EN3150 Pattern Recognition
Classification
Part 02

M. T. U. Sampath K. Perera,
Department of Electronic and Telecommunication Engineering,
University of Moratuwa.
(sampathk@uom.lk).
Semester 5 – Batch 20.

Electronic and Telecommunication Engineering

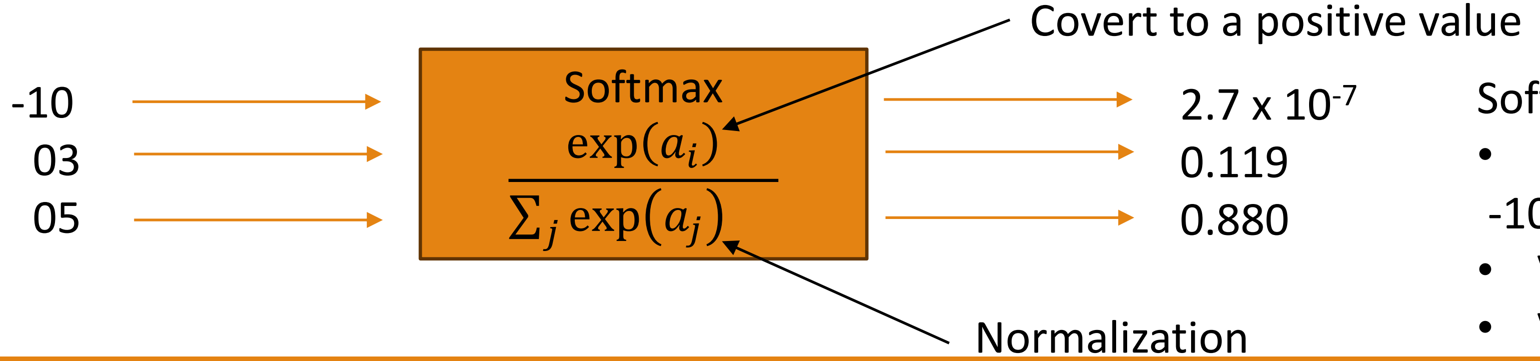
Multinomial logistic regression

- Also known as multiclass logistic regression
- For multiple K classes, there are K linear functions

$$p(y_i = C_k|\boldsymbol{x_i}, \mathbf{W}) = \frac{\exp(w_{0k}+w_{1k}x_{1,i}+\cdots+w_{Dk}x_{D,i})}{\sum_j \exp(w_{0k}+w_{1k}x_{1,i}+\cdots+w_{Dk}x_{D,i})} = \frac{\exp(\boldsymbol{w}_k^T\boldsymbol{x}_i)}{\sum_j \exp(\boldsymbol{w}_j^T\boldsymbol{x}_i)}$$

Softmax transformation

- $\mathbf{W}$ is K x (D+1) weight matrix and D is no of features.
- $p(y_i|x_i, \mathbf{W}) = \text{Cat}\,(y_i|\text{softmax}\,(\widetilde{\boldsymbol{W}}\boldsymbol{x}_i + \boldsymbol{b}))$. Here, $\boldsymbol{b} = [w_{01}, \ldots, w_{0K}]^{\boldsymbol{T}}$ is a K length vector*.



SoftMax

- Preserves the order of the input

$-10 < 03 < 05$ and $2.7 \times 10^{-7} < 0.119 < 0.880$

- Values are between 0 and 1
- Values sum up to 1

* $\boldsymbol{b}$ can be added to first column by considering dummy feature equal to 1, $\mathbf{W}= [\boldsymbol{b}\ \widetilde{\boldsymbol{W}}]$.

Multinomial logistic regression

$$\mathbf{C} = \begin{bmatrix} p(y_i = C_1 | x_i, \mathbf{W}) \\ \\ p(y_i = C_2 | x_i, \mathbf{W}) \\ . \\ . \\ . \\ \\ p(y_i = C_K | x_i, \mathbf{W}) \end{bmatrix} = \begin{bmatrix} \frac{\exp(\boldsymbol{w}_1^T \boldsymbol{x}_i)}{\sum_j \exp\left(\boldsymbol{w}_j^T \boldsymbol{x}_i\right)} \\ \frac{\exp(\boldsymbol{w}_2^T \boldsymbol{x}_i)}{\sum_j \exp\left(\boldsymbol{w}_j^T \boldsymbol{x}_i\right)} \\ . \\ . \\ . \\ \frac{\exp(\boldsymbol{w}_K^T \boldsymbol{x}_i)}{\sum_j \exp\left(\boldsymbol{w}_j^T \boldsymbol{x}_i\right)} \end{bmatrix}$$

Softmax transformation

Multinomial logistic regression

- How to learn weights? maximum likelihood estimation
- Let $a_{ik} = \boldsymbol{w}_k^T \boldsymbol{x}_i$ and $p(y_{ik} = 1 | \boldsymbol{x}_i, \boldsymbol{W}) = \tilde{y}_{ik}$
- Likelihood function

$$L(\boldsymbol{W}) = \prod_{i=1}^{N} \prod_{k=1}^{K} p(y_i = C_k | \boldsymbol{x}_i, \boldsymbol{W})^{c_{ik}} = \prod_{i=1}^{N} \prod_{k=1}^{K} \tilde{y}_{ik}{}^{c_{ik}}$$

- Negative likelihood function

$$NLL(\boldsymbol{W}) = - \sum_{i=1}^{N} \sum_{k=1}^{K} c_{ik} \log(\tilde{y}_{ik})$$

cross-entropy error function for the multiclass classification

Multinomial logistic regression

$$\begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_K \end{bmatrix} \Rightarrow \boxed{\text{Softmax}} \Rightarrow \begin{bmatrix} \tilde{y}_1 \\ \tilde{y}_2 \\ \vdots \\ \tilde{y}_K \end{bmatrix}$$

- Jacobian matrix

$$a_k = \boldsymbol{w}_k^T \boldsymbol{x}_i \qquad \tilde{y}_k = \frac{\exp(\boldsymbol{w}_k{}^T \boldsymbol{x}_i)}{\sum_j \exp(\boldsymbol{w}_j{}^T \boldsymbol{x}_i)} = \frac{\exp(a_k)}{\sum_j \exp(a_j)} \qquad \frac{\partial \tilde{y}_k}{\partial a_j} = \tilde{y}_k\left(\mathbb{I}(k=j) - \tilde{y}_j\right)$$

$$J = \begin{bmatrix} \frac{\partial \tilde{y}_1}{\partial a_1} & \frac{\partial \tilde{y}_1}{\partial a_2} & \cdots & \frac{\partial \tilde{y}_1}{\partial a_K} \\ \frac{\partial \tilde{y}_2}{\partial a_1} & \frac{\partial \tilde{y}_2}{\partial a_2} & \cdots & \frac{\partial \tilde{y}_2}{\partial a_K} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial \tilde{y}_j}{\partial a_1} & \frac{\partial \tilde{y}_j}{\partial a_2} & \cdots & \frac{\partial \tilde{y}_j}{\partial a_K} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial \tilde{y}_K}{\partial a_1} & \frac{\partial \tilde{y}_K}{\partial a_2} & \cdots & \frac{\partial \tilde{y}_K}{\partial a_K} \end{bmatrix}$$

For 3 classes Jacobian matrix is given by

$$\mathbf{J} = \begin{bmatrix} \frac{\partial \tilde{y}_1}{\partial a_1} & \frac{\partial \tilde{y}_1}{\partial a_2} & \frac{\partial \tilde{y}_1}{\partial a_3} \\ \frac{\partial \tilde{y}_2}{\partial a_1} & \frac{\partial \tilde{y}_2}{\partial a_2} & \frac{\partial \tilde{y}_2}{\partial a_3} \\ \frac{\partial \tilde{y}_3}{\partial a_1} & \frac{\partial \tilde{y}_3}{\partial a_2} & \frac{\partial \tilde{y}_3}{\partial a_3} \end{bmatrix} = \begin{bmatrix} \tilde{y}_1(1-\tilde{y}_1) & -\tilde{y}_1\tilde{y}_2 & -\tilde{y}_1\tilde{y}_3 \\ -\tilde{y}_2\tilde{y}_1 & \tilde{y}_2(1-\tilde{y}_2) & -\tilde{y}_2\tilde{y}_3 \\ -\tilde{y}_3\tilde{y}_1 & -\tilde{y}_3\tilde{y}_2 & \tilde{y}_3(1-\tilde{y}_3) \end{bmatrix}$$

$$\mathbf{J} = \frac{\partial \tilde{\boldsymbol{y}}}{\partial \boldsymbol{a}} = (\tilde{\boldsymbol{y}}\mathbf{1}^{\mathsf{T}}) \odot (\mathbf{I} - \mathbf{1}\tilde{\boldsymbol{y}}^{\mathsf{T}})$$

When $k = j$, $\mathbb{I}(k=j)$ =1 else 0. $\tilde{\boldsymbol{y}}\mathbf{1}^{\mathsf{T}}$ copies $\tilde{\boldsymbol{y}}$ across each column, and $\mathbf{1}\tilde{\boldsymbol{y}}^{\mathsf{T}}$ copies $\tilde{\boldsymbol{y}}$ across each row. **I**- identity matrix

Multinomial logistic regression

$$NLL(\boldsymbol{W}) = - \sum_{i=1}^{N} \sum_{k=1}^{K} c_{ik} \log(\tilde{y}_{ik})$$

$$\tilde{y}_{ik} = \frac{\exp(\boldsymbol{w}_k{}^T \boldsymbol{x}_i)}{\sum_j \exp(\boldsymbol{w}_j{}^T \boldsymbol{x}_i)} = \frac{\exp(a_{ik})}{\sum_j \exp(a_{ij})}$$

$$\frac{\partial \tilde{y}_k}{\partial a_j} = \tilde{y}_k (\mathbb{I}(k = j) - \tilde{y}_j)$$

$$a_{ik} = \boldsymbol{w}_k^T \boldsymbol{x}_i$$

- Consider i-th data sample

$$\nabla \boldsymbol{w}_k NLL(\boldsymbol{W})_i = - \sum_{k=1}^{K} \frac{\partial NLL(\boldsymbol{W})_i}{\partial \tilde{y}_{ik}} \frac{\partial \tilde{y}_{ik}}{\partial a_{ik}} \frac{\partial a_{ik}}{\partial \boldsymbol{w}_k}$$

$$= - \sum_{k=1}^{K} c_{ik} \frac{\tilde{y}_{ik}}{\tilde{y}_{ik}} \tilde{y}_k (\mathbb{I}(k = i) - \tilde{y}_{ik}) \boldsymbol{x}_i \quad = \sum_{k=1}^{K} (\tilde{y}_{ik} - c_{ik}) \boldsymbol{x}_i$$

- For all N data samples and all K classes

$$\mathrm{g}(\boldsymbol{W}) = \frac{1}{N} \sum_{i=1}^{N} \boldsymbol{x}_i (\tilde{\boldsymbol{y}}_i - \boldsymbol{c}_i)^T$$

1x K vector

(D+1) x 1 vector

(D+1) × K matrix

Multinomial logistic regression

- Stochastic gradient descent
- Update of weight (i is the sample index)

$$\mathrm{g}(\boldsymbol{W}) = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{x}_i(\widetilde{\boldsymbol{y}}_i - \boldsymbol{c}_i)^{\boldsymbol{T}}$$

$$\boldsymbol{W}_{i+1} \leftarrow \boldsymbol{W}_i - \alpha\boldsymbol{x}_i(\widetilde{\boldsymbol{y}}_i - \boldsymbol{c}_i)^{\boldsymbol{T}}$$

- Batch gradient descent

$$\boldsymbol{W}_{\boldsymbol{new}} \leftarrow \boldsymbol{W}_{\boldsymbol{old}} - \alpha\frac{1}{N}\sum_{i=1}^{N} \boldsymbol{x}_i(\widetilde{\boldsymbol{y}}_i - \boldsymbol{c}_i)^{\boldsymbol{T}}$$

Multinomial logistic regression

- Hessian of the NLL for multinomial logistic regression is given by

$$H(\boldsymbol{W}) = \frac{1}{N}\sum_{i=1}^{N} \left(\operatorname{diag}(\tilde{\boldsymbol{y}}_i) - \tilde{\boldsymbol{y}}_i\tilde{\boldsymbol{y}}_{\boldsymbol{i}}^{\boldsymbol{T}}\right) \otimes \boldsymbol{x}_i \, \boldsymbol{x}_{\boldsymbol{i}}^{\boldsymbol{T}}$$

- To develop a batch algorithm for the multiclass problem, we use the Newton-Raphson update.
- The IRLS algorithm involves evaluating the Hessian matrix

$\otimes$ Kronecker product

Probabilistic view of classification

- Probabilistic view of classification
  - **Discriminative classifier**
    - Directly fit the class posterior $p(y_i = C_k|\boldsymbol{x_i}, \boldsymbol{\theta})$
    - E.g., logistic regression, multi class logistic regression
  - **Generative classifier**
    - Model how to generate data using the conditional density $p(\boldsymbol{x_i}|y_i = Ck)$ and class priority $p(y_i = C_k)$. Then using Bayes rule

$$p(y = C_k|\mathbf{x}, \boldsymbol{\theta}) = \frac{p(y = Ck|\boldsymbol{\theta})p(\mathbf{x}|y = C_k, \boldsymbol{\theta})}{\sum_{C_k'} p(y = C_k'|\boldsymbol{\theta})p(\mathbf{x}|y = C_k', \boldsymbol{\theta})}$$

Generative models

generate the features $\boldsymbol{x}$
for each class

- Generative classifier

$$p(y = C_k|\mathbf{x}, \boldsymbol{\theta}) = \frac{p(y = C_k|\boldsymbol{\theta})p(\boldsymbol{x}|y = C_k, \boldsymbol{\theta})}{\sum_{C_k'} p(y = C_k'|\boldsymbol{\theta})p(\boldsymbol{x}|y = C_k', \boldsymbol{\theta})}$$

For two classes

$$p(y = C_1|\mathbf{x}, \boldsymbol{\theta}) = \frac{p(y = C_1|\boldsymbol{\theta})p(\boldsymbol{x}|y = C_1, \boldsymbol{\theta})}{p(y = C_1|\boldsymbol{\theta})p(\boldsymbol{x}|y = C_1, \boldsymbol{\theta}) + p(y = C_2|\boldsymbol{\theta})p(\boldsymbol{x}|y = C_2, \boldsymbol{\theta})}$$

$$= \frac{\mathbf{1}}{1+p(y = C_2|\boldsymbol{\theta})p(\boldsymbol{x}|y = C_2, \boldsymbol{\theta})/p(y = C_1|\boldsymbol{\theta})p(\boldsymbol{x}|y = C_1, \boldsymbol{\theta})} = \frac{\mathbf{1}}{1+\exp(-a(\boldsymbol{x}))}$$

$$= \mathrm{sigm}(a(\boldsymbol{x})) = \sigma(a(\boldsymbol{x}))$$

$$a(\boldsymbol{x}) = \frac{p(y = C_1|\boldsymbol{\theta})p(\boldsymbol{x}|y = C_1, \boldsymbol{\theta})}{p(y = C_2|\boldsymbol{\theta})p(\boldsymbol{x}|y = C_2, \boldsymbol{\theta})}$$

equivalent form of the posterior probabilities

logistic sigmoid function

squashing function

For multi-calss problem it is softmax function

*Although names says discriminant this is a generative model

Generative models

- Generative classifier

$$p(y = C_k|\mathbf{x}, \boldsymbol{\theta}) = \frac{p(y = C_k|\boldsymbol{\theta})p(\boldsymbol{x}|y = C_k, \boldsymbol{\theta})}{\sum_{C_k{}'} p(y = C_k{}'|\boldsymbol{\theta})p(\boldsymbol{x}|y = C_k{}', \boldsymbol{\theta})}$$

generate the features $\boldsymbol{x}$ for each class

- For two classes

$$p(y = C_1|\mathbf{x}, \boldsymbol{\theta}) = \frac{p(y = C_1|\boldsymbol{\theta})p(\boldsymbol{x}|y = C_1, \boldsymbol{\theta})}{p(y = C_1|\boldsymbol{\theta})p(\boldsymbol{x}|y = C_1, \boldsymbol{\theta}) + p(y = C_2|\boldsymbol{\theta})p(\boldsymbol{x}|y = C_2, \boldsymbol{\theta})}$$

- Linear discriminant analysis* $\log p(\boldsymbol{x}|y = C_k, \boldsymbol{\theta}) = \tilde{\boldsymbol{w}}^T\boldsymbol{x} + \text{const}$ (linear function of $\boldsymbol{x}$)

Generative models: Continuous inputs

- Class-conditional densities are Gaussian distributed

$$p(\boldsymbol{x}|y = Ck, \boldsymbol{\theta}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \Sigma_k) \triangleq \frac{1}{(2\pi)^{D/2}|\Sigma_k|^{1/2}} \exp\left[-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_k)^T {\Sigma_k}^{-1}(\mathbf{x} - \boldsymbol{\mu}_k)\right]$$

Multi–Variate Gaussian

$\boldsymbol{\mu}_k$ = mean vector

Σ_k= covariance matrix

- Assume that **same covariance matrix** is shared with all classes

$$p(\boldsymbol{x}|y = Ck, \boldsymbol{\theta}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \Sigma) \triangleq \frac{1}{(2\pi)^{D/2}|\Sigma|^{1/2}} \exp\left[-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_k)^T \Sigma^{-1}(\mathbf{x} - \boldsymbol{\mu}_k)\right]$$

Generative models: Continuous inputs

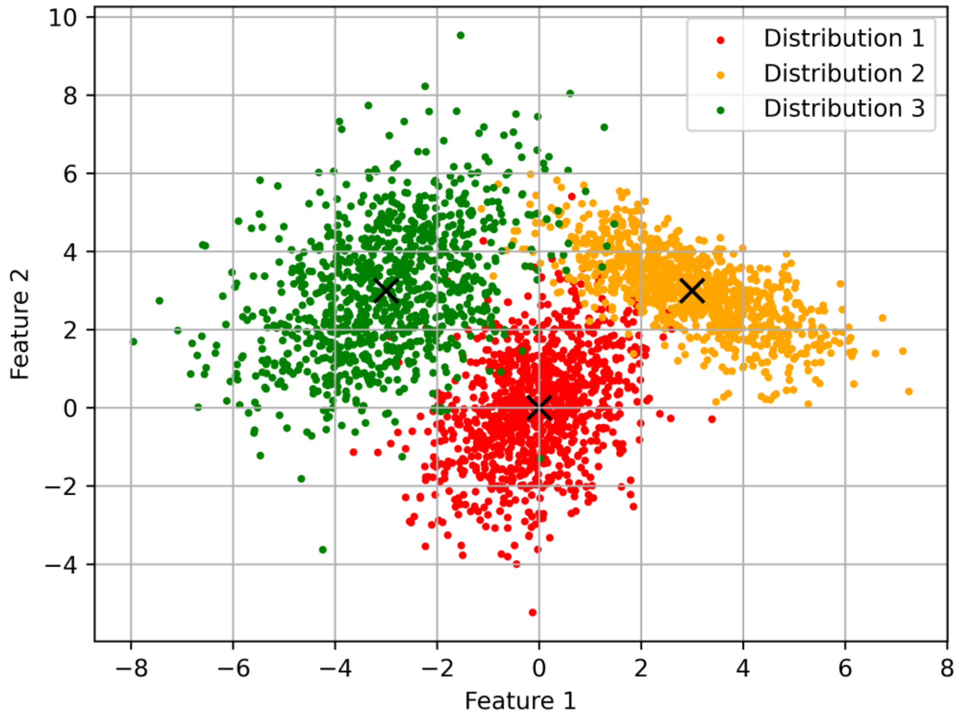
- Class-conditional densities are Gaussian distributed

$$p(\boldsymbol{x}|y = C_k, \boldsymbol{\theta}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \Sigma) \triangleq \frac{1}{(2\pi)^{D/2}|\Sigma|^{1/2}} \exp\left[-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)^T \Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu}_k)\right]$$

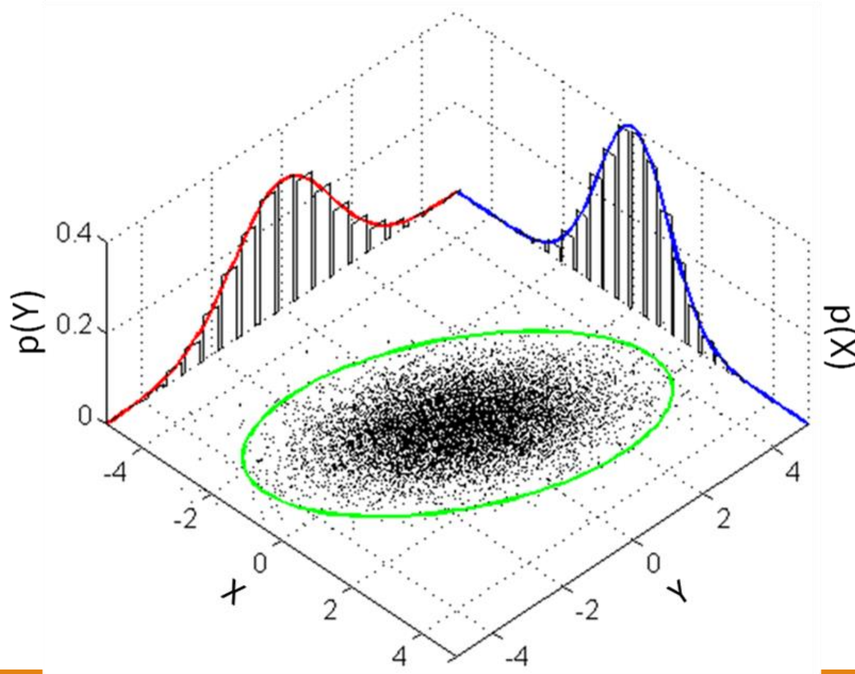
Multi–Variate Gaussian

$\boldsymbol{\mu}_k$ = mean vector

Σ= covariance matrix

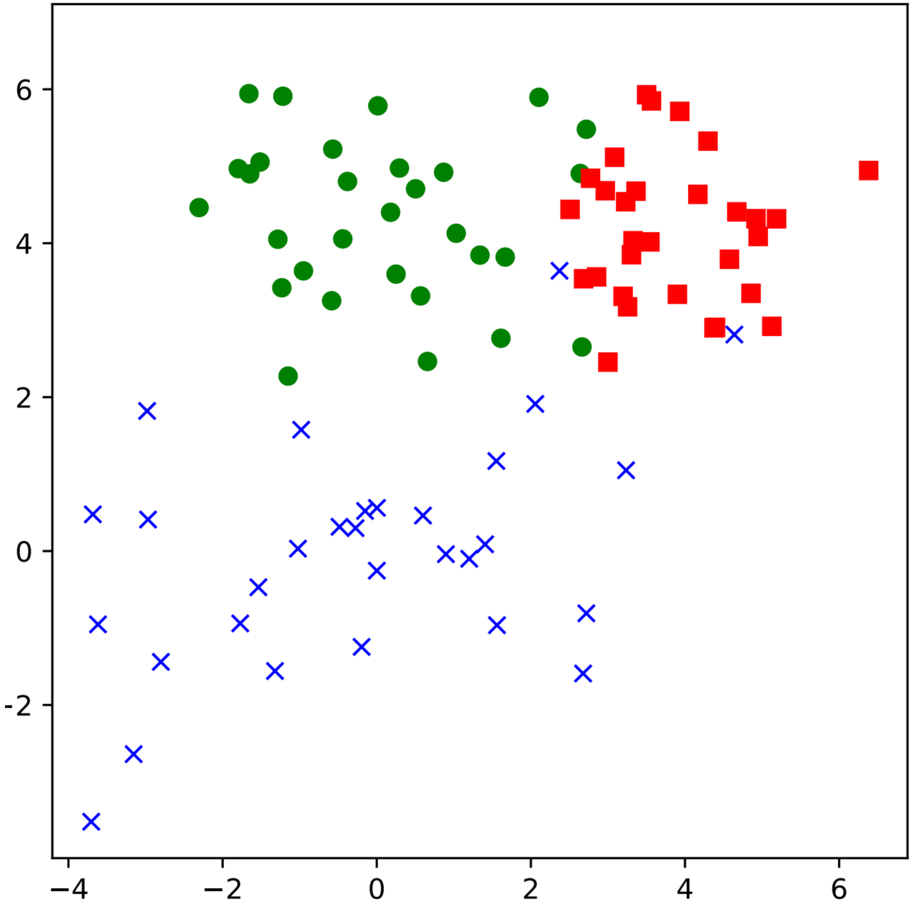


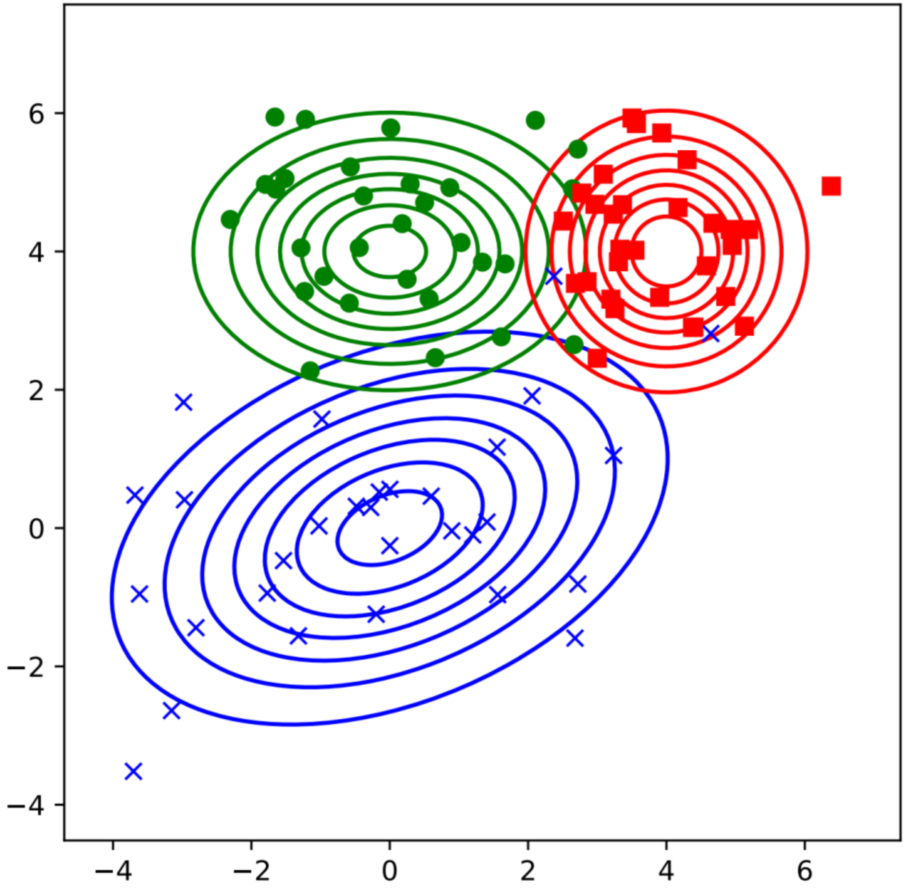
$$\mu_1 = \begin{bmatrix}0\\0\end{bmatrix}, C_1 = \begin{bmatrix}1 & 0.5\\0.5 & 2\end{bmatrix}$$
$$\mu_2 = \begin{bmatrix}3\\3\end{bmatrix}, C_2 = \begin{bmatrix}2 & -1\\-1 & 1\end{bmatrix}$$
$$\mu_3 = \begin{bmatrix}-3\\3\end{bmatrix}, C_3 = \begin{bmatrix}2 & 1\\1 & 3\end{bmatrix}$$



MultivariateNormal - Multivariate normal distribution - Wikipedia

Generative models: Continuous inputs





Kevin P. Murphy, *Machine Learning: A Probabilistic Perspective*, 2012.
probml.github.io/notebooks#discrim_analysis_dboundaries_plot2.ipynb

Generative models: Continuous inputs

➢ Class-conditional densities are Gaussian distributed

$$p(\boldsymbol{x}|y = C_k, \boldsymbol{\theta}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \Sigma) \triangleq \frac{1}{(2\pi)^{D/2}|\Sigma|^{1/2}} \exp\left[-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_k)^T \Sigma^{-1} (\mathbf{x} - \boldsymbol{\mu}_k)\right]$$

Multi–Variate Gaussian

$\boldsymbol{\mu}_k$ = mean vector

Σ= covariance matrix

➢ For two classes

$$\mathrm{p}(y = C_1|\mathbf{x}, \boldsymbol{\theta}) = \frac{p(y = C_1|\boldsymbol{\theta})p(\boldsymbol{x}|y = C_1, \boldsymbol{\theta})}{p(y = C_1|\boldsymbol{\theta})p(\boldsymbol{x}|y = C_1, \boldsymbol{\theta}) + p(y = C_2|\boldsymbol{\theta})p(\boldsymbol{x}|y = C_2, \boldsymbol{\theta})}$$

independent of C_1

prior probability of class C_1

$p(y = C_1|\boldsymbol{\theta}) = \pi_1$

$$\log(\mathrm{p}(y = C_1|\mathbf{x}, \boldsymbol{\theta})) = \log p(y = C_1|\boldsymbol{\theta}) + \log(p(\boldsymbol{x}|y = C_1, \boldsymbol{\theta})) + \text{constant}$$

$$= \log p(y = C_1|\boldsymbol{\theta}) + \log(\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_1, \Sigma)) + \text{constant}$$

$$= \log p(y = C_1|\boldsymbol{\theta}) + -\frac{1}{2}\log(|\Sigma|) - \frac{D}{2}\log(2\pi)\left[-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_1)^T \Sigma^{-1} (\mathbf{x} - \boldsymbol{\mu}_1)\right] + \text{constant}$$

$$= \log \pi_1 - \frac{1}{2}\log(|\Sigma|) - \frac{D}{2}\log(2\pi)\left[-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_1)^T \Sigma^{-1} (\mathbf{x} - \boldsymbol{\mu}_1)\right] + \text{constant}$$

Generative models: Continuous inputs

- Class-conditional densities are Gaussian distributed

$$p(\boldsymbol{x}|y = C_k, \boldsymbol{\theta}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \Sigma) \triangleq \frac{1}{(2\pi)^{D/2}|\Sigma|^{1/2}} \exp\left[-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_k)^T\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_k)\right]$$

Multi–Variate Gaussian
$\boldsymbol{\mu}_k$ = mean vector
Σ= covariance matrix

$$\log(\mathrm{p}(y = C_k|\mathbf{x}, \boldsymbol{\theta})) = \log p(y = Ck|\boldsymbol{\theta}) + \log(p(\boldsymbol{x}|y = Ck, \boldsymbol{\theta})) + \text{constant}$$

$$= \log \pi_k - \frac{1}{2}\log(|\Sigma|) - \frac{D}{2}\log(2\pi)\left[-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_k)^T\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_k)\right] + \text{constant}$$

$$= \log \pi_k - \frac{1}{2}\boldsymbol{\mu}_k{}^{\mathrm{T}}\Sigma^{-1}\boldsymbol{\mu}_k + \boldsymbol{x}^{\mathrm{T}}\Sigma^{-1}\boldsymbol{\mu}_k + \text{constant} - \frac{1}{2}\boldsymbol{x}^{\mathrm{T}}\Sigma^{-1}\boldsymbol{x}$$

$$= \boxed{\log \pi_k - \frac{1}{2}\boldsymbol{\mu}_k{}^{\mathrm{T}}\Sigma^{-1}\boldsymbol{\mu}_k} + \boldsymbol{x}^{\mathrm{T}}\boxed{\Sigma^{-1}\boldsymbol{\mu}_k} + \boxed{\text{constant} - \frac{1}{2}\boldsymbol{x}^{\mathrm{T}}\Sigma^{-1}\boldsymbol{x}}$$

$$p(y = C_1|\mathbf{x}, \boldsymbol{\theta}) = \sigma\big(a(\boldsymbol{x})\big) \Rightarrow \log(\mathrm{p}(y = Ck|\mathbf{x}, \boldsymbol{\theta})) = (\widetilde{\boldsymbol{w}}^T\boldsymbol{x} + \mathrm{w}_0)$$

$$\widetilde{\boldsymbol{w}} = \Sigma^{-1}\boldsymbol{\mu}_k$$

$$\mathrm{w}_0 = \log \pi_k - \frac{1}{2}\boldsymbol{\mu}_k{}^{\mathrm{T}}\Sigma^{-1}\boldsymbol{\mu}_k$$

Due to shared covariance matrix assumption, the quadratic part $\boldsymbol{x}^T\Sigma^{-1}\boldsymbol{x}$ cancels off and log $(\mathrm{p}(y = Ck|\boldsymbol{x}, \boldsymbol{\theta}))$ is a linear function of $\boldsymbol{x}$. This is called linear discriminant analysis.

Generative models: Continuous inputs

- Maximum likelihood estimation

$$p(\boldsymbol{x}|y = Ck, \boldsymbol{\theta}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}) \triangleq \frac{1}{(2\pi)^{D/2}|\Sigma|^{1/2}} \exp\left[-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)^T\Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu}_k)\right]$$

$$p(y = C_k|\mathbf{x}, \boldsymbol{\theta}) = \frac{p(y = Ck|\boldsymbol{\theta})p(\boldsymbol{x}|y = C_k, \boldsymbol{\theta})}{\sum_{C_k'} p(y = C_k'|\boldsymbol{\theta})p(\boldsymbol{x}|y = C_k', \boldsymbol{\theta})}$$

$$p(y = C_k|\mathbf{x}, \boldsymbol{\theta}) \propto p(y = Ck|\boldsymbol{\theta})p(\boldsymbol{x}|y = Ck, \boldsymbol{\theta})$$

- For two classes

$$p(y = C_1|\mathbf{x}, \boldsymbol{\theta}) \propto p(y = C_1|\boldsymbol{\theta})p(\boldsymbol{x}|y = C_1, \boldsymbol{\theta})$$

$$p(y = C_2|\mathbf{x}, \boldsymbol{\theta}) \propto p(y = C_2|\boldsymbol{\theta})p(\boldsymbol{x}|y = C_2, \boldsymbol{\theta})$$

$$p(y = C_1|\mathbf{x}, \boldsymbol{\theta}) \propto \pi_1\, p(\boldsymbol{x}|y = C_1, \boldsymbol{\theta})$$

$$p(y = C_2|\mathbf{x}, \boldsymbol{\theta}) \propto (1-\pi_1)\, p(\boldsymbol{x}|y = C_2, \boldsymbol{\theta})$$

$$p(y = C_1|\mathbf{x}, \boldsymbol{\theta}) \propto \pi_1\, \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_1, \boldsymbol{\Sigma})$$

$$p(y = C_2|\mathbf{x}, \boldsymbol{\theta}) \propto (1-\pi_1)\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_2, \boldsymbol{\Sigma})$$

- likelihood function

$$p(\boldsymbol{c}|\pi_1, \boldsymbol{\mu}_2, \boldsymbol{\mu}_1, \boldsymbol{\Sigma}) = \mathrm{L} = \prod_i [p(y = C_1|\mathbf{x}_i, \boldsymbol{\theta})]^{c_i} [p(y = C_2|\mathbf{x}_i, \boldsymbol{\theta})]^{1-c_i} = \prod_i [\pi_1 \mathcal{N}(\mathbf{x}_i|\boldsymbol{\mu}_1, \boldsymbol{\Sigma})]^{c_i} [(1-\pi_1)\mathcal{N}(\mathbf{x}_i|\boldsymbol{\mu}_2, \boldsymbol{\Sigma})]^{1-c_i}$$

Generative models: Continuous inputs

- Likelihood function

$$p(\boldsymbol{c}|\pi_1, \boldsymbol{\mu_2}, \boldsymbol{\mu_1}, \boldsymbol{\Sigma}) = \mathrm{L} = \prod_i [p(y = C_1|\mathbf{x_i}, \boldsymbol{\theta})]^{c_i} [p(y = C_2|\mathbf{x_i}, \boldsymbol{\theta})]^{1-c_i} = \prod_i [\pi_1 \mathcal{N}(\mathbf{x_i}|\boldsymbol{\mu_1}, \boldsymbol{\Sigma})]^{c_i} [(1-\pi_1)\mathcal{N}(\mathbf{x}|\boldsymbol{\mu_2}, \boldsymbol{\Sigma})]^{1-c_i}$$

- Log likelihood function

$$\sum_i c_\mathrm{i} \log [\pi_1 \mathcal{N}(\mathbf{x}|\boldsymbol{\mu_1}, \boldsymbol{\Sigma})] + (1 - c_\mathrm{i}) \log [(1-\pi_1) \mathcal{N}(\mathbf{x}|\boldsymbol{\mu_2}, \boldsymbol{\Sigma})]$$

- First maximize with respect to π_1

$$\mathrm{L}\pi_1 = \sum_i c_\mathrm{i} \log [\pi_1] + (1 - c_\mathrm{i}) \log [(1-\pi_1)]$$

$$\frac{\partial \mathrm{L}\pi_1}{\partial \pi_1} = 0 \qquad \pi_1 = \frac{\mathrm{N}_1}{N} = \frac{\mathrm{N}_1}{\mathrm{N}_1 + \mathrm{N}_2}$$

Generative models: Continuous inputs

- Likelihood function

$$p(\boldsymbol{c}|\pi_1, \boldsymbol{\mu_2}, \boldsymbol{\mu_1}, \boldsymbol{\Sigma}) = \mathrm{L} = \prod_i [p(y = C_1|\mathbf{x_i}, \boldsymbol{\theta})]^{c_i} [p(y = C_2|\mathbf{x_i}, \boldsymbol{\theta})]^{1-c_i} = \prod_i [\pi_1 \mathcal{N}(\mathbf{x_i}|\boldsymbol{\mu_1}, \boldsymbol{\Sigma})]^{c_i} [(1-\pi_1)\mathcal{N}(\mathbf{x}|\boldsymbol{\mu_2}, \boldsymbol{\Sigma})]^{1-c_i}$$

- Log likelihood function

$$\sum_i c_i \log [\pi_1 \mathcal{N}(\mathbf{x}|\boldsymbol{\mu_1}, \boldsymbol{\Sigma})] + (1 - c_i) \log [(1-\pi_1) \mathcal{N}(\mathbf{x}|\boldsymbol{\mu_2}, \boldsymbol{\Sigma})]$$

- First maximize with respect to $\boldsymbol{\mu_1}$

$$\mathrm{L}\boldsymbol{\mu_1} = \sum_i c_i \log [\pi_1 \mathcal{N}(\mathbf{x_i}|\boldsymbol{\mu_1}, \boldsymbol{\Sigma})] = \sum_i c_i \left[-\frac{1}{2}(\mathbf{x_i} - \boldsymbol{\mu}_1)^T \Sigma^{-1} (\mathbf{x_i} - \boldsymbol{\mu}_1)\right] + const$$

$$\frac{\partial \mathrm{L}\boldsymbol{\mu_1}}{\partial \boldsymbol{\mu_1}} = 0 \qquad \boldsymbol{\mu_1} = \frac{1}{\mathrm{N}_1} \sum_{i=1}^{N} c_i \, \mathbf{x_i} \qquad \text{Similarly } \boldsymbol{\mu_2} = \frac{1}{\mathrm{N}_2} \sum_{i=1}^{N} (1 - c_i) \, \mathbf{x_i}$$

Generative models: Continuous inputs

- Likelihood function

$$p(\boldsymbol{c}|\pi_1, \boldsymbol{\mu_2}, \boldsymbol{\mu_1}, \boldsymbol{\Sigma}) = \mathrm{L} = \prod_i [p(y = C_1|\mathbf{x_i}, \boldsymbol{\theta})]^{c_i} [p(y = C_2|\mathbf{x_i}, \boldsymbol{\theta})]^{1-c_i} = \prod_i [\pi_1 \mathcal{N}(\mathbf{x_i}|\boldsymbol{\mu_1}, \boldsymbol{\Sigma})]^{c_i} [(1-\pi_1)\mathcal{N}(\mathbf{x}|\boldsymbol{\mu_2}, \boldsymbol{\Sigma})]^{1-c_i}$$

- Log likelihood function

$$\sum_i c_\mathrm{i} \log [\pi_1 \mathcal{N}(\mathbf{x}|\boldsymbol{\mu_1}, \boldsymbol{\Sigma})] + (1 - c_\mathrm{i}) \log [(1-\pi_1) \mathcal{N}(\mathbf{x}|\boldsymbol{\mu_2}, \boldsymbol{\Sigma})]$$

- First maximize with respect to $\boldsymbol{\Sigma}$

$$\mathrm{L}_{\boldsymbol{\Sigma}} = \sum_i c_\mathrm{i} \log [\pi_1 \mathcal{N}(\mathbf{x_i}|\boldsymbol{\mu_1}, \boldsymbol{\Sigma})] = \sum_i -\frac{1}{2} c_\mathrm{i} \log(|\Sigma|) - \left[\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_k)^T \Sigma^{-1} (\mathbf{x} - \boldsymbol{\mu}_k)\right] + \mathrm{const}$$

$$\frac{\partial \mathrm{L}_{\boldsymbol{\Sigma}}}{\partial \boldsymbol{\Sigma}} = 0 \qquad \boldsymbol{\Sigma} = \frac{1}{N} \sum_{i=1}^{N} (\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T = \frac{1}{N} \left(\sum_{i=1}^{N} \mathbf{x}_i \mathbf{x}_i^T \right) - \bar{\mathbf{x}}\bar{\mathbf{x}}^T$$

Generative models: Continuous inputs

- Maximum likelihood estimation

$$\hat{\pi}_k = \frac{\mathrm{N_k}}{N} \qquad \widehat{\boldsymbol{\mu}_k} = \frac{1}{\mathrm{N}_k} \sum_{i:\, yi=k}^{N} \mathbf{x_i} \qquad \widehat{\boldsymbol{\Sigma}_k} = \frac{1}{\mathrm{N}_k} \sum_{i:y_i=k}^{N} (\mathbf{x}_i - \boldsymbol{\mu}_k)(\mathbf{x}_i - \boldsymbol{\mu}_k)^T$$

For tied variance ($\boldsymbol{\Sigma} = \boldsymbol{\Sigma}_\mathbf{k}$)

$$\widehat{\boldsymbol{\Sigma}} = \frac{1}{N} \sum_{i=1}^{N} (\mathbf{x}_i - \overline{\mathbf{x}})(\mathbf{x}_i - \overline{\mathbf{x}})^T = \frac{1}{N} \left(\sum_{i=1}^{N} \mathbf{x}_i \mathbf{x}_i^T \right) - \overline{\mathbf{x}}\overline{\mathbf{x}}^T$$

Generative models: Continuous inputs

- Quadrature discriminant analysis
- We drop the shared covariance matrix assumption

$$p(\boldsymbol{x}|y = Ck, \boldsymbol{\theta}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \Sigma_k) \triangleq \frac{1}{(2\pi)^{D/2}|\Sigma_k|^{1/2}} \exp\left[-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_k)^T {\Sigma_k}^{-1}(\mathbf{x} - \boldsymbol{\mu}_k)\right]$$

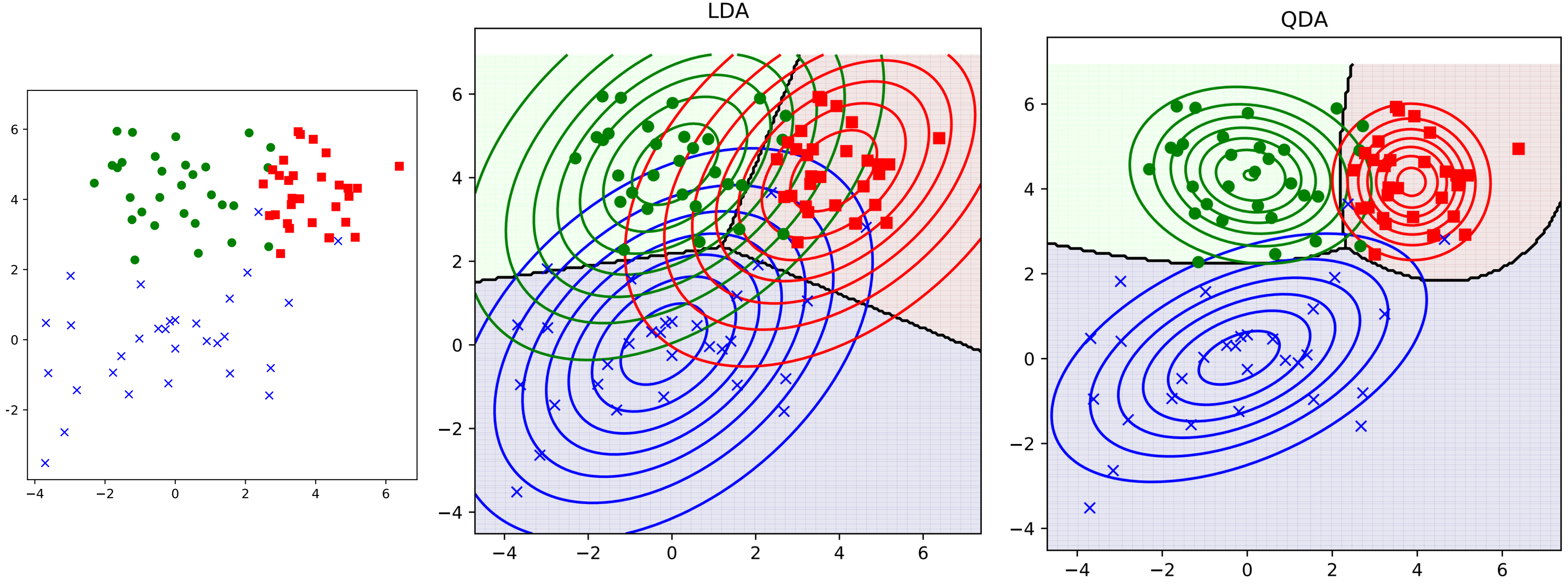
Multi–Variate Gaussian
$\boldsymbol{\mu}_k$ = mean vector
Σ_k= covariance matrix

$$p(y = C_k|\mathbf{x}, \boldsymbol{\theta}) = \frac{p(y = Ck|\boldsymbol{\theta})p(\mathbf{x}|y = C_k, \boldsymbol{\theta})}{\sum_{C_k{}'} p(y = C_k{}'|\boldsymbol{\theta})p(\mathbf{x}|y = C_k{}', \boldsymbol{\theta})}$$

Gaussian discriminant analysis or **GDA**

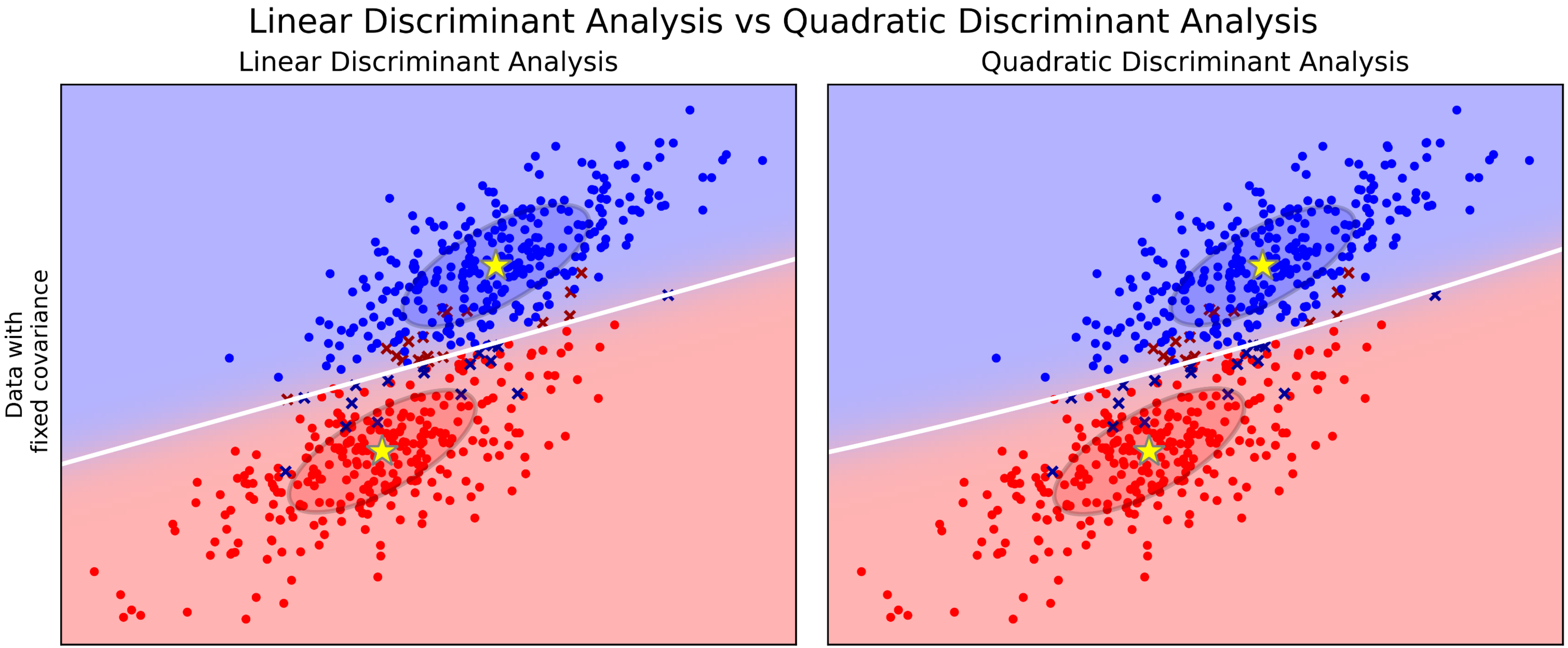
$$p(y = C_k|\mathbf{x}, \boldsymbol{\theta}) = \frac{\frac{1}{(2\pi)^{D/2}|\Sigma_k|^{1/2}} \exp\left[-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_k)^T {\Sigma_k}^{-1}(\mathbf{x} - \boldsymbol{\mu}_k)\right]}{\sum_{C_k{}'} \pi_{k'}|2\pi\Sigma_{k'}|^{-1/2} \exp\left[-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_{k'})^T \Sigma_{k'}^{-1}(\mathbf{x} - \boldsymbol{\mu}_{k'})\right]}$$

QDA & LDA



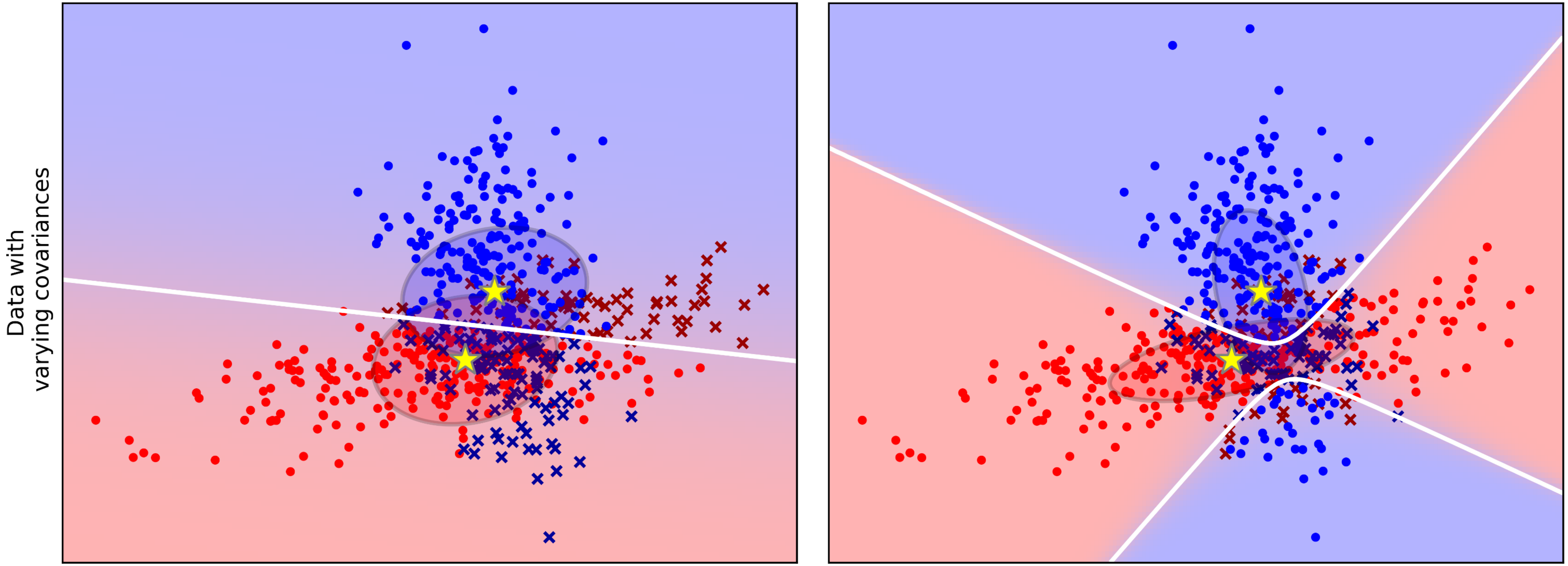
Kevin P. Murphy, *Machine Learning: A Probabilistic Perspective*, 2012.
probml.github.io/notebooks#discrim_analysis_dboundaries_plot2.ipynb

QDA & LDA



1.2. Linear and Quadratic Discriminant Analysis — scikit-learn 1.3.0 documentation

QDA & LDA



1.2. Linear and Quadratic Discriminant Analysis — scikit-learn 1.3.0 documentation

Advantages of generative classifiers

- Ease of fitting
- Handling missing features
- Class-specific learning
- Can handle unlabeled data
- Robustness to spurious features

Thank You

Q & A