AdaBoost and SVMs for Unbalanced Data Sets

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

***Abstract*—Support vector machines (SVM) were investigated for their use in single class and multi-class classification problems. The radial basis function was used as the kernel function, and the classifiers were trained for a variety of kernel sizes and regularization parameters in order to find the optimal classifier parameters for a single data set. After the optimal value was found for a single classifier, the AdaBoostM1 algorithm was implemented and an ensemble of weak learners was trained. The ensemble classifier trained with AdaBoostM1 had a higher accuracy than the single support vector machine for the Glass and Liver data set. The vowel data set, however, had a very high accuracy (96%) for the single classifier while the ensemble classifiers preformed poorly. As the vowel data set was the only example of a balanced data set while the others were unbalanced data sets, it can be concluded that the AdaBoostM1 algorithm improves the performance of a classifier on an unbalanced data set.**

I. Introduction

A supervised machine learning problem is one which a learning algorithm is presented a set of training data and attempts to find an unknown function which maps the training values to the correct answer. Typically the training set, denoted S, is a set of the form $\{(\vec{x}_1, y_1), (\vec{x}_2, y_2), \ldots, (\vec{x}_n, y_n)\}$ where $\vec{x}_i$ is vector of some features of the problem. Problem features are generally discrete or real valued items. Examples of problem features that could represent a person include height, weight, age, zip code, grade point average, starting salary, and telephone number. The y_i are the class of the training feature $\vec{x}_i$ belongs to; these might be University of Tennessee students or Carnegie Mellon students. In this examples students with a zip code of 15213 are likely to be Tartans, while students with a zip code of 37916 are like to be Volunteers. The challenge arises from examples have overlapping features; for example, this author as a former Tartan and current Vol would be difficult to classify by zip code. The learning algorithm's job is then to find a hypothesis h that correctly classifies a student as a Volunteer of Tartan based on the features provided. This learning process can then be defined as finding the hypothesis that has the least error (incorrect classifications) on the training data set while extending to examples outside of the training space.

A. Support Vector Machines

Support Vector Machines (SVM) are a supervised learning technique in which hyperplanes are constructed in a high dimensional space to which the features are mapped. SVMs find the hyperplanes that are the farthest away from all of mapped features in order to provide excellent training performance while still maintaining the ability to generalize to new instances; i.e. SVMs are maximal margin classifiers. 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

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

where $\phi(\vec{x})$ is a mapping to the higher dimensional space. The SVM is then learning 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.

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

The maximal margin is ensured by minimizing:

$$g(\vec{w}, \eta) = \frac{1}{2}\|\vec{w}\| + C\sum_{i=1}^{N}\zeta_i \tag{3}$$

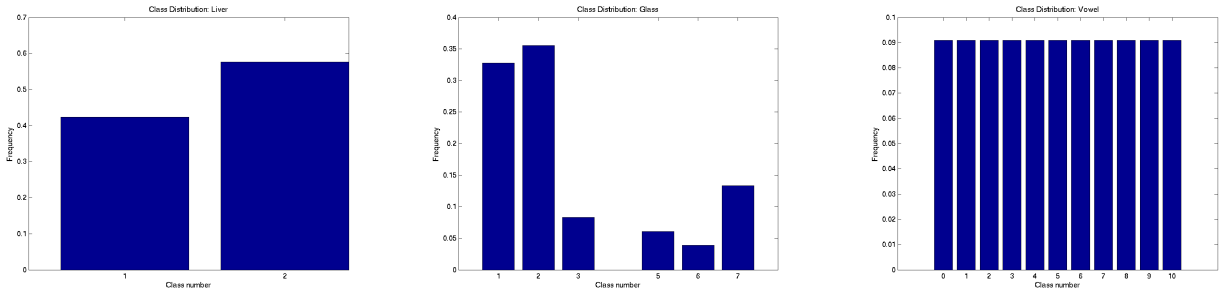
subject to:

$$y_i(\langle \vec{w}, \phi(\vec{x}) \rangle + b) \geq 1 - \zeta_i, \quad \zeta_i \geq 0 \tag{4}$$

where ζ_i is the ith slack variable and C is the regularization parameter [1]. This problem can be translated in to the Wolfe dual form, which can be solved with quadratic programing [1].

B. 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 balancing the data set 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]. Given an individual classifier h, an ensemble of classifiers can be constructed of a set of individual classifiers, $H = \{h_1, h_2, \ldots, h_n\}$. 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. Performance of the ensemble is enhanced as long as the individual classifiers are weak and have uncorrelated errors, as when any single classifier is incorrect the other classifiers in the ensemble might correctly classify the example.



(a) Liver (b) Glass (c) Vowel

Fig. 1: Distribution of Class Data

II. Methods

A. Support Vector Machines

Support vector machines were implemented with the LIBSVM library [2]. 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 where determined by a grid search of the parameter space. In searching for the optimal parameters with a grid search, five fold cross validation of the data was used. In addition, it was noticed for the vowel data sets that the large degree of accuracy was achieved by using a large amount of support vectors in which the SVM was essentially memorizing the training examples. This was mitigated by decreasing the error factor by passing the correct input argument to `svmtrain`.

B. AdaBoost

AdaBoost was implemented as an ensemble of support vector machines by maintaining a weight distribution of the misclassified 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 (Algorithm 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.

Algorithm 1 AdaBoost SVM

1: **procedure** AdaBoostSVM($\sigma_{init}, \sigma_{min}, \sigma_{step}, C, \vec{x}, \vec{y}$)
2: $w_i^1 \leftarrow 1/N \ \ \forall \ \ i = 1, \ldots, N$
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 σ 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: $w_i^{t+1} = \frac{w_i^t exp(-\alpha_i y_i h_t(\vec{x}))}{C_t}$
C_t is a normalization value, $\sum_{i=1}^{N} w_i^{t+1} = 1$
12: **end while**
return $f(\vec{x}) = sign \left(\sum_{t=1}^{T} \alpha_t h_t(\vec{x}) \right)$
13: **end procedure**

The choice of σ is critical for the convergence of the presented algorithm. Too small a σ and the RBFSVM tends to over fit the data and provide correlated classifiers (and thus their error will be correlated), while too large a value of σ and the classifier is too weak. The implemented algorithm only varied the value of σ because it is known that RBFSVM depend highly on σ and less on C so the performance of the classifier can be changed over a set C by simply changing σ [1]. In addition ensemble sizes for 5 to 150 were tested; these results are shown in Section III.

Several experiments were completed once the code base was written. Three data sets were obtained from the LIBSVM homepage [2]. These data sets were parsed and scaled with the tools provided in the LIBSVM package. First the optimal number of ensemble members was found by investigating the classification accuracy dependence on the number of support vectors. Next the AdaBoost implementation versus the accuracy achieved by the a single parameter grid search. Finally the

classification errors between the two methods are analyzed.

III. Results

It can be seen by Figure 1 that two of the data sets, the liver and glass, are imbalanced while the third, vowel, has each class equally represented. Furthermore it is observed that the glass data set is the most imbalanced; this is where the largest increase of performance due to the AdaBoostM1 is expected. The results of an SVM on the data sets is presented in Section III-A. In Section III-B the results are shown for using an ensemble methods.

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 liver and glass data set contour plots have many topological features, indicating that small variations in the RBFSVM parameters cause dramatic changes in the accuracy of the trained RBFSVM. The vowel data set does not display these features but rather has a plateau region atop a precipice in which the accuracy of the classifier does not change dramatically. The optimal classifier parameters are shown for the coarse parameter search in Table I and for the fine parameter search in Table II. C_{min} and C_{max} are the starting values of the grid search normalization parameter, σ_{min} and σ_{max} are the range of kernel size while C, σ and ϵ are the optimized normalization parameter, kernel size, and final accuracy respectively. The values for the fine parameter search were chosen to be 50% of the optimal values selected by the coarse parameter search.

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	1.00	3.89	23.53
Liver	1.00	10.00	-7.00	7.00	8.00	2.33	2.22
Vowel	1.00	10.00	-7.00	7.00	8.00	0.78	56.28

TABLE II: Fine Optimal Classifier Parameters

Data Set	C_{min}	C_{max}	σ_{min}	σ_{max}	C	σ	ϵ
Glass	0.50	1.50	1.94	5.83	1.50	3.69	68.89
Liver	4.00	12.00	1.17	3.50	10.69	1.36	74.00
Vowel	4.00	12.00	0.39	1.17	8.41	1.10	96.21

The confusion matrices of the RBFSVM classifiers are presented in Tables III - V.

TABLE III: Confusion Matrix for RBFSVM (Liver)

	1	2	3	4	5	6
1	0	0	13	0	4	1
2	0	1	20	2	4	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0

TABLE IV: Confusion Matrix for RBFSVM (Glass)

	1	2	3	4	5	6	7	8
1	0	0	0	0	0	0	0	0
2	0	8	0	0	3	0	0	0
3	3	7	0	1	1	0	0	0
4	0	1	0	0	1	0	0	0
5	0	0	0	0	0	0	0	0
6	1	0	1	0	0	0	0	0
7	0	0	1	0	1	0	0	0
8	0	2	2	1	0	0	0	0

TABLE V: Confusion Matrix for RBFSVM (Vowel)

	1	2	3	4	5	6	7	8	9	10	11
1	35	7	0	0	0	0	0	0	0	0	0
2	9	19	9	0	0	0	0	0	0	0	5
3	0	2	24	11	0	0	0	0	0	0	5
4	0	0	0	32	0	10	0	0	0	0	0
5	0	0	0	0	4	26	6	0	0	0	6
6	0	0	0	2	1	23	0	0	0	0	16
7	0	0	5	0	18	2	16	0	0	0	1
8	0	0	0	0	1	0	14	26	1	0	0
9	0	1	1	0	0	0	3	2	29	2	4
10	7	4	2	0	0	0	3	1	5	18	2
11	0	0	0	0	0	6	0	0	2	0	34

B. AdaBoostM1

The effect of the number of the classifiers in the ensemble is shown in Figure 5. It was observed for the unbalanced data sets the number of classifiers is the ensemble did not have a large effect (past a minimum amount of 20). Furthermore, it was surprising to note that the accuracy of the ensembles was constant for the liver data set, while the vowel data set showed an almost periodic trend that died out as the number of classifiers increased. The glass data set exhibited significant variation; it is thought that this occurs because of the well in the parameter space (Figure 3).

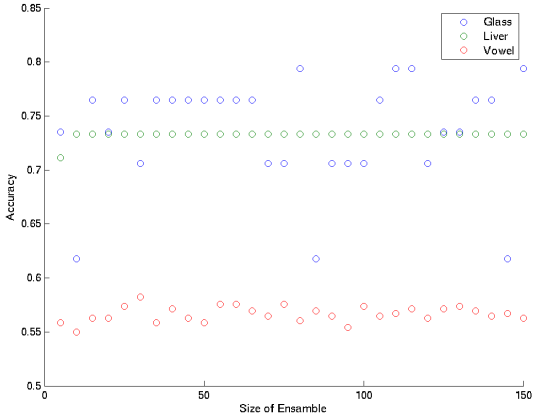
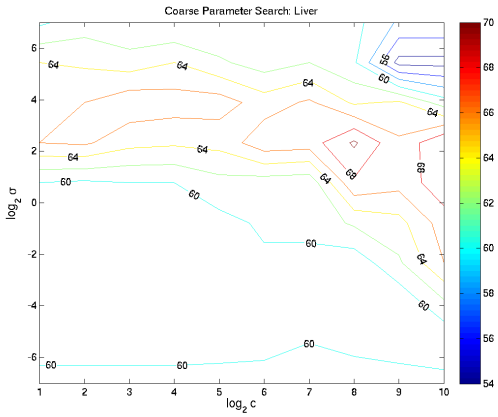
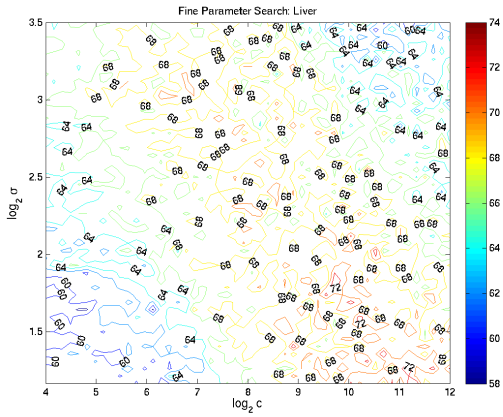


Fig. 5: Accuracy and Number of Components in Ensemble

The results using the implemented AdaBoostM1 algorithm are shown in Table VI, while the weights of the individual classifiers (representative of their accuracy) are shown in Figure 6. It is immediately observable that the AdaBoost algorithm

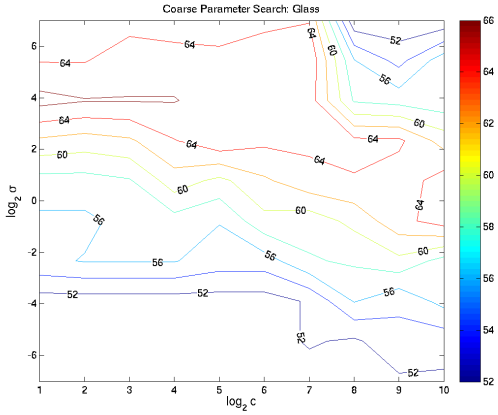


(a) Coarse Search

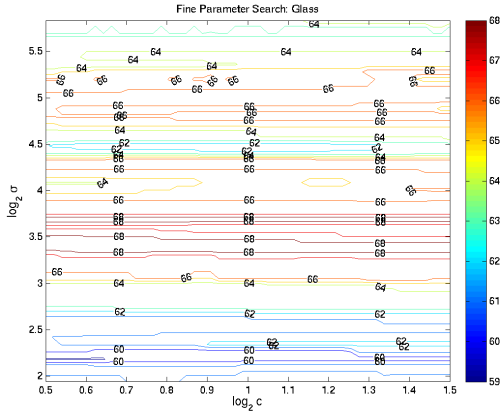


(b) Fine Search

Fig. 2: Parameter search for Liver Disorder

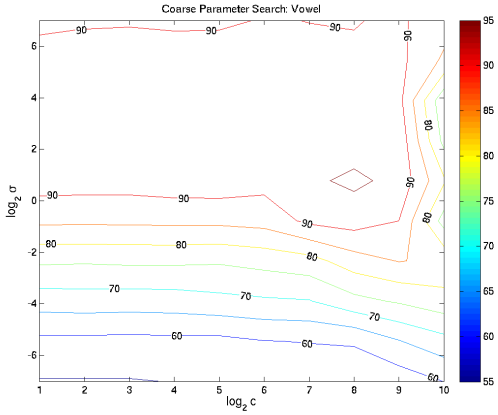


(c) Coarse Search

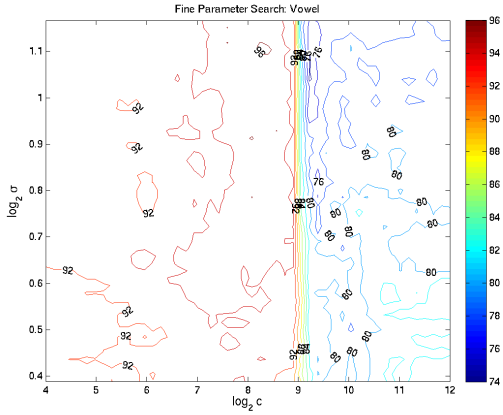


(d) Fine Search

Fig. 3: Parameter search for Glass Disorder



(a) Coarse Search



(b) Fine Search

Fig. 4: Parameter search for Vowel Disorder

increased the accuracy of the liver and glass data set, but failed to increase, and instead dramatically lowered the accuracy for the vowel data set. This is due to the vowel data set being balanced and each classifier being forced to become a weak classifier in the ensemble. When this ensemble is presented an input to classify, each classifier does not capture a specific region of the data set (as the data set is balanced), but instead in the voting scheme cancel each other out, resulting in poor accuracy.

TABLE VI: AdaBoost Classifier Values

Data Set	T	σ_{init}	C	ϵ
Glass	50	5.00	6.16	70.59
Liver	50	3.00	13.10	73.33
Vowel	50	2.00	10.00	56.49

Data Set	T	σ_{init}	C	ϵ
Glass	100	5.00	6.16	76.47
Liver	100	3.00	13.10	73.33
Vowel	100	2.00	10.00	55.84

T is the total number of classifiers trained, σ_{init} is the initial σ presented to AdaBoostM1, C is the constant RBFSVM normalization parameter, and ϵ is the accuracy of the ensemble method. Refer to Table II for the accuracy of individual RBFSVM.

The weight of each individual classifier for an ensemble of 150 members is shown in Figure 6. The confusion matrices of the AdaBoost ensemble classifiers are presented in Tables VII - IX.

TABLE VII: Confusion Matrix for AdaBoost Ensemble (Liver)

	1	2
1	10	8
2	4	23

TABLE VIII: Confusion Matrix for AdaBoost Ensemble (Glass)

	1	2	3	4	5	6
1	8	3	0	0	0	0
2	3	9	0	0	0	0
3	1	1	0	0	0	0
4	0	0	0	2	0	0
5	0	0	0	0	2	0
6	0	0	0	0	0	5

IV. Conclusions

Support vector machines were implemented on unbalanced data sets using the LIBSVM library. The AdaBoostM1 algorithm was implemented as an ensemble learning method, and the accuracy of the ensemble method was compared to that of the individual support vector machines. 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 ensemble methods because they tend to be representative of an RBFSVM with a relatively weak learning ability which generalizes better.

TABLE IX: Confusion Matrix for AdaBoost Ensemble (Vowel)

	1	2	3	4	5	6	7	8	9	10	11
1	36	6	0	0	0	0	0	0	0	0	0
2	9	19	9	0	0	0	0	0	0	0	5
3	0	2	24	11	0	0	0	0	0	0	5
4	0	0	0	32	0	10	0	0	0	0	0
5	0	0	0	0	4	26	6	0	0	0	6
6	0	0	0	2	1	23	0	0	0	0	16
7	0	0	5	0	18	2	16	0	0	0	1
8	0	0	0	0	1	0	14	26	1	0	0
9	0	1	1	0	0	0	4	2	28	2	4
10	8	4	2	0	0	0	3	1	4	17	3
11	0	0	0	0	0	6	0	0	3	0	33

Acknowledgments

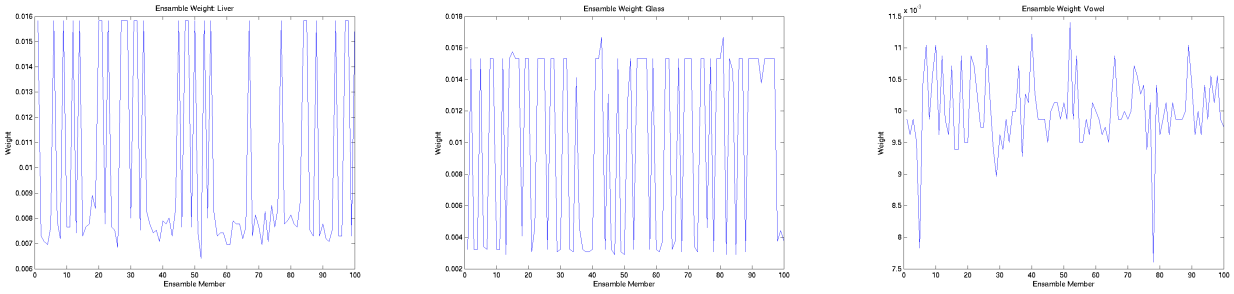
The conversations with Alan Nam of how to best use the LIBSVM library for the use of this project were extremely helpful. In addition Mike Franklin provided valuable comments on the structure of this project. Thanks are also due to Erica Lansberg for proof reading and editing, and to Eli Urffer (my little brother) for providing encourgement.

Finally, I would like to take this opportunity to thank both Dr. Parker and Mike for the tremendous job they did teaching machine learning. As an engineer I came into the class not knowing exactly what to expect, but left feeling that I have learned useful skills.

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.

[2] C. C. Chang and C. J. Lin, "LIBSVM: a library for support vector machines," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 2, no. 3, p. 27, 2011.



(a) Liver (b) Glass (c) Vowel

Fig. 6: Distribution of Ensemble Weights