Scale-Space Kernels for Additive Modeling

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 additive modeling techniques have been popularly used in many pattern recognition and machine learning related applications. The efficiency of any additive modeling technique relies significantly on the choice of the weak learner and the form of the loss function. In this paper, we propose a novel scale-space kernel based approach for additive modeling. Our method applies a few insights from the well-studied scale-space theory for choosing the optimal learners during different iterations of the boosting algorithms, which are simple yet powerful additive modeling methods. For each iteration of the additive modeling, weak learners that can best fit the current resolution of the data are chosen and then increase the resolution systematically. We demonstrate the results of the proposed framework on both synthetic and real datasets taken from the UCI machine learning repository. Though demonstrated specifically in the context of boosting algorithms, our approach is generic enough to be accommodated in general additive modeling techniques. Similarities and distinctions of the proposed algorithm with the popularly used radial basis function networks and wavelet decomposition method are also discussed.

Keywords: additive modeling, boosting, weak learner, wavelet decomposition, scale-space theory.

1 Introduction

In statistical pattern recognition, additive modeling 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 framework using '*boosting*' methods, which is a standard additive modeling algorithm popular in pattern recognition applications. In spite of its great success, boosting algorithms still suffer from a few open-ended issues such as the choice of the parameters for the weak learner. In this paper, we propose a novel scale-space based scheme for choosing optimal weak learners at multiple resolutions during the iterations in boosting or generic additive modeling. The proposed framework can model any arbitrary function using the boosting

N. da Vitora Lobo et al. (Eds.): SSPR&SPR 2008, LNCS 5342, pp. 714–723, 2008.

methodology at different resolutions of the data. In the scale-space based approach to statistical additive modeling, the weak learners are determined by analyzing the data at different resolutions. Our algorithm provides the flexibility of choosing the weak learners dynamically which will improve the modeling performance compared to using the static learners. For every learner in the additive model, the resolution is either maintained or doubled and a weak learner is modeled for fitting the data. To begin with, complete input data is considered for fitting a Gaussian model. As the iterations progress, the number of data points to be considered for fitting is reduced and a Gaussian model is fit to the data for every boosting iteration. We obtain an optimal Gaussian model that can fit the given number of data points at that particular resolution of interest in that iteration.

The scale-space concept allows for effective modeling of the dataset at any given resolution [6]. Scale-space based techniques have also been widely used for effective clustering [5]. In terms of analyzing a dataset at different resolutions, our approach closely resembles wavelet decomposition techniques which are effective tools in the field of multi-resolution signal analysis [4]. The main contributions of our work is: *Multi-resolution additive modeling by fitting the data using the concepts of scale-space theory.* This is an extended version of our previous work on scale-space based regression [7]. Though our method can be potentially applied with any base (or weak) learner, we chose to have Gaussian model because of its nice theoretical properties in the scale-space framework [8]. The rest of the paper is organized as follows: Section 2 shows the problem formulation in detail and discusses the concepts necessary to comprehend our algorithm. Section 3 describes our scale-space based additive modeling framework. Section 4 gives the experimental results of our algorithm on both synthetic and real datasets. Finally, Section 5 concludes our discussion with future research directions.

2 Preliminaries

This section gives details on boosting algorithm, introduces scale-space kernels and wavelet decomposition methods.

2.1 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. The goal of a regression problem is to obtain the function $F(\mathcal{X})$ that can approximate $\mathcal{Y}$. We chose to demonstrate the power of scale-space kernels in the context of Logitboost algorithm because of its popularity and the power of additive modeling. Let us denote $p(x)$ is an observation for $\hat{P}[y = 1|X = x]$. Algorithm 1 gives generic Logitboost algorithm to solve two class classification problem.

Algorithm 1. Logitboost Algorithm

Initialize $F^{(0)}(x_i) = 0$ and probabilities $p^{(0)}(x_i) = \frac{1}{2}$, for $i = 1, 2, .., N$
for $t = 1$ to T **do**

1. Compute response and weights for all datapoints

$$w_i^{(t)} = p^{(t-1)}(x_i) \cdot \left(1 - p^{(t-1)}(x_i)\right)$$

$$z_i^{(t)} = \frac{y_i - p^{(t-1)}(x_i)}{w_i^{(t)}}$$

2. Fit a weak learner $f^{(t)}$ to the training data using weights w_i.

$$f^{(t)} = arg\min_f \sum_{i=1}^{n} w_i^{(t)} \left(z_i^{(t)} - f(x_i)\right)^2$$

3. Update the Classifier and the probabilities

$$F^{(t)}(x_i) = F^{(t)}(x_i) + \frac{1}{2} f^{(t)}(x_i)$$

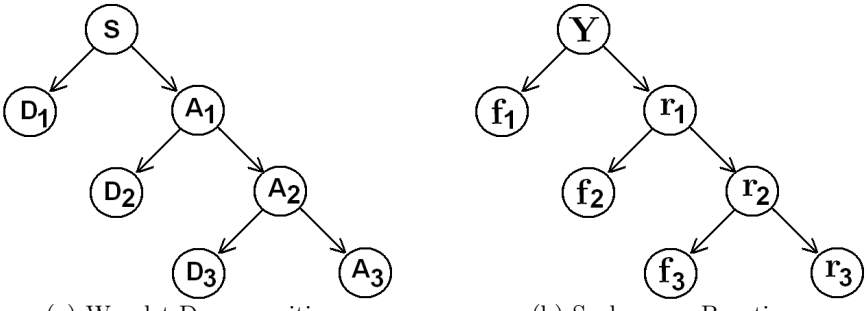
$$p^{(t)}(x_i) = \left(1 + exp(-2F^{(t)}(x_i))\right)^{-1}$$

end for
Output $C(X) = sign(F^{(t)}(x_i))$.

2.2 Scale-Space Kernels

In scale-space theory, mapping $p(x)$ is embedded into a continuous family $P(x, \sigma)$. Our method starts with an approximation of the entire dataset with Gaussian kernel of $\sigma = \infty$. 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 datapoint with a Gaussian kernel. Several practical considerations were made to make the implementation more efficient. More study on the scale-space kernels was made in the image processing literature and not much focus is given by the pattern recognition community.

Gaussian kernels are a simple and trivial choice for scale-space kernels which follow all the scale-space axioms [1]. The main reasons for choosing Gaussian kernels in the context of additive modeling as weak learners are that they are powerful universal approximators and they allow systematic hierarchical modeling. We choose to reduce the width by halves using the concepts of wavelet decomposition methods which were again well studied concepts in the context of handling image operations efficiently. Be it a classification, regression or image operation, one can consider in a generic way by treating them as a signal and the processing of this signal has to be done efficiently in order to obtain some interestingness of the signal. In this way, one can provide a unification framework



(a) Wavelet Decomposition (b) Scale-space Boosting

Fig. 1. Similarities and differences between wavelet decomposition methods and scale-space boosting. (a) Wavelet techniques decompose the signal into approximations and details. (b) Scale-space based additive modeling decomposes the function by weak learners and the corresponding residuals.

that combines several well-studied concepts in a unique manner to handle some key challenges efficiently.

2.3 Wavelet Decomposition

In signal processing applications, wavelet transformation constructs a family of hierarchically organized decompositions. At a given level j, the j-level approximation is called A_j and a deviation signal called the j-level detail D_j. A_1 corresponds to the approximation of the original signal corresponding to the low frequencies of A_0, whereas the detail D_1 corresponds to the high frequency deviation. The selection of a suitable level for the hierarchy will depend on the user needs and is usually chosen based on a desired low-pass cutoff frequency. Fig. 1 shows a comparison of wavelet decomposition with scale-space additive modeling. Successive approximations A_1, A_2 and A_3 with lower resolution of a given signal are built. The original signal S can be built by summing up all the details with the approximation at the highest frequencies. i.e. $S = A_3 + D_3 + D_2 + D_1$. Our scale-space boosting algorithm works in a similar manner. The frequencies in the wavelet domain correspond to resolutions in our scale-space based algorithm. 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 datapoints. The original target function ($\mathcal{Y}$) is decomposed using weak learners(f) and residuals(r). The final model at any given resolution is obtained by a weighted linear combination of the weak learners obtained so far. It should be noted the residual indicates the difference between the target function and the final learner (F) at that particular resolution. i.e. $r = |\mathcal{Y} - F|$.

3 Scale-Space Based Additive Modeling

Algorithm 2 describes our scale-space based approach for additive modeling. We demonstrate our approach on Logitboost algorithm, though our algorithm can be used to any additive modeling framework. The initial model is set to null or the mean value of the target values. Initially, the number of datapoints to be modeled will be the entire dataset. The feature values are first sorted (column-wise) independently which will facilitate the scale-space kernel based modeling that will be performed later on. As the iterations progress, the number of datapoints considered for fitting the weak regressor is retained or halved depending on the error of the model. For every iteration, the residual r is set to the absolute difference between the target value ($\mathcal{Y}$) and the final learner (F). The best multivariate Gaussian model will be fitted to this data at a given resolution. The details of the procedure *bestgaussianfit* which obtains the best Gaussian regressor at a particular resolution of the data is described later. The best Gaussian model corresponding to the resolution is obtained at every iteration and the model with the least error is chosen for that particular iteration.

Algorithm 2. Scale-space based 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**
  $r = |\mathcal{Y} - F|$
  $[\hat{f}_0, err_0] = bestgaussfit(\hat{\mathcal{X}}, r, N, d, n, idx)$
  $[\hat{f}_1, err_1] = bestgaussfit(\hat{\mathcal{X}}, r, N, d, n/2, idx)$
  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

Our algorithm obtains the weak regressors and models the data 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. For every iteration, the best weak Gaussian regressor is fit to the data based on a single feature value at a particular resolution. The basic idea of *bestgaussianfit* is to slide a Gaussian window across all the datapoints corresponding to each feature at a given resolution. Depending on the given resolution

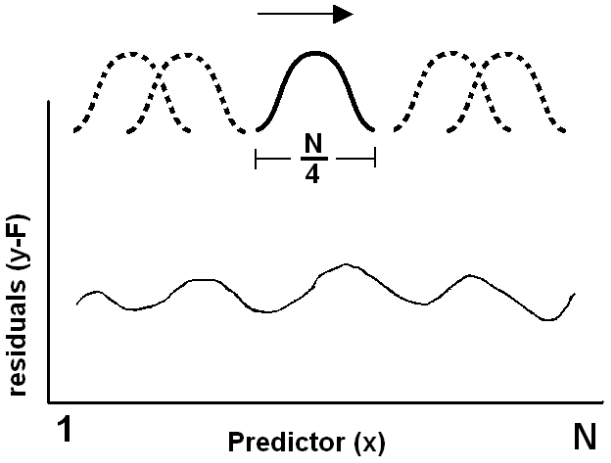


Fig. 2. Demonstration of the sliding Gaussian kernel. There are N datapoints and each has its corresponding response as a residual (y-F). A Gaussian kernel of resolution $\frac{N}{4}$ is slided across the entire range of data points to find the optimal fit. The optimal fit is obtained by minimizing the L_2 norm between the residual and the weighted Gaussian fit. The dotted Gaussians represent the sliding windows while the dashed Gaussian represent the active window.

(indicated by n datapoints), a Gaussian kernel containing n datapoints is slid across all the data points and the location where the minimal residual error is obtained. The concept of sliding kernels and active window is clearly illustrated in Fig. 2. Within each active window, the weight of individual data point is computed based on the normalized regression value.

The error is computed between the weak learner and the target regression values. It should be noted that the error is not computed using the given window alone. In other words, Gaussian kernel that will fit the data is computed based on the parameters computed within the window and the final error is computed based on the deviation in the entire dataset (not just the active window). This is a nontrivial aspect in our algorithm and failing to perform this might lead to erroneous results that might fit the data locally. This is due to the fact that every iteration, the result will be a local fit to the data and each of this weak regressor might not perform well globally. Finally, the best weak regressor across each feature that has the minimum error is returned.

The resolution is increased at a logarithmic scale. i.e. every time the resolution is doubled or 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 indicated that this change of resolution is optimal and also this logarithmic change of resolution has nice theoretical properties (they mimic wavelet decomposition methods).

3.1 Relation to Other Models

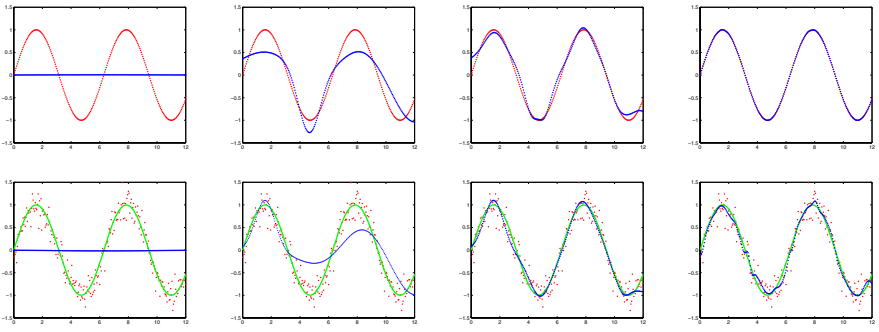
Our method appears to have close connection to popularly used Radial Basis Function (RBF) Networks [2]. The RBF networks rely on the fact that any continuous function can be approximated by a sum of appropriately chosen Gaussian functions. These networks are one form of neural networks which have a static Gaussian function for the hidden layer processing elements to attain non-linearity. The RBF networks compute the input to output map using local Gaussian approximators. During the training phase the centers and width of the Gaussians in the hidden units along with the weights of the connections is determined. In spite of their success, the RBF networks suffer from some of the popular drawbacks of neural networks like fixing the topology, including the number of hidden nodes and the active connections between the layers. Our algorithm contains very minimal user-defined parameters and doesn't suffer from these problems. It gives a systematic approach to automatically model the regression function as a linear combination of Gaussian kernels. Being complex models, the time taken for the RBF networks during the training and testing phases are very high compared to our boosting algorithm. Being powerful local approximators, RBF networks perform poorly on noisy data since the coefficients are adjusted so as to accommodate local data and these networks do not average out noise over the entire data space. Using our approach, we avoid the local approach to regression of RBF networks and at the same time benefit from other nice properties of these networks.

4 Experimental Results

All programs were written in MATLAB 7.0 and run on pentium IV 2.8 GHz machines. Experiments were performed on both synthetic and real datasets.

4.1 Synthetic Datasets

To demonstrate the robustness of the proposed scale-space boosting algorithm, we performed experiments on two synthetic regression datasets. Regression datasets were chosen to demonstrate the capability of our algorithm to obtain smooth functions. Using, boost stumps or trees, one might be able to obtain more accurate models in terms of metrics like mean square error. But, this will come at a cost of obtaining highly discontinuous models which are not desirable. Hence, we demonstrated the behaviour of algorithm on two synthetic sine wave datasets: 1) noise-free sine wave and 2) the sine wave with Gaussian noise. For the first dataset, 241 samples were generated from a sine wave (with x from 0 to 12 uniformly and $y = \sin(x)$). For the second dataset, Gaussian random noise with zero mean and 0.2 standard deviation is overlaid onto all the samples of the first dataset. Figure 3 depicts the experiment results using the two synthetic sine wave datasets: 1) noise-free sine wave in the first row and 2) sine wave with Gaussian noise in the second row. To understand the difference between the original and the final data, we overlaid the noise-free sine wave in the second



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

Fig. 3. Experimental results along with the number of iterations using two synthetic sinewave datasets. The first row is the sine wave without noise and the second row is the sine wave with Gaussian noise with μ=0 and σ=0.2. ($N(0, 0.2^2)$). The red color and blue color represent the original data and our algorithm results respectively. The green color in the second row represents the original sine wave without the Gaussian noise.

row in order to compare the system output results not only with the given noisy data but also with the original noise-free data.

4.2 Real-World Datasets

Different real-world datasets were chosen from the UCI machine learning repository [3] to demonstrate our scale-space based modeling technique. Since, the basic algorithm is for binary classification, we demonstrate the power of using scale-space kernels on datasets with 2 classes. One has to realize that the main claim of the paper is not about producing the best classification or regression algorithm. Rather, this paper tries to give a systematic understanding of different kinds of kernels that can be used in the additive modeling techniques. To the best of our knowledge, this is the first effort to combine the concepts of scale-space theory in the context of additive modeling. Additive modeling with smooth and continuous kernels will result in smooth functions for classifier boundary and regression functions. Hence, we compare the scale-space kernels with other static and dynamic kernels. 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. We used kernel width of $n/8$ for demonstrating the performance of static kernels.

The scale-space kernel doesn't suffer from the generalization problem as clearly illustrated by the results on the test data shown in Table 1. Experiments were conducted using five-fold cross validation method and the test results are reported for the algorithm using various kernels. Mean squared error is reported between both training and test cases. Scale-space kernels are competitive with the best possible kernels and can be generically be used for any dataset. Since,

Table 1. Performance of scale-space kernels with other kernels on different real-world datasets. Test error along with the standard deviation using five-fold cross validation is reported.

Dataset	Cancer	Ionosphere	Sonar	Bupa
static kernel -n/8	0.244± 0.0869	0.4118± 0.1024	0.9148± 0.1267	0.9453± 0.0607
Dynamic kernel	0.1898± 0.0338	0.3553 ± 0.063	0.7543± 0.0885	0.869 ±0.053
Scale-space kernel	0.1895 ± 0.043	0.3371± 0.092	0.7125± 0.1433	0.8962 ± 0.1291

obtaining the width of the kernel during the additive modeling process can be a challenging task, the use of scale-space kernels can resolve the problem by using adaptive step-sizes. The fact that the scale-space kernels converge much faster than static kernels make them much suitable for additive modeling algorithms. Also, one can see that the results of the scale-space kernels are fairly robust compared to other kernels.

5 Conclusions and Future Research

Recently, additive modeling techniques have received a great attention from several researchers working in a wide variety of applications in science and engineering. In this paper, we proposed a novel additive modeling framework that uses scale-space theory to obtain the optimal weak learner at every iteration. Advantages of our method compared to other popular models proposed in the literature are clearly demonstrated. As a continuation of this work, we would like to explore the idea of using other scale-space kernels (different from Gaussian) as our weak learners. Optimal choice of the n value to be chosen adaptively during each boosting iteration must be investigated more thoroughly depending on individual dataset to be modeled. More theoretical insights related to combination of additive models and scale-space kernels are yet to be investigated.

References

1. Babaud, J., Witkin, A.P., Baudin, M., Duda, R.O.: Uniqueness of the gaussian kernel for scale-space filtering. IEEE Transactions on Pattern Analysis and Machine Intelligence 8(1), 26–33 (1986)
2. Bishop, C.M.: Neural Networks for Pattern Recognition. Oxford University Press, Oxford (1995)
3. 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`
4. Graps, A.L.: An introduction to wavelets. IEEE Computational Sciences and Engineering 2(2), 50–61 (1995)
5. Leung, Y., Zhang, J., Xu, Z.: Clustering by scale-space filtering. IEEE Transactions on Pattern Analysis Machine Intelligence 22(12), 1396–1410 (2000)

6. Lindeberg, T.: Scale-Space Theory in Computer Vision. Kluwer Academic Publishers, Dordrecht (1994)
7. Park, J.-H., Reddy, C.K.: Scale-space based boosting for weak regressors. 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)
8. Sporring, J., Nielsen, M., Florack, L., Johansen, P.: Gaussian Scale-Space Theory. Kluwer Academic Publishers, Dordrecht (1997)