Third-year ISAE-SUPAERO engineering students
Research Area: Neuro & AI
December, 2020

Neuro & AI: Methods and Tools for Neuroergonomics

Introduction to Machine Learning

Nicolas Drougard[1]

[1]ISAE-SUPAERO DCAS, Toulouse, FRANCE

nicolas.drougard@isae-supaero.fr

Institut Supérieur de l'Aéronautique et de l'Espace
SUPAERO

Outline

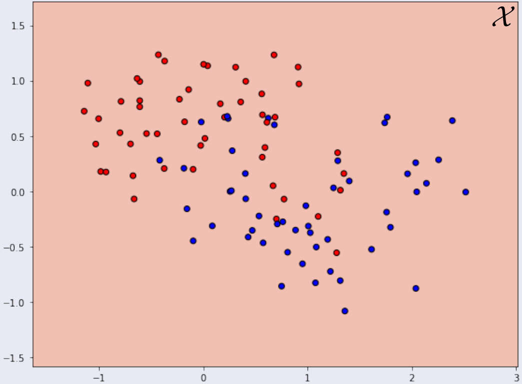
References

- [BBV04] Stephen Boyd, Stephen P Boyd, and Lieven Vandenberghe, *Convex optimization*, Cambridge university press, 2004, download link **here**.
- [HTF09] Trevor Hastie, Robert Tibshirani, and Jerome Friedman, *The elements of statistical learning: data mining, inference, and prediction*, Springer Science & Business Media, 2009, download link **here**.

Outline

Decision Tree

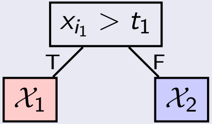
$\boxed{\mathcal{X}}$



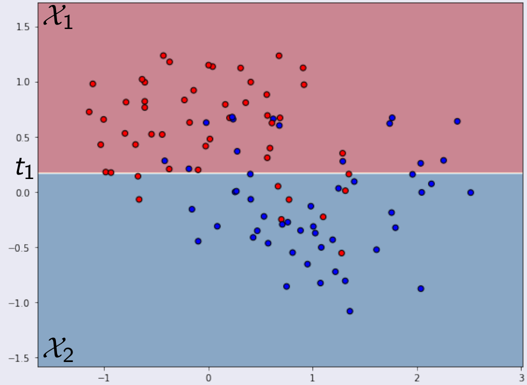
Splits using impurity criteria.

- sklearn: `tree.DecisionTreeClassifier`

Decision Tree



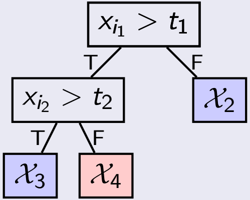
$i_1 = 2$



Splits using impurity criteria.

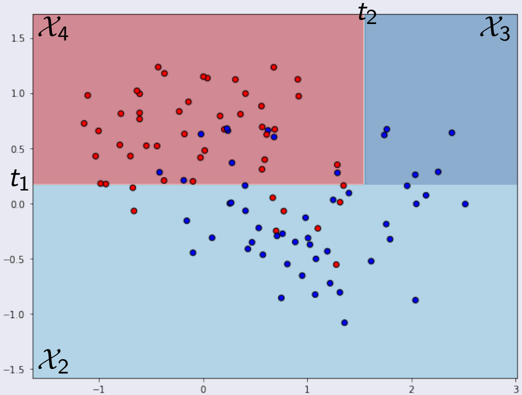
- sklearn: `tree.DecisionTreeClassifier`

Decision Tree



$i_1 = 2$

$i_2 = 1$

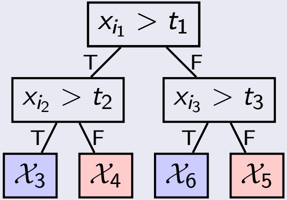


Splits using impurity criteria.

- sklearn: `tree.DecisionTreeClassifier`

Classical Supervised Learning Algorithms

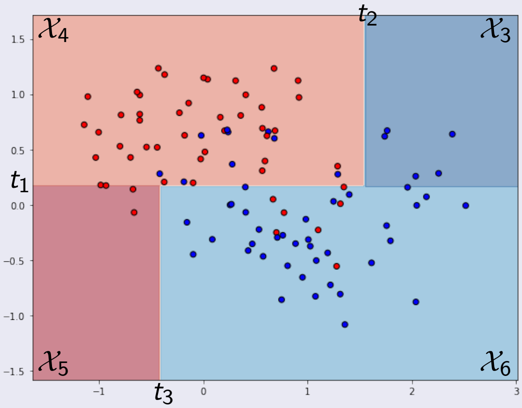
Decision Tree



$i_1 = 2$

$i_2 = 1$

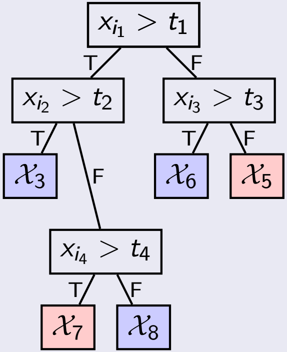
$i_3 = 1$



Splits using impurity criteria.

- sklearn: `tree.DecisionTreeClassifier`

Decision Tree

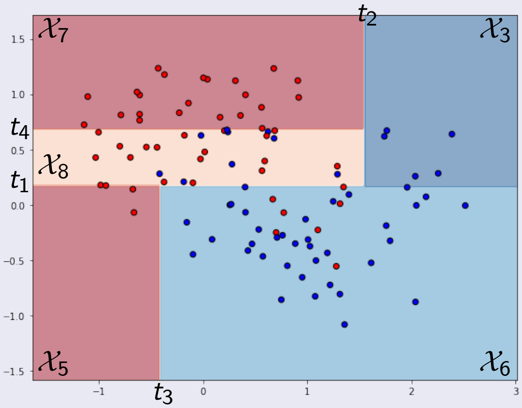


$i_1 = 2$

$i_2 = 1$

$i_3 = 1$

$i_4 = 2$

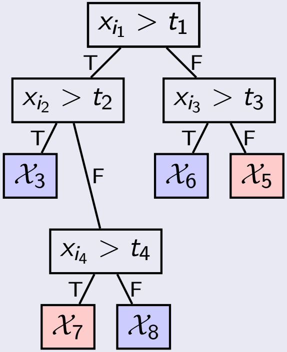


Splits using impurity criteria.

- sklearn: `tree.DecisionTreeClassifier`

Classical Supervised Learning Algorithms

Decision Tree



$i_1 = 2$

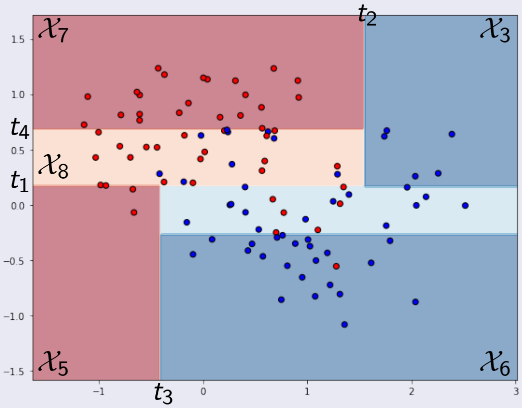
$i_2 = 1$

$i_3 = 1$

$i_4 = 2$

$i_5 = 2$

$\cdots$



Splits using impurity criteria.

- sklearn: `tree.DecisionTreeClassifier`

Let's use the following notations:

- $Pos = \sum_{i=1}^{n} \mathbb{1}_{\{y_i = y_+\}}$
- $Neg = \sum_{i=1}^{n} \mathbb{1}_{\{y_i = y_-\}}$

Gini Impurity index

$$G = 2\left(\frac{Pos}{n}\right)\left(\frac{Neg}{n}\right)$$

Shannon Entropy

$$S = -\frac{Pos}{n}\ln\left(\frac{Pos}{n}\right) - \frac{Neg}{n}\ln\left(\frac{Neg}{n}\right)$$

Note that if $Pos = 0$ (or $Neg = 0$), $G = S = 0$.
These impurity criteria are maximal when $Pos = Neg$: $G = \frac{2}{4} = \frac{1}{2}$, and $S = -\ln(\frac{1}{2}) = \ln(2)$.

Random Forest

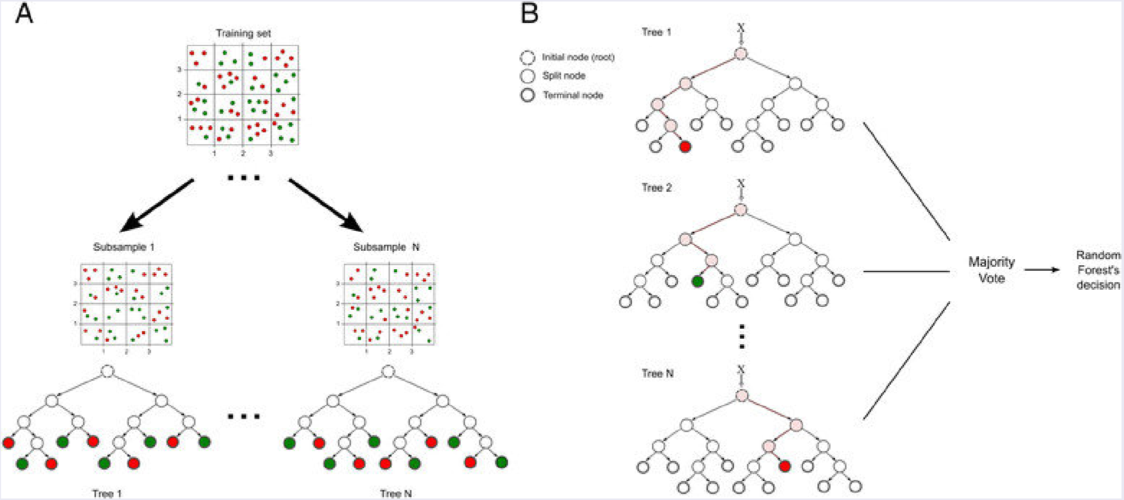


Figure from Machado et al. 2015,

▶ sklearn: `ensemble.RandomForestClassifier`

Linear Discriminant Analysis (LDA)

- Assumptions: $X \in \mathbb{R}^d$, $X \sim \mathcal{N}(\mu_y, \Sigma)$, $\forall y \in \mathcal{Y}$.
- Then, given $y \in \mathcal{Y}$, the density of X is:

$$f_y(x) = \frac{1}{(2\pi)^{\frac{d}{2}} \det(\Sigma)^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(x-\mu_y)^T \Sigma^{-1} (x-\mu_y)\right).$$

- Using the Bayes rule, assuming a prior probability $\mathbb{P}(Y = y)$, the posterior probability is:

$$\mathbb{P}(Y = y \mid X = x) = \frac{f_y(x)\mathbb{P}(Y = y)}{\sum_{y \in \mathcal{Y}} f_y(x)\mathbb{P}(Y = y)}.$$

- Decision $y_+ \Leftrightarrow \frac{\mathbb{P}(Y=y_+ \mid X=x)}{\mathbb{P}(Y=y_- \mid X=x)} \geqslant 1$, decision $y_- \Leftrightarrow \frac{\mathbb{P}(Y=y_+ \mid X=x)}{\mathbb{P}(Y=y_- \mid X=x)} < 1$.
- Decision function $\nu(x) = \ln\left(\frac{\mathbb{P}(Y=y_+ \mid X=x)}{\mathbb{P}(Y=y_- \mid X=x)}\right)$ [Reminder: $c(x) = y_+ \Leftrightarrow \nu(x) \geqslant 0$].

Decision function of LDA

Reminder:

- densities: $f_y(x) = \frac{1}{(2\pi)^{\frac{d}{2}} \det(\Sigma)^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(x-\mu_y)^T \Sigma^{-1} (x-\mu_y)\right)$,
- posterior probabilities: $\mathbb{P}(\,Y=y \mid X=x\,) = \frac{f_y(x)\mathbb{P}(\,Y=y\,)}{\sum_{y\in\mathcal{Y}} f_y(x)\mathbb{P}(\,Y=y\,)}$.

So, we can compute the decision function:

- $\frac{\mathbb{P}(\,Y=y_+ \mid X=x\,)}{\mathbb{P}(\,Y=y_- \mid X=x\,)} = \frac{f_{y_+}(x)\mathbb{P}(\,Y=y_+\,)}{f_{y_-}(x)\mathbb{P}(\,Y=y_-\,)}$.
- Decision function:

$$\begin{aligned}
\nu(x) &= \ln\left(\frac{\mathbb{P}(\,Y=y_+ \mid X=x\,)}{\mathbb{P}(\,Y=y_- \mid X=x\,)}\right) = \ln\left(\frac{f_{y_+}(x)}{f_{y_-}(x)}\right) + \ln\left(\frac{\mathbb{P}(\,Y=y_+\,)}{\mathbb{P}(\,Y=y_-\,)}\right) \\
&= \ln\left(\frac{\exp\left(-\frac{1}{2}(x-\mu_{y_+})^T \Sigma^{-1} (x-\mu_{y_+})\right)}{\exp\left(-\frac{1}{2}(x-\mu_{y_-})^T \Sigma^{-1} (x-\mu_{y_-})\right)}\right) + \ln\left(\frac{\mathbb{P}(\,Y=y_+\,)}{\mathbb{P}(\,Y=y_-\,)}\right)
\end{aligned}$$

Decision function of LDA

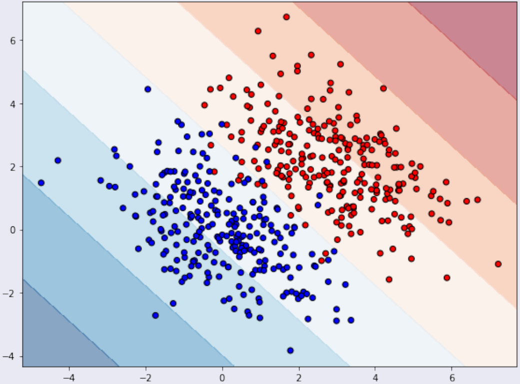
$$
\begin{aligned}
\nu(x) = \ln\left(\frac{\exp\left(-\frac{1}{2}(x-\mu_{y_+})^T\Sigma^{-1}(x-\mu_{y_+})\right)}{\exp\left(-\frac{1}{2}(x-\mu_{y_-})^T\Sigma^{-1}(x-\mu_{y_-})\right)}\right) + \ln\left(\frac{\mathbb{P}(Y=y_+)}{\mathbb{P}(Y=y_-)}\right) \\
= \quad -\frac{1}{2}(x-\mu_{y_+})^T\Sigma^{-1}(x-\mu_{y_+}) + \frac{1}{2}(x-\mu_{y_-})^T\Sigma^{-1}(x-\mu_{y_-}) + \ln\left(\frac{\mathbb{P}(Y=y_+)}{\mathbb{P}(Y=y_-)}\right) \\
= \quad \frac{1}{2}\Big(-x^T\Sigma^{-1}x + \mu_{y_+}^T\Sigma^{-1}x + x^T\Sigma^{-1}\mu_{y_+} - \mu_{y_+}^T\Sigma^{-1}\mu_{y_+} \\
+x^T\Sigma^{-1}x - \mu_{y_-}^T\Sigma^{-1}x - x^T\Sigma^{-1}\mu_{y_-} + \mu_{y_-}^T\Sigma^{-1}\mu_{y_-}\Big) + \ln\left(\frac{\mathbb{P}(Y=y_+)}{\mathbb{P}(Y=y_-)}\right) \\
= \quad \tfrac{1}{2} \quad \Big((\mu_{y_+}-\mu_{y_-})^T\Sigma^{-1}x + x^T\Sigma^{-1}(\mu_{y_+}-\mu_{y_-}) + (\mu_{y_-}-\mu_{y_+})^T\Sigma^{-1}(\mu_{y_-}+\mu_{y_+})\Big) + \ln(\ldots) \\
= \quad (\mu_{y_+}-\mu_{y_-})^T\Sigma^{-1}x + \frac{1}{2}(\mu_{y_-}-\mu_{y_+})^T\Sigma^{-1}(\mu_{y_-}+\mu_{y_+}) + \ln\left(\frac{\mathbb{P}(Y=y_+)}{\mathbb{P}(Y=y_-)}\right).
\end{aligned}
$$

Linear Discriminant Analysis (LDA)

The decision function is linear in x.

Prior & Gaussian parameter estimations:

- $\frac{\mathbb{P}(Y=y_+)}{\mathbb{P}(Y=y_-)} = \frac{Pos}{Neg}$.
- $\widehat{\mu_{y_+}} = \frac{1}{Pos}\sum_{i=1}^{n} \mathbb{1}_{\{y_i=y_+\}} x_i$,
  $\widehat{\mu_{y_-}} = \frac{1}{Neg}\sum_{i=1}^{n} \mathbb{1}_{\{y_i=y_-\}} x_i$.
- $\widehat{\Sigma} = \frac{1}{n-2}\left(\widehat{\Sigma_+} + \widehat{\Sigma_-}\right)$,
  $\widehat{\Sigma_+} = \sum_{i=1}^{n}(x_i-\mu_{y_+})^T(x_i-\mu_{y_+})\mathbb{1}_{\{y_i=y_+\}}$,
  $\widehat{\Sigma_-} = \sum_{i=1}^{n}(x_i-\mu_{y_-})^T(x_i-\mu_{y_-})\mathbb{1}_{\{y_i=y_-\}}$.

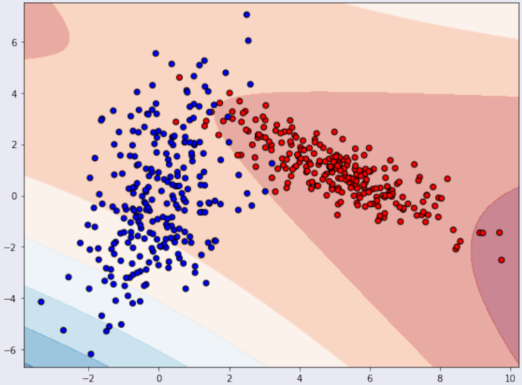


$$\Rightarrow \nu(x) = (\widehat{\mu_{y_+}} - \widehat{\mu_{y_-}})^T\Sigma^{-1}x + \frac{1}{2}(\widehat{\mu_{y_-}} - \widehat{\mu_{y_+}})^T\Sigma^{-1}(\widehat{\mu_{y_-}} + \widehat{\mu_{y_+}}) + \ln\left(\frac{Pos}{Neg}\right).$$

▶ sklearn: `discriminant_analysis.LinearDiscriminantAnalysis`

Quadratic Discriminant Analysis

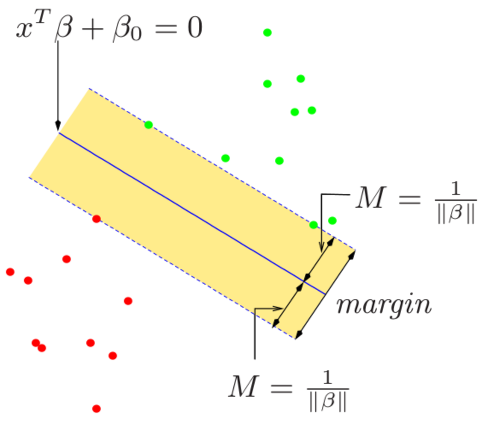
- Now the covariance matrix depends on the class: Σ_y.
- Other assumptions hold.
- In this case, the decision function is **quadratic** in x.



- sklearn: `discriminant_analysis.LinearDiscriminantAnalysis`

Support Vector Machine

- Decision function $\nu(x) = x^{\beta} + \beta_0$,
  with $\beta \in \mathbb{R}^d$, $\|\beta\| = 1$ and $\beta_0 \in \mathbb{R}$.
- $x^T\beta + \beta_0 = 0 \Leftrightarrow$ hyperplane orthogonal to β.
- $|x^T\beta + \beta_0| =$ distance $x \leftrightarrow$ hyperplane
- Encoding $y_+ = 1$, $y_- = -1$.
- By choosing $\beta \in \mathbb{R}^d$, $\beta_0 \in \mathbb{R}$,
  maximize the margin M
  subject to $y_i(x_i^T \frac{\beta}{\|\beta\|} + \frac{\beta_0}{\|\beta\|}) \geqslant M$, $\forall 1 \leqslant i \leqslant n$,
  i.e. subject to $y_i(x_i^T\beta + \beta_0) \geqslant M\,\|\beta\|$, $\forall i$.
  $\Updownarrow$ with $M\,\|\beta\| = 1$
  minimize $\|\beta\|$
  subject to $y_i(x_i^T\beta + \beta_0) \geqslant 1$, $\forall 1 \leqslant i \leqslant n$.

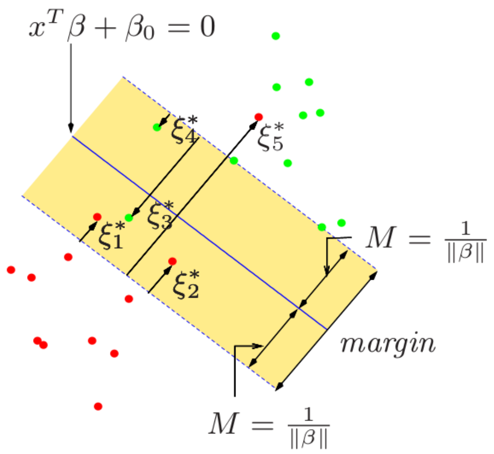


- sklearn: `svm.SVC`
- sklearn: `svm.SVR`

Support Vector Machine

If classes overlap, introduce ξ_i.

- Minimize $\|\beta\|$ subject to
  $$\begin{cases} y_i(x_i T\beta + \beta_0) \geqslant 1 - \xi_i, \ \forall 1 \leqslant i \leqslant n \\ \xi_i \geqslant 0, \ \text{and} \ \sum_i \xi \leqslant constant \end{cases}$$
  $\Updownarrow$ convex optimization [BBV04]
  minimize $\frac{1}{2}\|\beta\|^2 + C\sum_{i=1}^{n}\xi_i$
  subject to $\xi \geqslant 0$, $y_i(x_i^T\beta + \beta_0) \geqslant 1, \forall i$.
- High (resp. low) $C > 0$ prioritizes a good classification (resp. a large margin).



- sklearn: `svm.SVC`
- sklearn: `svm.SVR`

More details in [HTF09].

Support Vector Machine and kernels

Using a kernel $K : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$, the resulting decision function has the non linear form

$$\nu(x) = \sum_{i=1}^{n} \alpha_i y_i K(x, xi) + \beta_0.$$

Some popular kernels

- polynomial: $K(x, x') = (1 + \langle x, x' \rangle)^d$,
- radial basis: $K(x, x') = \exp(-\gamma \left\| x - x' \right\|^2)$,
- sigmoid: $\frac{1}{1+e^{-\langle x, x' \rangle}}$.

▶ sklearn: `svm.SVC` ▶ sklearn: `svm.SVR`

More details in [HTF09].

Outline

Remarks and tools for BCIs

Brain Computer Interfaces

- Difficulties with physiological data (e.g. EEG):
  - signal-to-noise ratio very low,
  - few small datasets (time/money consuming experiments),
  - high dimensionality,
  - non-stationary,
  - variability over humans (participants),
  - variability over time (sessions),
  - variability over experiments (settings).
- Different problems, increasing difficulty in prediction:
  - within-recording-session prediction (intra-session),
  - across-session within-subject prediction (intra-subject),
  - across-subject prediction (inter-subject).

Remarks and tools for BCIs

EEG tools and BCI evaluation

- mne.tools
- moabb.neurotechx.com

Institut Supérieur de l'Aéronautique et de l'Espace
10 avenue Édouard Belin – BP 54032
31055 Toulouse Cedex 4 – France
Phone: +33 5 61 33 80 80

www.isae-supaero.fr