# Machine Learning
# Exercise Session - Bagging, Boosting and RANSAC

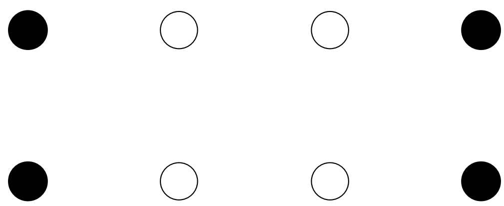
Professor: Aude Billard
Lecturer: Dr. Basilio Noris
contacts:
aude.billard@epfl.ch
basilio.noris@epfl.ch

Spring Semester 2017

## 1 Boosting with Decision Stump

Consider the following 2-class binary problem

$$\begin{aligned} C_{y=-1} &= \{(-3,-1),(-3,1),(3,-1),(3,1)\} \\ C_{y=1} &= \{(-1,-1),(-1,1),(1,-1),(1,1)\} \end{aligned}$$



Using the error and weight update rule of the discrete AdaBoost, answer the following question

a) What are the first 2 decision stumps, and what are their corresponding thresholds.

b) Are these sufficient to obtain perfect classification?

### Solution

a) First decision stump: $\varphi_1(x) = \begin{cases} 1, & \text{if } x > -2 \\ -1, & \text{otherwise} \end{cases}$

Second decision stump: $\varphi_2(x) = \begin{cases} 1, & \text{if } x < 2 \\ -1, & \text{otherwise} \end{cases}$

b) Not with Discrete AdaBoost. The final vote is obtained by linear combination of the weak learners. Regardless of the choice of weights assigned to each learner, the vote for the negative (black) samples on one of the two sides will be positive (the classifier function will be equal to 0 if all the weak learner weights are equal for instance). You need at minimum 3 weak learners to correctly classify this problem so that the third weight can play the role of a bias (a decision stump with all the points on one side of it, with a weight equal to the bias).

## 2 Weak Classifiers

Selecting the proper model of weak classifier is often a very important step when tackling new classification problems. Presented below are a number of such problems; for each dataset:

a) Propose a weak learner that will excel in classifying the data

b) Estimate the number of weak learners necessary to perform the task optimally

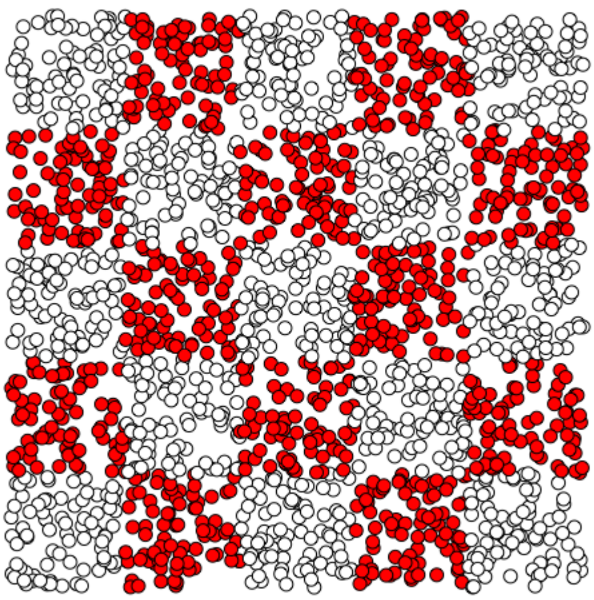


Figure 1: Checkerboard

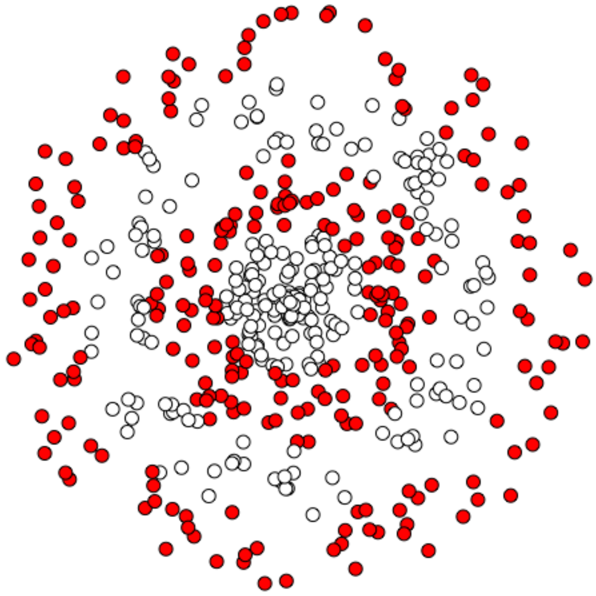


Figure 2: Concentric rings

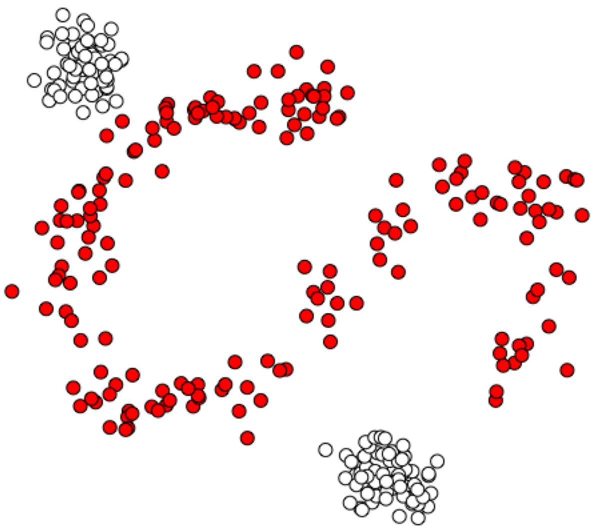


Figure 3: The serpent

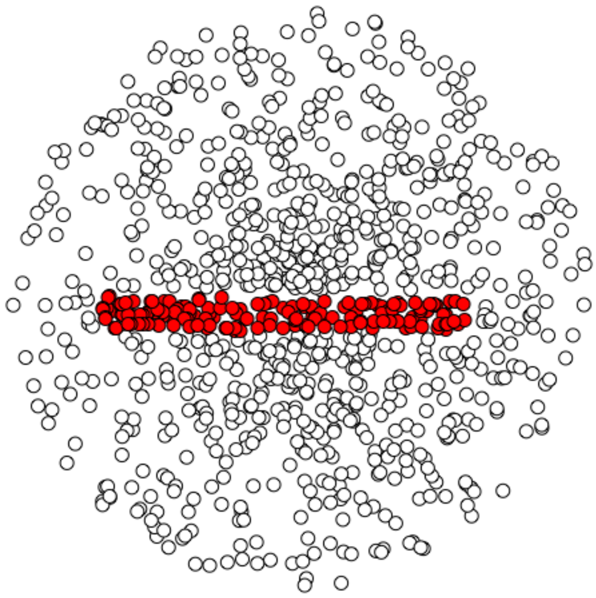


Figure 4: Stop sign

## Solution

Figure 1:

a) The easiest solution would be using random rectangles or decision stumps, as they will perfectly model the straight boundaries between the two classes. However, if the number of weak learners generated is too small, no rectangles might fit perfectly the boundaries of each 'cell'. In this case a "Random Circles" might better do the job as it is more robust to the initial random generation

b) Assuming that a sufficient number of weak learners have been generated, the system could perfectly classify the example using 12 random rectangles (there are 12 red squares and 13 white square, if you put equal weight one for each weak classifier on the red squares, there is no need for a bias w.l.) or 9 decision stumps (4 for each axis and one bias.)

Figure 2:

a) Random Circles are perfect for this task as they perfectly model the radial expansion of the classes

b) Three boundaries need to be modeled and 3 weak learners are sufficient to properly classify this data (on the contrary to Exercise 1, there is already an odd number of weak learner therefore no other one is required to add a bias.)

Figure 3:

a) Random Projection would allow to 'cut the edges' of the negative (white) class while maintaining perfect classification of the positive class. Random rectangles are another possible choice.

b) A minimum of 3 weak learners would be necessary to solve correctly this problem (see Exercise 1). One random rectangle is also enough.

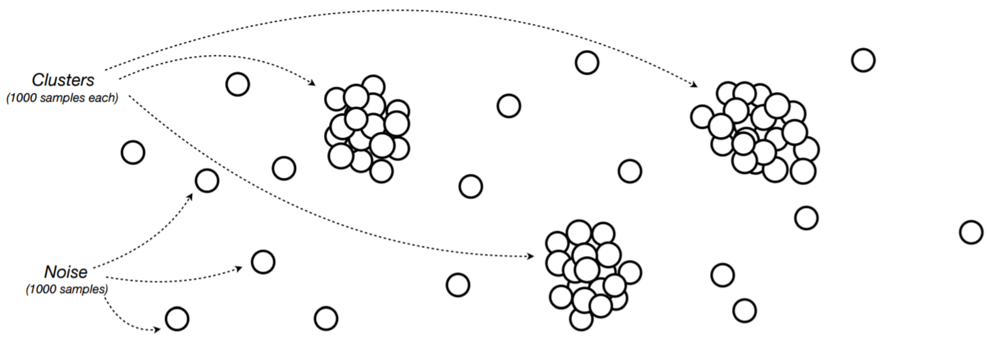
Figure 4:

a) Random rectangles would be a good starting point, if sufficient examples are generated. Otherwise, Decision stumps are well fitted to this problem.

b) A single random rectangle, if placed perfectly would do the job. Alternatively, 6 decision stumps would also suffice (3 to classify horizontally, 3 to classify vertically, see Exercise 1).

# 3 RANSAC

Consider the following distribution of points, where 3 clusters of 1000 points reside in a space with 1000 points that are randomly located (following a uniform distribution) :




a) Define a model and an objective function (consensus function) that are computationally efficient and should yield a good clustering

b) What is the probability that it will find the appropriate solution after 10 iterations?

c) How many iterations will ensure that the probability of finding an appropriate solution is greater than 99%?

d) Is that computationally efficient?

## Solution

a) We may pick 3 points at random and, by hardcoding a meaningful value of radius, calculate the distance of all the other points to the randomly picked points. To keep it simple, we may calculate the Manhattan distance.

b) The probability for a point being in a cluster is $1/4$, thus having 1 point in each of the 3 cluster has a probability of $1/4 \times 1/4 \times 1/4 = 1/64$. To calculate the probability that this happens at least once in 10 iterations,(which is what we need to define clusters) we need the probability of the points *not* being each in a cluster at each iteration, which is $1 - 1/64 = 63/64$, and we elevate it to the number of iterations. By subtracting it to 1 we have then the probability of having 1 point in each of the 3 clusters after 10 iterations, which is:

$$1 - (63/64)^{10} = 0.15$$

c) To calculate at which iteration we have 0.99 probability of finding 1 point in each of the 3 clusters, we use the same formula as is solution b) by changing the probability and substituting $x$ for the number of iterations:

$$1 - (63/64)^{x} = 0.99$$

Thus we can calculate x by taking the logarithm,

$$x \log(63/64) = \log(0.01)$$

which gives:

$$x = 292.4$$

Thus we need at least 293 iterations to ensure a probability higher than 99%.

d) Yes, it is: even though we need almost 300 iteration to get a 99% probability of finding an appropriate solution, the model is very simple and thus would require few calculations per iteration. The complexity is $\mathcal{O}(N)$, where $N$ is the number of points. This is very efficient compared to other methods.