

1 Programming

1.1 Best performance of Each Model

All these are measured without shuffling the data-set. If shuffle is turned on the results may vary.

ERM model for perceptron

```
python perceptron.py -m erm -d linearly-separable-dataset.csv -b 0.7
```

For best results use a bias factor of **b=0.7** to get 0 erm error in 2 iterations

iterations: 2 bias and weights: -2.8 [-11.3784905 5.65714419] empirical_error: 0.0

kfold model for perceptron

```
python perceptron.py -m kfold -d linearly-separable-dataset.csv -k 10 -b 0.8
```

For best results use **b=0.8** and **k=10** to get 0 mean error over all folds

ERM model for adaboost

```
python adaboost.py -d Breast_cancer_data.csv -m erm -t 95
```

For the least erm error in adaboost use 95 boosters above which the error wont decrease anymore for the given data-set.

kfold model for adaboost

```
python adaboost.py -d Breast_cancer_data.csv -m kfold -t 96
```

Mean validation error over 10 folds = 0.0703 and mean erm error of 0.041. The mean erm error will not go down significantly even if I use 150 boosters.

1.2 Observations

For the run-time of perceptron algorithm I can observe that the bias factor which is introduced plays a significant role in run-time of the algorithm. Instead of updating the bias with ± 1 we can play around this and as shown using a bias factor of 0.7 within 2 iterations the algorithm classifies as data points.

The perceptron algorithm will not terminate if the data is not linearly separable and will run in an infinite loop. To counter this I have hard-coded the maximum number of iterations to 15000 after which the program will exit. Also an error threshold is also introduced, if the error in the current iteration goes below the given threshold the algorithm will stop.

The perceptron algorithm did not terminate as the algorithm started to repeat the same weights and all points could not be classified correctly.

1.3 Plot findings

From the figure 1 we can clearly see that the erm error and the validation error both decreases as we increase the number of boosters. Also the validation error starts to increase once we cross 50 boosters which tells that we have start to over fit the data. So based on the error plot we can do **50 rounds** of boosting for the given dataset.

Execute `python adaboost.py -d path-to-data -m kfold -t 200 -p` to get the plot

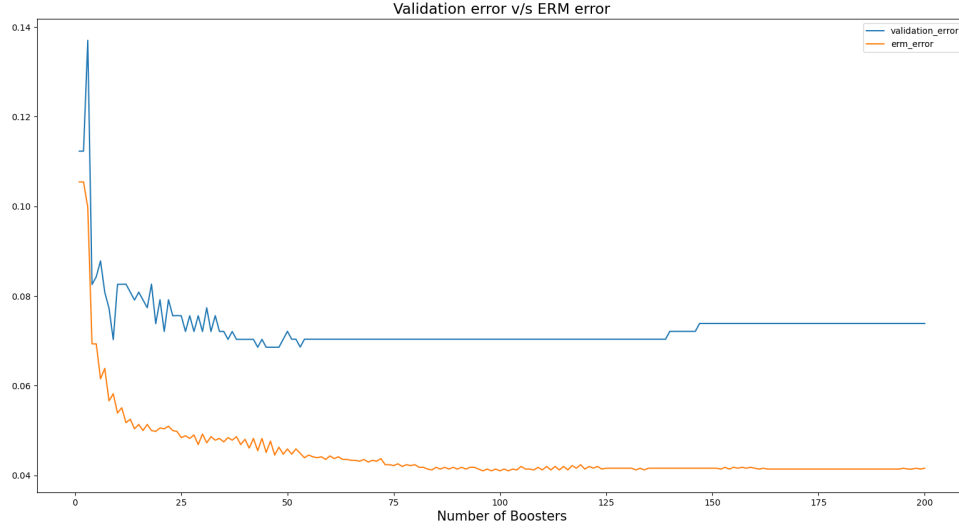


Figure 1: ERM error v/s validation error in Adaboost

2 README

2.1 Perceptron-Algorithm

perceptron.py supports the below parameters:

Option	Description	Default Value
-h, -help	show this help message and exit	
-d, -dataset	<i>path_to_dataset</i>	linearly-separable-dataset.csv
-m, -mode	Mode can either be erm/kfold	erm
-e, -epsilon	The ERM error below which the program will terminate	0.0
-b, -bias_factor	Factor which decides the rate at which bias factor will change	1.0
-k, -kfold	The number of folds. used if mode=kfold	10
-s, -shuffle	Shuffle the data set randomly	False

To run the perceptron algorithm in erm mode, execute

python perceptron.py -d path_to_data -m erm -e 0 -b 1.0

To run perceptron algorithm in kfold mode, execute

python perceptron.py -d path_to_data -m kfold -k 10 -e 0 -b 1.0

Note to shuffle the data-set add **-shuffle** or **-s** flag while running the command

2.2 AdaBoost

adaboost.py supports the below parameters:

Option	Description	Default Value
-h, -help	show this help message and exit	
-d, -dataset	<i>path_to_dataset</i>	Breast_cancer_data.csv
-m, -mode	Mode can either be erm/kfold	erm
-t, -trounds	Number of boosters	10
-k, -kfold	The number of folds. used if mode=kfold	10
-p, -plot	Plot the graph of validation v/s erm error. Only used if mode=kfold	False
-s, -shuffle	Shuffle the data set randomly	False

To run the adaboost algorithm in erm mode, execute

python adaboost.py -d path_to_data -m erm -t 10

To run adaboost algorithm in kfold mode with validation error v/s erm plot, execute **python adaboost.py -d path_to_data -m kfold -t 10 -k 10 -p**

Note to shuffle the data-set add **-shuffle** or **-s** flag while running the command.
Also if plot is not needed skip giving **-p** while running the command