Machine Learning

Linear and quadratic discriminant analysis

Sebastian Engelke

Master in Business Analytics

Université de Genève

Linear decision boundaries: hyperplanes

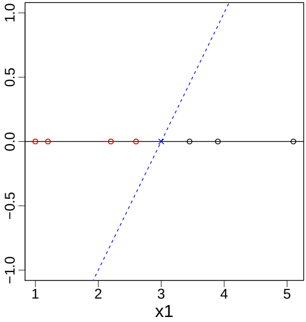
- Suppose we have only two classes, denoted by $\mathcal{G} = \{0, 1\}$.
- Linear classification methods produce `linear decision boundaries`, called `hyperplanes`.
- In a p-dimensional space, a hyperplane is a flat `affine subspace of dimension` $p-1$, and it can be written, for parameters $b_0, b_1, \ldots, b_p \in \mathbb{R}$, as the set

$$x \in \mathbb{R}^p : \qquad b_0 + b_1 x_1 + \cdots + b_p x_p = 0.$$

Example:

- For $p = 1$, the hyperplane is just a point

$$x \in \mathbb{R} : \qquad b_0 + b_1 x = 0 \quad \Leftrightarrow \quad x = -b_0/b_1.$$



Linear decision boundaries: hyperplanes

- Suppose we have only two classes, denoted by $\mathcal{G} = \{0, 1\}$.
- Linear classification methods produce `linear decision boundaries`, called `hyperplanes`.
- In a p-dimensional space, a hyperplane is a flat `affine subspace of dimension` $p-1$, and it can be written, for parameters $b_0, b_1, \ldots, b_p \in \mathbb{R}$, as the set

$$x \in \mathbb{R}^p : \qquad b_0 + b_1 x_1 + \cdots + b_p x_p = 0.$$

- Suppose a point $x \in \mathbb{R}^p$ is not on the hyperplane, then

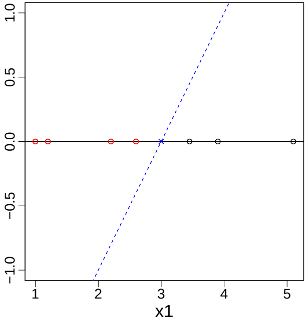
$$b_0 + b_1 x_1 + \cdots + b_p x_p > 0,$$

tells us that it lies on the `one side` of the hyperplane, or with "$<$" on the other.

Example:

- For $p = 1$, the hyperplane is just a point

$$x \in \mathbb{R} : \qquad b_0 + b_1 x = 0 \quad \Leftrightarrow \quad x = -b_0/b_1.$$



Linear decision boundaries: hyperplanes

- Suppose we have only two classes, denoted by $\mathcal{G} = \{0, 1\}$.
- Linear classification methods produce `linear decision boundaries`, called `hyperplanes`.
- In a p-dimensional space, a hyperplane is a flat `affine subspace of dimension` $p-1$, and it can be written, for parameters $b_0, b_1, \dots, b_p \in \mathbb{R}$, as the set

$$x \in \mathbb{R}^p : \qquad b_0 + b_1 x_1 + \cdots + b_p x_p = 0.$$

- Suppose a point $x \in \mathbb{R}^p$ is not on the hyperplane, then

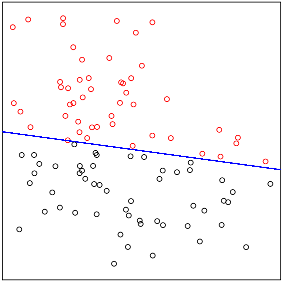
$$b_0 + b_1 x_1 + \cdots + b_p x_p > 0,$$

tells us that it lies on the `one side` of the hyperplane, or with "$<$" on the other.

Example:

- For $p = 2$, the hyperplane is a line

$$(x_1, x_2) \in \mathbb{R}^2 : \qquad b_0 + b_1 x_1 + b_2 x_2 = 0$$
$$\Leftrightarrow \quad x_2 = -b_0/b_2 - (b_1/b_2) x_1.$$



Linear decision boundaries: hyperplanes

- Suppose we have only two classes, denoted by $\mathcal{G} = \{0, 1\}$.
- Linear classification methods produce `linear decision boundaries`, called `hyperplanes`.
- In a p-dimensional space, a hyperplane is a flat `affine subspace of dimension` $p-1$, and it can be written, for parameters $b_0, b_1, \ldots, b_p \in \mathbb{R}$, as the set

$$x \in \mathbb{R}^p : \qquad b_0 + b_1 x_1 + \cdots + b_p x_p = 0.$$

- Suppose a point $x \in \mathbb{R}^p$ is not on the hyperplane, then

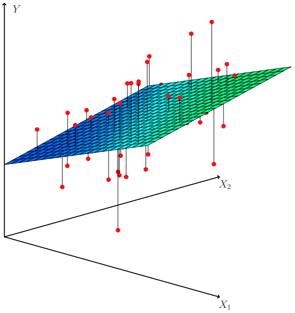
$$b_0 + b_1 x_1 + \cdots + b_p x_p > 0,$$

tells us that it lies on the `one side` of the hyperplane, or with "$<$" on the other.

Example:

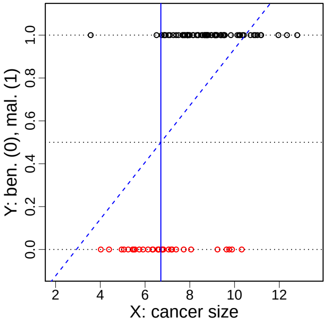
- For $p = 3$, the hyperplane is

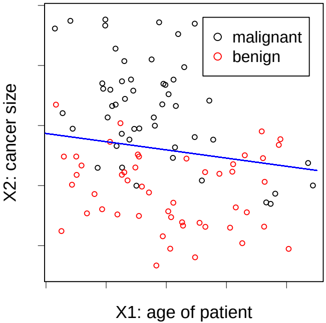
$$(x_1, x_2, x_3) \in \mathbb{R}^3 : \qquad b_0 + b_1 x_1 + b_2 x_2 + b_3 x_3 = 0.$$



Linear methods for classification

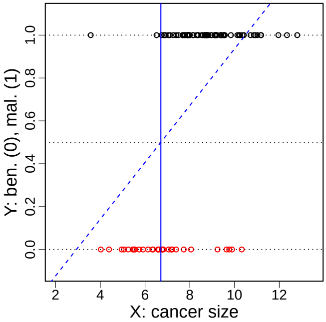
- Example: direct linear regression on $\mathcal{G} = \{0, 1\}$; usually not a good idea!

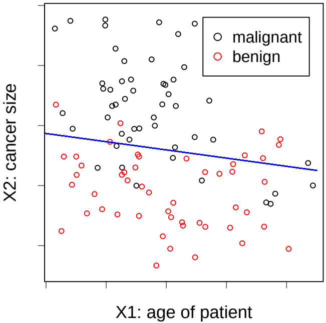




Linear methods for classification

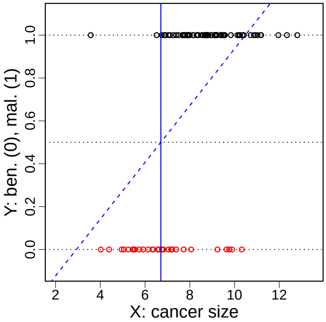
- Example: direct linear regression on $\mathcal{G} = \{0, 1\}$; usually not a good idea!
- Many other methods will produce linear decision boundaries.

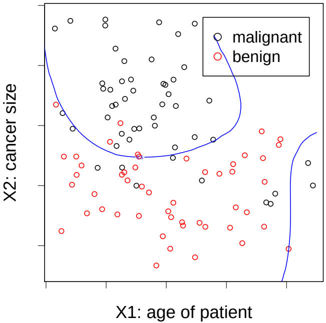




Linear methods for classification

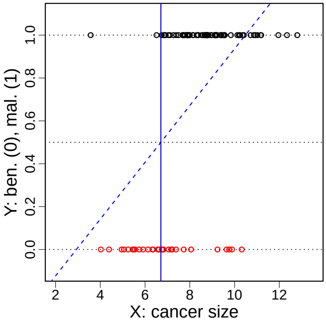
- Example: direct linear regression on $\mathcal{G} = \{0, 1\}$; usually not a good idea!
- Many other methods will produce linear decision boundaries.
- Decision boundaries of linear methods are linear only in the `feature space`.
- `Augmenting the feature` space by basis functions might lead to non-linear decision boundaries in the original space of $(X_1, \dots, X_p)$.
- Right: the model is linear in $X_1, X_2, X_1^2, X_2^2, X_1^3, X_2^3, X_1X_2, X_1X_2^2, X_1^2X_2, X_1^2X_2^2$.

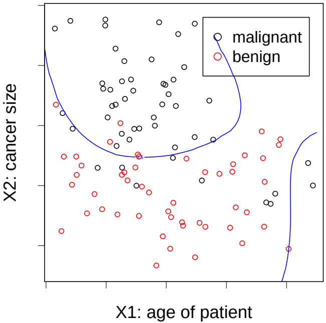




Linear methods for classification

- Example: direct linear regression on $\mathcal{G} = \{0, 1\}$; usually not a good idea!
- Many other methods will produce linear decision boundaries.
- Decision boundaries of linear methods are linear only in the `feature space`.
- `Augmenting the feature` space by basis functions might lead to non-linear decision boundaries in the original space of $(X_1, \ldots, X_p)$.
- Right: the model is linear in $X_1, X_2, X_1^2, X_2^2, X_1^3, X_2^3, X_1X_2, X_1X_2^2, X_1^2X_2, X_1^2X_2^2$.
- For more than two classes, linear classification methods produce `piecewise linear decision boundaries`.





Linear discriminant analysis

- **Setup**: Classify $Y \in \mathcal{G} = \{1, \ldots, q\}$ given predictors $X = (X_1, \ldots, X_p)$
- **Idea**: model the `class-conditional densities` g_j of $X \mid Y = j$, and the marginals $\pi_j = \mathrm{Pr}(Y = j)$, $j = 1, \ldots, q$. Deduce $\mathrm{Pr}(Y \mid X)$ from `Bayes' formula`:

$$\mathrm{Pr}(Y = j \mid X = x) = \frac{\mathrm{Pr}(X = x \mid Y = j)\mathrm{Pr}(Y = j)}{\mathrm{Pr}(X = x)} \propto g_j(x)\pi_j.$$

Linear discriminant analysis

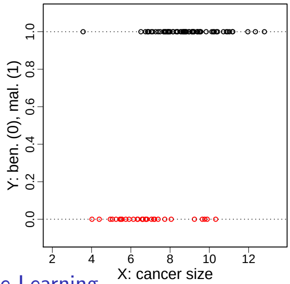
- **Setup**: Classify $Y \in \mathcal{G} = \{1, \ldots, q\}$ given predictors $X = (X_1, \ldots, X_p)$
- **Idea**: model the class-conditional densities g_j of $X \mid Y = j$, and the marginals $\pi_j = \mathrm{Pr}(Y = j)$, $j = 1, \ldots, q$. Deduce $\mathrm{Pr}(Y \mid X)$ from Bayes' formula:

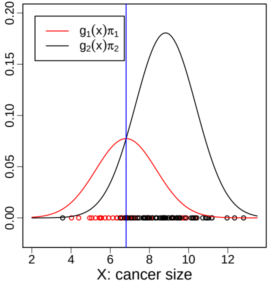
$$\mathrm{Pr}(Y = j \mid X = x) = \frac{\mathrm{Pr}(X = x \mid Y = j)\mathrm{Pr}(Y = j)}{\mathrm{Pr}(X = x)} \propto g_j(x)\pi_j.$$

- Linear discriminant analysis (LDA):
  (1) model $g_j(x)$ by a multivariate normal distribution,

$$g_j(x) = \frac{1}{(2\pi)^{p/2}(\det \mathbf{\Sigma}_j)^{1/2}} \exp\left\{-\frac{1}{2}(x - \mu_j)^T \mathbf{\Sigma}_j^{-1}(x - \mu_j)\right\};$$

  (2) assume the g_j have the same covariance matrix $\mathbf{\Sigma}_j = \mathbf{\Sigma}$ for all j.





LDA: estimation

- **Model parameters:**
  - the class/prior probabilities $\pi_1, \ldots, \pi_q \in (0, 1)$;
  - the mean vectors $\mu_1, \ldots, \mu_q \in \mathbb{R}^p$;
  - the covariance matrix $\mathbf{\Sigma} \in \mathbb{R}^{p \times p}$.
- **Estimation:** From a training sample $(y_1, x_1), \ldots, (y_n, x_n)$ we estimate the parameters of LDA by the usual `maximum likelihood estimators`

$$\widehat{\pi}_j = \frac{1}{n} \sum_{i=1}^{n} \mathbf{1}\{y_i = j\}$$

$$\widehat{\mu}_j = \frac{\sum_{i=1}^{n} x_i \mathbf{1}\{y_i = j\}}{\sum_{i=1}^{n} \mathbf{1}\{y_i = j\}}$$

$$\widehat{\mathbf{\Sigma}} = \frac{\sum_{j=1}^{q} \sum_{i=1}^{n} (x_i - \widehat{\mu}_j)(x_i - \widehat{\mu}_j)^T \mathbf{1}\{y_i = j\}}{n - q}$$

- This is very simple and `computationally fast`!
- Note: Different to linear regression, we don't use the numerical value of the y_i's but only the class!

LDA: prediction

- **Goal**: for a new input value $x_0 \in \mathbb{R}^p$ predict its unobserved class y_0.
- **Method**: compute the `posterior class probabilities` $\widehat{\Pr}(Y = j \mid X = x_0)$ and use the `Bayes classifier` resulting from these estimates (plug-in estimate), i.e.,

$$\widehat{y}_0 = \arg\max_{j=1,\dots,q} \widehat{\Pr}(Y = j \mid \boldsymbol{X} = x_0) = \arg\max_{j=1,\dots,q} \log \widehat{\Pr}(Y = j \mid X = x_0).$$

LDA: prediction

- **Goal**: for a new input value $x_0 \in \mathbb{R}^p$ predict its unobserved class y_0.
- **Method**: compute the posterior class probabilities $\widehat{\Pr}(Y = j \mid X = x_0)$ and use the Bayes classifier resulting from these estimates (plug-in estimate), i.e.,

$$\widehat{y}_0 = \arg\max_{j=1,\ldots,q} \widehat{\Pr}(Y = j \mid \boldsymbol{X} = x_0) = \arg\max_{j=1,\ldots,q} \log \widehat{\Pr}(Y = j \mid X = x_0).$$

- Note that

$$\begin{aligned} \log \widehat{\Pr}(Y = j \mid X = x_0) &= \log \widehat{g}_j(x_0) + \log \widehat{\pi}_j + \text{constant} \\ &= -\frac{1}{2}(x_0 - \widehat{\mu}_j)^T \widehat{\boldsymbol{\Sigma}}^{-1}(x_0 - \widehat{\mu}_j) + \log \widehat{\pi}_j + \text{constant} \qquad (1) \\ &= x_0^T \widehat{\boldsymbol{\Sigma}}^{-1} \widehat{\mu}_j - \frac{1}{2}\widehat{\mu}_j^T \widehat{\boldsymbol{\Sigma}}^{-1} \widehat{\mu}_j + \log \widehat{\pi}_j + \text{constant}, \end{aligned}$$

  where the constant includes terms that do not depend on j.
- **Geometric interpretation**: eq. (1) implies that LDA classifies x_0 in terms of the Mahalanobis distance, $d_{\widehat{\Sigma}^{-1}}(x, y) = x^T \widehat{\Sigma}^{-1} y$, of x_0 from the centers $\widehat{\mu}_1, \ldots, \widehat{\mu}_q$ (with a correction for the prior class probabilities).

LDA: decision boundaries

- The decision boundary that separates two classes $j, m \in \{1, \dots, q\}$ are all $x \in \mathbb{R}^p$ with

  $$\Pr(Y = j \mid X = x) = \Pr(Y = m \mid X = x),$$

  which is equivalent to

  $$\log \Pr(Y = j \mid X = x) - \log \Pr(Y = m \mid X = x) = 0.$$

  Replacing the probabilities by their expressions in terms of the Gaussian densities and the prior class probabilities we get the decision boundary

  $$\log \frac{\pi_j}{\pi_m} - \frac{1}{2}(\mu_j + \mu_m)^T \mathbf{\Sigma}^{-1}(\mu_j - \mu_m) + x^T \mathbf{\Sigma}^{-1}(\mu_j - \mu_m) = 0,$$

  a linear equation in x.

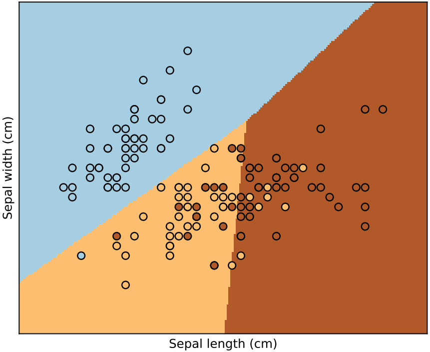
LDA: decision boundaries

- The decision boundary that separates two classes $j, m \in \{1, \dots, q\}$ are all $x \in \mathbb{R}^p$ with
$$\Pr(Y = j \mid X = x) = \Pr(Y = m \mid X = x),$$
which is equivalent to
$$\log \Pr(Y = j \mid X = x) - \log \Pr(Y = m \mid X = x) = 0.$$
Replacing the probabilities by their expressions in terms of the Gaussian densities and the prior class probabilities we get the decision boundary
$$\log \frac{\pi_j}{\pi_m} - \frac{1}{2}(\mu_j + \mu_m)^T \boldsymbol{\Sigma}^{-1}(\mu_j - \mu_m) + x^T \boldsymbol{\Sigma}^{-1}(\mu_j - \mu_m) = 0,$$
a linear equation in x.
- Hence the decisions boundaries separating any two classes j and m are hyperplanes in $\mathbb{R}^p$ of dimension $p - 1$.
- Thus the decision boundary that separates the q classes is piecewise linear.

Iris data set

For 150 measured plants; $p = 4$ predictors as response the three Iris species $\mathcal{G} = \{\texttt{setosa}, \texttt{versicolor}, \texttt{virginica}\}$ the plant belongs. We only use two predictors here.

```
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
iris = sklearn.datasets.load_iris()
X, y = iris.data[:, :2], iris.target  # we only keep the first two features: 'Sepal length', 'Sepal width'
lda = LinearDiscriminantAnalysis()
lda.fit(X, y)
x_min, x_max = X[:, 0].min() - .5, X[:, 0].max() + .5
y_min, y_max = X[:, 1].min() - .5, X[:, 1].max() + .5
h = .02  # step size in the mesh
xx, yy = np.meshgrid(np.arange(x_min, x_max, h), np.arange(y_min, y_max, h))
Z = lda.predict(np.c_[xx.ravel(), yy.ravel()]).reshape(xx.shape)
```



Extensions of LDA

- There are many possibilities to extend LDA. The Bayes' construction

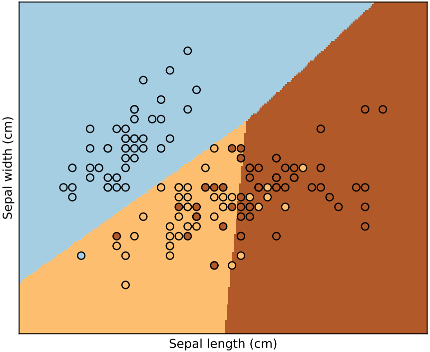
$$\Pr(Y = j \mid X = x) = \frac{\Pr(X = x \mid Y = j)\Pr(Y = j)}{\Pr(X = x)} = \frac{g_j(x)\pi_j}{\sum_{j=1}^{q} g_j(x)\pi_j}.$$

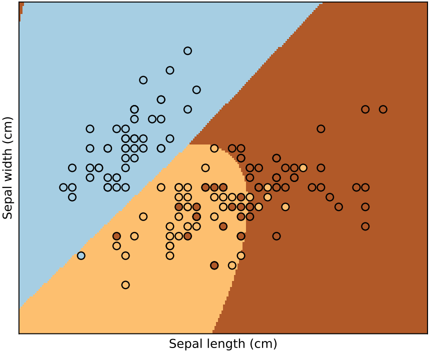
  can be used with other densities g_j.
- `Quadratic discriminant analysis` (QDA): model g_j by multivariate normals with different covariance matrices $\mathbf{\Sigma}_j$.

Iris data set: QDA

For 150 measured plants; $p = 4$ predictors as response the three Iris species $\mathcal{G} = \{\texttt{setosa}, \texttt{versicolor}, \texttt{virginica}\}$ the plant belongs. We only use two predictors here.

```
from sklearn.discriminant_analysis import QuadraticDiscriminantAnalysis
iris = sklearn.datasets.load_iris()
X, y = iris.data[:, :2], iris.target  # we only keep the first two features: 'Sepal length', 'Sepal width'
qda = QuadraticDiscriminantAnalysis()
qda.fit(X, y)
x_min, x_max = X[:, 0].min() - .5, X[:, 0].max() + .5
y_min, y_max = X[:, 1].min() - .5, X[:, 1].max() + .5
h = .02  # step size in the mesh
xx, yy = np.meshgrid(np.arange(x_min, x_max, h), np.arange(y_min, y_max, h))
Z = qda.predict(np.c_[xx.ravel(), yy.ravel()]).reshape(xx.shape)
```





Extensions of LDA

- There are many possibilities to extend LDA. The Bayes' construction

$$\Pr(Y = j \mid X = x) = \frac{\Pr(X = x \mid Y = j)\Pr(Y = j)}{\Pr(X = x)} = \frac{g_j(x)\pi_j}{\sum_{j=1}^{q} g_j(x)\pi_j}.$$

  can be used with other densities g_j.
- `Quadratic discriminant analysis` (QDA): model g_j by multivariate normals with different covariance matrices $\mathbf{\Sigma}_j$.
- `Mixture` models: use mixtures of Gaussians for the g_j.

Extensions of LDA

- There are many possibilities to extend LDA. The Bayes' construction

$$\mathrm{Pr}(Y = j \mid X = x) = \frac{\mathrm{Pr}(X = x \mid Y = j)\mathrm{Pr}(Y = j)}{\mathrm{Pr}(X = x)} = \frac{g_j(x)\pi_j}{\sum_{j=1}^{q} g_j(x)\pi_j}.$$

  can be used with other densities g_j.
- Quadratic discriminant analysis (QDA): model g_j by multivariate normals with different covariance matrices $\mathbf{\Sigma}_j$.
- Mixture models: use mixtures of Gaussians for the g_j.
- Naive Bayes: conditional independence assumption, $g_j(x) = \prod_{l=1}^{p} g_{j,l}(x_l)$.
- When some features are categorical, for example X_1, we can use

$$g_j(x) = \mathrm{Pr}(X_1 = x_1 \mid Y = j) g_j(x_2, \ldots, x_p \mid x_1),$$

  where $g_j(x_2, \ldots, x_p \mid x_1)$ is a density for $X_2, \ldots, X_p \mid X_1 = x_1, Y = j$.
- Non-parametric approaches: estimate g_j non-parametrically.

Comments on LDA

- LDA is easy and `fast to compute`.
- LDA provides estimates of the `posterior class probabilities`

$$\widehat{\Pr}(Y = j \mid X = x_0) = \frac{\widehat{g}_j(x_0)\widehat{\pi}_j}{\sum_{j=1}^{q} \widehat{g}_j(x_0)\pi_j}.$$

Comments on LDA

- LDA is easy and fast to compute.
- LDA provides estimates of the posterior class probabilities

$$\widehat{\Pr}(Y = j \mid X = x_0) = \frac{\widehat{g}_j(x_0)\widehat{\pi}_j}{\sum_{j=1}^{q} \widehat{g}_j(x_0)\pi_j}.$$

- LDA usually performs well and can compete with more complex methods.
- Why? Clearly not because the normality assumption is realistic, but because a linear decision boundary is often the best the data can support and LDA is a stable estimation method.
- We don't need to have good models for the g_j to obtain a good model for the posterior: complex g_j may not be visible in the decision (see figure).