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 algorthim attempts to create a hypothesis h which correctly classifiers 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.

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

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 kernal function used to map the input vector $\vec{x}$ into a higher dimension.

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

Something about how we are trying to elarge the margin

B. Ensamble Classifers

Given an individiual classifier h, an enamble of classifers can be constructed of a set of indvidual classifers, $H = h_1, h_2, ;h_n$.

As long as the individual classifers have uncorrelated errors when any single classifer is incorrect the other classifiers in the ensamble 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 classifer. Often classifiers are trained from unblananced data sets by artifically reblaning the dataset by sampling techniques; i.e. up-sampling (sampling more from the minoarty class) and down-sampling (sampling less from the majoirty class). Boosting is an ensamble learning method in which a set of weights is maintained over the training samples and adaptively adjusted after each training itteration according to the ones that are misclasified [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 classifer to the ensamble with the new weights.

II. Methods

A. Support Vector Machines

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

B. AdaBoost

AdaBoost was implemented as an ensamble of support vector machines by maintaining a weight distribution of the misclassificaiton error over all of the training examples. At each cycle t AdaBoost provides the learning algorthim with training examples $\vec{x}$ and a weight distriubtion w (initialized uniformly). The learning algorthim 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 ensamble whose signal, final hypothesis, is constructed by by weighting the individual classifers by their training errors.

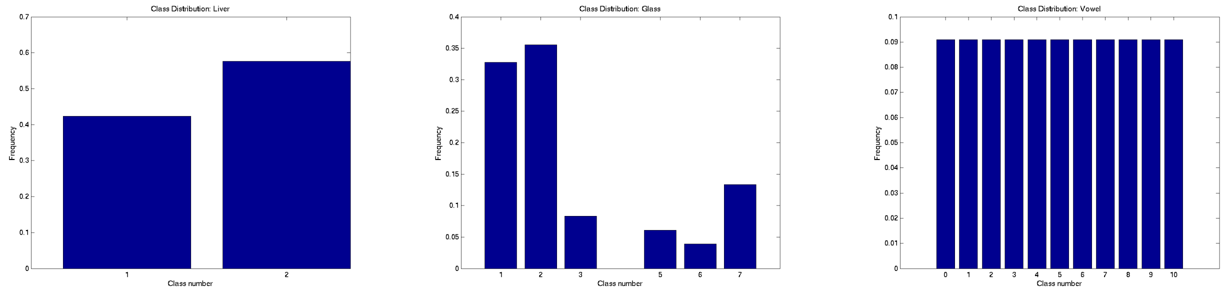
The AdaBoost algorthim (Algo 1 was extended to AdaBoostM1 in order to use the RBFSVM. This algorthim uses a fairly large RBFSVM kernal (a weak learning ability) for the first classifier, h_t. The classifier is then retrained with a smaller σ until the accuracy of the classifier has an accuracy of just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

After training was complete the ensamble method was tested.

III. Results

A. Parmater 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:     $w_i^1 \leftarrow 1/N \forall \quad i = 1, \dots, N$
3:     while $\sigma > \sigma_{min}$ do
   Train a RBFSVM component classifier
4:         $h_t \leftarrow$ ComponentClassifer($\vec{x}, \vec{w}$)
   Compute the error of that classifer
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 classifer
10:        $h_t \ : \ \alpha_t = \frac{1}{2} ln \left( \frac{1-\epsilon_t}{\epsilon_t} \right)$
   Update weights of training samples
11:        $w_i^{t+1} = \frac{w_i^t exp(-\alpha_i y_i h_t(\vec{x}))}{C_t}$
   $C_t$ is a normalization value, $\sum_{i=1}^{N} w_i^{t+1} = 1$
12:    end whilereturn $f(\vec{x}) = sign \left( \sum_{t=1}^{T} \alpha_t h_t(\vec{x}) \right)$
13: end procedure
```

Make some quantifications and discuss the data

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	6.68	3.32	68.33
Liver	1.00	10.00	-7.00	7.00	9.05	0.37	73.00
Vowel	1.00	10.00	-7.00	7.00	7.63	1.11	95.83

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

TABLE II: Fine Optimal Classifier Parameters

Data Set	C_{min}	C_{max}	σ_{min}	σ_{max}	C	σ	ϵ
Glass	3.34	10.03	1.66	4.97	6.16	4.97	68.89
Liver	4.53	13.58	0.18	0.55	13.10	0.22	74.00
Vowel	3.82	11.45	0.55	1.66	8.23	1.25	96.02

TABLE III: AdaBoost ClassiferValues

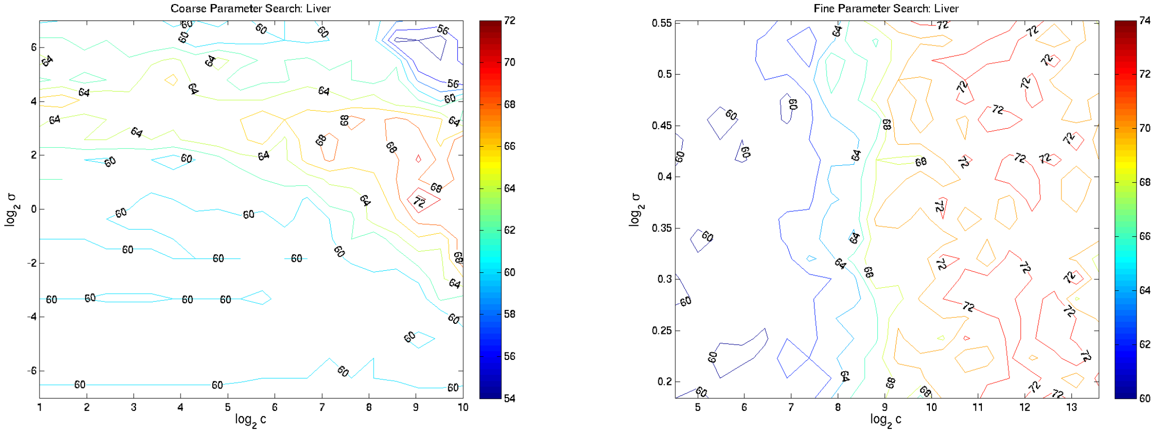
Data Set	T	σ_{init}	C	ϵ
Glass	50	5.00	6.16	76.47
Liver	50	3.00	13.10	73.33
Vowel	50	2.00	10.00	56.49

A. *Future Work*

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

References

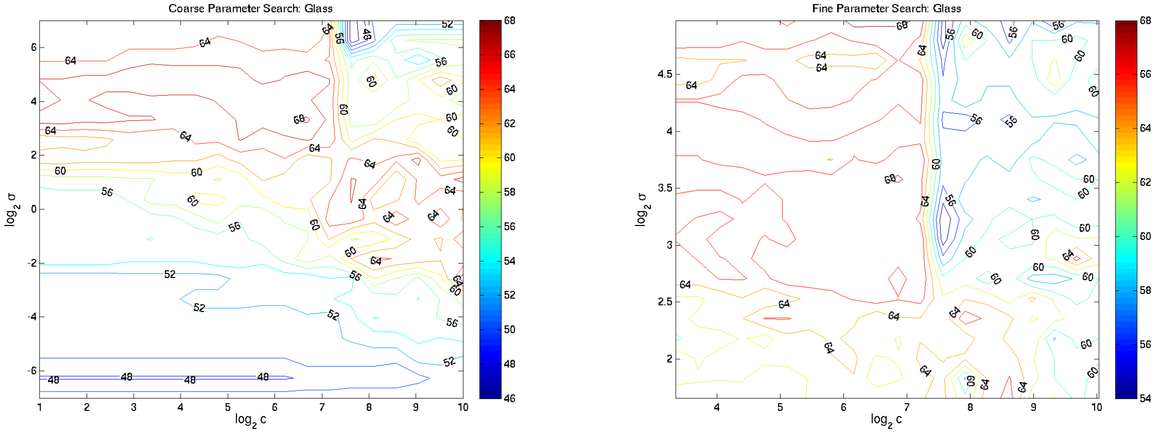
[1] X. Li, L. Wang, and E. Sung, “AdaBoost with SVM-based component classifiers,” *Engineering Applications of Artificial Intelligence*, vol. 21, pp. 785–795, Aug. 2008.



(a) Coarse Search

(b) Fine Search

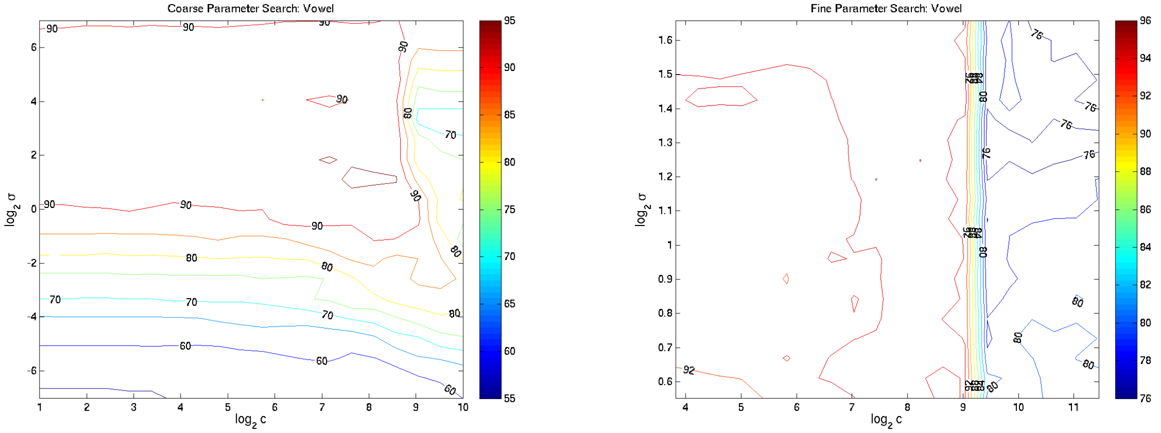
Fig. 2: Parameter search for Liver Disorder



(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