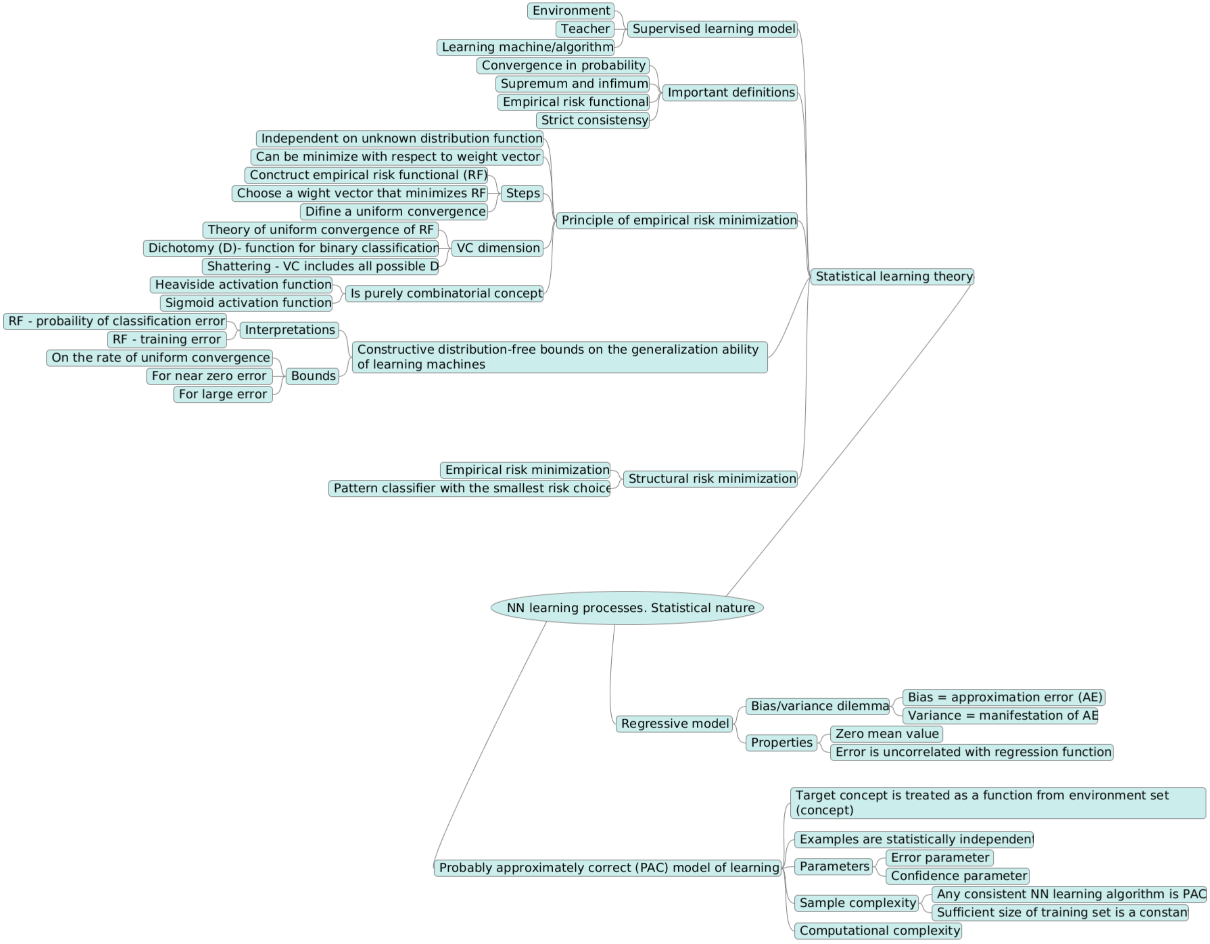
Neural Networks
- Homework 4 -

Petr Lukin, Ivan Vishniakou, Evgeniya Ovchinnikova

Lecture date: 24 October 2016

1 Mind map

Figure 1: Mind map. Chapter 2 (second part) from Haykins book.



NN learning processes. Statistical nature
Statistical learning theory
Supervised learning model
Environment
Teacher
Learning machine/algorithm
Important definitions
Convergence in probability
Supremum and infimum
Empirical risk functional
Strict consistensy
Principle of empirical risk minimization
Independent on unknown distribution function
Can be minimize with respect to weight vector
Steps
Conctruct empirical risk functional (RF)
Choose a wight vector that minimizes RF
Difine a uniform convergence
VC dimension
Theory of uniform convergence of RF
Dichotomy (D)- function for binary classification
Shattering - VC includes all possible D
Is purely combinatorial concept
Heaviside activation function
Sigmoid activation function
Constructive distribution-free bounds on the generalization ability of learning machines
Interpretations
RF - probaility of classification error
RF - training error
Bounds
On the rate of uniform convergence
For near zero error
For large error
Structural risk minimization
Empirical risk minimization
Pattern classifier with the smallest risk choice
Regressive model
Bias/variance dilemma
Bias = approximation error (AE)
Variance = manifestation of AE
Properties
Zero mean value
Error is uncorrelated with regression function
Probably approximately correct (PAC) model of learning
Target concept is treated as a function from environment set (concept)
Examples are statistically independent
Parameters
Error parameter
Confidence parameter
Sample complexity
Any consistent NN learning algorithm is PAC
Sufficient size of training set is a constan
Computational complexity

2 Exercises

Exercise 2.1

Consider the space of instances X corresponding to all points in the x, y plane. Give the VC dimension of the following hypothesis spaces:

Solution:

a. H_r = the set of all rectangles in the x,y plane. i.e. $H_r = \{((a < x < b) \wedge (c < y < d))|a, b, c, d \in IR\}$

Since $H_r = \{((a < x < b) \wedge (c < y < d))|a, b, c, d \in IR\}$ we have non-rotatable rectangles only with horizontal and vertical edges those are not tight to the center of coordinate plane.
The shattering is possible in case if we can select a certain number of points so it will be possible to choose this number of points minus one without taking into account the last one. In Fig. 2 we show how we can shatter 4 points in all possible combinations. However, we can not choose the five points and select four of them without the fifth one. Fig. 3 shows that it is not possible to shatter 5 points. One can see that to define the fifth point we already have four defined points on a rectangle. Fifth point can be either inside or on the edge. Therefore, VC dimension is 4.

Figure 2: Four points shattered by a rectungular.

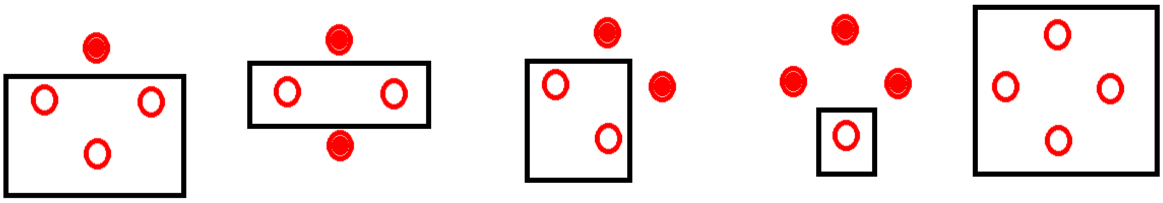
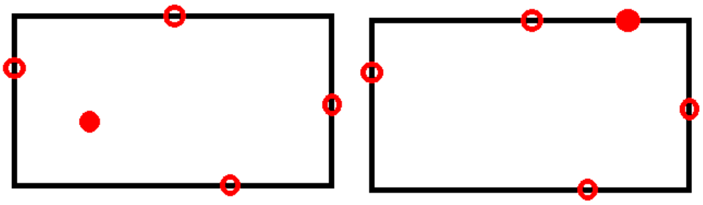


Figure 3: Five points not shattered by a rectungular.



b. H_c = the set of all circles in the x,y plane. Points inside the circle are classified as positive examples.

A circle can be defined by 3 points, so the VC is 3. In Fig. 4 we show 3 points shattering and in Fig. 5 that 4 points cannot be shattered.

Figure 4: Four points shattered by a rectungular.

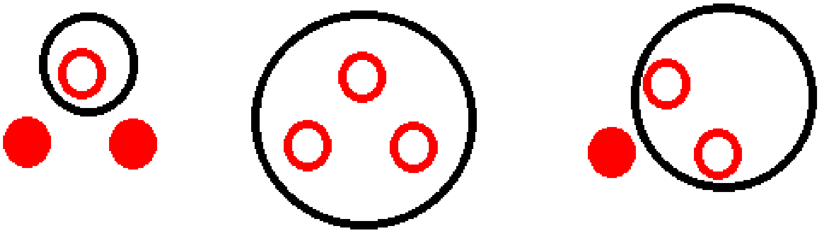
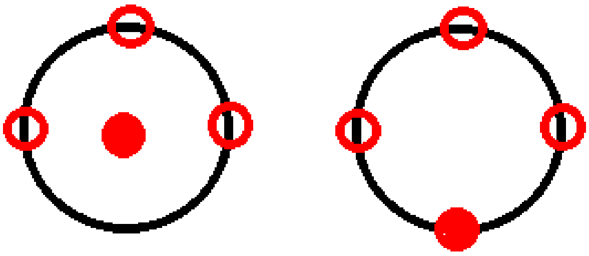


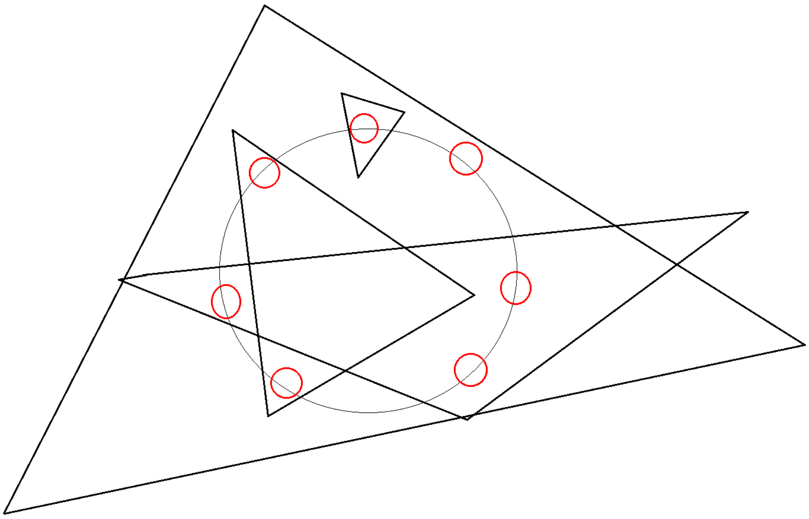
Figure 5: Five points not shattered by a rectungular.



c. H_t = the set of all triangle in the x,y plane. Points inside the triangle are classified as positive examples.

In this case we can always cut 3 points or less of the certain type, because the triangle has three vertices that can be easily seen when we put the points on a circle (see Fig. 8). So, it's not a problem to divide 3 from 4 as well as the less number of point and the all cases will be taken into account. In case of 8 points we can only divide 3 points, so the case 4-4 is missing. That is why VC is 7.

Figure 6: Four points shattered by a rectungular.



Exercise 2.2

Definition: consistent learner

- A learner is consistent if it outputs hypotheses that perfectly fit the training data, whenever possible. It is quite reasonable to ask that a learning algorithm be consistent, given that we typically prefer a hypothesis that fits the training data over one that does not.

  Task:

- Write a consistent learner for H_r from last Exercise (i.e. $H_r = \{((a < x < b) \wedge (c < y < d)) | a, b, c, d \in IR\}$). Generate a variety of target concept rectangles at random, corresponding to different rectangles in the plane. Generate random examples of each of these target concepts, based on a uniform distribution of instances within the rectangle from (0,0) to (100, 100).

Plot the generalization error as a function of the number of training examples, m. On the same graph, plot the theoretical relationship between e and m, for d = .95. Does theory fit experiment?

Solution:

Generalization error ($\hat{\epsilon}\epsilon(h)$) is given as the following:

for each sample $Z_i = 1\{h_i(x) \neq c(x)\}$ (Bernoully random variable):

$$\hat{\epsilon} = \frac{1}{m}\sum_{j=1}^{m} Z_i$$

Probability that the version space with respect to H_r and S (a sequence of training examples $m \geqslant 1$) is not ϵ-exhausted (with respect to c) is less than $|H|e^{-\epsilon m}$, where ϵ is an error. $0 < \epsilon < 0.5$, probability $0 < \delta < 0.5$.

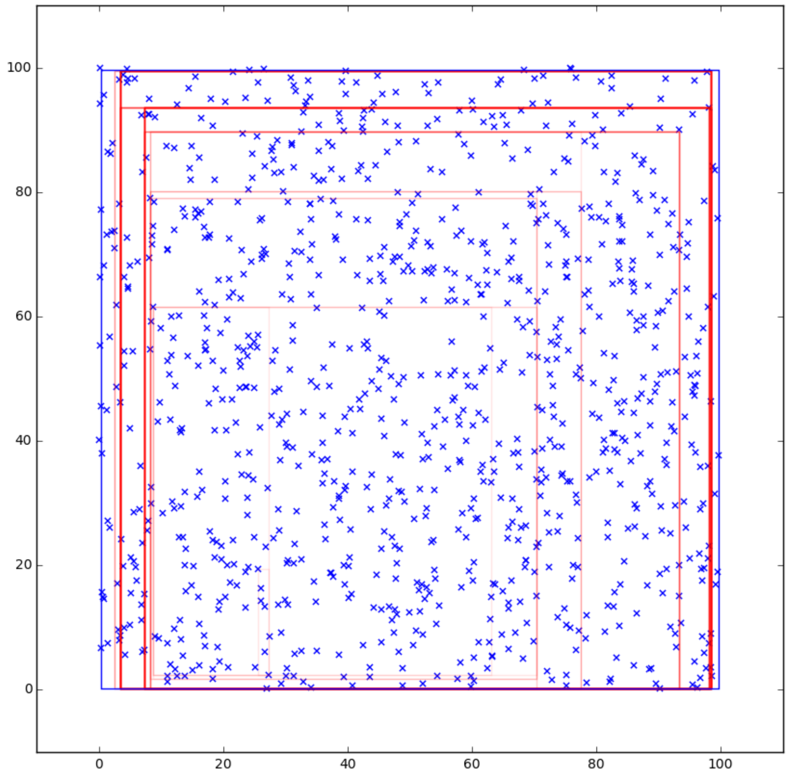
$\epsilon \leqslant \frac{1}{m}(lnH + ln\frac{1}{\delta})$, so:
$\epsilon \leqslant \frac{1}{m}(ln4 + ln\frac{1}{(1-0.95)})$

The rectangle learner is implemented simply by taking the x and y bounds of the given sample set. This always returns the most specific hypothesis:

```python
def learn_rect(samples):
    a = np.min([s[0] for s in samples])
    b = np.max([s[0] for s in samples])
    c = np.min([s[1] for s in samples])
    d = np.max([s[1] for s in samples])
    return a,b,c,d
```

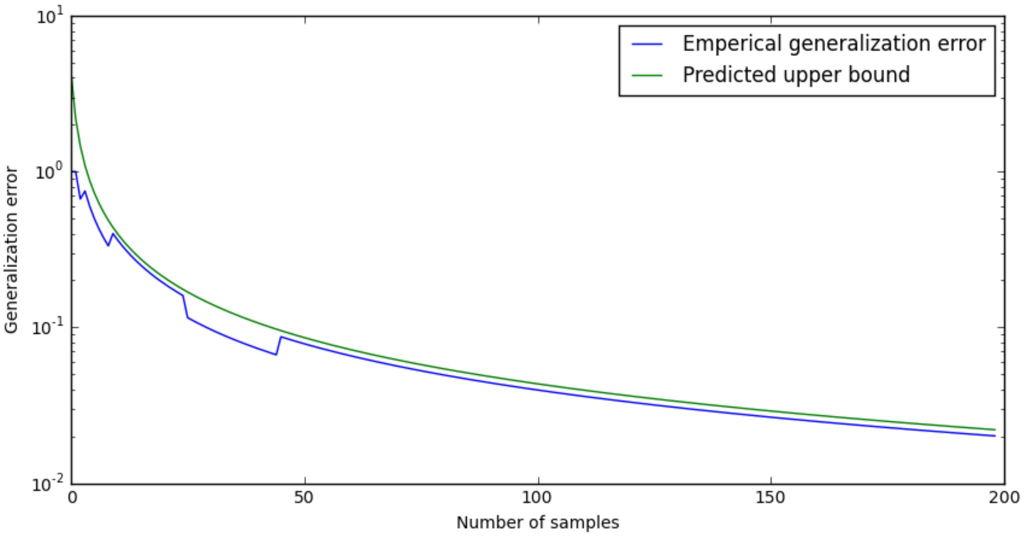
The concept is a rectangle $(0, 0, 100, 100)$ and the samples are drawn form it. The learnt hypothesis can be plotted as improving and covering more and more of the concept space as the number of introduced samples increases:

Figure 7: Improving hypothesis rectangles covering the concept space.



The resulting error can be plotted as well as the predicted shows their good correspondance:

Figure 8: Improving hypothesis rectangles covering the concept space.


Emperical generalization error
Predicted upper bound
Generalization error
10^1
10^0
10^{-1}
10^{-2}
Number of samples