Assignment III
Boosting and Gaussian Processes

Adam Kosiorek

I. OBJECTIVE

THE aim of the third assignment was to get insight into workings of Boosting and Gaussian Process (GP) algorithms by comparing performance and estimating uncertainty of several variants of the algorithms.

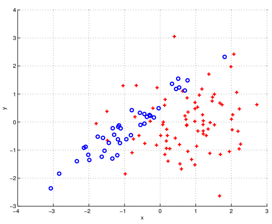
II. DATASET

Boosting algorithms were compared on the MNIST dataset is comprised of 60000 training and 10000 testing samples [2]. Each sample is a 28x28 pixel black and white image of a single centered digit. It is widely used as a benchmark for comparing machine learning algorithms.

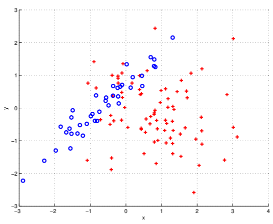
GPs were evaluated on an artificial 2-class dataset. It was randomly generated from a gaussian distribution with expected values and covariance matrices as denoted by $\bar{x}$ and Σ in equations 1 and 2.

$$\bar{x}_{positive} = \begin{pmatrix} 0.75 \\ 0 \end{pmatrix} \quad \bar{x}_{negative} = \begin{pmatrix} -0.75 \\ 0 \end{pmatrix} \tag{1}$$

$$\Sigma_{positive} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad \Sigma_{negative} = \begin{pmatrix} 1 & 0.95 \\ 0.95 & 1 \end{pmatrix} \tag{2}$$



(a)



(b)

Fig. 1: (a) trainset and (b) testset for GPs. **+** denotes positive samples and **o** denotes negative samples.

III. METHODS

Boosting is a means of combining weak classifiers, that is, classifiers only slightly better than random classifiers, in order to create a strong classifier. It does so by sequentially minimizing an exponential loss function by adding parameterized basis functions to the growing ensemble of weak classifiers. A basis function is any weak classifier capable of handling weighted data, *e.g.* decision trees or decision stumps.

Gaussian Process is a collection of random variables, any finite number of which have a joint Gaussian distribution. Since the number of variables can be inifnite, a Gaussian Distribution is a distribution over functions. Therefore, if we have a mean function and covariance function, we can sample from the Gaussian Process. The inifite number of variables can be handled due to marginalization property, which allows operation on finite vectors only.

IV. EXPERIMENTS AND RESULTS

All experiments were carried out on raw data (without any preprocessing) on a notebook with Intel i7-2670QM quad core CPU and 8Gb RAM in Matlab using Matlab Boosting Framework [4] and Gaussian Processes for Machine Learning (GPML) toolboxes [3]. Assignment code is available on GitHub [1].

A. Boosting

I compare AdaBoost, RUSBoost, LPBoost and TotalBoost algorithms trained and tested on small part of the dataset (1000 training and testing samples) and full dataset with different ensemble sizes. The ensemble size is a hyperparameter for the first two algorithms and it is automatically chosen for the last two, which are self-terminating. LPBoost terminated with 128 and 8 classifiers, while TotalBoost terminated with 28 and 51 classifiers when trained on 1000 samples and on the whole dataset respectively. AdaBoost significantly outperforms all other algorithms. According to [4] TotalBoost and LPBoost are ill-suited for problems with large number of observations, which is the case here, while RUSBoost is especially effective for unbalanced data classification problems. Obviously, increased size of the training set positively impact classification accuracy.

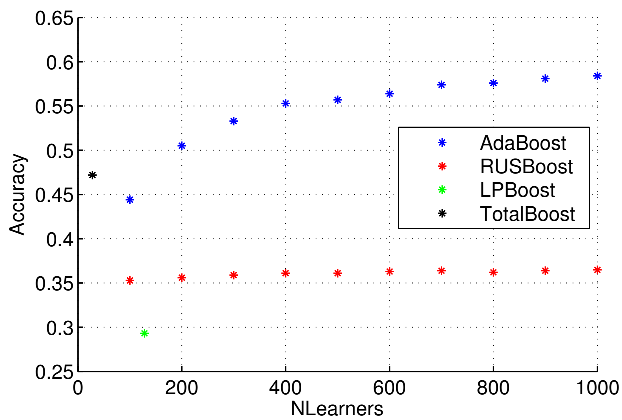


Fig. 2: Classification accuracy of Boosting classifiers trained and tested on 1000 samples.

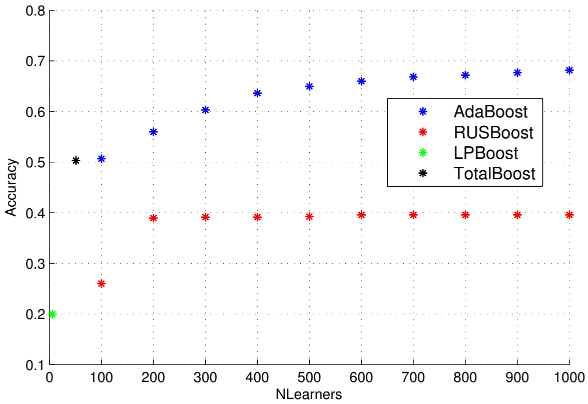


Fig. 3: Classification accuracy of Boosting classifiers trained and tested on the whole dataset.

B. Gaussian Processes

GPs output log probabilties instead of class labels. To compute accuracy and uncertainty I have taken the exponent of the log probability and thresholded it with a value of 0.5, taking class label $l = 1$ for probabilities greater than the threshold value and $l = -1$ for probabilities smaller than the threshold value. Table I shows accuracy of GPs with different covariance, likelihood and inference functions while figures 4 and 5 depict uncertainty histogram for one GP variants with the highest accuracy, namely with linear isotropic (LINiso) covariance function.

TABLE I: GP accuracy

covariance	likelihood	inference	accuracy
SEard	Erf	EP	0.62
SEiso	Erf	EP	0.63
LINard	Erf	EP	0.68
LINard	Gaus	Exact	0.55
LINard	Logistic	EP	0.68
LINiso	Erf	EP	0.68

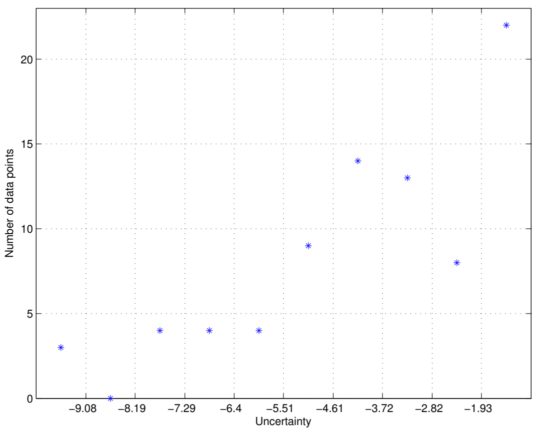


Fig. 4: GP classification uncertainty histogram for correctly classified test samples.

I believe that uncertainty value should be a positive real number between 0 and 1. According to the formula given in the

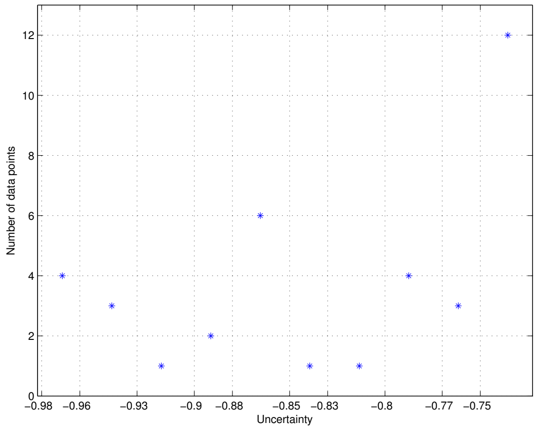


Fig. 5: GP classification uncertainty histogram for incorrectly classified test samples.

assignment, however, I obtained negative values. Also, since

C. Discussion

References

[1] Assignment Code: https://github.com/akosiorek/CSE/tree/master/MLCV/ex3/
[2] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. "Gradient-based learning applied to document recognition." Proceedings of the IEEE, 86(11):2278-2324, November 1998.
[3] C. E. Rasmussen and H. Nickisch, "Gaussian processes for machine learning (GPML) toolbox", The Journal of Machine Learning Research, 2010, 11, 3011-3015.
[4] Matlab Boosting Framework: http://www.mathworks.com/help/stats/ensemble-methods.html