Multi-resolution Boosting for Classification and Regression Problems

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

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

Abstract. Various forms of boosting techniques have been popularly used in many data mining and machine learning related applications. Inspite of their great success, boosting algorithms still suffer from a few open-ended problems that require closer investigation. The efficiency of any such ensemble technique significantly relies on the choice of the weak learners and the form of the loss function. In this paper, we propose a novel multi-resolution approach for choosing the weak learners during additive modeling. Our method applies insights from multi-resolution analysis and chooses the optimal learners at multiple resolutions during different iterations of the boosting algorithms. We demonstrate the advantages of using this novel framework for classification tasks and show results on different real-world datasets obtained from the UCI machine learning repository. Though demonstrated specifically in the context of boosting algorithms, our framework can be easily accommodated in general additive modeling techniques.

1 Introduction

In the field of data mining, ensemble methods have been proven to be very effective for not only improving the classification accuracies but also in reducing the bias and variance of the estimated classifier. We choose to demonstrate our multi-resolution based framework using '*boosting*' algorithm, which is a standard additive modeling algorithm popular in data mining and machine learning domains. The Boosting meta-algorithm is an efficient, simple, and easy to manipulate additive modeling technique that can use potentially any weak learner available [8]. 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" [3]. In simple terms, boosting algorithms combine weak learning models that are slightly better than random models. Recently, several researchers in other domains like computer vision, medical imaging have started using boosting algorithms extensively for real-time applications. Both classification and regression based boosting algorithms have been successfully

T. Theeramunkong et al. (Eds.): PAKDD 2009, LNAI 5476, pp. 196–207, 2009.

used in a wide variety of applications in the fields of computer vision [12], information retrieval [11], bioinformatics [9] etc. Inspite of their great success, boosting algorithms still suffer from a few open-ended issues such as the choice of the parameters for the weak learner. The framework proposed in this paper is more generally termed as "*Multi-resolution Boosting*", which can model any arbitrary function using the boosting methodology at different resolutions of either the model or the data. Here, we propose a novel boosting model that can take advantage of using the weak learners at multiple resolutions. This method of handling different resolutions and building effective models is similar to wavelet decomposition methods for multi-resolution signal analysis. In this work, we achieve this *multi-resolution* concept in the context of boosting algorithms by one of the following two ways:

- *Model-driven multi-resolution:* This is achieved by varying the complexity of the classification boundary. This approach will provide a systematic procedure that increases the complexity of the weak learner as the boosting iterations progress. This framework not only obtains weak learners in a systematic manner, but also reduces the over-fitting problem as discussed in Section 4.1 of this paper.
- *Data-driven multi-resolution:* This can be achieved by considering the data (not the model) at multiple resolutions during each iteration in the boosting algorithm. Our framework chooses the weak learners for the boosting algorithm that can best fit the current resolution and as the additive modeling iterations progress, the modeling resolution is increased. The amount of increase in the resolution follows from the theory of wavelet decomposition. Our algorithm provides the flexibility for dynamically choosing the weak learner compared to static learners with certain pre-specified parameters. This framework is discussed in Section 4.2 of this paper.

The main idea of the proposed framework is: *the use of Multi-resolution data (or model) driven fitting in the context of additive modeling using concepts that are similar to wavelet decomposition techniques.* The rest of the paper is organized as follows: Section 2 gives some relevant background on various boosting techniques and scale-space kernels. Section 3 shows the problem formulation in detail and discusses the concepts necessary to comprehend our algorithm. Section 4 describes both the model-driven and the data-driven multi-resolution boosting frameworks. Section 5 gives the experimental results of the proposed methods on real-world datasets and Section 6 concludes our discussion.

2 Relevant Background

Ensemble learning [4] is one of the most powerful modeling techniques that was found to be effective in a wide variety of applications in recent years. different ensemble techniques have been proposed in the literature and is still a very active area of research. Boosting is one of the most widely used algorithm that has

caught the attention of several researchers working in the areas of pattern recognition and machine learning [5]. A main advantage of boosting algorithms 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 weak learner. This is a very desirable property of the boosting algorithms that can be applied in several applications for predictive modeling [8,6]. The additive model provides a reasonable flexibility in choosing the optimal weak learners for a desired task. In this paper, we propose a novel multi-resolution framework for choosing optimal weak learners during the iterations in boosting. This approach allows for effective modeling of the dataset at any given resolution [10]. In terms of analyzing (or modeling) a given dataset at different resolutions, our approach closely resembles wavelet decomposition techniques which are effective tools in the field of multi-resolution signal analysis [7]. In the model-driven multi-resolution boosting framework, the models are built by increasing the complexity during the boosting process. The data-driven multi-resolution, on the other hand, considers the data at different resolutions decomposition techniques which are effective tools in the field of multi-resolution signal analysis. The main advantages of using this multiple resolution framework in the context of boosting are that they:

- allow systematic hierarchical modeling of the final target model.
- provide more flexibility by allowing the user to stop at a reasonable resolution and thus avoid the over-fitting problem.
- require very few pre-defined user parameters.
- avoid the use of strong learners in the beginning stages of modeling and progressively use them towards the end.

3 Problem Formulation

Let us consider N i.i.d. training samples $\mathcal{D} = (\mathcal{X}, \mathcal{Y})$ 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 $\mathcal{Y} \in \mathbb{R}^{N \times 1}$. 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. Now, we will discuss boosting algorithms applied to general classification problems. We choose to demonstrate the power of scale-space kernels in the context of Logitboost algorithm because of its popularity and its power of demonstrating the additive modeling nature.

Each boosting iteration performs the following three steps: (1) Computes response and weights for every datapoint. (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.

In the case of classification problems, the penalty function induced by the error estimation is given by:

$$L(y_i, F_t(x_i)) = I(y_i \neq F^{(t)}(x_i)) \tag{1}$$

where I denotes an indicator function which returns value 0, when $y_i \neq F^{(t)}(x_i)$ and 1 otherwise. In other words, the penalty term is 1 if the i^{th} sample is misclassified and 0 if it is correctly classified. Whether it is a classification or a regression problem, the main challenges in the boosting framework are the following: (i) The choice of the weak learner and (ii) The complexity of the weak learner. While choosing a weak learner model can be a complicated task in itself, tuning the right complexity for such a weak learner might be even more challenging. The multi-resolution framework proposed in this paper addresses the second issue.

The boosting framework discussed above works for classification problems and can be easily adapted to solve regression problems. In the case of regression problems, the penalty function is given by:

$$L(y_i, F^{(t)}(x_i)) = \|y_i - F^{(t)}(x_i)\|_p \tag{2}$$

where $\|\cdot\|_p$ indicates the L_p norm. We will consider $p = 2$ (namely, the Euclidean norm) in this paper. We formulate this multi-resolution boosting using the standard boosting algorithm with exponential L_2 norm loss function and demonstrate empirical results on classification problems. In our previous work [10], we have demonstrated the use of scale-space kernels in the data-driven boosting framework on several regression datasets.

Algorithm 1. Model-driven Multi-Resolution Boosting

Input: Data ($\mathcal{X}$), No. of samples (N), No. of iterations (T).
Output: Final model (F)
Algorithm:
Initialize the weight vector $W^{(1)}$ such that $w_i^1 = 1/N$ for $i = 1, 2, .., N$
$nplits = 1$
for $t = 1$ to T **do**
  $[\hat{f}_0, err_0] = Train(\mathcal{X}, W^{(t)}, nsplits)$
  $[\hat{f}_1, err_1] = Train(\mathcal{X}, W^{(t)}, nsplits + 2)$
  if $err_0 < err_1$ **then**
    $f_t = \hat{f}_0 \quad \epsilon_t = err_0$
  else
    $f_t = \hat{f}_1 \quad \epsilon_t = err_1$
    $nsplits = nsplits + 1$
  end if
  Compute $\alpha_t = \frac{1}{2} ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$
  Modify the training weight $w_i^{(t+1)}$ as follows:

$$w_i^{(t+1)} = \frac{w_i^{(t)} \cdot exp(-\alpha_t y_i f_t(x_i))}{z_t}$$

  where z_t is the normalization factor (chosen so that $\sum_{i=1}^{N} w_i^{(t+1)} = 1$)
end for
Output the final model $F(\mathcal{X}) = \sum_{t=1}^{T} \alpha_t f_t(\mathcal{X})$

4 Multi-resolution Boosting Framework

We will now describe both the model-driven and data-driven multi-resolution boosting algorithms. To demonstrate a reasonably wide applicability of the multi-resolution framework, we implement our framework using both the adaboost and logitboost algorithms. We show the model-driven multi-resolution algorithm using the adaboost framework for classification problems and the data-driven multi-resolution algorithm using the logitboost framework for regression problems. Though, we chose to demonstrate in this setting, the proposed framework is generic and can be applied to other additive modeling techniques used for solving classification and regression problems.

4.1 Model-Driven Multi-resolution Boosting

In the model driven boosting framework, the complexity of weak learner is modified as the boosting iterations progress. Changing the complexity of the weak model can be done in a very intuitive manner depending on the choice of the weak learner. For example, if decision trees are used as a weak learner, the resolution can be changed by changing the number of levels in the decision tree that is being considered. The initial boosting iterations use trees with only one level (or decision stumps) and later on the resolution can be increased by increasing the tree-depth. One has to note that the complexity of the modeling (or classification boundary) is significantly increased by changing the resolution. Algorithm 1 describes our model-driven multi-resolution boosting framework using the adaboost algorithm for a *binary classification* problem. The weight vector W is initialized to $1/N$ (uniform). The main algorithm runs for a predefined number (T) of iterations. The procedure $Train$ will obtain weak learner (and the corresponding training error) using the weights $W^{(t)}$. The number of splits ($nsplits$) is a parameter that determines the complexity of the model i.e. the more the number of splits in the weak learner, the more the complexity of the model. It is initialized to one at the beginning. As the iterations progress, the complexity of the weak learner is either retained or incremented depending upon the training error.

For every iteration, the training error of the current model is compared with the error of a slightly complex model (with $nsplits + 2$ nodes in the tree). If this new model performs well, then the complexity of the current model is increased ($nsplits = nsplits + 2$) and the re-weighting of the data points is computed using this new model. The weights are normalized (so that they sum to one) in every iteration. One can see that the algorithm appears to be working in a similar manner to the traditional Adaboost, except for the fact that the choice of the weak learner is made more systematically from simple to complex and is not chosen arbitrarily as done in the standard boosting procedure. In this manner, the algorithm increases the complexity of the weak learners chosen and the final weighted combinations of the selected weak learners are used as the final trained model. Hence, the model will have a very simple classification boundary

in the initial stages and the boundary becomes more and more complex as the iterations proceed.

4.2 Data-Driven Multi-resolution Boosting

In this section, we will describe the data-driven approach where we maintain the same complexity of the weak learner, but change the number of data points to be modeled during each boosting iteration. Algorithm 2 describes our data-driven multi-resolution boosting framework for a *regression problem*. As mentioned earlier, this approach is demonstrated using the logitboost algorithm. The initial model is set to null or to the mean value of the target values. The main program runs for a predefined number (T) of iterations. Initially, *res* is set to 1 indicating the simplest model possible (which will consider all the data points). The feature values are sorted independently by column-wise and the indices corresponding to each column are stored. As the iterations progress, the resolution considered for fitting the weak learner is retained or doubled depending on the error. In other words, depending on the error obtained at a given iteration, the resolution of the data is either maintained or increased for the next iteration. For every iteration, the residual r is computed depending on the difference between the target value ($\mathcal{Y}$) and the final model (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. Using the quasi-Newton's method the data to be modeled in the next iteration will be set to $-(I + 2rr^T)^{-1} \cdot r$. The best multivariate Gaussian model will be fitted to this data at a given resolution.

Theorem 4.1. *During each boosting iteration, the minimum of the loss function is achieved by setting $f = r$ and the Newton's update is chosen by setting $f = -(I + 2rr^T)^{-1} \cdot r$.*

Proof. We will discuss the derivations for the first derivative and the second derivative and show the Newton updates in the case of the boosting for regression problems. Consider the following exponential loss function:

$$L(y, F, f) = exp(\parallel y - F - f \parallel^2)$$

For the Newton's update equation, we need to compute the first and second derivatives with respect to $f(x)$ and evaluate them at $f(x) = 0$.

$$s(x) = \frac{\partial L(y, F, f)}{\partial f(x)}|_{f(x)=0} = 2exp(\parallel r - f \parallel)(r - f)|_{f=0} = 2 \cdot exp(r^T r) \cdot r$$

Taking the derivative again, we have

$$H(x) = \frac{\partial^2 L(y, F, f)}{\partial f(x)^2}|_{f(x)=0} = 2exp(\parallel r - f \parallel^2) \cdot I$$
$$+ 4exp(\parallel r - f \parallel^2) \cdot (r - f) \cdot (r - f)^T|_{f=0} = 2exp(r^T r) \cdot (I + 2rr^T)$$

Hence, the inverse of the Hessian becomes

$$H^{-1}(x) = \frac{(I + 2rr^T)^{-1}}{2exp(r^T r)}$$

Finally, the Newton's update is given as follows:

$$F(x) = F(x) - H(x)^{-1}s(x) = F(x) - (I + 2rr^T)^{-1} \cdot r$$

Hence, we plug-in the value $-(I + 2rr^T)^{-1} \cdot r$ as the regression value to be modeled using the weak regressor. Also, we can notice that the minimum of the loss function can also be obtained by equating the first derivative to zero.

$$2exp(\parallel r - f \parallel)(r - f) = 0 \Rightarrow r = f$$

In other words, by modeling the residual directly using the weak regressor, the minimum of the loss function can be obtained. *End of Proof*

The details of the procedure $bestfit$ which obtains the best weak model at a given resolution of the data is described in the next section. The main reason for retaining the resolution of the next iteration is that sometimes there might be more than one significant component at that given resolution. One iteration can model only one of these components. In order to model the other component, one has to perform another iteration of obtaining the best weak model at the same resolution. Increasing the resolution for the next iteration might fail to model the component accurately. 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 weak model corresponding to current resolution or next higher resolution is obtained at every iteration and the model with the lowest error is added to the final model.

For every iteration, the best weak model is fit to the data based on a single feature value at a given resolution. This is performed using the $bestfit$ function in the algorithm. One way of achieving the multi-resolution in this context is to use scale-space kernel to model a subset of data and handling the data in a multi-resolution fashion. The procedure $bestgaussfit$ (instead of $bestfit$) performs this task for a particular value of resolution. Additive modeling with smooth and continuous kernels will result in smooth functions for classifier boundary and regression functions. Gaussian kernels are a simple and a trivial choice for scale-space kernels that are powerful universal approximators. Also, Gaussian kernels allow generative modeling of a target function which is a good choice for many applications like object detection. The basic idea is to slide a Gaussian window across all the datapoints corresponding to each feature at a given resolution. Algorithm 3 contains two loops. The outer loop ensures that the Gaussian fit has to be computed for each feature and the inner loop corresponds to the sliding Gaussian. In other words, depending on the given resolution (indicated by n datapoints), a Gaussian kernel containing n datapoints is moved across all the data points and the location where the minimal residual error is obtained.

The result f is obtained by fitting a Gaussian kernel computed using weighted median (μ) and standard deviation (σ) for the datapoints within this window.

Algorithm 2. Data-driven Multi-Resolution Boosting

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

Algorithm 3. *bestgaussfit*

Input: Sorted feature data ($\hat{\mathcal{X}}$), No. of samples(N), No. of samples to fit Gaussian (n), Residual vector (r), Sorting indices (idx).
Output: Best fit Regressor ($\hat{f}$), Error (Err_{min})
Algorithm:
Err_{min} =MAXDOUBLE
for $i = 1 : d$ **do**
  for $j = 1 : N - n + 1$ **do**
    $\hat{x} = \hat{\mathcal{X}}(:, j : j + n - 1)$
    $\hat{r} = r(idx(j : j + n - 1, i))$
    $wgt(1 : n) = abs(\hat{r}(1 : n))/sum(abs(r))$
    $\mu = E_{wgt}(\hat{x}) = wgt^T * \hat{x}$
    σ = sqrt$(E_{wgt}((\mu - \hat{x})^2))$
    f =normpdf$(\hat{X}, \mu, \sigma)$
    β =sum$(\hat{r})$/sum$(f(j : j + n - 1))$
    $err = (r - \beta f)^T \cdot (r - \beta f)$
    if $err < Err_{min}$ **then**
      $Err_{min} = err$
      $\hat{f} = f$
    end if
    $f = min(\hat{f}(1 : d))$
  end for
end for
return $\{f, Err_{min}\}$

After obtaining the weak learner it must be scaled (scale factor is β) according to the target values. Finally, the error is computed between the weak learner and the target values. If the error with the new model is improved, the resolution is doubled (change at a logarithmic scale) or in other words, the number of datapoints considered to fit a Gaussian is halved. In fact, we can use any other heuristic to change the resolution more efficiently. Experimental results showed that this change of resolution is optimal and also this logarithmic change of resolution has nice theoretical properties as they mimic some of the wavelet decomposition methods.

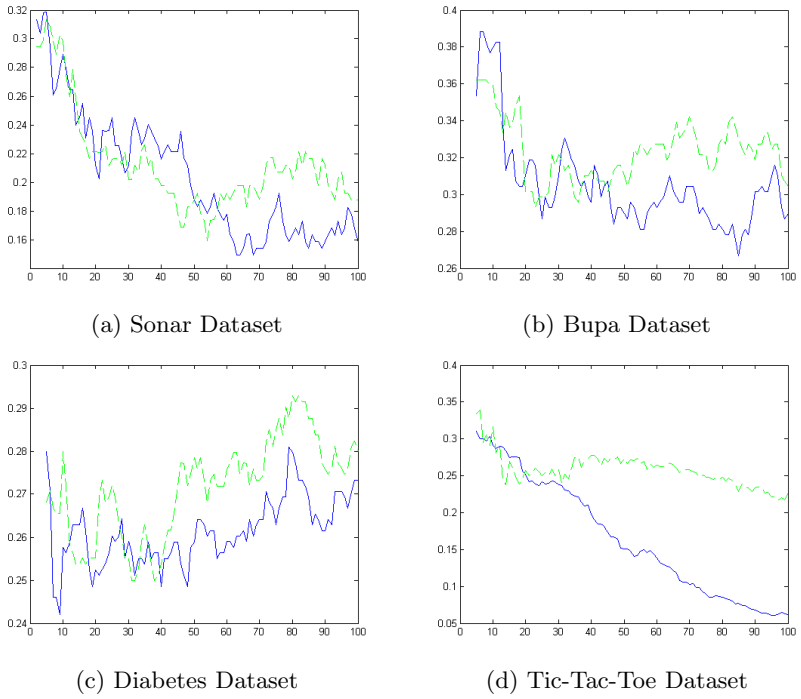
The multi-resolution aspect of our algorithm can be seen from the fact that the resolution of the data to be modeled is either maintained or increased as the number of iterations increase. In fact, one might interpret this approach as an improvement in the weak learner alone because the algorithm proposed here will obtain improved weak learner at every iteration and hence the overall boosting will have faster convergence. We consider that the main contribution of this paper is not just at the level of choosing a weak learner but it is at the junction between the choice of weak learner and the iterations in the boosting algorithm. Also, our algorithm obtains the weak models in a more systematic hierarchical manner. Most importantly, the increase in the resolution is monotonically non-decreasing, i.e. the resolution either remains the same or increased.

5 Experimental Results

We will now demonstrate our results on some real-world datasets. All experiments were run in MATLAB 7.0 and on a pentium IV 2.8 GHz machine. Six different real world binary classification datasets were chosen from the UCI machine learning repository [2]. Multi-class classification problems can also be performed using methods similar to [1]. Two different sets of experiments were conducted on these datasets to illustrate the power of multi-resolution boosting. In order to demonstrate the model-driven framework, decision trees at multiple resolutions (different number of levels in the decision tree) are considered, and in order to demonstrate the data-driven framework, Gaussian kernels are considered for fitting the data at multiple resolutions.

5.1 Results for Model-Driven Multi-resolution

Fig. 1 shows the test error results on different datasets during the boosting iterations. Comparisons are made between the standard Adaboost and the multi-resolution boosting framework. We can see that the error obtained using the multi-resolution boosting procedure is significantly lower compared to the standard procedure. This clearly illustrates the fact that the multi-resolution scheme is less prone to the over-fitting problem. Under this framework, during the initial iterations of boosting, decision stumps (trees with only one level of child nodes) are used. As the iterations proceed, more deeper trees (with levels greater than 2) are used for modeling. This way, a hierarchical approach is used for computing the classification boundary from low resolution to high resolution. Using a



(a) Sonar Dataset (b) Bupa Dataset

(c) Diabetes Dataset (d) Tic-Tac-Toe Dataset

Fig. 1. Test error during the boosting iterations on various datasets. The dashed line gives the error obtained using the standard Adaboost algorithm and the solid line gives the error obtained using the model-driven multi-resolution boosting algorithm.

high-resolution weak models will suffer from the problem of over-fitting. For example, by using a tree with many levels in the first few iterations in the boosting procedure might obtain a very complicated decision boundary which is prone to the over-fitting problem. Also, it will be expensive to use complex trees from the start of the boosting procedure when it is not required to have a complex decision boundary.

5.2 Results for Data-Driven Multi-resolution

We demonstrate the power of data-driven multi-resolution approach using scale-space kernels on binary classification problems. Additive modeling with smooth and continuous kernels will result in smooth functions for classifier boundary and regression functions. Since, obtaining the width of the kernel during the boosting process can be a challenging task, the use of scale-space kernels can resolve the problem by using adaptive step-sizes by a '*global-to-local*' fitting process. One cannot predetermine the reduction in the kernel width. In our multi-resolution framework, we choose to reduce it by halves using the concepts of wavelet decomposition methods which were well studied concepts in the context of handling image operations efficiently. We compare the performance of these scale-space

Table 1. Experimental results of Data-Driven Multi-Resolution boosting. Performance of scale-space kernels with other kernels on various real-world datasets. Test error along with the standard deviation using five-fold cross validation procedure is reported.

Dataset	Cancer	Ionosphere	Sonar	Bupa	Tic-Tac-Toe	Diabetes
Number of Samples	569	351	208	345	958	768
Number of Features	30	34	60	6	9	8
static kernel -n/2	0.1938±0.05	0.3647± 0.08	**0.6632± 0.13**	0.8785± 0.08	0.6947± 0.04	0.6765± 0.06
static kernel -n/4	0.1993 ±0.03	**0.333 ±0.08**	0.6697±0.07	0.9156 ± 0.11	0.5725± 0.02	0.6419±0.06
static kernel -n/8	0.244± 0.09	0.4118± 0.1	0.9148± 0.13	0.9453± 0.06	0.5644 ± 0.02	0.657±0.019
static kernel -8	0.7638 ±0.07	0.503 ±0.06	1.144 ± 0.15	0.9384± 0.11	0.5662± 0.03	0.7487±0.06
Dynamic kernel	0.1898± 0.03	0.3553 ± 0.06	0.7543± 0.09	**0.869 ±0.05**	0.5726± 0.03	0.6676±0.07
Exhaustive kernel	0.2325± 0.06	0.4243 ± 0.12	0.8068±0.30	0.9643± 0.12	0.5624±0.04	0.6546 ±0.07
Scale-space kernel	**0.1895 ± 0.04**	0.3371± 0.09	0.7125± 0.14	0.8962 ± 0.13	**0.5603± 0.038**	**0.6386± 0.05**

kernels with other static and dynamic kernels. Exhaustive kernel is the most expensive one which tries to fit a kernel of various widths during each iteration of boosting. Dynamic kernel (or random kernel) fits a kernel of random width during the boosting process. Static kernels will have static widths that do not change during the boosting process.

Compared to other static kernels of fixed width, the scale-space kernels do not suffer from the generalization problem as clearly illustrated by the results on the test data shown in Table 1. Scale-space kernels consistently perform better than the exhaustive or dynamic kernels. For some datasets, wider static kernels perform better than the scale-space kernels and for other datasets static kernels with lesser width perform better. However, scale-space kernels are competitive with the best possible kernels and can be generically used for any dataset. Overall, the scale-space kernels are less than twice as expensive as the static width kernels. One can also see that the results of the scale-space kernels are fairly robust compared to other kernels. This multi-resolution framework will provide a systematic hierarchical approach of obtaining the classification boundary in the context of additive modeling. One of the main reasons for using the scale-space framework is for faster convergence of the results by *dynamically choosing the weak regressors* during the boosting procedure. 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 fast convergence). The fact that the scale-space kernels converge much faster than static kernels make them more suitable for additive modeling algorithms. 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 predictive modeling.

6 Conclusion

Recently, additive modeling techniques have received a great attention from several researchers working in a wide variety of applications in science and engineering. Choosing optimal weak learners and setting their parameters during the modeling have been a crucial and challenging task. In this paper, we

proposed a novel boosting algorithm that uses multi-resolution framework to obtain the optimal weak learner at every iteration. We demonstrated our results for logitboost based regression problems on real-world datasets. Advantages of our method compared to existing methods proposed in the literature is clearly demonstrated. As a continuation of this work, we would like to perform the generalization of the multi-resolution approach for other ensemble learning techniques.

References

1. Allwein, E., Schapire, R., Singer, Y.: Reducing multiclass to binary: a unifying approach for margin classifiers. J. Machine Learning Research 1, 113–141 (2001)
2. Blake, C.L., Merz, C.J.: UCI repository of machine learning databases. University of California, Irvine, Dept. of Information and Computer Sciences (1998), `http://www.ics.uci.edu/~mlearn/MLRepository.html`
3. Breiman, L.: Arcing classifiers. The Annals of Statistics 26(3), 801–849 (1998)
4. Dietterich, T.G.: Ensemble methods in machine learning. In: First International Workshop on Multiple Classifier Systems, pp. 1–15 (2000)
5. Freund, Y., Schapire, R.E.: Experiments with a new boosting algorithm. In: International Conference on Machine Learning, pp. 148–156 (1996)
6. Friedman, J.H., Hastie, T., Tibshirani, R.: Additive logistic regression: A statistical view of boosting. Annals of Statistics 28(2), 337–407 (2000)
7. Graps, A.L.: An introduction to wavelets. IEEE Computational Sciences and Engineering 2(2), 50–61 (1995)
8. Hastie, T., Tibshirani, R., Friedman, J.: Boosting and Additive Trees. In: The Elements of Statistical Learning. Data Mining, Inference, and Prediction. Springer, New York (2001)
9. Hong, P., Liu, X.S., Zhou, Q., Lu, X., Liu, J.S., Wong, W.H.: A boosting approach for motif modeling using chip-chip data. Bioinformatics 21(11), 2636–2643 (2005)
10. Park, J.-H., Reddy, C.K.: Scale-space based weak regressors for boosting. In: Kok, J.N., Koronacki, J., Lopez de Mantaras, R., Matwin, S., Mladenič, D., Skowron, A. (eds.) ECML 2007. LNCS, vol. 4701, pp. 666–673. Springer, Heidelberg (2007)
11. Schapire, R., Singer, Y., Singhal, A.: Boosting and rocchio applied to text filtering. In: Proceedings of ACM SIGIR, pp. 215–223 (1998)
12. Viola, P.A., Jones, M.J.: Robust real-time face detection. International Journal of Computer Vision 57(2), 137–154 (2004)