Assignments 2

Joaquin Vanschoren, Mykola Pechenizkiy, Anne Driemel

1 Trees

1.1 Exercise 1: Experiment with trees (6 points, 1 each)

Import the Lymphography dataset from OpenML (http://www.openml.org/d/10) into R or Python. Then do the following:

- Study the structure of the dataset. For instance, plot the class distribution, explore which features seem important, etc.
- Remove the smaller classes so that you get a binary problem.
- Build a decision tree learning using CART (rpart in R) and one or several other decision tree learners such as C4.5 (j48 in R). Evaluate and visualize it if possible. Evaluate and visualize it if possible. Note: SKlearn only has CART (DecisionTreeClassifier) and a randomized tree (tree.ExtraTreeClassifier).
- How 'stable' are the trees returned by CART? What happens when you train it on random samples of the dataset?
- What is the difference in performance between pruned and unpruned trees? How much smaller are they?
- Is one of the decision trees better suited for this dataset? Why?

1.2 Exercise 2: Classification trees (1 point)

Consider the following toy training set:

Example	Sky	AirTemp	Humidity	Wind	Water	Forecast	Date?
1	sunny	warm	normal	strong	warm	same	yes
2	sunny	warm	high	strong	warm	same	yes
3	rainy	cold	high	strong	warm	change	no
4	sunny	warm	high	strong	cool	change	yes
5	sunny	warm	high	weak	warm	same	no

Learn a decision tree:

- What is the class entropy for the entire dataset?
- What is the information gain when you split the data using the Sky feature? Show your calculation.
- Construct a tree by following a simple decision tree learner:
  - Select a feature to split on according to its information gain. If multiple features are equally good, select the leftmost one.
  - Split the data and repeat until the tree is complete.

1.3 Exercise 3: Regression trees (1 point)

Consider the following toy training set S:

Example	a1	a2	Target
1	T	T	1.0
2	T	T	1.0
3	T	F	1.0
4	F	F	5.0
5	F	T	6.0
6	F	T	5.5

We are going to use weighted average variance of the subsets as the split heuristic H for every feature A:

$$H(A) = \sum_{v \in Values(A)} \frac{|S_{A=v}|}{|S|} \cdot Var[S_{A=v}]$$

where

$$Var[S_{A=v}] = \frac{1}{|S_{A=v}|} \cdot \sum_{i \in S_{A=v}} (Target_i - \overline{Target}[S_{A=v}])^2$$

$$\overline{Target}[S_{A=v}] = \frac{1}{|S_{A=v}|} \cdot \sum_{i \in S_{A=v}} Target_i$$

Which attribute (a1 or a2) will be put in the top node of the regression tree?

1.4 Exercise 4: Heuristics (2 points)

Let $p(k|t)$ be the percentage of elements of class k in node t of a classification tree. A typical classification rule is then predict the class with the highest $p(k|t)$:

$$\hat{k}|t = \arg\max_k p(k|t)$$

Now assume that instead of this rule, you randomly predict the class according to its estimated probability $p(k|t)$, hence class k is predicted with a probability $p(k|t)$.

- What is the estimated percentage of cases where class k is correct, but another class is predicted instead?
- What is then the estimated misclassification probability for the entire leaf t? Does this look familiar?

---

2 Evaluation

2.1 Exercise 1: ROC curves (4 points (1+2+1))

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 (e.g. 0.5). C is considered to predict + if its prediction is above the threshold, - otherwise.

example	1	2	3	4	5	6	7	8	9	10	11	12	13		
true label	+	+	+	+	+	+	+	-	-	-	-	-	-		
prediction A	+	+	-	-	+	+	-	-	+	-	-	-	-		
prediction B	+	+	+	+	-	+	+	-	+	-	+	-	-		
prediction 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		

Do the following:

- Plot A, B, and C on a ROC diagram. Use a number of different thresholds for C (at least 3).
- Assume that the classes are balanced, hence $P(+) = P(-) = 0.5$. The cost of a false positive and false negative are $C_{FP} = 1$ and $C_{FN} = 5$. Which classifier is best: A, B, or C with a threshold of 0.5? Show geometrically in the ROC diagram which models are optimal under this cost function.
- Draw the convex hull of the classifiers A, B, C. Which classifiers are never optimal? Which classifiers are optimal in a certain environment?

2.2 Exercise 2: Model selection (4 points (1+1+1+1))

Generate a 2-dimensional dataset with 1000 examples. Use, for instance, mlbench.threenorm in R or make_blobs in Python. Then run a k-Nearest Neighbor classifier, and optimize the value for k (keep k < 50).

- Use 10-fold crossvalidation and plot k against the misclassification rate. Which value of k should you pick?
- Do the same but with 100 bootstrapping repeats. What does the bias-variance trade-off look like for low and high values of k?
- What if you generate a dataset with 10000 examples, is the result still the same?
- Repeat step 1-3, this time optimizing the hyperparameters of a decision tree. Choose the hyperparameter(s) that you think have the largest impact and use sensible value ranges.

2.3 Exercise 3: Optimization (3 points (1+1+1))

Import the PenDigits dataset from OpenML (http://www.openml.org/d/32) into R or Python, and then create a separate test set with 20% of the instances using random stratified sampling. Then do the following:

- Optimize the main hyperparameters (at least 2) of a decision tree learner (or SVMs) with a random search. Use nested resampling on the training set to obtain a clean evaluation. Evaluate your optimized hyperparameter settings on the separate test set.
- Do the same, but this time don't use nested resampling: just optimize the hyperparameter settings on the training data. Do you get different optimized parameters? Also evaluate these on your separate test set. Which approach yields the best results? Explain your findings.
- Optimize these hyperparameters again (using nested resampling), but replace random search with a more intelligent approach, e.g. iterated F-racing. Plot the number of evaluations against the performance of the best hyperparameters up till then, for both approaches. Which approach finds good hyperparameter settings faster?

---

3 Kernel methods

3.1 Exercise 1: Kernel selection (2 points (1+1))

Generate a 2-dimensional dataset with 1000 examples. Use, for instance, mlbench.threenorm in R or make_blobs in Python. Study the effect of the choice of kernel by visualizing the results.

- Train an SVM on the entire dataset using a linear, polynomial and RBF kernel and visualize the results. Also evaluate the predictions using 10-fold cross-validation and AUC.
- Vary the C and the γ parameters and interpret the (visualized) results. Also explore some extreme values. Explain as good as possible what is happening.

3.2 Exercise 2: Landscape analysis (3 points (1+1+1))

Study how SVM hyperparameters interact on the Ionosphere dataset. Running this experiment can take a while, so start early and use a feasible grid search.

- Do a grid search to optimize the γ parameter of the RBF kernel in an SVM, using a 10-fold cross-validation. Use a log scale and keep C at its default value. Plot the results (γ vs. AUC performance).
- Train an SVM with the RBF kernel, and vary both C and γ on a log scale from 2^{-15} to 2^{15}. Explore how fine-grained this grid/random search can be, given your computational resources.
- Visualize the results in a plot $C \times \gamma \rightarrow \rho$ with ρ being the performance of the model (e.g. AUC) visualized as the color of the data point.

3.3 Exercise 3: Maximal margin classifier (2 points (0.5 points each))

Consider the following toy training set S:

Example	X_1	X_2	Target
1	3	4	Red
2	2	2	Red
3	4	4	Red
4	1	4	Red
5	2	1	Blue
6	4	3	Blue
7	4	1	Blue

- Sketch the observations. Sketch the optimal separating hyperplane, and provide the equation for this hyperplane of the form $\beta_0 + \beta_1 X_1 + \beta_2 X_2 = 0$. Indicate on your sketch the margin for the maximal margin hyperplane, as well as the support vectors of the maximal margin classifier.
- Describe the classification rule for the maximal margin classifier based on the equation you derived above. When do we classify Red, and when Blue?
- Will a slight movement of the 7th example affact the maximal margin hyperplane? Discuss.
- Draw an additional observation on the plot so that the two classes are no longer linearly separable by a hyperplane. Explain why.

3.4 Exercise 4: Kernels (3 points (1+1+1))

Prove the following statements about kernels:

- If k_1 and k_2 are kernel functions, and $\alpha, \beta \geqslant 0$, then $k(x, z) = \alpha k_1(x, z) + \beta k_2(x, z)$ is also a kernel function.
- If k is a kernel function, then $k(x, x) \geqslant 0 \ \forall x \in X$
- If k is a kernel function and $k(x, x) = 0 \ \forall x \in X$, then $k(v, w) = 0 \ \forall v, w \in X$