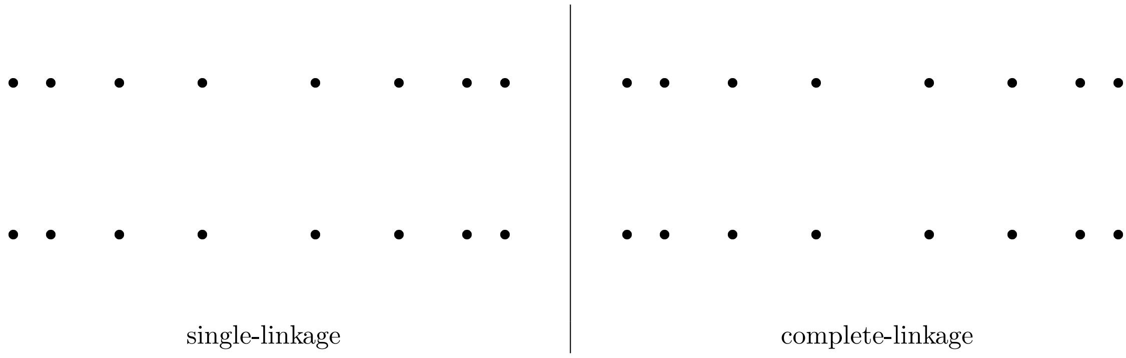
3 Exercise Session 3

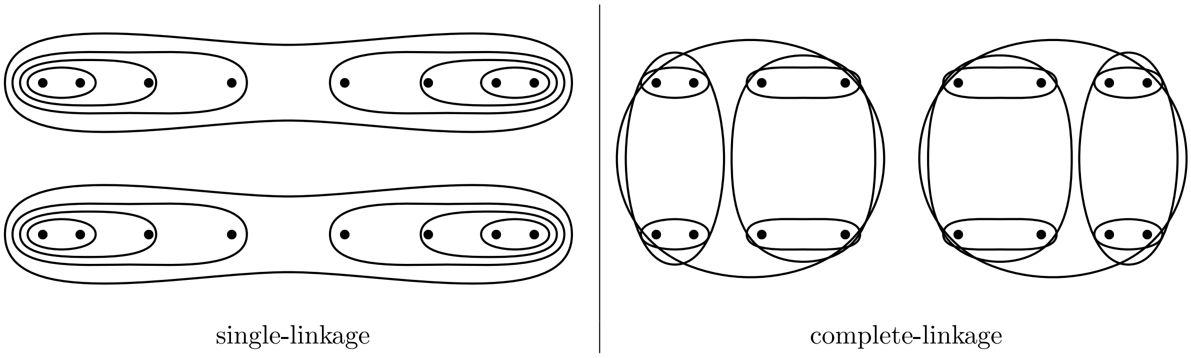
Exercises on clustering, hypothesis evaluation, computational learning theory, artificial neural networks and support vector machines.

3.1 Clustering

Consider the data set of points below. Draw the hierarchical clusters obtained by performing agglomerative clustering using single-linkage and complete-linkage.



Solution



3.2 Confidence intervals

1. **Confidence intervals on differences of accuracy between hypotheses**

   Suppose you have two hypotheses and want to know which one performs best. Both have been tested on a different test set of 40 examples. Results: hypothesis H_1 makes 24 correct predictions, hypothesis H_2 made 34 correct predictions.

   Build a 95% confidence interval for the difference in predictive accuracy between both hypotheses.

2. **Confidence interval on the error rate**

   A hypothesis h makes 10 errors on 65 predictions. Build a 90% confidence interval for its error rate. Give an upper bound u such that $error_{\mathcal{D}}(h) < u$ with 95% confidence. What is the 90% confidence upper bound?

Solution

1. Using accuracy estimates $\hat{p}_1$ and $\hat{p}_2$: the difference is estimated as

$$\hat{p}_1 - \hat{p}_2 \pm 1.96\sqrt{\frac{\hat{p}_1(1-\hat{p}_1)}{n_1} + \frac{\hat{p}_2(1-\hat{p}_2)}{n_2}}$$

which gives (approximately) $[-0.45, -0.05]$, i.e., the second hypothesis is most probably better than the first and the difference may vary from relatively small to quite large.

2. $\hat{p} = 10/65 = 2/13 = 15.4\%$; $\sigma_{\hat{p}} = 4.475\ \%$
   90% conf. int: $z_{90} = 1.64$, yields $15.4\% \pm 7.3\%$
   95% upper bound: $z_{90} = 1.64$, yields $15.4\% + 7.3\% = 22.7\%$
   90% upper bound: $z_{80} = 1.28$, yields $15.4\% + 5.7\% = 21.1\%$

3.3 Comparing two hypotheses on the same data

Now assume that you have tested two hypotheses on the same set of examples, and that the results are as follows:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
real	+	+	+	+	+	+	+	+	-	-	-	-	-	-	-	-
H_1	-	+	-	+	-	-	+	+	-	+	-	-	-	+	+	+
H_2	+	+	+	+	+	-	+	+	-	+	-	-	+	-	-	-

Apply McNemar's test for changes to compare the two hypotheses. Use the binomial distribution to determine how significant the result is. (Hint: the significance can be expressed as the probability that you would obtain results at least as extreme as the ones you have obtained if both hypotheses were equally good). What do you conclude?

The binomial distribution $P(x)$ is the probability of having x successes in n experiments if the probability of success is p_0.

$P(x) = \binom{n}{x} \cdot p_0^x \cdot (1 - p_0)^{n-x} \qquad \binom{n}{x} = \frac{n!}{x! \cdot (n-x)!}$

Solution

Assume a contingency table:

		h_1 correct	wrong
h_2	correct	a	b
	wrong	c	d

Then the hypotheses that we are testing are:

	Machine learning point of view	*Statistical point of view*
H_0	Classifiers h_1 and h_2 are equally good (or bad) $P(h_1 \text{ correct}) = P(h_2 \text{ correct})$ and $P(h_1 \text{ wrong}) = P(h_2 \text{ wrong})$	Homogeneity of outcomes $P(a) + P(b) = P(a) + P(c)$ and $P(c) + P(d) = P(b) + P(d)$ $\Leftrightarrow$ $P(b) = P(c) = 0.5$
H_A	Classifiers h_1 and h_2 are **not** equally good	$P(b) \neq P(c)$

We are using the binomial distribution P: each instance where h_1 and h_2 disagree is an *experiment* and we (arbitrarily) consider each instance where h_1 is correct a *success*. Hence, $n = 7$; observed counts of b and c are $b' = 6$ and $c' = 1$.

Now we compute the statistical significance, i.e. the probability that given our assumption H_0, two equally good classifiers disagree at least this much on 7 instances. Note that for this choice of H_A: $P(b) \neq P(c)$, we perform a two-tailed test, i.e. the extreme events we consider are where h_1 is correct on 1 instance or less **or** h_2 is correct on 1 instance or less, alternatively the number of successes $x \leq c'$ or $x \geq n - c'$:

$$P(x=0) + P(x=1) + P(x=6) + P(x=7) =$$

$$\binom{7}{0} 0.5^0 \ (1-0.5)^7 + \binom{7}{1} 0.5^1 \ (1-0.5)^6 + \binom{7}{6} 0.5^6 \ (1-0.5)^1 + \binom{7}{7} 0.5^7 \ (1-0.5)^0 =$$

$$0.5^7 + 7 \cdot 0.5^7 + 7 \cdot 0.5^7 + 0.5^7 = 16 \cdot 0.5^7 = 0.125 = \frac{1}{8}$$

In other words, if you compare two hypotheses that actually are equally good, there is a 1/8 chance of a result that is in favor of one hypothesis or the other at least as strongly as this one.

Now, following the statistical hypothesis testing procedure, we compare this value with the selected significance level α and stick to or reject H_0. For example, for the most common choices of $\alpha = 0.1, 0.05, 0.01$, we would conclude that given the data at hand, there is no sufficient evidence to reject the null hypothesis and hence we could keep assuming that classifiers h_1 and h_2 are equally good (or bad).

Furthermore, note that *(a)* simply computing the probability of the observed data, i.e. $P(x = 1)$, would underestimate the significance; and *(b)* for certain other choices of H_A, we would perform a one-tailed test, e.g. for H_A: $P(b) > P(c)$, that is h_2 is better than h_1, significance would be $P(x = 0) + P(x = 1)$.

3.4 ROC Curves

Given below is the real classification of 13 instances and the prediction made by classifiers A and B and a rank classifier C. Remember that a rank classifier can be turned into an ordinary classifier by providing a threshold; C is considered to predict + if its prediction is above the threshold, − otherwise.

	1	2	3	4	5	6	7	8	9	10	11	12	13
real	+	+	+	+	+	+	+	-	-	-	-	-	-
A	+	+	-	-	+	+	-	-	+	-	-	-	-
B	+	+	+	+	-	+	+	-	+	-	+	-	-
C	0.8	0.9	0.7	0.6	0.4	0.8	0.4	0.4	0.6	0.4	0.4	0.4	0.2

a. Plot A, B, and C on an ROC diagram.

b. Let $P(+) = P(-) = 0.5$, the cost of predicting a negative example to be positive $C_{FP} = 1$ and the cost of predicting a positive example to be negative $C_{FN} = 5$. Which classifier is best: A, B, or C used with a threshold of 0.5? Draw an “equal cost” line in the ROC diagram.

c. Draw the convex hull of the classifiers A, B and C.

d. Which classifiers are never optimal?

e. Which classifiers are optimal in a certain environment?

Solution

a. ROC diagram: Figure 1.

b. The expected cost of a classifier’s prediction is given by $\widehat{C} = C_{FP} \cdot FP \cdot P(-) + C_{FN} \cdot FN \cdot P(+)$. With $FN = 1 - TP$ we can write: $TP = \frac{C_{FP}P(-)}{C_{FN}P(+)} \cdot FP + \frac{C_{FN} \cdot P(+) - \widehat{C}}{C_{FN}P(+)}$. Substituting $C_{FP}/C_{FN} = 1/5$ and $P(-)/P(+) = 1$ we obtain $TP = 1/5 \cdot FP + c^{te}$. Lines of equal cost are therefore straight lines with a slope of 1/5. Drawing such a line makes clear that among the proposed classifiers B is best. This can also be seen by just computing the costs of the predictions on this data set: $\widehat{C}(A) = 1.16$, $\widehat{C}(B) = 0.52$, $\widehat{C}(C_{0.5}) = 0.797$. Also note that if a lower threshold would have been chosen for C, C could have been better.

c. The convex hull is shown with a dotted line on the ROC diagram.

d. Classifier A is never optimal because it is not on the convex hull.

e. Classifier B is optimal for example if $P(-)/P(+) = 1$ and $C_{FP}/C_{FN} = 3/7$. Also classifier C is optimal for certain environments. This is because B and all points of C are on the convex hull. Note that $C_{0.5}$ will never be better than both B and $C_{0.65}$.

3.5 VC-Dimension

Consider a 2-dimensional instance space. Take as a hypothesis space, the set of all hypotheses of the form “everything inside rectangle R is positive and everything outside it is negative”, where R can be any axis-parallel rectangle (that is, the sides of the rectangle are parallel to the X and Y axis). This is similar to the illustrations of rule learning in the lectures.

Show that the VC-dimension of this hypothesis space is at least 3.

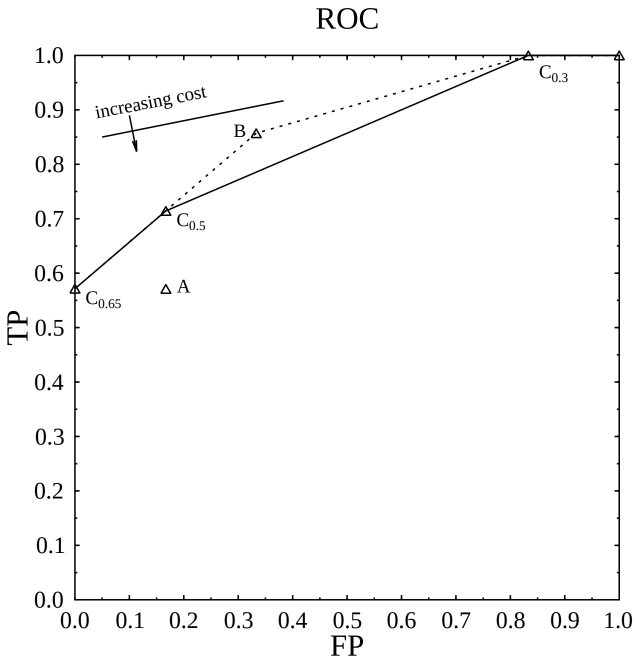


Figure 1: ROC diagram.

Optional: Show that the VC-dimension of this hypothesis space is at least 4.

Now consider a similar hypothesis space, but for each hypothesis that states "everything inside rectangle R is positive and everything outside it is negative" there is also a hypothesis "everything outside R is positive and everything inside is negative". Show that the VC-dimension of this hypothesis space is at least 4.

Solution

Consider 3 equidistant points (i.e., they form a triangle with 3 sides of equal length, see also Figure 2). For any assignment of + and − labels to these points you can draw a rectangle around the + points that does not include the − points. Hence, these 3 points can be shattered, so the VC-dimension must be at least 3.

Consider the configuration of 4 points in Figure 3. Enclosing any 3 out of these 4 with a rectangle is trivial. Enclosing any pair is also possible: for the "problematic" case of the upper and lower points, the rectangle can be made narrow enough so that it does not cover the leftmost and rightmost. For the other "problematic" case, a broad and low rectangle can be drawn such that it encloses just the leftmost and rightmost points. All other cases can be covered by drawing a rectangle that covers a leftmost/rightmost upper/lower quadrant.

Now consider 4 points as shown in Figure 2. For any labeling of these points with + and − you can draw a rectangle around the pluses OR a rectangle around the minuses (but usually not both). So these points can be shattered by the second hypothesis space, but not by the first. The VC-dimension of the second hypothesis space must therefore be at least 4.

3.6 Sample complexity

Using the concept of VC-dimension, compute an upper bound for the number of examples that may be needed to train a 2-input perceptron such that with 90% certainty it learns a hypothesis with true error $< 5\%$.

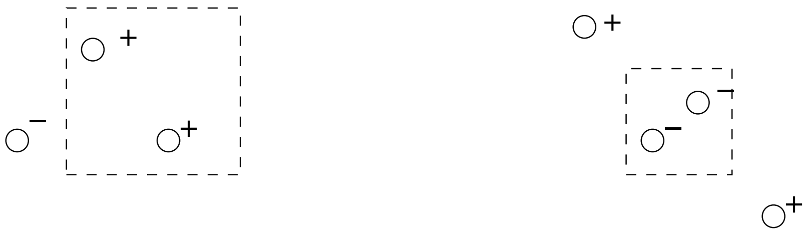


Figure 2: Shattering points with rectangles. (a) To the left, 3 points are shown, and a rectangle covering 2 out of these 3 points. For whatever +/− label assignment to the points, a rectangle consistent with it exists. (b) To the right, 4 points. If we label the "middle points" negative and the other positive, no rectangle exists that covers the positives and no negatives, but a rectangle exists that covers the negatives and no positives.

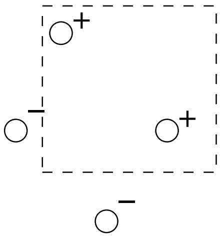


Figure 3: Here, 4 points are shown, and a rectangle covering 2 out of these 4 points. For whatever +/− label assignment to the points, a rectangle consistent with it exists. It should be clear that the VC-dimension of this hypothesis space is at least 4.

Solution

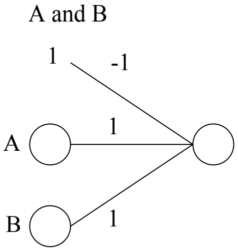
The VC dimension of a perceptron is 3, see course. Using the bound with VC dimension:

$$m \geq 1/\epsilon(4\log_2 2/\delta + 8VC(H)\log_2 13/\epsilon) = 20(4\log_2 20 + 8 * 3 * \log_2 260) > 4000$$

This bound does not seem very realistic.

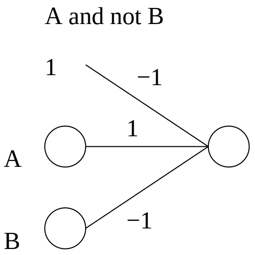
3.7 Logical Concepts with Perceptrons

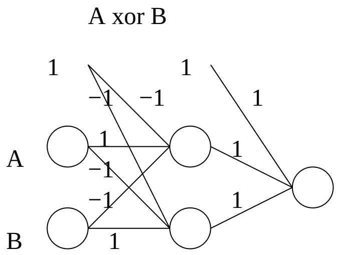
Typically, when perceptrons are used to implement boolean functions, `true` and `false` are encoded as 1 and −1 respectively, and the step transfer function is used. This is a perceptron that implements logical conjunction:



Design a two-input perceptron that implements the boolean function $A \wedge \neg B$, and a 2-layer network of perceptrons that implements A XOR B.

Solution





3.8 Decision Regions of Neural Networks

Consider a two-dimensional space XY; X and Y are inputs of a perceptron. We have seen that the decision surface of a perceptron is always a straight line. We also know that perceptrons can apply OR and AND operations to boolean values.

Now consider a neural network with two layers of perceptrons (a hidden layer with n perceptrons and an output layer of 1 perceptron). Based on the above observations, what kind of decision surface do you think such a network can at least form?

Now look at Figure 10.12 on page 187 in the course text and compare the decision regions seen there to the results you just obtained. What is your conclusion?

Solution

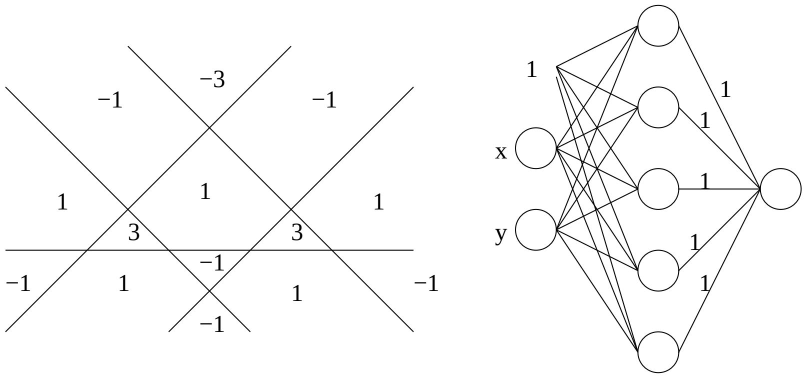


Figure 4: Output of a simple network with 5 hidden nodes, where the output node is not thresholded. All output weights are assumed to be 1 here. What decision regions could you get in this case by adding a threshold to the last node?

2-layer case: each perceptron in the hidden layer represents a straight line; the perceptron fires if a signal is on one side of the line. Viewing the outputs of the hidden layer as booleans, the output perceptron can combine these booleans with an AND operation. It will fire if the input signal is to one specific side of each line. This allows the network to represent any convex polygon.

More generally, the network can form decision regions bounded by n straight lines, where n is the number of nodes in the hidden layer. A simple example is shown in Figure 4.

In Fig. 10.12, the decision region corresponding to each single output node of the network indeed resembles an area bounded by a number of lines, however the lines are not straight. This is due to the nonlinear character of the network (using sigmoid functions instead of strict threshold functions).

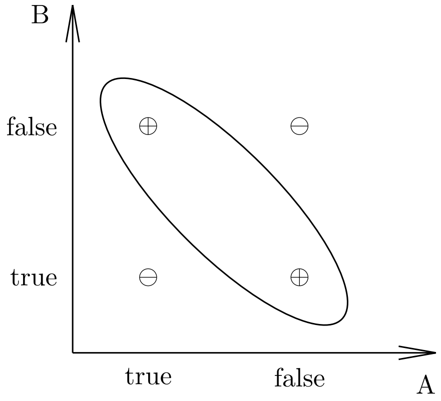


Figure 5: Decision surface for an SVM with a polynomial kernel.

3.9 Support Vector Machines

Is it possible to learn the XOR-function with a Support Vector Machine? If yes, under what conditions? If no, why not?

Solution

Yes, under the condition that a non-linear kernel-function is used. For example, the kernel $K(x, y) = (xy + 1)^2$ enables an SVM to use quadratic decision surfaces in a two dimensional space. Figure 5 shows how a quadratic surface can be used to learn the XOR-function in a two dimensional space.

See also the animated demonstration by Ehud Aharoni: `https://www.youtube.com/watch?v=3liCbRZPrZA`.

3.10 Using Weka: Homework

For more information on Weka, see Exercise 1.9 from Session 1. This exercise assumes you have installed Weka on your computer.

1. As in Exercise 1.9, download and unzip the file `datasets.zip` from Toledo (`Course Documents/Session 1/Weka Data Sets/`). Open the `weather.arff` file in the Weka Explorer.
2. Go to 'Cluster' tab and select the clustering algorithm SimpleKMeans. Click on the line behind the choose button. This shows you the parameters you can set and a button called 'More'. Pay attention to the choice of the distance function and the number of clusters.
3. In 'Cluster mode', select 'Classes to clusters evaluation' and attribute 'play'.
4. Click the start button to run the algorithm. What cluster centroids are obtained? Do clusters correspond to classes?
5. Right-click on the line in 'Result list' and select 'Visualize cluster assignments'. Try various axes to visualize the clustering. (Hint: Use the 'Jitter' slider to help visualize nominal attributes.)
6. (Optional) Vary the number of clusters or/and distance functions. Interpet the obtained results.