MACHINE LEARNING FROM DATA

Lab Session 8 – Decision trees and Random Forests

1. Goal

The goal of this session is to

- Learn how to use decision trees to solve a classification problem
- Learn to prune a decision tree
- Learn to train a random forest classifier
- Solve the Ionosphere classification problem with decision trees and random forests
- Solve the MNIST classification problem with random forests.

2. Instructions

- Download and uncompress the file Mlearn_Lab8.zip
- Answer the questions in the document Mlearn_Lab8_report_surname.pdf

3. Previous study

Read the slides corresponding to lecture 4.4: Decision trees.

In the first part of the session, you will play with a toy example, using a synthetic dataset with two classes and two features per class, to learn how to train and visualize a decision tree classifier.

In the second part you will use decision trees and random forest on the Ionosphere dataset, where the task is to classify radar returns from the ionosphere. Finally, you will train a random forest classifier to solve the MNIST classification problem

4. Ionosphere dataset

4.1. Characteristics of the dataset

The dataset can be found here: https://archive.ics.uci.edu/ml/datasets/ionosphere

This is the information provided for the dataset:

```
This radar data was collected by a system in Goose Bay, Labrador. This
system consists of a phased array of 16 high-frequency antennas with a
total transmitted power on the order of 6.4 kilowatts. See the paper
for more details. The targets were free electrons in the ionosphere.
"Good" radar returns are those showing evidence of some type of
structure in the ionosphere. "Bad" returns are those that do not; their
signals pass through the ionosphere.

Received signals were processed using an autocorrelation function whose
arguments are the time of a pulse and the pulse number. There were 17
pulse numbers for the Goose Bay system. Instances in this databse are
described by 2 attributes per pulse number, corresponding to the
complex values returned by the function resulting from the complex
electromagnetic signal.

Attribute Information:

-- All 34 are continuous
```

```
-- The 35th attribute is either "good" or "bad" according to the
definition summarized above. This is a binary classification task
```

4.2. Classification using decision trees

You will first train a decision tree with the default parameters and then you will try to reduce overfitting with two different strategies, controlling the minimum number of samples per leaf, and by pruning the tree.

Run the code and answer the following questions:

Q1: For the tree trained with the default parameters, copy the training, and test classification errors and the confusion matrices.

Q2: Visualize the tree and analyze the questions at each node. Compute the feature's importance. Which are the most relevant features for the classification task?

You will first try to reduce overfitting by controlling the minimum number of samples per leaf.

Edit the notebook.
Add the necessary code to:
- perform a grid search cross validation on hyperparameters: max_leaf_nodes and min_samples_split (use at least the values given in the notebook).
- make predictions on the train and test sets with the best value of the hyperparameter
- print classification report, error and confusion matrices for the train and test sets.
- show the grid search results as a Pandas dataframe.

Q3: Copy and analyze the results (errors, confusion matrices and plots) and compare with the results obtained with the default parameters.

Next, you will try to reduce overfitting by pruning the tree with the minimal cost-complexity algorithm.

Edit the notebook.
Add the necessary code to:
- perform a grid search cross validation with the following values of the parameter ccp_alpha: [0, 0.01, 0.05, 0.1, 0.15, 0.2, 0.25, 0.3].
- make predictions on the train and test sets with the best value of the hyperparameter
- print classification report, error and confusion matrices for the train and test sets.
- show the plots with train and validation errors for all the values of the hyperparameter

Q4: Copy and analyze the results (errors, confusion matrices and plots) and compare with the results obtained with the previous strategy.

4.3. Classification using random forests

Finally, you will train a random forest classifier with the 'entropy' criterion, and min_samples_leaf=5.

Q5: Copy and analyze the results (errors, confusion matrices and plots) and compare with the results obtained with decision trees.

5. MNIST dataset

Now you will have to train a random forest classifier to solve the MNIST classification problem.
First import and load the dataset, then select a subset of 15000 images for training and a subset of 2500 images for test.
Train a Random Forest classifier, selecting some of the hyperparameters by cross-validation. Use the name name 'rnd_clf' for the model
Finally, show classification report and confusion matrices for train and test sets, show and observe the importance of each feature (pixel) (code is given).

Q6: Copy and analyze the results (errors, confusion matrices and plots) and compare with the results obtained in Lab6 using Neural Networks.