Lecture 5

Gaussian Models - Part 1

Luigi Freda

ALCOR Lab
DIAG
University of Rome "La Sapienza"

November 29, 2016

Outline

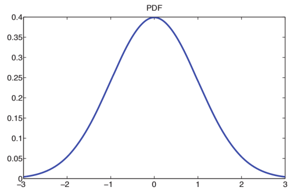
Outline

Univariate Gaussian (Normal) Distribution

- X is a continuous RV with values $x \in \mathbb{R}$
- $X \sim \mathcal{N}(\mu, \sigma^2)$, i.e. X has a **Gaussian distribution** or **normal distribution**

$$\mathcal{N}(x|\mu,\sigma^2) \triangleq \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}(x-\mu)^2} \qquad (= \; P_X(X = x))$$

- mean $\mathbb{E}[X] = \mu$
- mode μ
- variance $\mathrm{var}[X] = \sigma^2$
- precision $\lambda = \frac{1}{\sigma^2}$
- $(\mu - 2\sigma, \mu + 2\sigma)$ is the approx 95% interval
- $(\mu - 3\sigma, \mu + 3\sigma)$ is the approx. 99.7% interval

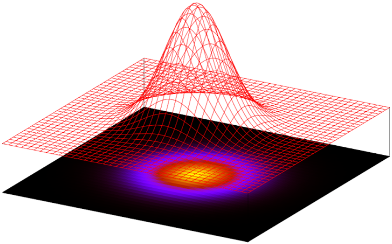


Multivariate Gaussian (Normal) Distribution

- $\mathbf{X}$ is a continuous RV with values $\mathbf{x} \in \mathbb{R}^D$
- $\mathbf{X} \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)$, i.e. $\mathbf{X}$ has a **Multivariate Normal** distribution (MVN) or **multivariate Gaussian**

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

- mean: $\mathbb{E}[\mathbf{x}] = \boldsymbol{\mu}$
- mode: $\boldsymbol{\mu}$
- covariance matrix: $\mathrm{cov}[\mathbf{x}] = \boldsymbol{\Sigma} \in \mathbb{R}^{D \times D}$ where $\boldsymbol{\Sigma} = \boldsymbol{\Sigma}^T$ and $\boldsymbol{\Sigma} \geq 0$
- precision matrix: $\boldsymbol{\Lambda} \triangleq \boldsymbol{\Sigma}^{-1}$
- spherical isotropic covariance with $\boldsymbol{\Sigma} = \sigma^2 \mathbf{I}_D$



Outline

MLE for an MVN

Theorem

Theorem 1

If we have N iid samples $\mathbf{x}_i \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$*, then the MLE for the parameters is given by*

1. $\boldsymbol{\mu}_{MLE} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{x}_i \triangleq \overline{\mathbf{x}}$
2. $\boldsymbol{\Sigma}_{MLE} = \frac{1}{N}\sum_{i=1}^{N} (\mathbf{x}_i - \overline{\mathbf{x}})(\mathbf{x}_i - \overline{\mathbf{x}})^T = \frac{1}{N}\left(\sum_{i=1}^{N} \mathbf{x}_i \mathbf{x}_i^T\right) - \overline{\mathbf{x}}\,\overline{\mathbf{x}}^T$

- this theorem states the MLE parameter estimates for an MVN are just the **empirical mean** and the **empirical covariance**
- in the **univariate case**, one has

$$\hat{\mu} = \frac{1}{N}\sum_{i=1}^{N} x_i \triangleq \overline{x}$$

$$\hat{\sigma}^2 = \frac{1}{N}\sum_{i=1}^{N} (x_i - \overline{x})(x_i - \overline{x})^T = \frac{1}{N}\left(\sum_{i=1}^{N} x_i x_i^T\right) - \overline{x}^2$$

MLE for an MVN

Theorem

proof sketch

- in order to find the MLE one should maximize the log-likelihood of the dataset
- given that $\mathbf{x}_i \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$

$$p(\mathcal{D}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \prod_i \mathcal{N}(\mathbf{x}_i|\boldsymbol{\mu}, \boldsymbol{\Sigma})$$

- the log-likelihood (dropping additive constants) is

$$l(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \log p(\mathcal{D}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{N}{2} \log |\boldsymbol{\Lambda}| - \frac{1}{2} \sum_i (\mathbf{x}_i - \boldsymbol{\mu}) \boldsymbol{\Lambda} (\mathbf{x}_i - \boldsymbol{\mu})^T + const$$

- the MLE estimates can be obtained by maximizing $l(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ w.r.t. $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$

homework: continue the proof for the univariate case

Outline

Generative Classifiers

probabilistic classifier

- we are given a dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$
- the goal is to compute the **class posterior** $p(y = c|\mathbf{x})$ which models the mapping $y = f(\mathbf{x})$

generative classifiers

- $p(y = c|\mathbf{x})$ is computed starting from the **class-conditional density** $p(\mathbf{x}|y = c, \boldsymbol{\theta})$ and the **class prior** $p(y = c|\boldsymbol{\theta})$ given that

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

- this is called a **generative classifier** since it specifies how to generate the feature vector $\mathbf{x}$ for each class $y = c$ (by using $p(\mathbf{x}|y = c, \boldsymbol{\theta})$)
- the model is usually fit by maximizing the joint log-likelihood, i.e. one computes $\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} \sum_i \log p(y_i, \mathbf{x}_i|\boldsymbol{\theta})$

discriminative classifiers

- the model $p(y = c|\mathbf{x})$ is directly fit to the data
- the model is usually fit by maximizing the conditional log-likelihood, i.e. one computes $\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} \sum_i \log p(y_i|\mathbf{x}_i, \boldsymbol{\theta})$

Outline

- we can use the MVN for defining the class conditional densities in a generative classifier

$$p(\mathbf{x}|y=c,\boldsymbol{\theta}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_c,\boldsymbol{\Sigma}_c) \qquad \text{for } c \in \{1,...,C\}$$

- this means the samples of each class c are characterized by a normal distribution
- this model is called **Gaussian Discriminative Analysis** (GDA) but it is a **generative classifier** (not discriminative)
- in the case $\boldsymbol{\Sigma}_c$ is diagonal for each c, this model is equivalent to a Naive Bayes Classifier (NBC) since

$$p(\mathbf{x}|y=c,\boldsymbol{\theta}) = \prod_{j=1}^{D} \mathcal{N}(x_j|\mu_{jc},\sigma_{jc}^2) \qquad \text{for } c \in \{1,...,C\}$$

- once the model is fit to the data, we can classify a feature vector by using the decision rule

$$\hat{y}(\mathbf{x}) = \underset{c}{\operatorname{argmax}} \log p(y=c|\mathbf{x},\boldsymbol{\theta}) = \underset{c}{\operatorname{argmax}} \Big[\log p(y=c|\boldsymbol{\pi}) + \log p(\mathbf{x}|y=c,\boldsymbol{\theta}_c) \Big]$$

- decision rule

$$\hat{y}(\mathbf{x}) = \underset{c}{\text{argmax}} \left[\log p(y = c|\boldsymbol{\pi}) + \log p(\mathbf{x}|y = c, \boldsymbol{\theta}_c) \right]$$

- given that $y \sim \text{Cat}(\boldsymbol{\pi})$ and $\mathbf{x}|(y = c) \sim \mathcal{N}(\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)$ the decision rule becomes (dropping additive constants)

$$\hat{y}(\mathbf{x}) = \underset{c}{\text{argmin}} \left[-\log \pi_c + \frac{1}{2} \log |\boldsymbol{\Sigma}_c| + \frac{1}{2} (\mathbf{x} - \boldsymbol{\mu}_c)^T \boldsymbol{\Sigma}_c^{-1} (\mathbf{x} - \boldsymbol{\mu}_c) \right]$$

which can be thought as a **nearest centroid classifier**

- in fact, with an uniform prior and $\boldsymbol{\Sigma}_c = \boldsymbol{\Sigma}$

$$\hat{y}(\mathbf{x}) = \underset{c}{\text{argmin}} \ (\mathbf{x} - \boldsymbol{\mu}_c)^T \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}_c) = \underset{c}{\text{argmin}} \ \|\mathbf{x} - \boldsymbol{\mu}_c\|_{\boldsymbol{\Sigma}}^2$$

- in this case, we select the class c whose center $\boldsymbol{\mu}_c$ is closest to $\mathbf{x}$ (using the **Mahalanobis distance** $\|\mathbf{x} - \boldsymbol{\mu}_c\|_{\boldsymbol{\Sigma}}$)

Mahalanobis Distance

- the covariance matrix $\mathbf{\Sigma}$ can be diagonalized since it is a symmetric real matrix

$$\mathbf{\Sigma} = \mathbf{U}\mathbf{D}\mathbf{U}^T = \sum_{i=1}^{D} \lambda_i \mathbf{u}_i \mathbf{u}_i^T$$

  where $\mathbf{U} = [\mathbf{u}_1, ..., \mathbf{u}_D]$ is an orthonormal matrix of eigenvectors (i.e. $\mathbf{U}^T\mathbf{U} = \mathbf{I}$) and λ_i are the corresponding eigenvalues ($\lambda_i \geq 0$ since $\mathbf{\Sigma} \geq 0$)

- one has immediately $\mathbf{\Sigma}^{-1} = \mathbf{U}\mathbf{D}^{-1}\mathbf{U}^T = \sum_{i=1}^{D} \dfrac{1}{\lambda_i} \mathbf{u}_i \mathbf{u}_i^T$

- the **Mahalanobis distance** is defined as $\|\mathbf{x} - \boldsymbol{\mu}\|_{\mathbf{\Sigma}} \triangleq \Big((\mathbf{x} - \boldsymbol{\mu})^T \mathbf{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}) \Big)^{1/2}$

- one can rewrite

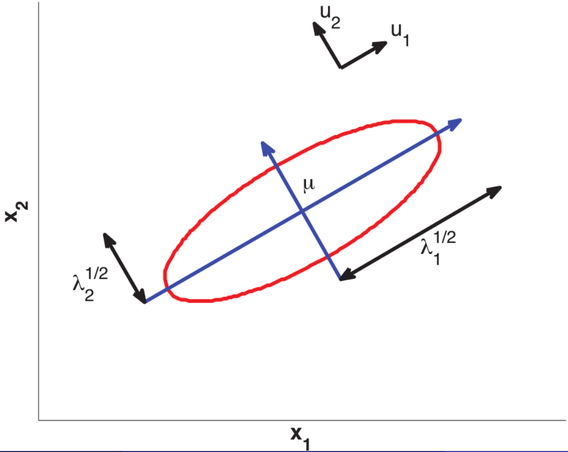
$$(\mathbf{x} - \boldsymbol{\mu})^T \mathbf{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}) = (\mathbf{x} - \boldsymbol{\mu})^T \left(\sum_{i=1}^{D} \frac{1}{\lambda_i} \mathbf{u}_i \mathbf{u}_i^T \right) (\mathbf{x} - \boldsymbol{\mu}) =$$

$$= \sum_{i=1}^{D} \frac{1}{\lambda_i} (\mathbf{x} - \boldsymbol{\mu})^T \mathbf{u}_i \mathbf{u}_i^T (\mathbf{x} - \boldsymbol{\mu}) = \sum_{i=1}^{D} \frac{y_i^2}{\lambda_i}$$

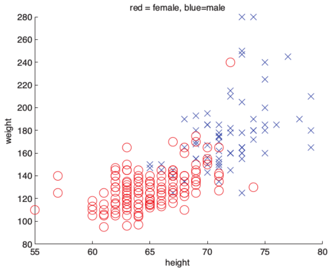
  where $y_i \triangleq \mathbf{u}_i^T (\mathbf{x} - \boldsymbol{\mu})$ (or equivalently $\mathbf{y} \triangleq \mathbf{U}^T (\mathbf{x} - \boldsymbol{\mu})$)

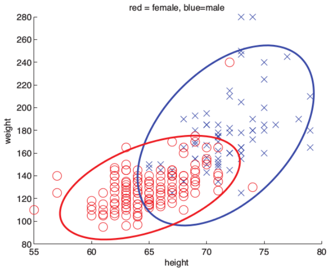
Mahalanobis Distance

- $\boldsymbol{\Sigma} = \mathbf{U}\mathbf{D}\mathbf{U}^T = \sum_{i=1}^{D} \lambda_i \mathbf{u}_i \mathbf{u}_i^T$
- $(\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}) = \sum_{i=1}^{D} \frac{y_i^2}{\lambda_i}$ (where $\mathbf{y} \triangleq \mathbf{U}^T(\mathbf{x} - \boldsymbol{\mu})$)
- *(1)* center w.r.t. $\boldsymbol{\mu}$ *(2)* rotate by $\mathbf{U}^T$ *(3)* get a norm weighted by the $\frac{1}{\lambda_i}$



Gaussian Discriminant Analysis
GDA





- *left*: height/weight data for the two classes male/female
- *right*: visualization of 2D Gaussian fit to each class
- we can see that the features are correlated (tall people tend to weigh more)

Outline

- the **complete class posterior** with Gaussian densities is

$$p(y = c|\mathbf{x}, \boldsymbol{\theta}) = \frac{\pi_c |2\pi \boldsymbol{\Sigma}_c|^{-1/2} \exp[-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_c)^T \boldsymbol{\Sigma}_c^{-1} (\mathbf{x} - \boldsymbol{\mu}_c)]}{\sum_{c'} \pi_{c'} |2\pi \boldsymbol{\Sigma}_{c'}|^{-1/2} \exp[-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_{c'})^T \boldsymbol{\Sigma}_{c'}^{-1} (\mathbf{x} - \boldsymbol{\mu}_{c'})}$$

- the **quadratic decision boundaries** can be found by imposing

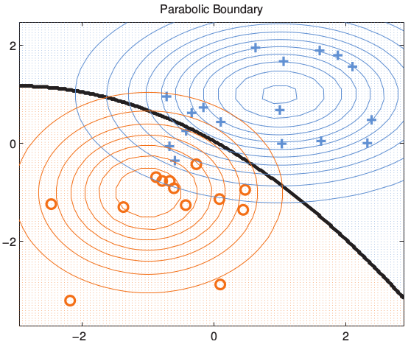
$$p(y = c'|\mathbf{x}, \boldsymbol{\theta}) = p(y = c''|\mathbf{x}, \boldsymbol{\theta})$$

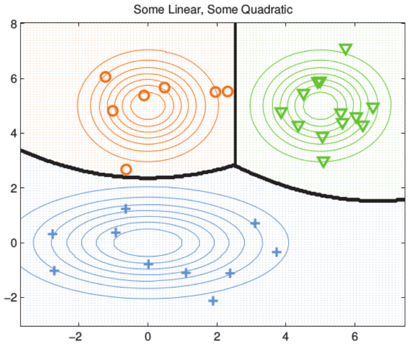
or equivalently

$$\log p(y = c'|\mathbf{x}, \boldsymbol{\theta}) = \log p(y = c''|\mathbf{x}, \boldsymbol{\theta})$$

for each pair of "adjacent" classes (c', c''), which results in the quadratic equation

$$-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_{c'})^T \boldsymbol{\Sigma}_{c'}^{-1} (\mathbf{x} - \boldsymbol{\mu}_{c'}) = -\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_{c''})^T \boldsymbol{\Sigma}_{c''}^{-1} (\mathbf{x} - \boldsymbol{\mu}_{c''}) + \text{constant}$$





- *left*: dataset with 2 classes
- *right*: dataset with 3 classes

Outline

- we now consider the GDA in the special case $\mathbf{\Sigma}_c = \mathbf{\Sigma}$ for $c \in \{1, ..., C\}$
- in this case we have

$$p(y = c|\mathbf{x}, \boldsymbol{\theta}) \propto \pi_c \exp\left[-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_c)^T \mathbf{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu}_c) \right] =$$

$$= \exp\left[-\frac{1}{2}\mathbf{x}^T \mathbf{\Sigma}^{-1}\mathbf{x} \right] \exp\left[\boldsymbol{\mu}_c^T \mathbf{\Sigma}^{-1}\mathbf{x} - \frac{1}{2}\boldsymbol{\mu}_c^T \mathbf{\Sigma}^{-1}\boldsymbol{\mu}_c + \log \pi_c \right]$$

- note that the quadratic term $-\frac{1}{2}\mathbf{x}^T \mathbf{\Sigma}^{-1}\mathbf{x}$ is independent of c and it will cancel out in the numerator and denominator of the complete class posterior equation
- we define

$$\gamma_c \triangleq -\frac{1}{2}\boldsymbol{\mu}_c^T \mathbf{\Sigma}^{-1}\boldsymbol{\mu}_c + \log \pi_c$$

$$\boldsymbol{\beta}_c \triangleq \mathbf{\Sigma}^{-1}\boldsymbol{\mu}_c$$

- we can rewrite

$$p(y = c|\mathbf{x}, \boldsymbol{\theta}) = \frac{e^{\boldsymbol{\beta}_c^T \mathbf{x} + \gamma_c}}{\sum_{c'} e^{\boldsymbol{\beta}_{c'}^T \mathbf{x} + \gamma_{c'}}}$$

- we have

$$p(y = c|\mathbf{x}, \boldsymbol{\theta}) = \frac{e^{\boldsymbol{\beta}_c^T \mathbf{x} + \gamma_c}}{\sum_{c'} e^{\boldsymbol{\beta}_{c'}^T \mathbf{x} + \gamma_{c'}}} \triangleq S(\boldsymbol{\eta})_c$$

where $\boldsymbol{\eta} \triangleq [\boldsymbol{\beta}_1^T \mathbf{x} + \gamma_1, ..., \boldsymbol{\beta}_C^T \mathbf{x} + \gamma_C]^T \in \mathbb{R}^C$ and the function $S(\eta)$ is the **softmax function** defined as

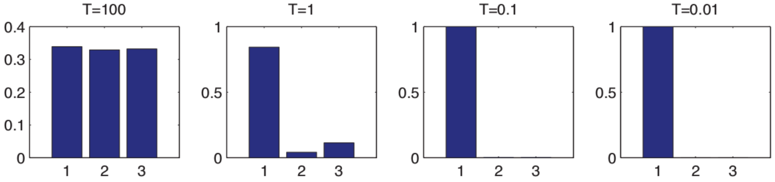
$$S(\boldsymbol{\eta}) \triangleq \left[\frac{e^{\eta_1}}{\sum_{c'} e^{\eta_{c'}}}, ..., \frac{e^{\eta_C}}{\sum_{c'} e^{\eta_{c'}}} \right]^T$$

and $S(\boldsymbol{\eta})_c \in \mathbb{R}$ is just its c-th component

- the softmax function $S(\eta)$ is so-called since it acts a bit like the max function. To see this, divide each component η_c by a **temperature** T, then

$$S(\boldsymbol{\eta}/T)_c = \begin{cases} 1 & \text{if } c = \underset{c'}{\operatorname{argmax}}\ \eta_{c'} \\ 0 & \text{otherwise} \end{cases} \qquad \text{as } T \to 0$$

- in other words, at low temperature $S(\boldsymbol{\eta}/T)_c$ returns the most probable state, whereas at high temperatures $S(\boldsymbol{\eta}/T)_c$ returns one of the states with a uniform probability (cfr. **Bolzmann distribution** in physics)



- softmax distribution $S(\boldsymbol{\eta}/T)$, where $\boldsymbol{\eta} = [3, 0, 1]^T$, at different temperatures T
- when the temperature is high (left), the distribution is uniform, whereas when the temperature is low (right), the distribution is "spiky", with all its mass on the largest element

- in order to find the decision boundaries we impose

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

which entails

$$e^{\boldsymbol{\beta}_c^T \mathbf{x} + \gamma_c} = e^{\boldsymbol{\beta}_{c'}^T \mathbf{x} + \gamma_{c'}}$$

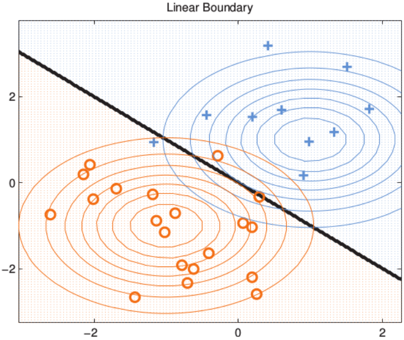
- in this case, taking the logs returns

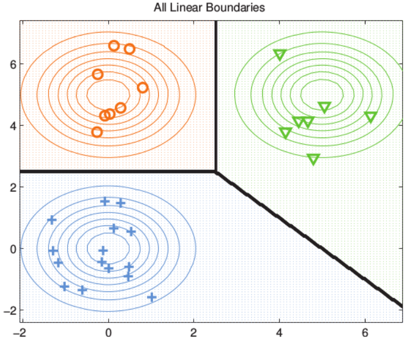
$$\boldsymbol{\beta}_c^T \mathbf{x} + \gamma_c = \boldsymbol{\beta}_{c'}^T \mathbf{x} + \gamma_{c'}$$

which in turn corresponds to a **linear decision boundary**[1]

$$(\boldsymbol{\beta}_c - \boldsymbol{\beta}_{c'})^T \mathbf{x} = -(\gamma_c - \gamma_{c'})$$

[1] in D dimensions this corresponds to an hyperplane, in 3D to a plane, in 2D to a straight line





- *left*: dataset with 2 classes
- *right*: dataset with 3 classes

Linear Discriminant Analysis
two-class LDA

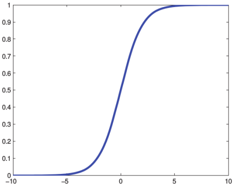
- let us consider an LDA with just two classes (i.e. $y \in \{0, 1\}$)
- in this case

$$p(y = 1|\mathbf{x}, \boldsymbol{\theta}) = \frac{e^{\boldsymbol{\beta}_1^T \mathbf{x} + \gamma_1}}{e^{\boldsymbol{\beta}_1^T \mathbf{x} + \gamma_1} + e^{\boldsymbol{\beta}_0^T \mathbf{x} + \gamma_0}} = \frac{1}{1 + e^{(\boldsymbol{\beta}_0 - \boldsymbol{\beta}_1)^T \mathbf{x} + (\gamma_0 - \gamma_1)}}$$

that is

$$p(y = 1|\mathbf{x}, \boldsymbol{\theta}) = \text{sigm}((\boldsymbol{\beta}_0 - \boldsymbol{\beta}_1)^T \mathbf{x} + (\gamma_0 - \gamma_1))$$

where $\mathrm{sigm}(\eta) \triangleq \frac{1}{1+\exp(-\eta)}$ is **the sigmoid function** (aka logistic function)



- the linear decision boundary is

$$(\boldsymbol{\beta}_0 - \boldsymbol{\beta}_1)^T \mathbf{x} + (\gamma_0 - \gamma_1) = 0$$

- if we define

$$\mathbf{w} \triangleq \boldsymbol{\beta}_1 - \boldsymbol{\beta}_0 = \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)$$
$$\mathbf{x}_0 \triangleq \frac{1}{2}(\boldsymbol{\mu}_1 + \boldsymbol{\mu}_0) - (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)\frac{\log(\pi_1/\pi_0)}{(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)}$$

  we obtain $\mathbf{w}^T \mathbf{x}_0 = -(\gamma_1 - \gamma_0)$

- the linear decision boundary can be rewritten as

$$\mathbf{w}^T(\mathbf{x} - \mathbf{x}_0) = 0$$

- in fact we have

$$p(y = 1|\mathbf{x}, \boldsymbol{\theta}) = \mathrm{sigm}(\mathbf{w}^T(\mathbf{x} - \mathbf{x}_0))$$

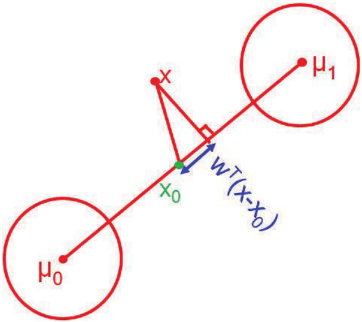
Linear Discriminant Analysis
two-class LDA

- we have

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

$$\mathbf{x}_0 \triangleq \frac{1}{2}(\boldsymbol{\mu}_1 + \boldsymbol{\mu}_0) - (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)\frac{\log(\pi_1/\pi_0)}{(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)}$$

- the linear decision boundary is $\mathbf{w}^T(\mathbf{x} - \mathbf{x}_0) = 0$
- in the case $\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2 = \mathbf{I}$ and $\pi_1 = \pi_0$, one has $\mathbf{w} = \boldsymbol{\mu}_1 - \boldsymbol{\mu}_0$ and $\mathbf{x}_0 = \frac{1}{2}(\boldsymbol{\mu}_1 + \boldsymbol{\mu}_0)$



MLE for GDA

- how to fit the GDA model?
- the simplest way is to use MLE
- let's assume iid samples, then it is $p(\mathcal{D}|\boldsymbol{\theta}) = \prod_{i=1}^{N} p(\mathbf{x}_i, y_i|\boldsymbol{\theta})$
- one has

$$p(\mathbf{x}_i, y_i|\boldsymbol{\theta}) = p(\mathbf{x}_i|y_i, \boldsymbol{\theta})p(y_i|\boldsymbol{\pi})$$

$$p(\mathbf{x}_i|y_i, \boldsymbol{\theta}) = \prod_c \mathcal{N}(\mathbf{x}_i|\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)^{\mathbb{I}(y_i=c)} \qquad p(y_i|\boldsymbol{\pi}) = \prod_c \pi_c^{\mathbb{I}(y_i=c)}$$

  where $\boldsymbol{\theta}$ is a compound parameter vector containing the parameters $\boldsymbol{\pi}$, $\boldsymbol{\mu}_c$ and $\boldsymbol{\Sigma}_c$
- the log-likelihood function is

$$\log p(\mathcal{D}|\boldsymbol{\theta}) = \left[\sum_{i=1}^{N}\sum_{c=1}^{C} \mathbb{I}(y_i = c) \log \pi_c\right] + \sum_{c=1}^{C}\left[\sum_{i:y_i=c} \log \mathcal{N}(\mathbf{x}_i|\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)\right]$$

  which is the sum of $C + 1$ distinct terms: the first depending on $\boldsymbol{\pi}$ and the other C terms depending both on $\boldsymbol{\mu}_c$ and $\boldsymbol{\Sigma}_c$
- we can estimate each parameter by optimizing the log-likelihood separately w.r.t. it

MLE for GDA

- the log-likelihood function is

$$\log p(\mathcal{D}|\boldsymbol{\theta}) = \left[\sum_{i=1}^{N} \sum_{c=1}^{C} \mathbb{I}(y_i = c) \log \pi_c \right] + \sum_{c=1}^{C} \left[\sum_{i:y_i=c} \log \mathcal{N}(\mathbf{x}_i | \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c) \right]$$

- for the class prior, as with the NBC model, we have

$$\hat{\pi}_c = \frac{N_c}{N}$$

- for the class conditional densities, we partition the data based on its class label, and compute the MLE for each Gaussian term

$$\hat{\boldsymbol{\mu}}_c = \frac{1}{N_c} \sum_{i:y_i=c}^{N_c} \mathbf{x}_i$$

$$\hat{\boldsymbol{\Sigma}}_c = \frac{1}{N_c} \sum_{i:y_i=c}^{N_c} (\mathbf{x}_i - \hat{\boldsymbol{\mu}}_c)(\mathbf{x}_i - \hat{\boldsymbol{\mu}}_c)^T$$

Posterior Predictive for GDA

- once the model is fit and the parameters are estimated we can make predictions by using a plug-in approximation

$$p(y = c|\mathbf{x}, \hat{\boldsymbol{\theta}}) \propto \hat{\pi}_c |2\pi\hat{\boldsymbol{\Sigma}}_c|^{-1/2} \exp[-\frac{1}{2}(\mathbf{x} - \hat{\boldsymbol{\mu}}_c)^T \hat{\boldsymbol{\Sigma}}_c^{-1}(\mathbf{x} - \hat{\boldsymbol{\mu}}_c)]$$

Overfitting for GDA

- the MLE is fast and simple, however it can badly overfit in high dimensions
- in particular, $\hat{\mathbf{\Sigma}}_c = \frac{1}{N_c} \sum_{i:y_i=c}^{N_c} (\mathbf{x}_i - \hat{\boldsymbol{\mu}}_c)(\mathbf{x}_i - \hat{\boldsymbol{\mu}}_c)^T \in \mathbb{R}^{D \times D}$ is singular for $N_c < D$
- even when $N_c > D$, the MLE can be ill-conditioned (close to singular)
- possible simple strategies to solve this issue (they reduce the number of parameters)
  - use NBC model/assumption (i.e. $\mathbf{\Sigma}_c$ are diagonal)
  - use LDA (i.e. $\mathbf{\Sigma}_c = \mathbf{\Sigma}$)
  - use diagonal LDA (i.e. $\mathbf{\Sigma}_c = \mathbf{\Sigma} = \text{diag}(\sigma_1^2, ..., \sigma_D^2)$) (following subsection)
  - use Bayesian approach: estimate full covariance by imposing a prior and then integrating out (following subsection)

Outline

Diagonal LDA

- the diagonal LDA assumes $\mathbf{\Sigma}_c = \mathbf{\Sigma} = \text{diag}(\sigma_1^2, ..., \sigma_D^2)$ for $c \in \{1, ..., C\}$
- one has

$$p(\mathbf{x}_i, y_i = c|\boldsymbol{\theta}) = p(\mathbf{x}_i|y_i = c, \boldsymbol{\theta}_c)p(y_i = c|\boldsymbol{\pi}) = \mathcal{N}(\mathbf{x}_i|\boldsymbol{\mu}_c, \mathbf{\Sigma})\pi_c = \prod_{j=1}^{D} \mathcal{N}(x_{ij}|\mu_{cj}, \sigma_j^2)$$

and taking the logs

$$\log p(\mathbf{x}_i, y_i = c|\boldsymbol{\theta}) = -\sum_{j=1}^{D} \frac{(x_{ij} - \mu_{cj})^2}{2\sigma_j^2} + \log \pi_c$$

- typically the estimates of the parameters are

$$\hat{\mu}_{cj} = \frac{1}{N_c} \sum_{i:y_i=c} x_{ij}$$

$$\hat{\sigma}_j^2 = \frac{1}{N - C} \sum_{c=1}^{C} \sum_{i:y_i=c} (x_{ij} - \hat{\mu}_{cj})^2 \qquad \text{(pooled empirical variance)}$$

- in high-dimensional settings, this model can work much better than LDA and RDA

Outline

Bayesian Procedure

- we now follow the full Bayesian procedure to fit the GDA model
- let's restart from the expression of the **posterior predictive** PDF

$$p(y = c|\mathbf{x}, \mathcal{D}) = \frac{p(y = c, \mathbf{x}|\mathcal{D})}{p(\mathbf{x}|\mathcal{D})} = \frac{p(\mathbf{x}|y = c, \mathcal{D})p(y = c|\mathcal{D})}{p(\mathbf{x}|\mathcal{D})}$$

- since we are interested in computing

$$c* = \underset{c}{\operatorname{argmax}} \;\; p(y = c|\mathbf{x}, \mathcal{D})$$

we can neglect the constant $p(\mathbf{x}|\mathcal{D})$ and use the following simpler expression

$$p(y = c|\mathbf{x}, \mathcal{D}) \propto p(\mathbf{x}|y = c, \mathcal{D})p(y = c|\mathcal{D})$$

- note that we didn't use the model parameters in the previous equation
- now we use the **Bayesian procedure** in which we integrate out the unknown parameters
- for simplicity we now consider a vector parameter $\boldsymbol{\pi}$ for the PMF $p(y = c|\mathcal{D})$ and a vector parameter $\boldsymbol{\theta}_c$ for the PDF $p(\mathbf{x}|y = c, \mathcal{D})$

Bayesian Procedure

- as for the PMF $p(y = c|\mathcal{D})$ we can integrate out $\boldsymbol{\pi}$ as follows

$$p(y = c|\mathcal{D}) = \int p(y = c, \boldsymbol{\pi}|\mathcal{D})d\boldsymbol{\pi}$$

- we know that $y \sim \text{Cat}(\boldsymbol{\pi})$ i.e. $p(y|\boldsymbol{\pi}) = \prod_c \pi_c^{\mathbb{I}(y=c)}$
- we can decompose $p(y = c, \boldsymbol{\pi}|\mathcal{D})$ as follows

$$p(y = c, \boldsymbol{\pi}|\mathcal{D}) = p(y = c|\boldsymbol{\pi}, \mathcal{D})p(\boldsymbol{\pi}|\mathcal{D}) = p(y = c|\boldsymbol{\pi})p(\boldsymbol{\pi}|\mathcal{D}) = \pi_c p(\boldsymbol{\pi}|\mathcal{D})$$

  where $p(\boldsymbol{\pi}|\mathcal{D})$ is the posterior w.r.t. $\boldsymbol{\pi}$
- using the previous equation in integral above we have

$$p(y = c|\mathcal{D}) = \int p(y = c, \boldsymbol{\pi}|\mathcal{D})d\boldsymbol{\pi} = \int \pi_c p(\boldsymbol{\pi}|\mathcal{D})d\boldsymbol{\pi} = \mathbb{E}[\pi_c|\mathcal{D}] = \frac{N_c + \alpha_c}{N + \alpha_0}$$

  which is the posterior mean computed for the **Dirichlet-multinomial** model (cfr lecture 4 slides)

Bayesian Procedure

- as for the PDF $p(\mathbf{x}|y=c,\mathcal{D})$ we can integrate out $\boldsymbol{\theta}_c$ as follows

$$p(\mathbf{x}|y=c,\mathcal{D}) = \int p(\mathbf{x},\boldsymbol{\theta}_c|y=c,\mathcal{D})d\boldsymbol{\theta}_c = \int p(\mathbf{x},\boldsymbol{\theta}_c|\mathcal{D}_c)d\boldsymbol{\theta}_c$$

  where for simplicity we introduce $\mathcal{D}_c \triangleq \{(\mathbf{x_i}, y_i) \in \mathcal{D} | y_i = c\}$
- we know that $p(\mathbf{x}|\boldsymbol{\theta}_c) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)$ where $\boldsymbol{\theta}_c = (\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)$
- we can use the following decomposition

$$p(\mathbf{x},\boldsymbol{\theta}_c|\mathcal{D}_c) = p(\mathbf{x}|\boldsymbol{\theta}_c,\mathcal{D}_c)p(\boldsymbol{\theta}_c|\mathcal{D}_c) = p(\mathbf{x}|\boldsymbol{\theta}_c)p(\boldsymbol{\theta}_c|\mathcal{D}_c)$$

  where $p(\boldsymbol{\theta}_c|\mathcal{D}_c)$ is the posterior w.r.t. $\boldsymbol{\theta}_c$
- hence one has

$$p(\mathbf{x}|y=c,\mathcal{D}) = \int p(\mathbf{x},\boldsymbol{\theta}_c|\mathcal{D}_c)d\boldsymbol{\theta}_c = \int p(\mathbf{x}|\boldsymbol{\theta}_c)p(\boldsymbol{\theta}_c|\mathcal{D}_c)d\boldsymbol{\theta}_c =$$

$$= \int\int \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_c,\boldsymbol{\Sigma}_c)p(\boldsymbol{\mu}_c,\boldsymbol{\Sigma}_c|\mathcal{D}_c)d\boldsymbol{\mu}_c d\boldsymbol{\Sigma}_c$$

Bayesian Procedure

- one has

$$p(\mathbf{x}|y=c,\mathcal{D}) = \int\int \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_c,\boldsymbol{\Sigma}_c)p(\boldsymbol{\mu}_c,\boldsymbol{\Sigma}_c|\mathcal{D}_c)d\boldsymbol{\mu}_c d\boldsymbol{\Sigma}_c$$

- the posterior is (see sect. 4.6.3.3 of the book)

$$p(\boldsymbol{\mu}_c,\boldsymbol{\Sigma}_c|\mathcal{D}_c) = \text{NIW}(\mathbf{m}_c,\boldsymbol{\Sigma}_c|\mathbf{m}_N^c,\kappa_N^c,\nu_N^c,\mathbf{S}_N^c)$$

- then (see sect. 4.6.3.6)

$$p(\mathbf{x}|y=c,\mathcal{D}) = \int\int \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_c,\boldsymbol{\Sigma}_c)\text{NIW}(\boldsymbol{\mu}_c,\boldsymbol{\Sigma}_c|\mathbf{m}_N^c,\kappa_N^c,\nu_N^c,\mathbf{S}_N^c)d\boldsymbol{\mu}_c d\boldsymbol{\Sigma}_c =$$

$$p(\mathbf{x}|y=c,\mathcal{D}) = \mathcal{T}(\mathbf{x}|\mathbf{m}_N^c,\frac{\kappa_N^c+1}{\kappa_N^c(\nu_N^c-D+1)}\mathbf{S}_N^c,\nu_N^c-D+1)$$

- let's summarize what we obtained by applying the Bayesian procedure
- we first found

$$p(y = c|\mathcal{D}) = \mathbb{E}[\pi_c|\mathcal{D}] = \frac{N_c + \alpha_c}{N + \alpha_0}$$

  and then

$$p(\mathbf{x}|y = c, \mathcal{D}) = \mathcal{T}(\mathbf{x}|\mathbf{m}_N^c, \frac{\kappa_N^c + 1}{\kappa_N^c(\nu_N^c - D + 1)}\mathbf{S}_N^c, \nu_N^c - D + 1)$$

- then combining everything in the starting posterior predictive we have

$$p(y = c|\mathbf{x}, \mathcal{D}) \propto p(\mathbf{x}|y = c, \mathcal{D})p(y = c|\mathcal{D}) =$$

$$= \mathbb{E}[\pi_c|\mathcal{D}]\mathcal{T}(\mathbf{x}|\mathbf{m}_N^c, \frac{\kappa_N^c + 1}{\kappa_N^c(\nu_N^c - D + 1)}\mathbf{S}_N^c, \nu_N^c - D + 1)$$

Credits

- Kevin Murphy's book