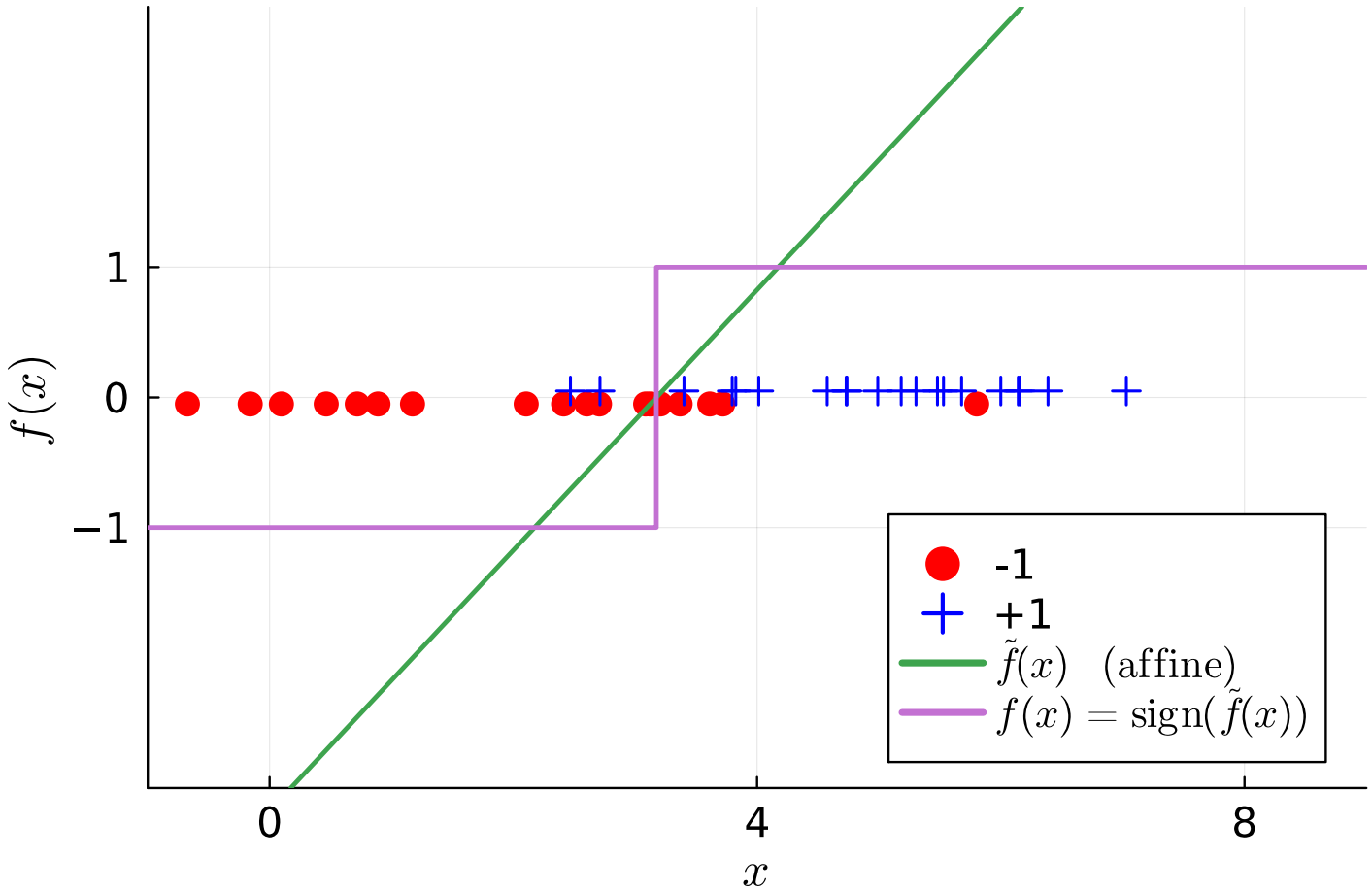
$$\tilde{f} : \mathbb{R}^d \mapsto \mathbb{R}$$
$$f : \mathbb{R}^d \mapsto \{\pm 1\}.$$

Pictures on next pages from decision boundary demo.

- For example, a "linear" classifier has $\tilde{f}(\boldsymbol{x}) =$
  - The $d+1$ classifier parameters are **weights** $\boldsymbol{w} \in \mathbb{R}^d$ and **bias** $b \in \mathbb{R}$.
  - We must **learn** these parameters from training data.
  - $\boldsymbol{w}^\top \boldsymbol{x} = \langle \boldsymbol{w}, \boldsymbol{x} \rangle$ is an

Decision boundary: 1D

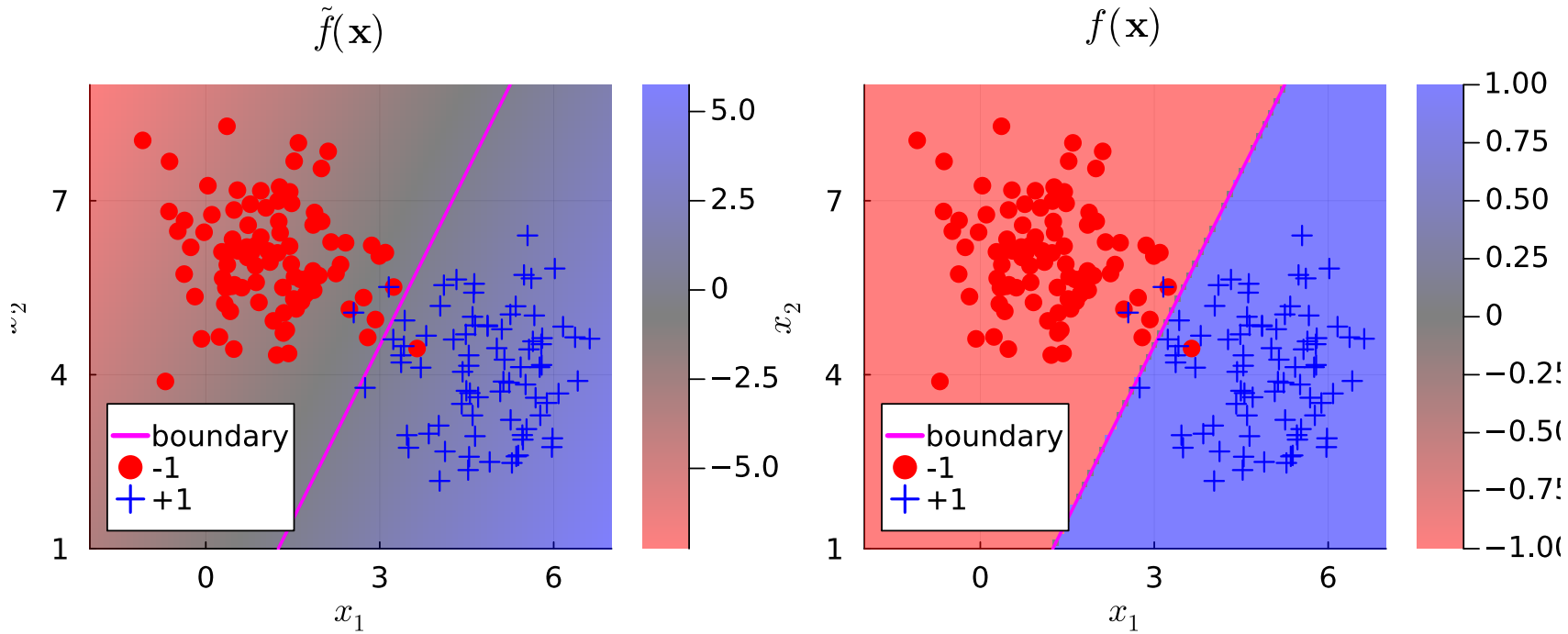
© 2026 Fessler



Here $\tilde{f}$ is an **affine function**. Geometrically the locus of points is a "line," but the function $\tilde{f}$ is not linear:

Decision boundary: 2D

© 2026 Fessler



We will describe several ways to define

The 0-1 loss, risk, and error probability

© 2026 Fessler

- If L is the **0-1 loss** ("all or nothing"):

$$L_{0\text{-}1}(y, \hat{y}) \triangleq$$

$$=$$

- then the **risk** is the **probability of error**:

$$R(f) =$$

- The expectation is w.r.t. the joint distribution $\mathrm{p}(\boldsymbol{x}, y)$.

Later we will discuss **surrogate losses** for the 0-1 loss for binary classification.

- Rewrite 0-1 loss:

$$L_{\text{0-1}}(y, f(\boldsymbol{x})) = \mathbb{I}_{\{y \neq f(\boldsymbol{x})\}} = \mathbb{I}_{\{y \neq \text{sign}(\tilde{f}(\boldsymbol{x}))\}} = \mathbb{I}_{\{y \cdot \tilde{f}(\boldsymbol{x}) \leq 0\}}.$$

- Hinge loss (used in **support vector machine** (**SVM**) methods):

$$L_{\text{hinge}}(y, \tilde{f}(\boldsymbol{x})) \triangleq \max\Big(0, 1 - y \cdot \tilde{f}(\boldsymbol{x})\Big).$$

- Logistic loss (used in **logistic regression**):

$$L_{\text{logistic}}(y, \tilde{f}(\boldsymbol{x})) \triangleq \log\Big(1 + \mathrm{e}^{-y \cdot \tilde{f}(\boldsymbol{x})}\Big).$$

  To be precise, this loss uses the negative log of the **standard logistic function** $z \mapsto 1/(1 + \exp(-z))$.

- They all are non-increasing functions of $y \cdot \tilde{f}(\boldsymbol{x})$.

Bayes classifier: definition

© 2026 Fessler

- A **classifier** is a function $f : \mathbb{R}^d \mapsto \mathcal{Y}$, *e.g.*, $\mathcal{Y} = \{1, \ldots, K\}$.
- The *best* classifier depends on the *data distribution* and the **performance measure**.
- One common performance measure is the **risk** defined by the **0-1 loss**, which equals the probability of error:

$$R(f) \triangleq \mathbb{E}_{\boldsymbol{X},Y}\left\{\mathbb{I}_{\{f(\boldsymbol{X}) \neq Y\}}\right\} = \mathsf{P}\{f(\boldsymbol{X}) \neq Y\}.$$

- The smallest **risk** (for the 0-1 loss) of any classifier is called the
  It is denoted:
- If $R(f) = R^*$, then classifier f is called a

CQ: risk for 0-1 loss

© 2026 Fessler

Q02.3 What is $R(f)$ when $f(\boldsymbol{x})$ always predicts class $\hat{k}$?

A: $R(f) = \underset{k \in \{1,\dots,K\}}{\arg\max}\ \pi_k g_k(\boldsymbol{x})$

B: $R(f) = \sum_{k \neq \hat{k}} \pi_k$

C: $R(f) = \sum_{k \neq \hat{k}} \pi_k g_k(\boldsymbol{x})$

D: $R(f) = \sum_{k \neq \hat{k}} g_k(\boldsymbol{x})$

E: $R(f) = \pi_{\hat{k}}$

Clearly this f is a terrible classifier.

How do we find an **optimal** classifier, *i.e.*, a **Bayes classifier**?

- **Theorem:** The following **maximum a posteriori** (**MAP**) estimator (a function from $\mathbb{R}^d$ to $\{1, \ldots, K\}$) is a **Bayes classifier**:

$$f^*(\boldsymbol{x}) =$$

Q02.4 Assuming the probability model is known perfectly, the **Bayes classifier** always (predicts the correct class, is unique).

A: (T,T) B: (T,F) C: (F,T) D: (F,F) **??**

Bayes classifier for binary classification

© 2026 Fessler

Consider the two-class case with labels $\mathcal{Y} = \{0, 1\}$ and recall

$$\eta_1(\boldsymbol{x}) \triangleq \mathsf{P}\{Y = 1 \mid \boldsymbol{X} = \boldsymbol{x}\} = 1 - \eta_0(\boldsymbol{x}).$$

By previous theorem, a **Bayes Classifier** here is

$$f^*(\boldsymbol{x}) \triangleq \underset{k \in \{0,1\}}{\arg\max}\, \eta_k(\boldsymbol{x}) = \begin{cases} 1, & \eta_1(\boldsymbol{x}) \geq \eta_0(\boldsymbol{x}) \\ 0, & \text{otherwise} \end{cases}$$

$$= \begin{cases} 1, & \eta_1(\boldsymbol{x}) \geq 1 - \eta_1(\boldsymbol{x}) \\ 0, & \text{otherwise} \end{cases} = \quad$$

- This classifier is sometimes called *the* **Bayes classifier**, but *any* classifier with $R = R^*$ is a **Bayes classifier**.
- Intuitive: if posterior probably of class 1 exceeds
  then predict

Continuing the two-class case, we have by **Bayes' rule**:

$$\begin{aligned}\eta_1(\boldsymbol{x}) &= \mathsf{P}\{Y = 1 \mid \boldsymbol{X} = \boldsymbol{x}\} \\ &= \frac{\mathsf{p}(\boldsymbol{x} \mid Y = 1)\,\mathsf{p}(Y = 1)}{\mathsf{p}(\boldsymbol{x})} \\ &= \frac{\pi_1 g_1(\boldsymbol{x})}{\pi_1 g_1(\boldsymbol{x}) + \pi_0 g_0(\boldsymbol{x})} = \frac{1}{1 + \frac{\pi_0 g_0(\boldsymbol{x})}{\pi_1 g_1(\boldsymbol{x})}}.\end{aligned}$$

Our Bayes classifier is equivalent to the

$$\frac{g_1(\boldsymbol{x})}{g_0(\boldsymbol{x})} \geq \frac{\pi_0}{\pi_1} \iff$$

Bayes classifier illustration

© 2026 Fessler

picture:

CQ Bayes classifier: risk examples

© 2026 Fessler

Consider four possible classification problems with $\mathcal{Y} = \{1, 2\}$:

1: $g_1 = U[3, 5],\ g_2 = U[3, 5]$

2: $g_1 = U[3, 6],\ g_2 = U[3, 7]$

3: $g_1 = U[3, 5],\ g_2 = U[6, 7]$

4: $g_1 = U[3, 7],\ g_2 = U[4, 5]$

Q02.5 Which problem leads to **Bayes risk** ($R^* = 0$, $R^* = \frac{1}{2}$)?
Enter one of these 2-digit numbers: 11, 12, 13, 14, 21, 22, 23, 24, 31 , 32, 33, 34, 41, 42, 43, 44, or "99" for none of these.

- In ML, generally the quantities $\{\pi_k, g_k(\boldsymbol{x}), \eta_k(\boldsymbol{x})\}$ are unknown, so the **Bayes classifier** is unknown.
- Instead, we **plug in** estimates *learned* from training data.
- We plug those estimates into the Bayes classifier formula to get a

Forthcoming examples:

- **linear discriminant analysis** (**LDA**) and **Naïve Bayes** classifiers have the form

$$f(\boldsymbol{x}) \triangleq \arg\max_k \hat{\pi}_k \hat{g}_k(\boldsymbol{x}).$$

(Naïve Bayes assumes the features are independent given the label Y.)

- **logistic regression** is essentially a **classifier** that has the form

$$f(\boldsymbol{x}) \triangleq \arg\max_k \hat{\eta}_k(\boldsymbol{x}).$$

- Other estimators are also possible.

Linear discriminant analysis: History

© 2026 Fessler

- **Fisher's linear discriminant** (Fisher, 1936) [4] seeks a "linear" classifier between two classes ($K = 2$), *i.e.*, $f(\boldsymbol{x}) = \text{sign}\left(\tilde{f}(\boldsymbol{x})\right)$, where $\tilde{f}$ is a linear (or affine) function of $\boldsymbol{x}$ [1, §9.2.6].
- "Note on discriminant functions" (Welch, 1939) [5] derives Bayes binary classifier when the distributions are known and specifies results for multivariate normal with the same covariance, where it coincides with Fisher's.
- Extensions for more classes with varying assumptions were made in Fisher (1938), Day and Sandomire (1942), Brown (1947), Rao (1952), Anderson (1958), Rao (1963) [6]

Linear discriminant analysis: Perspectives

© 2026 Fessler

- In early 20th century "discriminant" was used in two ways:
  - for generic classification problems without distributional assumptions
  - for hypothesis testing problems between multiple populations from known distributions.
- Today, **linear discriminant analysis** (**LDA**) is sometimes described either as
  - **Fisher's discriminant**
    (a heuristic rule with guarantees in certain Gaussian cases)
  - or as a **plug-in classifier**
    (our treatment) [1, §9.1].

Keep this in mind when reading about LDA elsewhere.

- Supervised classifier, starting with training data of feature vectors and labels: $\mathcal{D}_{\text{train}} = ((\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n))$.
- **LDA model** uses the **categorical distribution** (aka generalized Bernoulli) and conditional **multivariate Gaussian distributions**:

$$Y_1, \ldots, Y_n \overset{iid}{\sim}$$

$$\boldsymbol{X} \mid Y = k \sim$$

for some unknown class mean vectors $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K$ and

- Full data probabilistic model:

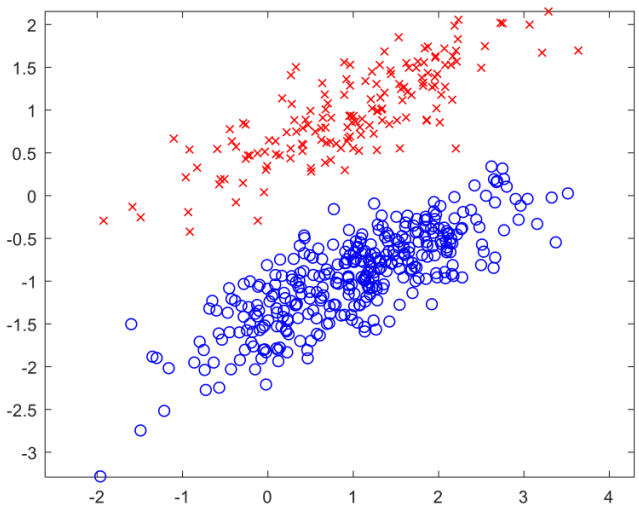
$$\mathsf{p}(\mathcal{D}_{\text{train}}) = \underbrace{\prod_{i=1}^{n}}_{} \mathsf{p}(\boldsymbol{x}_i \mid Y = y_i) \, \mathsf{P}\{Y = y_i\}$$

$$= \prod_{i=1}^{n} \mathsf{p}(\boldsymbol{x}_i \mid Y = y_i) \, \pi_{y_i}$$

Specifically, letting $|\boldsymbol{\Sigma}|$ denote the **determinant** of $\boldsymbol{\Sigma}$:

$$g_k(\boldsymbol{x}) = \mathrm{p}(\boldsymbol{x} \mid Y = k) = \frac{1}{\sqrt{|2\pi\boldsymbol{\Sigma}|}} \exp\left(-\tfrac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{x} - \boldsymbol{\mu}_k)\right).$$

- Thus $\mathrm{p}(\mathcal{D}_{\text{train}})$ is a **mixture of gaussians** (**MoG**) model.
- LDA is the **plug-in classifier** based on this model.
- We use training data to estimate



- **LDA** is the classifier obtained by plugging the following parameter estimates into the **Bayes classifier** formula:

$$\hat{\pi}_k =$$

$$\hat{\boldsymbol{\mu}}_k =$$

$$\widehat{\boldsymbol{\Sigma}} =$$

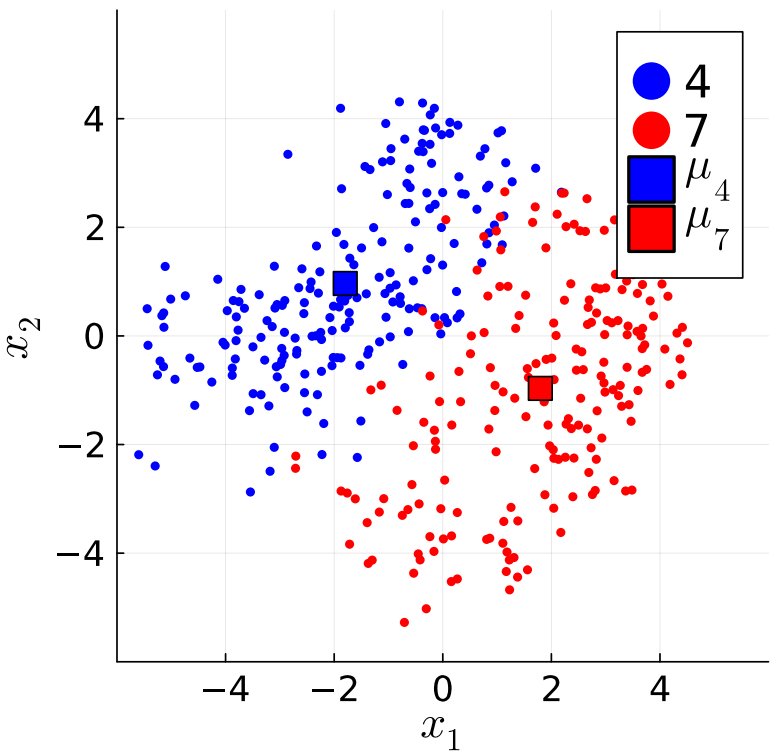
- These estimates are all **maximum likelihood estimates**
  - $\hat{\pi}_k$ is simply proportions (fractions or frequencies) of each class
  - $\hat{\boldsymbol{\mu}}_k$ is the
  - $\widehat{\boldsymbol{\Sigma}}$ is the

LDA: Estimates visualized

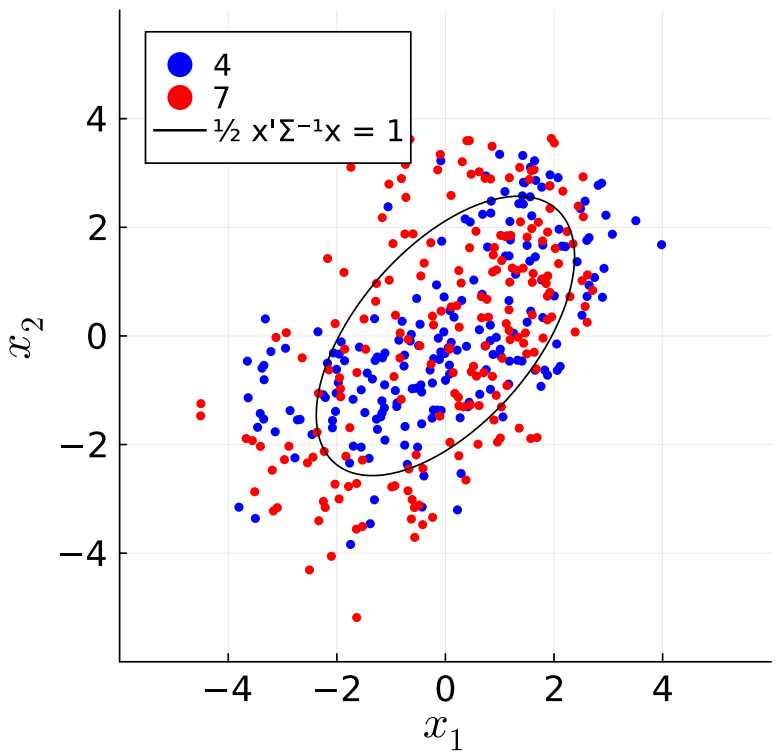
© 2026 Fessler

- Julia LDA demo with same data as k-NN demo.
- Visualize sample means $\{\hat{\boldsymbol{\mu}}_k\}$ and
- Ellipse is one contour line of $\phi(\boldsymbol{x}; \mathbf{0}, \widehat{\boldsymbol{\Sigma}})$, where $\frac{1}{2}\boldsymbol{x}^\top \widehat{\boldsymbol{\Sigma}} \boldsymbol{x} = 1$.

Train data with sample means



Train data de-meaned



- Recall the **Bayes classifier** is

$$f(\boldsymbol{x}) = \underset{k \in \{1,\dots,K\}}{\arg\max} \; \mathrm{P}\{Y = k \mid \boldsymbol{X} = \boldsymbol{x}\} = \underset{k}{\arg\max} \, \pi_k \, g_k(\boldsymbol{x}).$$

- Plug-in classifier:

$$f(\boldsymbol{x}) \triangleq \underset{k}{\arg\max} \, \hat{\pi}_k \, \hat{g}_k(\boldsymbol{x})$$

$$= \underset{1 \le k \le K}{\arg\max} \qquad \qquad (1)$$

where LDA's conditional generative model is based on

$$\phi(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) \triangleq |2\pi\boldsymbol{\Sigma}|^{-\frac{1}{2}} \exp\left(-\frac{1}{2} (\boldsymbol{x} - \boldsymbol{\mu})^\top \qquad (\boldsymbol{x} - \boldsymbol{\mu}) \right).$$

- One could classify $\boldsymbol{x}$ by evaluating (1) numerically.

- Plug-in covariance estimate:

$$\widehat{\boldsymbol{\Sigma}} = \frac{1}{n} \sum_{i=1}^{n} (\boldsymbol{x}_i - \hat{\boldsymbol{\mu}}_{y_i})(\boldsymbol{x}_i - \hat{\boldsymbol{\mu}}_{y_i})^\top$$

- Gaussian pdf uses the *inverse* of the covariance matrix.

Q02.6 Having $n \geq d$ is (necessary, sufficient) for $\widehat{\boldsymbol{\Sigma}}$ to be invertible.

A: (T,T) B: (T,F) C: (F,T) D: (F,F) **??**

LDA demo: Linear decision boundary

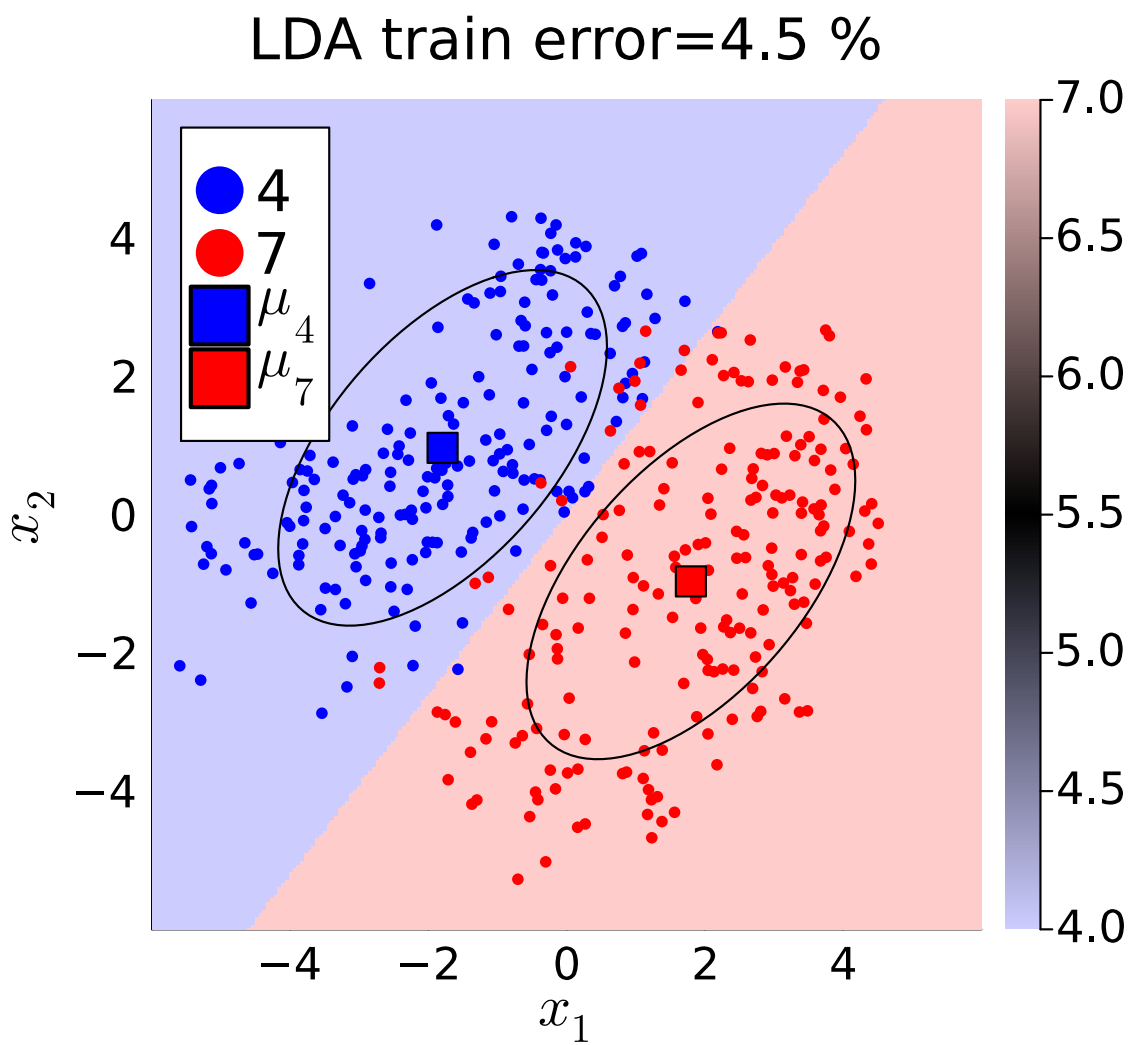
© 2026 Fessler

LDA classifier:

$f(\boldsymbol{x}) = \arg\max_k \hat{\pi}_k \, \hat{g}_k(\boldsymbol{x})$

Julia LDA demo

Next we derive this property for general K.



LDA is a "linear" classifier: General K

© 2026 Fessler

$$f(\boldsymbol{x}) = \arg\max_k \hat{\pi}_k \, \hat{g}_k(\boldsymbol{x}) = \arg\max_k \; \log \hat{\pi}_k + \log \phi(\boldsymbol{x}; \hat{\boldsymbol{\mu}}_k, \widehat{\boldsymbol{\Sigma}})$$

where

LDA is a linear classifier: Binary case

© 2026 Fessler

- Binary setting with $Y \in \mathcal{Y} = \{-1, 1\}$.
- Now the LDA classifier simplifies greatly to

$$f(\boldsymbol{x}) = \text{sign}(\boldsymbol{w}^\top \boldsymbol{x} + b)$$

where

$\boldsymbol{w} =$

$b =$

- Exercise: solve for these formulae by changing argmax to the sign of the difference.

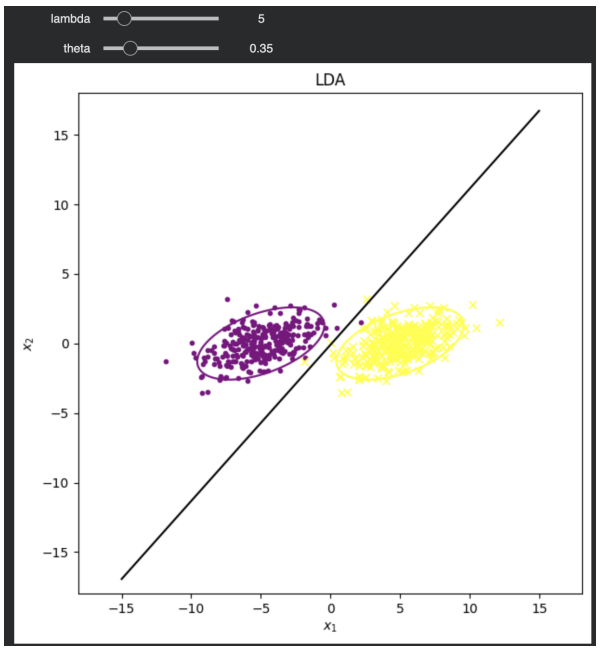
Q02.7 The number of hyperparameters for LDA is:

A: 0 B: 1 C: d D: n E: K

Synthetic data with $\boldsymbol{\mu}_{-1} = \left[\begin{smallmatrix} -5 \\ 0 \end{smallmatrix}\right]$ and $\boldsymbol{\mu}_1 = \left[\begin{smallmatrix} 5 \\ 0 \end{smallmatrix}\right]$ and covariance:

$$\boldsymbol{\Sigma} = \boldsymbol{R} \begin{bmatrix} \lambda & 0 \\ 0 & 1 \end{bmatrix} \boldsymbol{R}^\top, \qquad \boldsymbol{R} \triangleq \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}.$$

- If $\boldsymbol{\Sigma}$ is multiple of identity, decision boundary is orthogonal (with respect to the dot product) to $\hat{\boldsymbol{\mu}}_1 - \hat{\boldsymbol{\mu}}_{-1}$.
- Otherwise, decision boundary is skewed by covariance matrix.



Python colab demo with interactive λ and θ.

Generative vs discriminative

© 2026 Fessler

- A machine learning algorithm is **generative** if it is based on a full probabilistic model of the observed data:
  - Assume some functional form or probabilistic model for $\{\pi_k, g_k(\boldsymbol{x})\}$.
  - Estimate parameters of $\pi_k, g_k(\boldsymbol{x})$ from training data.
  - Use plug-in and **Bayes' rule** to calculate/estimate $\eta_k(\boldsymbol{x})$ from $\{\hat{\pi}_k, \hat{g}_k\}$.
- An algorithm is **discriminative** if it assumes a partial or no probabilistic model.
  - Assume some functional form or probabilistic model for $\eta_k(\boldsymbol{x})$.
  - Estimate (*i.e.*, learn) $\eta_k(\boldsymbol{x})$ directly from training data.
- In both cases, classify $\boldsymbol{x}$ using $\arg\max_k \hat{\eta}_k(\boldsymbol{x})$.
- See [1, §9.4] [7] for discussions.

Parametric vs nonparametric

© 2026 Fessler

- A learning algorithm is **parametric** if the number of model parameters* is independent of sample size n.
- An algorithm is **nonparametric** if the number of model parameters* grows with sample size n.
- *Model parameters = parameters needed to represent the function (=model) produced by the learning algorithm
- Example.
  Using a histogram as a probability **density estimator** is:
  - **parametric** if the number of bins is
  - **nonparametric** if the **number of bins**

CQ: LDA properties

© 2026 Fessler

Q02.8 The LDA classifier is

A: generative & parametric

B: generative & nonparametric

C: discriminative & parametric

D: discriminative & nonparametric

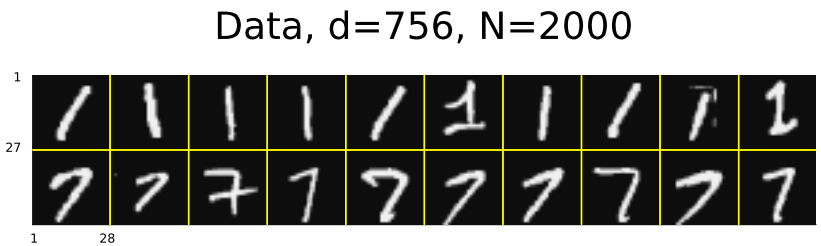
E: none of these

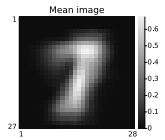
Q02.9 The LDA classifier always exists for any $\mathcal{D}_{\text{train}}$ with $n > d$.

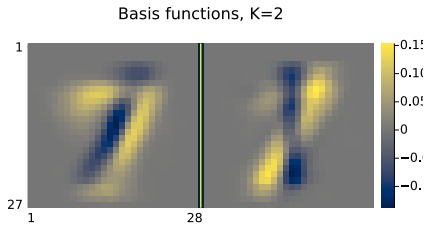
A: True B: False **??**

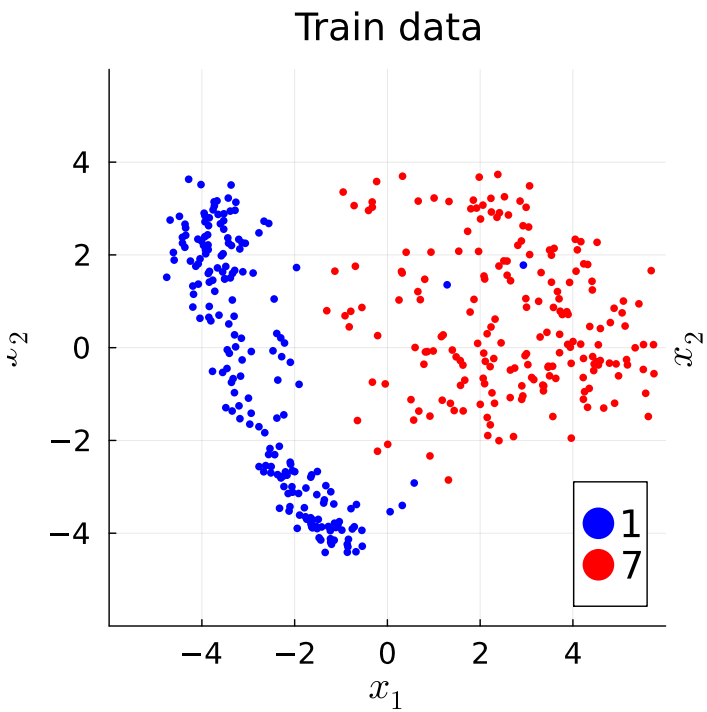
Classifier demo for digits 1,7: Data

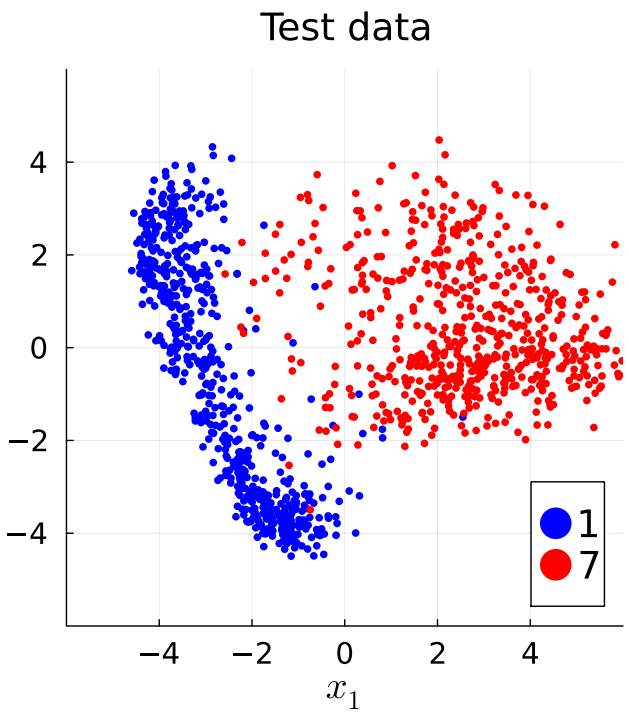
© 2026 Fessler







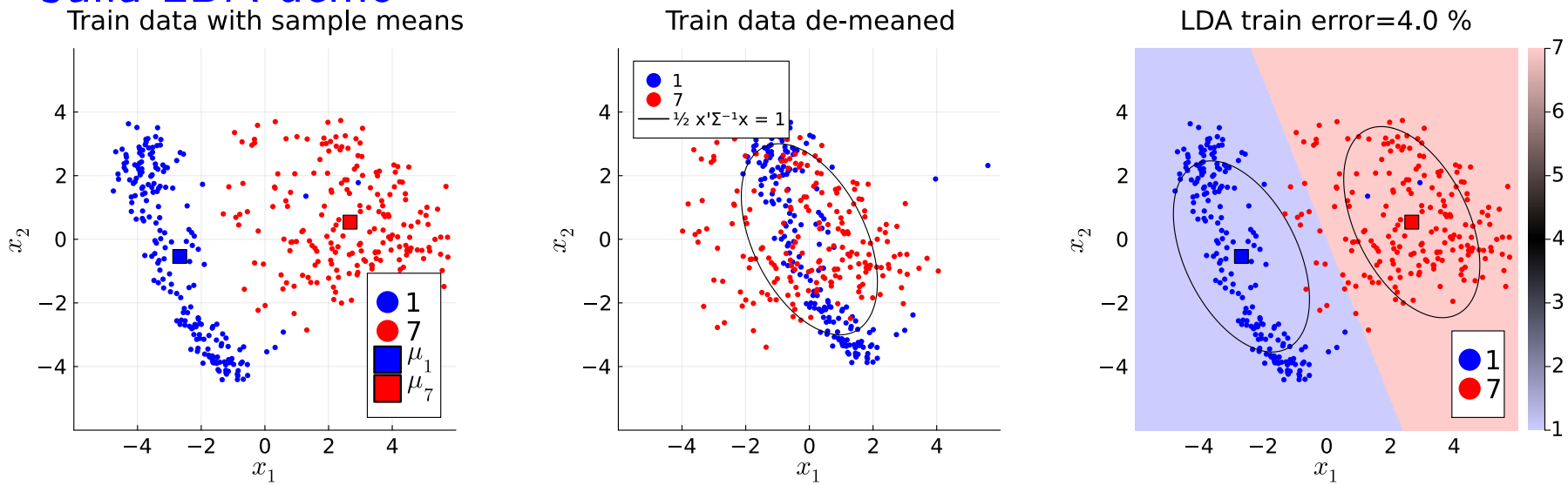




LDA classifier for digits 1,7

© 2026 Fessler

Julia LDA demo



The covariance matrices $\boldsymbol{\Sigma}_k$ differ *greatly* between the two classes.

Quadratic discriminant analysis: Data model

© 2026 Fessler

- Training data $\mathcal{D}_{\text{train}} = ((\boldsymbol{x}_1, y_1), \dots, (\boldsymbol{x}_n, y_n))$.
- **quadratic discriminant analysis** (**QDA**) probability model:

$$\begin{aligned} \text{labels:} \quad & Y_1, \dots, Y_n \overset{iid}{\sim} \text{discrete}(\pi_1, \dots, \pi_K) \\ \text{features:} \quad & \boldsymbol{X} \mid Y = k \sim \mathcal{N}(\boldsymbol{\mu}_k, \quad), \quad k = 1, \dots, K \end{aligned}$$

for some unknown $\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_K$ and Equivalently:

$$g_k(\boldsymbol{x}) = |2\pi \quad|^{-1/2} \exp\left(-\frac{1}{2} (\boldsymbol{x} - \boldsymbol{\mu}_k)^\top \quad (\boldsymbol{x} - \boldsymbol{\mu}_k) \right).$$

- QDA is the plug-in rule based on this model.
- We use training data $\mathcal{D}_{\text{train}}$ to estimate

Quadratic discriminant analysis: Estimates

© 2026 Fessler

- QDA is the classifier obtained by plugging the following parameter estimates into the Bayes classifier formula:

$$\hat{\pi}_k = \frac{n_k}{n}, \qquad n_k = |\{i : y_i = k\}|$$

$$\hat{\boldsymbol{\mu}}_k = \frac{1}{n_k} \sum_{i \,:\, y_i = k} \boldsymbol{x}_i$$

$$\widehat{\boldsymbol{\Sigma}}_k =$$

- These estimates are all **maximum likelihood estimates**.
- $\hat{\boldsymbol{\mu}}_k$ is the **sample mean** for class k.
- $\widehat{\boldsymbol{\Sigma}}_k$ is the
- All model parameters for $g_k(\boldsymbol{x}) = \mathrm{p}(\boldsymbol{x} \mid Y = k)$ are estimated from *only* the samples with label k, as is typical in many Bayesian methods [3, p. 91].

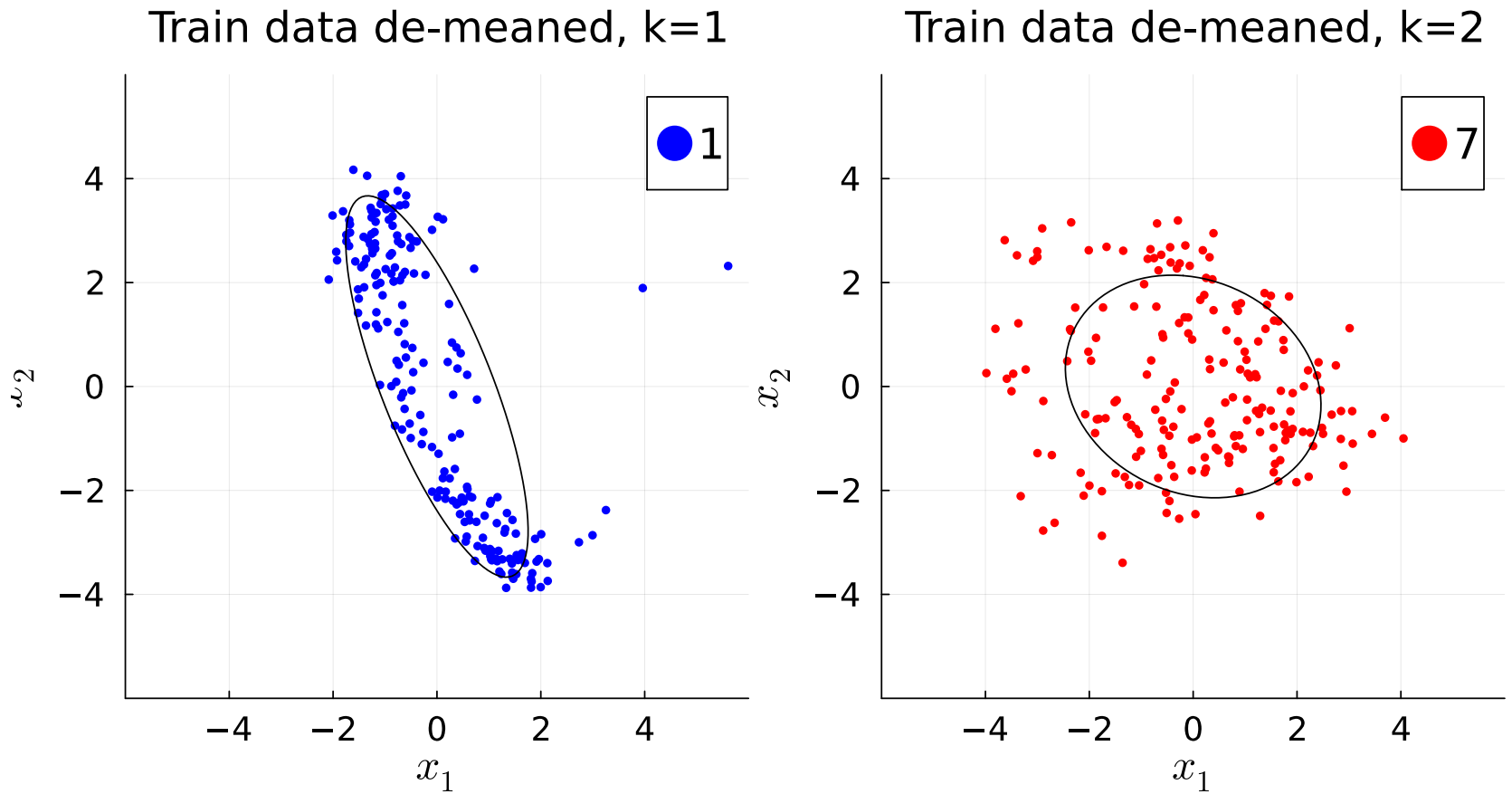
QDA estimates illustrated

© 2026 Fessler

Within-class sample covariance matrix:

$$\widehat{\boldsymbol{\Sigma}}_k = \frac{1}{n_k} \sum_{i\,:\,y_i = k} (\boldsymbol{x}_i - \hat{\boldsymbol{\mu}}_k)(\boldsymbol{x}_i - \hat{\boldsymbol{\mu}}_k)^\top$$

Julia QDA demo



Quadratic discriminant analysis: Binary case

© 2026 Fessler

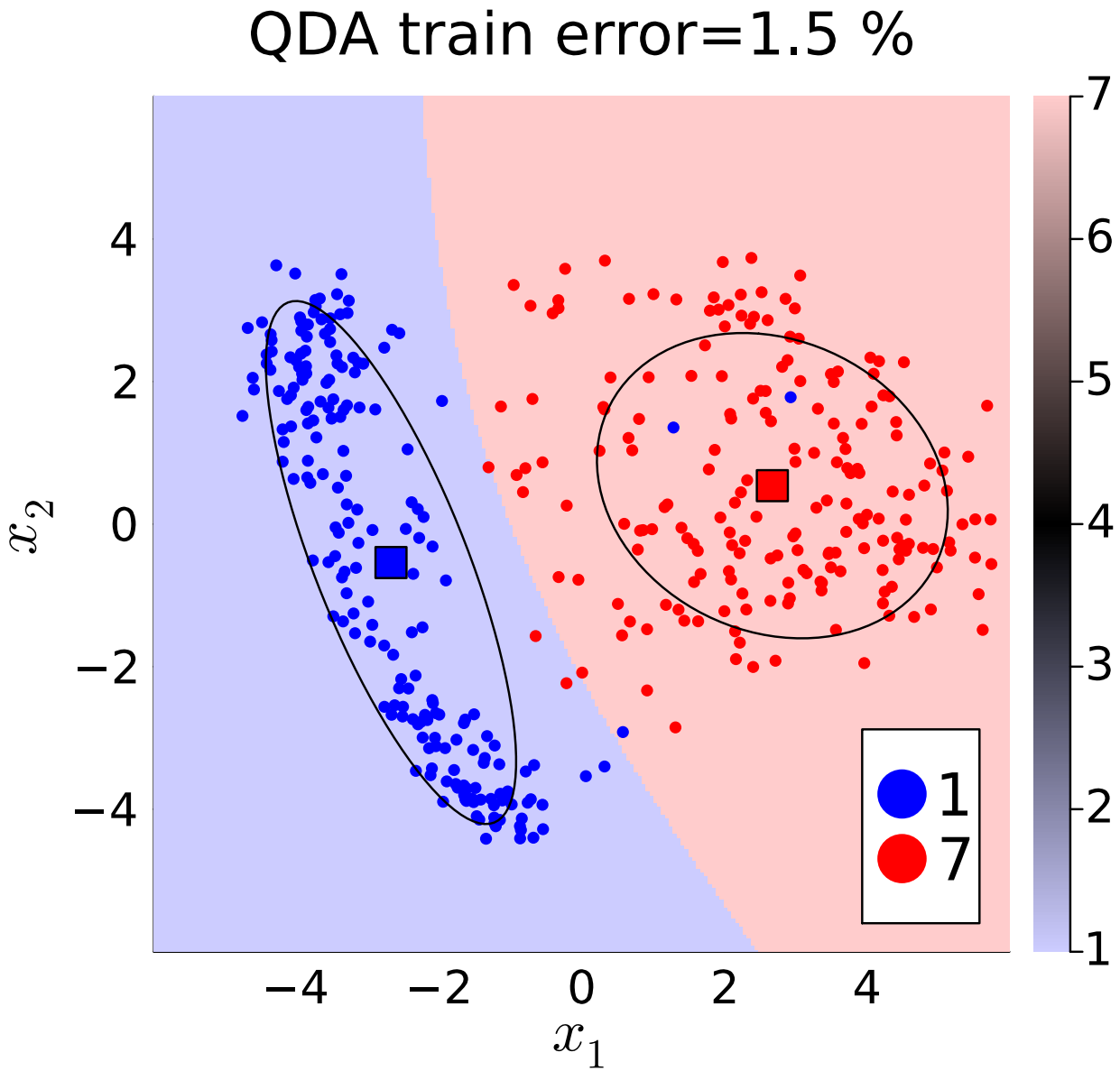
- Binary setting, $Y \in \{-1, 1\}$.
- The QDA classifier is

$$f(\boldsymbol{x}) = \arg\max_k \hat{\pi}_k \hat{g}_k(\boldsymbol{x}) = \text{sign}\left(\frac{1}{2} \boldsymbol{x}^\top \boldsymbol{A} \boldsymbol{x} + \boldsymbol{b}^\top \boldsymbol{x} + c \right),$$

  for certain matrix $\boldsymbol{A}$, vector $\boldsymbol{b}$, scalar c.
- The decision boundaries are quadratics.

- Exercise: Determine $\boldsymbol{A}, \boldsymbol{b}, c$.
  You will again use sign of the difference instead of argmax, and then you manipulate the relevant terms of the Gaussian likelihoods.

Julia QDA demo



Fisher's original iris data (application)

© 2026 Fessler





Image attributions:
setosa: CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=170298,
versicolor: CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=248095, and
virginica: By Frank Mayfield - originally posted to Flickr as Iris virginica shrevei BLUE FLAG, CC BY-SA 2.0, https://commons.wikimedia.org/w/index.php?curid=9805580

Fisher's original iris data

© 2026 Fessler

- 150 observations made on $K = 3$ classes of the iris flower: Iris setosa, Iris versicolor, and Iris virginica.
- $d = 4$ features:
  - **sepal** length
  - sepal width
  - petal length
  - petal width.

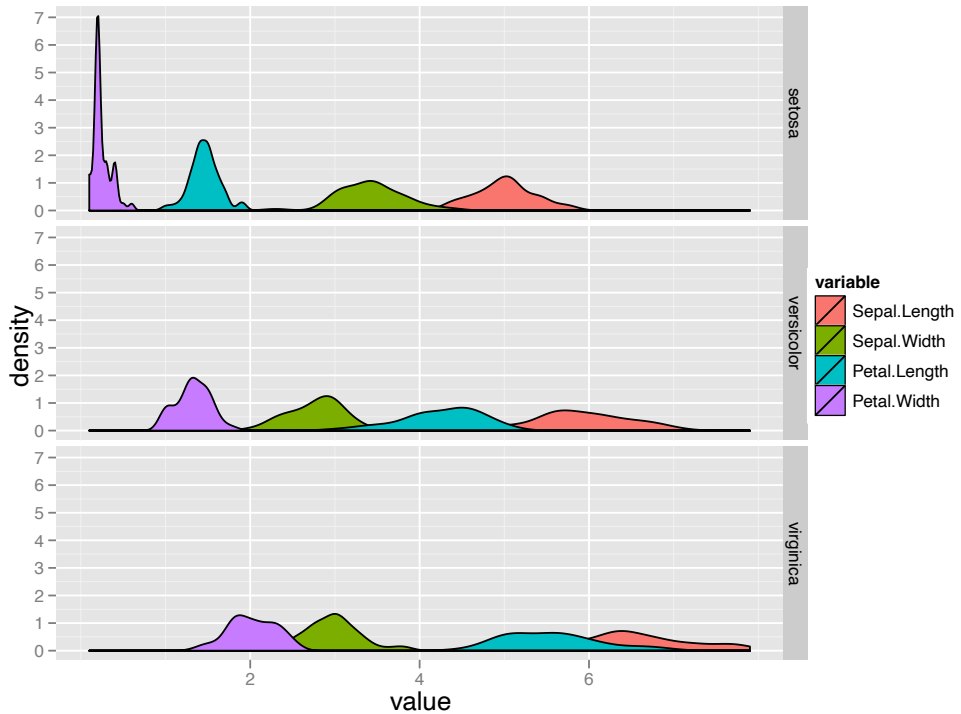


Figure courtesy Larry Wasserman

https://www.stat.cmu.edu/~larry/=sml/linearclassification.pdf

Q02.10 We could classify accurately based just on petal length.

A: True B: False **??**

CQ:: LDA accuracy

© 2026 Fessler

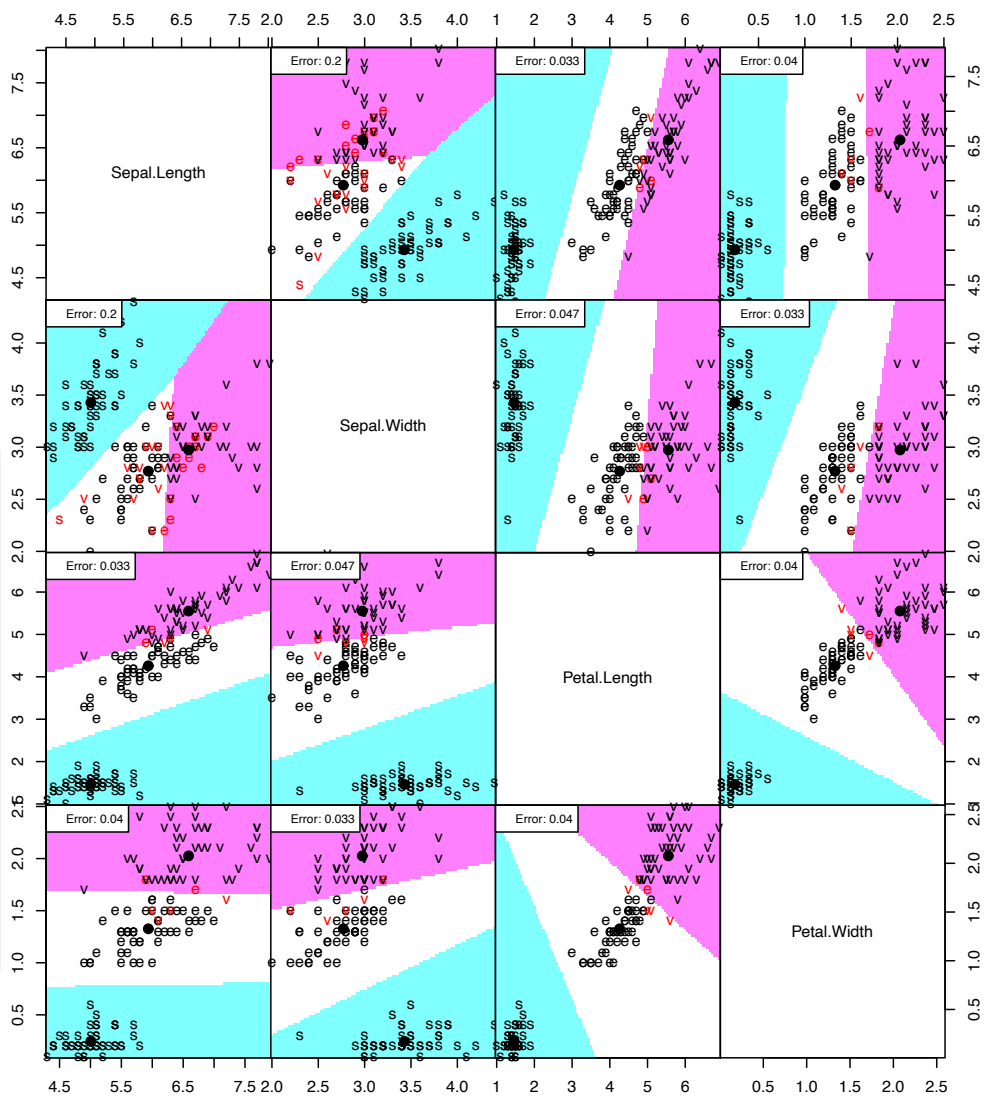
Classifying the Iris data using LDA. The multiple figure array illustrates the classification of observations based on LDA for every combination of two features. The classification boundaries and error are obtained by simply restricting the data to a given pair of features before fitting the model. In these plots, "s" represents the class label Iris setosa, "e" represents the class label Iris versicolor, and "v" represents the class label Iris virginica. The red letters illustrate the misclassified observations.

Q02.11 Which class of flower is most accurately classified by LDA per these plots?

A: Setosa ("s")

B: Versicolor ("e")

C: Virginica ("v")



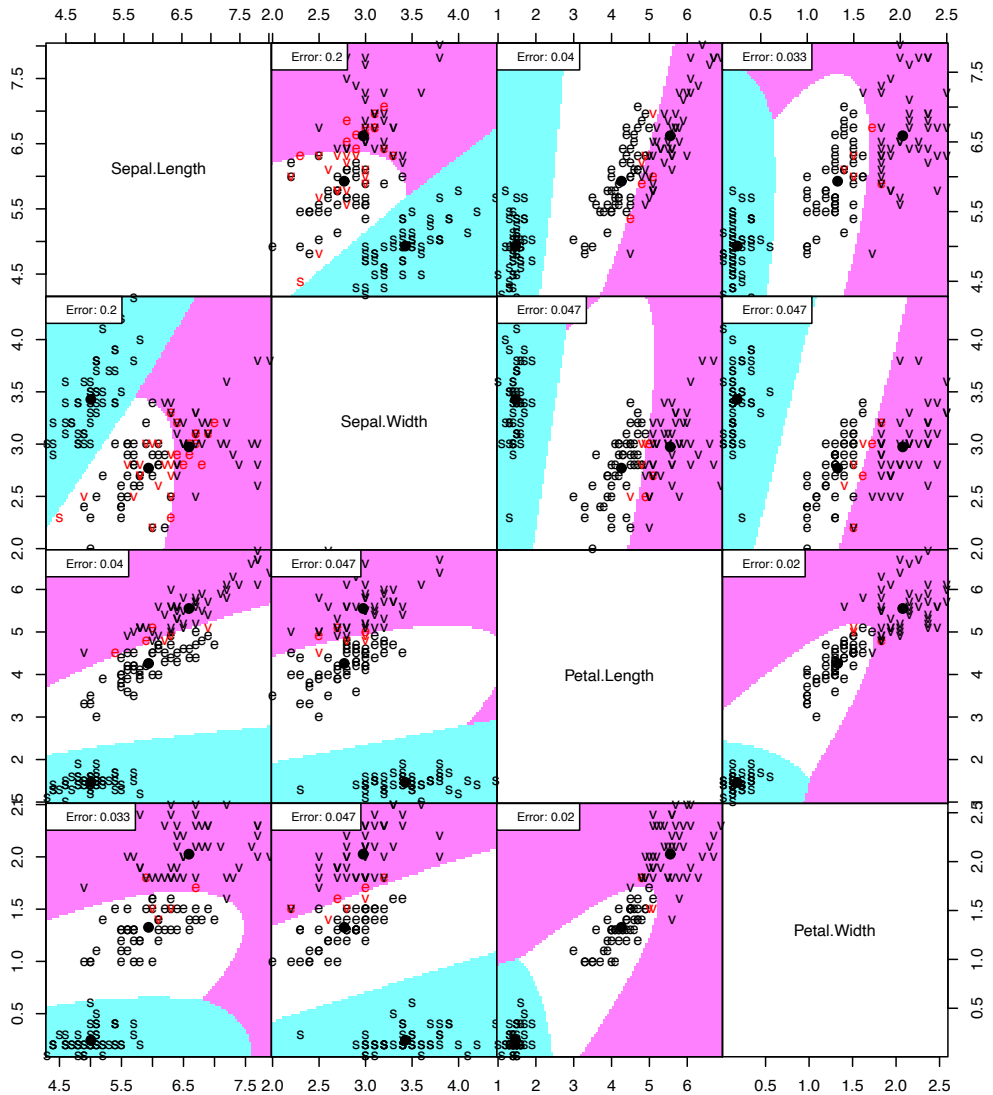
QDA on Fisher's original iris data

© 2026 Fessler

Classifying the Iris data using QDA. The multiple figure array illustrates the classification of observations based on QDA for every combination of two features. The classification boundaries are displayed and the classification error by simply casting the data onto these two features are calculated. In these plots, "s" represents the class label Iris setosa, "e" represents the class label Iris versicolor, and "v" represents the class label Iris virginica. The red letters illustrate the misclassified observations. Overall training error is 0.02.

Figure & text courtesy Larry Wasserman

https://www.stat.cmu.edu/~larry/=sml/linearclassification.pdf



Linear discriminant analysis: Summary

© 2026 Fessler

- One of the very earliest ML algorithms invented (c. 1930s)
- Binary classifier is "linear:"

$$f(\boldsymbol{x}) = \text{sign}\Big(\tilde{f}(\boldsymbol{x})\Big), \quad \tilde{f}(\boldsymbol{x}) = \boldsymbol{w}^\top \boldsymbol{x} + b$$

- $\boldsymbol{w}$ and b are functions of parameter estimates $\{\hat{\pi}_k\}$, $\{\hat{\boldsymbol{\mu}}_k\}$, $\widehat{\boldsymbol{\Sigma}}^{-1}$
- Requires $n \geq d$ although extensions are possible.
- Many applications throughout history, although better methods now available.
- **Logistic regression** almost always preferable
  - makes weaker assumptions
  - learns $f(\boldsymbol{x}) = \text{sign}\big(\boldsymbol{w}^\top \boldsymbol{x} + b\big)$ directly from training data
- Optional: read about **Mahalanobis distance** in Prof. Scott's notes on Canvas.

References I

The Annals of Eugenics, established 1925, is now named The Annals of Human Genetics. While **Fisher** was an undisputed statistical giant, he also was part of an awful history that we should not ignore [8, 9].

[1] K. P. Murphy.
Probabilistic machine learning: An introduction.
MIT Press, 2022.

[2] C. Bishop.
Pattern recognition and machine learning.
Springer, 2006.

[3] R. O. Duda, P. E. Hart, and D. G. Stork.
Pattern classification.
Wiley, New York, 2001.

[4] Ronald A Fisher.
The use of multiple measurements in taxonomic problems.
Annals of eugenics, 7(2):179–188, 1936.

[5] Bernard L Welch.
Note on discriminant functions.
Biometrika, 31(1/2):218–220, 1939.

[6] Somesh Das Gupta.
Theories and methods in classification: a review.
Discriminant analysis and applications, pages 77–137, 1973.

References II

[7] Andrew Ng and Michael Jordan.
On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes.
In T. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems*, volume 14. MIT Press, 2001.

[8] Walter Bodmer, RA Bailey, Brian Charlesworth, Adam Eyre-Walker, Vernon Farewell, Andrew Mead, and Stephen Senn.
The outstanding scientist, RA Fisher: his views on eugenics and race.
Heredity, 126(4):565–576, 2021.

[9] Eric Michael Johnson.
Ronald Fisher is not being 'cancelled', but his eugenic advocacy should have consequences, 2021.
https://thisviewoflife.com/ronald-fisher-is-not-being-cancelled-but-his-eugenic-advocacy-should-have-consequences/.