Kernel Query By Committee (KQBC)

Ran Gilad-Bachrach
ranb@cs.huji.ac.il

Amir Navot
anavot@cs.huji.ac.il

Naftali Tishby
tishby@cs.huji.ac.il

School of Computer Science and Engineering and
Interdisciplinary Center for Neural Computation
The Hebrew University, Jerusalem, Israel

Abstract

The *Query By Committee* (QBC) algorithm is the only algorithm in the *active learning* framework that has a full theoretical justification. It was proved that it can reduce exponentially the number of labels needed for learning. Unfortunately, a naive implementation of this algorithm is impossible due to impractical time complexity. In this paper we make another step toward a practical implementation. The running time of our novel method does not depend on the input dimension but only on the number of labels obtained already. Moreover, it can be expressed in term of inner products only, and therefore we get a *kernel* version of the QBC algorithm.

1 Introduction

Active Learning models [4] allow the student some control over the learning process. The student has the ability to ask questions and thus to direct the teacher to guide him in places where he needs her assistance the most. This is in contrary to *Passive Learning* models, such as *PAC* [18] or *Online Learning* [14], where the student just observes the examples chosen by the teacher. The exact questions the student is allowed to ask change between different models. This option to ask question may reduce dramatically the total amount of guidance needed in order to gain a given performance level. This is mostly desirable as in most learning tasks the teacher guidance is expensive.

One common active learning model allows the student to use *Membership Queries (MQ) [18],* i.e. to present the teacher with instances and ask for correct labels. The MQ is a powerful oracle. Problems such as learning constant depth circuits [13] or learning finite automatons [11] are learnable with MQ but can not be learned in passive models. Yet this model has major problem: the student tend to present questions that have no correct answer (see [2]). However, we are interested in another model, the *Filtering* model, that doesn't suffer from this flaw.

In the *Filtering* model the teacher present the questions and the student decides to which of them he want the teacher to give the correct answer. More formally, consider instances

drawn at random from some underlying distribution. Each random instance is presented to the student, if he thinks that knowing the correct label will be very helpful for the learning process, he can query for it. The motivation behind the filtering model is that in many cases random instances are easy to obtain while the labels are "hard to get" or "expensive". Consider for example document classification task, in this case instances can be collected automatically from the World-Wide-Web. However labeling these instances can be labor intensive task since someone has to read these documents. In this case labeling 1000 instances, i.e. documents, becomes almost intractable task.

Probably the most noticeable algorithm in this field is the *Query By Committee* (QBC) algorithm [15]. When presented with an instance, the student decide weather to query for its label or not according to a vote he holds between hypotheses which were correct so far. In [7, 16] this algorithm was analyzed. It has been shown that the number of queries for labels that the algorithm will make is $O((d/g)\log(1/\epsilon))$ where ϵ is the required accuracy, d is the VC-dimension and g is some constant depending on the concept class and the underlying distribution. Note that in passive learning the size of the sample needed for learning is $O(d/\epsilon)$ hence in terms of the accuracy ϵ there is an exponential reduction in the number of labels needed. This is very promising, but there is a problem: a naive implementation of QBC has unreasonable time complexity. The main obstacle in implementing QBC is the part in which the algorithm has to hold a vote between hypotheses that were correct so far, i.e. hypotheses in the *version space*. Due to this difficulty, the QBC was used only as an inspiration for real-world tasks.

Practitioner have used active learning for various applications: text categorization [12], part of speech tagging [5], learning structure in Bayesian networks [17], speech recognition [9] and many more. In all of the reported experiments the results were in favor of active versus passive learning. However, all the algorithms tested lack theoretical guaranty.

As far as we know, [1] were the first to present algorithm which is theoretically sound and somewhat practical (i.e. polynomial) at the same time. It has been shown that in the case of learning linear separators, the problem of holding the vote between consistent hypotheses for the QBC algorithm can be reduced to the problem of sampling from convex bodies or computing the volume of such bodies. However, the algorithm assumes that the hypotheses are presented explicitly in the feature space and the complexity depends on the dimension of this space. Therefore, using it for problems given in high dimension is impossible and usage of kernel functions [3] is impossible as well.

In this paper we extend the results in [1] and show that it is possible to implement the QBC algorithm for learning linear separators with complexity which depends only on the number of queries made. Since the goal of the algorithm is to reduce the number of queries made, we expect this number to be small. Using this new method it is possible to express the algorithm in terms of inner products only and the complexity is independent of the input dimension, thus it can be applied with kernels[1]. The main tool in our new technique is a projection of the version space on the instances we have seen so far and the instance we would like to label. We show that sampling from the projected body preserve the information we are interested at.

2 The Query By Committee Algorithm and Linear Separation

The query by committee (QBC) algorithm was presented by Sueng et al. [15] and analyzed in [7, 16]. The algorithm assumes the existence of some underlying probability measure over the hypotheses class. At each stage, the algorithm holds the *version-space*: the set

[1] Although the per sample complexity of the algorithm does not depend on the input dimension, due to the results of [7] we expect the number of queries needed in order to learn to grow as the dimension grows.

of hypotheses which were correct so far. Upon receiving a new instance the algorithm has to decide whether to query for its label are not. This is done by randomly selecting hypotheses from the version-space and checking the prediction they make for the label of the new instance. The algorithm is presented as algorithm 1.

Algorithm 1 Query By Committee [15]

The algorithm receives a required accuracy ϵ and a required confidence level $1 - \delta$ and iterates over the following procedure:

1. Receive an unlabeled instance x.
2. Randomly select two hypotheses h_1 and h_2 from the version space, use these hypotheses to obtain two predictions for the label of x.
3. **If** the two predictions disagree **then** query the teacher for the correct label of x.
4. **If** no query for a label was made for the last t_k consecutive instances
   then randomly select an hypothesis from the version space and return it as an approximation to the target concept.
   else return to the beginning of the loop (step 1).

where k is the number of queries made so far and $t_k = \frac{2\pi^2(k+1)^2}{3\epsilon\delta} \ln \frac{2\pi^2(k+1)^2}{3\delta}$.

Freund et al. [7] defined the term *expected information gain* and were able to prove that hypotheses class which have *lower bound on the expected information gain:* $g > 0$, can benefit from using the QBC algorithm: they will need only

$$O\left(\frac{d}{g}\log\frac{1}{\epsilon}\right)$$

labels in order to achieve an accuracy ϵ (where d is the VC dimension).

The main class for which [7] prove that there exist a lower bound on the expected information gain is the class of *linear separators* endowed with the uniform distribution. The class of linear separators is very powerful if kernels are allowed. However a major building block is missing in the QBC algorithm: a method of randomly selecting two hypotheses from the version space (step 3 in algorithm 1) this is especially difficult when kernels are in use.

Several authors tried to address the problem of sampling from the version-space for the case of linear separators. [8] presented a method of Gibbs sampling using random walks. Although the technique presented can tolerate noise and works with kernels, the authors do not provide any guaranty for the correctness of their algorithm and its complexity. In [1] the problem of sampling from the version space was converted to the problem of sampling from convex bodies, and algorithm for solving the later problem was used[2].

We now turn to present the new result of this paper.

3 Kernel QBC - A New Method for Sampling the Version-Space

As discussed before, a good method for sampling the version-space is the key point in turning the QBC algorithm into a practicale solution for learning problems. Therefore we now focus on the problem of sampling the version-space when the concept class is linear separators through the origin in $\mathbb{R}^d$ endowed with the uniform distribution.

Our main result is the following theorem:

[2]The problem of sampling convex bodies or computing their volume is an NP-hard problem [6], however both problems can be approximated to any finite precision.

Theorem 1 *There exist an algorithm which receives a set of labeled instances* $\{(x_i, y_i)\}_1^k$*, an accuracy parameter* $\epsilon > 0$ *and an instance* x *and returns an hypothesis* h *such that*

$$|Pr_h [h \cdot x > 0] - Pr_{w \in V} [w \cdot x > 0]| < \epsilon \tag{1}$$

where V *is the version space:*

$$V = \left\{ w \in \mathbb{R}^d \ \textit{s.t.} \ \ \|w\| = 1 \ \textit{and} \ \forall 1 \leq i \leq k \quad y_i w \cdot x_i > 0 \right\} \tag{2}$$

The running time of this algorithm is $\left(\frac{k}{\epsilon}\right)^{O(1)}$ *and the algorithm uses the instances only through inner products.*

In [1] the same problem was addressed. However the solution provided in [1] was polynomial in the input dimension (i.e. d in our notation) and could not work with kernels. They used the same idea of converting the problem to sampling from convex bodies and used known algorithm for solving the later problem. As in our case this algorithm can not sample from the uniform distribution but only to approximate it in the sense of (1), but as discussed in [1] this does not pose a major problem.

Theorem 1 proves that QBC can be implemented efficiently even when kernels are used. In order to see that recall that QBC samples two hypotheses h_1 and h_2 from the version-space V. QBC uses h_1 and h_2 in order to get predictions for the label of x, hence we can change the method of sampling h_1 and h_2 as long as we sample the predictions correctly.

Each instance splits the version-space into two segments. One segment holds the hypotheses which label the new instance $+1$ and the other segment holds the hypotheses which label it -1. Since we are only interested in the predicted labels, it suffices to sample each segment with the right probability, while the distribution inside each segment is insignificant. The key idea is a low dimensional projection of the version-space on the sub-space spanned by $x, x_1 \ldots, x_k$. The dimension of the projection depends only on the number of labeled instances we have seen so far, while the original dimension of the input is insignificant. An illustration of this process is given in figure figure 1.

We now turn to prove our main theorem.

We begin with establishing a notation for the discussion:

- Let $\{(x_i, y_i)\}_{i=1}^k$ be the labeled instances we already saw.
- Let V be the current version space as defined in (2).
- Let x_0 be a new instance for which we would like to decide weather to query for its label or not. Therefore we would like to sample the labels different hypotheses in the version space assign to x_0. Note that for ease of notation we denote by x_0 the new instance.
- Let $S = \text{span}\{x_0, x_1, \ldots, x_k\}$.
- For any $s \in S$ such that $\|s\| = 1$ we denote by $T(s)$ the set

  $$T(s) = \left\{ w \in V \ : \ \exists \lambda > 0, t \in S^\top \ s.t. \ w = \lambda s + t \right\}$$

  this is the set of all completions of s in V.

Some properties of $T(s)$ are given in the following lemma.

Lemma 1 *The following properties of* $T(s)$ *hold:*

1. *If* $w \in T(s)$ *then* $\textit{sign}(w \cdot x_0) = \textit{sign}(s \cdot x_0)$.

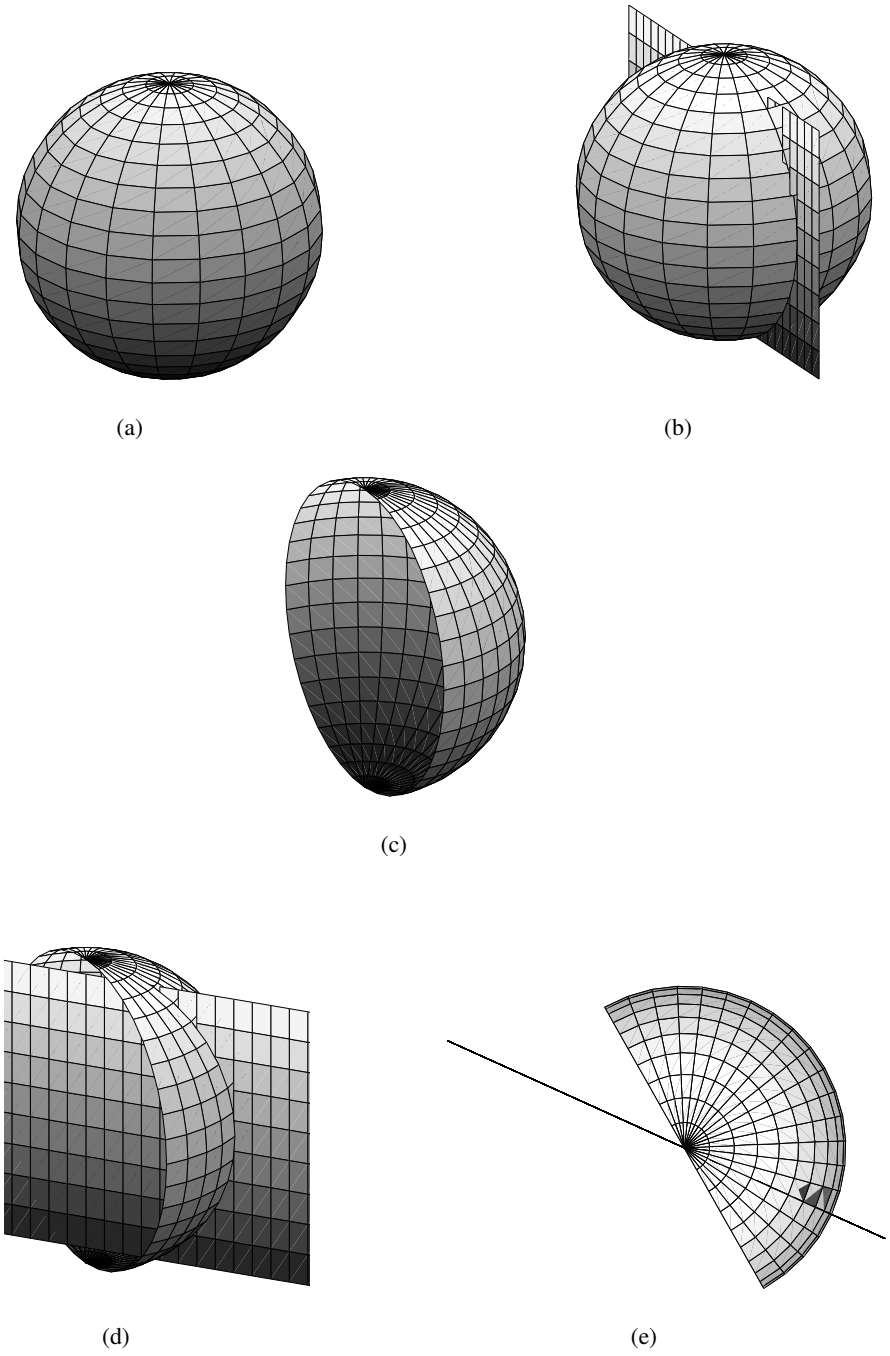

Figure 1: An illustration of the proof theorem 1 for $\mathbb{R}^3$: The hypotheses class is the 2-dimensional sphere (1(a)). The perpendicular plane to a given instance x_1 splits the sphere into two halves according to the label each hypothesis assigns to x_1 (1(b)). Once the true label of x_1 is revealed only one of the sphere halves remains as the version-space (1(c)). Given a new instance x, the plane perpendicular to it splits the version space, as before, into two segments, according to the labels assigned to x (1(d)). In order to decide whether to query for the label of x we need to estimate the relative proportions between the two segments. The projection of the version space on the plane spanned by x and x_1 preserves this proportion (1(e)). Therefore the estimation can be done in low dimension.

2. *Let $s \in S$ be such that $\|s\| = 1$ and let $w = \lambda s + t$ such that $\|w\| = 1$, $\lambda > 0$ and $t \in S^{\top}$ then $w \in T(s)$ iff $s \in V$.*

3. *$T(s) \neq \phi$ iff $s \in V \cap S$.*

4. *For s_1, s_2 if $T(s_1) \cap T(s_2) \neq \phi$ then $s_1 = s_2$.*

5. *If s_1 and s_2 are such that $T(s_1), T(s_2) \neq \phi$ then there exists a rotation Z of $\mathbb{R}^d$ such that $T(s_1) = ZT(s_2)$.*

Proof: The proof is straightforward:

1) Let $w \in T(s)$ then $w = \lambda s + t$ for $\lambda > 0$ and $t \in S^{\top}$. Since $x_0 \in S$ we have that $x_0 \perp t$ and thus $w \cdot x_0 = \lambda s \cdot x_0$ and since $\lambda > 0$ we have that $\text{sign}(w \cdot x_0) = \text{sign}(s \cdot x_0)$.

2) Let w and s be as defined in the lemma. Assume that $w \in T(s)$ then $w \in V$ and thus for $i = 1, \ldots, k$ we have that $y_i w \cdot x_i > 0$. Since $x_i \in S$, $\lambda > 0$ and $t \perp x_i$we have that $y_i s \cdot x_i > 0$ and thus $s \in V$. On the other hand, if $s \in V$ then $y_i s \cdot x_i > 0$ and using the same argument we have that $y_i w \cdot x_i > 0$ and therefore $w \in T(s)$.

3) This property follows immediately from property 2 by choosing $w = s$.

4) Let $w \in T(s_1) \cap T(s_2)$. Therefore $w = \lambda_1 s_1 + t_1 = \lambda_2 s_2 + t_2$ where $t_1, t_2 \in S^{\top}$ therefore $\lambda_1 s_1 - \lambda_2 s_2 = t_2 - t_1$ and thus $\lambda_1 s_1 - \lambda_2 s_2 \in S \cap S^{\top}$ and so $\lambda_1 s_1 - \lambda_2 s_2 = 0$ and thus $\lambda_1 s_1 = \lambda_2 s_2$. Since $\lambda_1, \lambda_2 > 0$ and $\|s_1\| = \|s_2\| = 1$ we conclude that $s_1 = s_2$.

5) Let s_1, s_2 be such that $T(s_1)$ and $T(s_2)$ are non empty. Since $\|s_1\| = \|s_2\| = 1$ there exists a rotation Z of S such that $Zs_2 = s_1$. Z can be extended to operate on $\mathbb{R}^d$ by defining $Z(s + t) = Z(s) + t$ for $s \in S$ and $t \in S^{\top}$. Let $w \in T(s_2)$ then $w = \lambda s_2 + t$ for $\lambda > 0$ and $t \in S^{\top}$. Therefore $Z(w) = \lambda s_1 + t$. ¿From property 3 we have that $s_1 \in T(s_1)$, using property 2 we have that $Z(w) \in T(s_1)$ and therefore $ZT(s_2) \subseteq T(s_1)$. Since Z is invertible and due to symmetry we have that $Z^{-1}T(s_1) \subseteq T(s_2)$ and by applying Z to both sides we have $T(s_1) \subseteq ZT(s_2)$ which complete the proof.

□

The simple lemma just presented is the key to our new algorithm, instead of sampling w from V we will sample s from $V \cap S$. First we will show that this will yield the right probabilities and later we will show how sampling from $V \cap S$ can be done.

Lemma 2 *Let V be the current version space, and S be the sub space spanned by the labeled instances $x_0, x_1, \ldots, x_k$. Then*

$$Pr_{w \in V}\left[w \cdot x_0 \geq 0\right] = Pr_{s \in S \cap V}\left[s \cdot x_0 \geq 0\right] \tag{3}$$

Proof: Lemma 1 shows that $T(s)$ breaks V into equivalence classes. Let P be the projection such that $w \in V$ is projected to s such that $w \in T(s)$. Since $T(s)$ is non-empty iff $s \in V \cap S$ then P induces a distribution on $V \cap S$, we denote by D this distribution.

Using properly 1 in lemma 1 we have that every $w \in T(s)$ assigns the same label to x_0 and thus

$$\Pr_{w \in V}\left[w \cdot x_0 \geq 0\right] = \Pr_{s \sim D}\left[s \cdot x_0 \geq 0\right]$$

next we would like to prove that D is the uniform distribution over $S \cap V$. Recall that the uniform distribution is invariant to rotations hence the density of $T(s)$ in V is the same for every $s \in V \cap S$. The density of $T(s)$ in V is by definition the density of s in $V \cap S$ according to the distribution D, hence D is the uniform distribution.

□

Using the lemmas just proved we can prove out main theorem

Proof (theorem 1): Lemma 2 shows that for our purpose it is enough to sample uniformly from $S \cap V$. Recall from the definitions that $S = \text{span}\{x_0, x_1, \ldots, x_k\}$ thus S is a t dimensional space where $t \leq k+1$.

Since for $s \in S \cap V$ and $\lambda > 0$ we have that s and λs assign the same label to x_0 we can "give depth" to $S \cap V$ by defining the convex body

$$C = \{\lambda s \ : \ 0 < \lambda \leq 1 \text{ and } s \in S \cap V\}$$

and sample uniformly from C.

Let $v_1, \ldots, v_t$ form an orthogonal basis for S. We can rewrite C in this basis

$$C = \left\{ \sum_{j=1}^{t} \alpha_j v_j \ : \ \forall 1 \leq i \leq k \quad y_i \left(\sum_{j=1}^{t} \alpha_j v_j \right) \cdot x_i > 0 \text{ and } \left\| \sum_{j=1}^{t} \alpha_j v_j \right\| \leq 1 \right\}$$

The body C is a convex body. Since $v_1, \ldots, v_t$ form an orthogonal basis we have that $\left\| \sum_{j=1}^{t} \alpha_j v_j \right\| = \sqrt{\sum_{j=1}^{t} \alpha_j^2}$. We can also write $v_j = \sum_{r=0}^{k} c_r^j x_r$ and thus have a definition of C which uses only inner products of the different x_i's. Finally we can sample the α_j's as presented in algorithm 2, since the v_j's are orthonormal. This sampling can be done in time $O\left(\frac{k}{\epsilon}\right)^{O(1)}$ (see [10]) such that (1) will hold. □

Algorithm 2 Sampling the label

Given a set of labeled instances $\{(x_i, y_i)\}_{i=1}^{k}$ and a new instance x_0:

1. Find $v_1, \ldots, v_t$ which forms an orthonormal basis to $S = \text{span}\{x_0, \ldots, x_k\}$.
2. Calculate c_r^j for $r \in [0, k]$ and $j \in [1, t]$ such that $v_j = \sum_r c_r^j x_i$
3. Use an algorithm for sampling from convex bodies to get $\alpha_1, \ldots, \alpha_t$ from the body

   $$K = \left\{ \alpha_1, \ldots, \alpha_t \ : \ \forall i = 1, \ldots, k \quad y_i \sum_{r=0}^{k} \left(\sum_{j=1}^{t} \alpha_j c_r^j \right) x_r \cdot x_i > 0 \text{ and } \sum_{r=1}^{t} \alpha_r^2 \leq 1 \right\}$$

   (Note that $\sum_j \alpha_j v_j \cdot x_i = \sum_{r=0}^{k} \left(\sum_{j=1}^{t} \alpha_j c_r^j \right) x_r \cdot x_i$)
4. return sign $\left(\sum_{r=0}^{k} \left(\sum_{j=1}^{t} \alpha_j c_r^j \right) x_r \cdot x_0 \right)$

4 Summery of Result and Further Study

In this paper we presented a novel technique for implementing the QBC algorithm for learning linear separators. The new algorithm can be expressed in terms of inner products only and its complexity depends only on the number of queries made, thus it is applicable with kernels as well. The key idea in our new technique is to project the version-space on the lower dimension space spanned by the previous labeled instances and the new given instance. Altogether we made a big step toward an applicable implementation of QBC.

The main thing which is holding us from going further and experiment with this new algorithm is that the "state-of-the-art" algorithms for sampling from convex bodies have complexity of $O^*(n^3)$ (where n is the dimension of the body to be sampled and O^* means that

we neglect log factors) for each sample made and $O^*(n^5)$ for preprocessing. In the terms of active learning this means that every time we make a query for a true label we have to suffer $O^*(k^5)$ operations (preprocessing) where k is the number of true labels seen so far. After that, each time we are presented with an instance we use $O^*(k^3)$ operations. This complexity prevents it from being useful at the moment, because we cannot use more than about 15 queries, which is too few for any real word problem. However, since the area of sampling from convex bodies is very active, we expect the efficiency of the solutions to increase in the next few years. Moreover, if computers will be faster enough to allow about 100 queries in the future, the algorithm will be useful already.

References

[1] R. Bachrach, S. Fine, and E. Shamir. Query by committee, linear separation and random walks. *TCS*, 284(1), 2002.

[2] E. B. Baum and K. Lang. Query learning can work poorly when human oracle is used. In *International Joint Conference in Neural Netwroks*, 1992.

[3] C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 1998.

[4] D. Cohn, L. Atlas, and R. Ladner. Improving generalization with active learning. *Machine Learning*, 15(2):201–221, 1994.

[5] I. Dagan and S. Engelson. Committee-based sampling for training probabilistic classifiers. *Proceedings of the 12th International Conference on Machine Learning*, 1995.

[6] G. Elekes. A geometric inequality and the complexity of computing volume. *Discrete and Computational Geometry*, 1, 1986.

[7] Y. Freund, H. Seung, E. Shamir, and N. Tishby. Selective sampling using the query by committee algorithm. *Macine Learning*, 28:133–168, 1997.

[8] T. Graepel and R. Hebrich. The kernel gibbs sampler. In *NIPS*, 2001.

[9] D. Hakkani-Tur, G. Riccardi, and A. Gorin. Active learning for automatic speech recognition. In *ICASSP*, 2002.

[10] R. Kannan, L. Lovasz, and M. Simonovits. Random walks and an $o^*(5)$ volume algorithm for convex bodies. *Random Structures and Algorithms*, 11:1–50, 1997.

[11] M. Kearns and U. Vazirani. *An Introduction To Computational Learning Theory*. The MIT Press, 1994.

[12] Ray Liere and Prasad Tadepalli. Active learning with committees for text categorization. In *AAAI-97,*, 1997.

[13] N. Linial, Y. Mansour, and N. Nissan. Constant-depth circuits, fourier transform and learnability. *Jour. Assoc. Comput. Mach.*, 40:607–620, 1993.

[14] N. Littlestone. *Mistake Bounds and Logarithmic Linear-threshold Learning Algorithms*. PhD thesis, University of California Santa Cruz, 1989.

[15] H. S. Seung, M. Opper, and H. Sompolinsky. Query by committee. *Proc. of the Fith Workshop on Computational Learning Theory*, pages 287–294, 1992.

[16] P. Sollich and D. Saad. Learning from queries for maximum information gain in imperfectly learnable problems. *Advances in Neural Information Systems*, 7:287–294, 1995.

[17] S. Tong and D. Koller. Active learning for structure in bayesian networks. In *International Joint Conference on Artificial Intelligence*, 2001.

[18] L. G. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.