Assignment 1: Decision Trees and Nearest Neighbor Classification

In this assignment we will explore the use of decision trees and nearest neighbor classifiers to learn morphological classes. We will make use of standard implementations of these classifiers in Weka (https://www.cs.waikato.ac.nz/ml/weka/), so the first thing to do is to download and install Weka (if you haven't done so already).

We will use two word morphology data sets describing the formation of inflected word forms from lemma information for the plural forms of German nouns (german_plural.arff) and the past tense forms of English verbs (english_past_tense.arff), respectively. The data is in the **arff** format, which is the standard Weka format and more or less self-explanatory. If you want to know more about the format, consult the Weka documentation.

Data Exploration

Your first task is to get acquainted with the data sets. Start Weka, choose the Explorer GUI, and open the German plural data set (under the Preprocess tab). You can now see that the data set contains 25168 instances, each of which is a German noun lemma, and that each instance has 9 attributes, of which the **class** attributes encode the plural formation of the noun and is the attribute that we want to learn to predict. The attribute **frequency** records corpus frequency, the attribute **gender** represents the gender of the noun, and the attributes **p1-p6** give a phonological representation of the last two syllables of the base form. Specifically, p1, p2 and p3 represent the onset, nucleus and coda of the penultimate syllable, and p4, p5 and p6 those of the ultimate syllable. By selecting different attributes, you can inspect their type, value set and distribution in the data set.

Next open the English past tense data set, which consists of 4330 verb lemmas and where the class to predict is the past tense formation rule. Attributes are similar to the ones found in the German data set, except that the phonological representation covers the last three syllables (p1-p3: antepenultimate, p4-p6: penultimate, p7-p9: ultimate) and that there is no gender attribute.

Once you understand how the data sets are composed, you should analyze the informativeness of the different features using **information gain**. To do this, choose Select attributes in Weka and then choose InfoGainAttributeEval as the Attribute Evaluator (which also requires you to choose Ranker as the Search Method). This will give you a ranking of the features in terms of information gain.

Report: Which features are most informative for the two data sets? Try to explain why some features are more informative than others.

Decision Trees

The second task is to induce decision trees for predicting the plural form of a German noun and the past tense form of an English verb. Open the Classify tab and choose **J48** in the folder **trees** as the classifier. (J48 is Weka's implementation of the C4.5 algorithm.) Build decision trees for both data sets and analyze their performance.

Compare **training error** (Test option: Training set) to **test error** (Test option: Cross-validation) and see whether there are signs of overfitting. Also compare tree induction with and without pruning (click on options next to the Classifier choice to switch the parameter **unpruned** from false to true) and see how this affects the size of the tree as well as the relation between training and test error.

One of the advantages of decision trees, compared to many other learning algorithms, is that the induced classifier can be interpreted as a set of rules for classifying new instances. What rules can you find in the trees you have induced? Do they make sense?

Report:

1. How accurate are the decision tree classifiers for the two data sets? Look at overall accuracy as well as precision and recall for specific classes.

2. How does training error relate to test error? What is the effect of pruning?

3. Can you make sense of the rules implicit in the trees? Consider especially the pruned tree for the English past tense data.

K-Nearest Neighbor

The third task is to use k-nearest neighbor classification to predict the plural form of a German noun and the past tense form of an English verb. Open the Classify tab and choose **IBk** in the folder **lazy** as the classifier. Apply the classifier to both data sets and analyze its performance.

Compare **training error** (Test option: Training set) to **test error** (Test option: Cross-validation). Vary the number of neighbors used to predict the class (click on options next to the Classifier choice to change the value of the parameter **KNN**) and see how this affects training and test error.

One of the properties of (simple) nearest neighbor classification is that all features are given equal weight, which means that irrelevant features could hurt classification accuracy. Check whether you can improve accuracy by removing features. Compare the best accuracy to that obtained with decision trees.

Report:

1. How accurate are the nearest neighbor classifiers for the two data sets? Look at overall accuracy as well as precision and recall for specific classes.

2. What is the effect of varying the k parameter? Can you improve accuracy by removing less informative features?

3. Does k-nearest neighbor perform better or worse than decision trees? Can you force the nearest neighbor classifier to behave like the pruned decision tree on the English past tense data?

VG assignment

The above tasks are sufficient to obtain a pass grade (G) in this assignment. To obtain a pass with distinction (VG), all the assignments must be carried out without major errors, and there is an additional task you should solve.

It is fairly obvious that the performance of a machine learning task depends on the amount of training data that is available, but the amount of data required to reach a certain level of performance in a specific task varies. Your last task is to relate the size of the training data to the performance you can achieve on the two data sets.

Train your classifiers using the best hyperparameters you found. for both prediction tasks on a varying amount of training data ranging from just a few examples to the full data set and test the resulting classifiers for each training set size. Do this for both types of classifiers (decision trees and kNN). Check if you can find a set of hyperparameters that work better for smaller training set size and compare the learning curves. Take a look at the rules in the decision trees for various training set sizes.

Report: Describe how you ran these experiments and present the resulting learning curves, both in tabular and in graphical form. Report and discuss any interesting observations you made.

Submission

You should submit a written report (3-5 pages) reporting your results on all three tasks. The report should be submitted through Studentportalen by the 13th April.