

Spectral Partitioning for Boundary Estimation

Terry Windeatt

Centre for Vision, Speech and Signal Proc., School of EE, IT & Maths,
University of Surrey, Guildford, Surrey, United Kingdom GU2 5XH
t.windeatt@surrey.ac.uk

Abstract

We propose a spectral technique for analysing intermediate feature space of multiple classifier decisions, which enables a separable subset of patterns to be extracted. The method is applied to finding a set of patterns that are inconsistently classified, a random subset of which is left out of the training set of each expert in a multiple classifier framework.

Introduction

Achieving optimal performance for a pattern recognition system is not necessarily consistent with obtaining the best performance for a single classifier; indeed, combining multiple classifiers may lead to improved generalisation performance compared with any constituent classifier. However certain conditions need to be satisfied to realise the performance improvement, in particular that the individual (base) classifiers be not too highly correlated. Various techniques have been devised to reduce correlation between classifiers before combining, including: (i) reducing dimension of training set to give different feature sets, (ii) incorporating different types of base classifier, (iii) designing base classifiers with different parameters for same type of classifier, and (iv) resampling training set so each classifier is specialised on different subset [Kit97] [Win99]. In this paper, we propose a spectral technique for characterising correlation of multiple classifier decisions so that a maximally separable subset can be extracted from the training set. We present results for correlation reduction technique (iv), which uses different training sets. Training on subsets appears to work well for unstable classifiers, such as neural networks and decision trees, in which a small perturbation in the training set may lead to a significant change in constructed classifier. It is this instability property that we exploit here and investigate performance improvement for combining multiple MLP networks on partitioned subsets.

Training set perturbations and Combining

Effective methods for improving unstable predictors based on perturbing the training set prior to combining, include Bagging [Bre96] and Boosting [Fre97]. Bagging and Boosting are both voting algorithms and achieve impressive results for some problems by combining hard-level classifications. By hard-level we mean that the combination is taken after the final decision of each classifier is taken, in contrast with soft-level combination which uses the outputs of each classifier prior to taking individual decisions. Bagging (from Bootstrap Aggregating) forms replicate training sets by sampling with replacement, and combines the resultant classifications with a simple majority vote. Boosting, which combines with a fixed weighted vote is more complex than Bagging in that the distribution of the training set is adaptively changed based upon the performance of sequentially constructed classifiers. Each new expert is used to adaptively filter and re-weight the training set, so that the next expert in the sequence has increased probability of selecting patterns that have been previously misclassified. The performance improvement in Boosting is normally characterised as turning a series of *weak learners* into a *strong learner* [Fre97]. Some studies have compared Boosting and Bagging and noted that, while Boosting outperforms Bagging for many problems, in some circumstances Boosting actually increases the classification error. One possible reason for the difference is the fact that Boosting does not appear to handle noise well [Rat98] [Qui96].

In our approach to designing multiple classifiers, we use a different form of filtering from that used in Boosting, in which each sequential expert divides the training data into correctly and incorrectly classified sets. In contrast, we partition the training data into maximally separable subsets based on hard-level decisions of all constructed experts considered in parallel. In this paper, we are not concerned with incorporating the filtering technique into Boosting methods. Here we consider noisy problems and we assume that the base classifier is a *strong learner* so we suspect that conventional Boosting-type methods are

unable to improve performance. We estimate patterns near the optimal boundary after extracting subsets, and construct an approximate perceptron network for combining intermediate features. The procedure is recursive with random fraction of previous estimate of boundary patterns left out of the training set of each individual classifier. The update method and combining technique may be viewed as a form of regularisation that allows for mistrust in the data to reduce overfitting. This viewpoint is taken in [Rat98], in which Adaboost is characterised as a simulated annealing process, and the cost function modified to improve regularisation.

In this paper, we restrict ourselves to the 2-class case, and since the hard-level intermediate feature space is binary we consider a spectral representation of a binary-to-binary mapping. However, this is not really a restriction; the background behind our approach is similar to the error-correcting code scheme in [Die95], and it should be straightforward to use such codes to handle the multi-class case.

Spectral approach to partitioning

We first describe a modification to the proposed partitioning algorithm which handles noisy and possibly contradictory information in the spectral domain, and incorporates a check for separability of a subset. We use this method with hard-level classifications in intermediate feature space to partition and resample the training set. Spectral content of the resulting partition is also used to construct a perceptron network that is an alternative to combining by weighted vote. Our partitioning method is based on the Sequential Learning Algorithm [Mar90], which is an exact technique and guaranteed to learn an arbitrary Boolean function with a single hidden layer perceptron network.

Our measure of separability is based on the concept of k -monotonicity which comes from threshold logic theory and provides necessary and increasingly sufficient conditions for separability of a Boolean data set. We derive separability constraints by enhancing the representation of a Boolean function as follows [Win97]. For a completely specified Boolean function we attach a second binary value to each component that we call sensitivity (σ), and for convenience we label the j th component $x_j^{\sigma_j}$. To find $x_j^{\sigma_j}$ from a truth table, compare input patterns that are unit Hamming Distance (D_H) apart and assign σ_j according to whether the target response differs. We therefore obtain some structural information about network implementation, namely whether a change in binary value of x_j gives rise to a change in target value. This information is implicit in the original representation, but by making it explicit and

interpreting it as excitatory and inhibitory spectral contributions, we can derive separability constraints for a subset of patterns. In the following description, we assume binary values 1 and 0, and a simple modification of the formulae is required for other coding schemes such as +1 and -1.

Rule for assigning σ_j for completely specified case

For all X_1, X_2 such that $\{ |X_1 \oplus X_2| = 1, x_{1j} \neq x_{2j} \}$

$$\text{Assign } \sigma_{1j} = \sigma_{2j} = |f(X_1) - f(X_2)|, \quad j=1,2,\dots,p$$

where $X_m = (x_{m1}, x_{m2}, \dots, x_{mp})$

and components x_{mj} and target $f(X_m) \in \{0,1\}$

$$D_H(X_m, X_n) = |X_m \oplus X_n| \quad \oplus = \text{modulo-2 sum}$$

Spectral contribution and monotonicity constraints

Define $\vec{X}_m = (x_{m1}^{\sigma_1}, x_{m2}^{\sigma_2}, \dots, x_{mp}^{\sigma_p})$

where j th component of $\vec{X}_m, x_{mj}^{\sigma_j} \in \{0^0, 0^1, 1^0, 1^1\}$

Now $\sigma_j = 1$ simply indicates the existence of a pattern pair unit D_H apart in the j th component that provides a net contribution to the spectrum [Win97] (Rademacher-Walsh or alternative ordering). We define the spectral contribution of the j th component for pattern \vec{X}_1 as excitatory if $x_{1j} = f(X_1)$ and inhibitory if $x_{1j} \neq f(X_1)$. We label the spectral contribution of the j th component σ_j^+ or σ_j^- according to the sign of $(-1)^{|x_{1j} - f(X_1)|}$ so that we can sum the positive and negative contributions separately.

Summing σ_j^+ and σ_j^- over all patterns ($\sum_{\vec{X}} \sigma_j$) gives

the first order spectral coefficients, decomposed into excitatory and inhibitory contributions. In [Win97] we specified tests for separability for a subset of patterns based upon k -monotonic criterion, formulated as a constraint on spectral contributions. For example, the existence of two patterns \vec{X}_1 and \vec{X}_2 satisfying

$$f(X_1) \neq f(X_2), \quad x_{1j} = x_{2j}, \quad \sigma_{1j} = \sigma_{2j} = 1$$

provides evidence that the subset containing those two patterns is non-separable because both $\sum_{\vec{X}} \sigma_j^+ > 0$

and $\sum_{\vec{X}} \sigma_j^- > 0$, implying that the 1-monotonic

constraint is violated for j th component. Constraints for k -monotonicity, $k > 1$ involve relations between pairs of components.

Rule for assigning σ_j for incompletely specified case

We propose a simple modification to the counting technique for incompletely specified, noisy and possibly contradictory patterns. Assuming no evidence to weight one component more than another, the contribution from each pattern pair is inversely proportional to D_H and equally shared. Thus, all pattern pairs contribute to the spectral summation, not just the nearest neighbour (unit D_H) as in the completely specified case.

For all X_1, X_2 such that $f(X_1) \neq f(X_2)$

$$\text{Assign } \sigma_j = |x_{1j} - x_{2j}| |X_1 \oplus X_2|^{-1}$$

where σ_j is the spectral contribution of j th component and, as before, we label spectral contributions σ_j^+ and σ_j^- to differentiate between excitatory and inhibitory contributions. Note that a pair of contradictory patterns has identical pattern components and therefore has no net spectral contribution, $\sigma_j^+ = 0$ and $\sigma_j^- = 0$. We have not experimented with more sophisticated distance measures.

Extracting Subsets

The Sequential Learning algorithm [Mar90] extracts maximally separable subsets by finding and removing patterns causing non-separability at each step. In our approach we use k -monotonic criterion to identify patterns for removal, and for the experiments reported here we sum the evidence for $k=1$ without considering $k > 1$. In order to sort the patterns according to degree of separability we assign a heuristic measure h to sort patterns according to sum of normalised excitatory and inhibitory contributions as follows:

$$h = \sum_{j=1}^p \left[\text{signum} \left(\sum_X \sigma_j^+ - \sum_X \sigma_j^- \right) \left(\frac{\sigma_j^+}{\sum_X \sigma_j^+} - \frac{\sigma_j^-}{\sum_X \sigma_j^-} \right) \right]$$

where $\text{signum}()$ ensures that sign of the j th contribution to h is based on the larger of $\sum_X \sigma_j^+$ and $\sum_X \sigma_j^-$.

Figure 1 shows a typical plot of cumulative sum of h as the number of patterns in the set is increased for Gaussian data discussed in the next section. Note that the sum is zero for the empty and full set and achieves a maximum when the increase and decrease in h is approximately balanced. We checked that this heuristic function had a clearly defined maximum for at least the first three extracted subsets, and we used the peak as a threshold for the number of patterns to extract.

Results

In these experiments, we estimate patterns close to the optimal boundary (the boundary set) by extracting subsets. We then recursively repeat the extraction with a

random fixed fraction of the boundary set left out of the training set of each individual classifier. For all experiments, we used a 3 hidden-node MLP base classifier using Levenberg-Marquardt optimisation algorithm. The classifier is run B times with random initial weights, and terminated after 50 cycles.

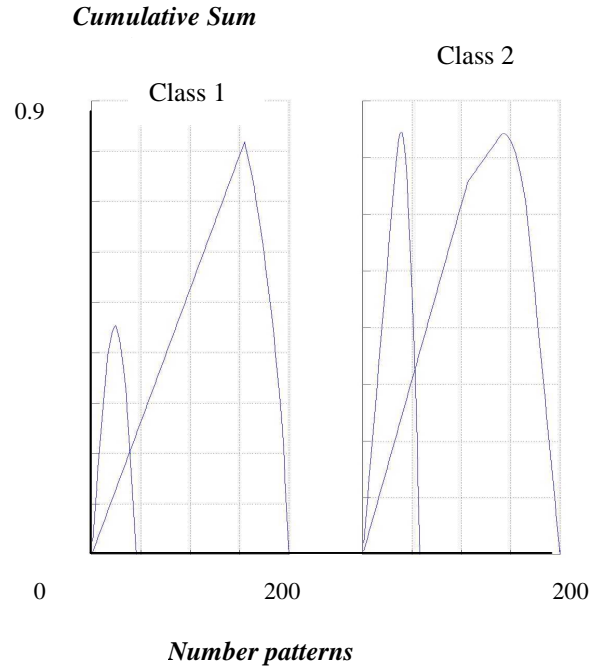


Figure 1: Cumulative sum of h versus number of patterns for two extracted subsets with class 1 and class 2

Weighted combination of B base classifier outputs is performed using a single layer perceptron. The orientation weights of the perceptron are fixed at values proportional to the first order spectral coefficients, found from a single node of the Sequential Learning algorithm as described in the previous section. Although the spectral count uses hard-level information to determine orientation weights, the bias is learned using gradient descent with soft-level values applied to perceptron inputs. For comparison, we also report results for combination by majority vote.

For N runs of the combiner we determine the confidence that a pattern does not belong to one of the first two extracted subsets by counting the number of times it remains after the first two subsets are extracted. Dividing by N gives a number between 0 and 1 for the confidence, and by varying a confidence threshold, $threshB$ we change the number of patterns in the boundary set. We recursively repeat $B \times N$ runs, but leaving out a fraction bf of the boundary set as defined by $threshB$.

We define level1 combining as the result of combining the outputs of B runs of the base classifier. L1CS denotes the level1 error rate after combining by perceptron with orientation weights set by spectral counting. Similarly L1CM denotes combining by majority vote. The reported experiments are for $B=50, N=50$.

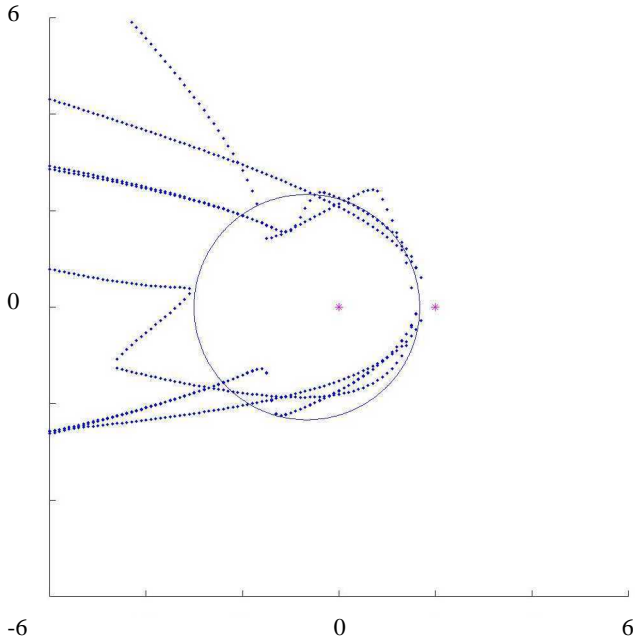


Figure 2: decision boundaries of four base classifiers

We use a simple 2D overlapping Gaussian, with {mean (0,0), var. 1} and {mean (2,0), var. 4} with 200 patterns from class 1 and 2 respectively [Yee92]. The Bayes boundary is circular for this problem with Bayes error rate of 18.49%. The problem is useful as a developmental tool to visualise decision boundaries and understand how the Bayes boundary is approximated. Typical individual decision boundaries with respect to the circular Bayes boundary are given for four base classifiers in Figure 2 (unequal spacing of the points is an artefact of plotting routine). For this data we checked that Bootstrapping on the full training set with majority vote gave no further improvement in generalisation over that obtained from combining with random initial weights without Bootstrapping.

Fig 3 shows boundary patterns for the Gaussian problem as $threshB$ is varied in 0.1 intervals. For this problem, patterns in the first and second extracted subsets correspond to patterns correctly and incorrectly classified respectively. In Figure 4 we see how the decision boundary varies with bf .

To test the combined classifier, we used 30,000 independently generated patterns and repeated ten times. Figure 5 shows how the error rate varies with $threshB$.

	Base Rec0	L1CS Rec0	Base Rec1	L1CS Rec1
mean	21.98	19.21	22.6	18.95
std	0.9	0.060	0.58	0.062

Table 1: error rates %, Rec0: initialisation ($threshB=1$) Rec1: $threshB=0.7, bf=0.5$

Table 1 compares error rates for base and combined classifiers before and after first recursive estimate. Leaving out boundary patterns from training sets appears to increase slightly the mean bias of the base classifiers, but reduces bias of the combiner. We also tried ten different seeds for the training set, and at $threshB = 0.7, bf = 0.5$, mean error(L1CS) = 19.01 %, std=0.060 and mean error(L1CM)=19.14%, std=0.096. The minimum mean error rate(L1CS) was 18.85 %.

Fig 6 shows how the relative size of the extracted subset varies with $threshB$ for the first recursive estimate. Note that at $threshB \cong 0.7$, the extracted subsets are of equal size, and we performed further experiments (not reported here) which showed that the size remained stable for at least 3 more recursive estimates. However we have not experimented enough to determine whether this is a viable way of finding $threshB$. Also we have not experimented with the use of a validation set for determining $threshB$.

We tested the proposed method with the diabetes data set [Mer98], whose performance is allegedly difficult to improve with Boosting, due to noise [Qui96]. We kept the parameters of the base classifier unchanged although the combined training rate for 3 hidden-nodes (14%) indicated significant overfitting. Experiments are repeated ten times with random 50/50 splits for training and testing. Figure 7 shows how the error rate varies with $threshB$; Figure 8 gives a typical plot of the cumulative sum of h for the two classes, similar to Figure 1.

Results are not reported but we also tried some variants of combining. We tried combining by non-linear single hidden layer MLP using 2,3,4 nodes of Sequential Learning. Initial orientation weights were set according to the spectral counts as in the single perceptron, and bias weight learned by BackPropagation, but we observed no significant improvement in performance. Also level2 combining (L2CS and L2CM) gave similar result to mean (L1CS). However we did observe improvement in generalisation for level2 combining when $threshB$ was varied over the N runs of the level1 combiner.

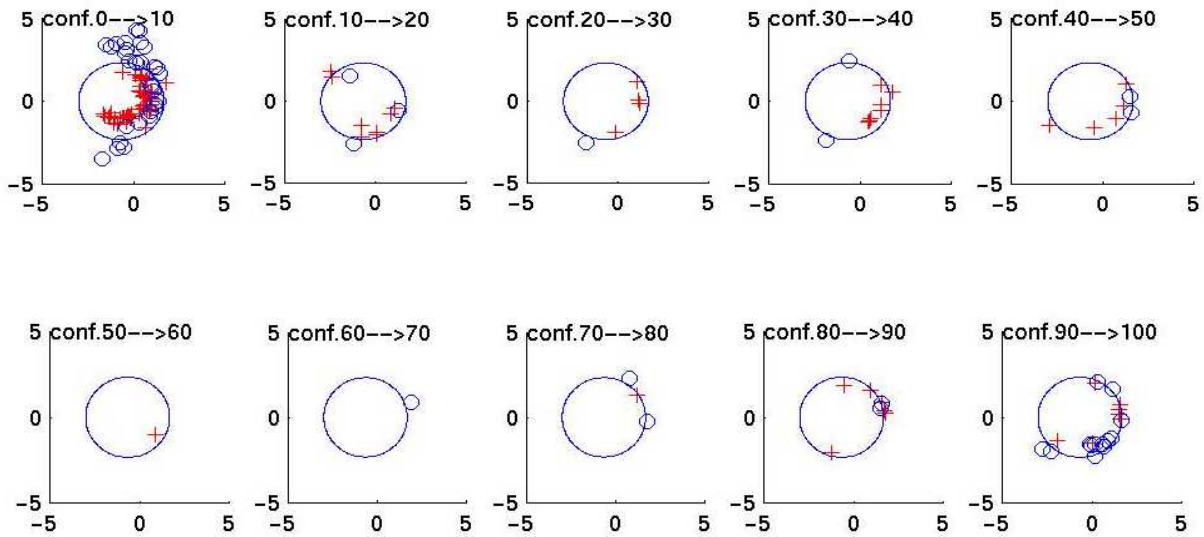


Figure 3: Boundary patterns in 10% confidence intervals

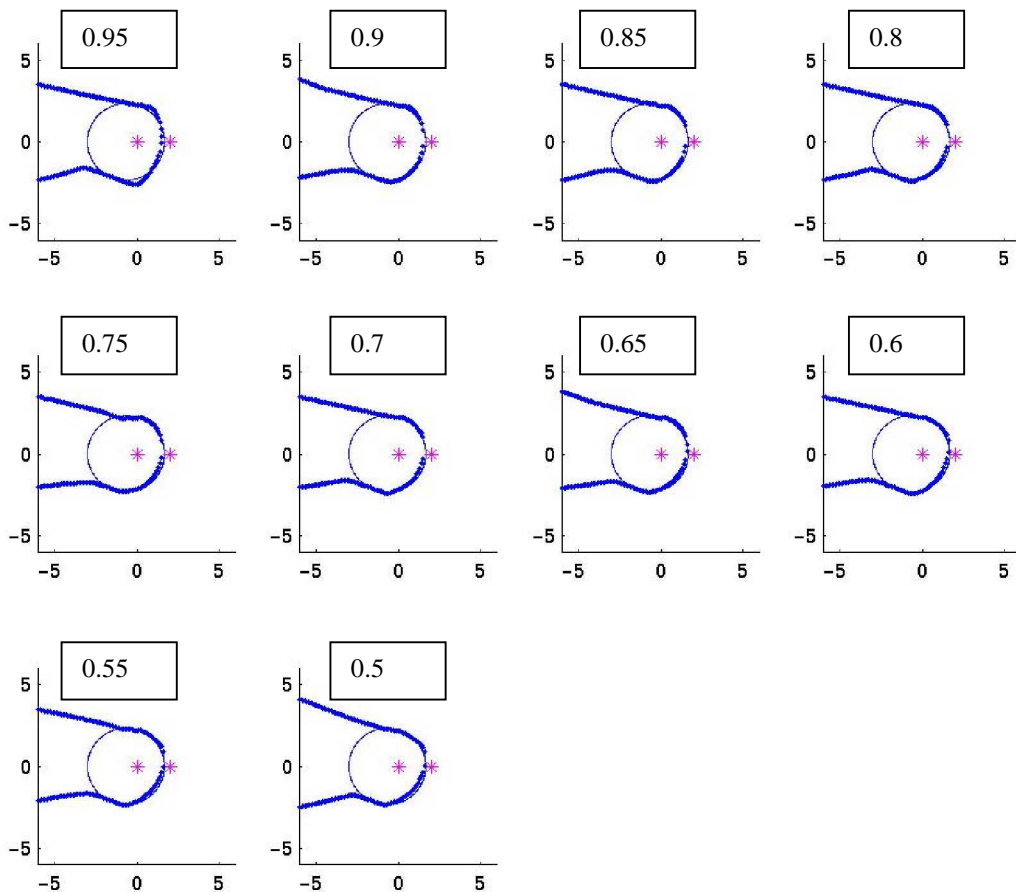


Figure 4: Combined classifier decision boundaries as bf is varied, $threshB = 0.7$

Conclusion

We have presented an approach based on spectral representation of intermediate feature space for improving generalisation in a multiple classifier framework. Correlation between experts is reduced by identifying patterns close to the decision boundary and leaving out a random fraction from the training set of each classifier. We will report separately on the sensitivity of the proposed technique to choice of problem, and to choice of important parameters such as number of hidden-nodes of base classifier.

References

- [Bre97] Breiman L., Bagging Predictors, Machine Learning, 24(2), 1997, pp123-40.
- [Die95] Dietterich T.G., Bakiri G., Solving multiclass learning problems via error-correcting output codes, J. of Artificial Intelligence Research 2, 1995, pp263-286.
- [Fre97] Freund Y., Schapire R.E.. A decision-theoretic generalisation of on-line learning and an application to boosting, J. of Computer and System Science, 55(1), 1997, pp119-139.

[Kit97] Kittler J., Hojjatoleslami A., Windeatt T., Strategies for combining classifiers employing shared and distinct pattern representations, Pattern Recognition Letters, August 1997, Vol.18, No.11-13, pp.1373-1377.

[Mar90] Marchand M., Golea M., Rujan P., A convergence theorem for sequential learning in two-layer perceptrons, Europhys. Lett. 11(6), pp 487-492.

[Mer98] Merz C.J., Murphy P.M., UCI repository of machine learning databases, 1998

[Qui96] Quinlan J.R., Bagging, Boosting and C4.5, in Proc 13th Conf. on AI, MIT press, 1996, pp725-730.

[Rat98] Ratsch G., Onoda T., Soft margin for Adaboost, Tech Report 021, NeuroCOLT, Berlin, Aug 1998.

[Win97] Windeatt T., Tebbs R., Spectral technique for hidden layer neural network training, Pattern Recognition Letters, December, 1997, Vol.18, No.8, pp.723-731.

[Win99] Windeatt T., Ghaderi R, Adaboost and neural networks, ESANN99, Bruges, April, 1999.

[Yee92] Yee P., Classification requirements involving Backpropagation and RBF networks, tech report 249, McMasters Univ, Ontario, 1992.

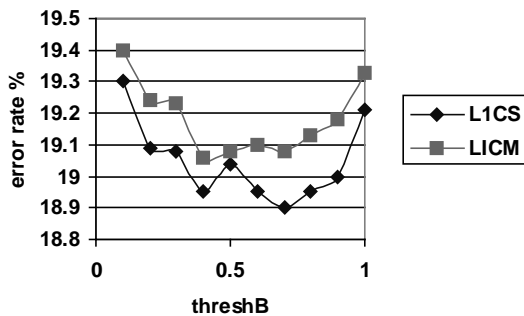


Fig. 5: error rate for varying boundary set threshold, $bf = 0.5$

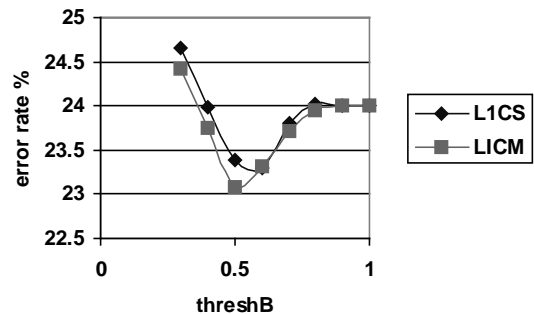


Fig. 7: error rate for varying boundary set threshold, $bf = 0.5$ Diabetes data

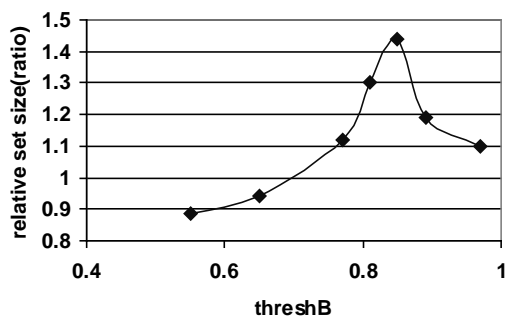


Fig. 6: relative size of boundary set estimate after one recursion vs threshB

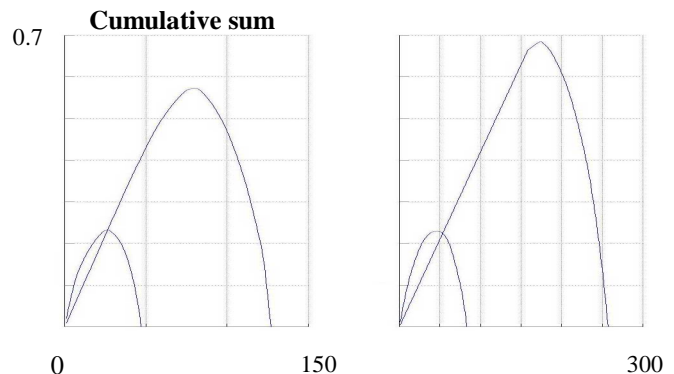


Figure 8: Cumulative sum of h versus number of patterns, Diabetes data