Machine learning: Exercises for September 14, 2017

Henning Christiansen
Roskilde University
http://www.ruc.dk/~henning/, henning@ruc.dk

1 Transforming data to obtain better features for learning

Sometimes the raw dataset given for training does not fit a given model class, and it may be the case that a transformation of data may help.

We may construct new attributes, identified as features, from the given attributes.[1] We will use the linear regression program from last week as our "learning engine" and consider how we can transform data into a more useful form.

There are statistical techniques aimed at identifying the most important features or attributes in case of perhaps very long data tuples. However, these methods are not of interest in the present course, and the purpose of this exercises is to get familiar with the idea of transforming data to obtain the right features.

1.1 Identify those features that matter, and those that are uninteresting

The file `featureFix1.csv` is a data file, which has three columns, we may call them x_1, x_2 and x_3, and your task is to argue which two columns that are best for classification tasks (and which one you may throw away as unimportant).

More precisely, given a dataset (here: of triples=3-tuples) that is representative of some population $\mathscr{P}$, you must produce a model that measure of how will a new and unseen triple can be seen as belong to $\mathscr{P}$.

You should argue in terms of average error for the possible models, and support the conclusion by graphical plots of the relevant models.

1.2 Combing two attributes into one

The file `featureFix2.csv` is a data file which has three columns, x_1, x_2 and x_3. The data are constructed such that a sum of two the columns are suited as a feature to include in a linear model

[1] The word "feature" is often used as synonymous with our use of "attribute", so you may also talk about constructing new features from existing ones.

(correlated with the remaining column). As above, your task is to argue which two colunms that should be added together (element-wise!) in order to obtain the best classification.[2] You would need to use a spreadsheet program such as Excel or similar; remember to save the file in the `csv` format, and be aware that Excel in some cases may replace the commas by semicolons, so that you have to repair the file using a flat text editor.

You should argue in terms of average error for the possible models, and support the conclusion by graphical plots of the relevant models.

2 Automatic detection of painters

The purpose of this exercise is to try out supervised learning for recognizing different painter's works. We should use the HIC system which you may find at `http://ruc.dk/~henning/HIC`. On moodle you can find a collection of digitized paintings by Monet, Picasso and van Gogh that should be used for training and validation.[3]

HIC is based on learning of ensembles of a special kind of decision trees (the lecture will give a short introduction), described in the paper

> P. Geurts, D. Ernst, and L. Wehenkel. Extremely randomized trees. *Machine Learning*, 63(1):3–42, 2006.

The mentioned website includes the source code for HIC written in Processing and an easy to read User's Guide.

2.1 Get familiar with HIC

Download the system, get it to run and train a model on the included test images to distinguish between *busses* and *dinos*. Use the Learn and Validate buttons as described in the User's Guide and try to interpret the result of Validate. Try also the other commands in the HIC window. – There is not much to conclude here, but when you have done this, you should know how to use the system.

2.2 Learn

Now to the interesting part. Download the digitized paintings by Monet, Picasso and van Gogh. Notice that the labelling of files are implictly given by the subfolder names, as HIC expects it.

Divide it into training and validation data and test (cf. the course note "A gentle introduction ... '') and see how good results you can obtain.[4]

[2] Your teacher is aware that this is a highly artificial set-up, but it shows a point.

[3] Warning: The images have been harvested by web search without caring about copyright, so you should not distribute them further!!

[4] Warning: Training may take some time, especially if your computer is a bit old.

There is an element of randomization in HIC's algorithms, so you may try to run the same experiment twice to check whether this results in any significant chance. If you have time, try different splittings of the data sample into training and validation part to see if it makes some difference..

2.3 Can you improve the result by changing the learning parameters?

The source code window `constantDefinition` has a number of constants that determines how HIC learns.[5] Some of them likely gives not much sense to you so ignore those, but try to see if you can improve the performance of the learned model.

[5] Such constants are normally called "learning parameters", which is something very different from the (model) parameters we have considered so far.