Machine Learning I: Foundations
Exercise Sheet 7

Prof. Marius Kloft TA:Billy Joe Franks

23.06.2021
Deadline: 22.06.2021

1) (MANDATORY) 10 Points

In this question we investigate what influences models to overfit or underfit. Please list every ML model we have investigated so far in ML1. For each model list what influences the models power, i.e. what would have to be changed in the model for it to potentially overfit, underfit, or fit well. Try to list each possible change for each model (most models have methods to change their representative power without changing their regularization).

- **K-nearest neighbor.** The obvious choice is the parameter k. Increasing k increases underfitting, while decreasing k increases overfitting and the models power. Another non-obvious choice, which is especially important to name for KNN is the size of the dataset. The size of a dataset in fact influences almost every ML model. However, for KNN its power is proportional to the amount of data provided. With more data KNN can describe more complex functions. The dataset size will not be mentioned from this point onward, even though it does typically influence overfitting and underfitting, it does not influence a models power for most ML models.
- **Nearest centroid classifier.** It might not be obvious, but the dimensions of the dataset influence the NCCs power. Just as we learned later on kernelization can turn linear models into non-linear ones. NCC can be kernelized, by rerepresenting its distance calculation. Some kernels have themselves hyperparameters that influence the models power. For instance the gaussian kernel has its width parameter σ^2, which influence the models power directly. This kind of hyperparameter, i.e. kernelization or explicit feature mapping, is again one that actually holds for most ML models, even though it is typically only mentioned for models whose predictive function is strongly limited like linear classifiers.
- **Hard-margin SVM.** It has the same property as the NCC.
- **Soft-margin SVM.** In addition to the hard-margin SVM, this classifier has a hyperparameter C which influences overfitting and underfitting. However, C does not influence the models power, i.e. changing C does not influence the type of function that can be represented, unless we are considering certain kernelization, then C can directly influence the functions, however this is outside the scope of this small exercise.
- **ML models optimized by SGD.** Generally SGD reduces overfitting, however it does not typically influence a models power.
- **Neural Networks.** The architecture of a neural network generally influences its power. The "bigger" it is, i.e. the more layers and neurons per layer it has, the higher the models power. The activation function as well as specific layer choices, for instance a batch-norm layer, might also influence the models power, beyond just improving the models trainability.
- **Regularization.** Regularization can be added to most ML models, however, regularization will only influence a models power, if the models function space contains functions with differing powers. For instance linear classifiers will not change their power if regularized, they will change their generalization. NNs on the other hand will change their function space based on regularization and will be less powerful for stronger regularization.

2) (MANDATORY) 10 Points

Consider you are given an ML model together with hyperparameters for adjusting the models power, i.e. you have options to change whether the model overfits, underfits, or fits well. There might be an infinite amount of possibilities for setting the hyperparameters, but you may assume that each hyperparameter can only be set to a rational number. You may also assume each hyperparameter influences the models power continuously, i.e. if the hyperparameter increases the power increases and vice versa. This exercise is not about finding the optimal procedure, which is an active research question. This exercise is about your approach and you will be graded on how good your approach is given the information provided in the lecture.

a) Describe a methodology for using these adjustments to determine the appropriate power for a given problem, be exact with your methodology. Your methodology should set the hyperparameters somehow.

The simplest approach is using cross-validation. We try out different hyperparameters and choose the best among them according to CV. However, since there might be an infinite amount of hyperparameters how do we choose a finite subset of these? One typical approach in practice is gridsearch. For each hyperparameter some lowest and highest value is choosen somehow, we will leave this as a hyperparameter, and then we search in the resulting cube of the remaining hyperparameters according to some grid, which is another hyperparameter. Then we simply try out all possibilities and choose the hyperparameters that fit best.

b) Now consider your methodology as the entire training procedure resulting in a model. Does your model have hyperparameters?

As mentioned above the gridsearch has hyperparameters, in addition CV itself also has hyperparameters, like the number of splits.

c) When might your model overfit or underfit, i.e. how can you determine your models power?

First of all the choice of the domain of the gridsearch is directly choosing the original hyperparameters, which could change the power. Next, the total amount of hyperparameter choices searched might influence the models power, since each choice could represent a different function. Thus the density of the gridsearch influences the models power. Lastly, the number of splits in CV influences underfitting and overfitting of the model, even though it does not influence its power.

d) How do your models hyperparameters influence computation times?

In this simple example the number of evaluations is directly proportional to the total computation time. The number of evaluations is proportinal to the number of points in the grid, as well as, the number of splits of CV. And even worse, these are influenced multiplicatively. Consider we split up our cube in hyperparameter i by s_i and that we have n hyperparameters and we are considering t-times k-fold CV. Then the total computation time is proportinal to

$$tk \prod_{i=1}^{n} s_i.$$

3) Solve programming task 7.