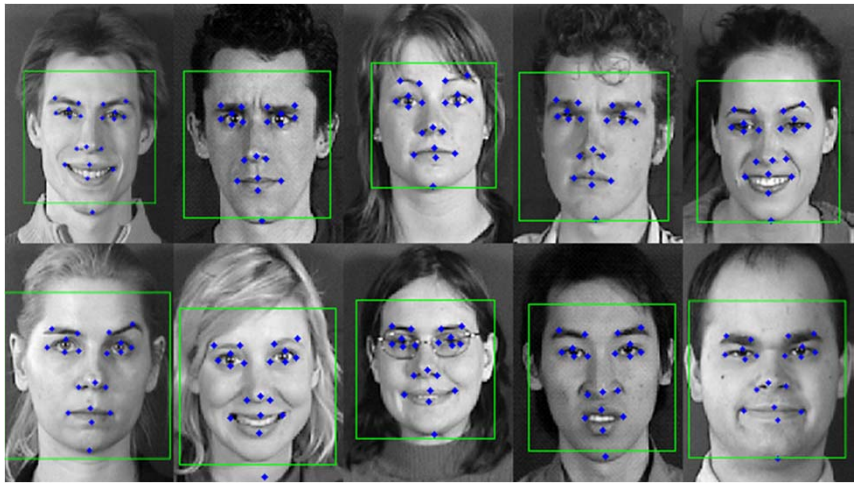
Learning from Data
3. Classification: Gaussian Discriminant Analysis

Spring 2017

Classification

- Supervised (preditive) learning: given examples with labels, predict labels for all unseen examples
  - Classification:
    - label = category,
    - $\mathbf{y} \in \mathcal{Y} = \{1, \ldots, m\}$, m = number of classes
    - $\ell(\mathbf{x}, y, h) = 1(h(\mathbf{x}) \neq y)$, Risk = $P(Y \neq h(\mathbf{X})) = P(\text{Error})$



$\mathbf{x}$ = facial geometry features

y = gender label

Gaussian Discriminant Analysis

- Parametric model for data
- Generative learning despite “discriminant” in name
- Feature vectors: $\mathbf{x} \in \mathcal{X} = \mathbb{R}^d$
- Model:

$$
\begin{aligned}
p(\mathbf{x}, y|\theta) &= p(y|\theta) \cdot p(\mathbf{x}|y, \theta) \\
&= p(y) \cdot \mathcal{N}(\boldsymbol{\mu}_y, \Sigma_y)(\mathbf{x}) \\
&= p(y) \cdot \frac{1}{\left(\sqrt{2\pi}\right)^d \sqrt{\det\left(\Sigma_y\right)}} \exp\left\{ -\frac{1}{2}\left(\mathbf{x} - \boldsymbol{\mu}_y\right)^\top \Sigma_y^{-1} \left(\mathbf{x} - \boldsymbol{\mu}_y\right) \right\}
\end{aligned}
$$

- Parameters: $\theta = \{(\underbrace{p(y)}_{\text{scalar}}, \underbrace{\boldsymbol{\mu}_y}_{d\times 1 \text{ vector}}, \underbrace{\Sigma_y}_{d\times d \text{ matrix}}), y = 1, \ldots, m\}$
- Total number of scalar parameters: $(1+d+d(d+1)/2)m$

Bayes classifier (θ known): for 0-1 loss, MPE rule = MAP rule

$$
\begin{aligned}
h_{\mathrm{MAP}}(\mathbf{x}) &= \arg \max_{y=1,\ldots,m} p(y|\mathbf{x},\theta) \\
&= \arg \min_{y} \left[-\ln p(\mathbf{x}|y,\theta) - \ln p(y) \right] \\
&= \arg \min_{y} \left[\frac{1}{2} \underbrace{(\mathbf{x}-\boldsymbol{\mu}_y)^\top \Sigma_y^{-1} (\mathbf{x}-\boldsymbol{\mu}_y)}_{\text{class-dependent quadratic in } \mathbf{x}} + \underbrace{\frac{1}{2} \ln \det\left(\Sigma_y\right) - \ln p(y)}_{\text{class-dependent scalar offset}} \right] \\
&= h_{\mathrm{QDA}}(\mathbf{x})
\end{aligned}
$$

- Quadratic Discriminant Analysis (QDA)

Bayes classifier (θ known): for 0-1 loss, MPE rule = MAP rule

- $\sqrt{\mathbf{z}^\top \Sigma^{-1} \mathbf{z}} = \|\mathbf{z}\|_\Sigma$ can be proved to behave just like the length of $\mathbf{z}$, i.e., a norm, although weighted by Σ

- Thus, $(\mathbf{x} - \boldsymbol{\mu})^\top \Sigma^{-1} (\mathbf{x} - \boldsymbol{\mu})$ is like the squared (weighted) distance between $\mathbf{x}$ and $\boldsymbol{\mu}$

- $\sqrt{(\mathbf{x} - \boldsymbol{\mu})^\top \Sigma^{-1} (\mathbf{x} - \boldsymbol{\mu})}$ is called the Mahalanobis distance between $\mathbf{x}$ and $\boldsymbol{\mu}$. It reduces to the Euclidean distance when Σ is the identity matrix.

- It is the multidimensional generalization of the idea of measuring how many standard deviations is a value from the mean.

Bayes classifier (θ known): for 0-1 loss, MPE rule = MAP rule

- The set of all x such that $p(\mathbf{x}|y,\theta) =$ constant is given by:

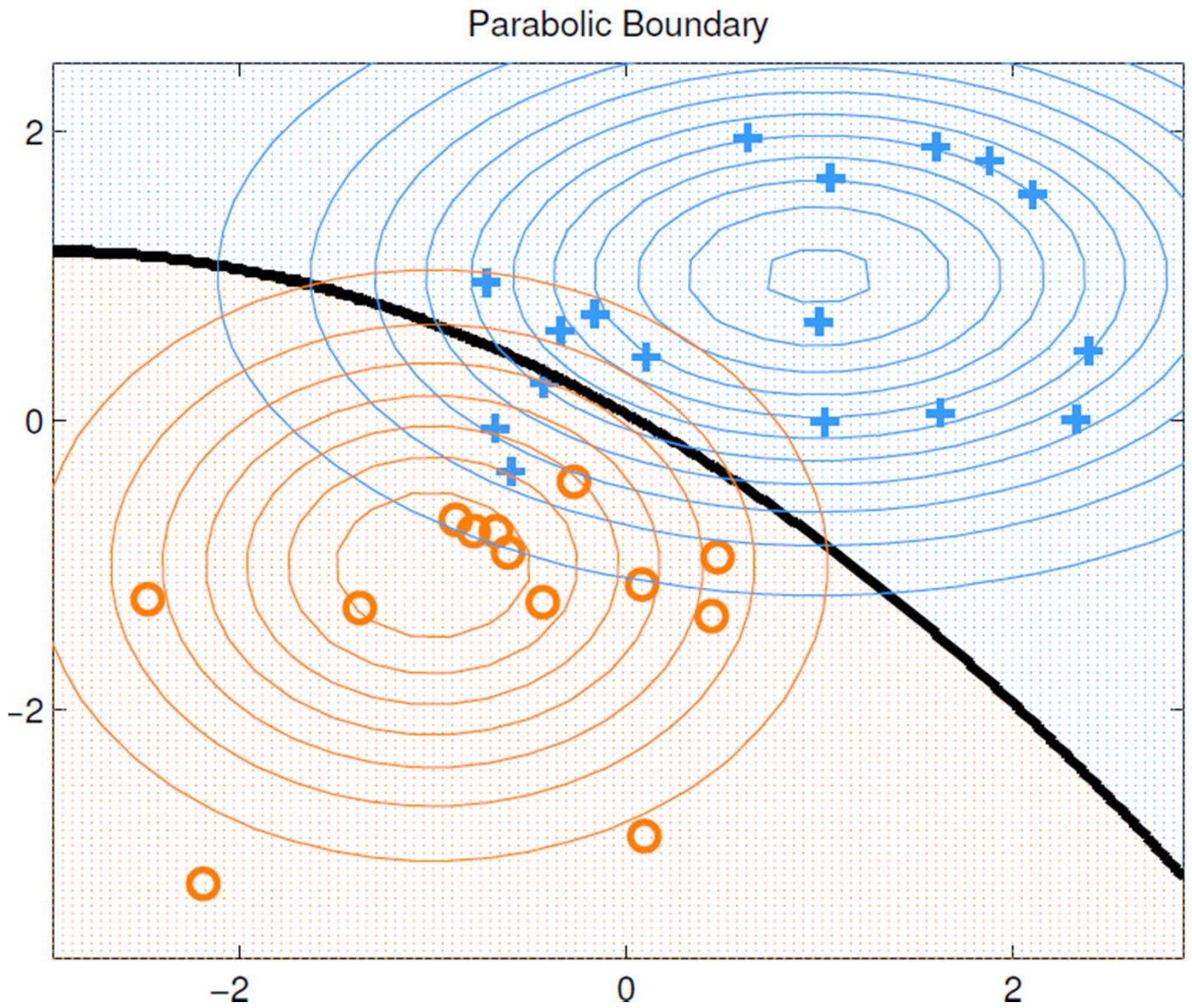
$$(\mathbf{x} - \boldsymbol{\mu}_y)^\top \Sigma_y^{-1} (\mathbf{x} - \boldsymbol{\mu}_y) = \text{constant}(y)$$

- This is the equation of a d-dimensional ellipsoid centered at $\boldsymbol{\mu}_y$.The principal directions of the ellipsoidal-shaped spread are determined by the orthonormal eigenvectors of Σ_y .

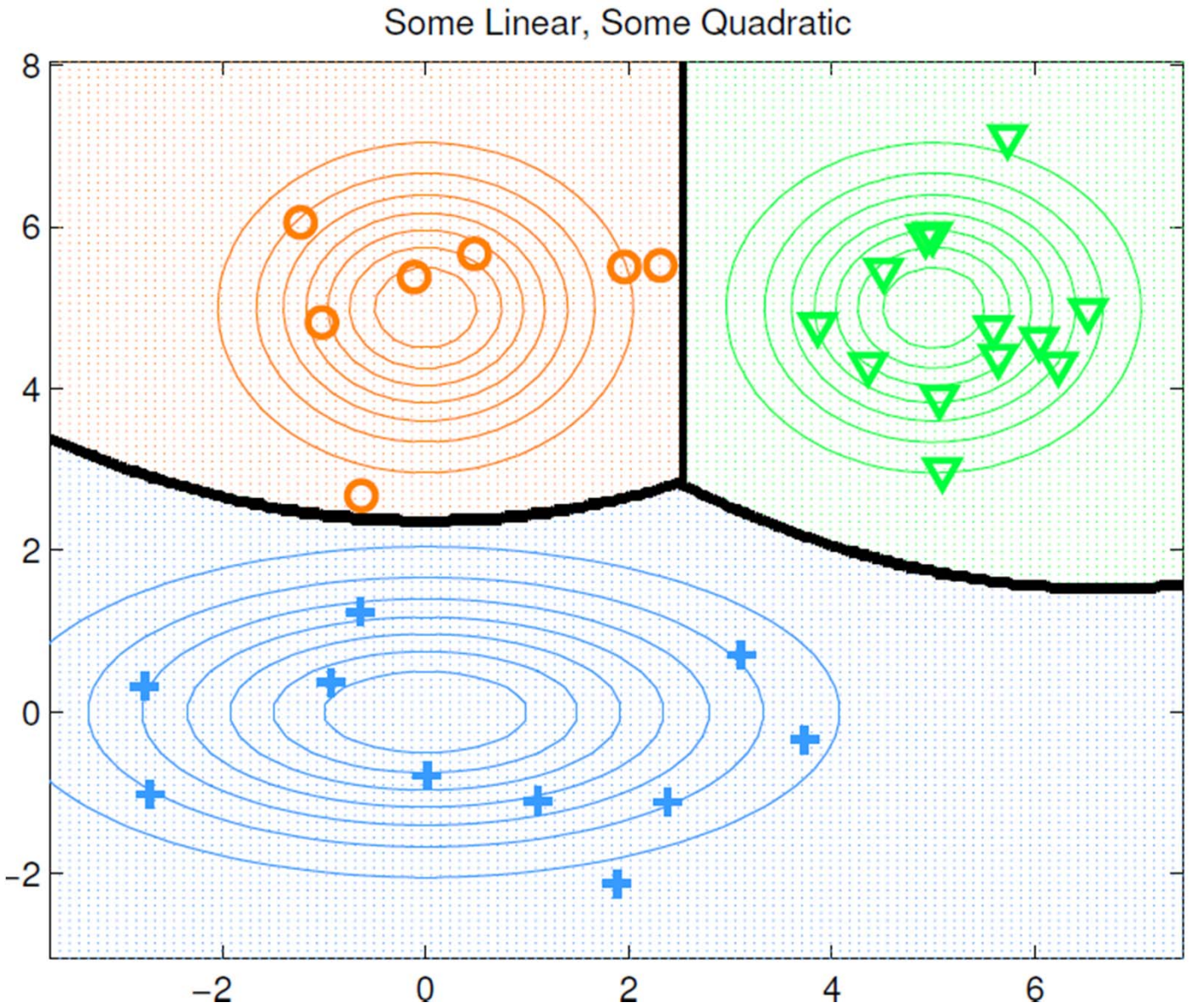
- QDA rule in words: map $\mathbf{x}$ to the class whose mean vector $\boldsymbol{\mu}_y$ is closest to it. Here, closeness is measured by a suitably weighted and offset-adjusted distance between $\mathbf{x}$ and $\boldsymbol{\mu}_y$.

- Decision boundaries between classes: typically piece-wise parabolic, but can be linear (planar) if Σ_y's of adjacent classes are equal.

QDA visualization: m = 2, d = 2



QDA visualization: $m = 3$, $d = 2$



Linear Discriminant Analysis (LDA)

- Special case of QDA when all classes have the same covariance matrix: $\Sigma_y = \Sigma$ for all y
- Then the quadratic rule becomes a linear rule:

$$
\begin{aligned}
h_{\mathrm{LDA}}(\mathbf{x}) &= \arg\min_y \left[\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_y)^\top \Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu}_y) + \frac{1}{2}\ln\det\big(\Sigma\big) - \ln p(y) \right] \\
&= \arg\min_y \Bigg[\underbrace{\frac{1}{2}\mathbf{x}^\top\Sigma^{-1}\mathbf{x}}_{\text{does not depend on } y} - \boldsymbol{\mu}_y^\top\Sigma^{-1}\mathbf{x} + \frac{1}{2}\boldsymbol{\mu}_y^\top\Sigma^{-1}\boldsymbol{\mu}_y - \ln p(y) \Bigg] \\
&= \arg\max_y \left[\underbrace{\big(\boldsymbol{\mu}_y^\top\Sigma^{-1}\big)}_{\boldsymbol{\beta}_y^\top}\mathbf{x} \underbrace{-\frac{1}{2}\boldsymbol{\mu}_y^\top\Sigma^{-1}\boldsymbol{\mu}_y + \ln p(y)}_{\gamma_y} \right] \\
&= \arg\max_y \Big[\underbrace{\boldsymbol{\beta}_y^\top\mathbf{x}}_{\text{projection}} + \underbrace{\gamma_y}_{\text{offset}} \Big]
\end{aligned}
$$

Linear Discriminant Analysis (LDA)

$$h_{\text{LDA}}(\mathbf{x}) = \arg\max_y \left[\boldsymbol{\beta}_y^\top \mathbf{x} + \gamma_y\right]$$

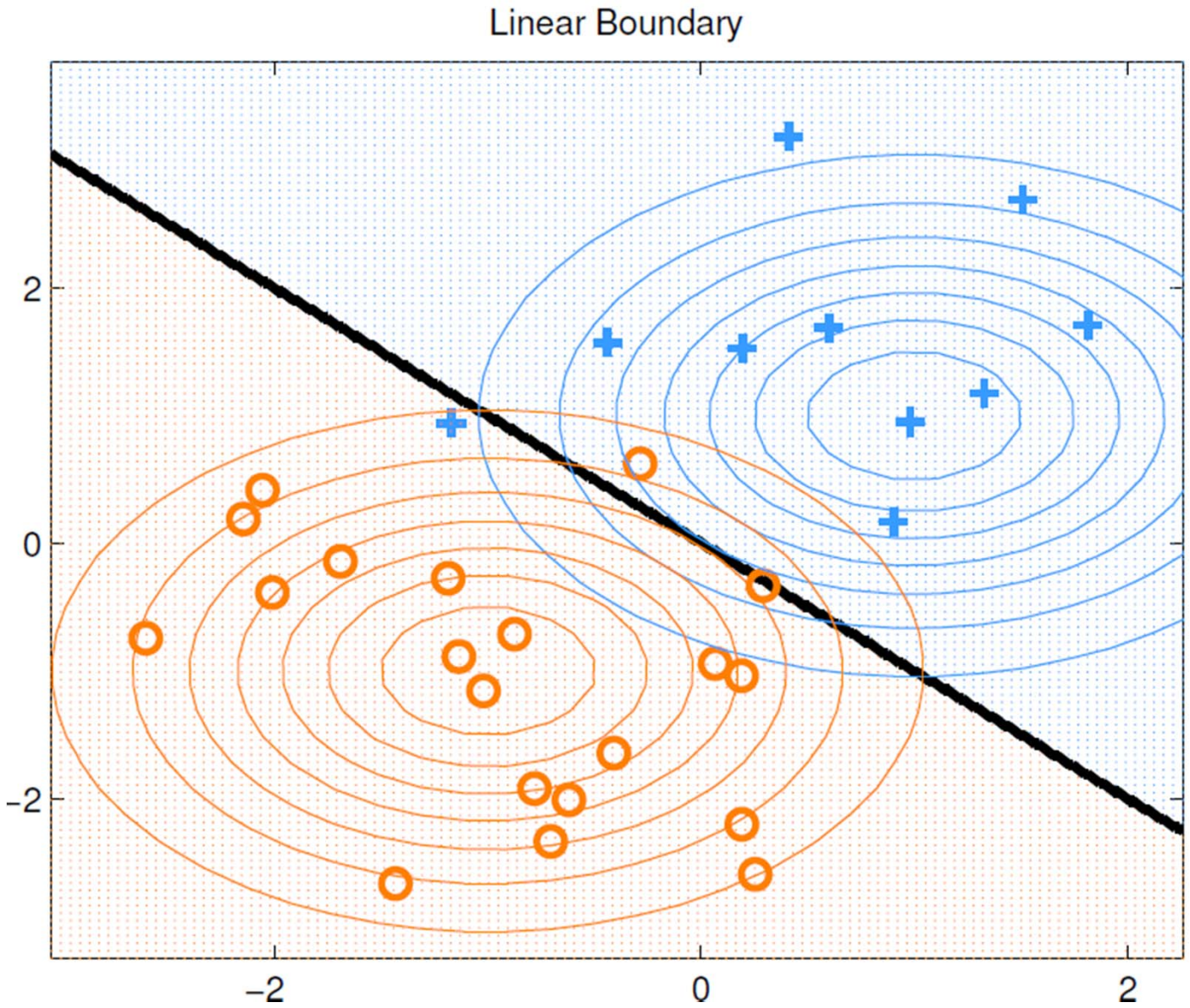
- The inner product of two vectors is proportional to the correlation between their components:

$$\mathbf{b}^\top \mathbf{x} = b_1 \cdot x_1 + \ldots + b_d \cdot x_d$$

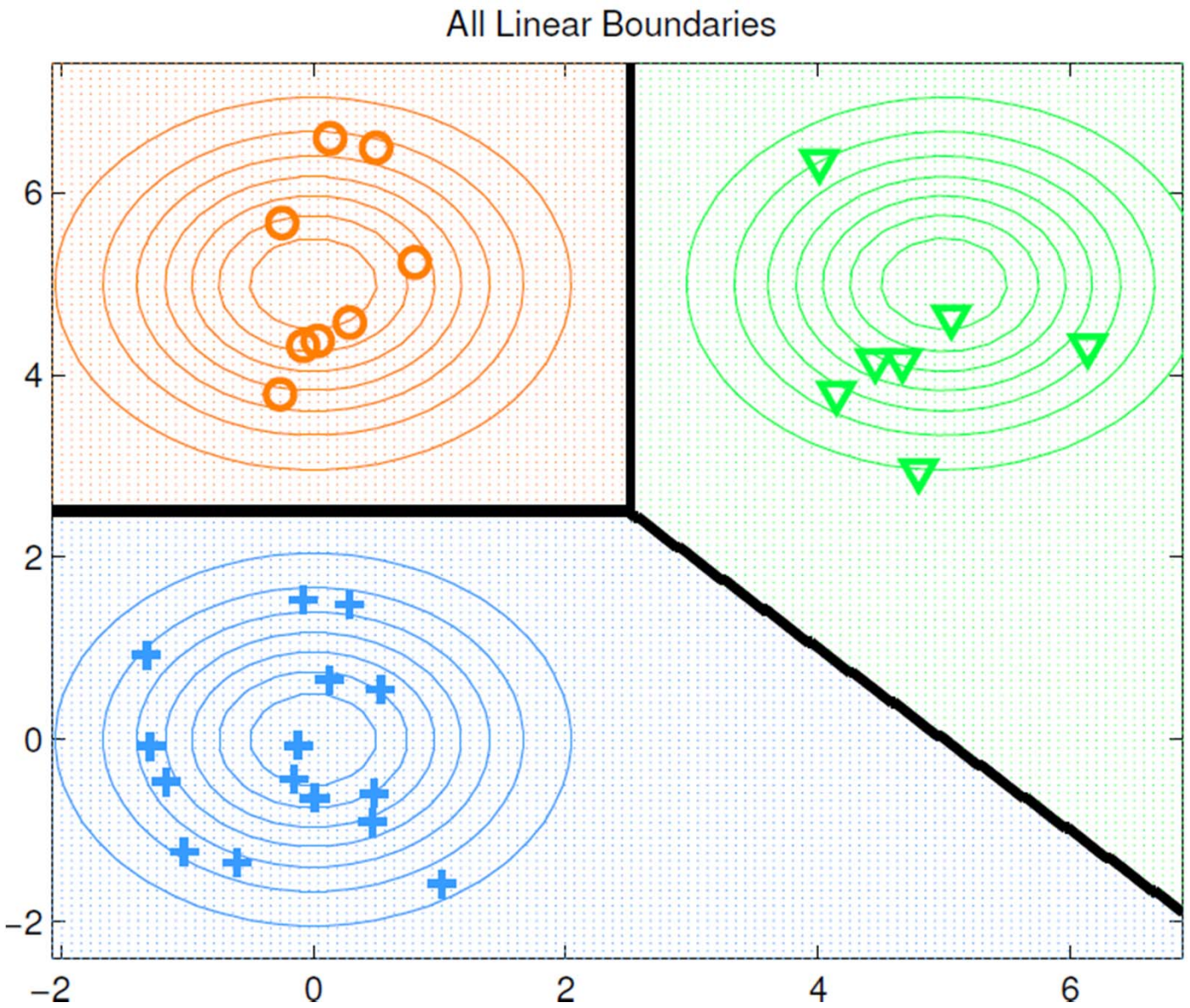
- LDA rule in words: map $\mathbf{x}$ to the class whose mean vector $\boldsymbol{\mu}_y$ is maximally correlated with it. Here, the correlation is measured by a suitably weighted and offset-adjusted correlation between the components of $\mathbf{x}$ and $\boldsymbol{\mu}_y$.

- Decision boundaries between classes always piece-wise linear (planar): equation of boundary between class i and j: $(\boldsymbol{\beta}_i - \boldsymbol{\beta}_j)^\top \mathbf{x} + (\gamma_i - \gamma_j) = 0$

LDA visualization: $m = 2$, $d = 2$



LDA visualization: $m = 3$, $d = 2$



2-class LDA

- For 2 classes (binary classification), the LDA rule simplifies a bit more.
- Classes $y = 0$ (null hypothesis) versus $y = 1$ (alternative hypothesis)

$$\boldsymbol{\beta}_y = \Sigma^{-1}\boldsymbol{\mu}_y, \quad \gamma_y = -\frac{1}{2}\boldsymbol{\mu}_y^\top \Sigma^{-1}\boldsymbol{\mu}_y + \ln p(y),$$

$$\gamma_1 - \gamma_0 = -\frac{1}{2}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)^\top \Sigma^{-1}(\boldsymbol{\mu}_1 + \boldsymbol{\mu}_0) + \big(\ln p_Y(1) - \ln p_Y(0)\big)$$

$$\mathbf{w} := (\boldsymbol{\beta}_1 - \boldsymbol{\beta}_0) = \Sigma^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)$$

$$\widetilde{\boldsymbol{\mu}} := \frac{1}{2}(\boldsymbol{\mu}_1 + \boldsymbol{\mu}_0) - (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)\frac{\big(\ln p_Y(1) - \ln p_Y(0)\big)}{(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)^\top \Sigma^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)}$$

2-class LDA

$$h_{\mathrm{LDA}}(\mathbf{x}) = \arg\max_y \left[\boldsymbol{\beta}_y^\top \mathbf{x} + \gamma_y\right]$$

$$\equiv \boldsymbol{\beta}_1^\top \mathbf{x} + \gamma_1 \underset{y=0}{\overset{y=1}{\gtrless}} \boldsymbol{\beta}_0^\top \mathbf{x} + \gamma_0$$

$$\left(\boldsymbol{\beta}_1 - \boldsymbol{\beta}_0\right)^\top \mathbf{x} + \left(\gamma_1 - \gamma_0\right) \underset{y=0}{\overset{y=1}{\gtrless}} 0$$

$$\mathbf{w}^\top \left(\mathbf{x} - \widetilde{\boldsymbol{\mu}}\right) \underset{y=0}{\overset{y=1}{\gtrless}} 0$$

$$\Rightarrow h_{\mathrm{LDA}}(\mathbf{x}) = 1\left(\mathbf{w}^\top \left(\mathbf{x} - \widetilde{\boldsymbol{\mu}}\right) > 0\right)$$

- Form of a thresholding rule: project $\mathbf{x}$ onto $\mathbf{w}$ and threshold projection with threshold τ

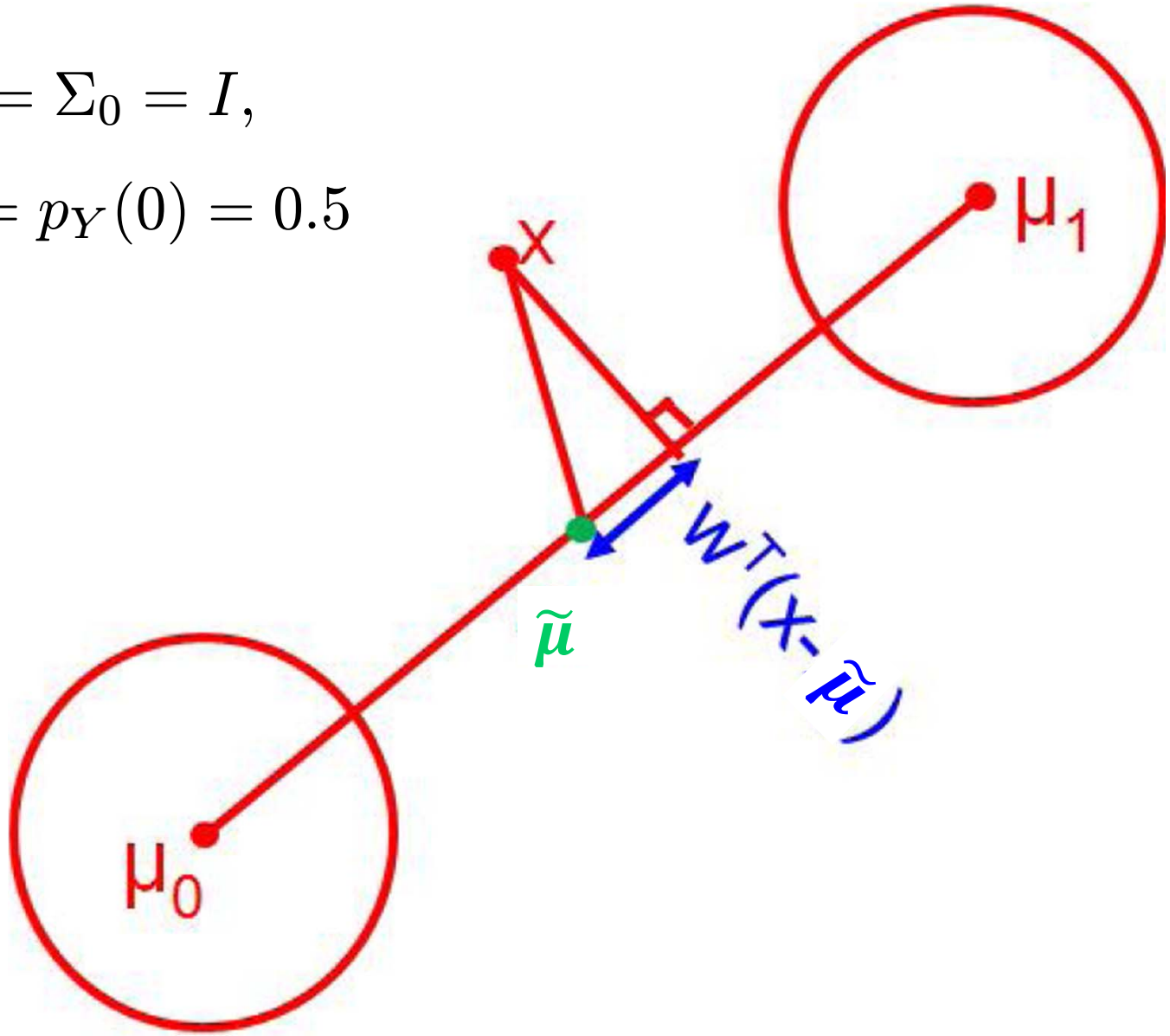
$$h_{\mathrm{LDA}}(\mathbf{x}) = 1\left(g(\mathbf{x}) > \tau\right),$$

$$g(\mathbf{x}) = \mathbf{w}^\top \mathbf{x}, \tau = \mathbf{w}^\top \widetilde{\boldsymbol{\mu}}$$

2-class LDA

$$\Sigma_1 = \Sigma_0 = I,$$

$$p_Y(1) = p_Y(0) = 0.5$$



LDA posterior distribution

- posterior pmf: $p_{\mathrm{LDA}}(y|\mathbf{x},\theta) = \dfrac{e^{\boldsymbol{\beta}_y^\top \mathbf{x}+\gamma_y}}{\sum_{j=1}^m e^{\boldsymbol{\beta}_j^\top \mathbf{x}+\gamma_j}} = \mathrm{softmax}(y|\boldsymbol{\eta})$

- where $\boldsymbol{\eta} = \left(\boldsymbol{\beta}_1^\top \mathbf{x}+\gamma_1, \ldots, \boldsymbol{\beta}_m^\top \mathbf{x}+\gamma_m\right)^\top = \left(\eta_1, \ldots, \eta_m\right)^\top$
- and $\mathrm{softmax}(y|\boldsymbol{\eta}) = \dfrac{e^{\eta_y}}{\sum_{j=1}^m e^{\eta_j}}$ behaves like the max, because

$$\lim_{T\to 0} \mathrm{softmax}(y|\boldsymbol{\eta}/T) = \begin{cases} 1/k & \text{if } y \in \arg\max_j \eta_j, k = \#\text{maximizers} \\ 0 & \text{otherwise} \end{cases}$$

- T: like "temperature" → at low temperatures, the distribution "spends most of its time" in high probability states; at high temperatures, it visits all states uniformly:

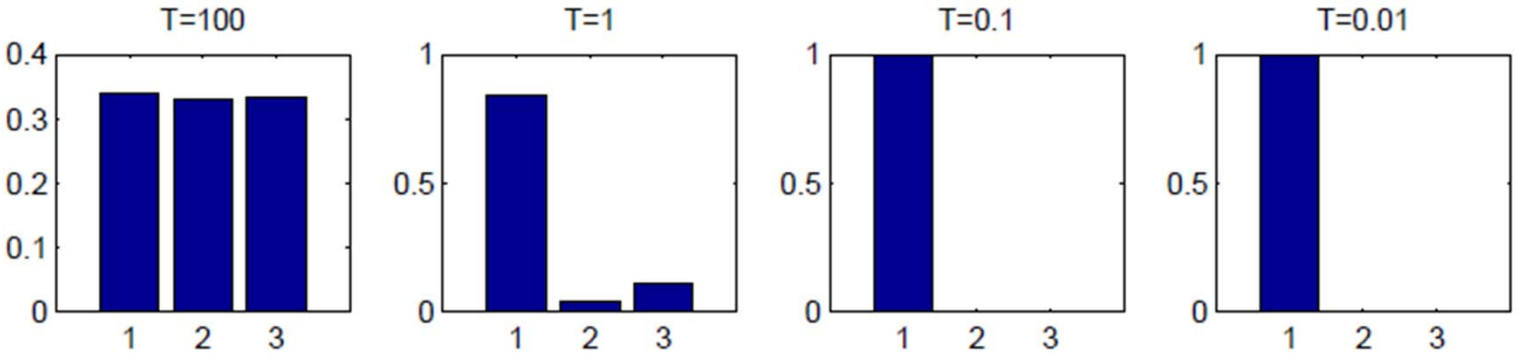
$$\lim_{T\to\infty} \mathrm{softmax}(y|\boldsymbol{\eta}/T) = \frac{1}{m} \text{ for all } y$$

Softmax function illustration

- Softmax → similar to Boltzmann distribution in Statistical Physics

$$\text{softmax}(y|\boldsymbol{\eta}) = \frac{e^{\eta_y}}{\sum_{j=1}^{m} e^{\eta_j}}$$

$$\lim_{T\to 0} \text{softmax}(y|\boldsymbol{\eta}/T) = \begin{cases} 1/k & \text{if } y \in \arg\max_j \eta_j, k = \#\text{maximizers} \\ 0 & \text{otherwise} \end{cases}$$

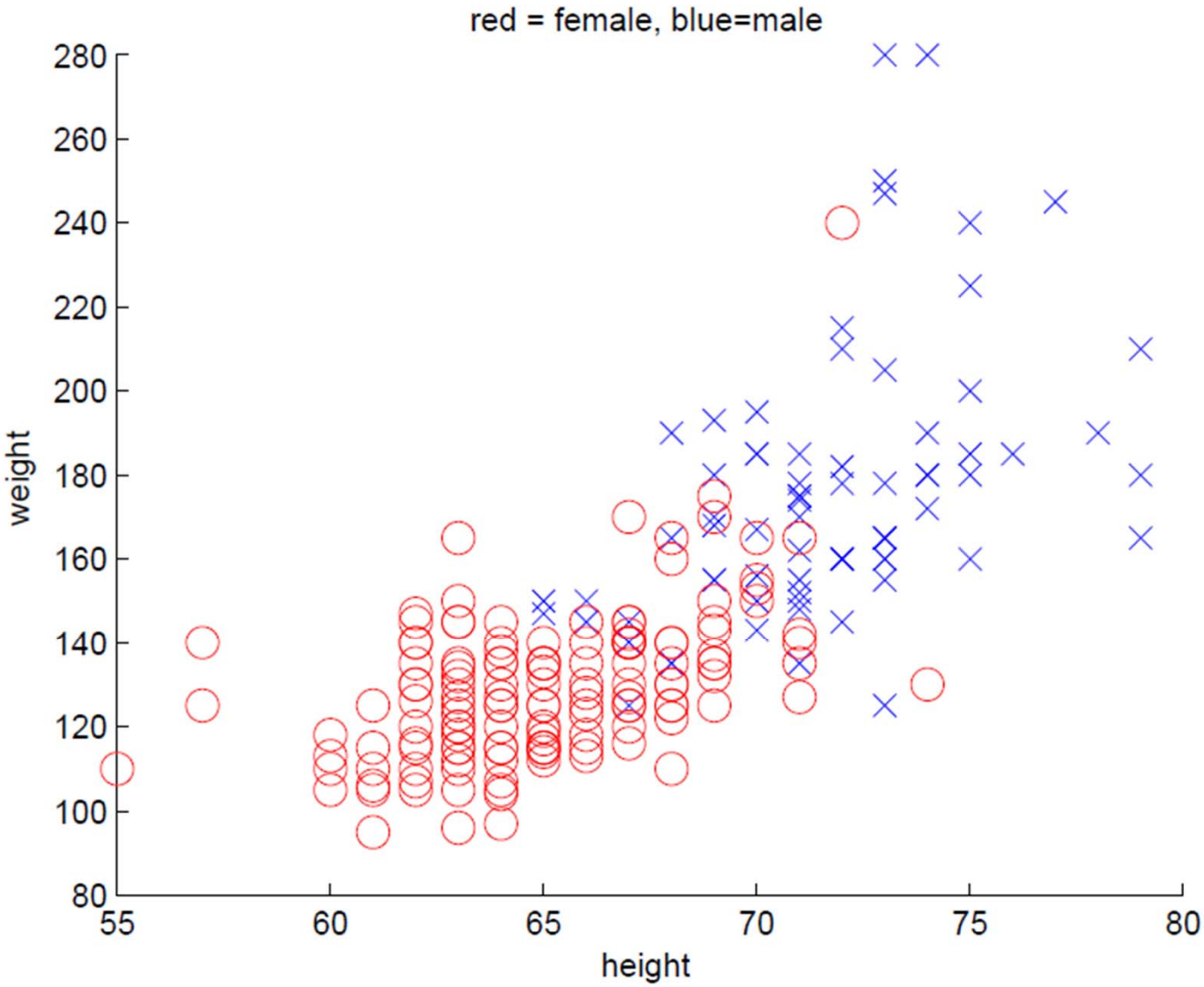


Almost Uniform (high temperature)

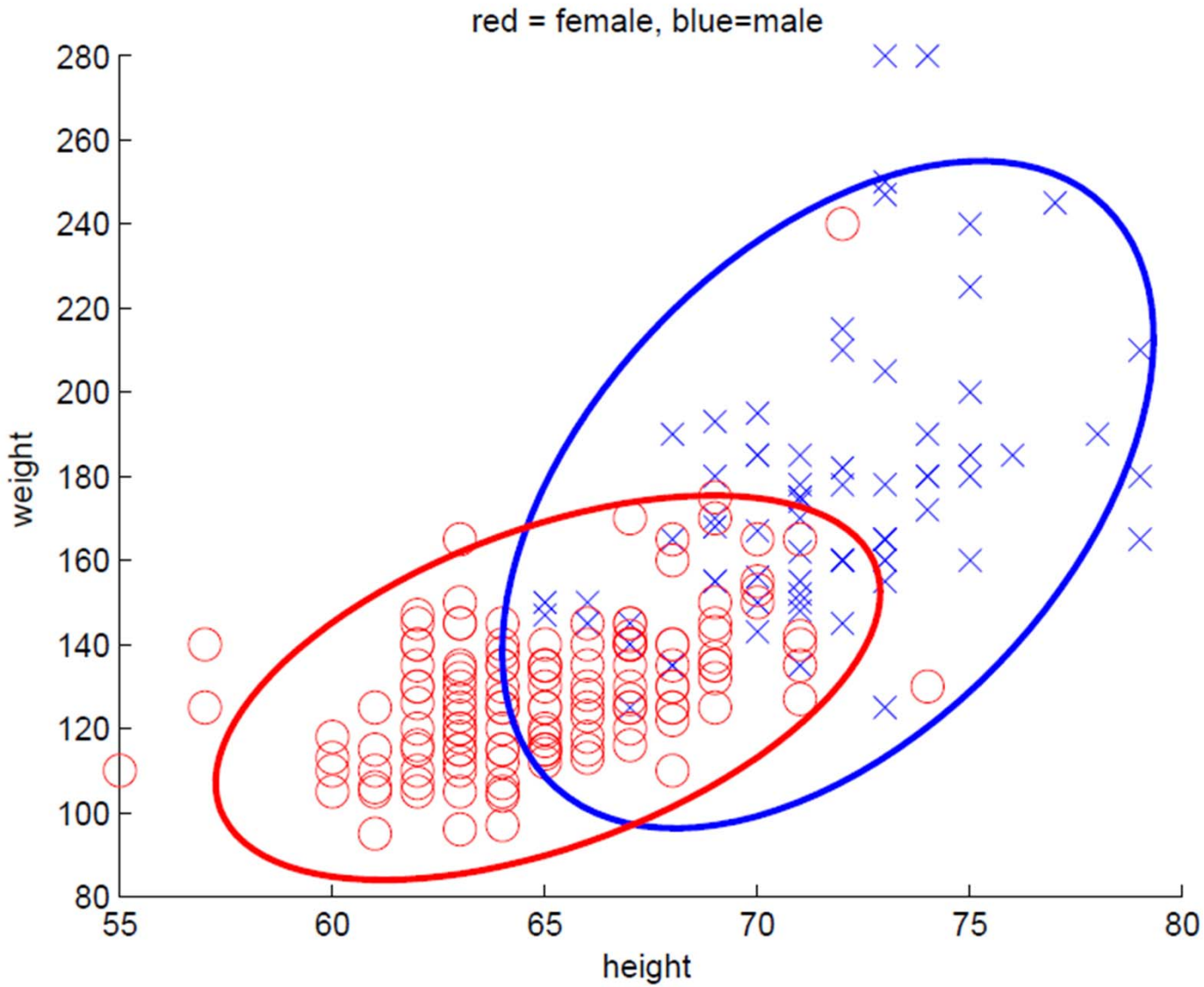
$\boldsymbol{\eta} = (3, 0, 1)^{\mathrm{T}}$

"Spiky: most of the mass is on the largest element (low temperature)

Example 2-class 2-D labeled training set



Learned 2-class Gaussian model (QDA)



Learning θ from data

- Empirical parameter estimates (intuitive): for all $y = 1,\ldots,m$

$$n_y = |\{j \in [1:n] : y_j = y\}| = \sum_{j=1}^{n} 1(y_j = y)$$

$$\widehat{p}(y) = \frac{n_y}{n}$$

$$\widehat{\boldsymbol{\mu}}_y = \frac{1}{n_y} \sum_{j \in [1:n]: y_j = y} \mathbf{x}_j$$

$$\text{QDA: } \widehat{\Sigma}_y = \frac{1}{n_y} \sum_{j \in [1:n]: y_j = y} \left(\mathbf{x}_j - \widehat{\boldsymbol{\mu}}_y\right)\left(\mathbf{x}_j - \widehat{\boldsymbol{\mu}}_y\right)^\top$$

$$\text{LDA: } \widehat{\Sigma} = \frac{1}{n} \sum_{j=1}^{n} \left(\mathbf{x}_j - \widehat{\boldsymbol{\mu}}_{y_j}\right)\left(\mathbf{x}_j - \widehat{\boldsymbol{\mu}}_{y_j}\right)^\top = \sum_{y=1}^{m} \widehat{p}(y)\widehat{\Sigma}_y$$

- Turns out that these are also ML (frequentist) estimates of θ based on data! + they are asymptotically consistent!

Proofs of ML expressions for θ

- Skipped. See Problem 2.5 in Assignment 2 + supplementary notes.

Matrix-vector expressions for emprical means and covariances

$$
\begin{aligned}
\text{Feature matrix:} \quad & \mathbb{X} &&= [\mathbf{x}_1, \ldots, \mathbf{x}_n] \\
\text{Empirical feature mean:} \quad & \widehat{\boldsymbol{\mu}}_x &&= \tfrac{1}{n}\mathbb{X}\mathbf{1}_n,\ \mathbf{1}_n = n \times 1 \text{ all ones vector} \\
\text{Centered feature:} \quad & \widetilde{\mathbf{x}}_j &&= \mathbf{x}_j - \widehat{\boldsymbol{\mu}}_x,\ j = 1, \ldots, n \\
\text{Centered feature matrix:} \quad & \widetilde{\mathbb{X}} &&= [\widetilde{\mathbf{x}}_1, \ldots, \widetilde{\mathbf{x}}_n] \\
& &&= \mathbb{X}\left(I_n - \tfrac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top\right) \\
\text{Empirical feature cov.mtx.:} \quad & \widehat{\Sigma}_x &&= \tfrac{1}{n}\widetilde{\mathbb{X}}\widetilde{\mathbb{X}}^\top \\
\text{Centered feature Gram mtx.:} \quad & &&= \widetilde{\mathbb{X}}^\top\widetilde{\mathbb{X}} \\
\text{Label matrix:} \quad & \mathbb{Y} &&= [y_1, \ldots, y_n] \\
\text{For real-valued labels,} & & & \\
\text{Emp. feature-label cross cov.mtx.:} \quad & \widehat{\Sigma}_{xy} &&= \tfrac{1}{n}\widetilde{\mathbb{X}}\widetilde{\mathbb{Y}}^\top
\end{aligned}
$$

Learning θ from data

- Expected values of parameter estimates: for all $y = 1,\ldots,m$

$$E\left[\widehat{p}(y)\right] = p(y)$$

$$E\left[\widehat{\boldsymbol{\mu}}_y \middle| y_1, \ldots, y_n, n_y > 0\right] = \boldsymbol{\mu}_y$$

$$\text{QDA: } E\left[\widehat{\Sigma}_y \middle| y_1, \ldots, y_n, n_y > 0\right] = \frac{(n_y - 1)}{n_y}\Sigma_y$$

$$\text{LDA: } E\left[\widehat{\Sigma} \middle| y_1, \ldots, y_n, n_1 > 0, \ldots, n_m > 0\right] = \frac{(n - m)}{n}\Sigma$$

- $\Rightarrow$ the covariance estimates are biased.
- Unbiased covariance estimates:
  - QDA: $\hat{\Sigma}_y \rightarrow \frac{n_y}{n_y - 1}\hat{\Sigma}_y$
  - LDA: $\hat{\Sigma} \rightarrow \frac{n}{n-m}\hat{\Sigma}$

Final learned QDA/LDA rule

- Plug-in decision rule: plug in the estimates of all the means and the covariance(s) into the Bayes decision rule, i.e., set: $\theta \rightarrow \hat{\theta}_{\mathrm{ML}}$ in the Bayes rule
- Note: QDA and LDA decision rules contain the inverse of the estimated covariance matrices
- Computational complexity (rough estimates assuming $d < n$ keeping only dominant terms):

Method	Resource	Training	Testing (per sample)
LDA	Time:	$O(nd^2+md^2+d^3)$	$O(md)$
	Memory:	$O(nd+md+d^2)$	$O(md)$
QDA	Time:	$O(nd^2+md^3)$	$O(md^2)$
	Memory:	$O(nd+md^2)$	$O(md^2)$

Regularized estimation of Σ

- if $d > n_y$, ML estimates of the covariance matrices will not be invertible (they are singular): sum of n_y rank-1 matrices can have rank at most n_y → more parameters than data

- Even when $d < n_y$, the ML covariance matrix estimates can be ill-conditioned, i.e., close to singular

- This is the overfitting problem that we have previously discussed→ the complexity of the model is greater than the amount of data

Strategies for preventing overfitting

Regularized QDA / LDA (RDA):

Key idea: Let A be a symmetric positive semi-definite matrix (i.e., a covariance matrix) which may not be invertible. Let B be any symmetric positive-definite matrix. So B is invertible. Then $C = B + A$ is a symmetric positive-definite matrix and is therefore invertible.

Proof: $$\forall z \neq 0, z^\top C z = \underbrace{z^\top B z}_{>0} + \underbrace{z^\top A z}_{\geq 0} > 0.$$

Regularization Strategy 1:

$$\widehat{\Sigma}_{\text{reg}} = \lambda I_d + (1-\lambda)\left(\widehat{\Sigma}_{\text{ML}}\right), \quad \lambda \in [0,1], \quad I_d = d \times d$$ identity matrix, $\lambda =$ tuning parameter set by cross validation (controls degree of regularization)

Strategies for preventing overfitting

Regularized QDA / LDA (RDA):

Observation: Without loss of generality (w.l.o.g.) we can assume that all diagonal elements of the ML estimate of the feature covariance matrix (the empirical covariance) are positive, i.e., $\forall i,\ \hat{\Sigma}_{ML}(i,i) > 0$, because otherwise it would imply that some feature is constant across all training examples. If so, then that feature is essentially "useless" for decision making and can be removed from the very beginning. Thus, w.l.o.g.

$$\operatorname{diag}\left(\widehat{\Sigma}_{\mathrm{ML}}\right) = \begin{pmatrix} \widehat{\Sigma}_{\mathrm{ML}}(1,1) & 0 & \cdots & 0 \\ 0 & \widehat{\Sigma}_{\mathrm{ML}}(2,2) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \widehat{\Sigma}_{\mathrm{ML}}(d,d) \end{pmatrix} > 0$$

and is therefore invertible.

Strategies for preventing overfitting

Regularization Strategy 2:

$$\widehat{\Sigma}_{\text{reg}} = \lambda \text{diag}\left(\widehat{\Sigma}_{\text{ML}}\right) + (1-\lambda)\left(\widehat{\Sigma}_{\text{ML}}\right), \quad \lambda \in [0,1]$$

Here, λ controls the degree of regularization and can be set via cross validation.

It can be proved that this is the MAP (Bayesian) estimate of Σ using an inverse-Wishart prior $\pi(\Sigma)$ for Σ with tuning parameter λ. The proof of this result is outside the scope of this class.

Special case with $\lambda = 1$: $\widehat{\Sigma}_{\text{reg}} = \text{diag}\left(\widehat{\Sigma}_{\text{ML}}\right)$. Equivalent to assuming (naïvely) that all d components are independent even if they might not be → "Gaussian Naïve Bayes"

Fisher's LDA for 2 classes (m = 2)

- Similar to LDA
- Gaussian assumption not made
- But the optimality criterion is changed
- Model:

$$\text{Given} \quad \underbrace{Y = y}_{P(Y=y)=p_y} \;,\; X = \underbrace{\boldsymbol{\mu}_y}_{\text{“signal ”}} + \underbrace{Z_y}_{\text{zero-mean “noise” with cov } \Sigma_y}$$

- Idea: project $X = \mathbf{x}$ along a direction $\mathbf{w}$ and threshold the projection $\mathbf{w}^T\mathbf{x}$ to make class decision
- What is the best direction?
- How to choose the threshold?

Fisher's LDA for 2 classes (m = 2)

$$\text{Given} \quad \underbrace{Y = y}_{P(Y=y)=p_y} \;,\; X = \underbrace{\boldsymbol{\mu}_y}_{\text{“signal ”}} + \underbrace{Z_y}_{\text{zero-mean “noise” with cov } \Sigma_y}$$

- Given $Y = y$, the projection $\mathbf{w}^\mathsf{T}X$ is a scalar random variable with mean $\mathbf{w}^\mathsf{T}\boldsymbol{\mu}_y$ and variance $\mathbf{w}^\mathsf{T}\Sigma_y\mathbf{w}$
- Define the between-class variance or "signal power" of the projection as:

$$\sigma^2_{\text{between}}(\mathbf{w}) = (\mathbf{w}^\mathsf{T}\boldsymbol{\mu}_1 - \mathbf{w}^\mathsf{T}\boldsymbol{\mu}_0)^2$$

- Define the within-class variance or "noise power" of the projection as:

$$\sigma^2_{\text{within}}(\mathbf{w}) = p_0\mathbf{w}^\mathsf{T}\Sigma_0\mathbf{w} + p_1\mathbf{w}^\mathsf{T}\Sigma_1\mathbf{w} = \mathbf{w}^\mathsf{T}(p_0\Sigma_0 + p_1\Sigma_1)\mathbf{w}$$

- Measure of separation between classes:

$$SNR(\mathbf{w}) = \frac{\sigma^2_{\text{between}}(\mathbf{w})}{\sigma^2_{\text{within}}(\mathbf{w})} = \frac{(\mathbf{w}^\mathsf{T}\boldsymbol{\mu}_1 - \mathbf{w}^\mathsf{T}\boldsymbol{\mu}_0)^2}{p_0\mathbf{w}^\mathsf{T}\Sigma_0\mathbf{w} + p_1\mathbf{w}^\mathsf{T}\Sigma_1\mathbf{w}}$$

loosely called signal-to-noise ratio (SNR)

Fisher's LDA for 2 classes (m = 2)

$$\text{Given} \quad \underbrace{Y = y}_{P(Y=y)=p_y} \;,\; X = \underbrace{\boldsymbol{\mu}_y}_{\text{“signal ”}} + \underbrace{Z_y}_{\text{zero-mean “noise” with cov } \Sigma_y}$$

- Best $\mathbf{w}$? One which maximizes the SNR:

$$\mathbf{w}_{\text{Fisher}} = \arg\max_{\mathbf{w}} SNR(\mathbf{w}) = \arg\max_{\mathbf{w}} \frac{\left(\mathbf{w}^\top(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)\right)^2}{\mathbf{w}^\top \Sigma \mathbf{w}}$$

  where $\Sigma = (p_0\Sigma_0 + p_1\Sigma_1)$

- Main result: If Σ is invertible, then

$$\mathbf{w}_{\text{Fisher}} = \Sigma^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)$$

- Expression matches $\mathbf{w}$ in LDA
- As in LDA, in practice, all parameters will be estimated from data via empirical averages: means, covariances (may need regularization), and class prevalences:

$$\widehat{\mathbf{w}}_{\text{Fisher}} = \left(\hat{p}_0\hat{\Sigma}_0 + \hat{p}_1\hat{\Sigma}_1\right)^{-1}(\widehat{\boldsymbol{\mu}_1} - \widehat{\boldsymbol{\mu}_0})$$

Fisher's LDA for 2 classes (m = 2)

$$\text{Given} \quad \underbrace{Y = y}_{P(Y=y)=p_y} \;,\; X = \underbrace{\boldsymbol{\mu}_y}_{\text{“signal ”}} + \underbrace{Z_y}_{\text{zero-mean “noise” with cov } \Sigma_y}$$

- Final decision rule: $h_{\text{Fisher}}(\mathbf{x}) = 1(\widehat{\mathbf{w}}_{\text{Fisher}}{}^{\top}\mathbf{x} > \tau)$

- Threshold τ can be chosen as in LDA, but in practice it is treated as a tuning parameter and its value is set via a cross-validation analysis

Fisher's LDA for 2 classes (m = 2)

- Main result: If Σ is invertible, then

$$\mathbf{w}_{\text{Fisher}} = \Sigma^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0) = \arg\max_{\mathbf{w}} \frac{\left(\mathbf{w}^\top(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)\right)^2}{\mathbf{w}^\top \Sigma \mathbf{w}}$$

where $\Sigma = (p_0\Sigma_0 + p_1\Sigma_1)$

Proof: Define $\mathbf{v} = \sqrt{\Sigma}\mathbf{w}$. Then,

$$\frac{\left(\mathbf{w}^\top(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)\right)^2}{\mathbf{w}^\top \Sigma \mathbf{w}} = \frac{\left(\mathbf{w}^\top \sqrt{\Sigma}\sqrt{\Sigma}\Sigma^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)\right)^2}{\mathbf{w}^\top \sqrt{\Sigma}\sqrt{\Sigma}\mathbf{w}} = \frac{\left(\mathbf{v}^\top \sqrt{\Sigma}\mathbf{w}_{\text{Fisher}}\right)^2}{\mathbf{v}^\top \mathbf{v}} = \frac{\left|\langle \mathbf{v}, \sqrt{\Sigma}\mathbf{w}_{\text{Fisher}}\rangle\right|^2}{\|\mathbf{v}\|^2}$$

$$\leq \frac{\|\mathbf{v}\|^2 \left\|\sqrt{\Sigma}\mathbf{w}_{\text{Fisher}}\right\|^2}{\|\mathbf{v}\|^2} = \left\|\sqrt{\Sigma}\mathbf{w}_{\text{Fisher}}\right\|^2$$

where in the last-but-one step we used the Cauchy-Schwartz inequality. Equality can be attained by setting $\mathbf{v} = \sqrt{\Sigma}\mathbf{w}_{\text{Fisher}}$, i.e., $\mathbf{w} = \mathbf{w}_{\text{Fisher}}$