LA-UR- 03-0545

Approved for public release; distribution is unlimited.

Title:	Weighted Order Statistic Classifiers with Large Rank-Order Margin
Author(s):	Reid Porter, Don Hush, James Theiler, Maya Gokhale
Submitted to:	International Conference on Machine Learning

Form 836 (8/00)

Weighted Order Statistic Classifiers with Large Rank-Order Margin

Reid Porter RPORTER@LANL.GOV
Don Hush[1] DHUSH@LANL.GOV
James Theiler JT@LANL.GOV
Maya Gokhale MAYA@LANL.GOV
Nonproliferation and International Security Division,
[1] Computer and Computational Sciences Division,
Los Alamos National Laboratory, NM 87545, USA

Abstract

We describe how Stack Filters and Weighted Order Statistic function classes can be used for classification problems. This leads to a new design criteria for linear classifiers when inputs are binary-valued and weights are positive. We present a rank-based measure of margin that can be directly optimized as a standard linear program and investigate its effect on generalization error with experiment. Our approach can robustly combine large numbers of base hypothesis and easily implement known priors through regularization.

1. Introduction

Generalized median filters are a continuing topic of interest in image and signal processing. Their robustness to outliers has made them a popular alternative in applications where linear filters perform poorly. An intuitive understanding of how median-type filters can be used for classification is gained through analogy. The linear classifier model, or finite impulse response filter, can be considered a direct generalization of the weighted average and sample mean. In a similar way, the median filter can be generalized to weighted median and weighted order statistic function classes. A more detailed discussion of this analogy can be found in (Arce 1998).

Our analysis and method of optimization are based on properties known as threshold decomposition and stacking. This approach has been used to successfully design generalized median filters for signal enhancement and noise suppression problems. Our novel contribution is application of threshold decomposition and stacking to large margin classification. We discuss the relevant literature in Section 2.

A number of non-linear classifiers have been suggested that are related to generalized median function classes, including min-max classifiers (Yang and Maragos 1995), fuzzy min-max classifiers (Simpson 1992) and morphological networks (Ritter and Sussner 1996), (Sussner 1998). However, in all cases threshold decomposition and stacking were not discussed.

A second interpretation for generalized median classifiers comes from the ensemble approach to machine learning. The median, or majority vote is used by bagging (Breiman 1996). In Section 4 we show how the weighted order statistic function class is equivalent to a linear classifier with binary inputs and positive weights. This method of combining ensembles is used by the adaboost m.1 algorithm (Schapire, Freund et al. 1998). Direct optimization of margin for this function class has also been considered by a number of authors (Mason, Bartlett et al. 1999), (Grove and Schuurmans 1998). However, the relationship to weighted order statistics was not considered. This relationship leads us to a significantly different design criteria, and an efficient method of optimization.

The key concepts which extend threshold decomposition and stacking to classification problems are described in Section 3. This includes a new measure of margin based on rank-order. Our results are applicable to a large number of ordered function classes including weighted order statistics, stack filters and all digital mappings: $\{0,1\}^D \rightarrow \{0,1\}$ (see Section 6).

In Section 4, we describe practical methods of optimization for the weighted order statistic function class. We show how the rank-order margin can be directly optimized as a standard linear program. We investigate the relationship between rank-order margin and generalization with 4 two-class classification problems in Section 5. We summarize and suggest future research efforts in Section 6.

2. Background

2.1 Threshold Decomposition

Threshold decomposition is a useful tool for understanding a large family of non-linear functions (Fitch, Coyle et al. 1984). We describe threshold decomposition for D inputs: $\vec{X} = [X_1, X_2, \ldots, X_D]$, where each input takes on values from a finite set of $2M+1$ quantization levels: $X_i \in Q \triangleq \{-M, \ldots -1, 0, 1, 2, \ldots, M\}$. The concept is readily extended to real-valued inputs (L.Yin and Neuvo 1994), (Arce 1998). We define a threshold function as:

$$T^m(X) = \begin{cases} 1, \; if \; X \geq m \\ 0, if \; X < m \end{cases} \tag{1}$$

Threshold decomposition of $\vec{X}$ produces a set of $2M$ binary vectors $\left[\vec{x}^{-M+1}, \ldots, \vec{x}^{0}, \ldots \vec{x}^{M}\right]$ where $\vec{x}^m = T^m(\vec{X})$. To clarify our notation, the i^{th} element of $\vec{x}^m$ is $x_i^m = T^m(X_i)$. The original inputs can be exactly recovered from this decomposition by:

$$X_i = -M + \sum_{m=-M+1}^{M} x_i^m \tag{2}$$

2.2 Stack Filters

If we define a binary function $f : \{0,1\}^D \rightarrow \{0,1\}$ operating on each binary input vector $\vec{x}^m$, then the function:

$$S_f(\vec{X}) = \sum_{m=-M+1}^{M} f(\vec{x}^m) \tag{3}$$

defines a stack filter when $f(\vec{x}^m)$ possesses the stacking property. The stacking property requires $f(\vec{x}^m) \leq f(\vec{x}^l)$ whenever $x_i^m \leq x_i^l$ for all i. A necessary and sufficient condition for $f(\bullet)$ to possess the stacking property, is that, it is a Positive Boolean Function (PBF) (Wendt, Coyle et al. 1986). PBFs are the subset of Boolean functions that can be expressed without complements of the input variables. Figure 1 illustrates the stack filter architecture, and corresponding PBF for a three input median function.

2.3 Weighted Order Statistics

If we restrict $f(\bullet)$ within the stack filter architecture (3) to a linearly separable PBF, the sub-class of Weighted Order Statistics (WOS) is defined. A linearly separable PBF can be expressed as:

$$f(x_1, \ldots, x_D) = T^0\left(\sum_{i=1}^{D} W_i x_i - R\right) \tag{4}$$

where $W_i, R \in \mathbb{R}$, $W_i, R \geq 0$ and $x_i \in \{0,1\}$ for all i. We will refer to this model as a Positive weighted Binary Perceptron (PBP). This class has an equivalent integer domain interpretation:

$$Wos_f(\vec{X}) = R^{th}\text{largest}\{W_1 \Diamond X_1, W_2 \Diamond X_2, \ldots W_D \Diamond X_D\} \tag{5}$$

where $W_i \Diamond X_i = \overbrace{X_i, X_i, \ldots, X_i}^{W_i \; times}$. By choosing weights W_i and threshold R, median, weighted median, order statistic and the weighted order statistic function classes can be implemented (Yli-Harja, Astola et al. 1991). Pseudo-code for calculating a real-valued WOS can be found in (Arce 1998). In figure 1, the PBP for a 3 input median function has $\forall W_i = 1$ and $R = 1.5$.

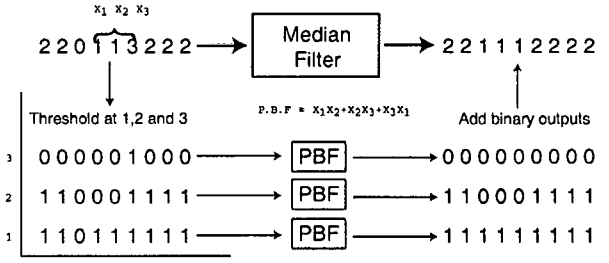


Figure 1. Threshold decomposition architecture for a 3-input median filter.

2.4 Mean Absolute Error

Stack filter optimization can be posed as a linear program for Mean Absolute Error (MAE) loss functions (E.J.Coyle 1988). Consider the stack filter (3) with input $\vec{X} \in Q^D$ and desired output $Y \in Q$:

$$L_{MAE}(S_f) = E\left[\left|Y - S_f(\vec{X})\right|\right] \tag{6}$$

where $E[\;]$ is the expected value over the joint distribution $\vec{X} \times Y$.

$$L_{MAE}(S_f) = E\left[\left|\sum_{m=-M+1}^{M} \left(y^m - f(\vec{x}^m)\right)\right|\right] \tag{7}$$

$$L_{MAE}(S_f) = \sum_{m=-M+1}^{M} E\left[\left|y^m - f(\vec{x}^m)\right|\right] \tag{8}$$

Equation (7) follows from the application of threshold decomposition. By the stacking property, all nonzero errors within the sum of (7) are the same sign leading to (8). An intuitive interpretation of the optimization problem was presented by Coyle (E.J.Coyle 1988): At each level m, the function $f(\bullet)$ observes a binary input

vector $\{0,1\}^D$ and must decide, by outputting a one or zero, whether the desired output is greater than or equal to m. The optimization problem is equivalent to completing the 2^D row truth table, which defines $f(\bullet)$, subject to the stacking constraints.

3. Stack Filter Classification

3.1 Mirrored Input Space

For classification problems, the desired signal is $Y \in \{-1,1\}$. We will threshold the stack filter model at 0 to produce the desired $\{-1,1\}$output. That is, $S_f(\bullet) \geq 0$ for $Y = 1$ and $S_f(\bullet) < 0$ for $Y = -1$. How well can a stack filter be expected to solve this type of problem? It is important to realize that the filter output is always equal to one of the filter inputs. This suggests that it would perform very badly since if $X_i >= 0$ for all *i*, then the stack filter must assign the sample to class $Y = 1$.

This *one-sided* limitation can be overcome by supplementing the inputs to the function with mirrored input samples: $\vec{X}' = [X_1,...,X_D,-X_1,...,-X_D]$. This approach was suggested within the context of a mirrored threshold decomposition architecture by (Paredes and Arce 2001). They demonstrate stack filters with band-pass filtering capabilities. They point out that a stack filter, without mirrored inputs, is limited to low-pass behavior. This result is also observed in mathematical morphology, and many algorithms have both foreground and background components (Serra 1982).

For classification, the mirrored input space provides an absolute reference point for the stack filter at 0. That is, the median filter, applied to the mirrored input space will always output 0 (for sake of argument we assume the space is also augmented with the constant 0 term). An alternative interpretation, suggested by the weak-ordering property of stack filters around the median (Lin, III et al. 1994), is that the hard decision boundary at 0 has been replaced with a relative boundary defined by the median. For the rest of this paper we assume that the mirrored input space is used and $\vec{X} = \vec{X}'$.

3.2 Data Dependent Threshold Decomposition

In the threshold decomposition architecture presented, the threshold levels must be specified. The stack filter output is always one of the input samples, and therefore it is possible to define the threshold levels by the input samples without loss of generality. Let $SI_f(\vec{X})$ be a stack index filter defined as:

$$SI_f(\vec{X}) = \sum_{m=1}^{2D} f(\vec{x}^{X^{mth}}) \tag{9}$$

where the notation X^{mth} is the mth smallest sample in the sorted mirrored input set $\vec{X} = [X_1,...,X_D,-X_1,...,-X_D]$. To clarify our notation, recall that:

$$x_i^{X^{mth}} = T^{X^{mth}}(X_i) = \begin{cases} 1, \text{ if } X_i \geq X^{mth} \\ 0, \text{if } X_i < X^{mth} \end{cases}$$

There is a one-to-one correspondence between a Stack Index filter and a Stack filter:

$$S_f(\vec{X}) = X^{[SI_f(\vec{X})]th} \tag{10}$$

With this approach we have a finite set of *2D*, scale invariant, threshold levels to consider.

3.3 Large Margin Loss Functions

A useful concept from machine learning is large margin classification. With this technique we consider not just misclassification error, but also the distance from the decision boundary. A number of theorems from the Probably Approximately Correct machine-learning framework, bound generalization error in terms of margin, for a number of function classes. These theorems generally take the following form, which we simplify from (Schapire, Freund et al. 1998):

Theorem 1: Let S be a sample of N examples chosen independently at random according to the distribution D over $X \times Y$. Then for $\delta > 0$, with probability $1-\delta$ over the random choice of the training set S, every function $h(\bullet)$in a hypothesis class $\mathcal{H}$ satisfies the following bound for all $\gamma > 0$:

$$P_D[yh(x) \leq 0] \leq P_S[yh(x) \leq \gamma] + \varepsilon(\delta, N, |\mathcal{H}|, \gamma)$$

where $P_D[A]$ and $P_S[A]$ are the probabilities of event A when an example *(x,y)* is chosen from D and S respectively.

Here γ is the margin and increasing γ reduces ε, but increases P_S.

3.4 Rank-Order Margin

We now describe large margin classification for the stack filter function class (3). Consider misclassification loss, where we simply count the number of mistakes:

$$L_M(S_f) = E\left[T^0\left(S_f(\bullet)\right) \neq \frac{1}{2}(Y+1)\right] \tag{11}$$

By the stacking property, this reduces to misclassification loss at level 0 of the threshold decomposition architecture:

$$L_M(S_f) = E\left[f^0(\bullet) \neq \frac{1}{2}(Y+1)\right] \tag{12}$$

We define misclassification loss at margin γ for the Stack filter as:

$$L_M^{\gamma}(S_f) = E\left[T^{\gamma}\left(S_f(\bullet)\right) \neq \frac{1}{2}(Y+1)\right] \tag{13}$$

To specify γ we must specify threshold levels $Q \triangleq \{-M,...-1,0,1,2,...,M\}$. A convenient choice is the finite set $\{1,2,..,2D\}$ defined in the stack index function class (9). For the discrete distribution based on samples, γ-margin misclassification loss is defined as:

$$\hat{L}_M^{\gamma}(S_f) = \sum_{\forall Y=-1} f(\vec{x}^{\,X^{[D-\gamma+2]/h}}) + \sum_{\forall Y=1} 1 - f(\vec{x}^{\,X^{[D+\gamma]/h}}) \tag{14}$$

where γ has integer values $\gamma \in [1,2,\ldots,D]$. This loss function is illustrated in Figure 2.

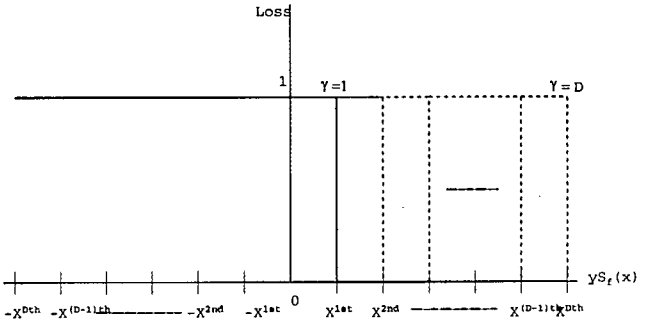


Figure 2. Rank-order margin loss functions.

In Figure 3, we illustrate rank-order margin with an example. The mirrored input sample $\vec{X}$ belongs to class $Y=1$. A particular stack filter $S_f(\vec{X})$ produces a correct classification, $S_f(\vec{X}) \geq 0$, when it outputs either 1 or 3. The corresponding margins are $\gamma=1$ and $\gamma=3$ respectively.

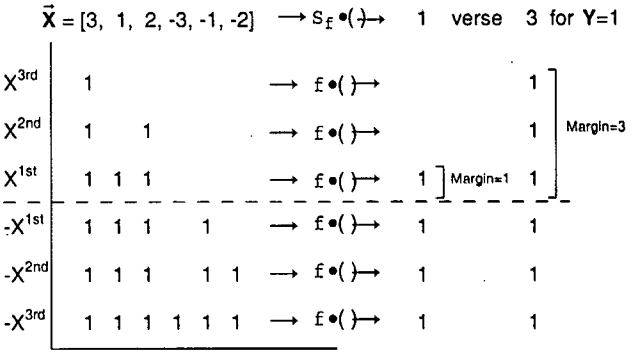


Figure 3: Example of rank-order margin.

4. Weighted Order Statistic Classification

The techniques of Section 3 were developed for the Stack Filter function class. The optimization problem in this case requires $\mathcal{O}\left(2^D\right)$ variables. The WOS sub-class can be represented with 2D+1 variables, and therefore is computationally less demanding and analogous to linear classifiers. For the remainder of the paper we investigate this sub-class.

For classification, the stacking property links the WOS integer domain interpretation (5) to a binary-input linear classifier with positive weights (PBP). That is, once thresholded at 0, a WOS function class is equivalent to the PBP function class. Direct optimization of margin for the PBP function class was described as NP hard in (Mason, Bartlett et al. 1999). The WOS interpretation has led us to a rank-order margin, and a tractable loss function. From (4) and (14):

$$\hat{L}_M^{\gamma}(S_f) = \sum_{\forall Y=-1} 1 - T^0(\vec{W}.\vec{z}^{\,X^{[D-\gamma+2]/h}}) + \sum_{\forall Y=1} 1 - T^0(\vec{W}.\vec{z}^{\,X^{[D+\gamma]/h}}) \tag{15}$$

where $\vec{z}^{\,m}$ is the augmented and transformed input $[Y\vec{x}^{\,m}\ Y]$ and $\vec{W}^T = [W_1,\ldots,W_{2D},R]$ is from (4). This binary-input linear classifier design problem is also subject to the additional stacking constraints $W_i, R \geq 0$ for all i. There are a number of algorithms used to solve (15). We optimize the perceptron criteria, with a linear program, that also includes stacking constraints and an opportunity for L1 regularization:

$$\min_{\tau,w} \sum_{i=1}^{N'} \tau_i + \alpha \sum_{i=1}^{2D} W_i$$
$$subject\ to \qquad (16)$$
$$\tau_i \geq 0 \ and \ \tau_i \geq b_i - \vec{W}.\vec{z}_i$$
$$and \quad W_i, R \geq 0 \ for\ all\ i$$

The positive constants b_i are required to avoid a trivial solution. In all experiments we used a value of 1. The linear program has $N+2D+1$ variables where N is the number of training pairs, and D is the original input space. In our experiments we investigate relationships between rank-order margin (15) and generalization. We found $\alpha = 0$ caused problems for our LP solver and we used $\alpha = 1/4D$.

5. Experiments

5.1 Basis Expansion

Non-linear basis expansion is often used to increase the flexibility of linear models for classification and regression. The WOS classifier is inherently non-linear and it is therefore possible to consider linear basis expansion of input variables. WOS non-linearity results from the hard-limit thresholding of input variables. This suggests an alternative perspective, where the WOS classifier is used to combine ensembles of simple classifiers with binary outputs.

In the experiments that follow, we implement non-linear basis expansion using data centered spheres:

We have a N sample training set $\{\vec{X}^1, Y^1, \ldots, \vec{X}^N, Y^N\}$, where each $\vec{X}^i = [X_1, X_2, \ldots, X_D]$ and Y is the associated class label: $X_i \in \mathbb{R}$, $Y \in \{-1, 1\}$. A new pattern $\vec{X}$ is mapped to an N dimensional vector $\vec{M} = [M_1, M_2, \ldots, M_N]$, $M_i \in \mathbb{R}$, defined by:

$$M_i = \left\| \vec{X} - \vec{X}^i \right\| - t_i \qquad (17)$$

for $i = 1 \ldots N$ and $\|\bullet\|$ is the Euclidean distance.

The threshold t_i is a free parameter. We perform two experiments, and t_i is treated in two different ways.

5.2 Maximizing Margin in Boosted Ensembles

Our first experiment is similar to one used to investigate DOOM (Mason, Bartlett et al. 1999). We chose 4 two-class data sets from the UCI Machine Learning Repository (Blake and Merz 1998). Each problem was divided randomly into training and test sets of pre-specified size summarized in Table 1. Each experiment was performed 25 times and the results were averaged.

Table 1. Training and test set sizes for the 4 data sets.

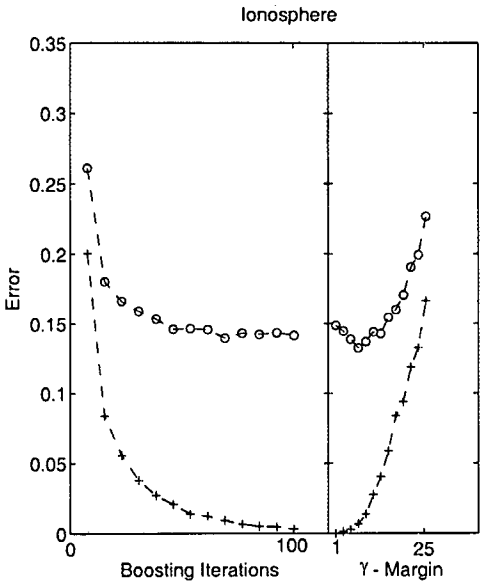
DATA SET	TRAINING	TESTING
IONOSPHERE	100	250
SONAR	58	150
BREAST CANCER	182	512
HEART DISEASE	96	200

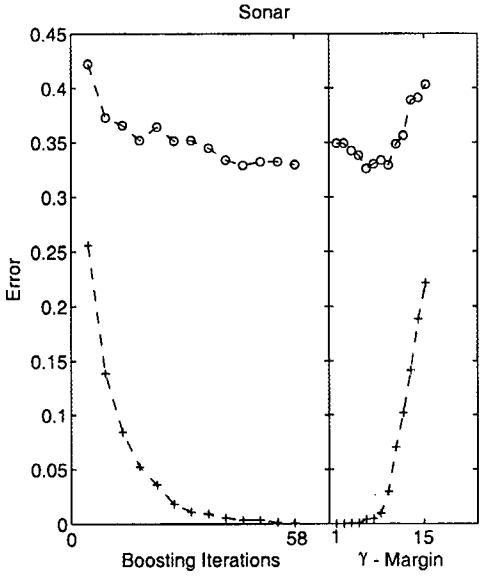
We used the AdaBoost m.1 algorithm to incrementally construct base hypothesis and PBP model. The *weak learner* selects, through exhaustive search, M_i from $\vec{M}$ and the associated threshold t_i that will minimize training error. For convenience, we chose the number of boosting iterations to be equal to the number of training points. The training and test errors are shown on the left hand side of plots in Figure 4.

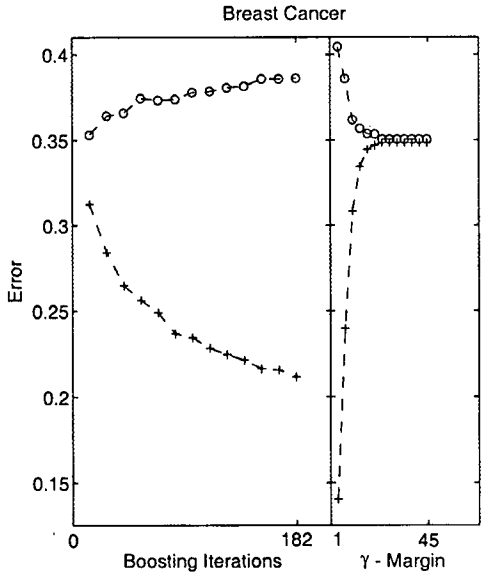
By the final iteration, this procedure produced models of dimension equal to the training column in Table 1. We augmented this model with mirrored (or negative) input samples as described in Section 3.1. This doubles the model dimension. We then re-optimized the PBP weights using a linear program (16) at specific values of margin, γ according to (15). To keep computation times reasonable we only evaluated γ between 1 and 25% of the original input dimension D. The training and test errors for the WOS classifier are shown on the right hand side of plots in Figure 4.

5.3 Discussion

γ-margin curves appear consistent with the hypothesis that rank-order margin is related to generalization error. WOS optimization at 0-margin produces *equal or smaller* training error and *equal or greater* test errors compared to Boosting. As we increase margin γ, we observe decreasing test errors and increasing training errors. This trend continues until (presumably) training error becomes the dominant contributor to test error. In all 4 problems, the WOS classifier achieves *equal or smaller* test error than at any point in the Boosting procedure.







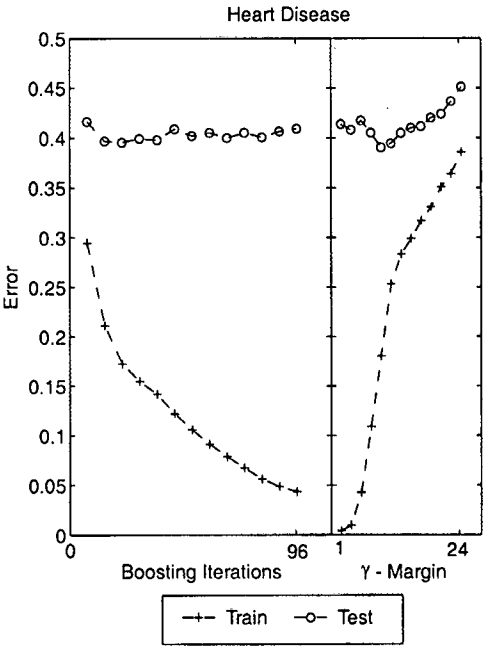


Figure 4. Training and test errors found by (left side) boosting data centered spheres and (right side) applying WOS classifier to final iteration at increasing margin.

The Breast Cancer result is significantly different from the other 3 experiments. We observe that Boosting achieves the best test error of 0.356 on the first iteration. By iteration 182 the boosted ensemble shows significant over-fitting. At *0*-margin, the WOS classifier also over-fits the 182 base hypotheses. However, as margin is increased, the WOS classifier approach can sacrifice substantial training error and reduce test error to 0.350.

5.4 Fixed Basis and Parameter Regularization

The constructive approach used by Boosting is one way to choose relevant binary base hypothesis. However, it is likely that the M_i and t_i selected by Boosting are not optimal for the WOS classifier. An alternative is to provide the entire set of base hypothesis considered by the weak learner as input. This requires us to provide all possible values of t_i, increasing the input dimension to $\mathcal{O}\left(N^2\right)$.

In our second experiment, we used the same data sets as experiment 1 and divided training and test sets according to Table 1. We also used the same data centered spheres, but randomly selected a subset of 25 training points. The dimension of M was therefore reduced from N, used previously, to 25. For each M_i we then used the same 25 training points, after projection (17), to calculate 24 values of t_i (we used the average of consecutive neighbors). This resulted in a 600 dimensional problem (1200 mirrored input space) for all experiments.

We applied the WOS classifier with increasing values of rank-order margin γ. Training and test errors are shown in Figure 5. After each optimization, we also calculated the L1 regularization term from (16). This value is also

shown in Figure 5. We do not provide the axis (and hence magnitude) for this measurement.

5.5 Discussion of Results

In all problems we observe very similar performance to experiment 1, suggesting the WOS classifier is scalable to high dimensional problems. To investigate this further (e.g. 100 training points and a 9900 dimension expansion) is a topic of future work and computational requirements will demand a more careful implementation.

In 3 of the 4 experiments we observe a monotonic increase in the L1 norm as margin increases and test error decreases. This indicates maximizing rank-order margin and parameter regularization will affect the classifier model in different ways. For the breast cancer problem there is a sharp decrease in the L1 norm. Note that the boosting result for this problem did not improve after the first iteration. We propose the explanation is therefore problem specific. That is, the combination of data centered spheres and PBP function class, is poorly matched to the problem, and produces a large number of noisy, or irrelevant, input dimensions.

6. Summary and Future Work

We have a presented a new approach to large-margin classification for the PBP function class. We developed the concept of rank-order margin and found its relationship to generalization performance was supported by experiment. Rank-order margin has two significant properties:

- The finite set of γ-margin loss functions can be directly optimized as a standard linear program.
- Margin maximization appears independent of parameter regularization mechanisms.

Our experiments have centered on the Weighted Order Statistic function class. However, our approach is immediately applicable to larger (and richer) function classes. Similar to (4), a family of non-linearly separable PBFs can be expressed as polynomial functions of input variables (Yli-Harja, Astola et al. 1991). This is analogous to quadratic and higher-order polynomials that extend linear classifiers.

Generalized Stack Filters (Lin and Coyle 1990) extend Stack Filters by supplying $f(\bullet)$ with inputs from multiple levels (and possibly all levels) of the threshold decomposition architecture. We used this approach, in experiment 2, when we applied multiple constants t_i to input variables. A second extension allows the function $f(\bullet)$ to vary from one level to the next subject to stacking constraints described in (Lin and Coyle 1990). Lin and Coyle (1990) also show that for any digital function $g(\bullet): Q^D \rightarrow Q$ there exists a Generalized Stack Filter, $GS_f(\bullet)$, such that $GS_f(\bullet) = g(\bullet)$.

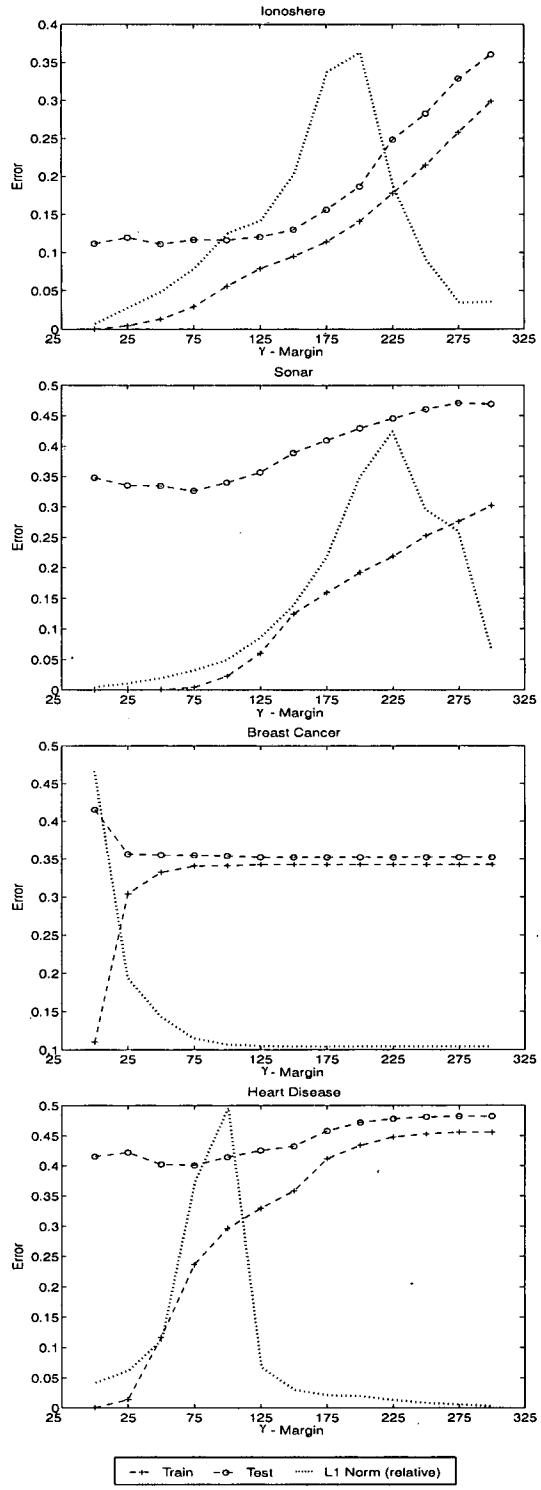


Figure 5. Training, test errors and L1 Norm as rank-order margin is increased.

A problem with using larger function classes will be computation time. For example, the stack filter

optimization problem for D inputs, is $\mathcal{O}\left(2^D\right)$. For Generalized Stack Filters the problem is $\mathcal{O}\left(N^{D+1}\right)$ (Lin and Coyle 1990) (we assume threshold levels are defined by N data points). A potential solution to this large optimization problem is stochastic descent type algorithms suggested by (Yoo, Fong et al. 1999).

Acknowledgements

This work was supported by the Department Of Energy's Deployable Adaptive Processing initiative and Los Alamos National Laboratory's Real World Machine Learning directed research program. We gratefully acknowledge Damien Eads, Neal Harvey and Simon Perkins for fruitful discussions and programming support.

References

Arce, G. R. (1998). "A General Weighted Median Filter Structure Admitting Negative Weights." *IEEE Trans. Signal Processing* 46(12).

Blake, C. L. and C. J. Merz (1998). *UCI Repository of machine learning databases*, University of California, Irvine, Dept. of Information and Computer Sciences.

Breiman, L. (1996). "Bagging predictors." *Machine Learning* 24(2): 123-140.

E.J.Coyle (1988). "Stack Filters and the Mean Absolute Error Criterion." *IEEE Trans. Acoust., Speech, Signal Processing* 36(8): 1244-1254.

Fitch, J. P., E. J. Coyle, et al. (1984). "Median filtering by threshold decomposition." *IEEE Trans. Acoust., Speech, Signal Processing* ASSP-32.

Grove, A. J. and D. Schuurmans (1998). *Boostin in the limit: Maximizing the margin of learned ensembles*. AAAI-98: Proceedings of the Fifteenth National Conference on Artifical Intelligence, Madison, WI.

L.Yin and Y. Neuvo (1994). "Fast adaption and performance characteristics of fir-wos hybrid filters." *IEEE Trans. Signal Processing* 42.

Lin, J. and E. J. Coyle (1990). "Minimum Mean Absolute Error Estimation over the Class of Generalized Stack Filters." *IEEE Transactions on Acoustics, Speech and Signal Procesing* 38(4): 663-678.

Lin, L., G. B. A. III, et al. (1994). "Stack filter lattices." *Signal Processing* 38: 277-297.

Mason, L., P. Bartlett, et al. (1999). "Improved Generalization through Explicit Optimization of Margins." *Machine Learning* 0(1-11).

Paredes, J. L. and G. R. Arce (2001). "Optimization of Stack Filters based on Mirrrored Threshold Decomposition." *IEEE Transactions on Signal Processing* 49(6).

Ritter, G. X. and P. Sussner (1996). *An introduction to morphological neural networks*. 13th International Conference on Pattern Recognition, Vienna, Austria.

Schapire, R. E., Y. Freund, et al. (1998). "Boosting the margin: a new explanation for the effectiveness of voting methods." *Annals of Statistics* 26(5).

Serra, J. (1982). Image Analysis and Mathematical Morphology. New York, Academic Press.

Simpson, P. K. (1992). "Fuzzy min-max neural networks -Part 1: Classification." *IEEE Transactions on Neural Networks* 3: 776-786.

Sussner, P. (1998). *Morphological Perceptron Learning*. Joint Conference on the Science and Technology of Intelligent Systems, Maryland, IEEE.

Wendt, P. D., E. J. Coyle, et al. (1986). "Stack Filters." *IEEE Transactions on Acoustics, Speech and Signal Procesing* 34(4): 898-911.

Yang, P. and P. Maragos (1995). "Min-Max Classifiers: Learnability, Design and Application." *Pattern Recognition* 28(6): 879-899.

Yli-Harja, O., J. Astola, et al. (1991). "Analysis of the Properties of Median and Weighted Median Filters Using Threshold Logic and Stack Filter Representation." *IEEE Transactions on Signal Processing* 39(2): 395-410.

Yoo, J., K. L. Fong, et al. (1999). "A Fast Algorithm for Designing Stack Filters." *IEEE Transactions on Image Processing* 8(8): 1014-1028.