Active Learning with Committees for Text Categorization

Ray Liere
lierer@research.cs.orst.edu

Prasad Tadepalli
tadepall@research.cs.orst.edu

Department of Computer Science, Oregon State University,
Dearborn Hall 303, Corvallis, OR 97331-3202, USA

Abstract

In many real-world domains, supervised learning requires a large number of training examples. In this paper, we describe an active learning method that uses a committee of learners to reduce the number of training examples required for learning. Our approach is similar to the Query by Committee framework, where disagreement among the committee members on the predicted label for the input part of the example is used to signal the need for knowing the actual value of the label. Our experiments are conducted in the text categorization domain, which is characterized by a large number of features, many of which are irrelevant. We report here on experiments using a committee of Winnow-based learners and demonstrate that this approach can reduce the number of labeled training examples required over that used by a single Winnow learner by 1-2 orders of magnitude.

1. Introduction

The amount of textual information that is available in electronic form has grown exponentially in recent years. Automating the task of indexing, categorizing, and organizing these electronic documents will make it easier and cheaper for people to find relevant written materials.

The goal of *text categorization* is to assign each document to the appropriate categories, based on the semantic content of the document. A knowledge engineering approach to text categorization involves designing rules that correctly categorize the documents. Our goal is to develop automatic methods for text categorization through the application of machine learning techniques.

The text categorization domain has several characteristics that make it a difficult domain for the use of machine learning, including a very large number of input features (10,000+), high levels of attribute and class noise, and a large percentage of features that are irrelevant. As a result, the use of supervised learning requires a relatively large number of labeled examples.

We have been working on developing methods that will dramatically reduce the number of labeled examples needed in order to train the system, without incurring unacceptable decreases in prediction accuracy. Our approach so far has utilized very little in the way of any preprocessing that is specific to the handling of text, so we believe that many of our results will also apply to other similarly-difficult machine learning domains.

Active learning refers to machine learning methods that allow the learning program to exert some control over the examples on which it learns [Cohn94]. *Query by Committee* (QBC) is one specific type of active learning which starts with a committee of all possible hypotheses. Each feature vector is presented to the committee. A high degree of disagreement among the hypotheses as to the predicted value of the label indicates that the example will be very informative, and so the actual label is requested. The label is then used to remove all hypotheses from the committee that do not predict the actual label [Freund92, Seung92, Freund95].

The learning methods which we have investigated are similar to QBC, in that they use disagreement among the committee members to determine the need for requesting the actual value of each example's label from the teacher. Unlike QBC, our committee consists of a small finite number of hypotheses, which are updated with learning. We use Winnow as the learning algorithm. A small learning rate is used to make the method robust against noise. We use majority voting to determine the prediction of the committee.

The purpose of this paper is to present results of experiments that demonstrate the effectiveness of active learning with committees, and also to analyze the sources of this effectiveness. We performed experiments on 4 different systems which vary in terms of whether or not they use active learning and whether or not they use committees for prediction. Our experiments indicate that active learning with committees can, as compared to supervised learning with a single learner, result in learning methods that use far fewer labeled examples but still achieve the same accuracy.

2. Previous Research

2.1 Active Learning

"Active learning" in its most general sense refers to any form of learning wherein the learning algorithm has some degree of control over the examples on which it is trained. One active learning approach is the membership query paradigm, in which the learner can construct new sets of inputs and request that the teacher provide their labels [Angluin88]. In this paper, we are specifically considering the type of active learning in which there exists a set of examples, and the learner chooses which of these it will use for learning. Typically, the cycle proceeds as follows. The teacher presents the learner with the feature vector portion of an example (i.e., the example without the label). The learner examines the feature vector and then decides whether or not to ask for the label. If the label is requested, then the example is considered as having been used as a training example by the learner.

There have been some promising results in the active learning area. Cohn, Atlas, and Ladner developed the theory for an active learning method called *selective sampling* and then applied it to some small to moderate sized problems as a demonstration of the viability of this new approach [Cohn94]. Lewis and Gale developed a method called *uncertainty sampling*, which is similar conceptually to selective sampling, but which is specifically meant for use in text categorization. Their method selects for labeling those examples whose membership is most unclear by using an approximation based on Bayes Rule, certain independence assumptions, and logistic regression. Since the method was developed for text categorization, it is able to handle noise as well as large numbers of features [Lewis94].

While approaches and results vary, these and other studies have concluded that active learning greatly improves learning efficiency by reducing the number of labeled examples used [Board87, Freund92, Dagan95].

2.2 Query by Committee (QBC)

Query by Committee (QBC) is a learning method which uses a committee of hypotheses to decide for which examples the labels will be requested. It also uses the committee to determine the prediction of the label. Since QBC exerts some control over the examples on which it learns, it is one form of active learning.

QBC maintains a committee of hypotheses consistent with the labeled examples it has seen so far – a representation of the version space. For many real-world problems, the committee is infinite. Each training example is presented to the algorithm unlabeled. An even number of hypotheses (usually 2) are chosen at random, given the attribute values, and asked to predict the label. If their predictions form a tie, then the example is assumed to be maximally informative, the algorithm requests the actual label from the teacher and updates the version space [Freund92, Seung92, Freund95].

Freund, Seung, Shamir, and Tishby analyzed QBC in detail and showed that the number of examples required in this learning situation is logarithmic in the number of examples required for random example selection learning [Freund92]. Dagan and Engelson proposed a similar method, termed *committee-based sampling*, for selecting examples to be labeled [Dagan95]. The informativeness of an example (and so the desirability of having it labeled) is indicated by the entropy of the predictions of the various hypotheses in the committee.

3. Approach

Our *active learning with committees* approach uses a form of QBC for deciding whether or not to see the label, Winnow for updating the hypotheses in the committee, and majority voting for prediction of the labels for the test examples. Although it may not be surprising that the choice of good examples allows one to learn with fewer examples, it is not easy to know how to *select* good examples, especially in the presence of noise. Random selection of examples is no better than passive learning.

3.1 Deciding to See the Label

QBC offers the benefit of a logarithmic reduction in the number of labeled training examples needed. However, QBC needs to maintain all possible hypotheses consistent with the training data – the version space – in some form [Seung92]. This is the committee. When data is noisy, this will not be possible. When there is a very large number of candidate hypotheses, explicitly representing them will not be practical. In text categorization, we have data that is noisy. Because of the large number of attributes, text categorization typically also entails a large number of possible hypotheses.

Our approach is to use a committee with a small number of hypotheses. Once presented with an unlabeled example, we do the following: two randomly chosen members of the committee are given the unlabeled example and asked to predict the label. If their predictions disagree, then we ask to see the actual label.

3.2 Updating the Hypotheses

After the label is seen, the learners adjust the hypotheses in the committee. Typically, each member of the committee learns individually. We chose Winnow as the learning algorithm [Littlestone88]. Winnow is especially suited to large attribute spaces and to situations in which there is a large percentage of irrelevant features. Winnow also has a relatively low space and time complexity and is easy to implement. And, Winnow has been used successfully in other noisy text-based applications [Roth96].

Actually, "Winnow" refers to a quite large family of algorithms [Littlestone89]. We have thus far used one of the more general (and simpler) Winnow algorithms – WINNOW2 in [Littlestone88], with some modifications from [Littlestone91]. We will hereafter refer to the algorithm that we use as simply "Winnow".

Conceptually, think of each document as being represented by a data point in some feature space. What the Winnow algorithm does is try to pass a hyperplane through the "cloud" of document data points so that the points representing all of the documents that *are* in the specified category lie on one side of the hyperplane, and the points representing all of the documents that are *not* in the category lie on the other side of the hyperplane. If there exists such a separating hyperplane, then the data is termed *linearly separable*. Knowing the equation of this hyperplane, we can predict the category membership of new documents by simply seeing on which side of the hyperplane they fall.

Winnow starts with the hyperplane in some initial location and then adjusts the location of the hyperplane gradually as it learns. The hyperplane is moved by multiplying the coefficients in its equation by a constant (which is one of the parameters given to Winnow). A Winnow learner modifies the location of the hyperplane only when it encounters an example that it does not already classify correctly.

3.3 Committee Prediction

Once the learning process has been completed, the committee needs to make predictions for previously unseen inputs. The idea behind using a committee to make predictions is that a committee of several members might be able to outperform a single member [Freund92, Seung92, Breiman96]. Each member predicts a label, and these votes are then combined using majority vote.

4. Experimental Results

4.1 The Systems to be Compared

We examined the performance of 4 different learning systems. We not only compared them to see which system is "the best", but we also looked into why. In order to meet this second goal, we constructed the systems in somewhat of a boolean building block fashion, where each system does/does not have particular features.

The systems are:

– *active-majority*: the learner is a committee of Winnows which uses disagreement between two randomly chosen members to determine which labels to obtain from the teacher. Prediction is made by that same committee, using majority rule.

– *passive-majority*: the learner is a committee of Winnows which passively accepts all labels from the teacher. Prediction is made by that same committee, using majority rule.

– *active-single*: the learner is a committee of Winnows which uses disagreement between two randomly chosen members to determine which labels to obtain from the teacher. However, in the prediction phase, only a single member of the committee is used. For all predictions in a particular trial, a committee member chosen at random makes the predictions.

– *passive-single*: the learner is a single Winnow which passively accepts all labels from the teacher. Prediction is by that same Winnow. This can be thought of as the "base case" – a single supervised learner and predictor.

4.2 Test Bed

All of our experiments were conducted using the titles of newspaper articles from the Reuters-22173 corpus [Reuters], hereafter "Reuters". The Reuters corpus is a collection of 22,173 Reuters newswire articles ("documents") from 1987. It is a 25Mb full text corpus. Each article has been assigned to categories by human indexers. Typical categories are "grain", "gold", "Canada", and "trade". An article may be assigned to any number of categories, including none.

The Reuters-22173 corpus contains formatting errors, misspellings, and garbled/missing/reordered sections. This is actually good, in that it is typical of most real-world data.

We were very conservative in how we preprocessed the data. We specifically did *not* correct any misspellings, either in the text of the articles, or in the names of the categories assigned to each article. Neither did we remove any illegal category names. The preprocessing step converts the raw data in the corpus into sets of labeled examples. The preprocessor unpacks the corpus, performs a rough structural parse, separates text and category information, constructs a table of existing categories, tokenizes the text, constructs a dictionary of text tokens, and prepares the labeled examples. In these experiments, each token in a document is a feature. Each feature is boolean-valued – either the token does or does not appear in the document. The labels are also boolean-valued – the document either is or is not in each category.

There are several possible tokenizing methods. The experiments reported in this paper tokenize text by separating the text stream at whitespace or punctuation.

Full corpus statistics for Reuters (after our preprocessing):

– 22,173 documents
– 21,334 unique tokens in titles (maximum – the actual number depends on the tokenizing method used)
– 679 categories

4.3 Repeated Trials

A variety of approaches have been utilized in previous research using the Reuters corpus [Hayes90, Lewis91, Apte94]. There are some differences among researchers as to which articles in the corpus are used, and also there are differences in how the corpus is split into training and test sets. Normally researchers use one of 3 standard corpus setups, and so it is predetermined which articles will be used for training, which will be used for testing, and which will not be used at all. Procedurally, the training and test sets are constructed, and then the system being examined learns and is tested in the normal manner. Typically a single trial is executed for each category, and the results are averaged over all categories to obtain a measure of overall system performance.

We are mainly interested at this point in comparisons among our learning systems. In particular, we want to compare their performance for categories most likely to have ample training data. We used the 10 most frequently occurring topic categories, as listed in [Lewis91], for our experiments. We performed repeated trials for each category, using randomly chosen training-test splits. We used the entire corpus, and split it into 21,000 training examples and 1,173 test examples. We used titles only for our tests.

For our experiments, we did the following. We generated a random training-test split. Then for each of the categories and for each of the systems, we ran a trial – we trained the system, gathering data during and after training.

We analyzed the data gathered while learning was occurring as well as the final results. The main results that we examined were the number of labeled training examples used, accuracy (fraction of documents correctly classified), elapsed processor time used, space used during learning, and space used during prediction. Our main statistical tool was the repeated measures analysis of variance test, hereafter "anova". The null hypothesis is that all of the systems actually perform the same on the average. We performed single factor tests, with the factor being the system.

All of the systems were initialized in the same manner, so that comparisons among the systems would be fair. How one does initialization is important in several learning algorithms, including Winnow. We compute initial positions of the hyperplanes so that they approximately bisect the space of all *possible* data points. We can compute this knowing only the number of attributes in the data. We chose this method because it allows the learners to start the learning process at a reasonable location, and it does not use any information about any actual data values. Individual committee members are randomly initialized to slightly different hyperplanes so that they represent different initial hypotheses.

We used a committee with 7 members. This value was determined by trial and error. We found that using a very large committee increased the space and time complexity of the algorithm without any significant increase in accuracy. With a large number of committee members, we obtained large amounts of duplication. We ran tests with as many as 1,000 committee members, but normally found that, in terms of committee predictive behavior, there were perhaps only 5 - 20 different prediction patterns. We found that using a very small committee resulted in the committee becoming very sensitive to 1 or perhaps 2 of the members, and the behavior of the committee began to approach that of a single member.

4.4 Results

Figure 1 shows elapsed processor time as a function of the number of training examples used, for each of the 4 systems. There is one dot on the graph for each trial. Observe that, for each system, the dots form very tight clusters. We have added labeled boxes around each of the clusters. (Since the passive-majority and passive-single systems are supervised learners, their boxes collapse into lines). Figure 1 dramatically illustrates that, in these experiments, the differences among the systems in terms of both the number of labeled examples used and the elapsed execution time were quite large. Figure 1 also shows that the variation in the behavior within each system, for both the number of training examples used and elapsed processor time, was quite small.

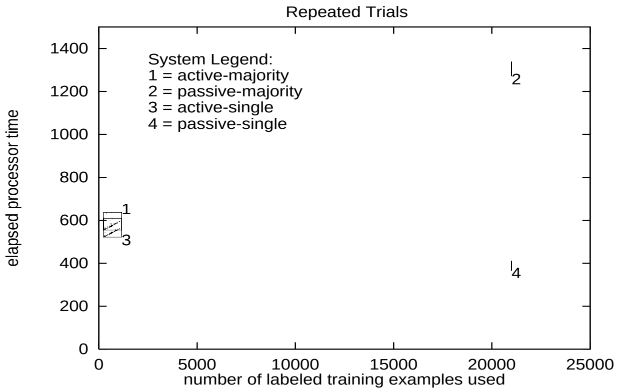

Figure 1: Elapsed Time versus Number of Training Examples Used

Anova indicates that which system one uses has a significant effect on both the number of training examples used ($p < 0.0001$) and on the elapsed execution time ($p < 0.0001$).

Note that, from Figure 1, one can also conclude that if one has an abundant source of cheap labeled examples, then perhaps the passive-single system is the best choice. It uses the least amount of processor time (on the average). However, one often does not have as many labeled examples as one wants. In these situations, active learning (the active-majority and active-single systems) is beneficial. From Figure 1, it would appear that the 2 active learning systems are very similar, in that they have similar average values both of number of training examples used and elapsed processor time. How would one choose between these 2 systems?

Figure 2 shows the average accuracy for each of the 4 systems as a function of the number of training examples used. This is a learning trace, showing how accuracy varies for each system as it learns. (Note the use of a log scale). We can see that the systems employing active learning use many fewer examples than those using supervised learning, which is consistent with the results in Figure 1. However, Figure 2 also shows that the 4 systems end up with very similar final accuracies, while the path that each takes to get there is different. The fact that the systems are all about the same in terms of final accuracy justifies our looking at other system characteristics (such as number of training examples used and elapsed processor time) as metrics on which to base our comparison of the systems. As regards the previously posed question as to whether the active-majority or the active-single system is better, one can see in Figure 2 that active-majority is, during the learning process, more accurate than active-single, with convergence to a common value occurring only towards the end of the learning process. This difference is a consideration in situations where one has a limited number of training examples available. Figure 2 indicates that active-majority would be the better system in this situation, since its accuracy is, on the average, always greater than that of the other systems.

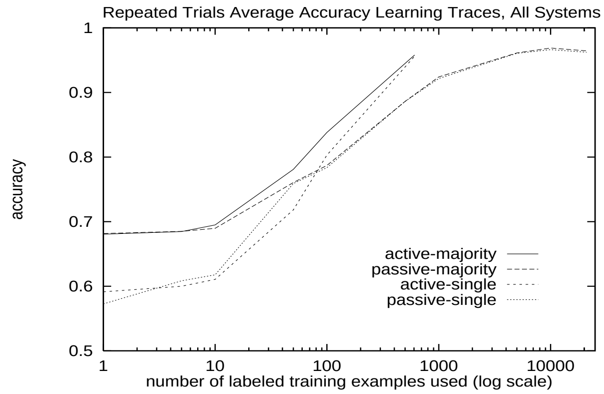

Figure 2: Average Accuracy by System

Finally, it is appropriate to examine the final accuracies in detail, since one of our goals is to develop systems that reduce the number of training examples used without unduly reducing predictive accuracy. The above arguments as to which system is better in various situations would certainly need to be modified if the systems' final accuracies were quite different. We can see in Table 1 that the differences in accuracy are relatively small. That is, all 4 of the systems gave similar levels of accuracy.

system	*mean*	*std dev*
active-majority	0.9581	0.0202
passive-majority	0.9645	0.0320
active-single	0.9562	0.0207
passive-single	0.9622	0.0331

Table 1: Final Accuracies

Anova indicates that the system used does *not* have a significant effect on final accuracy.

5. Conclusions

Of the systems tested, active learning with committees (the active-majority system) is the best approach when one has a limited supply of labeled examples. This approach achieves accuracies that are the same as those obtained by the other systems, but uses only 2.9% as many training examples as the supervised learners. It also requires less execution time than a committee of supervised learners that uses majority rule for prediction. Because it has the best average accuracy as learning progresses, the active-majority system is also the best one for applications in which learning is halted (and prediction commences) after a certain period of elapsed time, such as when interactive processing is occurring with a human being.

If labeled examples are cheap, then the passive-single system is the best approach, as it gives the smallest elapsed time on the average – it uses 29 - 69% of the time required by the other systems tested. This is because the passive-single system does not spend any time performing computations needed to decide whether or not to accept each label – it accepts all labels. Neither does it spend time updating several learners, nor does it spend time having several individual committee members make predictions in order to determine the prediction of the committee as a whole – this committee has only one member.

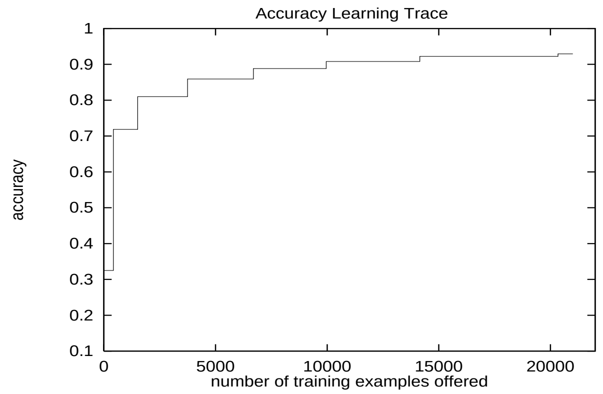

Figure 3: Accuracy versus Examples Offered (for one trial)

We present here a brief intuitive argument as to why the active-majority method works so well. Please see [Freund92] and [Freund95] for a more detailed discussion of this aspect of active learning, as it relates to QBC. Figure 3 shows accuracy as a function of the number of training examples *offered* to the learner (versus used by the learner), with accuracy calculated at each hundredth example used (to smooth out the effects of noise and to make the steps more visible). Initially, assuming that the hypotheses in the committee are sufficiently diverse, two randomly chosen hypotheses disagree on an example with a significantly high probability. Hence labels are requested for a significant fraction (about ½) of the examples. As the learning progresses, each hypothesis approaches the optimal target hypothesis, and hence the diversity between the different hypotheses decreases. As a result, the informativeness of an example as measured by the probability of disagreement between two randomly chosen hypotheses decreases, and the distance between two successive label requests increases. This effect is demonstrated by the horizontal portions of the steps in Figure 3 becoming longer as learning occurs.

Figure 2 suggests that the best accuracy we can hope for, given this data and the type of learner we are using, is about 96 - 97%. This suggestion is based on the observation that the 2 supervised learners seem to reach a plateau which thousands of additional training examples do not alter greatly. This conclusion is consistent with the fact that the data is noisy, and also the data is probably not, in general, linearly separable.

6. Future Work

We would of course like to make the methods even more accurate. One possibility is to increase the number of hypotheses that give different predictive behaviors. This seems to be heading in the direction of full QBC, but one has to find a way to handle noise and the fact that (since we are using Winnow) the data is not linearly separable.

One would like active learning methods that were able to, at least to some degree, operate in a batch mode. By this, we mean that the learner would tell the teacher (human) that it needs the following *n* examples labeled, rather than asking that examples be labeled one at a time.

We would also like to adapt these methods to information retrieval. The task in information retrieval is to respond to queries with documents that satisfy the query. However, it has been found that users are often much better at deciding whether or not a particular document is of interest than they are at expressing that interest in a query language. User input in the form of *relevance feedback* significantly increases retrieval effectiveness [Croft95]. We can think of relevance feedback as allowing the system to learn the user's intentions by asking for the labels for selected examples, and use the active learning with committees paradigm to decide which examples to present to the user.

7. Acknowledgements

This research was partially supported by the National Science Foundation under grant number IRI-9520243. The availability of the Reuters-22173 corpus [Reuters] and of the |STAT Data Manipulation and Analysis Programs [Perlman] has greatly assisted in our research.

8. References

[Angluin88] Dana Angluin, Queries and Concept Learning, *Machine Learning* 2:319-342, 1988

[Apte94] Chidanand Apté, Fred Damerau, Sholom M. Weiss, Automated Learning of Decision Rules for Text Categorization, *ACM TOIS* 12(2):233-251, July 1994

[Board87] Raymond A. Board, Leonard Pitt, Semi-Supervised Learning, Department of Computer Science, University of Illinois at Urbana-Champaign, Report No. UIUCDCS-R-87-1372, September 1987

[Breiman96] Leo Breiman, Bagging Predictors, *Machine Learning* 24(2):123-140, August 1996

[Cohn94] David Cohn, Les Atlas, Richard Ladner, Improving Generalization with Active Learning, *Machine Learning* 15(2):201-221, May 1994

[Croft95] W. Bruce Croft, Effective Text Retrieval Based on Combining Evidence from the Corpus and Users, *IEEE Expert* 10(6):59-63, December 1995

[Dagan95] Ido Dagan, Sean P. Engelson, Committee-Based Sampling for Training Probabilistic Classifiers, in *Proceedings: ICML95*, 1995, p. 150-157

[Freund92] Yoav Freund, H. Sebastian Seung, Eli Shamir, Naftali Tishby, Information, Prediction, and Query by Committee, NIPS92, p. 483-490

[Freund95] Yoav Freund, H. Sebastian Seung, Eli Shamir, Naftali Tishby, Selective Sampling Using the Query by Committee Algorithm, July 1995, to appear in *Machine Learning*

[Hayes90] Phillip J. Hayes, Peggy M. Andersen, Irene B. Nirenburg, Linda M. Schmandt, TCS: A Shell for Content-Based Text Categorization, in *Proceedings of the 6th IEEE CAIA*, 1990, IEEE, p. 320-326

[Lewis91] David D. Lewis, Representation and Learning in Information Retrieval, Ph.D. Thesis, University of Massachusetts at Amherst, COINS Technical Report 91-93, December 1991

[Lewis94] David D. Lewis, William A. Gale, A Sequential Algorithm for Training Text Classifiers, in *Proceedings: SIGIR'94*, p. 3-12

[Littlestone88] Nick Littlestone, Learning Quickly When Irrelevant Attributes Abound: A New Linear-Threshold Algorithm, *Machine Learning* 2(4):285-318, 1988

[Littlestone89] Nicholas Littlestone, Mistake Bounds and Logarithmic Linear-Threshold Learning Algorithms, University of California at Santa Cruz, UCSC-CRL-89-11, March 1989

[Littlestone91] Nick Littlestone, Redundant Noisy Attributes, Attribute Errors, and Linear-Threshold Learning Using Winnow, COLT'91, p. 147-156

[Perlman] Gary Perlman, |STAT version 5.4, software and documentation, available from: `ftp:/archive.-cis.ohio-state.edu/pub/stat/`

[Reuters] Reuters-22173 corpus, a collection of 22,173 indexed documents appearing on the Reuters newswire in 1987; Reuters Ltd, Carnegie Group, David Lewis, Information Retrieval Laboratory at the University of Massachusetts; available via ftp from: `ciir-ftp.-cs.umass.edu:/pub/reuters1/corpus.tar.Z`

[Roth96] Dan Roth, Applying Winnow to Context-Sensitive Spelling Correction, ICML96, p. 182-190

[Seung92] H. S. Seung, M. Opper, H. Sompolinsky, Query by Committee, COLT92, p. 287-294