Technical Report UALR06-02: Using Active Learning with Integrated Feature Selection

Hemant Joshi
Applied Science Dept.
2801 S University Ave.
Little Rock, AR, USA 72204
hmjoshi@ualr.edu

Xiaowei Xu
Information Science Dept.
2801 S University Ave.
Little Rock, AR, USA 72204
xwxu@ualr.edu

ABSTRACT

In this paper we present two very popular aspects in supervised Machine Learning algorithms: feature selection and active learning paradigm. Feature selection algorithms are widely used in Machine Learning to reduce the feature space representing given data samples. Active learning is very popular supervised Machine Learning technique that has been effectively used in Text Classification tasks to reduce training time and achieve high accuracy with small labeling cost[1]. Using feature selection integrated with active learning seems like a great idea as it combines faster learning with smaller feature space dimensionality. Though promising, we observe through various experiments that feature selection when integrated within active learning process yields inferior accuracy results. We believe that the changing Feature space representation hurts the very core of active learning strategy. We present the rationale behind poor accuracy and prove it empirically.

Categories and Subject Descriptors

I.2 [**ARTIFICIAL INTELLIGENCE**]: Natural Language Processing,Text analysis; I.2.6 [**Learning**]: [Machine Learning]

General Terms

Integrating Feature selection into active Learning, Feature space reduction

Keywords

Active Learning, Feature Selection

1. INTRODUCTION

In this paper we present two very popular aspects in supervised Machine Learning algorithms: feature selection and active learning paradigm. Feature selection algorithms are widely used in Machine Learning to reduce the feature space representing given data samples. Active learning is very popular supervised Machine Learning technique that has been effectively used in Text Classification tasks to reduce training time and achieve high accuracy with small labeling cost. In section 1.1 we look at Active Learning approach and then in section 1.2 we focus on the concept of Feature Selection.

1.1 Active Learning Paradigm

Several Machine Learning techniques (supervised and unsupervised) have been proposed for various classification problems. In most of such problems, the class labels for data samples are known and that means splitting the actual data into training set and test dataset is easy. Common practice has been that at least *60* to *66*% of the total data is used for training the classifier (supervised learning). The rest *33* to *40*% is then used for predicting class labels and determining the accuracy of the algorithm. Two obvious observations can be made in this scenario. First observation is that larger training data will most likely result in better yield. Second observation is that the classification algorithm can be as good as the training data. If we choose right amount of training samples and train classifier on particularly those samples that the classifier finds most difficult to predict class label for, then the accuracy of the classification is likely to improve.

Active learning [5],[3] is a type of learning paradigm popular in Machine Learning with focus on quick training and smaller training data size. In a typical supervised learning situation, training data samples are randomly selected and no particular qualitative criteria are put forth for building the model. The pre-condition to selecting training data should be that the classifier learns something new with each training sample. Using active learning approach, we select training samples iteratively and train the classifier only on the samples that provide some new information.

The data samples which are most difficult for the active learner to predict are also the likely candidates to provide new information. Training in this fashion will not only increase accuracy but will also build more robust models. Another advantage of active learning is that the training cost is less and provides maximum achievable yield by the model. Using oracle (human expert) for every sample that we need to label would be laborious as well as time consuming. It is not practical to have an individual or a group of individuals label millions of training samples. On the other hand, if the training samples offer nothing new to the learner algorithm,

[1]The cost to label a data sample by a human expert is quiet high with respect to time and efforts involved.

then the efforts of labeling the sample are in vein. Active learning methodology can identify samples that the classifier will learn new information from. We can label only those training samples that are identified by the active learner and build a new model. This process can continue till satisfactory accuracy level has been reached.

Tong et. al present pool based active learning approach with text classification problems and prove its usability. The accuracy achieved by active learning algorithm is comparable to the accuracy of training using random sampling but with very few training samples. The time and money saved in labeling only those samples that are important is a big plus of this approach. Tong et. al. present active learning with Support Vector Machine (SVM) algorithm[6][2]. The approach is as follows:

1. Train SVM on randomly selected instances (*20* samples) from a pool of labeled samples. Note training accuracy of the model.
2. Find distance of all data from the separating hyperplane and sort them in ascending order (See Figure *1*).
3. Select the data points (*20* samples) that are closest to the hyperplane and label them.
4. Use new set of samples along with existing training data to train the SVM (*step 1*) Continue this process until, accuracy doesn't increase anymore or until satisfactory training accuracy has been achieved.

In the next sub section we look at the process of typical feature selection.

1.2 Process of Feature Selection

Feature selection techniques are widely used in Machine Learning[9]. Feature selection as the name suggests allows selection of prominent or important features from the pool of all possible features. Feature space representation is important in Machine Learning problems like classification, categorization etc. Consider the case of supervised Text Classification [4] problems. Correct features used for classification can significantly improve the performance of Machine Learning algorithms. Also as only important features are considered, it often reduces training as well as prediction time. Feature selection can be manual or automatic. Though manual feature selection is good especially in case of text based classification problems, it involves *2* issues:

(i) Time for selecting the right features could be long depending on the type of classification problem, expertise of the Oracle[2] and the number of features

(ii) Domain knowledge of human expert(s) is necessary from the point of view of how it affects classification

Automatic feature selection do not rely upon domain expertise. Instead they take into account various Feature Selection Algorithms [8] to understand which features truly classification accuracy. Thus, it saves time and effort by ranking all the features according to their importance in determining the class label of a particular data sample.

χ^2test, Information Gain and GainRatio are popular Feature Selection Algorithms widely used. Selecting the most important features is the key to most efficient classification algorithm. Though SVMs do not actually depend on the quantity of features, they do map the given feature space to kernel space. Typically Feature Selection is done only once at the beginning before training on the labeled data. Feature Selection also reduces dimensionality of the Feature Space (*FS*) leaving only the most important features for consideration. Usually, the words occurring in documents are considered as features representing the document. Thus feature selection identifies the words that contribute the most towards determining class label of a particular document.

In this paper, we study the possibility of integrating feature selection into active Machine Learning process. We start by analyzing typical active learning process as practiced by the leading researchers.We also pay close attention to the process of feature selection. In section *2* , we observe individual strengths and advantages of using active learning and feature selection respectively. In the experiment section, we present our experimental results with conclusion. Finally, we present analysis of results and conclusion. Please refer to all tables and figures at the end of this paper in the Appendix section.

2. INTEGRATING FEATURE SELECTION AND ACTIVE LEARNING

We consider the problem of Text Classification for the blog dataset of *3.2 million* blog posts. After removing stop words and common words, we have about *3 million* words. The words such as aaarrgghhh, aaah etc. are not uncommon in blog posts. Thus it is important to reduce feature space by using Feature Selection algorithm. We have to find feature sub-space of the actual feature space that represents the data in a manner that can be exploited for classification problems. Also the blog dataset provided [3] is not labeled. Hence the traditional supervised Machine Learning algorithms could not be employed. We need learning algorithm capable of learning from small data and ask for which more samples need to be labeled to increase its accuracy. Active learning allows shorter training time and higher accuracy with small number of training data.

We consider active learning ideal for classification of huge unlabeled data because it allows iterative training of only those samples that the classification algorithm finds the most difficult to predict. We used SVMs for active learning based classification. Using various kernels, the feature space of the given data samples could be mapped the kernel space where the classes might be separable. We find the distance of all samples from the SVM decision boundary and train SVM only on those samples that are closest to the decision boundary. The rationale behind selecting the data points closest to the decision boundary is that the data points (or blog posts in this case) that are closest to the decision boundary are the most difficult for SVM to predict. Hence iteratively training SVM on only closest points to the decision boundary, redraws the decision boundary so as to accommodate the newly labeled instances. This ensures increasing training accuracy till the algorithm achieves a consistent accuracy. Active learning techniques as explained earlier are known for quick learning convergence to higher accuracy levels.

We introduce the idea of using feature selection integrated within the active learning process. The feature selection has been used at the beginning of active learning to limit the feature space. Raghavan et. al [1] used manual feature selection with active learning to take advantage of smaller feature space as well as higher accuracy in short time. Active feature sampling [7] has also been researched by Veeramachaneni et. al. where they identify the features important for

[2]Oracle is the human expert involved in Supervised Learning

[3]Blog Track at http://trec.nist.gov

classification on the fly. We propose integrated feature selection into active learning process. For each iteration of active learning, feature space can be reduced. Our initial results by using GainRatio Feature selection integrated into active learning of blogs did not yield good results. Hence we decided to investigate the reason for poor performance using relatively smaller dataset for which all class labels are known. We used *20 newsgroup dataset* [4] for all experiments described in this paper. The *20 newsgroup dataset* has been widely used in Text Classification tasks. We considered binary classification problem between two newsgroups namely *rec.sport.baseball* and *sci.crypt*. We used linear kernel SVMs and GainRatio Feature selection technique for all the results presented here. In the next section we will discuss the details of experiments conducted and discuss the results.

3. EXPERIMENTS

rec.sport.baseball and *sci.crypt* newsgroups consist of *1,000* newsgroup postings each. From *2,000* newsgroup posts, we found *21,423* unique words after removing stop words and common words. No stemming was performed. The reported accuracy of this particular classification using linear kernel SVMs in the literature is 95.8959%. This accuracy reported is using all features and entire dataset (2000 samples) for training and 10 fold cross validation. We used libSVM software [5] with modified source code in order to find distance of all points from decision boundary.

For all experiments discussed in this section, we consider 10 fold cross validation accuracy and actual testing accuracy. The accuracy reported in most of the literature regarding active learning is *10* fold cross validation accuracy. In many practical applications, actual testing accuracy is not known due to various reasons such cost of labeling, size of test data etc. In the case of *rec.sport.baseball* and *sci.crypt* newsgroups the test data class labels are already known. Table *1* shows traditional active learning accuracy as reported by Tong et. al. The training time is short and training dataset is small as well. If we consider human labeling cost to label each sample, active learning will save a lot of time and money in this case.

From Table *1*, a clear increasing trend can be seen in the column labeled *10 fold training accuracy*. Accuracy increases rapidly and achieves consistently high accuracy levels with few training samples. We start with 20 randomly selected [6] samples. In each iteration we add 20 new samples labeled to the training set as determined by the distance from decision boundary. As shown in Figure 2, so called '*knee effect*' can be observed[3] when the number of training samples is *60* or *80* samples. Thus, it can be concluded that training accuracy of over *96%* can be achieved with as little as *60* or *80* training samples. We would like to note that though *10* fold cross validation accuracy reflects the actual testing accuracy, it does not necessarily follow actual testing accuracy closely. High accuracy of *95%* was observed when we continue active learning iteratively until *280* samples have been labeled. Next, we present different feature space reduction approaches and their accuracy results in comparison with conventional active learning.

3.1 *Adding all features iteratively from training samples*

In this experiment, we considered all features from the training samples. In the original experiment, all features from training as well as test data were considered. Here the assumption is that the training data and its features will represent the actual feature space. Consider Table *2* for more details.

We used 1,122 features from initial 20 newsgroup posts randomly selected (*10* from each class) and added features from newly selected *20* samples that are closest to the decision boundary. It is evident from Figure *3* that the *10* fold cross validation accuracy as well as testing accuracy are poor and inconsistent through iterations. We believe that the feature space represented by training samples was not separable for the classification task. Insufficient feature space representation resulted in poor accuracy levels. We refer readers to section *4* for more discussion on why we think this approach did not work.

3.2 *Using all features with Document frequency*

We use Document Frequency, $DF = 2$ in these experiments. Experimental results indicate that iteratively increasing the size of feature space according to training samples doe not have advantage with active learning approach. One plausible reason for poor accuracy could be that the training data was not able to represent important features for classification. In this sub section we discuss the results obtained by using most frequently occurring words in the two newsgroups (See Table *3*). This is similar to conventional active learning except that instead of all features we use only most commonly occurring features. We compiled the list of *11,666* from the entire dataset which occur in at least 2 newsgroup postings. These words are the most common features to represent the feature space. Most common is not the same as most important. The assumption here is that most commonly occurring features will certainly represent the entire feature space better than the previous approach. In this case, the number of features is constant and so the feature space representation does not change with iterations. We did notice improved accuracy levels compared to the earlier approach (sub section *3.1*). Figure *4* shows the different levels with this approach.

3.3 *Using Top 100 features ranked by GainRatio Feature Selection*

Motivated by the increased accuracy obtained using most frequently occurring features, we decided to exploit the GainRatio feature selection algorithm. We Used only top *100* features ranked from the 20 samples selected according to the active learning distance criterion. With each iteration top *100* new features were added. As Table *4* and Figure *5* will indicate, we did not achieve expected accuracy levels. On a side note, selecting top *200* ranked features instead of *100* did not improve accuracy much.

Consider two samples A and B representing Feature Space FS_0 as shown below:

$$A = \{f_1, f_2, f_3, f_4\} \Rightarrow +1$$

$$B = \{f_1, f_3, f_5\} \Rightarrow -1$$

[4]http://people.csail.mit.edu/jrennie/20Newsgroups/
[5]http://www.csie.ntu.edu.tw/
sim cjlin/libsvm
[6]10 samples are selected from *rec.sport.baseball* and *sci.crypt* each

If we select features that occur in at least 2 samples, then

$$A = \{f_1, f_3\} \Rightarrow \textit{+1}$$

$$B = \{f_1, f_3\} \Rightarrow \textit{-1}$$

which contradicts the class label prediction. For the same features f_1 and f_3, we are training the SVM to predict classes labels *+1* and *-1* at the same time. This certainly decreases the accuracy of SVMs. In a second case, let us consider two more samples such that

$$C = \{f_4, f_5, f_6\} \Rightarrow \textit{+1}$$

$$D = \{f_7, f_8, f_9\} \Rightarrow \textit{-1}$$

Suppose in the iteration-*0*, only samples A and B are selected. Then the Feature Space, $FS_0 = \{f_1, f_2, f_3, f_4, f_5\}$ consists of only *5* features. Using Feature Space FS_0, samples C and D are represented as $C = \{f_4, f_5\}$ and $D = \{\emptyset\}$ Thus with reduced feature space either represents data incorrectly (sample C) or insufficiently (sample D).

In the next section, we analyze the impact of integrating Feature Selection into active learning process and present the rationale behind poor accuracy results.

4. ANALYSIS AND DISCUSSION

Figure *6* shows comparison of *10* fold cross validation accuracy of *3* aforementioned approaches with conventional active learning approach. The blue dotted line with big squares is the baseline accuracy obtained by using all features and entire dataset for training.

Figure *7* shows the comparison of testing accuracy of the *3* approaches with the typical active learning accuracy response with increasing training samples. Testing accuracy does not fluctuate as much as the 10 fold cross validation accuracy, probably because the testing data size is about *10* times the size of any of the folds used for cross validation. Overall trend is upward indicating increase in accuracy. Conventional active learning still outperforms any of the proposed approaches.

Feature space reduction does not seem to work well when used integrated with active learning. We would like to find out why the highly regarded active learning approach does not work well when integrated with feature selection. For typical high dimensionality Machine Learning problems, feature selection is necessary. In fact, feature selection in some cases can reduce the noise, thus increasing classification accuracy. In order to analyze the problem, let us assume that in iteration-*0*, *20* samples S_0 were randomly selected and represented in Feature Space FS_0,

$$FS_0 \subseteq FS$$

where FS is the feature space of all features.

Active learning approach finds next *20* samples closest to the decision boundary by mapping Feature Space FS_0 to the kernel space. *Twenty* new samples, S_new selected contribute new features to the Feature Space FS_new. When the two feature space namely, FS_0 and FS_new are combined, we get joint feature space FS_1

$$FS_1 \subseteq FS$$

in iteration-*1*. We believe that samples S_new which were closest to the decision boundary in FS_0 are no longer closer to the boundary when mapped in feature space FS_1. This is because the feature space changed from the time closest samples were identified and the same samples are no longer closer to the decision boundary. It is crucial to the success of active learning algorithm to make sure that the algorithm always learns labels of the samples closest to the decision boundary (difficult to predict). With new Feature Space FS_1, the selected samples S_new are no longer closer to the decision boundary. In order to prove this we monitored the distance of the selected samples S_new in iteration- *0* again using conventional active learning approach. We found that the particular newsgroup post id *611* was closest to the decision boundary in FS_0 (See Table 6). Next, we find distances of all samples in S_new again, this time mapped using Feature Space FS_1 rather than original Feature Space FS_0. We find that newsgroup posting id *611* is no longer closest the decision boundary. In fact, there are as many as *90* other samples that are closer to the decision boundary than post id *611*. According to new feature space FS_1, newsgroup posting id *923* is closest to the decision boundary. Post Id *923* did not figure in top *20* samples closest in Feature Space FS_0 and hence was not selected for active learning. This proves our hypothesis that the reason for the poor accuracy levels with active learning and integrated feature space reduction approach is the changing Feature Space itself. The two ideas collide here. The active learning algorithm needs constant feature space to predict boundary distances and integrating feature selection keeps changing feature space.

5. CONCLUSION

In this paper we study the effect of integrating Feature Selection into active learning process. We present results using different feature space reduction approaches to corroborate our claim that feature selection is not well suited to be integrated into active learning. Finally, we also analyze and prove the reason for why exactly the Feature Selection fails when integrated into active learning. Two approaches of active learning and feature selection when integrated do not compliment each other because of their inherent properties.

In future, we would like to investigate other possible feature space reduction techniques to try and improve the accuracy of active learning approach by using smaller feature space representation.

6. REFERENCES

[1] R. J. Hema Raghavan, Omid Madani. Interactive feature selection. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence*, 2005.

[2] T. Joachims. Text categorization with suport vector machines: Learning with many relevant features. In *ECML '98: Proceedings of the 10th European Conference on Machine Learning*, pages 137–142, London, UK, 1998. Springer-Verlag.

[3] G. Schohn and D. Cohn. Less is more: Active learning with support vector machines. In *Proc. 17th International Conf. on Machine Learning*, pages 839–846. Morgan Kaufmann, San Francisco, CA, 2000.

[4] F. Sebastiani. Machine learning in automated text categorization. *ACM Comput. Surv.*, 34(1):1–47, 2002.

[5] S. Tong and D. Koller. Support vector machine active learning with applications to text classification. *J.*

Mach. Learn. Res., 2:45–66, 2002.

[6] V. Vapnik. *The Nature of Statistical Learning Theory.* Springer, N.Y., 1995.

[7] S. Veeramachaneni and P. Avesani. Active sampling for feature selection. In *ICDM '03: Proceedings of the Third IEEE International Conference on Data Mining*, page 665, Washington, DC, USA, 2003. IEEE Computer Society.

[8] Y. Yang and J. O. Pedersen. A comparative study on feature selection in text categorization. In *ICML '97: Proceedings of the Fourteenth International Conference on Machine Learning*, pages 412–420, San Francisco, CA, USA, 1997. Morgan Kaufmann Publishers Inc.

[9] L. Yu and H. Liu. Efficient feature selection via analysis of relevance and redundancy. *J. Mach. Learn. Res.*, 5:1205–1224, 2004.

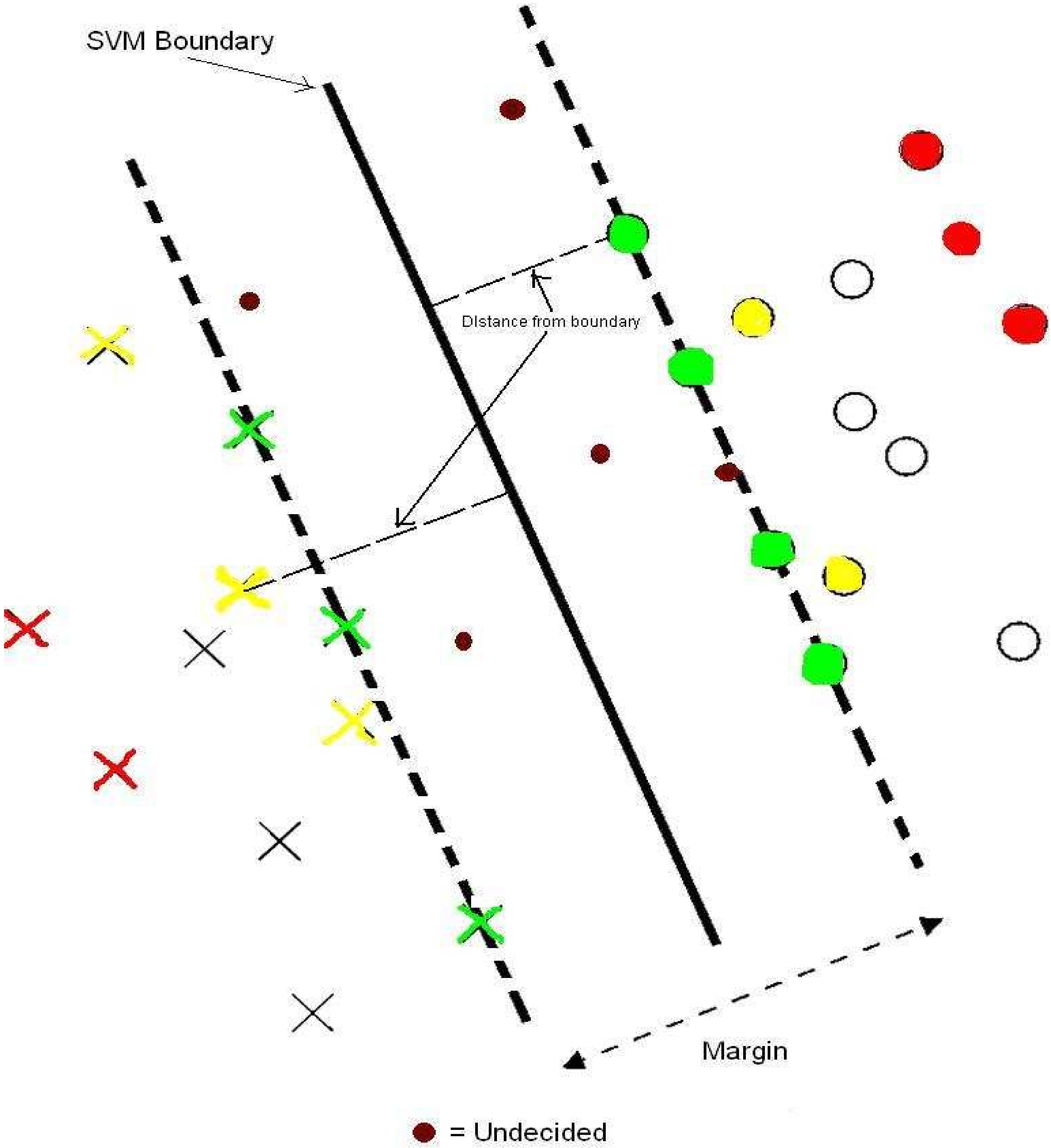

Figure 1: Active Learning Approach with Support Vector Machines (Linear kernel space)

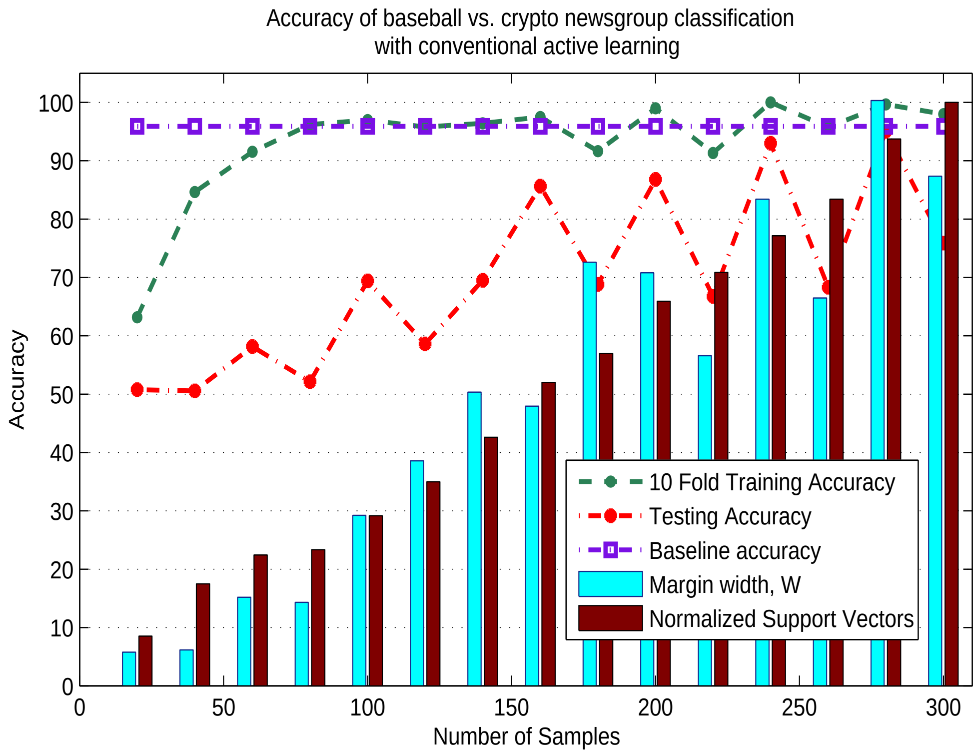

Figure 2: Baseball vs. crypto results obtained using conventional active learning approach

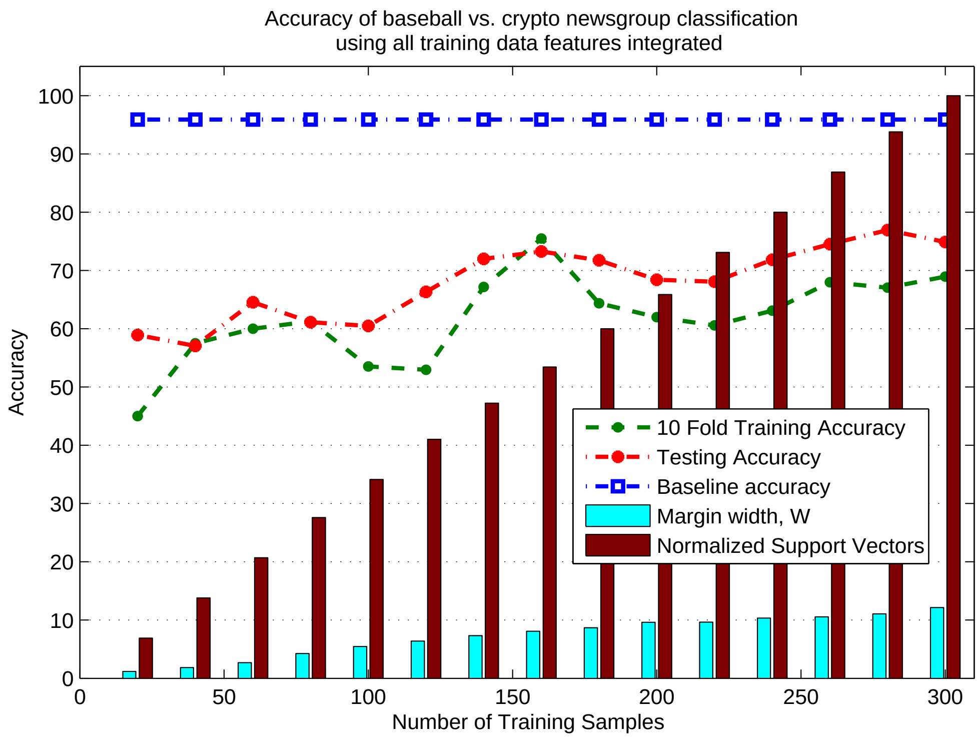

Figure 3: Active Learning results obtained by integrating all features from training data.

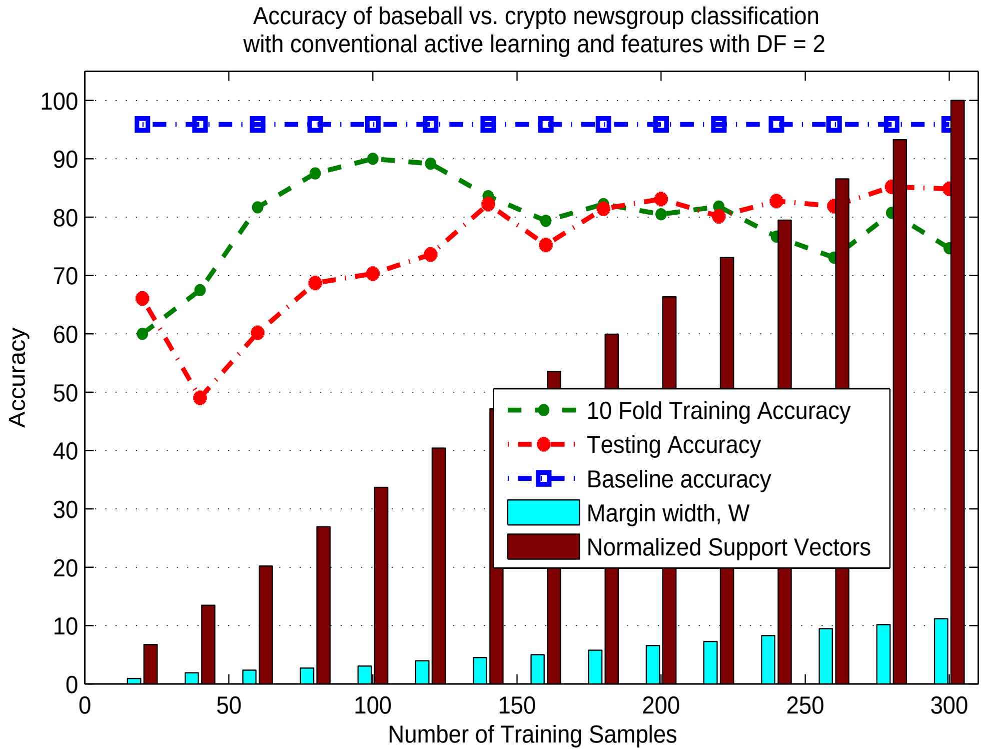

Figure 4: Active Learning results obtained by integrating all features from training data.

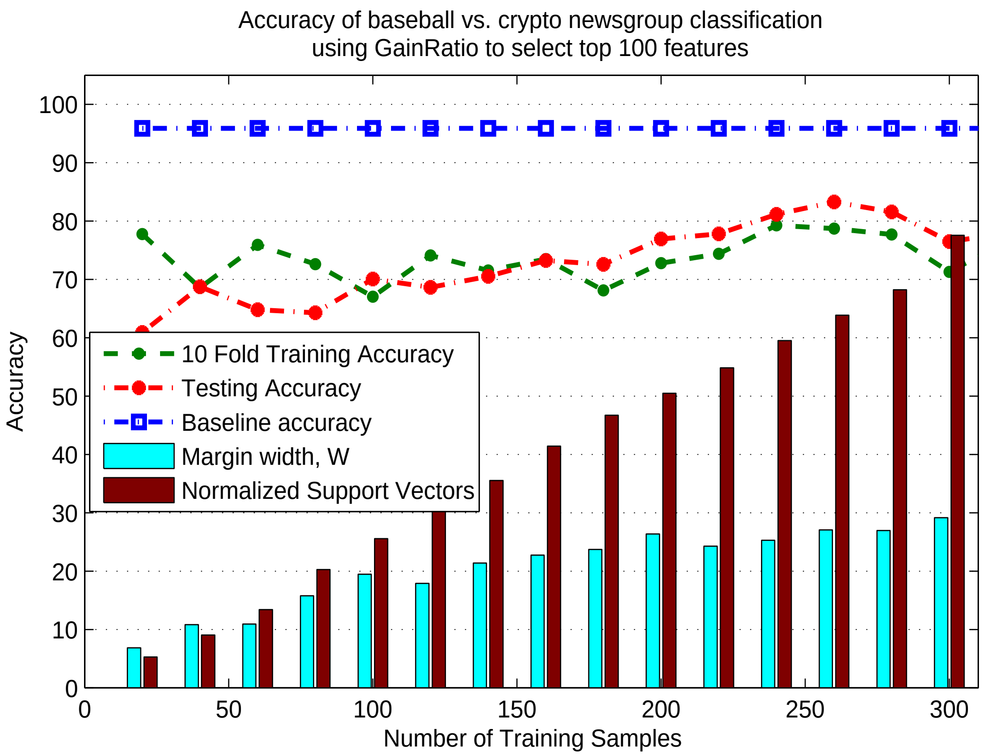

Figure 5: Accuracy of baseball vs. crypto newsgroup classification using top 100 features integrated ranked by GainRatio Feature Selection Algorithm

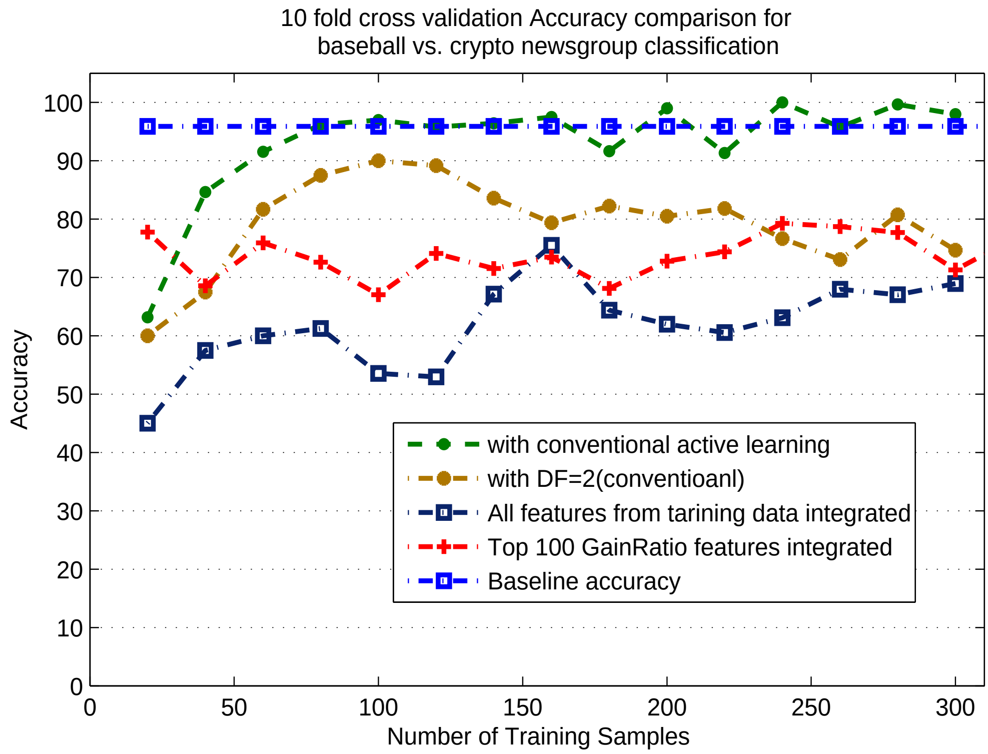

Figure 6: Comparison of 10 fold cross validation training accuracy for baseball vs. crypto classification

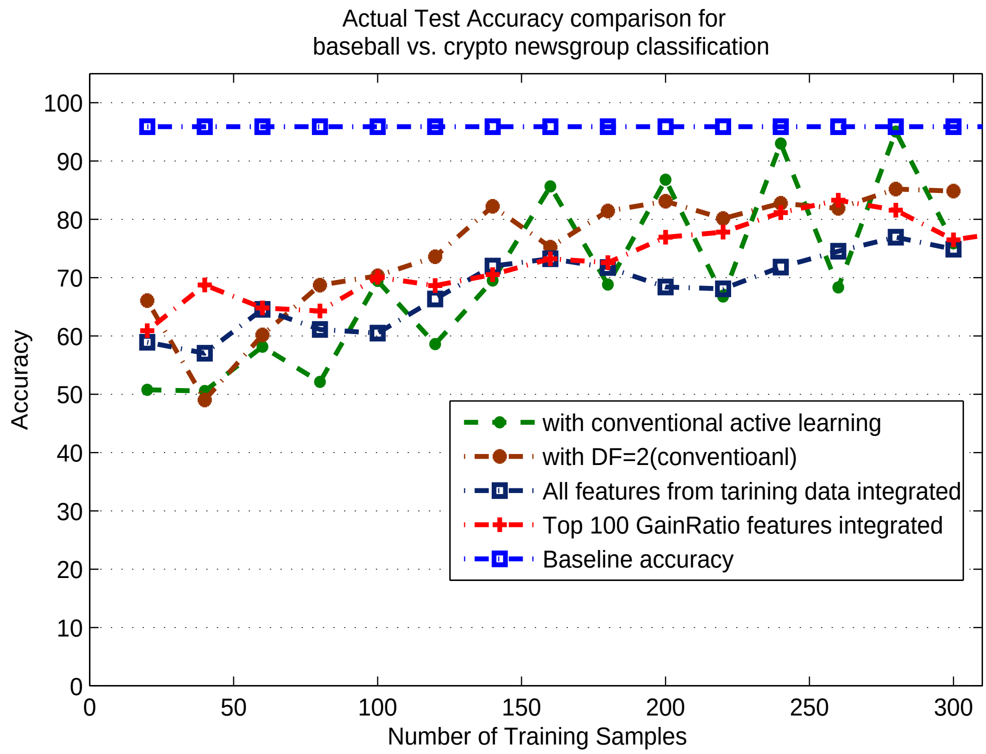

Figure 7: Comparison of testing accuracy for baseball vs. crypto classification

Table 1: Results of baseball vs. crypto with conventional active learning

No. Samples	No. of Support Vectors	Width of Margin, W	Training accuracy	Test accuracy
20	19	5.75814	63.1579	50.7508
40	39	6.155959	84.6154	50.5506
60	50	15.173018	91.5254	58.1582
80	52	14.295347	96.2025	52.1021
100	65	29.215246	96.9697	69.4194
120	78	38.548423	95.7983	58.6086
140	95	50.335364	96.4029	69.5195
160	116	47.941245	97.4843	85.6356
180	127	72.623777	91.6201	68.8188
200	147	70.78726	98.995	86.7868
220	158	56.579423	91.3242	66.7668
240	172	83.423831	100	92.993
260	186	66.491486	95.7529	68.3183
280	209	100.327119	99.6416	95.045
300	223	87.352862	97.9933	75.8759

Table 2: Results of baseball vs. crypto using features from only training samples integrated in active learning

No. Samples	No. of Support Vectors	No. of Features	Width of Margin, W	Training accuracy	Test accuracy
20	20	1.20407	1122	45	58.9007
40	40	1.855487	2050	57.5	57.0352
60	60	2.706077	2563	60	64.5582
80	80	4.272734	2997	61.25	61.1222
100	99	5.459783	3523	53.5354	60.4709
120	119	6.400225	4238	52.9412	66.3327
140	137	7.334479	4824	67.1533	71.994
160	155	8.084401	5248	75.4839	73.2465
180	174	8.693689	5603	64.3678	71.7435
200	191	9.614243	6071	61.9792	68.3868
220	212	9.661555	7129	60.5634	68.0681
240	232	10.343336	7626	63.0901	71.8218
260	252	10.56318	8243	67.9842	74.5245
280	272	11.066488	8490	67.033	76.9269
300	290	12.136982	9397	68.942	74.8749

Table 3: Results of baseball vs. crypto using all features with minimum Document Frequency, DF = 2 (Please note that features were *not* integrated into active learning process)

No. Samples	No. of Support Vectors	No. of Features	Width of Margin, W	Training accuracy	Test accuracy
20	20	0.91004	11666	60	66.0661
40	40	1.892279	11666	67.5	48.999
60	60	2.344454	11666	81.6667	60.1602
80	80	2.700645	11666	87.5	68.7187
100	100	3.052834	11666	90	70.3203
120	120	3.952858	11666	89.1667	73.5736
140	140	4.497396	11666	83.5714	82.2322
160	159	5.011395	11666	79.375	75.2252
180	178	5.767384	11666	82.2222	81.4314
200	197	6.557269	11666	80.5	83.0831
220	217	7.252904	11666	81.8182	80.1301
240	236	8.262037	11666	76.6667	82.7327
260	257	9.473864	11666	73.0769	81.8819
280	277	10.160205	11666	80.7143	85.1852
300	297	11.154414	11666	74.6667	84.8348

Table 4: Results of baseball vs. crypto using top 100 features ranked by GainRatio Feature Selection integrated into active learning

No. Samples	No. of Support Vectors	No. of Features	Width of Margin, W	Training accuracy	Test accuracy
20	17	6.840158	100	77.7778	60.8851
40	29	10.836183	200	68.5714	68.7243
60	43	10.908059	300	75.9259	64.7973
80	65	15.765109	400	72.6027	64.2933
100	82	19.46107	500	67.033	70.0684
120	102	17.881529	600	74.1071	68.6366
140	114	21.362983	700	71.5385	70.5095
160	133	22.753337	800	73.4694	73.2344
180	150	23.725653	900	68.0982	72.579
200	162	26.367851	1000	72.7778	76.9348
220	176	24.25711	1100	74.3719	77.7834
240	191	25.29436	1200	79.2627	81.1579
260	205	27.069396	1300	78.7234	83.2572
280	219	26.940332	1400	77.6892	81.5603
300	249	29.149229	1500	71.2687	76.4914
320	254	28.880698	1600	76.6551	77.9181
340	264	28.268802	1700	79.1391	82.4659
360	271	26.659014	1800	82.1317	85.1515
380	290	26.790886	1900	77.7778	81.887
400	321	29.08536	2000	73.487	82.006

Table 5: Comparison of documents ranked closest to the decision boundary in Run-0 (Feature Space FS_0) and Run-1 (Feature Space FS_1)

Closest docs ranked in Run-0	Closest docs ranked in Run-1
611	923
794	1920
812	783
1691	925
1193	927
956	1289
1663	1794
1356	318
881	1029
1062	1037
1161	1048
1106	1084
1708	1097
1730	1188
1422	1274
12	1351
421	1408
1667	1421
606	1543
607	1569