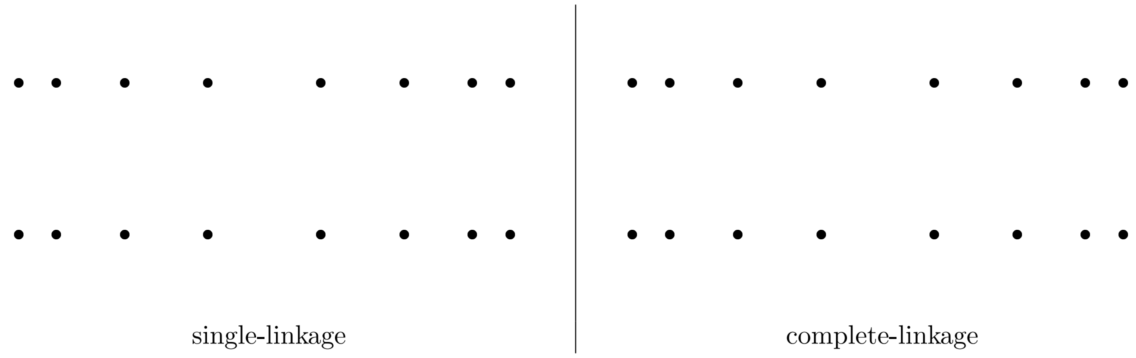
3 Exercise Session 3

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.



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?

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.

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

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?

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.

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

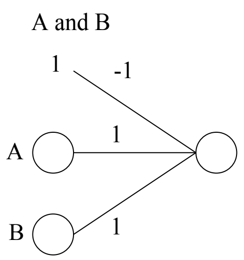
Now consider a similar hypothesis space, but for each hypothesis that states "everything inside rectangle R is positive and everything outside it is negative" there is also a hypothesis "everything outside R is positive and everything inside is negative". Show that the VC-dimension of this hypothesis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

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 $< 5\%$.

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.

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?

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?

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.