A Collection of Machine Learning Exercises

Barbora Hladká — Martin Holub — Vilém Zouhar

April 15, 2019

Contents

Acknowledgement

This work was supported by the Faculty of Mathematics and Physics, Charles University by the Student Faculty Grant *Sbírka příkladů ze strojového učení*, 2018.

Introductory remarks

Part I – Analytical tasks

Task 1
Binary classifier evaluation for different cut-off score values

Exercise 1a Accuracy, precision, and recall of classifier

Consider the table below where true classes of eight test examples are presented. In addition, their scores assigned by a predictor are provided. For a final binary classification of the test examples to $+1$ and -1 classes use this rule: if the score is equal to or greater than a given cut-off score, a classifier predicts $+1$, otherwise -1. Work with the cut-off score values 5, 3, 1, -3, -6 and for each value compute accuracy, precision, and recall of the classifier.

example	true class	score
$\mathbf{x}_1$	$+1$	7
$\mathbf{x}_2$	$+1$	4
$\mathbf{x}_3$	-1	2
$\mathbf{x}_4$	-1	1
$\mathbf{x}_5$	-1	-1
$\mathbf{x}_6$	$+1$	-4
$\mathbf{x}_7$	-1	-5
$\mathbf{x}_8$	-1	-6

Exercise 1b Receiver Operating Characteristic Curve of a Classifier

Plot an ROC curve of the classifier developed in the previous exercise.

Task 2
Confusion matrix for a binary classifier

Exercise 2a Confusion matrix

A binary classifier was evaluated using a set of 1,000 test examples in which 50 % of all examples are negative. It was found that the classifier has 60 % sensitivity and 70 % accuracy. Write the confusion matrix.

Exercise 2b Compute Precision, F_1-measure, and Specificity of Classifier

Using the confusion matrix created in the previous exercise compute the classifier's precision, F_1-measure, and specificity.

Task 3
Precision and ROC curve

Exercise 3a Point on ROC

To test a binary classifier, a data set consisting of 100 positive and 400 negative examples was used. It turned out that the ROC curve goes through the point TPR = FPR = 0.2. Calculate Precision at this point.

Task 4
Linear regression evaluation

Exercise 4a Variance explained

The table below provides a training data set containing five examples, a feature X, and a qualitative target attribute Y.

Example	X	Y
1	0	5
2	1	4
3	3	3
4	5	2
5	6	1

What is the proportion of the explained variance in the target attribute Y by the linear regression on $\mathbf{x}$ if the regression parameter estimates are $\hat{\theta}_0 = 5$ and $\hat{\theta}_1 = -1$?

Task 5
Ensemble classifier

Consider a binary classification problem and three binary classifiers C_1, C_2, and C_3 with statistically independent results. Single classifiers have the following error rates: $e_1 = 40\%$, $e_2 = 40\%$, $e_3 = 50\%$, resp. Using the three classifiers we can build a majority voting ensemble.

Exercise 5a Improving accuracy

Will the ensemble help to improve the classification accuracy? Why?

Exercise 5b Accuracy

What will be the ensemble classifier accuracy?

Part II – Experiments with the College data set

The tasks in this part relate to the `College` data set, which is part of the `ISLR` package. This data set contains statistics for 777 US Colleges from the 1995 issue of US News and World Report. The following 18 attributes are used. Their description is taken from the `ISLR` documentation.[1]

Attribute	Description
`Private`	A factor with levels No and Yes indicating private or public university
`Apps`	Number of applications received
`Accept`	Number of applications accepted
`Enroll`	Number of new students enrolled
`Top10perc`	Pct. new students from top 10 % of H.S. class
`Top25perc`	Pct. new students from top 25 % of H.S. class
`F.Undergrad`	Number of fulltime undergraduates
`P.Undergrad`	Number of parttime undergraduates
`Outstate`	Out-of-state tuition
`Room.Board`	Room and board costs
`Books`	Estimated book costs
`Personal`	Estimated personal spending
`PhD`	Pct. of faculty with Ph.D.'s
`Terminal`	Pct. of faculty with terminal degree
`S.F.Ratio`	Student/faculty ratio
`perc.alumni`	Pct. alumni who donate
`Expend`	Instructional expenditure per student
`Grad.Rate`	Graduation rate

To run the code snippets in the following tasks, `islr` package should be installed first.

```
library(islr)
```

[1] https://cran.r-project.org/web/packages/ISLR/ISLR.pdf

Task 7
Linear regression

Perform a multiple linear regression using all the features to predict `Apps`. Provide an interpretation of each hypothesis parameter in the model.

Task 8
Data preprocessing for classification

Exercise 8a New binary target attribute

Create a binary attribute `Apps01` that contains a 1 if `Apps` contains a value equal to or above its median, and a 0 if `Apps` contains a value below its median. Create a single data set `d` containing both `Apps01` and the other College features. Compute entropy of `Apps01`.

Exercise 8b Feature scaling

Scale all the features using `scale()` function.

Exercise 8c Data split

Split the data into `d.train` training set and `d.test` test set 80:20.

Exercise 8d MFC classifier

Make a trivial classifier without using the features to predict `Apps01` and evaluate it on `d.test`. Compute its accuracy.

Task 9
Logistic regression

To get the data for this exercise, you have to do Task 8 first.

Exercise 9a Running the algorithm

Perform logistic regression on `d.train` in order to predict `Apps01` using all the features on `d.test`.

Exercise 9b Error rate

Compute the training error rate of `model`. Produce a confusion matrix comparing the true target test values to the predicted target values. Compute the test error rate.

Exercise 9c Parameter interpretation

Provide an interpretation of each hypothesis parameter in `model`.

Task 10
Decision tree algorithm

To get the data for this exercise, you have to do Task 8 first.

Exercise 10a Running the algorithm

Perform decision tree algorithm on `d.train` to predict `Apps01`. Create a plot of the tree.

Exercise 10b Error rate

Compute the training error rate. Compute the test error rate.

Exercise 10c Tuning complexity parameter (cp)

Tune the cp parameter. Choose the best value of cp, and evaluate your model again. What is the best value of cp? Why? Explain it explicitly. Compute the accuracy of the model with your best cp.

Task 11
Naive Bayes algorithm

To get the data for this exercise, you have to do Task 8 first.

Exercise 11a Running the algorithm

Perform Naive Bayes algorithm on `d.train` to predict `Apps01` using all the features. Test it on `d.test`. Compute precision and recall.

Task 12
k-NN algorithm

To get the data for this exercise, you have to do Task 8 first.

Exercise 12a Spliting the data

Randomly split `d.train` into eight folds to perform 8-fold cross validation.

Exercise 12b Running the algorithm

Perform k-NN with several values of k in order to predict `Apps01` using all the features. Plot 8-fold cross validation error rate for different values of k.

Task 13
Ridge regularization

Consider `Apps` as the target value.

Exercise 13a Pearson correlation

Calculate the Pearson correlation coefficient between `Apps` and the numerical features. Put the features in descending order by the coefficient.

Exercise 13b

Split the data into a training set and a test set 80:20 and create a model for ridge regression using the training set with lambda chosen by cross-validation.

Exercise 13c Mean Squared Error

Evaluate the model on the test set using the Mean Squared Error measure (MSE).

Exercise 13d Feature scaling

Before training a Ridge regression model scale all the numerical features but the target attribute in the training and test sets created in 13a. Create a model for Ridge regression using the training set with lambda chosen by cross-validation. Evaluate the model on the test set using MSE and compare the result with the result from 13c.

Part III – Verb Pattern Recognition

Verb Pattern Recognition (VPR) is the task of verb meaning disambiguation. It is a task from the field of natural language processing. The goal of VPR is to classify a given verb in sentences with a pattern number. In the following tasks we focus on *cry* verb for which three patterns `1`, `4`, `7` are considered to describe its meaning.[2] We provide a sample sentence for each pattern:

`1` His advice to stressful women was: ‘ If you **cry**, do n’t cry alone.

`4` You can hear them screaming and banging their heads, **crying** that they want to go home.

`7` Identifying areas which **cry** out for improvement or even simply areas of muddle and ...

We use two more tags:

`u` for unclassifiable cases, i.e., a verb cannot be assigned to a particular pattern

`x` for not a verb, i.e. the given word occurring in the sentence is not a verb

Data sets provided for the VPR tasks

The development data set contains 224 examples where each sentence is represented by a unique sentence id (not important for the tasks), a feature vector (273 binary values) and a target pattern `tp` (categorical target value), $tp \in \{1, 4, 7, u, x\}$.

The gold data set contains 250 examples with true classifications represented as tuples `sentence_id;tp;` These examples were automatically classified using F1 classifier and 60 of them were manually annotated by four group of annotators. The six data sets are needed to address Tasks 15, 16, and 17:

	Size	**File**
Development data set	224 examples	`cry.devel.working.csv`
Gold data set	250 examples	`cry-gs.csv`
	classified by F1 classifier	`cry-F1.csv`
	60 examples annotated by group A	`cry-A.csv`
	60 examples annotated by group B	`cry-B.csv`
	60 examples annotated by group C	`cry-C.csv`
	60 examples annotated by group D	`cry-D.csv`

Brief comment on features

The features are divided into three feature families: morphological features (MF.*), syntactic features (STA.*), and semantic features (SEM.*).

[2] It is not necessary to explain why the patterns are called in this way.

Task 15
Target class and its properties

Exercise 15a Histogram

Draw a histogram for the target class in the gold standard data.

Exercise 15b Entropy

Compute the entropy of the target class in the gold standard data.

Task 16
Annotator agreement

Work with the annotations provided by the groups A, B, C, and D.

Exercise 16a Confusion matrix for annotators

Display the confusion matrix for groups A and B.

Exercise 16b Cohen's Kappa

Compute the Cohen's kappa value between groups A and B, A and C, A and D, B and C, B and D.

Exercise 16c Fleiss' Kappa

Compute the Fleiss's Kappa between groups A and B and C and D (all groups together).

Task 17
Classifier evaluation

Work with the gold dataset and the output of F1 classifier.

Exercise 17a Confusion matrix for classifier

Display the confusion matrix for F1 classifier.

Exercise 17b Accuracy of classifier

Compute F1's accuracy.

Task 18
Measure of feature frequency

Exercise 18a Feature frequency

For each binary feature compute its feature frequency and visualize distribution of feature frequencies using histogram.

Exercise 18b Contingency table

Create a contingency table for the feature frequencies.

Exercise 18c Constant features

Which features are constant, i.e. $fr(A) = 0$ or $fr(A) = 224$?

Exercise 18d Feature filtering

Assume the condition $min(fr(A), n - fr(A)) \geq frt$, where n is the dataset size, and frt is a parameter called frequency threshold $frt = 0, 1, \ldots, n$. Draw a plot that shows how the number of features that fulfill the given condition depends on the frt value (frt on x-axis, number of features on y-axis).

Task 19
Information gain

Exercise 19a Information gain of binary features

For each feature A compute the information gain $\mathrm{IG}(A, tp)$. Put the features in descending order by information gain. Visualize the information gain of the first 30 features.

Part IV – Virtual Ligand Screening

The task of *Virtual Ligand Screening* (VLS) comes from the field of cheminformatics. Virtual screening is a computational method used in early stage of the drug discovery process. It searches very large libraries of molecules to identify a selection of molecules that are most likely to bind to a target *receptor*, i.e. the target of the examined drug. The selected molecules are then laboratory tested, which is financially demanding. Therefore it is important that an automatic VLS method should show very good precision.

Ligands are molecules that can bind to a receptor, while *decoys* either bind only temporarily or do not bind at all. Based on previous research, a set of already recognized ligands and decoys is available and will be used as a development data set for developing predictors that classify molecules either as ligands or decoys.

Ligands are *active* molecules that are necessary for living organisms. It naturally happens, however, that in time some active molecules lose their effect and should be replaced. Since the human body is a dynamic environment, the active molecules are replaced by slightly different ones with similar but not really identical properties. We can imagine that for some of the new molecules we do not know their actual status, whether they are ligands or not. In the following VLS tasks, new molecules form a test data set.

Data sets provided for the VLS tasks

All data are related to one particular receptor called *AA2AR*. In both development and test data sets each example represents a small molecule and consists of a number of quantitative characteristics, the values of which are specific to each molecule and make a feature vector. The last attribute in the vector (`active`) represents the target value. In the development data sets each example is labeled with a true target value, while the test set is blind, which means without true target values. Development dataset D is divided into two disjoint parts D1 and D2. Development examples in D will be used for building and tuning machine learning models. Finally the model evaluated as the best one will be used for prediction on the given test set T. The size of the provided data sets is shown in the following table.

	Data set	File	Data set size
Development data sets (labeled examples)	D1	`devel1.csv`	6,300 examples
	D2	`devel2.csv`	2,100 examples
	D = D1 ∪ D2		8,400 examples in total
Test data set (blind)	T	`test.blind.csv`	1,722 examples

Why did we split the development examples into two parts and why do we make distinction between them? Technically, the reason is that the test data set T and the development data set D are not two representative samples of an identical population, although they should be considered to be similar. The test data set T corresponds to the "new molecules" that were not recognized yet, and their population differs from the "known" population of D. The difference between D1 and D2 should help to tune the developed models towards the unknown population of T. In fact, the "new molecules" in T were collected by the same technique as the molecules in D2 but later on. Therefore they are more similar to the molecules in D2 than in D1.

Data sets D1 and D2 look very similar, but have slightly different statistical properties. Generally they can be mixed, but in some experiments they should be used separately and differently. The relationship between D2 and D1 is similar to the relationship between T and D.

Brief comment on features

All features available in the given data sets are quantitative characteristics of molecules related either to their structure or to other chemical properties. Just for illustration here is a short description of selected ones:

- `MolWt` – The average molecular weight of the molecule
- `HeavyAtomMolWt` – The average molecular weight of the molecule ignoring hydrogens
- `ExactMolWt` – The exact molecular weight of the molecule
- `NumValenceElectrons` – The number of valence electrons the molecule has
- `NumRadicalElectrons` – The number of radical electrons the molecule has
- `NHOHCount` – The number of NHs or OHs

Technical remarks

Note that all VLS tasks should be read one after another. The code provided in the VLS tasks is usually a follow-up to the earlier ones. Also, all codes are maximally simplified as they aim to be as clear and comprehensible as possible.

To run the code snippets in the following VLS tasks, a few R packages should be installed first:

```
library(rpart)
library(rpart.plot)
library(adabag)
library(randomForest)
library(ROCR)
```

Task 20
Simple data analysis and feature filtering

Exercise 20a Positive/negative proportion

Determine the proportion of the binary target values in data sets D1 and D2. Then look at the available features. Remove all features that have constant values in data set D. Determine the number of remaining discrete and continuous features.

Exercise 20b Feature frequency

For each discrete feature determine the number of different values. Make a plot that shows how many features have a certain number of values.

Exercise 20c Information gain

For each discrete feature A compute its information gain, i.e. the mutual information IG(C,A) between the feature A and the target class C. Then sort all discrete features decreasingly according to information gain and draw a plot.

Task 21
Simple Decision Tree (DT) model for automatic classification

We should keep in mind the purpose of this classification task. Molecules from the test set that are classified as ligands are to be laboratory tested, which is rather costly. This is why our models for classification should prefer high precision. Hence to optimize our classification model we will try to maximize a particular area under the ROC curve rather than to simply minimize error rate. Since precision naturally decreases with increasing FPR, we will measure and optimize the AUC just up to $FPR \leq 10\%$, hereafter denoted by AUC_{10}.

Exercise 21a Cross validation preparation

Prepare your development data set D1 for 10-fold cross validation process. Since positive and negative examples are highly unbalanced in D1, the division into CV folds should be done carefully and identical number of positive examples in all 10 folds should be kept. Each CV run will use 5670 training examples and 630 test examples.

Exercise 21b Simple decision tree

Build a Decision Tree model for binary classification using `rpart` function with the default parameter settings. Use only D1 for training and evaluate the model using 10-fold cross-validation. In each run of the CV process compute AUC_{10}. Then report its mean, standard deviation, and confidence interval.

Exercise 21c Prediction on D2

Train your DT model using the whole D1 set. Then make prediction on D2. Again, compute AUC_{10}. Is it in line with your estimate computed in subtask 21b)?

Exercise 21d Tuning cp

Using the same CV process as in subtask 21b optimize the complexity parameter (cp) to maximize the mean of AUC_{10}. For different cp values report the mean of AUC_{10}, its standard deviation, and confidence interval for the mean (use t-test at significance level $\alpha = 5\%$).

Then choose an "optimal" cp value using the following heuristics. First find the cp value for which the mean of AUC_{10} reaches its maximum, and then choose a maximum cp value for which the mean of AUC_{10} does not decrease more than by 1SE below its maximum. SE stands for standard error, estimated as sample standard deviation divided by square root of the sample size.

Finally, use the tuned cp value and train your DT model again using the whole D1 set. Make prediction on D2 and compare your result with previous experiments.

Exercise 21e Decision tree plot

Visualize and compare the decision trees trained previously and used for prediction on D2.

Task 22
Random Forest models for automatic classification

Exercise 22a Optimizing AUC_{01}

Run the Random Forest learning method, evaluate it and tune its parameters to maximize AUC_{01}. Draw a plot that shows how the performance depends on selected parameters. Choose optimal parameter values.

Exercise 22b Performance and the number of tres in the ensemble

For a fixed value of `mtry` plot the dependency of AUC_{10} on `ntry`.

Exercise 22c Comparison

Draw the ROC curve for the Random Forest algorithm with `ntree=300, mtry=20`. Compare the results with the baseline model obtained in Task 21. Which models are better and why?

Task 23
Adaboost models for automatic classification

Exercise 23a Optimizing AUC_{01}

Run the Adaboost learning method, evaluate it and tune its parameters to maximize AUC_{01}. Evaluate the model, then draw a plot that shows how the performance depends on selected parameters. Choose optimal parameter values.

Exercise 23b Comparison

Compare the results with the baseline model obtained in 21 and 22. Which models are better and why?

Task 24
Logistic regression models for automatic classification

Exercise 24a Optimizing AUC_{01}

Run the Logistic regression learning method, evaluate it and tune its parameters to maximize AUC_{01}.

Exercise 24b Graphical representation

Draw a plot that shows how the performance depends on selected parameters. Choose optimal parameter values.

Exercise 24c Comparison

Compare the results with the baseline model obtained in 21. Which models are better and why?