Pattern Recognition - 8th lab
Validation and one-hot encoding

Viktor Kocur
viktor.kocur@fmph.uniba.sk

DAI FMFI UK

13.4.2020

Splitting the data

Training set

So far we have only worked with the training set. In other words we used all of our data to set the model parameters.

Test set

If we want to show that our model is reliable it is necessary to save some data for testing. The testing data are used at the very end when our model has been fit. We use them just for evaluation and not for selecting the method, its parameters or hyperparameters.

Data split

Validation set

Since the test set is not used to choose the method etc. we need to separate one more set for this purpose. The validation set is used to choose the optimal method and its hyperparameters.

Data split

Data splits are based on the amount and type of data as well as the selected model. With neural nets we need a lot of training data therefore common splits are 80/10/10. With other methods 60/20/20 is sufficient. Some datasets have a predetermined split of 40/20/40.

Validation

Hyperparameters

We select the hyperparamters on the validation set. These are paramters/settings which change how the model is trained and how the prediction works. For SVM this can be the choice of kernel function and its scale. For kNN it is for example the k value or the selected metric.

Validation

We train our model with various parameters (we just build the model in case of kNN) on the training set. We test these models on the validation set. We select some metric of reliability such as classification accuracy. Based on the results we select the hyperparameters.

Validation - exercise

Exercise

Split the data from the previous lab to train/val/test with 60/20/20 split. Choose the best k parameters for the kNN classifier and the best metric.

Sound representation

Sometimes the data is ordered by class or in some other regular form. It is therefore necessary whether the split is meaningful. Ideally we want the same amount examples for each class in the set.

Cross-validation

Cross-validation

If we do not have enough data we do not split the data into training and validation sets. We split the data into n approximately similar subsets. We then train the model on all but one of the subsets and use the remaining set as the validation set. We repeat this process n times and select our hyperparameters based on their average.

Matlab

```
Mdl = fitcknn(X, y, 'NumNeighbors', k);
CVMdl = crossval(Mdl)
loss = kfoldLoss(CVMdl)
```

Automating validation

Automatic hyperparameter selection

Matlab is able to find the optimatl hyperparameters for most of the fitc.. functions. If you choose to use this make sure you check the help so you know what your selected hyperparameters mean.

Matlab

```
Mdl = fitcknn(X,Y,'OptimizeHyperparameters','auto')
```

Categorical data

Categorical data

Sometimes we get data in a categorical form: one of the features comes from a finite set of possibilities. For example yes/no, or student/working/unemployed/retired. Some methods we have used cannot use these kinds of data. Specifically SVM and Linear classifier cannot. kNN in Matlab is able to work with categorical data using a special metric, but it is necessary to set it up.

Problem with simple conversion

Why not do a sipmle conversion

There is a simple way to convert the data by giving each categorical value a number. This is not ideal. One of the reasons is that for example if we have categories pedestrian:0, bicycle:1, motorcycle:2, car:3, van:4, truck:5. Then an average of a truck and motorcycle is a car, which is not meaningful. However we can sometimes use this approach for data such as grades: A, B, C, D, E, Fx, where we expect an average of B and D to be a C.

One-hot encoding

One-hot encoding

To avoid the probles of the previous slide we can use so-called one-hot encoding. We convert every categorical feature with m different categories into a feature vector of size m, so that every element of the vector corresponds to one category. We set this element to 1 and all of the rest to 0. So for example if we had data with speeds and categories of vehicles (20, bicycle), (58, car) we would obtain vectors (20, 0, 1, 0, 0, 0, 0) and (58, 0, 0, 0, 1, 0, 0).

One-hot endoding Matlab

dummyvar

d = dummyvar(c) - returns one-hot encoding for a categorical column vector c

categorical

c = categorical(r) - returns a categorical type for a vector r. We use it mostly to convert from cell type to categorical.

categories

cats = categories(c) - returns the categories from the categorical vector c.

One-hot kódovanie Matlab

load patients

Load the data of cardiology patients using load patients.

Exercise

Convert categorical features to one-hot encoding and train an SVM. Do not use the name and the hospital information. The goal for prediction is whether the patient is a smoker.