# Query by Committee, Linear Separation and Random Walks

Ran Bachrach, Shai Fine, and Eli Shamir*

Institute of Computer Science
The Hebrew University, Jerusalem, Israel
{ranb,fshai,shamir}@cs.huji.ac.il

**Abstract.** Recent works have shown the advantage of using *Active Learning* methods, such as the *Query by Committee* (QBC) algorithm, to various learning problems. This class of Algorithms requires an oracle with the ability to randomly select a consistent hypothesis according to some predefined distribution. When trying to implement such an oracle, for the linear separators family of hypotheses, various problems should be solved. The major problem is time-complexity, where the straight-forward *Monte Carlo* method takes exponential time.
In this paper we address some of those problems and show how to convert them to the problems of sampling from convex bodies or approximating the volume of such bodies. We show that recent algorithms for approximating the volume of convex bodies and approximately uniformly sampling from convex bodies using random walks, can be used to solve this problem, and yield an efficient implementation for the QBC algorithm. This solution suggests a connection between random walks and certain properties known in machine learning such as $\epsilon$-net and support vector machines. Working out this connection is left for future work.

* Partially supported by project I403-001.06/95 of the German-Israeli Foundation for Scientific Research [GIF].

# 1 introduction

In the *Active Learning* paradigm [3] the learner is given access to a stream of unlabeled samples, drawn at random from a fixed and unknown distribution and for every sample the learner decides whether to query the teacher for the label. Complexity in this context is measured by the number of requests directed to the teacher along the learning process. The reasoning comes from many real life problems where the teacher's activity is an expensive resource. For example, if one would like to design a program that classifies articles into two categories ("interesting" and "non-interesting") then the program may automatically scan as many articles as possible (e.g. through the Internet). However, articles which the program needs the teacher's comment (tag) - the teacher must actually read, and that is a costly task. The *Query By Committee* (QBC) algorithm [11] is an Active Learning algorithm acting in the *Bayesian* model of concept learning [8] i.e. it assumes that the concept to be learned is chosen according to some fixed distribution known to the learning algorithm. The algorithm uses three oracles: The *Sample* oracle returns a random sample $x$, the *Label* oracle returns the label(tag) for a sample, and the *Gibbs* oracle returns a random hypothesis from the version space. The algorithm gets two parameters - accuracy ($\alpha$) and reliability ($\beta$) - and works as follows:

1. Call *Sample* to get a random sample $x$.
2. Call *Gibbs* twice to obtain two hypotheses and generate two predictions for the label of $x$.
3. **If** the predictions are not equal
   **Then** call *Label* to get the correct label for $x$.
4. **If** *Label* was not used for the last $t_k$[1] consecutive samples, where $k$ is the current number of labeled samples,
   **Then** call *Gibbs* once and output this last hypothesis
   **Else** return to the beginning of the loop (step 1).

A natural mean for tracing the progress of the learning process is the rate at which the size of the version space decreases. We adopt the notion of information gain as the measure of choice for the analysis of the learning process:

**Definition 1 (Haussler et. al. [8]).** *Let $\mathcal{V}_x = \{h \in \mathcal{V} | h(x) = c(x)\}$ be the version space after sample $x$ had been labeled*

- *The instantaneous information gain is $\mathcal{I}(x, c(x)) = -\log \Pr_{h\in\mathcal{V}}[h \in \mathcal{V}_x]$*
- *The expected information gain of a sample $x$ to is*

$$\begin{aligned}\mathcal{G}(x|\mathcal{V}) &= \mathcal{H}(\Pr_{h\in\mathcal{V}}[h(x) = 1]) \\ &= \Pr_{h\in\mathcal{V}}[h(x) = 1] \cdot \mathcal{I}(x, 1) + \Pr_{h\in\mathcal{V}}[h(x) = -1] \cdot \mathcal{I}(x, -1)\end{aligned} \tag{1}$$

*where $H$ is the binary entropy, i.e. $\mathcal{H}(p) = -p \log p - (1-p)\log(1-p)$.*

[1] $t_k = \frac{2\pi^2(k+1)^2}{3\alpha\beta} \ln \frac{2\pi^2(k+1)^2}{3\beta}$ is a correction of the expression given at ([6]).

**Theorem 1 (Freund et. al. [6]).** *If a concept class $C$ has VC-dimension $0 < d < \infty$ and the expected information gain from the queries to* Label *Oracle made by QBC are uniformly lower bounded by $g > 0$ bits, then the following holds with probability larger than $1 - \beta$ over the choice of the target concept, the sequence of samples, and the choices made by QBC:*

1. *The number of calls to* Sample *is $m_0 = (\frac{d}{\alpha\beta g})^{O(1)}$.*
2. *The number of calls to* Label *is smaller then*[2] $k_0 = \frac{10(d+1)}{g} \ln \frac{4m_0}{\beta}$.
3. *The algorithm guarantees that*

$$\Pr_{c,h,QBC}[\Pr_x[h(x) \neq c(x)] \geq \alpha] \leq \beta \tag{2}$$

The main theme governing the proof of this theorem is the capability to bound the number of queries made by QBC in terms of $g$, the lower bound for the expected information gain: If the algorithm asks to tag all $m$ samples then $\Pr\left[\sum_i^m \mathcal{G}(x_i|\mathcal{V}) \geq (d+1)(\log \frac{\alpha m}{d})\right] < \frac{d}{em}$, meaning the accumulated information gain grows logarithmically with $m$. Obviously, when filtering out samples the accumulated information gain cannot be larger. On the other hand, $kg$ is a lower bound on the accumulated expected information gain from $k$ tagged samples. These two observations suggest that $kg \leq (d+1)(\log \frac{\alpha m}{d})$, which results in a bound on $k$ and implies that the gap between consecutive queried samples is expected to grow until the stop-condition will be satisfied. Theorem 1 can be augmented to handle general class of filtering algorithms: Let $L$ be an algorithm that filters out samples based on an internal probability assignment and previous query results. Using a stop-condition identical to the one used by the QBC algorithm and following the basic steps at the proof of theorem 1, one may conclude similar bounds on the number of calls $L$ makes to *Sample* and *Label* oracles.

By stating a lower bound on the expected information gain, Freund et. al. were able to identify several classes of concepts as learnable by the QBC algorithm. Among them are classes of perceptrons (linear separators) defined by a vector $w$ such that for any sample $x$:

$$c_w(x) = \begin{cases} +1, \text{if } x \cdot w > 0 \\ -1, \text{otherwise} \end{cases} \tag{3}$$

with the restriction that the version space distribution is known to the learner and both sample space and version space distributions are almost uniform. A question which was left open is how to efficiently implement the *Gibbs* oracle and thus reduce QBC to a standard learning model (using only *Sample* and *Label* oracles). It turns out that this question falls naturally into a class of approximating problems which got much attention in recent years: How to get an efficient approximation of volume or random sampling of rather complex defined and dynamically changing spaces. Moreover, unraveling the meaning of random walks

[2] $k_0 = O(\log \frac{1}{\alpha\beta})$ results in an exponential gap between the number of queries made to *Label*, comparing to "regular" algorithms.

employed by these approximate counting methods seems to have interesting implications in learning theory. Let us focus on the problem of randomly selecting hypotheses from the version space and limit our discussion to classes of linear separators: There are several known algorithms for finding a linear separator (e.g. the perceptron algorithm), but none of them suffice since we need to *randomly* select a separator in the version space. A possible straightforward solution is the use of the *Monte Carlo* mechanism: Assuming (as later we do) that the linear separators are uniformly distributed, we randomly select a point in the unit sphere[3], identifying this point as a linear separator, and check whether it is in the version space. If not, proceed with the sampling until a consistent separator will be selected. This process yields several problems, the most important of which is efficiency. Recall that the QBC algorithm assumes a lower bound $g > 0$ for the expected information gain from queried samples. Let $p \leq 1/2$ be such that $\mathcal{H}(p) = g$. Having $k$ tagged samples, the probability to select a consistent separator is smaller then $(1-p)^k$. This implies that the expected number of iterations the Monte Carlo algorithm makes until it finds a desired separator is greater then $(1-p)^{-k}$. If the total number of samples the algorithm uses is $m$, and $k$ is the number of tagged samples, then the computational complexity is at least $\Omega(m(1-p)^{-k})$. Plugging in the expected value for $k$ in the QBC algorithm, i.e. $\frac{10(d+1)}{g} \ln \frac{4m}{\beta}$, the Monte Carlo implementation results in a computational complexity exponential in $g$, $d$ (the VC-dimension, i.e. $n+1$ in our case) and also a depends on $m^2$. Furthermore, if the version space decreases faster then it's expected value, then finding a consistent linear separator will take even longer time. The algorithms we suggest in this paper work in time polynomial in $n, g$ and depend on $mk^{O(1)}$, i.e., they are exponentially better in terms of the VC-dimension and $g$ and also have better polynomial factors in terms of $m$. We also avoid the problem of rapid decrease in the size of the version space by employing a detecting condition.

### 1.1 Mathematical Notation

The sample space $\Omega$, is assumed to be a subset of $\Re^n$ and therefore a sample is a vector in $\Re^n$. A linear separator is a tuple $\{v, \text{offset}\}$ where $v$ is a vector in $\Re^n$ and offset $\in \Re$. To simplify notation we shall assume that each sample $x$ is taken from $\Re^{n+1}$ forcing $x_1 = 1$. Hence a linear separator is just a vector in $\Re^{n+1}$.The concept to be learned is assumed to be chosen according to a fixed distribution $\mathcal{D}$ over $\Re^{n+1}$. We denote by $x^i$ a queried sample and $t^i$ the corresponding tag, where $t^i \in \{-1, 1\}$. The version space $\mathcal{V}$ is defined to be $\mathcal{V} = \{v | \forall i (\langle x^i, v\rangle \cdot t^i) > 0\}$. Let $Y^i = t^i \cdot x^i$. Then a vector $v$ is a linear separator if $\forall i \ \langle Y^i, v\rangle > 0$. Using matrix notation we may further simplify notation by setting $Y^i$ to be the $i$'th row of matrix $A$ and writing $\mathcal{V} = \{v | Av > 0\}$.

[3] pick $n$ normally distributed variables $\zeta_1 \ldots \zeta_n$ and normalize them by the square root of the sum of their squares

### 1.2 Preliminary Observations

Upon receiving a new sample $x$, the algorithm needs to decide whether to query for a tag. The probability for labeling $x$ with $+1$ is: $P^+ = Pr_{\mathcal{D}}[v \in V^+]$ where $\mathcal{D}$ is the distribution induced on the version space $\mathcal{V}$ and $V^+ = \{v|Av > 0, \langle x, v\rangle \geq 0\}$. Similarly we define $V^-$ and $P^-$ which correspond to labeling $x$ with $-1$. The QBC algorithm decides to query for a tag only when the two hypotheses disagree on $x$'s label and this happens with probability $2P^+P^-$. Thus $P^+$ and $P^-$ are all we need in order to substitute *Gibbs* oracle and make this decision. Normalizing $||v|| = 1$, the version space of linear separators becomes subset of $n$ dimensional sphere $S^n$. Under the uniform distribution on $S^n$, the value of $P^+$ (and $P^-$) can be obtained by calculating $n+1$ dimensional volume: $P^+ = \frac{\text{Vol}(V^+)}{\text{Vol}(\mathcal{V})}$. Now $\mathcal{V}$, $V^+$ and $V^-$ are convex *simplexes* [4] in the $n+1$ dimensional unit ball. Having $P^+$ and $P^-$, we can substitute the *Gibbs* oracle: Given a set of tagged samples and a new sample $x$, query the label of $x$ with probability $2P^+P^-$.

### 1.3 Few Results about Convex Bodies

In order to simulate the *Gibbs* oracle we seek efficient methods for calculating the volume of a convex body and uniformly sampling from it. Very similar questions relating convex bodies have been addressed by Dyer et. al. [4], Lovasz and Simonovits [9] and others.

**Theorem 2 (The Sampling Algorithm [9]).** *Let $K$ be a convex body such that $K$ contains at least $2/3$ of the volume of the unit ball ($B$) and at least $2/3$ of the volume of $K$ is contained in a ball of radius $m$ such that $1 \leq m \leq n^{3/2}$. For arbitrary $\epsilon > 0$ there exist a sampling algorithm that uses $O(n^4 m^2 \log^2(1/\epsilon)(n \log n + \log(1/\epsilon))$ operations on numbers of size $O(\log n)$ bits and returns a vector $v$ such that for every Lebesgue measurable set $L$ in $K$: $|\Pr(v \in L) - \frac{Vol(L)}{Vol(K)}| < \epsilon$*

The algorithm uses random walks over the convex body $K$ as its method of traverse and it reaches every point of $K$ with almost equal probability. To complete this result, Grotchel et. al. [7] used the ellipsoid method to find an affine transformation $T_a$ such that given a convex body $K$, $B \subseteq T_a(K) \subseteq n^{3/2}B$ . Assume that the original body $K$ is bounded in a ball of radius $R$ and contains a ball of radius $r$ then the algorithm finds $T_a$ in $O(n^4(|\log r| + |\log R|))$ operations on numbers of size $O(n^2(|\log r| + |\log R|))$ bits. Before we proceed, let us elaborate on the meaning of these results for our needs: When applying the sampling algorithm to the convex body $\mathcal{V}$, we will get $v \in \mathcal{V}$. Since $V^+$ and $V^-$ are both simplexes, then they are Lebesgue measurable, hence $|\Pr(v \in V^+) - P^+| < \epsilon$. Note that we are only interested in the proportions of $V^+$ and $V^-$ and the use of the affine transformation $T_a$ preserve these proportions. The sampling algorithm

[4] the term *simplex* is used to describe a conical convex set of the type $K = \{v \in \Re^n | Av > 0, ||v|| \leq 1\}$. Note that it is a nonstandrd use of this term.

enables Lovasz and Simonovits to come out with an algorithm for approximating the volume of a convex body:

**Theorem 3 (Volume Approximation algorithm [9]).** *Let $K$ be a convex body*[5] *in $\Re^n$. There exists a volume approximating algorithm such that upon receiving error parameters $\epsilon, \delta \in (0,1)$ and numbers $R$ and $r$ such that $K$ contains a ball of radius $r$ and is contained in a ball of radius $R$ centered at the origin, the algorithm outputs a number $\zeta$ such that with probability at least $1-\delta$*

$$(1-\epsilon)\,Vol(K) \leq \zeta \leq (1+\epsilon)\,Vol(K) \tag{4}$$

*The algorithm works in time polynomial in $|\log R| + |\log r|$,$n$,$1/\epsilon$ and $\log(1/\delta)$.*

For our purposes, this algorithm can estimate the expected information gained from the next sample. It also approximates the value of $P^+$ (and $P^-$) and thus we may simulate the *Gibbs* oracle by choosing to query for a tag with probability $2P^+(1-P^+)$. Both the sampling algorithm and the volume approximation algorithm require the values of $R$ and $r$. Since in our case all convex bodies are contained in the unit ball $B$, then fixing $R = 1$ will suffice and we are left with the problem of finding $r$. However, it will suffice to find $r$ such that $\frac{r^*}{4} \leq r \leq r^*$ where $r^*$ is the maximal radius of a ball contained in $K$. Moreover, we will have to show that $r$ is not too small. The main part of this proof is to show that if the volume of $V^+$ is not too small, then $r$ is not too small (lemma 1). Since we learn by reducing the volume of the version space, this lemma states that the radius decreases in a rate proportional to the learning rate.

## 2 Modified Query By Committee Algorithms

In this section we present two variants of the QBC algorithm: the first uses approximation of the volume of convex bodies, while the second uses the technique for sampling from convex bodies. Both algorithms are efficient and maintain the exponential gap between labeled and unlabeled samples. QBC" is especially interesting from computational complexity perspective, while the mechanism in the basis of QBC' enables the approximation of the maximal a-posteriori (MAP) hypothesis in $Poly(\log m)$ time as well as direct access to the expected information gain from a query.

### 2.1 Using Volume Approximation in the QBC Algorithm (QBC')

Every Sample $x$ induces a partition of the version space $\mathcal{V}$ into two subsets $V^+$ and $V^-$. Since $\mathcal{V} = V^+ \cup V^-$ and they are disjoint, then $\mathrm{Vol}(\mathcal{V}) = \mathrm{Vol}(V^+) + \mathrm{Vol}(V^-)$ and $P^+ = \frac{\mathrm{Vol}(V^+)}{\mathrm{Vol}(V^+)+\mathrm{Vol}(V^-)}$. Hence, approximating these volumes results in approximation of $P^+$ that we can use instead of the original value. In order to use the volume approximation algorithm as a procedure, we need to

[5] $K$ is assumed to be given using a separation oracle.

bound the volumes of $V^+$ and $V^-$, i.e. find balls of radii $r^+$ and $r^-$ contained in $(V^+)$ and $(V^-)$ respectively. If both volumes are not too small then the corresponding radii are big enough and may be calculated efficiently using convex programming. If one of the volumes is too small then we are guaranteed that the other one is not too small since $\mathrm{Vol}(\mathcal{V})$ is not too small (lemma 7). It turns out that if one of the radii is very small then assuming that the corresponding part is empty (i.e. the complementary part is the full version space) is enough for simulating the *Gibbs* oracle (lemma 3). The QBC' algorithm follows:

Given $\alpha, \beta > 0$ and let $k$ be the current number of labeled samples, define $t_k = \frac{1024\pi^4(k+1)^4 \log \frac{8\pi^2(k+1)^2}{3\beta}}{9\alpha^2\beta^2}$ , $\delta_k = \frac{3\beta}{4\pi^2(k+1)^2}$ , $\epsilon_k = \frac{\alpha\delta_k}{48}$ , $\mu_k = \frac{2\epsilon_k}{n}$

1. Call *Sample* to get a random sample $x$.
2. Use convex programming to simultaneously calculate the values of $r^+$ and $r^-$, the radii of balls contained in $V^+$ and $V^-$.
3. **If** $min(r^+, r^-) < \mu_k(max(r^+, r^-))^n$
   **Then** assume that the corresponding body is empty and **goto** 6.
4. Call the volume approximation algorithm with $r^+$ (and $r^-$) to get $\zeta^+$ (and $\zeta^-$) such that
$$(1 - \epsilon_k)\mathrm{Vol}(V^+) \leq \zeta^+ \leq (1 + \epsilon_k)\mathrm{Vol}(V^+)$$
   with probability greater then $1 - \delta_k$
5. Let $\widehat{P^+} = \frac{\zeta^+}{\zeta^+ + \zeta^-}$, with probability $2\widehat{P^+}(1 - \widehat{P^+})$ call *Label* to get the correct label of $x$.
6. **If** *Label* was not used for the last $t_k$ consecutive samples,
   **Then** call the sampling algorithm with $\epsilon = \frac{\beta}{2}$ to give an hypothesis and stop. **Else** return to the beginning of the loop (step 1).

With probability greater then $1 - \beta$ the time complexity of QBC' is $m$ times $Poly(n, k, \frac{1}{\alpha}, \frac{1}{\beta})$ (for each iteration). The number of samples that the algorithm uses is polynomial in the number of samples that the QBC algorithm uses. Furthermore, the exponential gap between the number of samples and number of labeled samples still holds.

### 2.2 Using Sampling in the QBC algorithm (QBC")

Another variant of QBC is the QBC" algorithm which simulate the *Gibbs* oracle by sampling, almost uniformly, two hypothesis from the version space:

Given $\alpha, \beta > 0$ and let $k$ be the current number of labeled samples, define $\beta_k = \frac{3\beta}{\pi^2(k+1)^2}$ , $t_k = \frac{8\log 4/\beta_k}{\alpha\beta_k}$ , $\epsilon_k = \frac{\alpha\beta_k}{32}$ , $\mu_k = \frac{2}{n}\min(\epsilon_k, 1 - \epsilon_k)$

1. Call *Sample* to get a random sample $x$.
2. Call the sampling algorithm with $\epsilon_k$ and $r$ to get two hypotheses from the version space.

3. **If** the two hypotheses disagree on the label of $x$
   **Then** use convex programming to simultaneously calculate the values of $r^+$ and $r^-$, the radii of the balls contained in $V^+$ and $V^-$.
4. **If** $min(r^+, r^-) \geq \mu_k(max(r^+, r^-))^n$
   **Then** call *Label* to get the correct label and choose $r^+$ (or $r^-$), the radius of a ball contained in the new version space, to be used in step 2.
5. **If** *Label* was not used for the last $t_k$ consecutive samples,
   **Then** call the sampling algorithm with $\epsilon = \frac{\beta}{2}$ to give an hypothesis and stop.
   **Else** return to the beginning of the loop (step 1).

QBC" is very similar to QBC' but with one major difference: calculating new radius is conducted almost only when the version space changes and this happens only after querying for a tag. Hence each iteration takes $O(n^7 \log^2(1/\epsilon)(n \log n + \log(1/\epsilon))$ operations and an extra $Poly(n, 1/\epsilon, 1/\delta, k)$ is needed when a tag is asked. Due to the exponential gap between the number of samples and number of tagged samples, this algorithm may be attractive for practical use.

## 3 Deriving the Main Results

We start the analysis by presenting a useful lemma which bounds the radii of a ball contained in a simplex as a function of its volume. The algorithms for estimating the volume of a convex body, or sampling from it, work in time polynomial in $\log r$ ($R = 1$ in our case). The following lemma, gives a lower bound on the size of $r$, hence will be useful for analyzing the time-complexity of our algorithms.

**Lemma 1.** *Let $K$ be a convex body contained in the $n$-dimensional unit ball $B$ (assume $n > 1$). Let $v = Vol(K)$ then there is a ball of radius $r$ contained in $K$ such that*

$$r \geq \frac{1}{n} \frac{Vol(K)}{Vol(B)} = \frac{v\Gamma(\frac{n+2}{2})}{n\pi^{n/2}} \tag{5}$$

*Proof.* We shall assume that $K$ is given by a finite set of linear inequalities with the constraint that all the points in $K$ are within the unit ball. Let $r^*$ be the supremum of the radii of balls contained in $K$, and let $r \geq r^*$. We denote by $\partial K$ the boundary of $K$. Then $\partial K$ has a derivative at all points apart from a set of zero measure.

We construct a set $S$ by taking a segment of length $r$ for each point $y$ of $\partial K$ such that $\partial K$ has a derivative at $y$. We take the segment to be in direction which is orthogonal to the derivative at $y$ and pointing towards the inner part of $K$ (see figure 1).

The assumption that $r \geq r^*$ implies $K \subset S$ up to a set of zero measure. To show this we look at the following: Let $x \in K$, then there is $\hat{r}$ which is the maximal radius of a ball contained in $K$ centered at $x$. If we look at the ball of radius $\hat{r}$ than it intersects $\partial K$ at least at one point. At this point, denote $y$,

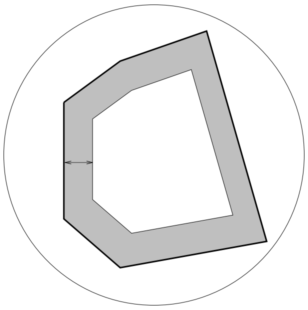

**Fig. 1.** An example of the process of generating the set $S$ (the gray area) by taking orthogonal segments to $\partial K$ (the bold line). In this case, the length $r$ (the arrow) is too small.

the derivative of $\partial K$ is the derivative of the boundary of the ball (the derivative exists). Hence the orthogonal segment to $y$ of length $\hat{r}$ reaches $x$. Since $\hat{r} \leq r^* \leq r$ then the orthogonal segment of length $r$ from $y$ reaches $x$. The only points that might not be in $S$ are the points in $\partial K$ where there is no derivative, but this is a set of measure zero. Therefore

$$\mathrm{Vol}(S) \geq \mathrm{Vol}(K) = v \tag{6}$$

But, by the definition of $S$ we can see that

$$\mathrm{Vol}(S) \leq |\int_{\partial K} r| = r\mathrm{Vol}(\partial K) \tag{7}$$

Note that $\mathrm{Vol}(\partial K)$ is $n-1$ dimensional volume. Hence, we conclude that $v \leq r\mathrm{Vol}(\partial K)$ or $r \geq \frac{v}{\mathrm{Vol}(\partial K)}$. This is true for every $r \geq r^*$ therefore

$$r^* \geq \frac{v}{\mathrm{Vol}(\partial K)} \tag{8}$$

It remains to bound the size of $\mathrm{Vol}(\partial K)$. In [1] appendix C we show that the convex body with maximal surface is the unit ball itself (this result can be obtained from the isoperimetric inequalities as well). Since the volume of $n$-dimensional ball is $\frac{r^n \pi^{n/2}}{\Gamma(\frac{n+2}{2})}$ the $n-1$ dimension volume of its surface is $\frac{nr^{n-1}\pi^{n/2}}{\Gamma(\frac{n+2}{2})}$. Substituting $\mathrm{Vol}(\partial K)$ with the volume of the surface of the unit ball in equation (8) we conclude that $r^* \geq \frac{v\Gamma(\frac{n+2}{2})}{n\pi^{n/2}}$. □

*Remark 1.* Perhaps the lemma is known, but we could not find a reference. Eggleston [5] (also quoted at M. Berger's book *Géométrie*, 1990) gives a rather difficult proof of the lower bound $r \geq \frac{\min \mathrm{width}(K)}{2\sqrt{n}}$ for $n$ odd (for even $n$ the bound is somewhat modified). The two lower bounds seem unrelated. Moreover, $\mathrm{width}(K)$ seems hard to approximate by sampling.

Convex programming provides the tools for efficiently estimating the radius of the ball contained in a simplex as shown in the statement of the next lemma (proved in App. A):

**Lemma 2.** *Let $K$ be a simplex contained in the unit ball defined by $k$ inequalities. Let $r^*$ be the maximal radius of a ball contained in $K$. Using the ellipsoid algorithm, it is possible to calculate a value $r$, such that $r^* \geq r \geq r^*/4$, in time which is Poly$(n, |\log r^*|, k)$.*

We are now ready to present the main theorem for each variant of the QBC algorithm. However, since the difference in the analysis of the two algorithms provides no further insight we chose to describe here only the proof of QBC" which seems more adequate for practical use. The interested reader may find a detailed proofs for both algorithms in the comprehensive version [1].

**Theorem 4.** *Let $\alpha$,$\beta > 0$, let $n > 1$ and let $c$ be a linear separator chosen uniformly from the space of linear separators. Then algorithm QBC" (as described in section 2.2) returns an hypothesis $h$ such that*

$$Pr_{c,h}[Pr_x[c(x) \neq h(x)] > \alpha] < 1 - \beta \tag{9}$$

*using $m_0 = (n, 1/\alpha, 1/\beta)^{O(1)}$ samples and $k_0 = O(n \log \frac{m_0}{\beta})$ labeled samples. With probability greater then $1 - \beta$, the time complexity is $m$ (the number of iterations) times Poly$(n, k, \frac{1}{\alpha}, \frac{1}{\beta})$ (the complexity of each iteration).*

*Proof.* Our proof consists of three parts: the correctness of the algorithm, it's sample complexity and computational complexity. We base the proof on three lemmas which are specified at App. A.

In lemma 4 we show that if $q$ is the probability that QBC will ask to tag a sample $x$ and $\hat{q}$ is the probability that QBC" will ask to tag the same sample then $|q - \hat{q}| \leq 4\epsilon_k$. Using this result, we show in lemma 6 that if the algorithm didn't ask to tag more then $t_k$ consecutive samples and $\mathcal{V}$ is the version space then

$$\Pr_{c,h\in\mathcal{V}}[\Pr_x[c(x) \neq h(x)] > \alpha] \leq \beta/2 \tag{10}$$

therefore if we stop at this point and pick an hypothesis approximately uniformly from $\mathcal{V}$ with picking accuracy $\beta/2$, we get an algorithmic error of no more then $\beta$ and this completes the proof of the correctness of the algorithm.

We now turn to discuss the sample complexity of the algorithm. Following Freund et. al. [6] we need to show that at any stage in the learning process there is a uniform lower bound on the expected information gain from the next query. Freund et. al. showed that for the class of linear separators such a lower bound exists, when applying the original QBC algorithm under certain restriction on the distribution of the linear separators and the sample space. In corollary 2 we show that if $g$ is such a lower bound for the original QBC algorithm then there exists a lower bound $\hat{g}$ for the QBC" algorithm such that $\hat{g} \geq g - 4\epsilon_k$. Since $\epsilon_k \leq \alpha\beta$ we get a uniform lower bound $\hat{g}$ such that $\hat{g} \geq g - 4\alpha\beta$, so for $\alpha\beta$ sufficiently small $\hat{g} \geq g/2$. I.e. the expected information QBC" gain from each query it makes, is "almost" the same as the one gained by QBC.

Having this lower-bound on the expected information gain and using the augmented Theorem 1 , we conclude that the number of iterations QBC" will make is bounded by $\mathrm{Poly}(n, \frac{1}{\alpha}, \frac{1}{\beta}, \frac{1}{g})$ while the number of queries it will make is less then $\mathrm{Poly}(n, \log\frac{1}{\alpha}, \log\frac{1}{\beta}, \frac{1}{g})$.

Finally we would like to discuss the computational complexity of the algorithm. There are two main computational tasks in each iteration QBC" makes: First, the algorithm decides whether to query on the given sample. Second, it has to compute the radii of balls contained in $V^+$ and $V^-$.

The first task is done using twice the algorithm which approximate uniform sampling from convex bodies in order to obtain two hypotheses. The complexity of this algorithm is polynomial in $|\log\epsilon_k|, n, |\log r|$ where $\epsilon_k$ is the accuracy we need, $n$ is the dimension and $r$ is a radius of a ball contained in the body. In lemma 7 we bound the size of $r$ by showing that if the algorithm uses $m$ samples then with probability greater then $1-\beta$, in every step $r \geq \frac{\beta\left(\frac{n}{em}\right)^n}{n}$. Since we are interested in $\log r$ this bound suffice.

The other task is to calculate $r^+$ and $r^-$, which is done using convex programming as shown in lemma 2. It will suffice to show that at least one of them is "big enough" since we proceed in looking for the other value for no more then another $\log\mu_k$ iterations (a polynomial value): From lemma 7 we know that $\mathrm{Vol}(\mathcal{V}) \geq \frac{\beta\left(\frac{n}{em}\right)^n \pi^{n/2}}{\Gamma\left(\frac{n+2}{2}\right)}$ at every step. At least one of $V^+$ or $V^-$ has volume of at least $\frac{\mathrm{Vol}(\mathcal{V})}{2}$. Using the lower bound on the volume of $\mathcal{V}$ and lemma 1 we conclude that the maximal radius $r \geq \frac{\beta\left(\frac{n}{em}\right)^n}{2n}$.

We conclude that with probability greater then $1-\beta$ the time-complexity of each iteration the QBC" algorithm is $\mathrm{Poly}(n, k, \log\frac{1}{\beta}, \log\frac{1}{\alpha})$. In the final iteration, the algorithm returns an hypothesis from the version space. Using the sampling algorithm this is done in polynomial time. Combined with the fact that there are at most $m_0$ iterations the statement of the theorem follows. $\square$

## 4 Conclusions and Further Research

In this paper we presented a feasible way to simulate the *Gibbs* oracle picking almost uniformly distributed linear separators and thus reduce *Query by Committee* to a standard learning model. To this purpose, we used convex programming and formed a linkage to a class of approximation methods related to convex body sampling and volume approximation. These methods use random walks over convex bodies, on a grid which depends on the parameters $\epsilon$ and $\delta$, in a similar fashion to PAC algorithms. It seems that such random walks could be described using $\epsilon$-net terminology or alike. We thus suggest that this connection have further implication in Learning Theory.

Freund et. al. [6] assumed the existence of the *Gibbs* oracle and essentially used the information gain only for proving convergence of the QBC algorithm (Theorem 1). The use of the volume approximation technique (i.e. QBC') provides a direct access to the instantaneous information gain. This enable us to suggest another class of algorithms, namely *Information Gain Machines*, which make use of the extra information. Combining our results with the ones presented by Shawe-Taylor and Williamson [12] and McAllester [10] may allow to obtain generalization error estimates for such information gain machines and make use of QBC' ability to produce maximum a-posteriori estimate.

The two algorithms presented have to estimate a radius of a ball contained in a convex body, which in our case is the version space $\mathcal{V}$. Finding the center of a large ball contained in the version space is also an essential task in the theory of support vector machines (SVM) [2]. In this case the radius is the margin of the separator. It is also clear that QBC is a filtering algorithm which seeks samples that are going to be in the support set of the current version space. This similarity implies a connection between the two paradigms and is the subject of our current research.

## 5 Acknowledgment

We would like to thank also R. El-Yaniv, N. Linial and M. Kearns for their good advices.

## References

[1] R. Bachrach, S. Fine, and E. Shamir. Query by committee, linear separation and random walks - full version. 1998. Available at `http://www.cs.huji.ac.il/labs/learning/Papers/MLT_list.html`.

[2] C. Burges. A tutorial on support vector machines for pattern recognition. 1998. Available at `http://svm.research.bell-labs.com/SVMdoc.html`.

[3] D. Cohn, L. Atlas, and R. Ladner. Training connectionist networks with queries and selective sampling. *Advanced in Neural Information Processing Systems 2*, 1990.

[4] M. Dyer, A. Frieze, and R. Kannan. A random polynomial time algorithm for approximating the volume of convex bodies. *Journal of the Association for Computing Machinery*, 38, Number 1:1–17, 1991.

[5] H. G. Eggleston. *Convexity*. Cambridge Univ. Press, 1958.

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

[7] L. Grotchel, L. Lovasz, and A. Schrijver. *Geometric algorithms and Combinatorial Optimization*. Springer-Verlag, Berlin, 1988.

[8] D. Haussler, M. Kearns, and R. E. Schapie. Bounds on the sample complexity of bayesian learning using information theory and the vc dimension. *Machine Learning*, 14:83–113, 1994.

[9] L. Lovasz and M. Simonovits. Random walks in a convex body and an improved volume algorithm. *Random Structures and Algorithms*, 4, Number 4:359–412, 1993.

[10] D. A. McAllester. Some pac-bayesian theorems. *Proc. of the Eleventh Annual Conference on Computational Learning Theory*, pages 230–234, 1998.

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

[12] J. Shawe-Taylor and R. C. Williamson. A pac analysis of a bayesian estimator. *Proc. of the Tenth Annual Conference on Computational Learning Theory.*, 1997.

## A lemmas for QBC"

**Corollary 1.** *Let $K$ be a simplex in the unit ball such that the maximal radius of the ball it contains is $r$. then*

$$\frac{r^n \pi^{n/2}}{\Gamma(\frac{n+2}{2})} \leq \mathit{Vol}(K) \leq \frac{rn\pi^{n/2}}{\Gamma(\frac{n+2}{2})} \tag{11}$$

*Proof.* The lower bound is obtained by taking the volume of the ball with radius $r$ contained in $K$, and the upper bound is a direct consequence of lemma 1. □

*Proof (of lemma 2).* Let $A$ be the matrix representation of the $k$ inequalities defining $K$. Given a point $x \in K$, we would like to find $r^*$, the maximal radius of a ball centered at $x$ and contained in $K$. Any ball contained in $K$ and centered at $x$ is also contained in the unit ball (which is centered at the origin). Hence, its radius, $r$, must satisfy $||x|| + r \leq 1$. The ball with the maximal radius meets the boundary of $K$ and at these points the boundary of $K$ is tangent to that ball. If the boundary is defined by $A_i y = 0$, then the minimal distance between this boundary and the point $x$ is given by $|\langle A_i, x\rangle| = |A_i x|$ (assuming $A_i$ is normalized such that $||A_i||_2 = 1$). Since $x \in K$ then $|A_i x| = A_i x$, which implies that for any ball with a radius $r$ centered at $x$ and contained in $K$, $\forall i\ A_i x \geq r$. Else, the ball meets the spherical boundary thus $||x|| + r \geq 1$. This last discussion suggest that finding $r^*$ may be expressed as an optimization problem

$$r^* = \arg\max_r \{r | \exists x \text{ s.t. } ||x|| + r \leq 1, Ax \geq r\} \tag{12}$$

It is easy to see that this is a *convex programming* problem: Fix $r$ and assume that $x$ does not satisfies one of the conditions which defines the optimization problem. If there exists $i$ such that $\lambda = A_i x < r$ then the hyper-plane defined by $\{y | A_i y = \lambda\}$ is a separating hyper-plane. Otherwise, if $||x|| + r > 1$ then the hyper plane defined by $\{y | \langle y - x, x\rangle = 0\}$ is a separating hyper plane (this is the orthogonal hyper plane to the segment from the origin to $x$).

To conclude we need to show that the *ellipsoid algorithm* can do the job efficiently. First notice that $r^*$ is always bounded by 1. At lemma 7 we were able to show that $r^*$ is not worst then exponentially small. For our purposes, finding an $r$ such that $r \geq r^*/4$ will suffice. Note that if $r \geq r^*/4$ then the volume of a ball with the radius $r$ centered at $x$ and contained in $K$ is at least $(\frac{r^*}{4})^n \mathrm{Vol}(B)$ where $\mathrm{Vol}(B)$ is the volume of the $n$ dimensional unit ball. Hence, it is not worst than exponentially small in $\log(r^*)$ and $n$. Since $r^*$ is not too small, efficiently of the *ellipsoid algorithm* is guaranteed. □

**Lemma 3.** *Let $K_1$, $K_2$ be two convex simplexes and let $r_1$, $r_2$ be the maximal radii of corresponding contained balls. For $i = 1, 2$ let $v_i = \mathit{Vol}(K_i)$. Define $p = \frac{v_1}{v_1+v_2}$, if $r_1 \leq \frac{\mu r_2^n}{n}$ then $p \leq \mu$.*

*Proof.* From corollary 1 it follows that $v_1 \leq \frac{r_1 n \pi^{n/2}}{\Gamma(n+2/2)}$ and $v_2 \geq \frac{r_2^n \pi^{n/2}}{\Gamma(n+2/2)}$. Therefore

$$\frac{v_1}{v_2} \leq \frac{nr_1}{r_2^n} \tag{13}$$

Let $\mu \leq 1$ . Substituting $r_1 \leq \frac{\mu r_2^n}{n}$ we conclude that

$$p = \frac{v_1}{v_1 + v_2} \leq \frac{v_1}{v_2} \leq \mu \tag{14}$$

□

**Lemma 4.** *For any sample $x$, let $q$ be the probability that the QBC algorithm will query for a tag and let $\hat{q}$ be the similar probability for QBC". Then*

$$|q - \hat{q}| \leq 4\epsilon_k \tag{15}$$

*Proof.* We start by analyzing the case that the radii size condition of QBC" is satisfied, i.e. $min(r^+, r^-) \geq \mu_k(max(r^+, r^-))^n$. Let $p = \Pr_{u \in \mathcal{V}}[u(x) = 1]$, then $q = 2p(1-p)$.

The QBC" algorithm samples two hypotheses, $h_1$ and $h_2$, from the version-space $\mathcal{V}$. The sampling algorithm guarantees that

$$|\Pr_{h_1 \in \mathcal{V}}[h_1 \in V^+] - \Pr[V^+]| \leq \epsilon_k \tag{16}$$

and the similarly for $V^-$ and $h_2$. Denote $a = \Pr_{h_1 \in \mathcal{V}}[h_1 \in V^+]$ and $b = \Pr_{h_2 \in \mathcal{V}}[h_2 \in V^-]$. Since $h_1$ and $h_2$ are independent random variables then $\hat{q} = 2ab$. Therefore, in order to bound $|q - \hat{q}|$ we need to maximize $|2p(1-p) - 2ab|$ subject to the restrictions $|p - a| \leq \epsilon_k$ and $|(1-p) - b| \leq \epsilon_k$. It is easy to check that the maximum is achieved when $|a - p| = \epsilon_k$ and $|b - (1-p)| = \epsilon_k$ and therefore

$$|2p(1-p) - 2ab| \leq 2\epsilon_k + 2\epsilon_k^2 \leq 4\epsilon_k \tag{17}$$

We now consider the case where the radii ratio condition fails. Without loss of generality, assume $r^+ < \mu_k(r^-)^n$. From lemma 3 it follows that $p < n\mu_k$ and by the definition of $\mu_k$ we get $p < 2\min(\epsilon_k, 1 - \epsilon_k)$ which means that $q < 4\min(\epsilon_k, 1-\epsilon_k)(1 - 2\min(\epsilon_k, 1-\epsilon_k)) < 4\epsilon_k$. Therefore, by defining $\hat{q} = 0$ we maintain the difference $|q - \hat{q}| \leq 4\epsilon_k$. □

**Lemma 5.** *Let $L$ be any samples filtering algorithm. For any sample $x$, let $q$ be the probability that the QBC algorithm will query for a tag and let $\hat{q}$ be the corresponding probability for $L$ and assume $|q - \hat{q}| \leq \gamma$. Let $g$ be a lower bound on the expected information gain for QBC. Then there exists a lower bound on the expected information gain for $L$, denoted by $\hat{g}$, such that*

$$\hat{g} \geq g - \gamma \tag{18}$$

*Proof.* Let $r(x)$ be the density function over the sample space $X$. Let $p(x)$ be the probability that $x$ is tagged 1, i.e. $p(x) = \Pr_{u \in \mathcal{V}}[u(x) = 1]$, then $q(x) = 2p(x)(1 - p(x))$. Since $g$ is a lower bound on the expected information gain for the QBC algorithm then

$$g \leq \int_x r(x)q(x)\mathcal{H}(p(x))dx \tag{19}$$

since $|q - \hat{q}| \leq \gamma$, the expected information gain for $L$ is bounded by

$$\int_x r(x)\hat{q}(x)\mathcal{H}(p(x))dx \geq \int_x r(x)q(x)\mathcal{H}(p(x))dx - \gamma \geq g - \gamma \tag{20}$$

taking a close enough lower bound $\hat{g}$ for the leftmost term, the statement of the lemma follows. □

**Corollary 2.** *Let $g$ be a lower bound on the expected information gain of the QBC algorithm. From lemmas 4 and 5 it follows that there exists $\hat{g} \geq g - 4\epsilon_k$ such that $\hat{g}$ is a lower bound on the expected information gain of QBC".*

**Lemma 6.** *Assume that after getting $k$ labeled samples, algorithm QBC" does not query for a tag in the next $t_k$ consecutive samples. If $c$ is a concept chosen uniformly from the version space and $h$ is the hypothesis returned by QBC", then*

$$Pr_{c,h}[Pr_x[c(x) \neq h(x)] > \alpha] \leq \beta/2 \tag{21}$$

*Proof.* We define *bad pair* to be a pair of hypotheses from the version space that differ on more then proportion $\alpha$ of the samples. We will want the algorithm to stop only when the queries it made form an $\left(\alpha, \frac{\beta_k}{4}\right)$-net, i.e. if two hypotheses are picked independently from the version space, the probability that they form a bad pair is less then $\frac{\beta_k}{4}$. We will show that if the algorithm did not make a query for $t_k$ consecutive samples, then the probability that the queries sampled do not form an $\left(\alpha - \frac{\beta_k}{4}\right)$-net, is bounded by $\beta_k/4$.

Let $W = \{(h_1, h_2)\,|\Pr_x[h_1(x) \neq h_2(x)] \geq \alpha\}$. If $\Pr[W] \leq \beta_k/4$ then the probability that $(c, h)$ is a bad pair is bounded by $\beta_k/4$ (when picked uniformly).We would like to bound the probability that $\Pr[W] > \beta_k/4$ when QBC" didn't query for a tag for $t_k$ at the last consecutive samples:

If $\Pr[W] > \beta_k/4$, then the probability that the QBC algorithm will query for a tag is greater then $\alpha\beta_k/4$. From lemma 4 we conclude that the probability that the QBC" algorithm will query for a tag is greater then $\alpha\beta_k/4 - 4\epsilon_k$. Plugging in $\epsilon_k = \frac{\alpha\beta_k}{32}$ we conclude that the probability that QBC" will query for a tag is greater then $\alpha\beta_k/8$. Therefore, the probability that it won't query for a tag $t_k$ consecutive samples is bounded by $(1 - \alpha\beta_k/8)^{t_k}$. Using the well known relation $(1 - \varepsilon)^n \leq e^{-n\varepsilon}$ and plugging in $t_k = \frac{8 \log 4/\beta_k}{\alpha\beta_k}$ it follows that

$$(1 - \frac{\alpha\beta_k}{8})^{t_k} \leq e^{-t_k\frac{\alpha\beta_k}{8}} = e^{-\frac{8\alpha\beta_k \log 4/\beta_k}{8\alpha\beta_k}} = e^{\log(\frac{\beta_k}{4})} = \frac{\beta_k}{4} \tag{22}$$

Hence, $\Pr_{x_1, x_2, \ldots}[\Pr[W] > \beta_k/4] \leq \beta_k/4$. Thus if the algorithm stops after $t_k$ consecutive unlabeled samples, the probability of choosing an hypothesis which forms a bad pair with the target concept is lower then $\beta_k/2$, since the probability of $W$ being bigger then $\beta_k/4$ is less then $\beta_k/4$, and if it is smaller than $\beta_k/4$ then the probability for mistake is bounded by $\beta_k/4$. Since $\beta_k = \frac{3\beta}{\pi^2(k+1)^2}$ it follows that $\sum_k \beta_k = \beta/2$ and we get the stated result. □

**Lemma 7.** *Let $a > 0$ and let $m$ be the number of calls to the Sample Oracle that QBC" (or QBC') makes (assume $m \geq n$) then the following holds with probability greater then $1 - 2^{-a}$:*

*Each intermediate version space the algorithm generates has a volume greater than $\frac{2^{-a}\left(\frac{d}{em}\right)^d \pi^{n/2}}{\Gamma\left(\frac{n+2}{2}\right)}$, and there is a ball of radius greater than $\frac{2^{-a}\left(\frac{d}{em}\right)^d}{n}$ contained in it.*

*Proof.* The final version space, the one that is being used when the algorithm stops, is the smallest of all intermediate version spaces. Moreover, if the algorithm was not filtering out samples, but querying labels for all samples, then the final version space would have been smaller. Since we are interested in the worst case, we will assume that the algorithm did not filter out any sample.

Fix $X$ a vector of samples, while $X^m$ is its first $m$ samples. Let $c$ be a concept chosen from $\Omega$. $X^m$ divides $\Omega$, the set of all concepts, to equivalence sets, two concepts are in the same set if they give the same label to all $m$ samples in $X^m$. If $\Omega$ has a VC-dimension $d$, we know from *Sauer's lemma* that the number of equivalence sets is bounded by $\left(\frac{em}{d}\right)^d$. Using the distribution $\Pr_c$ over $\Omega$ it follows that

$$\Pr_c\left[-\log \Pr[C] \geq a + d \log \frac{em}{d}\right] \leq 2^{-a} \tag{23}$$

where $C$ is the equivalence set of $c$. We now turn to discuss the special case of linear separators and the uniform distribution over the unit ball, i.e. $\Omega$. Note that if $\mathcal{V}$ is the version space after getting the labels for $X^m$, then $\mathcal{V} = C$. Therefore $\Pr[C] = \frac{\mathrm{Vol}(\mathcal{V})}{\mathrm{Vol}(B)}$ where $\mathrm{Vol}(B)$ is the volume of the $n$-dimensional unit ball. Using (23) we get

$$\Pr_c\left[-\log \mathrm{Vol}(\mathcal{V}) \geq a + d \log \frac{em}{d} - \log \mathrm{Vol}(B)\right] \leq 2^{-a} \tag{24}$$

Assume that $-\log \mathrm{Vol}(\mathcal{V}) \leq a + d \log \frac{em}{d} - \log \mathrm{Vol}(B)$ (this is true with probability greater then $1 - 2^{-a}$), from lemma 1 we know that there is a ball in $\mathcal{V}$ with radius $r$ such that $r \geq \frac{\mathrm{Vol}(\mathcal{V})\Gamma\left(\frac{n+2}{2}\right)}{n\pi^{n/2}}$. We conclude that there is a ball in $\mathcal{V}$ with radius $r$ such that

$$r \geq \frac{2^{-a}\left(\frac{d}{em}\right)^d \mathrm{Vol}(B)\Gamma\left(\frac{n+2}{2}\right)}{n\pi^{n/2}} \tag{25}$$

□