

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.



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. $H_r = \{((a < x < b) \wedge (c < y < d)) | a, b, c, d \in \mathbb{R}\}$

Since $H_r = \{((a < x < b) \wedge (c < y < d)) | a, b, c, d \in \mathbb{R}\}$ 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 rectangular.

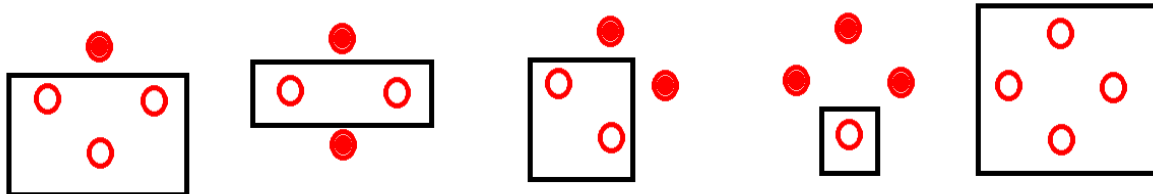
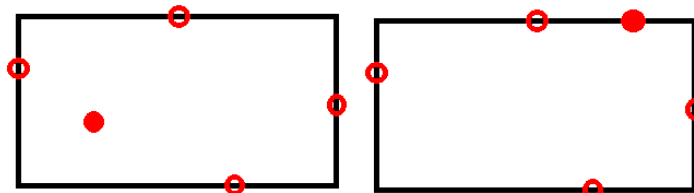


Figure 3: Five points not shattered by a rectangular.



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 rectangular.

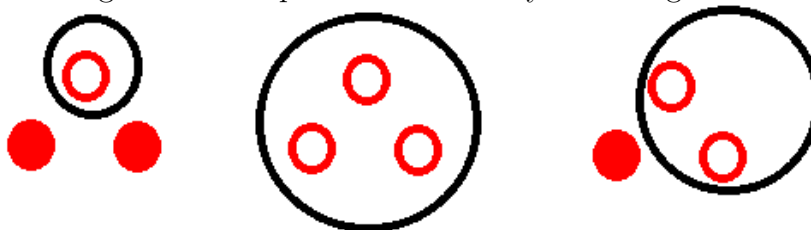
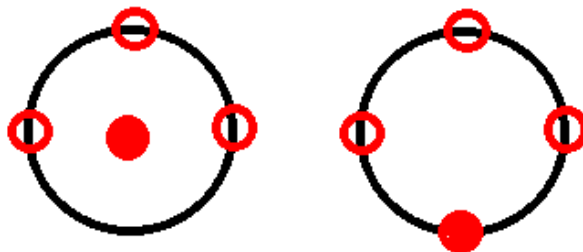


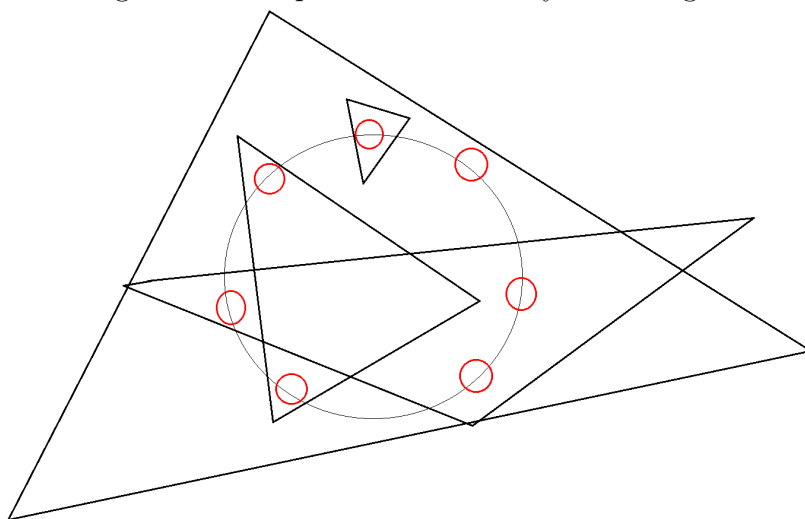
Figure 5: Five points not shattered by a rectangle.



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 rectangle.



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 ($\hat{e}(h)$) is given as the following:

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

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

Probability that the version space with respect to H_r and S (a sequence of training examples $m \geq 1$) 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 $0 < \delta < 0.5$.

$$\epsilon \leq \frac{1}{m}(\ln H + \ln \frac{1}{\delta}), \text{ so:}$$

$$\epsilon \leq \frac{1}{m}(\ln 4 + \ln \frac{1}{(1-0.95)})$$

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:

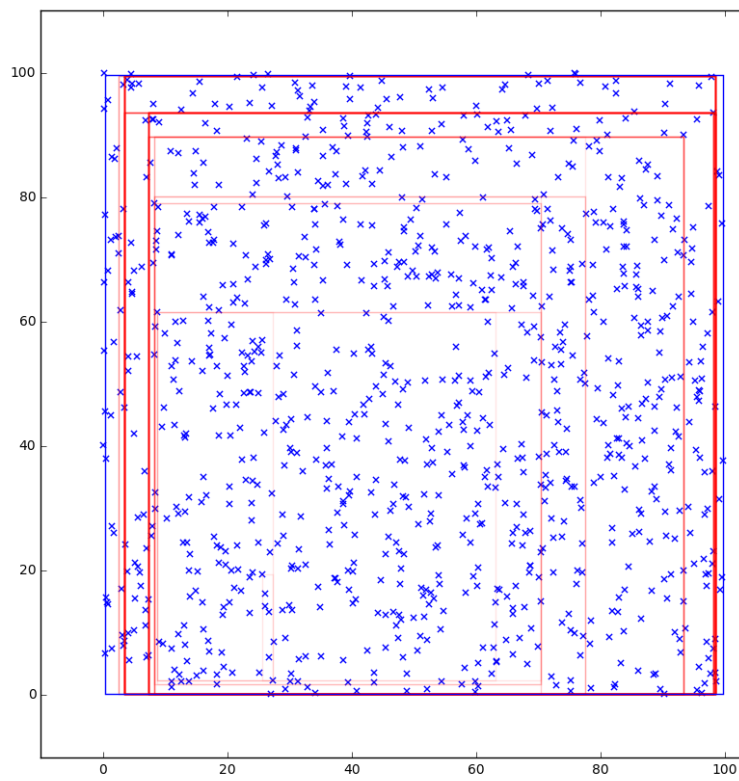
```

1 def learn_rect(samples):
2     a = np.min([s[0] for s in samples])
3     b = np.max([s[0] for s in samples])
4     c = np.min([s[1] for s in samples])
5     d = np.max([s[1] for s in samples])
6     return a, b, c, d

```

The concept is a rectangle (0, 0, 100, 100) and the samples are drawn from 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 correspondence:

Figure 8: Improving hypothesis rectangles covering the concept space.

