

AdaBoost and SVM's for Unbalanced Data Sets

Matthew J. Urffer
University of Tennessee
Knoxville, Tennessee, 37916
Email: matthew.urffer@gmail.com

Abstract—

I. INTRODUCTION

Given a set S of classified training examples a supervised learning algorithm attempts to create a hypothesis h which correctly classifies each class.

A. Support Vector Machines

Support Vector Machines (SVM) are

For a binary classification the decision function of the SVM is the dot product of the weight vector and the training example in the feature space added to a bias vector as shown in Equation 1.

$$f(\vec{x}) = \langle \vec{x}\phi(\vec{x}) \rangle + \vec{b} \quad (1)$$

where $\phi(\vec{x})$ is a mapping to the higher dimensional space. The learning in the SVM is then finding the optimal values of the weight vector \vec{w} and the basis \vec{b} .

The radial basis function (Equation 2) is a common kernel function used to map the input vector \vec{x} into a higher dimension.

$$k(\vec{x}_i, \vec{x}_j) = \exp\left(-\frac{\|\vec{x}_i - \vec{x}_j\|^2}{2\sigma^2}\right) \quad (2)$$

Something about how we are trying to elarge the margin

B. Ensemble Classifiers

Given an individual classifier h , an ensemble of classifiers can be constructed of a set of individual classifiers, $H = h_1, h_2, \dots, h_n$.

As long as the individual classifiers have uncorrelated errors when any single classifier is incorrect the other classifiers in the ensemble might correctly classify the example.

C. Boosting

Unbalanced data sets (data sets in which a majority of the values come from one class, see Figure 1) are difficult for classification schemes to learn because the minority class is not well represented and tends to be thought as noise for the classifier. Often classifiers are trained from unbalanced data sets by artificially rebalancing the dataset by sampling techniques; i.e. up-sampling (sampling more from the minority class) and down-sampling (sampling less from the majority class). Boosting is an ensemble learning method in which a set of weights is maintained over the training samples and adaptively adjusted after each training iteration according to the ones that are misclassified [1]. By maintaining a weight distribution over all of the training examples, these weights

could be updated to emphasize the training examples that are misclassified incorrectly. These incorrectly classified examples could then be learned in a refinement of the classifier or by training adding a new classifier to the ensemble with the new weights.

II. METHODS

A. Support Vector Machines

Support vector machines were implemented with the libsvm library [1]. Radial basis functions were used as the kernel space for the SVM to create RBFSVM. There are then two parameters that need to be determined, C the regularization parameter and σ the width of the kernel. These parameters were determined by a grid search of the parameter space.

B. AdaBoost

AdaBoost was implemented as an ensemble of support vector machines by maintaining a weight distribution of the misclassification error over all of the training examples. At each cycle t AdaBoost provides the learning algorithm with training examples \vec{x} and a weight distribution w (initialized uniformly). The learning algorithm is trained to generate a classifier h_t and the weight distribution is updated to reflect the predicted results; easy training examples (in which the classifier is very certain) have their weights lowered, while hard samples have their weights increased. This process continues for T cycles. Finally, the AdaBoost combines all of the component classifiers into the ensemble whose signal, final hypothesis, is constructed by weighting the individual classifiers by their training errors.

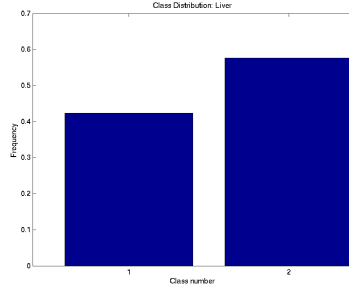
The AdaBoost algorithm (Algo 1) was extended to AdaBoostM1 in order to use the RBFSVM. This algorithm uses a fairly large RBFSVM kernel (a weak learning ability) for the first classifier, h_t . The classifier is then retrained with a smaller σ until the accuracy of the classifier has an accuracy of just over 50%. At this point the classifier is added to the ensemble, along with its weight based on its accuracy. The weights of the training samples are then updated to reflect the training examples that the classifier struggled with. This process is then repeated for the next classifier in the ensemble until all of the ensemble is full.

After training was complete the ensemble method was tested.

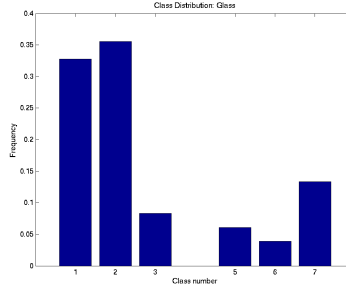
III. RESULTS

A. Parameter Search

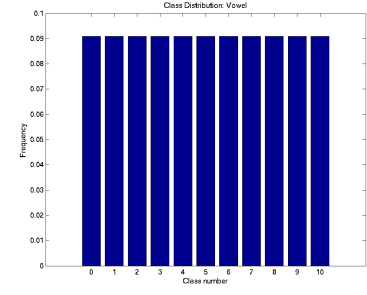
The parameter search for the optimal C and σ parameters is shown in the contour plots of Figures 2, 3 and 3. The optimal



(a) Liver



(b) Glass



(c) Vowel

Fig. 1: Distribution of Class Data

Algorithm 1 AdaBoostSVM

```

1: procedure ADABOOSTSVM( $\sigma_{init}, \sigma_{min}, \sigma_{step}, C, \vec{x}, \vec{y}$ )
2:    $w_i^1 \leftarrow 1/N \forall i = 1, \dots, N$ 
3:   while  $\sigma > \sigma_{min}$  do
     Train a RBFSVM component classifier
4:    $h_t \leftarrow \text{ComponentClassifier}(\vec{x}, \vec{w})$ 
     Compute the error of that classifier
5:    $h_t : \epsilon_t = \sum_{i=1}^N w_i^t, y_i \neq h_t(\vec{x}_i)$ 
     Decrease  $\sigma$  until a weak classifier
6:   if  $\epsilon_t > 0.5$  then
7:      $\sigma = \sigma - \sigma_{step}$ 
8:     go to 3
9:   end if
     Set weight of component classifier
10:   $h_t : \alpha_t = \frac{1}{2} \ln \left( \frac{1-\epsilon_t}{\epsilon_t} \right)$ 
     Update weights of training samples
11:   $w_i^{t+1} = \frac{w_i^t \exp(-\alpha_t y_i h_t(\vec{x}))}{C_t}$ 
      $C_t$  is a normalization value,  $\sum_{i=1}^N w_i^{t+1} = 1$ 
12:  end while return  $f(\vec{x}) = \text{sign} \left( \sum_{t=1}^T \alpha_t h_t(\vec{x}) \right)$ 
13: end procedure

```

TABLE II: Fine Optimal Classifier Parameters

Data Set	C_{min}	C_{max}	σ_{min}	σ_{max}	C	σ	ϵ
Glass	5.00	15.00	0.18	0.55	14.47	0.22	74.44
Liver	5.00	15.00	0.18	0.55	12.89	0.24	74.00
Vowel	5.00	15.00	0.55	1.66	15.00	1.19	99.43

accuracy, while too high a value tended to also have a low accuracy. Relatively large values of σ are then suited for the ensemble methods because they tend to be representative of an RBFSVM which a relatively weak learning ability which generalizes better.

A. Future Work

Future work might be to change to use the decision value instead of the output class.

REFERENCES

- [1] X. Li, L. Wang, and E. Sung, "AdaBoost with SVM-based component classifiers," *Engineering Applications of Artificial Intelligence*, vol. 21, pp. 785–795, Aug. 2008.

Make classifier parameters are shown for the coarse parameter search in Table I and for the fine parameter search in Table II.

TABLE I: Coarse Optimal Classifier Parameters

Data Set	C_{min}	C_{max}	σ_{min}	σ_{max}	C	σ	ϵ
Glass	1.00	10.00	-7.00	7.00	10.00	0.37	71.67
Liver	1.00	10.00	-7.00	7.00	10.00	0.37	74.00
Vowel	1.00	10.00	-7.00	7.00	10.00	1.11	99.24

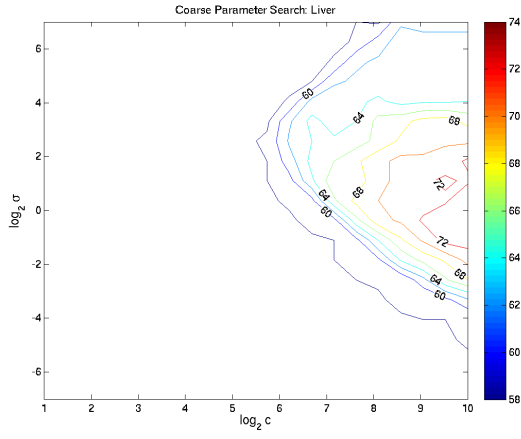
B. AdaBoostM1

IV. CONCLUSIONS

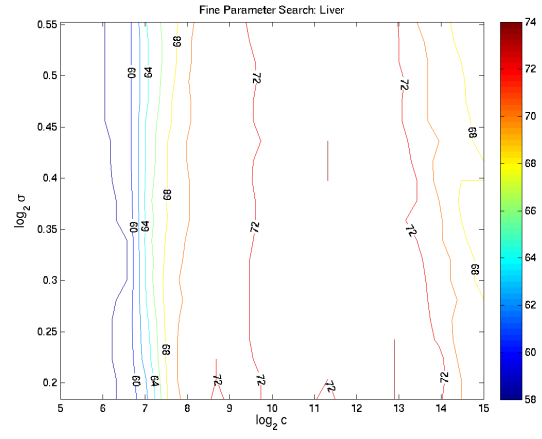
Support vector machines were implemented on unbalanced data sets

The effect of having a large σ can be observed by examining Figures 2, 3 and 3. Larger values of σ tended to have a low

Make some quantifications and discuss the data

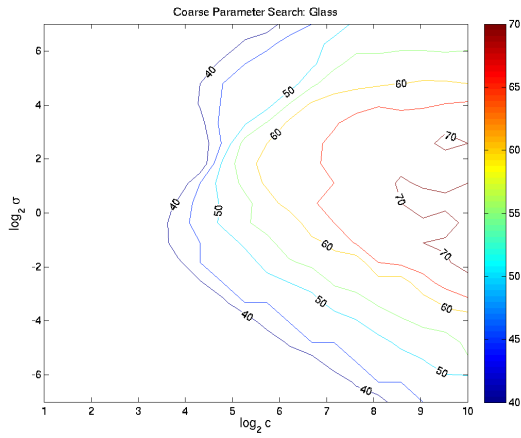


(a) Coarse Search

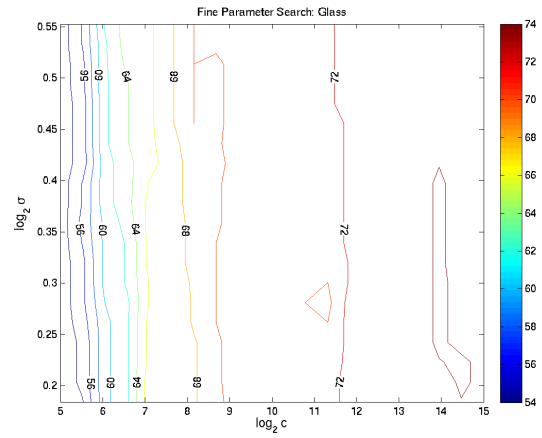


(b) Fine Search

Fig. 2: Parameter search for Liver Disorder

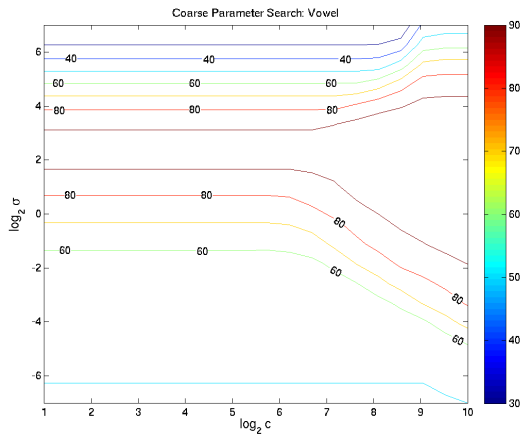


(c) Coarse Search

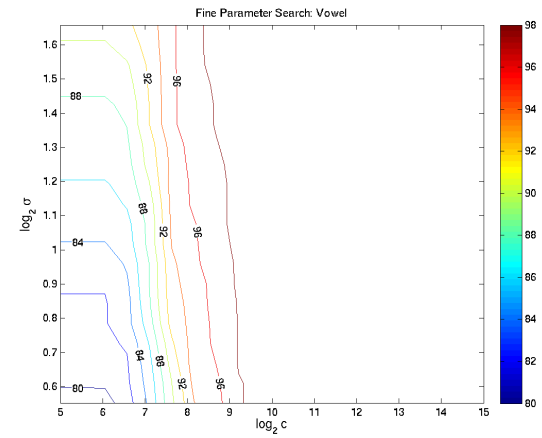


(d) Fine Search

Fig. 3: Parameter search for Glass Disorder



(a) Coarse Search



(b) Fine Search

Fig. 4: Parameter search for Vowel Disorder