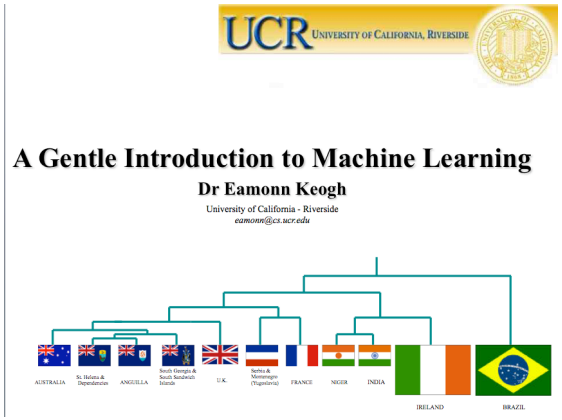
classification
(and then some…)
algorithms

Carlos Soares

[based on materials kindly provided by E. Keogh and J.M. Moreira]

U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO



reference materials

U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

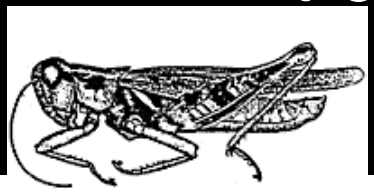
- JMM et al. ch. 8+9+10+12
  - also covers regression which will only be addressed later

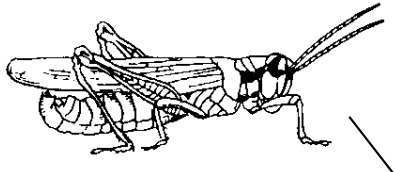
plan

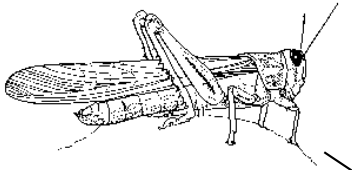
U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

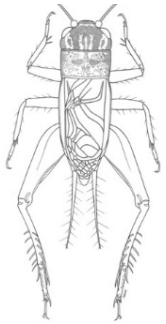
- understanding ML algorithms
  - algorithm vs geometric intuition
  - we've got neural nets, so why waste time?
- some popular algorithms
  - Linear Classifiers
  - Nearest Neighbors
  - Decision Trees
  - Naive Bayes
  - Neural Networks
  - Support Vector Machines
- ... with hyperparameters that may have to be tuned

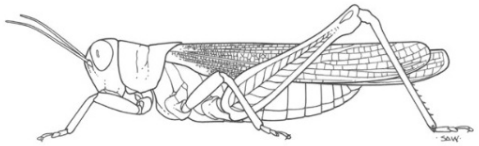
running (hopping?) example

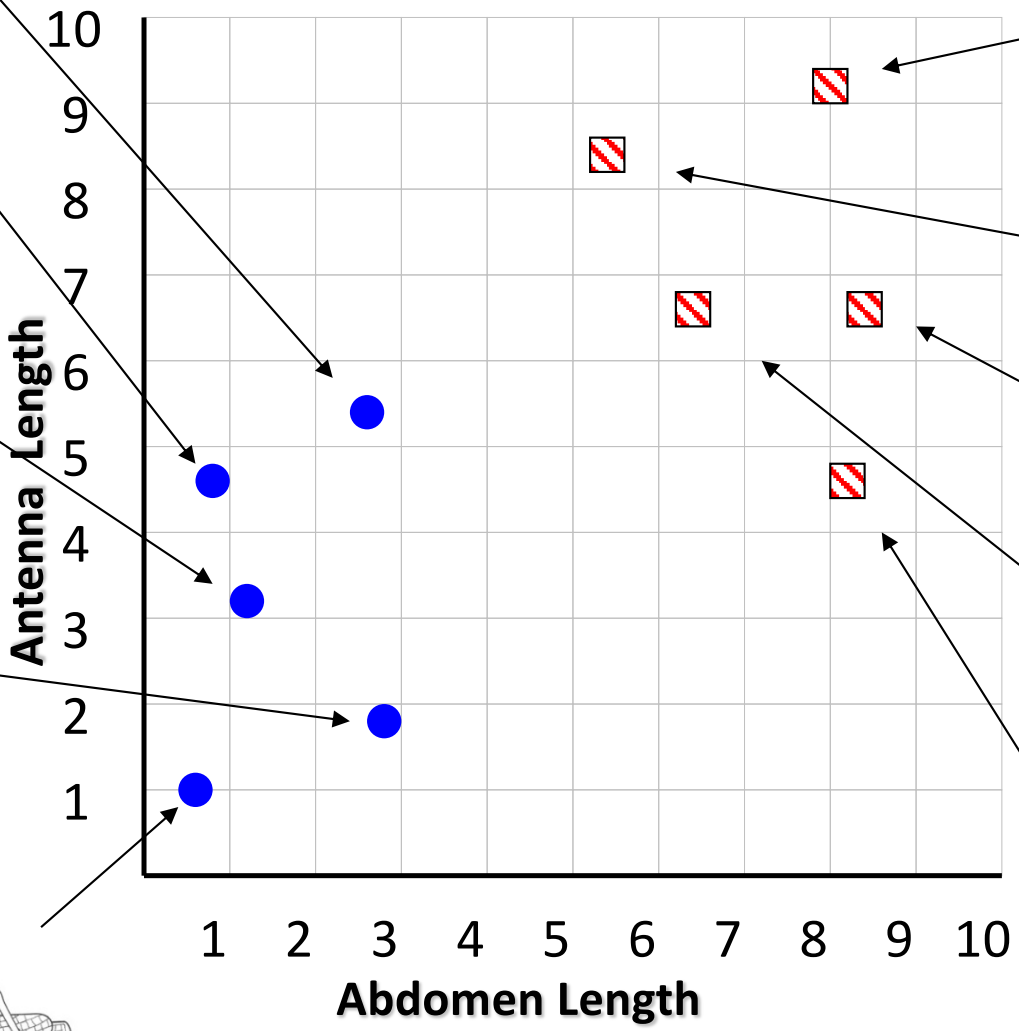


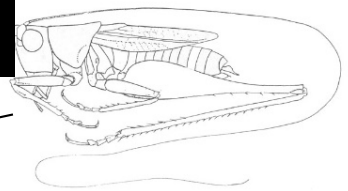


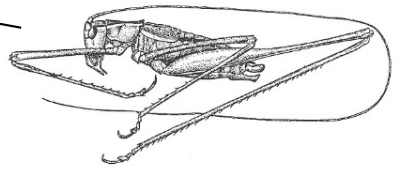


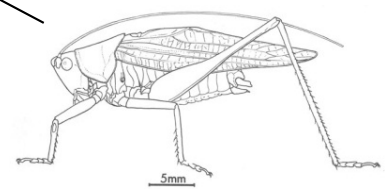


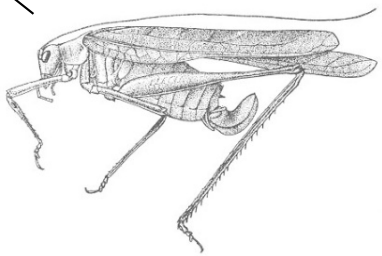












Grasshoppers

Katydids

LD vs DT: algorithm

U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

LD

1. Compute the d-dimensional mean vectors for the different classes from the dataset.
2. Compute the scatter matrices (in-between-class and within-class scatter matrix).
3. Compute the eigenvectors ($\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_d$) and corresponding eigenvalues ($\boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2, \ldots, \boldsymbol{\lambda}_d$) for the scatter matrices.
4. Sort the eigenvectors by decreasing eigenvalues and choose k eigenvectors with the largest eigenvalues to form a $d \times k$ dimensional matrix $\boldsymbol{W}$ (where every column represents an eigenvector).
5. Use this $d \times k$ eigenvector matrix to transform the samples onto the new subspace. This can be summarized by the matrix multiplication: $\boldsymbol{Y} = \boldsymbol{X} \times \boldsymbol{W}$ (where $\boldsymbol{X}$ is a $n \times d$-dimensional matrix representing the n samples, and $\boldsymbol{y}$ are the transformed $n \times k$-dimensional samples in the new subspace).

https://sebastianraschka.com/Articles/2014_python_lda.html

DT

Algorithm Hunt decision tree induction algorithm.

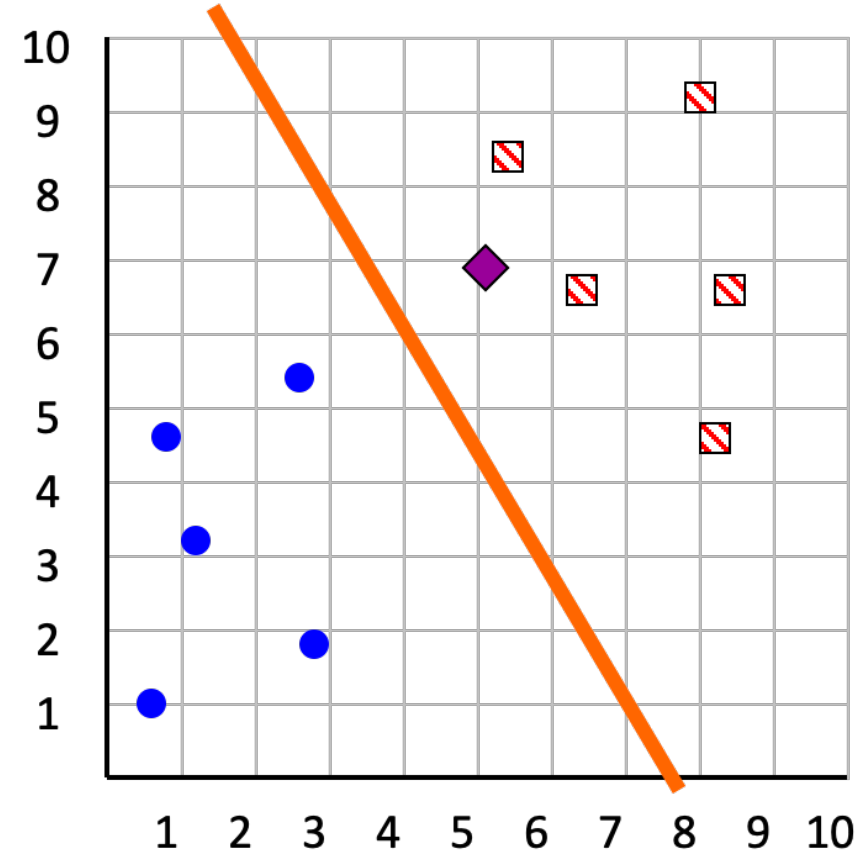
```
1: INPUT D_train current node training set
2: INPUT p the impurity measure
3: INPUT n the number of objects in the training set
4: if all objects in D_train belongs to the same class y then
5:     The current node is a leaf node labeled with class y
6: else
7:     Select a predictive attribute to split D_train using the impurity measure p
8:     Split D_train in subsets according to its current values
9:     Apply Hunt algorithm to each subset
```

JMM et al (ch. 10)

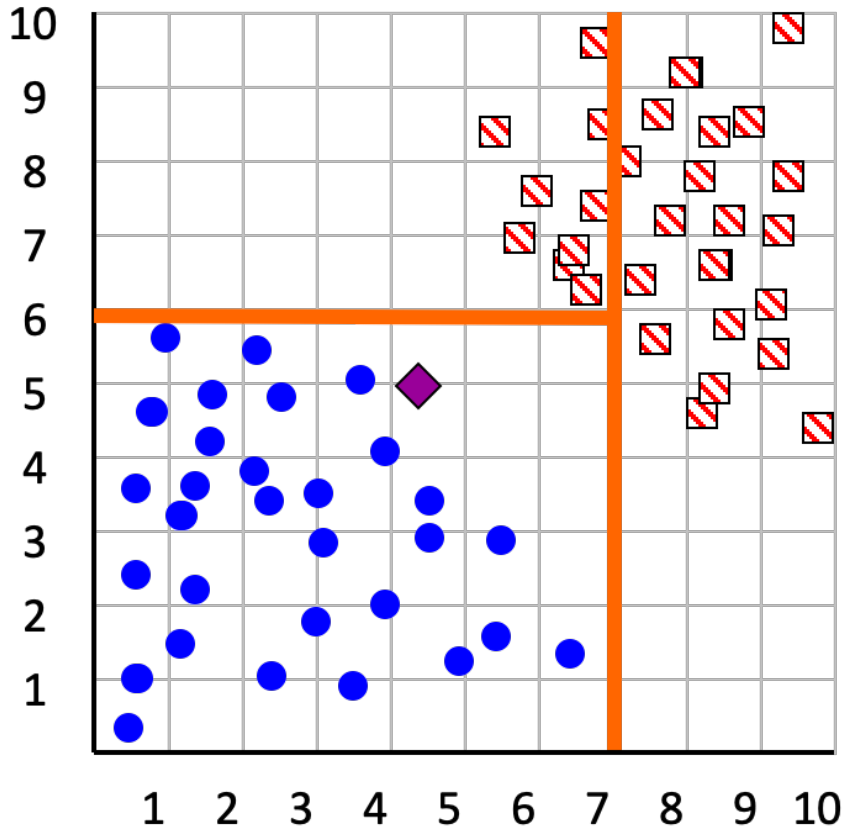
LD vs DT: geometric intuition

U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

LD

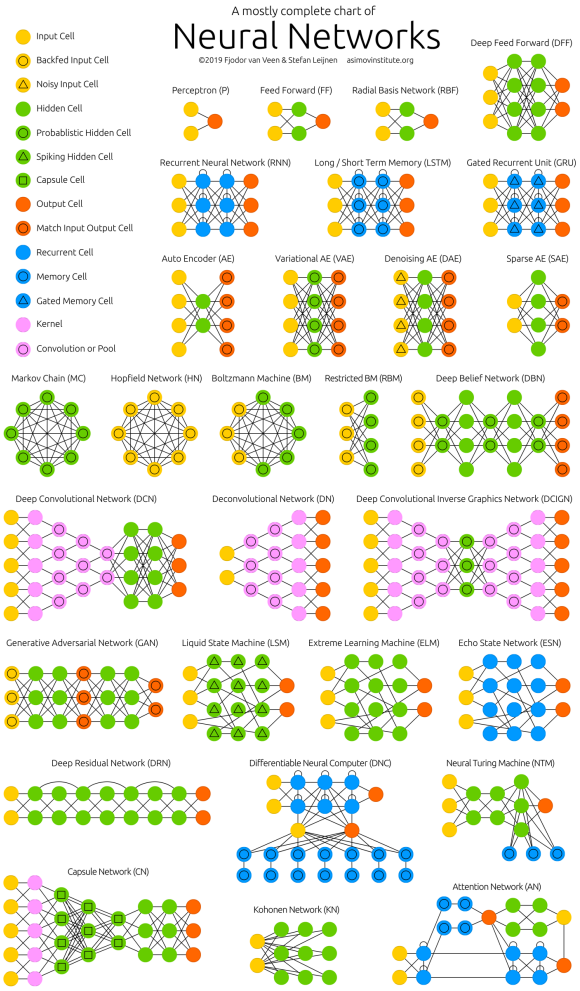


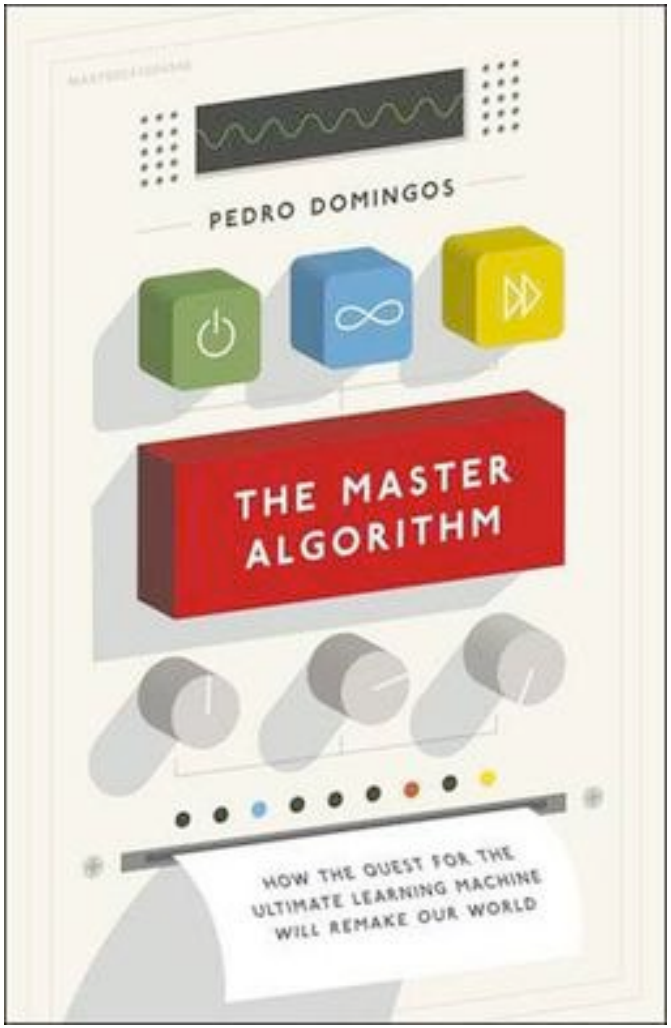
DT



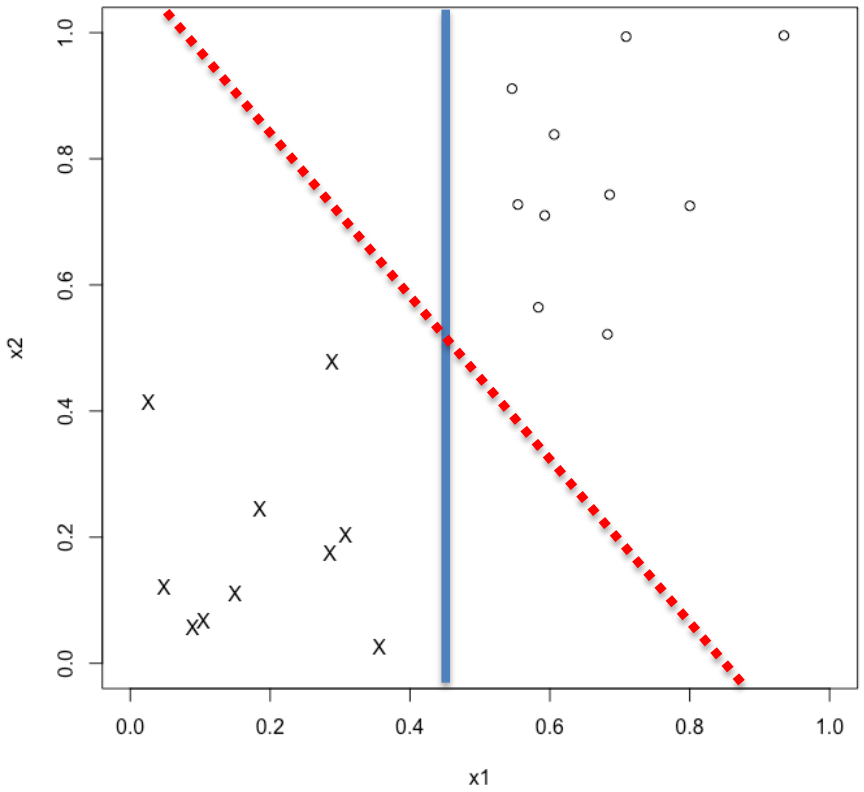
wait, is this really a problem?

U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO





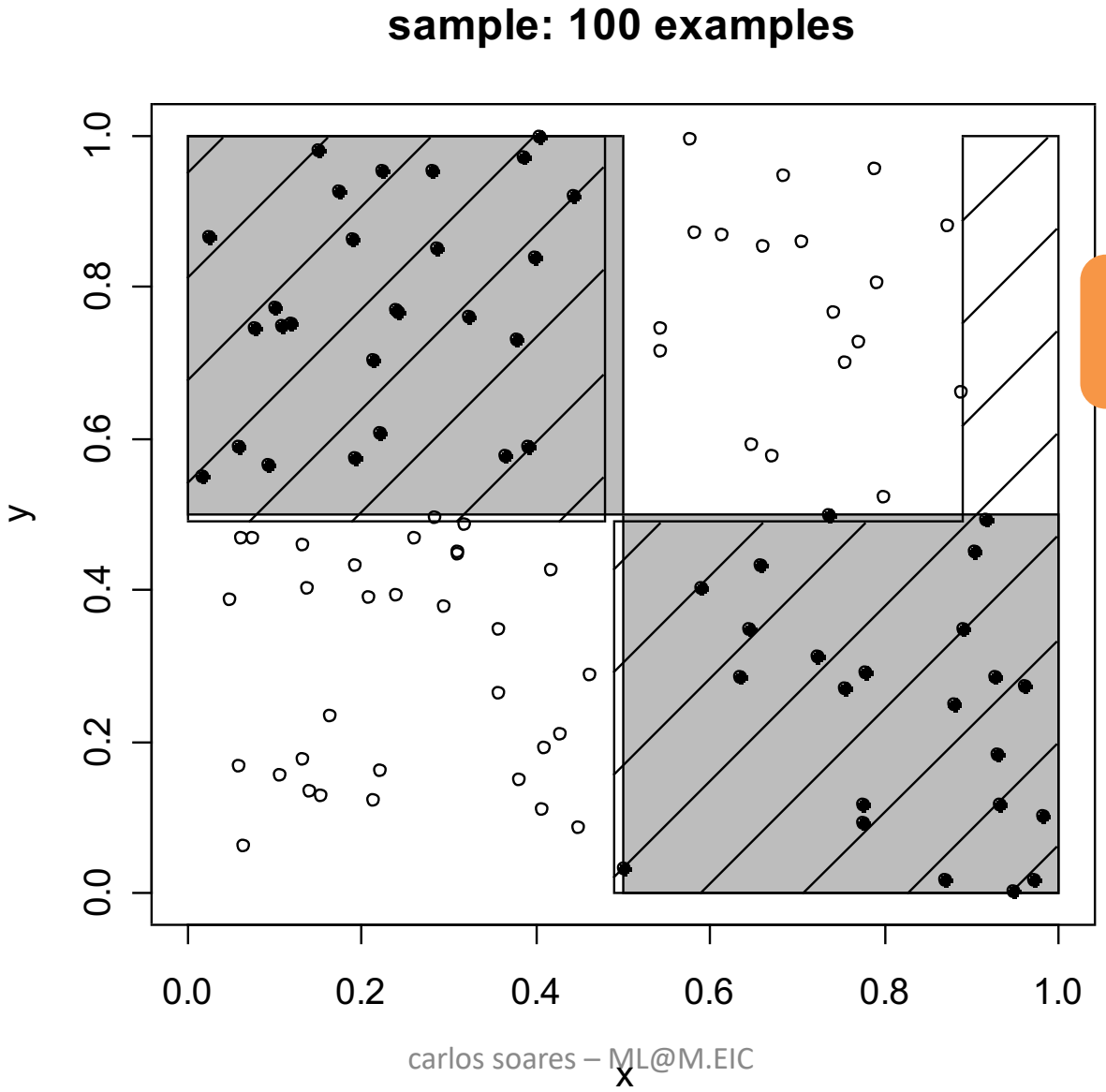
bias: can't live with it



- given
  - dataset
  - learning algorithm
- not every model is possible
  - e.g. DT and LR
  - … but not DT and LR

DT may be suitable for a problem

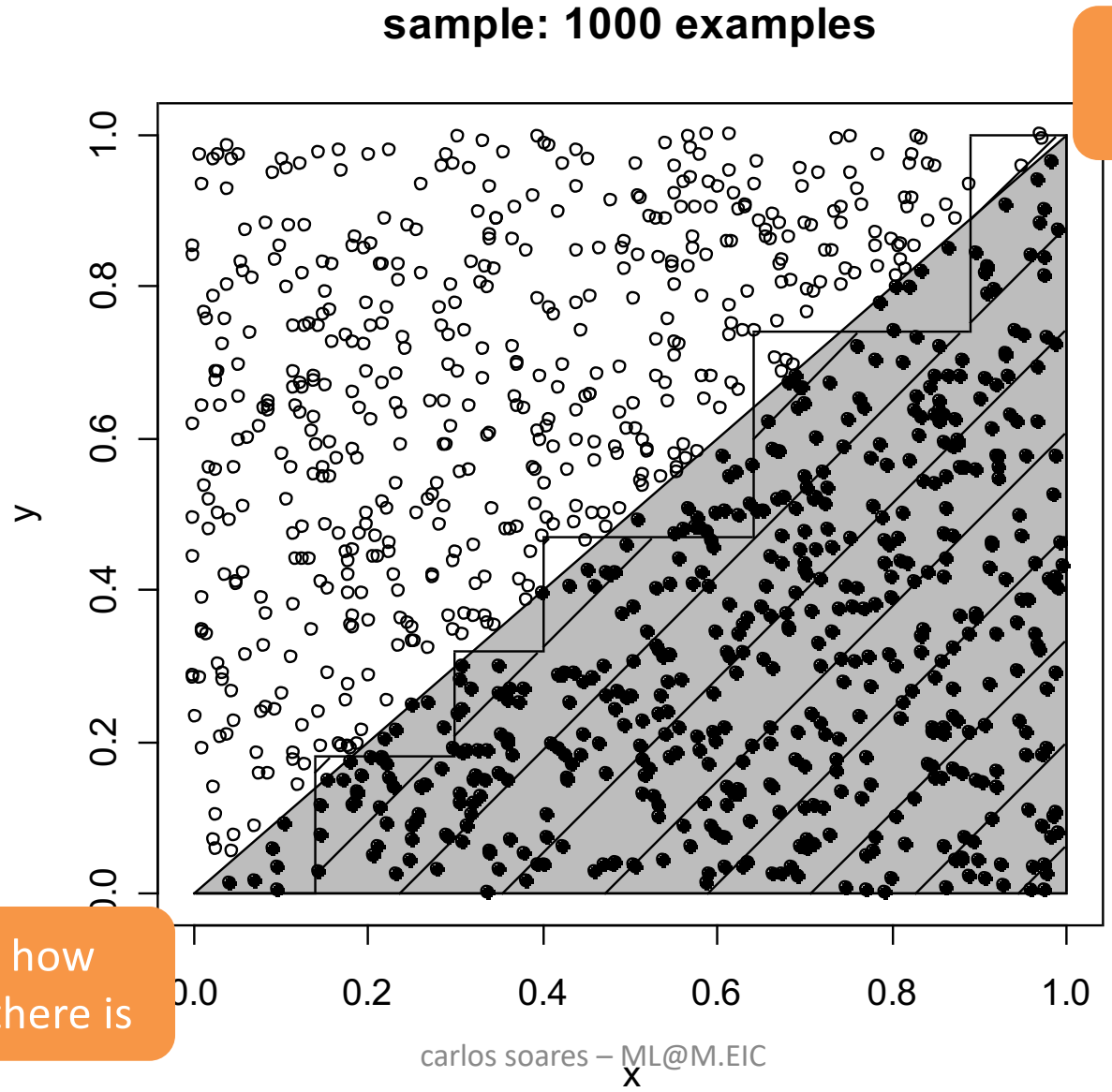
U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO



even if not perfect

… or not

U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO



… or how simple it is

no matter how much data there is

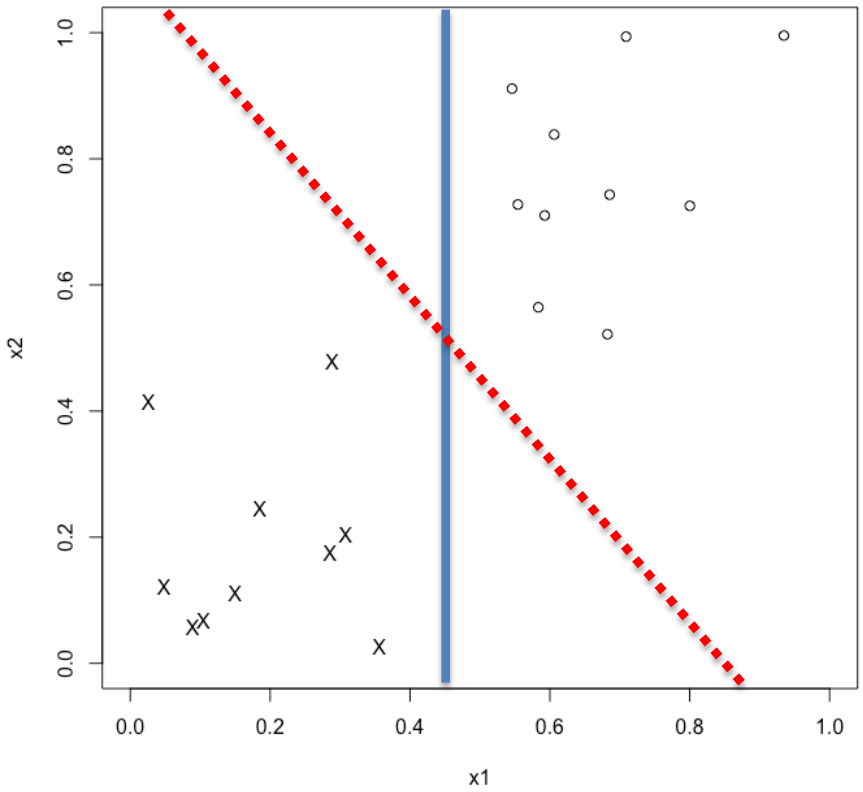
… and can't live without it

U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

- bias-free learning is futile (Mitchell 97, Ch. 2)
  - an algorithm that assumes nothing concerning the function it is trying to learn has no rational basis to classify unknown cases

- bias = criteria to prefer one model relative to another
- … so, how to select the best model if all models are considered equally suitable?

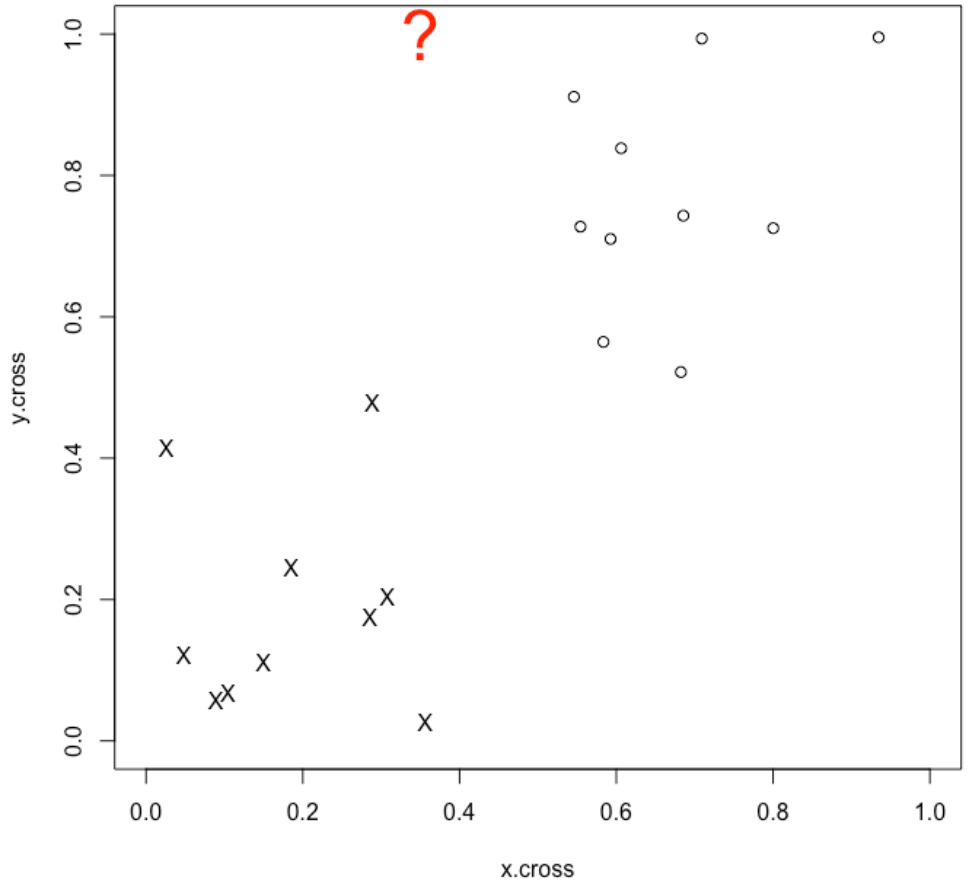
the bias-free algorithm



- … can learn any model
  - e.g. the DT and the LR
- … but doesn't have any preference for one over the other
  - … or for one over the other

… right?

U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO



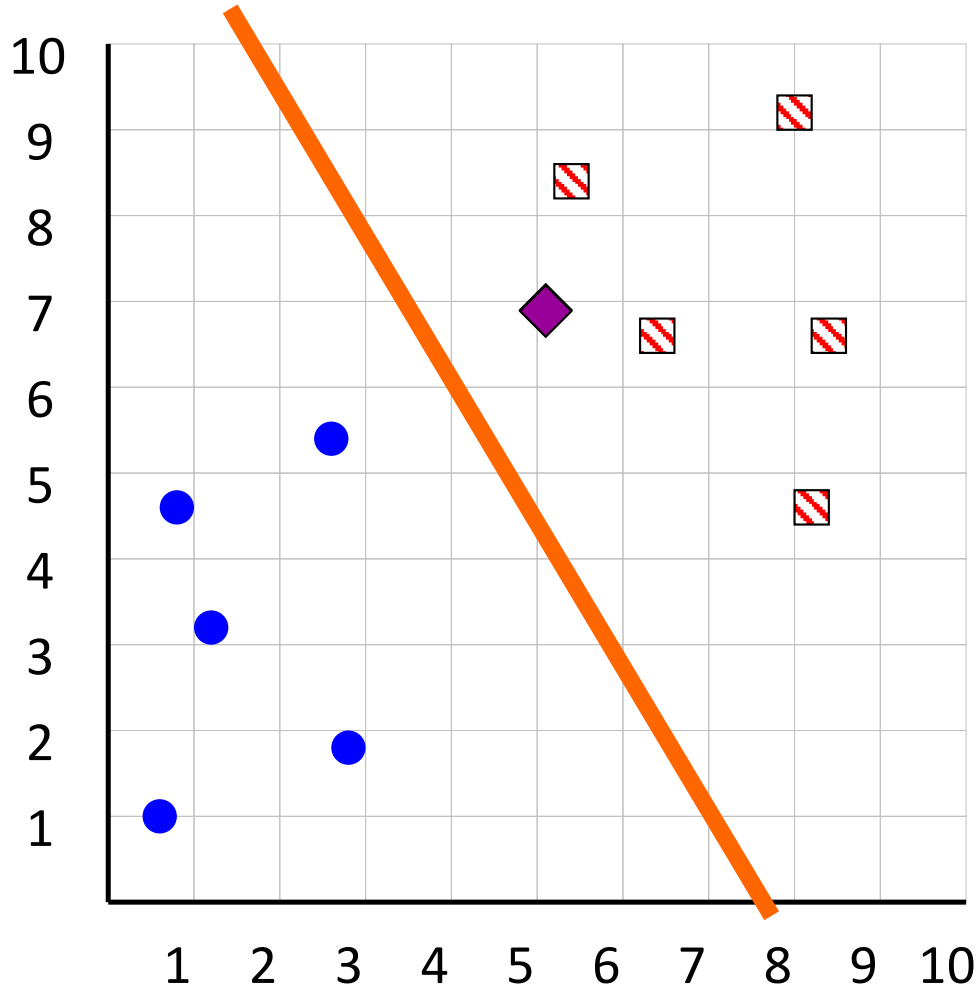
gps

U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

- understanding ML algorithms
  - algorithm vs geometric intuition
  - we've got neural nets, so why waste time?
- some popular algorithms
  - Linear Classifiers
  - Nearest Neighbors
  - Decision Trees
  - Naive Bayes
  - Neural Networks
  - Support Vector Machines
- ... with hyperparameters that may have to be tuned

Simple Linear Classifier

R.A. Fisher
1890-1962



If previously unseen instance above the line
then
class is **Katydid**
else
class is **Grasshopper**

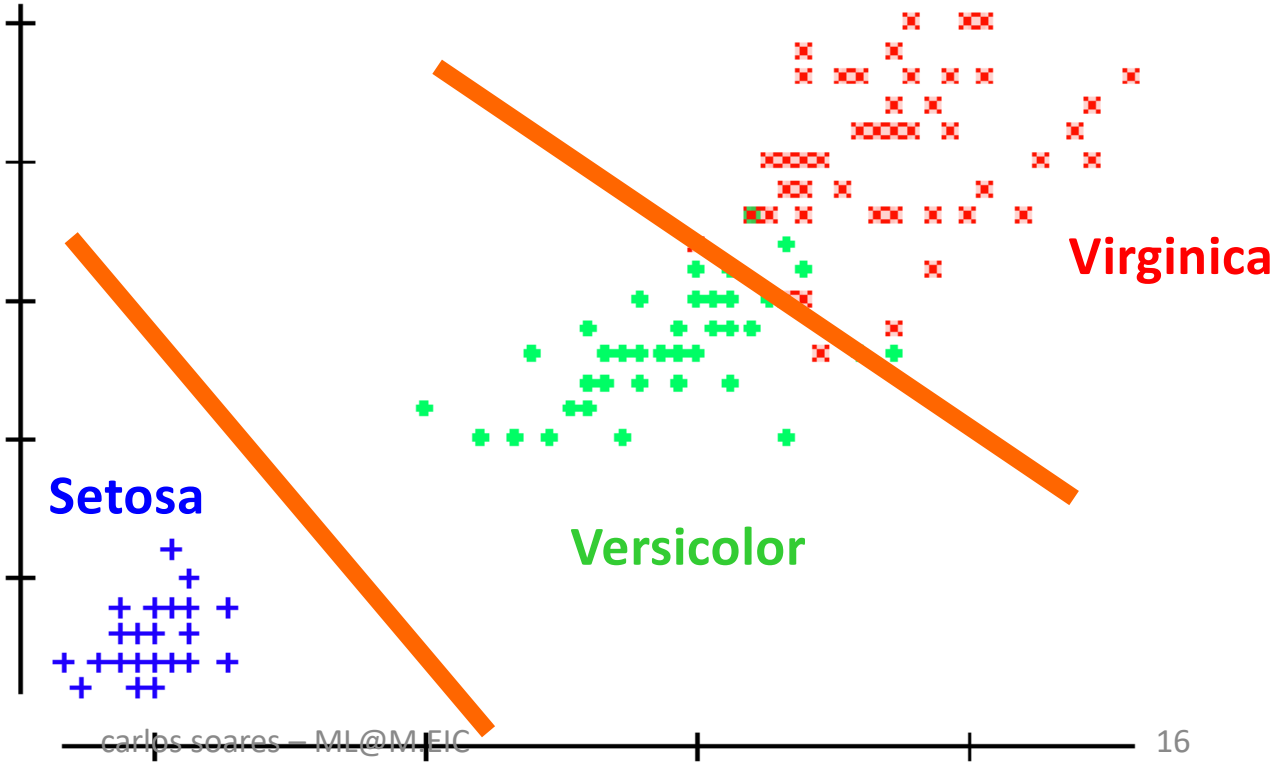
Katydids
Grasshoppers

Linear Classifier for 3+ Classes

U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

- linear classifier for N classes
- fit N-1 lines
  - **Setosa** vs **Virginica**/**Versicolor**
  - then
  - **Virginica** vs **Versicolor**.

If petal width > 3.272 – (0.325 * petal length)
then class = **Virginica**
Elseif petal width…



LD: pros and cons

U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

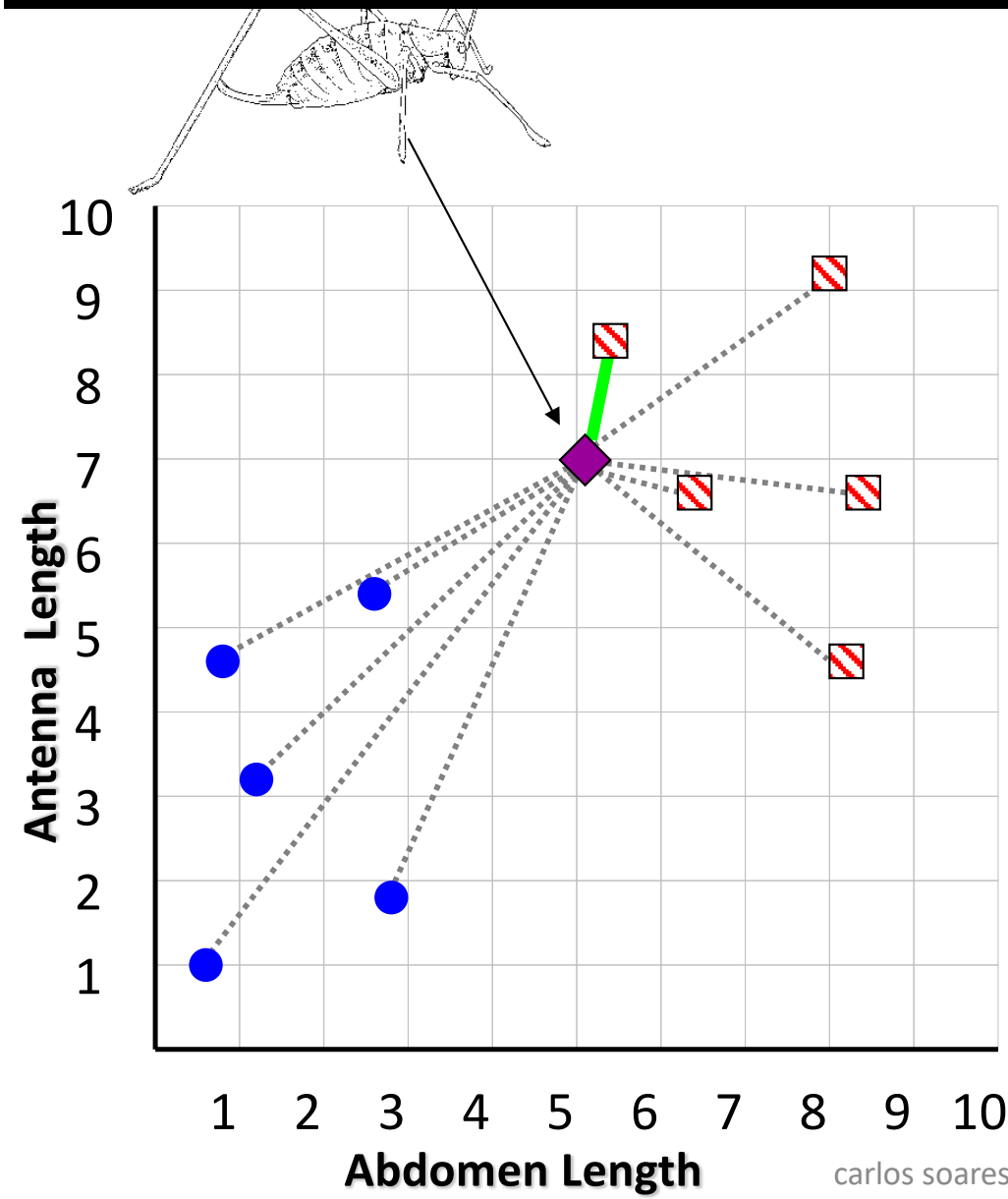
pros

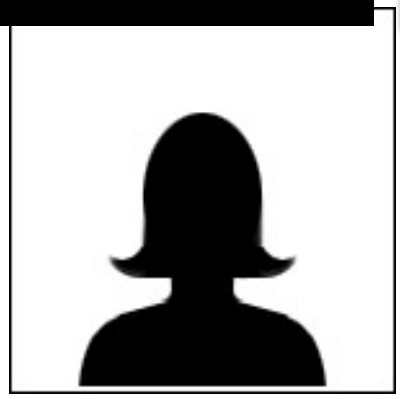
- interpretable model
- easy to configure

cons

- not directly applicable to qualitative features
- interpretability affected by correlated features
- sensitive to outliers

Nearest Neighbor Classifier





Evelyn Fix
1904-1965

Joe Hodges
1922-2000

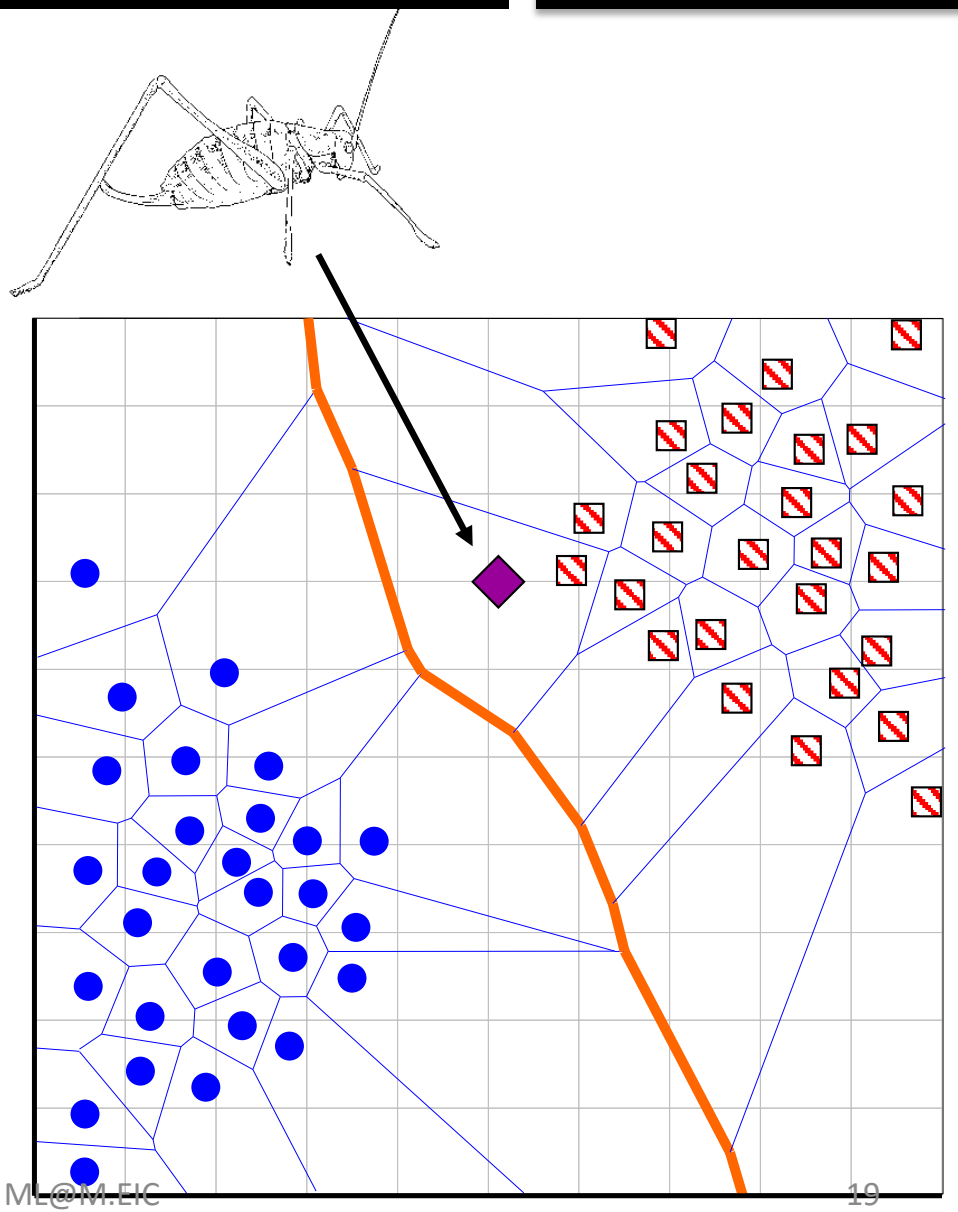
If the **nearest** instance to the previously unseen instance **is a Katydid**
    class is **Katydid**
else
    class is **Grasshopper**

Katydids
Grasshoppers

Geometrical Interpretation of Nearest Neighbor Classifier

U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

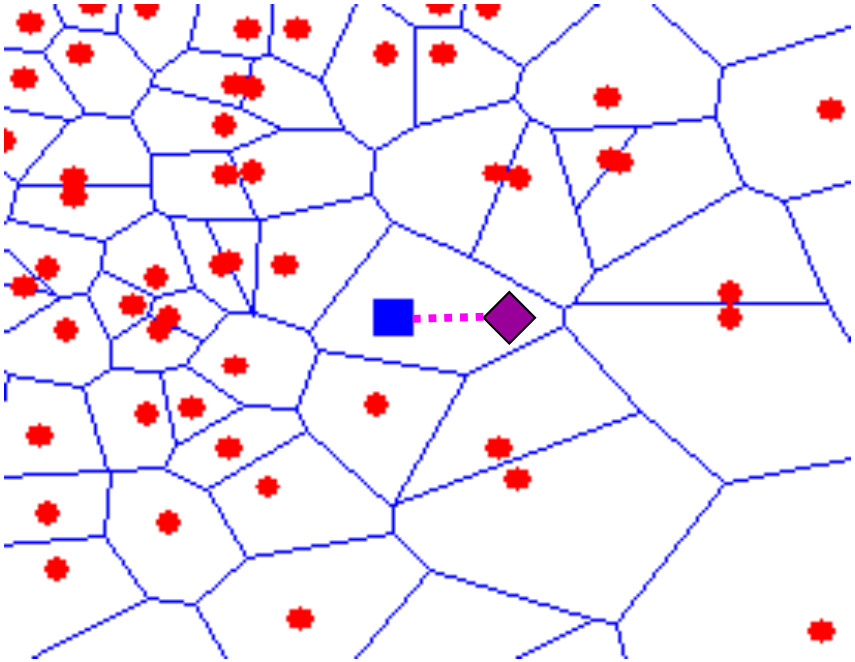
- decision surface of a NN classifier
  - also called Dirichlet Tessellation,
  - … Voronoi diagram
  - … or Theissen regions
- implicit
  - never defined explicitely
  - divide the space into regions “belonging” to each instance
  - … and corresponding class



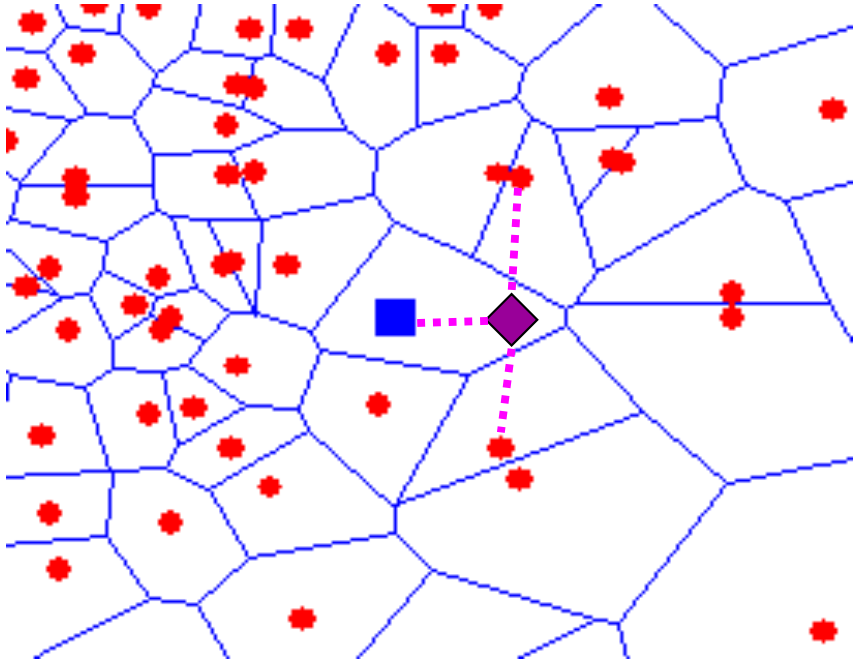
KNN Algorithm

U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

- generalization of the nearest neighbor algorithm
  - find the nearest K instances
  - each one represents a vote
- K is typically chosen to be an odd number



K = 1

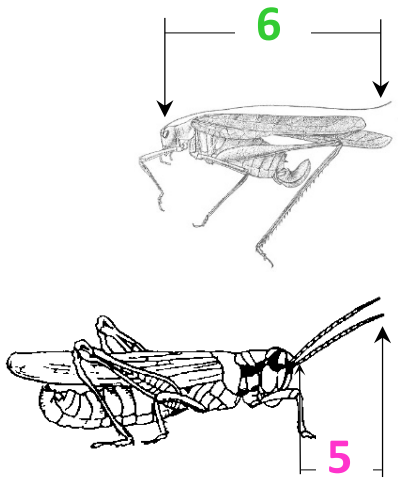


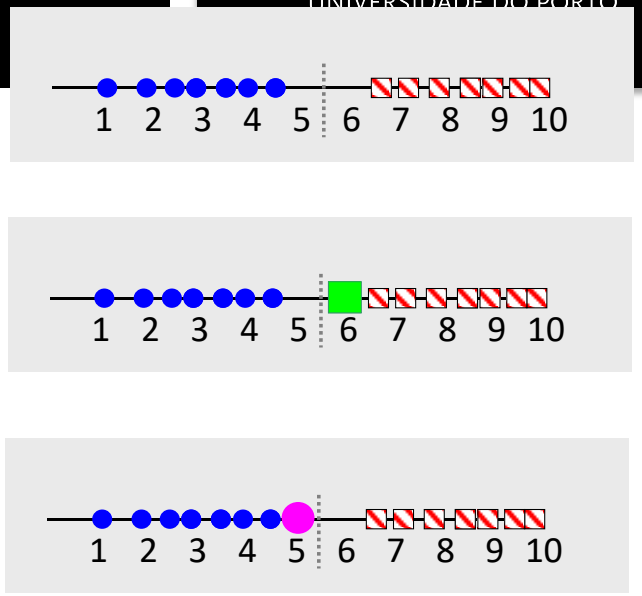
K = 3

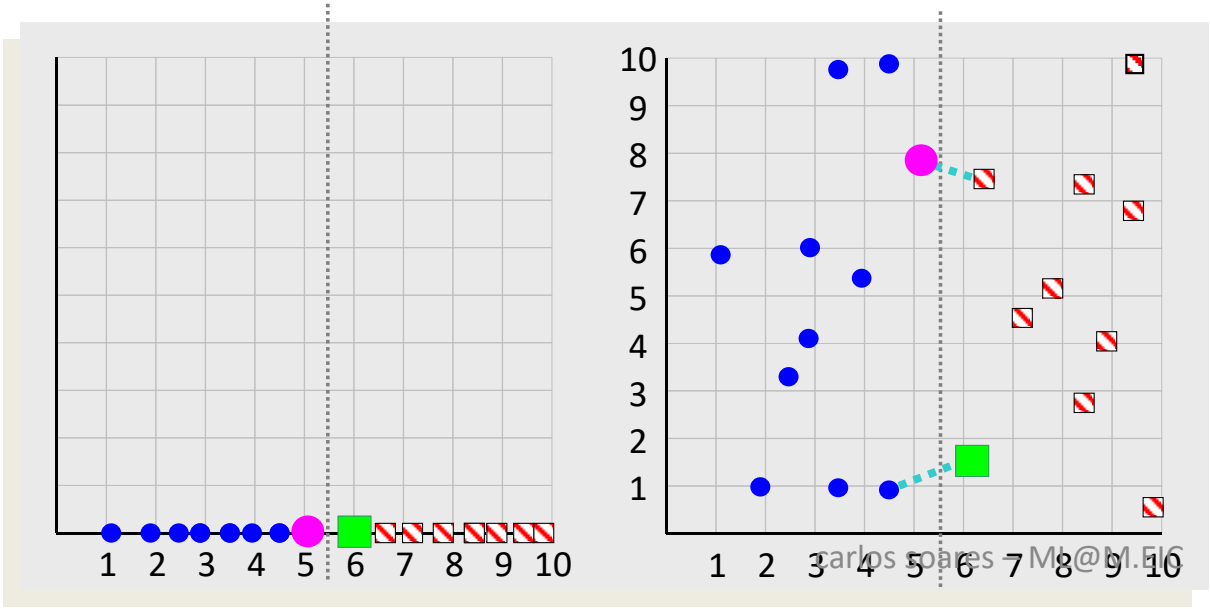
Sensitivity to Irrelevant Attributes (1/2)

Suppose the following is true, if an insects antenna is longer than 5.5 it is a **Katydid**, otherwise it is a **Grasshopper**.

Using just the antenna length we get perfect classification!







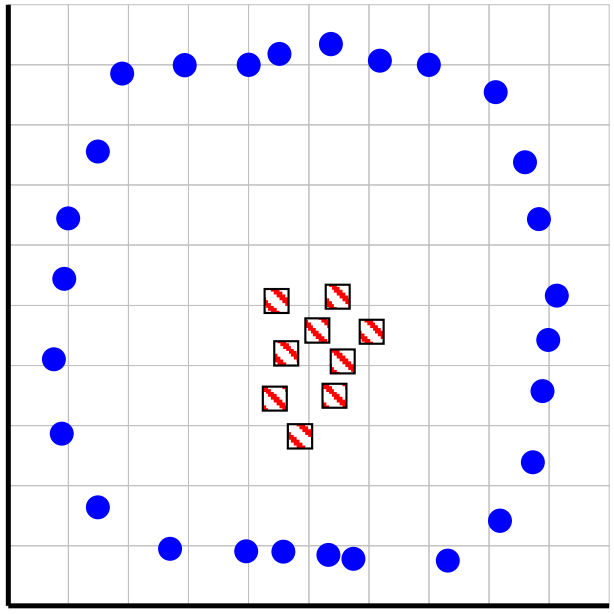
Suppose however, we add in an **irrelevant** feature, for example the insects mass.

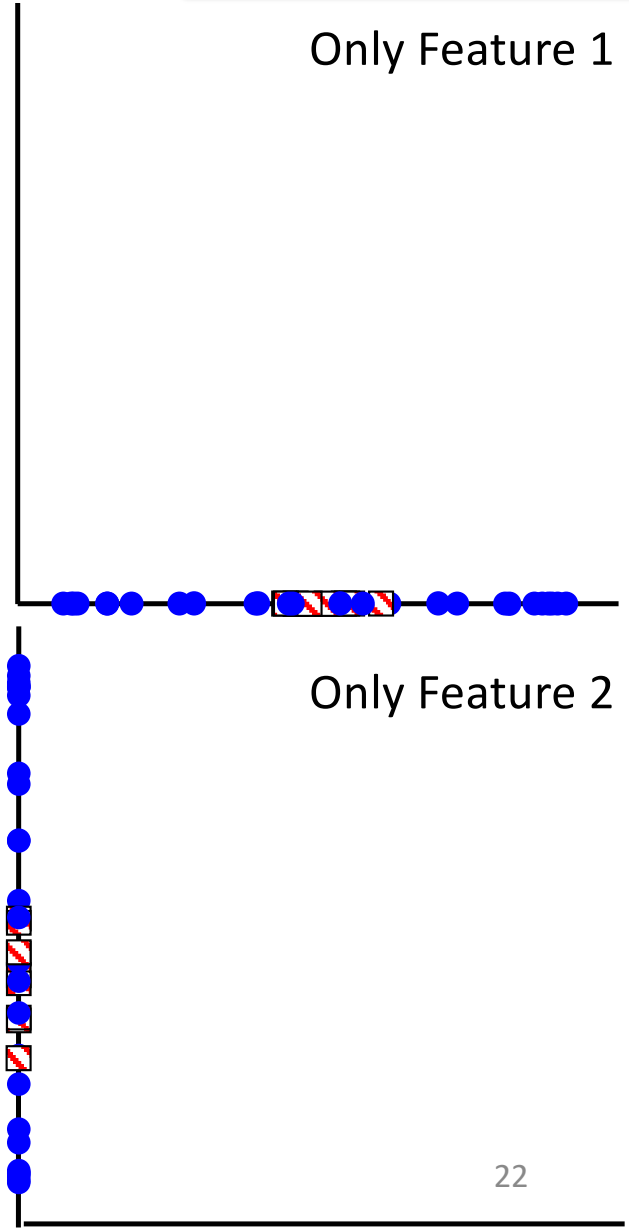
Using both the antenna length and the insects mass with the 1-NN algorithm we get the wrong classification!

why search over feature subsets can fail

U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

- problem
  - data with 100 features
  - Features 1 and 2 give perfect classification
  - … remaining 98 are irrelevant
- poor results
  - 100 features
  - Feature 1 alone
  - Feature 2 alone
- only one subset gives good results
  - $2^{100} - 1$ possible subsets





NN: pros and cons

U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

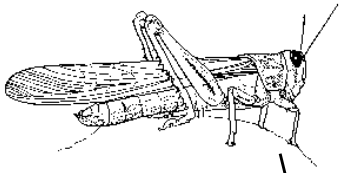
pros

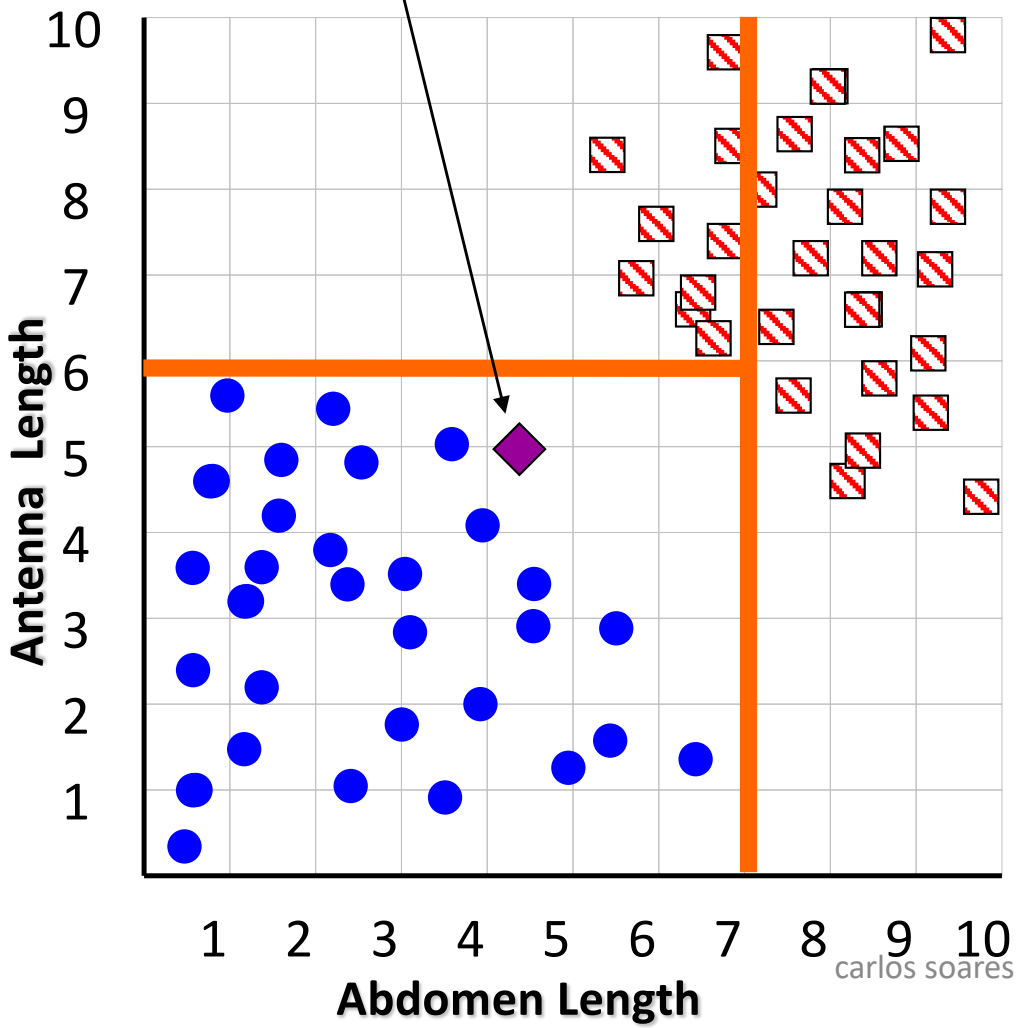
- simple to implement
- handles correlated features
  - Arbitrary class shapes
- defined for any distance measure

cons

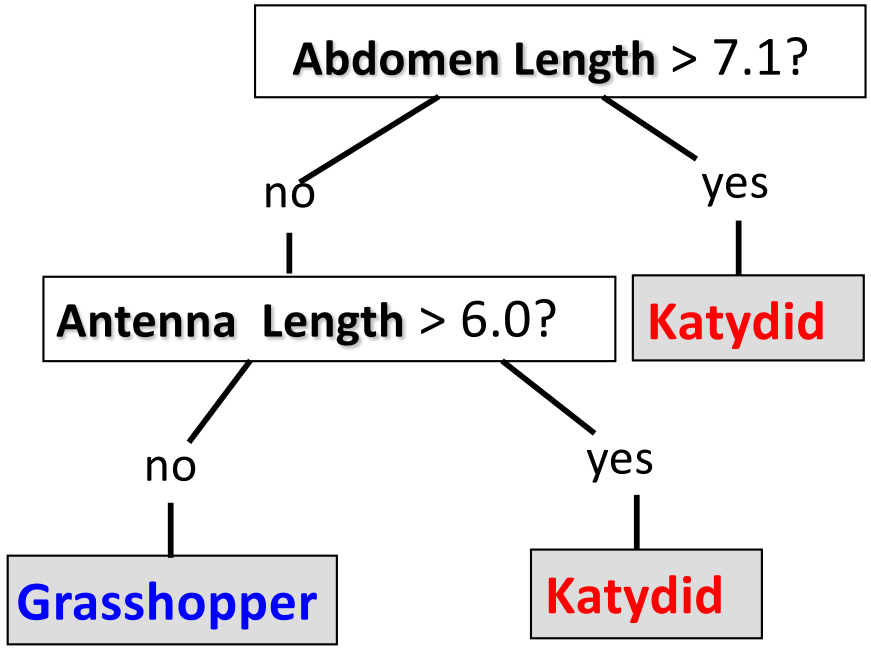
- very sensitive to irrelevant features
- slow classification time for large datasets
- works best for real valued datasets
- hard to interpret

Decision Tree Classifier





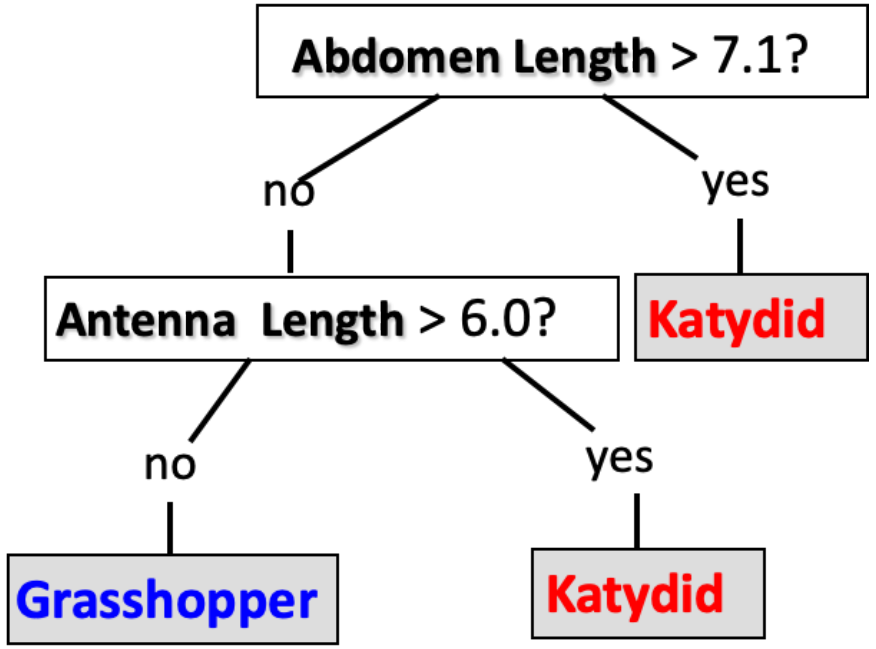
Ross Quinlan



avoiding overfitting in DT

U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

- prepruning
  - halt tree construction early
- postpruning
  - get a sequence of progressively pruned trees
    - removing branches from a "fully grown" tree
  - use a set of data different from the training data to decide which is the "best pruned tree"



DT: pros and cons

U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

pros

- easy to read
- robust to many data anomalies
  - outliers
  - missing values
  - correlated attributes
  - irrelevant attributes
  - scale of attributes

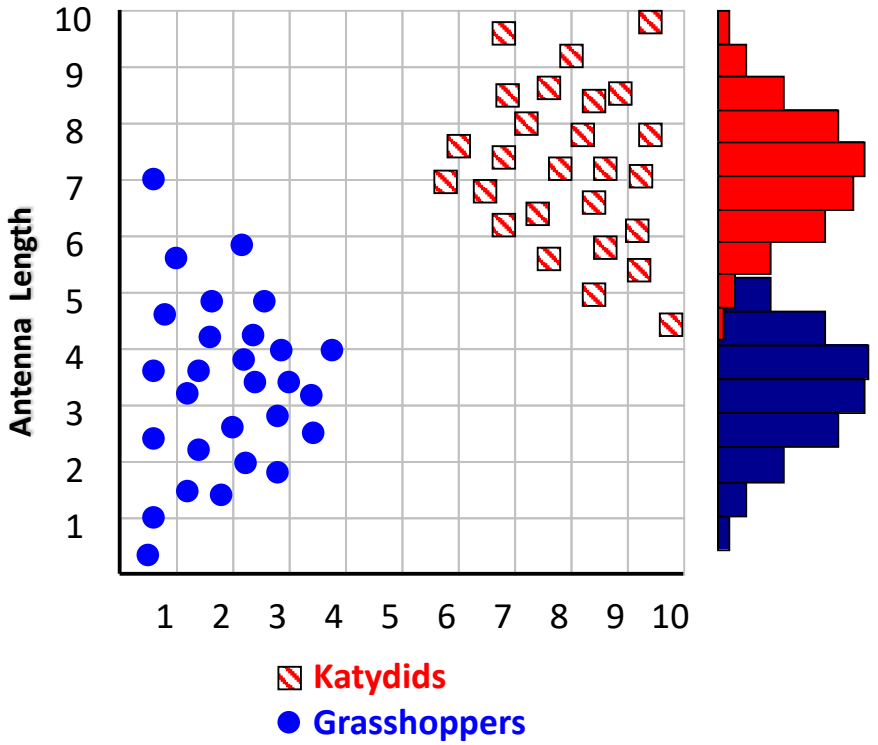
cons

- not necessarily easy to interpret
- sensitive to overfitting

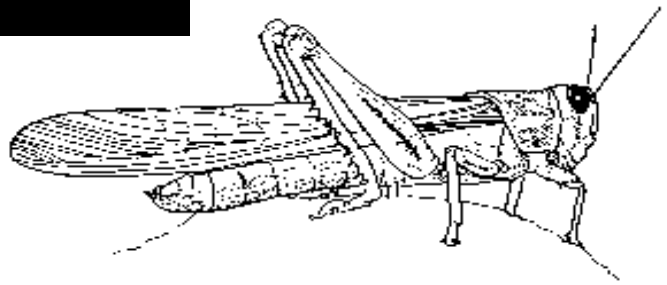
the distribution of bugs

U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

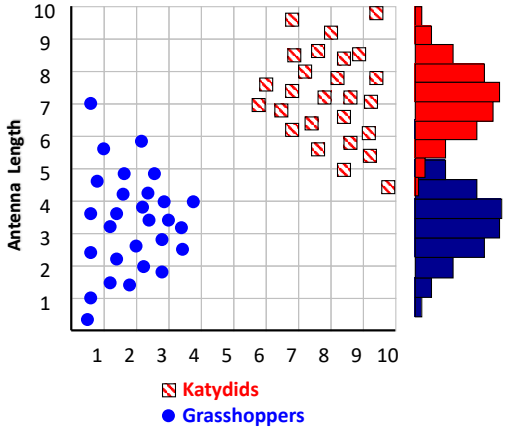
- class histogram
  - the more data available
  - … the more reliable

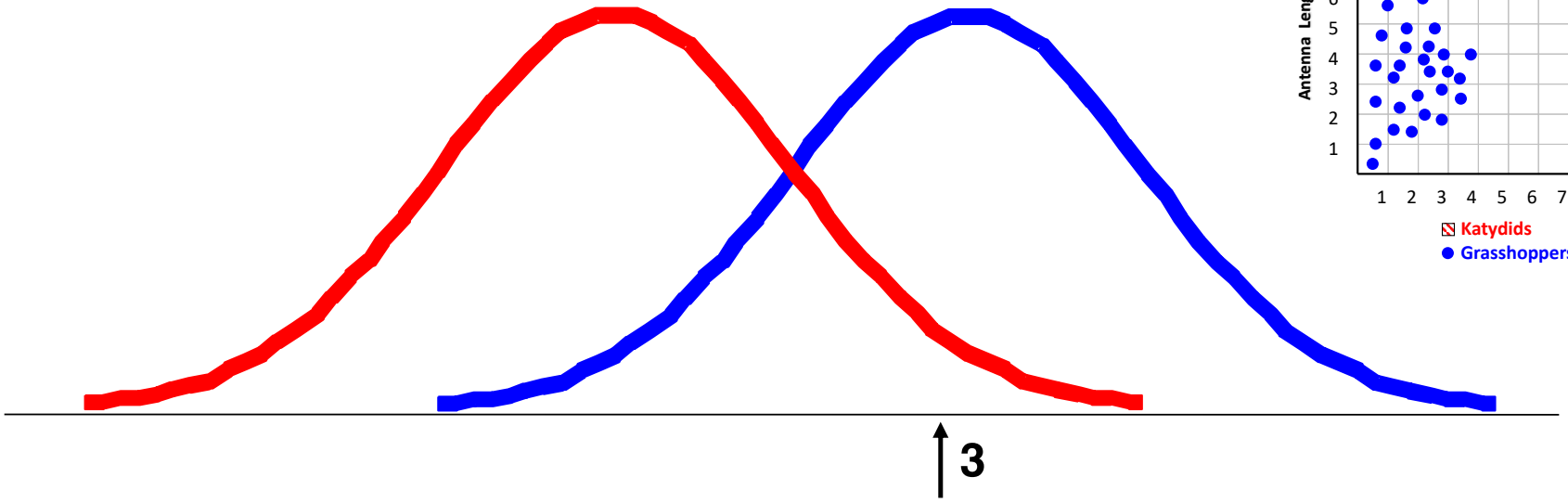


… can be used for prediction



- insect with 3 units long antennae
- … more *probably* a **Grasshopper** or a **Katydid**?
  - given the distributions of antennae lengths we have seen





$p(c_j \mid d)$ = probability of class c_j, *given* that we have observed d

… but is it reliable?

U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

how much evidence supports
$p(c_j \mid d)$?

Name	Over 170cm	Eye	Hair length	Sex
Drew	No	Blue	Short	Male
Claudia	Yes	Brown	Long	Female
Drew	No	Blue	Long	Female
Drew	No	Blue	Long	Female
Alberto	Yes	Brown	Short	Male
Karin	No	Blue	Long	Female
Nina	Yes	Brown	Short	Female
Sergio	Yes	Blue	Long	Male

Bayes classifies

U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

- Bayesian classifiers use **Bayes theorem**, which says

$$p(c_j \mid d) = \frac{p(d \mid c_j)\, p(c_j)}{p(d)}$$

  - $p(c_j \mid d)$ = probability of instance d being in class c_j,

This is what we are trying to compute

  - $p(d \mid c_j)$ = probability of generating instance d given class c_j,

We can imagine that being in class c_j, causes you to have feature d with some probability

  - $p(c_j)$ = probability of occurrence of class c_j,

This is just how frequent the class c_j, is in our database

  - $p(d)$ = probability of instance d occurring

This can actually be ignored, since it is the same for all classes

… assuming independence

U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

if attributes have independent distributions

$$p(d|c_j) = p(d_1|c_j) * p(d_2|c_j) * ....* p(d_n|c_j)$$

The probability of class c_j generating instance d, equals….

The probability of class c_j generating the observed value for feature 1, multiplied by..

The probability of class c_j generating the observed value for feature 2, multiplied by..

NB: pros and cons

U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

pros

- interpretable model
- easy to configure
- robust to irrelevant features
- fast
  - to train
    - single scan
  - to classify

cons

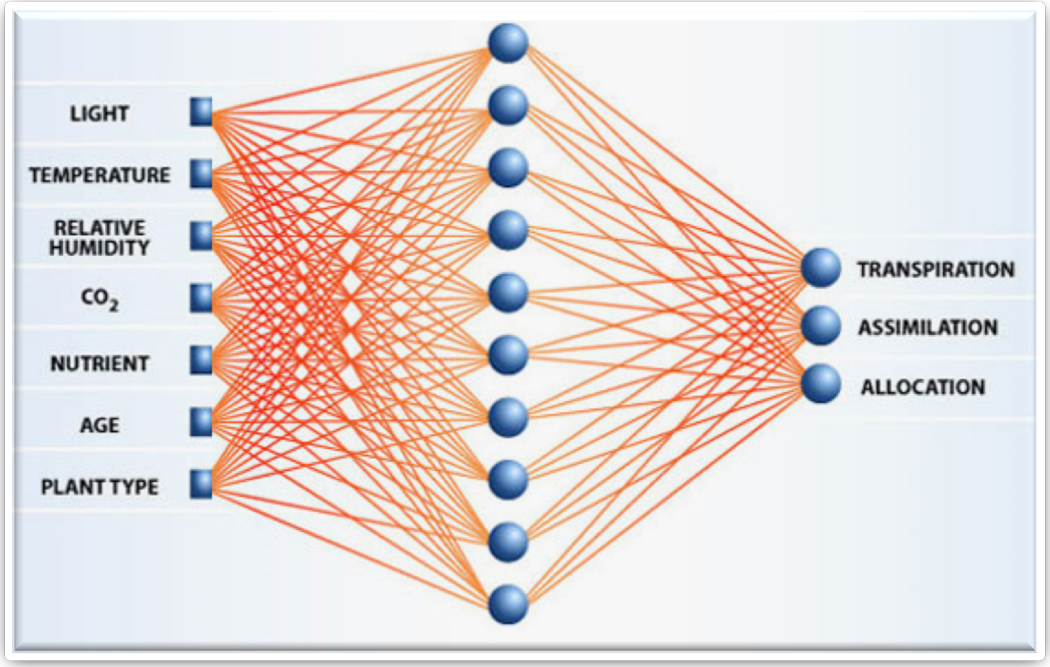
- assumes independence of features
- not directly applicable to numerical features

neural network learning

U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

- set of neurons
  - input/output units
- connected
- … with weights

- (supervised) learning
  - adjust weights to ensure outputs to given inputs are the expected ones
- predicting
  - feed input values
  - collect outputs



http://aemc.jpl.nasa.gov/activities/bio_regen.cfm

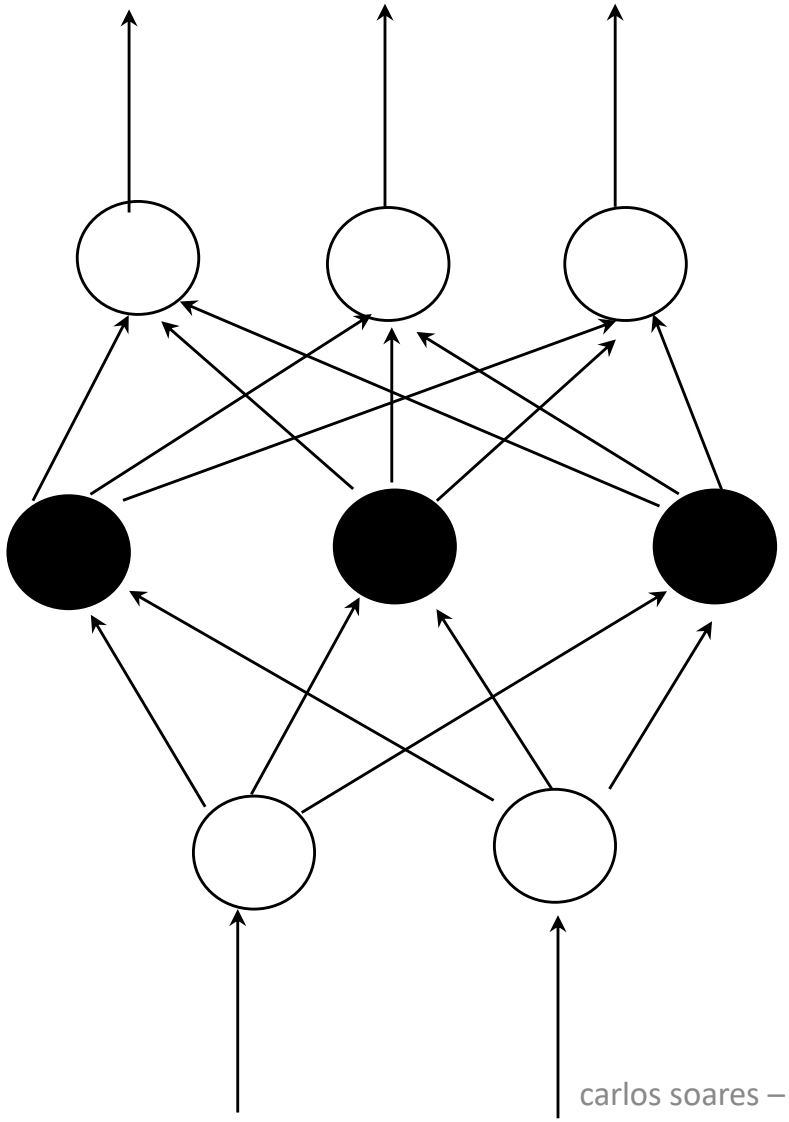
summary: feedforward

U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

output nodes

hidden nodes

input nodes



$$\hat{y}_k = 1/1+e^{-w_{0k}+\sum_{i\in\{\text{all units feeding into unit k}\}} w_{ik}y_i}$$

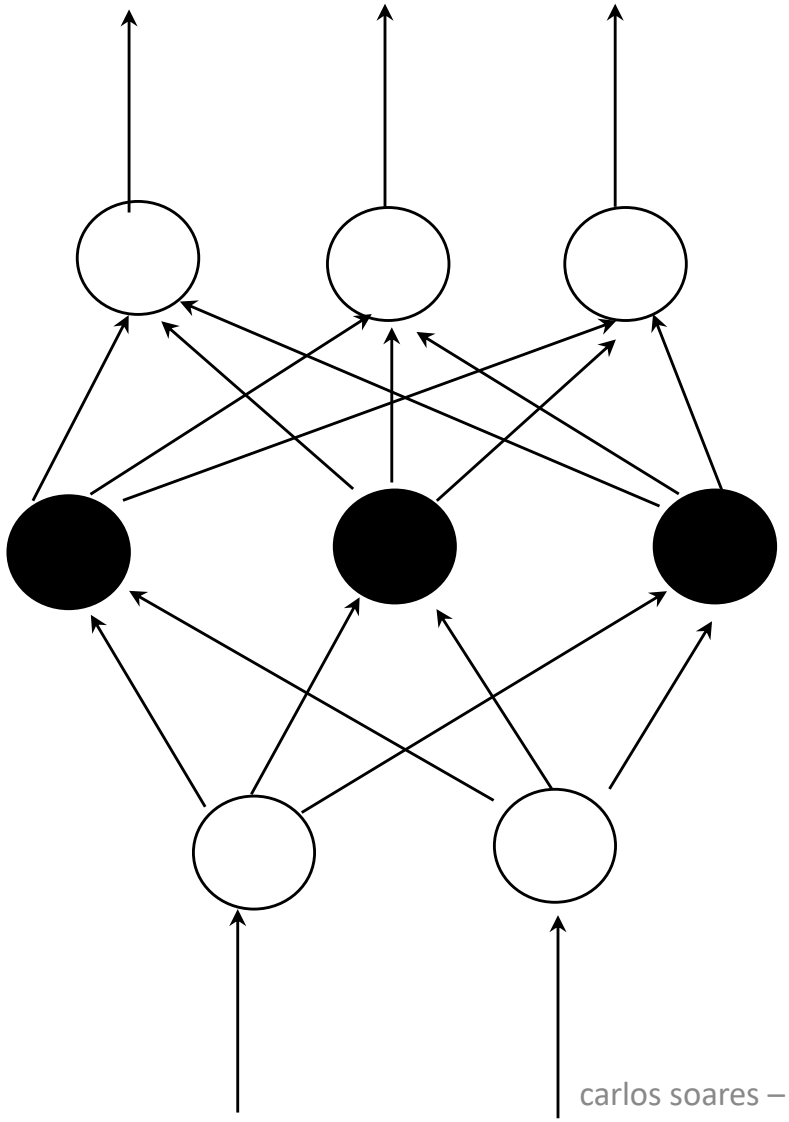
$w_{j,k}$ weights

$$y_i = 1/1+e^{-w_{0i}+\sum_{j\in\{\text{all units feeding into unit } i\}} w_{ji}x_j}$$

$w_{i,j}$ weights

x_i input values

summary: backprop



$$\delta_k = \hat{y}_k(1-\hat{y}_k)(\hat{y}_k - y_k)$$

$$w_{j,k}^{new} = w_{j,k} + r \cdot y_j \cdot \delta_k$$

$$\delta_j = y_j(1-y_j) \sum_{k \in \{\text{all units fed by this unit}\}} \delta_k$$

$$w_{i,j}^{new} = w_{i,j} + r \cdot x_i \cdot \delta_j$$

NN: pros and cons

U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

pros

- universal
  - fit any continuous function
- versatile
  - output may be one or more discrete and real values
- online
  - application and learning are intertwined
- robust
  - errors and noisy data
- fast
  - … application to new examples
- parallel

cons

- slow
  - … training
- low usability
  - empirical parameter tuning
  - network topology and learning rate
- low interpretability
  - understand the weights
- low adaptability
  - not easy to incorporate domain knowledge

universal approximator: the whole story

U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

- mlps are a class of universal approximators
  - 1 hidden layer

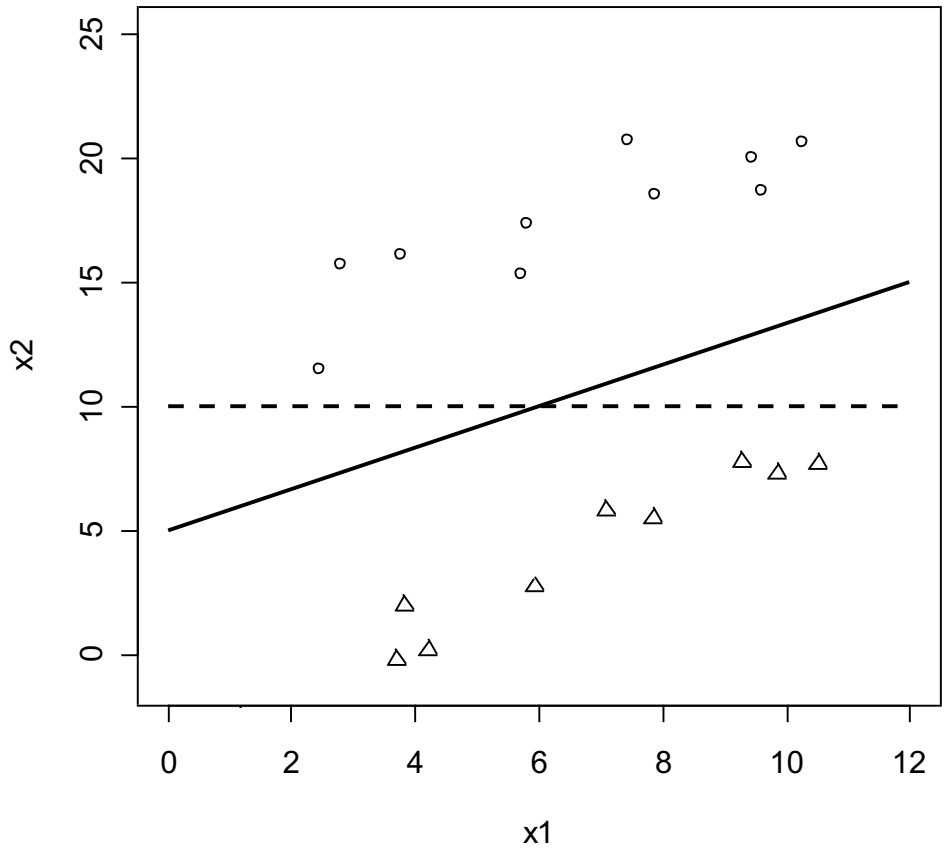
- so what's the catch?
  - provided sufficiently many hidden units…

overview

U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

- Support Vector Machines (SVM)
  - linear learning machines with maximization of margin
    - **better separation between classes**
  - duality
    - **higher robustness to the curse of dimensionality**
  - kernel trick
    - **non-linear models**
- according to Bennet & Campbell, “Support Vector Machines: Hype of Hallelujah?”, SIGKDD Explorations, 2000
  - geometric intuition
  - elegant math
  - theoretical guarantees
  - practical algorithms

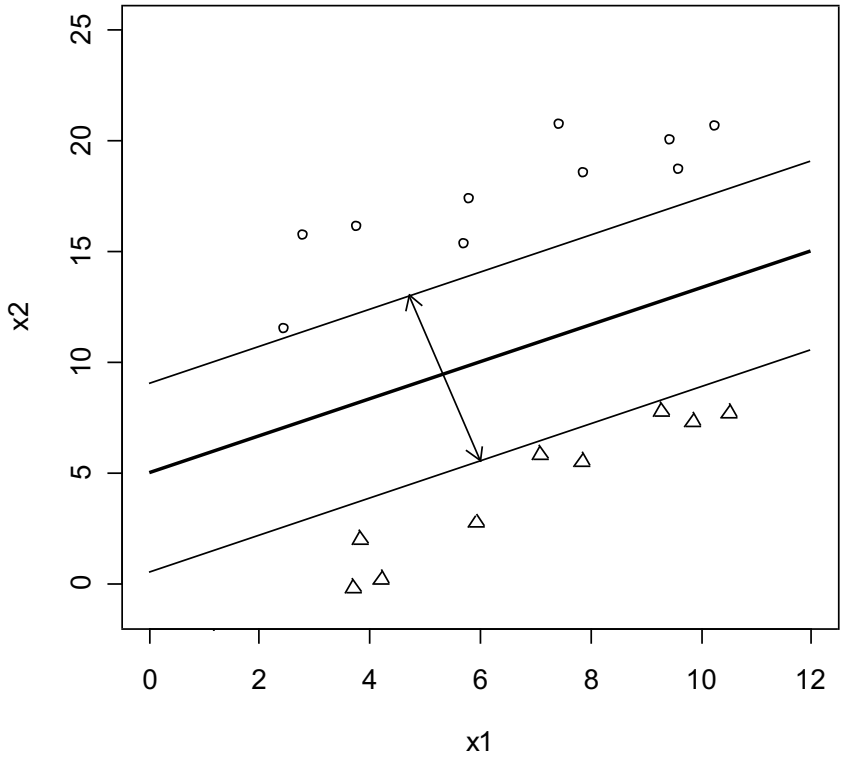
best separating hyperplane?



margin maximization: intuition

U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

- margin, γ:
  - sum of the shortest distances between points of each class and the separating hyperplane
- margin maximization
  - hyperplane "farthest away" from both classes simultaneously
- less risk of overfitting!



margin maximization: formal definition

U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

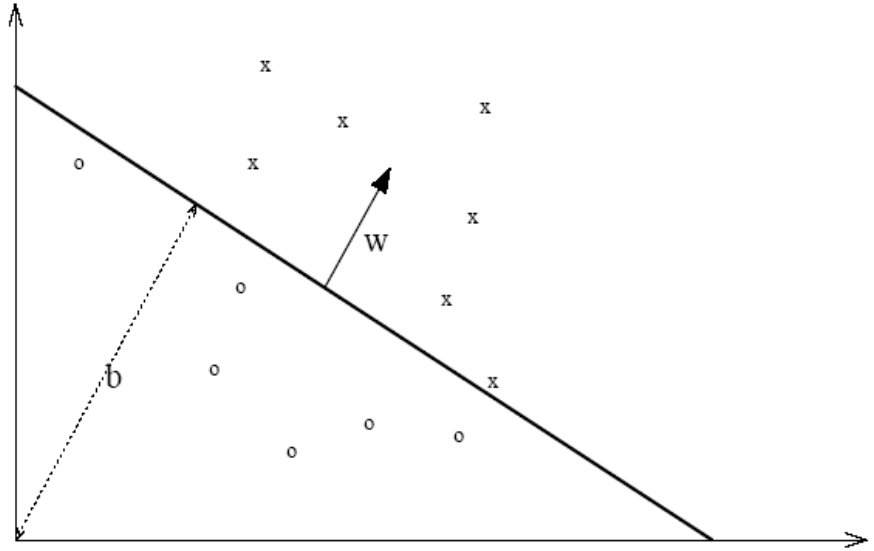
$$f(\mathbf{x}) = \langle \mathbf{w} \cdot \mathbf{x} \rangle + b$$

- it can be proved that

$$\gamma = \frac{1}{\|\mathbf{w}\|_2}$$

- … thus, maximizing the margin is the same as

$$\begin{aligned} &\text{minimize} && \|\boldsymbol{w}\|_2 \\ &\text{subject to} && y_i(\langle \boldsymbol{w} \cdot \boldsymbol{x}_i \rangle + b) \geq 1 \\ &\text{the constraints} && i = 1, \ldots, l \end{aligned}$$



- quadratic programming problem
  - i.e. a lot of work in the area of optimization in this problem can be reused, most importantly…

fonte: www.kernel-machines.org

… duality

U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

- dual problem

$$\text{maximize} \quad \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l} y_i y_j \alpha_i \alpha_j \langle \boldsymbol{x}_i \cdot \boldsymbol{x}_j \rangle$$

$$\text{subject to the constraints} \quad \sum_{i=1}^{l} y_i \alpha_i = 0$$

$$\alpha_i \geq 0, i = 1, \ldots, l$$

- linear prediction problem

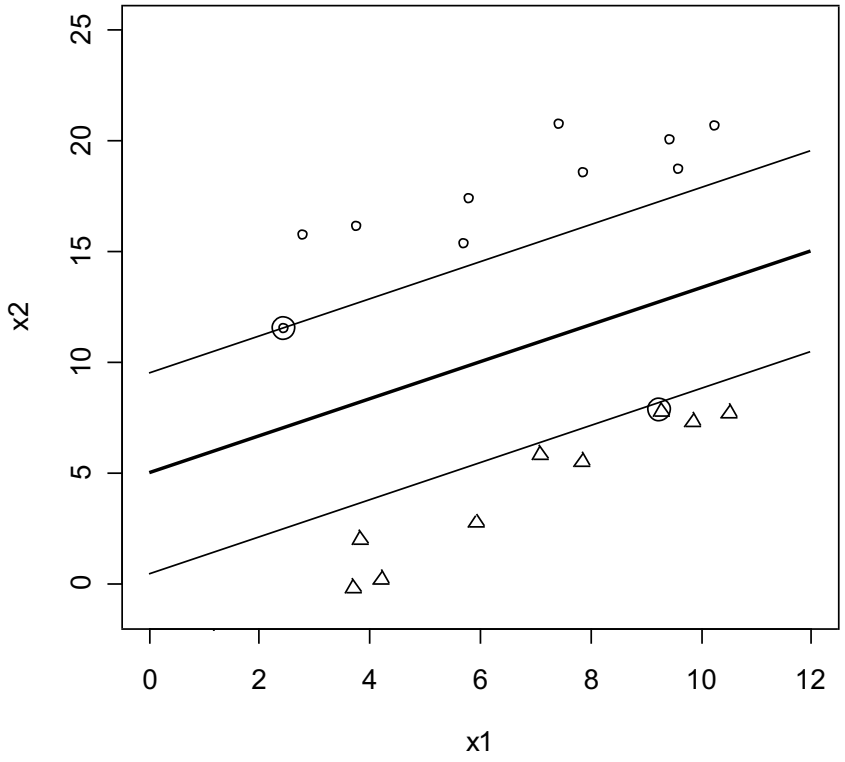
$$f(\boldsymbol{x}_{new}) = \sum_{i=1}^{l} \alpha_i y_i \langle x_i \cdot \boldsymbol{x}_{new} \rangle + b$$

model

U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

- subset of examples that determine the margin
  - frontier
  - other points are irrelevant

$$f(\boldsymbol{x}_{new}) = \sum_{i=1}^{l} \alpha_i y_i \langle x_i \cdot \boldsymbol{x}_{new} \rangle + b$$



pros

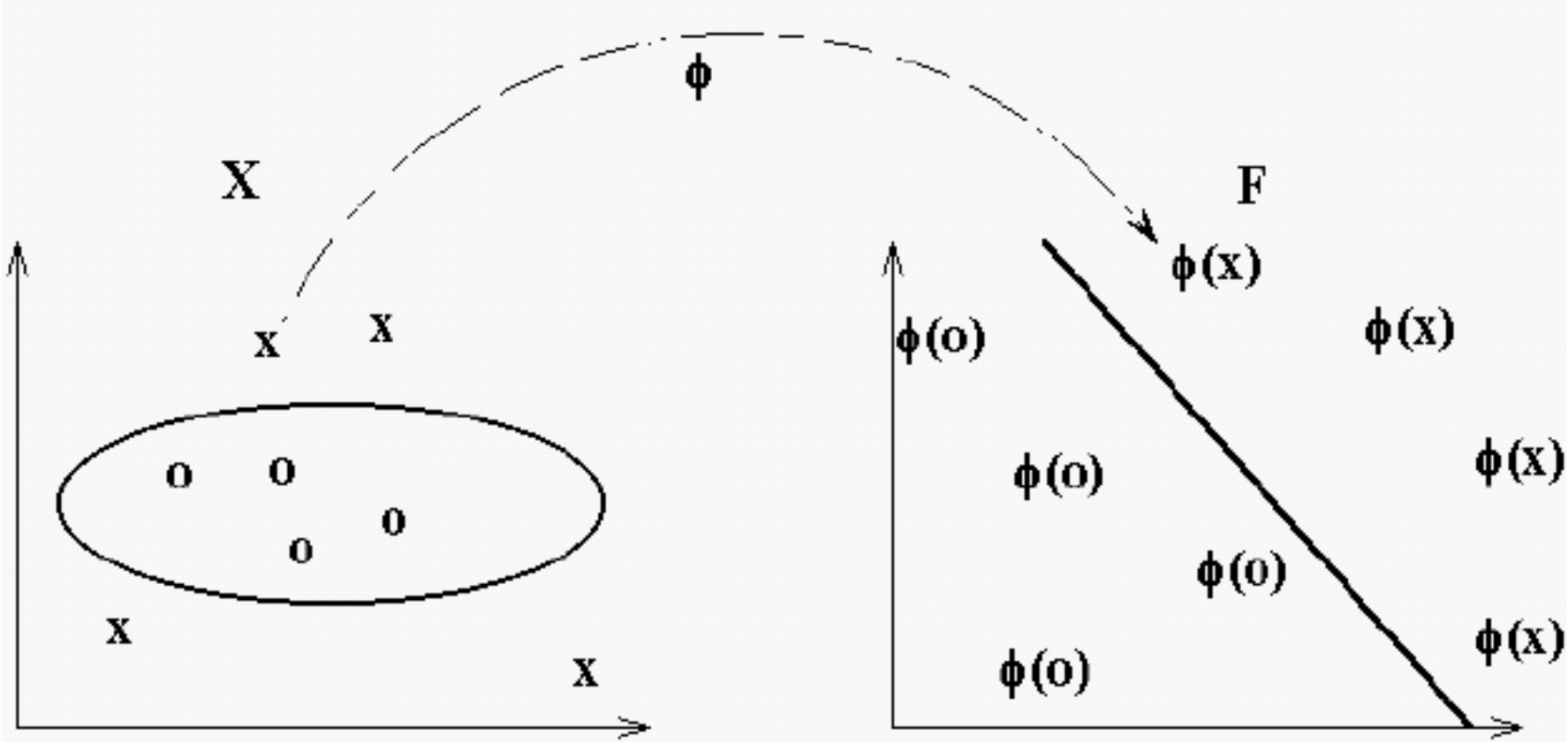
U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

$$\sum_{i=1}^{l}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{l} y_i y_j \alpha_i \alpha_j \left\langle \mathbf{x}_i \cdot \mathbf{x}_j \right\rangle$$

- QP methods are very mature
- statistical learning theory
  - bounds to the generalization error based on the training error
- results independent of initial conditions
  - order of presentation of examples
  - initializations
- convex problem
  - no local minima
  - … reducing the probability of overfitting
- dual is independent of number of attributes
  - minimizing effect of the curse of dimensionality

non-linear functions

- map attributes to space where linear discrimination is possible



fonte: www.kernel-machines.org

but, which space?

U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

- "A complex pattern-classification problem cast in a high-dimensional space nonlinearly is more likely to be linearly separable than in low-dimensional space"
  - Cover (95)
- i.e., the bigger, the better… or maybe not
  - theoretical challenge: curse of dimensionality
  - practical challenge: computational resources
    - e.g. memory

dual problem in space…

U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

- dual problem

$$\text{maximize} \quad \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l} y_i y_j \alpha_i \alpha_j \langle \phi(x_i) \cdot \phi(x_j) \rangle$$

$$\text{subject to the constraints} \quad \sum_{i=1}^{l} y_i \alpha_i = 0$$

$$\alpha_i \geq 0, i = 1, \ldots, l$$

- (non-)linear prediction function

$$f(\boldsymbol{x}_{new}) = \sum_{i=1}^{l} \alpha_i y_i \langle \phi(x_i) \cdot \phi(\boldsymbol{x}_{new}) \rangle + b$$

the kernel trick

U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

- kernel $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \langle \phi(x_i) \cdot \phi(x_j) \rangle$
- … integrated in the previous method

$$\text{maximize} \quad \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l} y_i y_j \alpha_i \alpha_j K(\boldsymbol{x}_i, \boldsymbol{x}_j)$$

$$\text{subject to the constraints} \quad \sum_{i=1}^{l} y_i \alpha_i = 0$$

$$\alpha_i \geq 0, i = 1, \ldots, l$$

$$f(\boldsymbol{x}_{new}) = \sum_{i=1}^{l} \alpha_i y_i K(\boldsymbol{x}_i, \boldsymbol{x}_{new}) + b$$

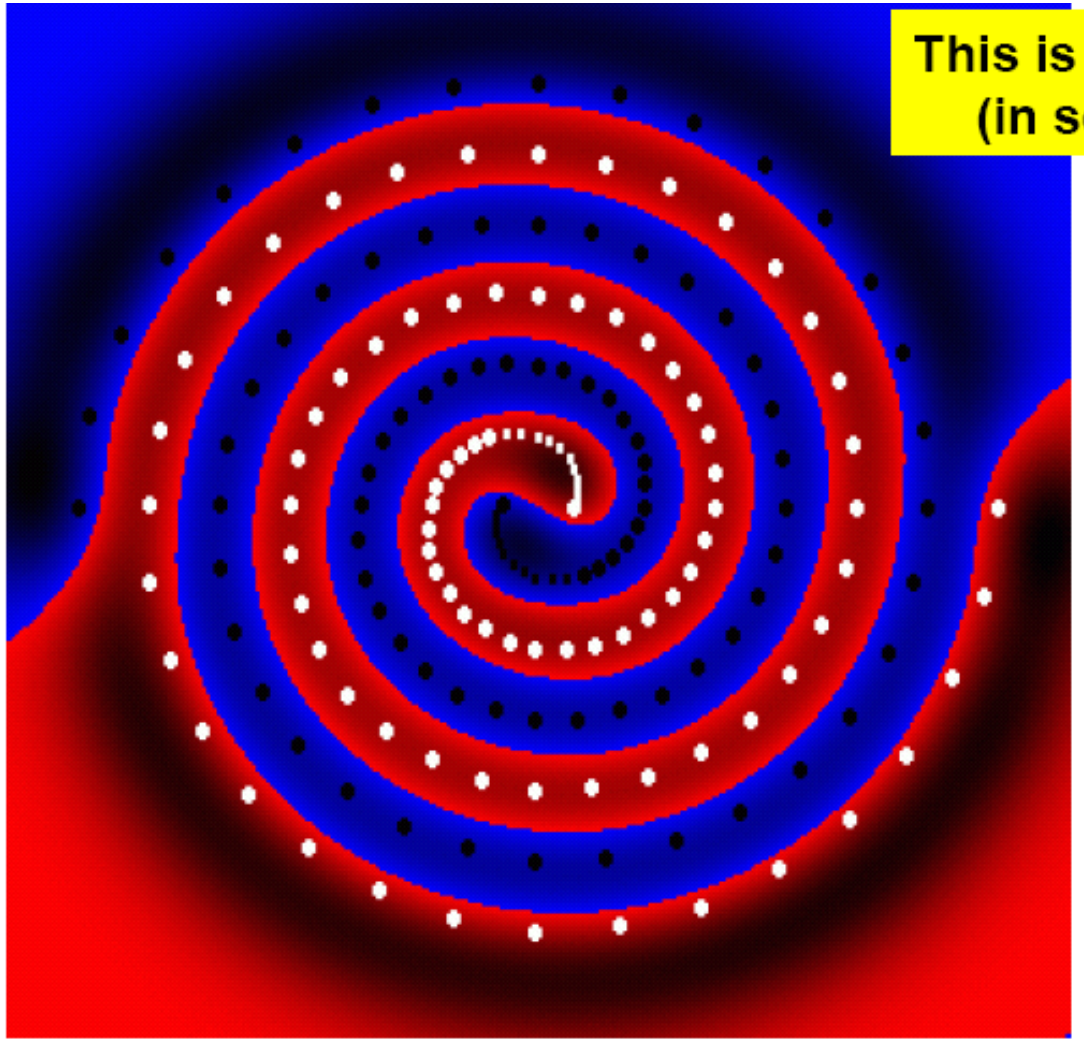
kernels + maximization of the margin with the dual

U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

$$K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \langle \phi(x_i) \cdot \phi(x_j) \rangle$$

- pros: kernel
  - problem projected to higher dimension space…
    - not necessarily...
    - but potentially infinite...
  - implicit
    - no need to compute ϕ(x)...
    - … which may be unknown!
- pros: margin maximization
  - dual doesn't depend on the number of variables
    - minimize the effect of the curse of dimensionality
  - problem remains convex
    - unique solution

gaussian kernel



$$e^{\frac{-\left\|\mathbf{x}_i-\mathbf{x}_j\right\|^2}{2\sigma^2}}$$

fonte: www.kernel-machines.org

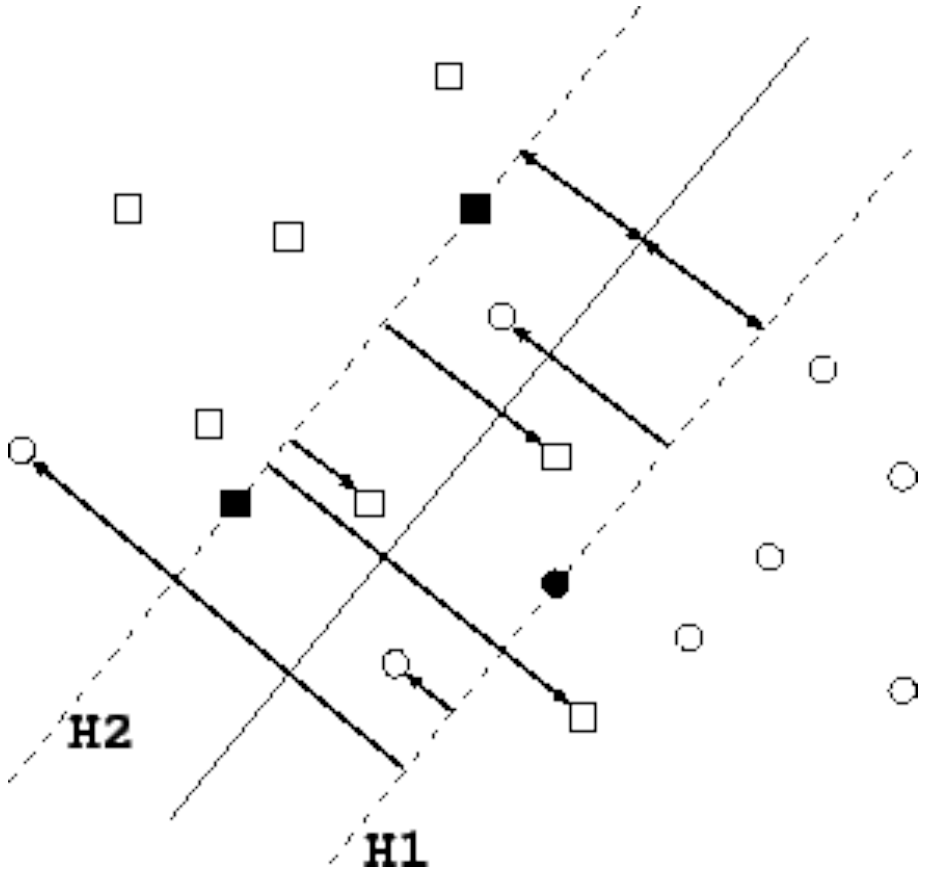
relax the objetive function

U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

- soft margin
  - maximize the margin
  - minimize error

$$\frac{1}{2}\|\mathbf{w}\|_2 + C\sum_{i=1}^{l} z_i$$

- C, regularization constant
  - compromise between the importance of the margin and the error
  - yet another parameter…
- most common type of SVM



SVM Soft Margin: dual problem

U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

$$\text{maximize} \quad \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l} y_i y_j \alpha_i \alpha_j K(\boldsymbol{x}_i, \boldsymbol{x}_j)$$

$$\text{subject to} \quad \sum_{i=1}^{l} y_i \alpha_i = 0$$

$$\text{the constraints} \quad C \geq \alpha_i \geq 0, i = 1, \ldots, l$$

- only one difference
  - weight of each example is limited

gps

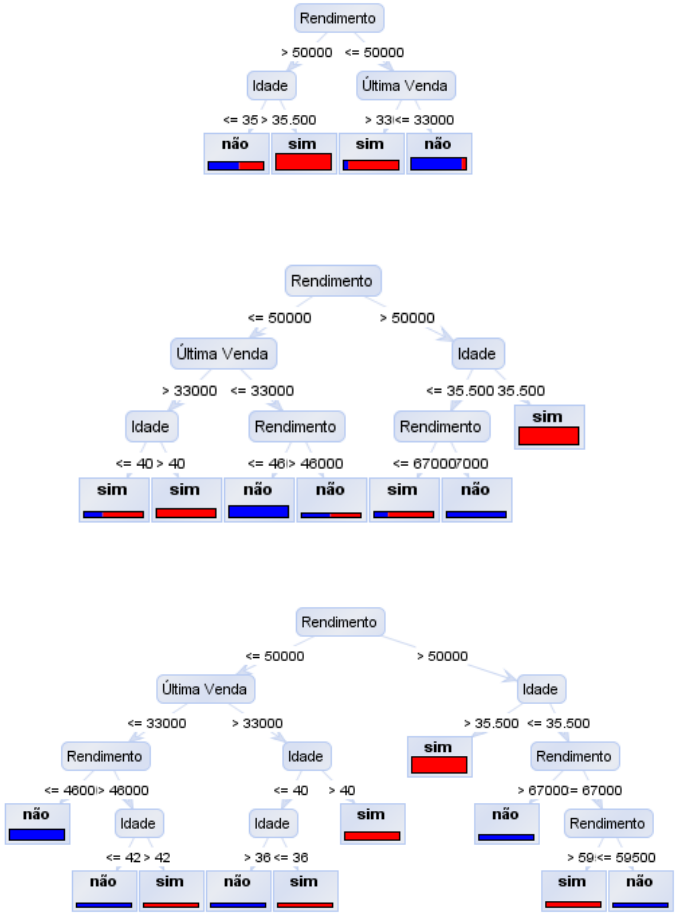
U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

- understanding ML algorithms
  - algorithm vs geometric intuition
  - we've got neural nets, so why waste time?
- some popular algorithms
  - Linear Classifiers
  - Nearest Neighbors
  - Decision Trees
  - Naive Bayes
  - Neural Networks
  - Support Vector Machines
- ... with hyperparameters that may have to be tuned

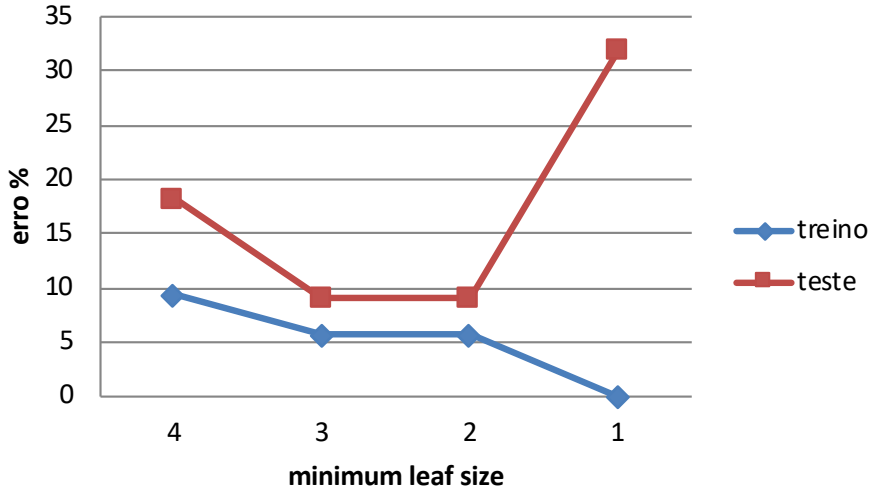
remember: overfitting in DTs

U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

- hyperparameter: minimum leaf size



- … affects predictive performance of DT model



hyperparameter tuning

U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

grid search

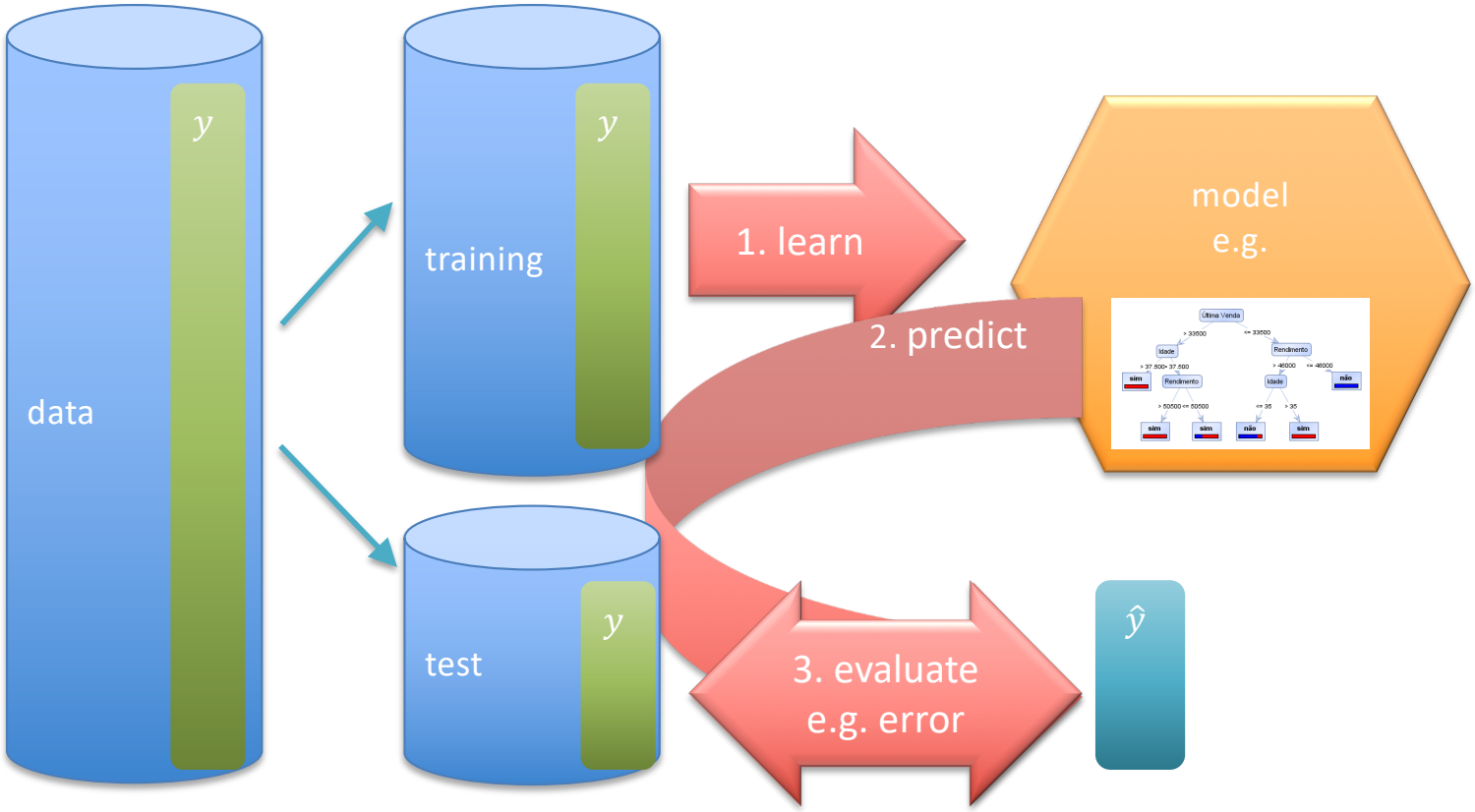
- define grid
  - e.g. minimum leaf size: {5, 25, 50, 100}
  - … maximum depth: {3, 5, 10}
- learn and evaluate models for all combinations
- choose best

random search

- define domain
  - e.g. minimum leaf size: {5,… 100}
  - … maximum depth: {3, …, 10}
- generate combinations randomly
- learn and evaluate models for all combinations
- choose best

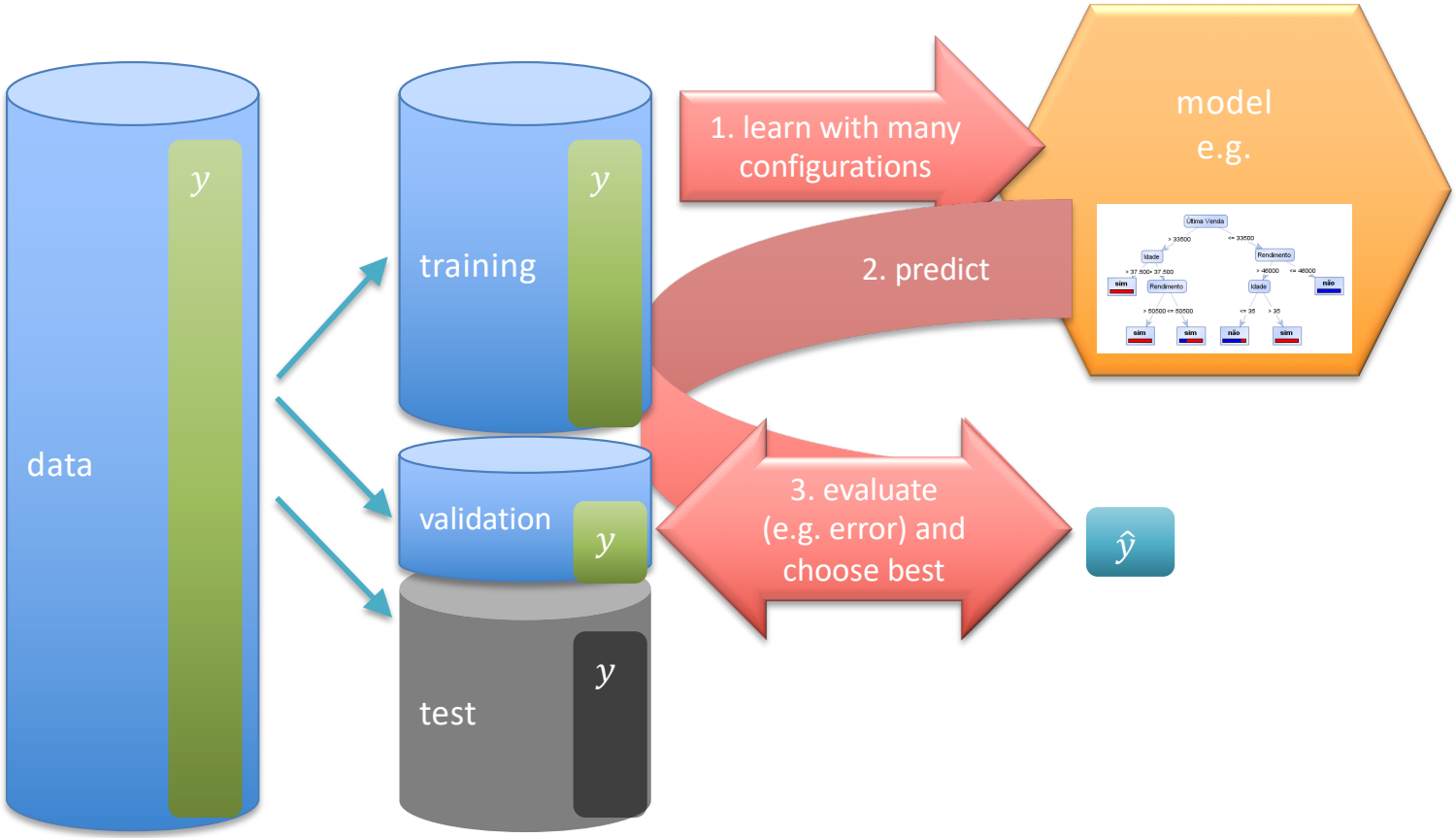
evaluation methodology: is this still suitable?

U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO



evaluation methodology with hyperparameter tuning (1/2)

U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO



evaluation methodology with hyperparameter tuning (2/2)