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 among the few algorithms in the *active learning* framework that has some theoretical justification. Freund et al[7] proved that QBC can reduce the number of labels needed for learning exponentially, if the version space can be randomly sampled. Unfortunately, a naive implementation of this algorithm is generally impossible due to impractical time complexity. In this paper we make another step toward a practical implementation of QBC, by combining it with Kernel methods. The running time of our method does not depend on the input dimension but only on the number of obtained labels. Moreover, the algorithm requires only inner products of the labeled data points, yielding a general *kernel* version of the QBC algorithm.

1 Introduction

Active Supervised Learning models [4] allow the student some control over the learning process. The student has the ability to make queries and direct the "teacher" to the input domains for which more assistance is needed. This is in contrast to the more common *Passive Learning* theoretical models, such as *PAC* [18] or *Online Learning* [14], where the student obtains labeled examples chosen by the teacher. The precise nature of the student's queries vary between different models, but the query option can dramatically reduce the total amount of supervision needed to guarantee a given performance level. This is desirable especially for the common cases for which labeled data, that requires teacher's assistance, 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 query its correct label. The MQ is known to be a powerful oracle. Problems such as learning constant depth circuits [13] and learning finite automata [11] are efficiently learnable with MQ, but not in passive models. Yet MQ has a major drawback: the student tends to present questions that have no correct or clear label (see [2]). For this reason we are interested in another mechanism - data *Filtering* - that doesn't suffer from this flaw.

In the *Filtering* model the teacher presents the data (questions) but the student decides for which data point to query for the correct label. More specifically, consider instances drawn at random from some underlying distribution over the instance space. Each random instance is presented to the student who quiries for the label only if he estimate it will be helpful for the learning process. The motivation behind this filtering model is that often random instances are easy to obtain while their labels are "hard to get" or "expensive". Consider, for example, a document classification task. In this case documents can be collected automatically from the World-Wide-Web, but labeling these documents can be labor intensive. In this case manual labeling of thousands of documents becomes almost impractical.

One of the most interesting algorithms in this filtering domain is the *Query By Committee* (QBC) algorithm [15]. When presented with an instance, the student decides weather to query for its label according to a "vote" he holds among a "committee" of randomly selected hypotheses from the version space. In [7, 16] this algorithm was analyzed. It has been shown that the number of label queries required by the algorithm is $O((d/g)\log(1/\epsilon))$, where ϵ is the required accuracy (generalization error), d is the VC-dimension of the concept class, and g depends on the geometry of the class as well as on the underlying distribution. Notice that in passive learning the sample size needed for ϵ-generalization is $O(d/\epsilon)$, hence in terms of ϵ there is an exponential saving in the number of needed labels. This seems very promising but there is a serious caveat: a naive implementation of QBC requires unreasonable time complexity. The main obstacle in implementing QBC is in selecting the committee of random hypotheses that were correct so far, i.e. hypotheses in the *version space*. This difficulty, which has to do with uniform sampling of version-spaces in high dimensions, is the main reason why QBC is not used for real-world applications.

Practitioner have used active learning for various applications, such as text categorization [12], part of speech tagging [5], structure learning in Bayesian networks [17], and speech recognition [9]. In all of the reported experiments the results favor active versus passive learning. However, all those applications lack any theoretical guarantee.

To the best of our knowledge [1] were the first to present a QBC algorithm which is both theoretically sound and has polynomial time complexity. It has been shown there that for learning linear separators ("preceptrons"), selecting the committee and voting in QBC can be reduced to the problem of uniform sampling from convex bodies in high dimensions, or equivalently - estimating their volume. The algorithm assumes, however, that the hypotheses are explicitly presented in the feature space and that the time complexity critically depends on the dimension of the version space. Thus, applying it to problems in high dimensions turns impossible and so is the usage of kernel functions (see [3]).

In this paper we extend the results in [1] and show that it is in fact possible to implement QBC for learning linear separators with time complexity that depends *only* on the number of queries and not on the class properties. Since the goal of the algorithm is to reduce the number of queries made, which can not be too large in practice anyway, we expect this number to be small and we thus obtain a significant theoretical improvement. Moreover, using our new method it is possible to express the algorithm in terms of only *inner products* of data points. Hence the complexity is independent of the input dimension and it can be implemented with kernels[1]. The main technical component in our new algorithm is a projection of the version space on the *labeled* instances seen so far and on the new instance we would like to label. We show that sampling from this projected space preserves the information-gain, which we need to maintain.

[1]Notice that while the per-sample time-complexity does not depend on the input dimension, as shown by [7], we do expect the number of queries to grow with the dimension.

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 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 required accuracy ϵ and confidence $1 - \delta$ 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 $t_k = \frac{2\pi^2(k+1)^2}{3\epsilon\delta} \ln \frac{2\pi^2(k+1)^2}{3\delta}$.

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 is 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.

In the case of the linear separators the version space takes the form:

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

where $x_1, \ldots, x_k$ are the instances for which a query for a label was made and the labels $y_1, \ldots, y_k \in \{\pm 1\}$ where obtained.

Several authors tried to address the problem of sampling from the version-space (2) in 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 don't provide any guaranty for the correctness of their algorithm and it's complexity. In [1] the problem of sampling from the version space was converted to the problem of sampling from convex bodies, and used algorithm for solving the later problem [2].

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

[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.

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

Assume that the hypotheses class is the class of linear separators through the origin in $\mathbb{R}^d$ and there is a uniform prior over this class. After the student have seen a set of labeled instances $\{(x_i, y_i)\}_{i=1}^k$ the current version space can be described as

$$V = \left\{ w \in \mathbb{R}^d \ : \ \|w\| = 1 \text{ and } \forall i \ y_i w \cdot x_i > 0 \right\}$$

Since the prior was uniform, the posterior after observing $\{(x_i, y_i)\}_{i=1}^k$ is uniform over V.

According to the QBC algorithm we should sample two hypotheses h_1, h_2 from V and compare the labels assigned to a new instance x_0. We do something slightly different. Assume that y is a random variable which is distributed as the posterior of the label of x_0, i.e.

$$\begin{aligned} \Pr_y [y = 1] &= \Pr_{w \in V} [w \cdot x_0 > 0] \\ \Pr_y [y = -1] &= \Pr_{w \in V} [w \cdot x_0 < 0] \end{aligned}$$

We can sample twice the random variable y and query for the label of x_0 only if the two samples of y disagree.

We now demonstrate how the task of sampling the label can be done. We start with a simple example.

3.1 A Proof by Grape-Fruit

The hypotheses class of linear separators is a ball. Each labeled instance induces a cut in this ball. Assume that the hypothesis class is a grape-fruit. Furthermore assume that the current version space consists of two adjacent slices of this grape-fruit. One of these slices is the set of hypotheses in the version space which label a new instance x_0 with the label $+1$ while the other slice is the set of hypotheses in the version space which label x_0 with the label -1.

The main observation we would like to make is as follows: the relative ratio of the volumes of the two slices equals exactly to the relative area of the slices if we make a cut through the grape-fruit as demonstrated in figure 1.

Note that although we are interested in the 3-dimensional volume, it suffices in the case of the grape-fruit to look at a 2-dimensional area. We are now about to turn to the general case. In this case we will look at d-dimensional separators (where d is considered to be very large) and a $k+1$ dimensional cut.

3.2 A Rigorous Proof of the "Grape-Fruit" 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 i.e.

$$V = \left\{ w \in \mathbb{R}^d \ : \ \|w\| = 1 \text{ and } \forall i \ y_i w \cdot x_i > 0 \right\}$$

- Let x_0 be a new instance for which we would like to decide weather to query for it's label or not. Therefore we would like to sample the labels different hypotheses in the version space assign to x_0.
- Let $S = \text{span}\{x_0, x_1, \ldots, x_k\}$.

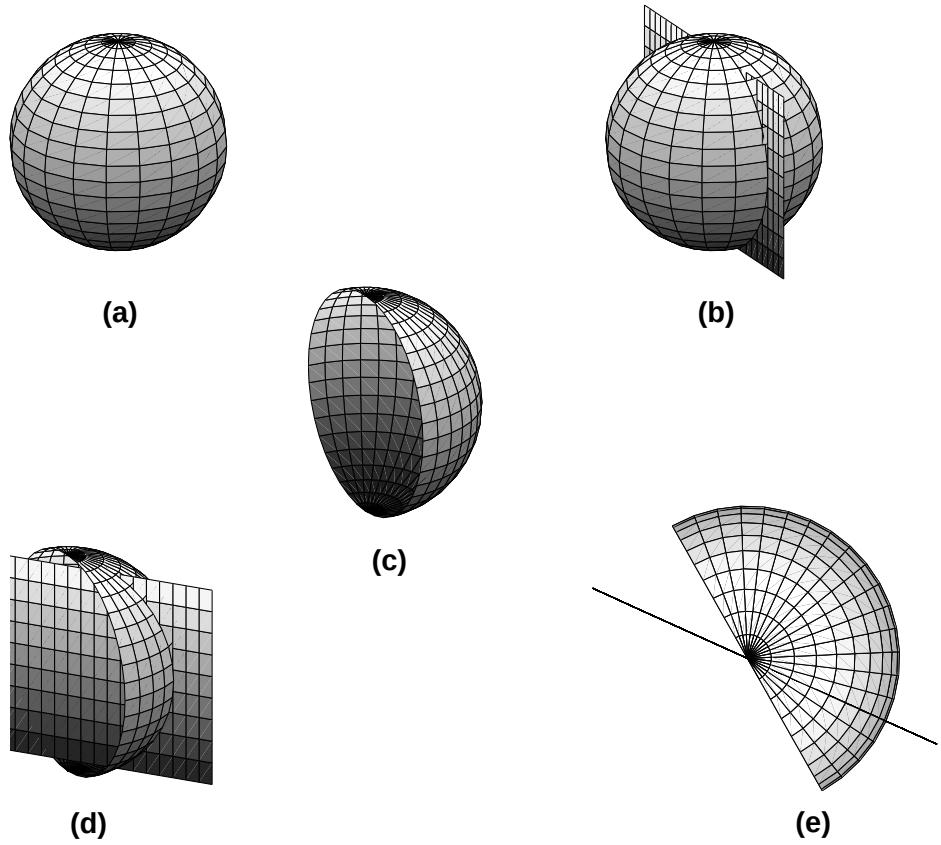

Figure 1: An illustration of the proof theorem **??** for $\mathbb{R}^3$: The hypotheses class is the 2-dimensional sphere (a). The perpendicular plane to a given instance x_1 splits the sphere into two halves according to the label each hypothesis assigns to x_1 (b). Once the true label of x_1 is revealed only one of the sphere halves remains as the version-space (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 (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 (e). Therefore the estimation can be done in low dimension.

- 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.

We now turn to prove several properties of $T(s)$

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

1. *If $w \in T(s)$ then sign$(w \cdot x_0)$ = sign$(s \cdot x_0)$.*
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 sign$(w \cdot x_0)$ = 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 $Z s_2 = s_1$. 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 completes 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*

$$Pr_{w \sim U(V)}\left[w \cdot x_0 \geq 0\right] = Pr_{s \sim U(S \cap V)}\left[s \cdot x_0 \geq 0\right] \quad (1)$$

where $U(\cdot)$ is the uniform distribution.

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

$$\Pr_{w \sim U(V)}[w \cdot x_0 \geq 0] = \Pr_{s \sim D}[s \cdot x_0 \geq 0]$$

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.

□

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 $t \leq k+1$. Also recall that V, the version space, is defined as $V = \{w : \forall i \in [1, \ldots, k]\ y_i w \cdot x_i > 0 \text{ and } \|w\| = 1\}$.

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 define 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 redefine K as follows

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

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 $v_j = \sum_{i=0}^{k} c_i^j x_i$ and thus have a definition of C which uses only inner products of the different x_i's. The exact solution is presented in algorithm 2.

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_i^j for $i \in [0, k]$ and $j \in [1, t]$ such that $v_j = \sum_i c_i^j x_i$
3. Use an algorithm for sampling from convex bodies [10] to get $\alpha_1, \ldots, \alpha_t$ from the body

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

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

---

4 Summery and Further Study

In this paper we present a novel technique for implementing the QBC algorithm for learning linear separators. This technique provides a more realistic, yet rigrouse, implementation of the QBC algorithm. The time-complexity of our algorithm depends only on the number of queries made and not on the input dimension or the VC-dim of the class. Furthemore, our technique requres only inner products of the labeled data points - thus can be implemented with kernels as well.

The main point holding us from practical implementations is the current "state-of-the-art" in efficient sampling from convex bodies. Best known convex sampling algorithms [10] have computational complexity of $O^*(n^3)$, where n is the input dimension and O^* indicate neglected log factors, for each sample and $O^*(n^5)$ for preprocessing. In terms of active learning using KQBC this means that for every label-query requires $O^*(k^5)$ preprocessing operations, where k is the number of labels obtained so far. Each time we are presented with a new labeled instance require another $O^*(k^3)$ operations. This computational complexity is still too high for most applications at this point. There is hope, however, since sampling from convex bodies is a very active research area and expect efficiency of the algorithms to improve in the coming years.

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 classifi ers. *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.