HyperParameter Tuning

Hyper Parameters & Tuning

1. Hyper parameters are like handles available to the modeler to control the behavior of the algorithm used for modeling

2. Hyper parameters are supplied as arguments to the model algorithms while initializing them. For e.g. setting the criterion for decision tree building
   "dt_model = DecisionTreeClassifier(criterion = 'entropy')"

3. To get a list of hyper parameters for a given algorithm, call the function get_params()…for e.g. to get support vector classifier hyper parameters
   1. from sklearn.svm import SVC
   2. svc= SVC()
   3. svc.get_params()

4. Hyper parameters are not learnt from the data as other model parameters are. For e.g. attribute coefficients in a linear model are learnt from data while cost of error is input as hyper parameter.

Hyper Parameters & Tuning

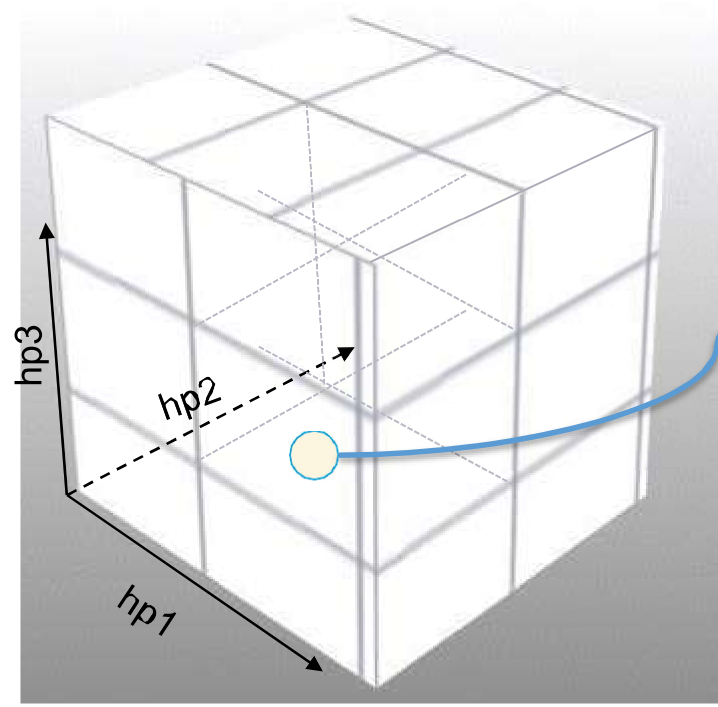
5. Fine tuning the hyper parameters is done in a sequence of steps
    1. Selecting the appropriate model type (regressor or classifier such as sklearn.svm.SVC())
    2. Identify the corresponding parameter space
    3. Decide the method for searching or sampling parameterspace;
    4. Decide the cross-validation scheme to ensure model will generalize
    5. Decide a score function to use to evaluate the model

6. Two generic approaches to searching hyper parameter space include
    1. GridSearchCV which exhaustively considers all parameter combinations
    2. RandomizedSearchCV can sample a given number of candidates from a parameter space with a specified distribution.
7. While tuning hyper parameters, the data should have been split into three parts – Training, validation and testing to **prevent data leak**

8. The testing data should be separately transformed * using the same functions that were used to transform the rest of the data for model building and hyper parameter tuning

* Any transformation where rows influence each other. For e.g. using zscore. OneHotCode transformation does not come into this category. It can be done before splitting the data

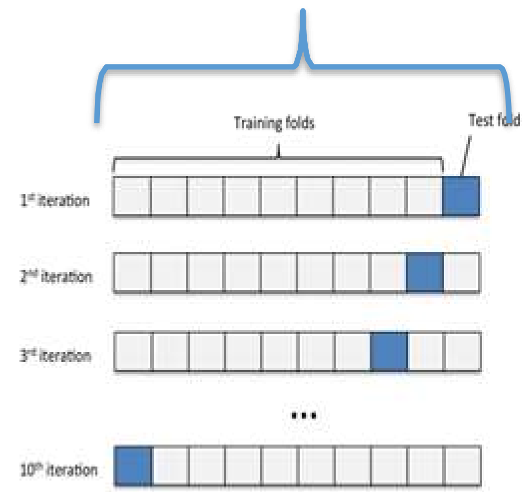
GridSearchCV



Hyper parameter space

One combination of hyper parameters used K times to train and test. The avg score of the K times is the score associated with this combination

This will repeat for all possible combinations i.e. all the cells in the space.



Hyper Parameters & Tuning (GridsearchCV/ RandomizedSearchCv)

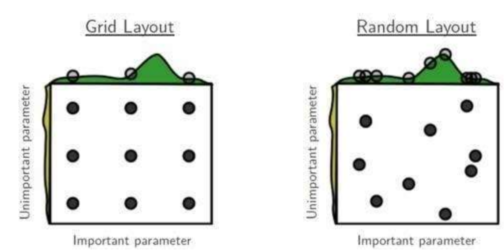
RandomizedSearchCV –

1. Random search differs from grid search. Instead of providing a discrete set of values to explore on each hyperparameter (parameter grid), we provide a statistical distribution.

2. Values for the different hyper parameters are picked up at random from this combine distribution

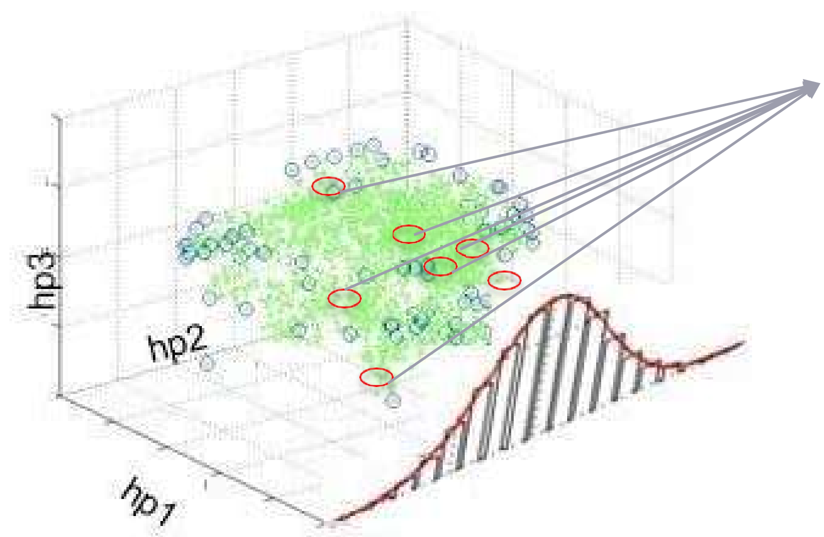
3. The motivation to use random search in place of grid search is that for many cases, hyperparameters are not *equally* important.

*A Gaussian process analysis of the function from hyper-parameters to validation set performance reveals that **for most data sets only a few of the hyper-parameters really matter, but that different hyper-parameters are important on different data sets**. This phenomenon makes grid search a poor choice for configuring algorithms for new data sets. - [Bergstra, 2012]*



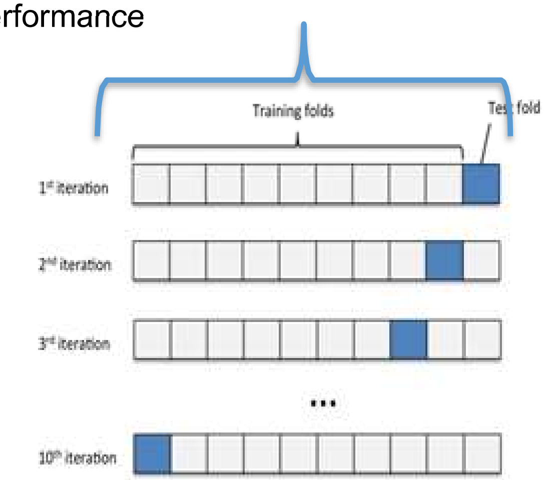
Picture by [Bergstra, 2012]

RandomizedSearchCV



Hyper parameter space

Randomly pick up n-iter samples from the hyper parameter distribution as sample, Use it K times and find avg performance



4. In contrast to GridSearchCV, not all combinations are evaluated. A fixed number of parameter settings is sampled from the specified distributions.
5. The number of parameter settings that are tried is given by n_iter
6. <u>If all parameters are presented as a list, sampling without replacement is performed</u>. <u>If at least one parameter is given as a distribution, sampling with replacement is used</u>. It is highly recommended to use continuous distributions for continuous parameters
7. Randomsearch has higher chance of hitting the right combination than gridsearch.