Machine Learning Exam

With Answers

27 March 2013

The exam is **open book**: you can use Alpaydin's "Introduction to Machine Learning" (as a print of the PDF or as a proper book) as well as the lecture slides and any notes you've taken. You can use a calculator. The questions add up to *80 points* in total.

Good luck!

Answers in bold italics.

Questions

1. **Short answers, no justification required** *(3 Points for each question)*

   (a) (**True** or **False**): When a decision tree is grown to full depth, it is more likely to fit the noise in the data.
   True

   (b) (**True** or **False**): When the hypothesis space is richer, overfitting is more likely.
   True

   (c) (**True** or **False**): Support Vector Machines explicitly maximise the distance between training instances and the separating hyperplane.
   True

   (d) (**True** or **False**): Instance-based methods such as k-Nearest Neighbour cannot handle categorical inputs such as colour.
   False

   (e) (**True** or **False**): k-Means clustering requires labelled (e.g. positive/negative) training data.
   False

2. **Decision Trees**

   (a) *(5 points)* Suppose we are training a decision tree on a dataset that contains f binary features. The dataset contains a very large number of examples ($N \gg f$). What is the maximum depth of the decision tree?

The maximum depth is f. Each attribute can be used once at most to define a split; after that, no further differentiation among records is possible (the second termination condition for the decision tree algorithm as presented in the lecture).

(b) *(5 points)* Consider training a decision tree on a similar dataset with f real-valued features. What can you say about the maximum depth now?

The crux is that real-valued features introduce the need for thresholded splits. This means that individual attributes may be used more than once with new thresholds. This means that f no longer determines the maximum depth, but N does. Answers like 'unlimited depth' are also considered valid.

(c) *(5 points)* Some implementations of the decision tree algorithm terminate if all features have information gain below a certain threshold. Does this help prevent overfitting?

It could help, since it reduces the space of possible hypotheses (limits the expressivity of the decision trees).

(d) *(5 points)* Can you mention a drawback of the stopping criterion described above?

The motivation I gave in the lecture is that the XOR problem cannot be solved when terminating in this situation.

3. **Instance-Based Learning**

(a) *(5 points)* In instance-based learning, you can scale distances for some features by multiplying their values with some factor. Suppose that you select a scale factor of 0.001 for some feature f. How does this influence the relative importance of f for classifications made with the resulting model?

f becomes less important because the distances between cases in terms of f become smaller.

(b) *(3 points)* Consider the dataset in table 1 with one real-valued input x and one binary output y. We are going to use k-nearest neighbour with unweighted Euclidean distance to predict y for a given x.

What is the predicted class of 1-NN for a new data-point $x = 1.5$?

Table 1: Instance-based learning data set

X	Y
-0.1	-
0.7	+
1.0	+
1.6	-
2.0	+
2.5	+
3.2	-
3.5	-
4.1	+
4.9	+

- because the nearest neighbour is 1.6

(c) *(3 points)* What is the predicted class of 3-NN for that same data-point $x = 1.5$?

+ because the nearest neighbours are 1.6 (-), 1.0 (+) and 2.0 (+)

(d) *(4 points)* Leave-one-out cross-validation is cross-validation using each separate data-point as the hold-out sample once, i.e., first use $\langle -0.1, - \rangle$ as the test set, then $\langle 0.7, + \rangle$, and so on.
What is the leave-one-out cross-validation error of 1-NN on this dataset? Give your answer as the number of misclassifications.

4 misclassifications: the points where the nearest neighbour has a different class than the points themselves. These are $0.1, 1.6, 2.0$ *and* 4.1.

4. **Clustering**

(a) *(5 points)* The k-Means algorithm assigns points to the cluster defined by the nearest centroid. Can you explain why this algorithm is sensitive to outliers (data points with extreme values)?

Outliers assigned to a cluster have a relatively high influence on the centroid position because the centroid will shift towards them as the algorithm minimises the sum of squared Euclidean distances within the cluster. Thus, outliers may cause the centroid to be moved away from the true cluster centre.

(b) *(5 points)* Propose a way to make the calculation of the centroids more robust to outliers.

simple acceptable answers would be:

- ***taking the median of the points instead of the mean***

- ***when averaging points to recalculate the centroid, weighting them by their distance (using a kernel function) to the centroid (as it is currently defined)***
- ***taking random samples of the dataset instead of the whole dataset in each iteration (with a small enough subset that would preserve the shape of the cluster, but would make the inclusion of the outliers more unlikely). This one has the advantage of also reducing k-Means' significant computational cost***

	A	**B**	**C**	**D**	**E**
A	0				
B	2	0			
C	4	3	0		
D	10	7	9	0	
E	8	5	6	1	0

Table 2: Distances between 5 data points

(c) *(5 points)* Consider the set of 5 data points in table 2: it lists the distances between the data points. Draw the dendrogram that would result from a single-link agglomerative hierarchical clustering of it.

See Fig. 1.

5. **Hypothesis Comparison and Cross Validation**

(a) *(5 points)* Explain why it is not enough to simply report hypothesis performance (for instance, the error rate), but you should also report a confidence interval or a similar statistical measure.

The measured performance of a hypothesis depends on the sample (e.g., the test set) on which it was tested. Even if we assume that the test set consists of independently drawn cases from the true population, the measured performance is an estimate of the true performance. It is necessary to give an indication of the reliability of this estimate, for instance with a confidence interval.

(b) *(5 points)* One of your lab partners suggests adapting the predictions made with a linear classifier to allow for graded outcomes that may deliver a smoother ROC curve: she suggests outputting for each record in the validation database the distance from the decision boundary rather than just -1 or 1 depending on which side of the boundary that record is on. Thus, records classified

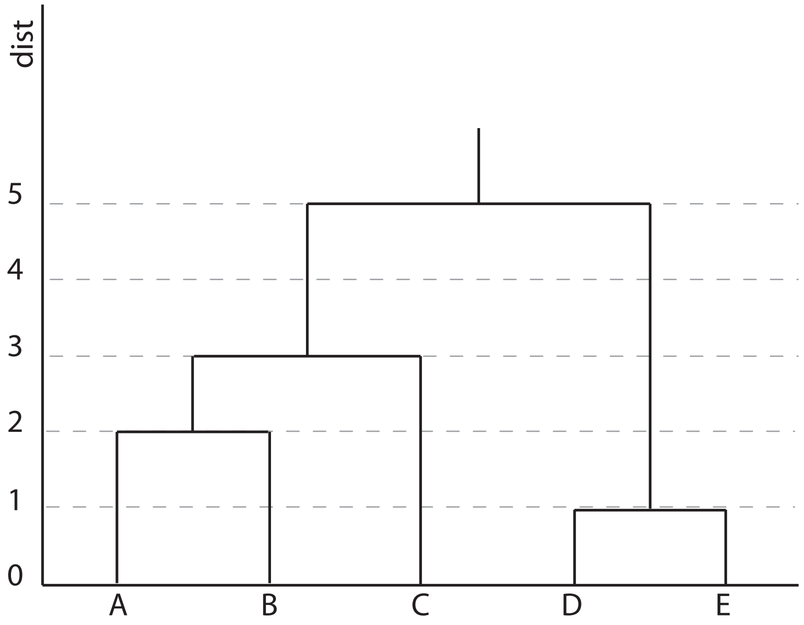

Figure 1: The dendrogram from table 2.

as positive would have a positive value that increases with the distance to the decision boundary, while records classified as negative would have a negative value that decreases similarly.

Is this a good idea? Why (not)?

It would indeed result in smoother ROC curves because it allows for a more fine-grained build-up of the curve. The essential insight here is that the distance from the decision boundary can be interpreted as an indication of the confidence of the classification: if a case lies further away, the classifier is less likely to be mistaken than when it lies on or close to the boundary.

(c) *(5 points)* Can you suggest a similar modification for decision trees?

To give an estimate of confidence for a decision tree instead of just a classification, the most straightforward idea is to output the likelihood of each class in the leaf of the tree that each case belongs to. This is analogous to the output calculation for regression trees.