Scale-space based Weak Regressors for Boosting

Jin-Hyeong Park[1] and Chandan K. Reddy[2]

[1] Integrated Data Systems Department, Siemens Corporate Research, Princeton, NJ-08540, USA. E-mail: jin-hyeong.park@siemens.com
[2] Department of Computer Science, Wayne State University, Detroit, MI-48202, USA. E-mail: reddy@cs.wayne.edu

Abstract. Boosting is a simple yet powerful modeling technique that is used in many machine learning and data mining related applications. In this paper, we propose a novel scale-space based boosting framework which applies scale-space theory for choosing the optimal regressors during the various iterations of the boosting algorithm. In other words, the data is considered at different resolutions for each iteration in the boosting algorithm. Our framework chooses the weak regressors for the boosting algorithm that can best fit the current resolution and as the iterations progress, the resolution of the data is increased. The amount of increase in the resolution follows from the wavelet decomposition methods. For regression modeling, we use logitboost update equations based on first derivative of the loss function. We clearly manifest the advantages of using this scale-space based framework for regression problems and show results on different real-world regression datasets.

1 Introduction

In statistical machine learning, boosting techniques have been proven to be effective for not only improving the classification/regression accuracies but also in reducing the bias and variance of the estimated classifier. The most popular variant of boosting, namely the AdaBoost (Adaptive Boosting) in combination with trees has been described as the "best off-the-shelf classifier in the world" [1]. In simple terms, boosting algorithms build multiple models from a dataset, using some learning algorithm that need not be a strong learner. Boosting algorithms are generally viewed as functional gradient descent schemes and obtain the optimal updates based on the global minimum of the error function [2]. In spite of its great success, boosting algorithms still suffer from a few open-ended problems such as the choice of the parameters for the weak regressor.

In this paper, we propose a novel boosting framework for regression problems using the concepts of scale-space theory. In the scale-space based approach to boosting, the weak regressors are determined by analyzing the data over a range of scales (or resolutions). Our algorithm provides the flexibility of choosing the weak regressor dynamically compared to static weak regressor with certain pre-specified parameters. For every iteration during the boosting process, the resolution is either maintained or doubled and a weak regressor is used for fitting

the data. This method of manipulating different resolutions and modeling them accurately looks similar to wavelet decomposition methods for multi-resolution signal analysis. Throughout this paper, we used a Gaussian kernel as an approximate (or weak) regressor for every iteration during boosting. The data is modeled at multiple resolutions and the final boosted (or additive) model will combine the weak models obtained at various resolutions. In this way, we propose a hierarchical (or scale-space) approach for modeling the data using Gaussian kernels. This approach is similar to decomposing a signal using wavelets. Basically, the low frequency components in wavelet decomposition correspond to fitting a Gaussian for the entire dataset and the high frequency components correspond to fitting fewer data points. We formulate this scale-space based boosting regressor using logitboost with exponential L_2 norm loss function. Though our method can be potentially applied with any base regressor, we chose to have Gaussian model because of its nice theoretical scale-space properties [3].

The rest of this paper is organized as follows: Section 2 gives some relevant background on various boosting techniques. It also gives the problem formulation in detail and discusses the concepts necessary to comprehend our algorithm. Section 3 describes our scale-space based boosting algorithm for regression problems. Section 4 gives the experimental results of our algorithm on different real-world datasets. Finally, Section 5 concludes our discussion with future research directions.

2 Background

Ensemble learning is one of the fundamental data mining operations that has become popular in recent years. As opposed to other popular ensemble learning techniques like bagging [4], boosting methods reduce the bias and the variance simultaneously. A comprehensive study on boosting algorithms and their theoretical properties are given in [5]. One main advantage of boosting methods is that the weak learner can be a black-box which can deliver only the result in terms of accuracy and can potentially be any model [2]. The additive model provides a reasonable flexibility in choosing the optimal weak learners for a desired task. Various extensions for the original adaboost algorithm had also been proposed in the literature [6–8]. A detailed study on L_2 norm based classification and regression is given in [9].

In this paper, we propose a novel scale-space based scheme for choosing optimal weak regressors during the iterations in boosting regression problems. The scale-space concept allows for effective modeling of the dataset at a given resolution. The theory of scale-space for discrete signals was first discussed in [10]. Data clustering is one of the most successful applications of the scale-space based techniques [11]. Gaussian kernels have been extensively studied in this scale-space framework [3]. The scale-space based weak regressors will allow systematic hierarchical modeling of the regression function. They also provide more flexibility and can avoid over-fitting problem by allowing the user to stop modeling after a certain resolution.

2.1 Problem Specification

Let us consider N i.i.d. training samples with d features $\mathcal{D} = (\mathcal{X}, \mathcal{Y})$ consisting of samples $(\mathcal{X}, \mathcal{Y}) = (x_1, y_1), (x_2, y_2), .., (x_N, y_N)$ where $\mathcal{X} \in \mathbb{R}^{N \times d}$ and $\mathcal{Y} \in \mathbb{R}^{N \times 1}$. Let us denote $x_i \in \mathbb{R}^{N \times d}$ as i^{th} data point in the d-dimensional feature space. For the case of binary classification problems, we have $y_i \in \{-1, +1\}$ and for regression problems, y_i takes any arbitrary real value. In other words, the univariate response $\mathcal{Y}$ is continuous for regression problems and discrete for classification problems. The goal of a regression problem is to obtain the function $F(\mathcal{X})$ that can approximate $\mathcal{Y}$.

The basic idea of boosting is to repeatedly apply the weak learner to modified versions of the data, thereby producing a sequence of weak regressors $f^{(t)}(x)$ for $t = 1, 2, .., T$ where T denotes predefined number of iterations. Each boosting iteration performs the following three steps: (1) Computes response and weights for every data point. (2) Fits a weak learner to the weighted training samples and (3) Computes the error and updates the final model. In this way, the final model obtained by boosting algorithm is a linear combination of several weak learning models. It was also proved that boosting algorithms are stage-wise estimation procedures for fitting an additive logistic regression model [5]. We derive the scale-space boosting algorithm based on this spirit.

2.2 Boosting for Regression

In the case of regression problems, the penalty function is given by:

$$L(y_i, F^{(t)}(x_i)) = \|y_i - F^{(t)}(x_i)\|_p \tag{1}$$

where $\| \cdot \|_p$ indicates the L_p norm. We will consider $p = 2$ namely the Euclidean norm in this paper.

Proposition 1 *[5] The Adaboost algorithm fits an additive logistic regression model by using quasi-Newton method using the analytical Hessian matrix updates for minimizing the expected value of the loss function.*

Let us consider the following exponential loss function

$$J(f) = exp(\| \ y - F - f \ \|^2) \tag{2}$$

Let us now define the residual r as the absolute difference between $\mathcal{Y}$ and F. We chose to use first derivative updates (for faster convergence) by choosing the weak regressor using the residual ($f = r$).

2.3 Scale-space Kernels

Let us consider the general regression problem which is a continuous mapping $p(x) : \mathbb{R}^d \rightarrow \mathbb{R}$. In scale-space theory, $p(x)$ is embedded into a continuous family $P(x, \sigma)$. Our method starts with an approximation of the entire dataset with

Gaussian kernel of $\sigma = 0$. As the resolution (or scale) increases, the sigma value is reduced and eventually converges to zero. In our case, the highest frequency (or resolution) corresponds to fitting every data point with a Gaussian kernel. In simple terms, one can write the new kernel $p(x, \sigma)$ as a convolution of $p(x)$ with a Gaussian kernel $g(x, \sigma)$. As described earlier, choosing optimal σ value during every iteration of boosting becomes a challenging task. In other words, one cannot predetermine the reduction in the σ value. We choose to reduce it by halves using the concepts of wavelet decomposition methods. In signal processing applications, wavelet transformation constructs a family of hierarchically organized decompositions [12]. The frequencies in the wavelet domain correspond to resolutions in our scale-space based algorithm. The original target function ($\mathcal{Y}$) is decomposed using weak regressors(f) and residuals(r). The final regression model at any given resolution is obtained by a weighted linear combination of the weak regressors obtained so far.

3 Scale-space based Framework

Algorithm 1 describes our scale-space based approach for boosting regression problems. The initial regressor is set to the mean value of the target values. The main program runs for a predefined number (T) of iterations. To make the problem simpler, 1) we control the resolution of the kernel using the number of data samples; 2) we fit the target values, $\mathcal{Y}$, only using one feature, $\mathcal{X}_i, i \in [1, d]$, at a time. Initially, the number of data points to be modeled is set to the total number of samples in the dataset. $\mathcal{X}_i$'s are sorted independently by column-wise and the indices corresponding to each column are stored. This will facilitate the Gaussian based regression modeling that will be performed later on. For every iteration, the best kernel is fit to the data based on a single feature, $\mathcal{X}_i, i \in [1, d]$, at a particular resolution. The procedure $bestkernelfit$ performs this task for a resolution corresponding to n data points. We used Gaussian weak regressors as our kernels since the Gaussian kernels are one of the most popular choice for scale-space kernel. The basic idea is to slide a Gaussian window across all the sorted data points corresponding to each feature, $\mathcal{X}_i, i \in [1, d]$, at a given resolution.

As the iterations progress, the number of data points considered for fitting the weak regressor is retained or halved depending on the error of the model. In other words, depending on the error at a given iteration, the resolution of the data is maintained or increased for the next iteration. For every iteration, the residual r is set to the absolute difference between the target value ($\mathcal{Y}$) and the final regressor (F). By equating the first derivative of the loss function to zero, we will set the residual as the data to be modeled during the next iteration using another weak regressor. The main reason for retaining the resolution in the next iteration is that sometimes there might be more than one significant component at that particular resolution. One iteration can model only one of these components. In order to model the other components, one has to perform another iteration of obtaining the best Gaussian regressor at the same resolution.

Increasing the resolution for the next iteration in this case might fail to model the component accurately. Only after ensuring that there are no more significant components at a given resolution, our algorithm will increase the resolution for the next iteration. Hence, the best Gaussian regressor corresponding to n or $n/2$ data points is obtained at every iteration and the model with the least error added to the final regressor. The main aspect of our algorithm, which is the scale-space, can be seen from the fact that the resolution of the data to be modeled is either maintained or increased as the number of iterations increase. Hence, the algorithm proposed here can be more generally termed as "*scale-space based Boosting*" that can model any arbitrary function using the boosting scheme with scale-space based weak regressors. Our algorithm obtains the weak regressors and models the data in a more systematic (hierarchical) manner. Most importantly, the change in resolution is monotonically non-decreasing, i.e. the resolution either remains the same or increased.

Algorithm 1 Scale-space Boosting

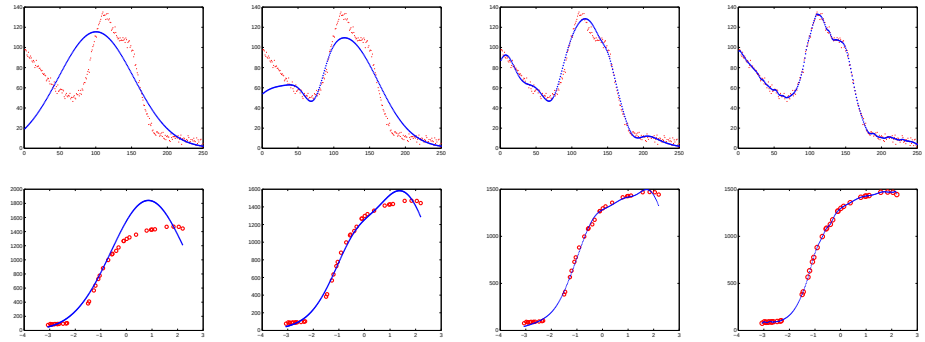
Input: Data ($\mathcal{D}$), No. of samples (N), No. of iterations (T).
Output: Final Regressor (F)
Algorithm:
set $n = N, \quad F = \emptyset$
for $i = 1 : d$ **do**
  $[\hat{\mathcal{X}}, idx(:,i)] =$sort$(\mathcal{X}(:,i))$
end for
for $t = 1 : T$ **do**
  $r = |\mathcal{Y} - F|$
  $[\hat{f}_0, err_0] = bestkernelfit(\hat{\mathcal{X}}, r, N, d, n, idx)$
  $[\hat{f}_1, err_1] = bestkernelfit(\hat{\mathcal{X}}, r, N, d, n/2, idx)$
  if $err_0 < err_1$ **then**
    $F = F + \hat{f}_0$
  else
    $F = F + \hat{f}_1$
    $n = n/2$
  end if
end for
return F

4 Experimental Results

We performed experiments using two non-linear regression datasets from NIST StRD (Statistics Reference Datasets [13]). We selected two datasets : (1) *Gauss3* from the category of average level of difficulty containing 250 samples with 1 predictor variable (x) and 1 response variable (y). (2) *Thurber* from the category of high level of difficulty containing 37 samples with 1 predictor variable (x) and 1 response variable (y). Figure 1 shows experimental results on these two datasets

using the proposed scale-space boosting algorithm after 1, 5, 10 and 50 iterations are shown. We also ran our experiments on the following more complicated real world datasets:

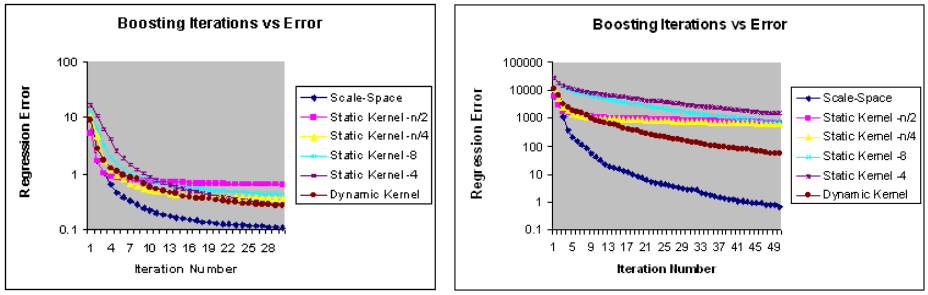
– **Diabetes** [14] dataset contains 43 samples with 2 predictor variables.
– **Ozone** [1] dataset contains 330 samples with 8 predictor variables.
– **Abalone** [15] dataset contains 4177 samples with 8 predictor variables.



(a) Iteration 1 (b) Iteration 5 (c) Iteration 10 (d) Iteration 50

Fig. 1. Experimental results for *Gauss3* (first row) and *Thurber* (second row) datasets after 1,5,10 and 50 iterations.

4.1 Discussion



(a) Diabetes (b) Ozone

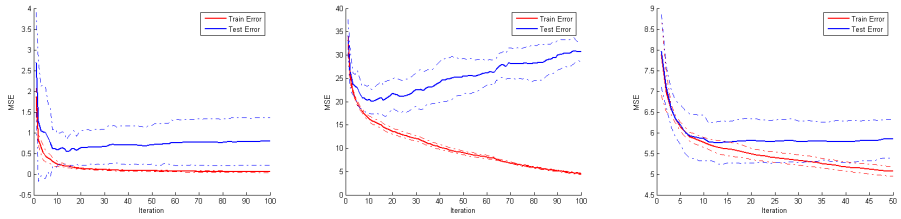
Fig. 2. Convergence of the regression error during the boosting procedure (training phase) using scale-space kernel and other static and dynamic kernels of various widths.

The scale-space boosting algorithm is very effective in reducing the error quickly in the first few iterations. Significant reduction in the training error occurs within first 10 boosting iterations. By using scale-space kernels, one can achieve the optimal point (point where the over-fitting starts) within this first few iterations. Usually this point is obtained after at least 40 boosting iterations in the case of static kernels as shown in Fig. 2, which gives the convergence of the regression error during the boosting iterations. Clearly the behavior of the convergence is similar to static kernels of very less width but the error is much lesser in the case of scale-space kernel. The main reason for using the scale-space framework is for faster convergence of the results by *dynamically choosing the weak regressors* during the boosting procedure. One can also see the comparison between the convergence behaviour of a randomly chosen dynamic kernel versus the scale-space kernel. Choosing an optimal weak regressor by exploring all possibilities might yield a better result, but it will be computationally inefficient and infeasible for most of the practical problems. For such problems, scale-space kernels will give the users with a great flexibility of adaptive kernel scheme at a very low computational effort (also considering the fact of speedy convergence). To the best of our knowledge, this is the first attempt to use the concepts of scale-space theory and wavelet decomposition in the context of boosting algorithms for any regression modeling.

We also demonstrate that the scale-space framework does not suffer from the over-fitting problem. Fig. 3 shows the train and test errors during the boosting iterations along with the standard deviation using 5-fold cross validation scheme for the different datasets. For improving the computational efficiency, the sliding window kernel in the $bestkernelfit$ procedure is moved in steps of multiple data points rather than individual data points. One other advantage of using the scale-space based boosting approach is that it obtains smooth regression functions (approximators) at different level of accuracies as shown in our results. This cannot be achieved by using a decision tree or a boosting stump though they might yield lower RMSE values for prediction. Hence, our comparisons were specifically made with other smooth kernels that were used in the literature.

5 Conclusions and Future Research

Recently, boosting have received great attention from several researchers. Choosing optimal weak regressors and setting their parameters during the boosting iterations have been a challenging task. In this paper, we proposed a novel boosting algorithm that uses scale-space theory to obtain the optimal weak regressor at every iteration. We demonstrated our results for logitboost based regression problems on several real-world datasets. Similarities and differences of our method compared to other popular models proposed in the literature are also described. Extensions to Adaboost framework and use of scale-space kernels in classification problems are yet to be investigated. Effects of different loss functions in this scale-space boosting framework will also be studied in the future.



(a) Diabetes (b) Ozone (c) Abalone

Fig. 3. Results of training and test error on different datasets using 5-fold cross validation. The solid lines indicate the mean of the error and the dashed lines indicate the standard deviation in the errors.

References

1. Breiman, L.: Arcing classifiers. The Annals of Statistics **26**(3) (1998) 801–849
2. Hastie, T., Tibshirani, R., Friedman, J.: Boosting and Additive Trees. In: The Elements of Statistical Learning. Data Mining, Inference, and Prediction. Springer-Verlag New York (2001)
3. Sporring, J., Nielsen, M., Florack, L., Johansen, P.: Gaussian Scale-Space Theory. Kluwer Academic Publishers (1997)
4. Breiman, L.: Bagging predictors. Machine Learning **24**(2) (1996) 123–140
5. Friedman, J.H., Hastie, T., Tibshirani, R.: Additive logistic regression: A statistical view of boosting. Annals of Statistics **28**(2) (2000) 337– 407
6. Zemel, R.S., Pitassi, T.: A gradient-based boosting algorithm for regression problems. Neural Information Processing Systems (2000) 696–702
7. Schapire, R.E., Singer, Y.: Improved boosting using confidence-rated predictions. Machine Learning **37**(3) (1999) 297–336
8. Zhu, J., Rosset, S., Zou, H., Hastie, T.: Multi-class adaboost. Technical Report 430, Department of Statistics, University of Michigan (2005)
9. Buhlmann, P., Yu, B.: Boosting with the l2 loss: Regression and classification. Journal of American Statistical Association **98**(462) (2003) 324–339
10. Lindeberg, T.: Scale-space for discrete signals. IEEE Transactions on Pattern Analysis Machine Intelligence **12**(3) (1990) 234–254
11. Leung, Y., Zhang, J., Xu, Z.: Clustering by scale-space filtering. IEEE Transactions on Pattern Analysis Machine Intelligence **22**(12) (2000) 1396–1410
12. Mallat, S.: A theory for multiresolution signal decomposition : the wavelet representation. IEEE Transaction on Pattern Analysis and Machine Intelligence **11** (1989) 674–693
13. Information Technology Laboratory, N.I.o.S., (NIST), T.: Nist strd (statistics reference datasets). http://www.itl.nist.gov/div898/strd/
14. Hastie, T., Tibshirani, R. In: Generalized additive models. Chapman and Hall, London (1990) 304
15. Blake, C., Merz, C.: UCI repository of machine learning databases. http://www.ics.uci.edu/ mlearn/MLRepository.html, University of California, Irvine, Dept. of Information and Computer Sciences (1998)