Advanced Course in Machine Learning Spring 2010

Selective Sampling

Handouts are jointly prepared by Shie Mannor and Shai Shalev-Shwartz

In supervised learning we typically assume that training and test samples are drawn from the same IID distribution. In this paradigm the learner is passive and has no effect on the information it receives. In the query paradigm, the learner is given the power to ask questions. We will ask the question what does the learner gain from this additional power?

The selective sampling paradigm that we study here is also known as the "query filtering paradigm." The idea is that the learner has access to a stream of inputs drawn at random from the input distribution. The learner sees every input and then has to decide if to query a teacher for the correct label. One can easily imagine application in speech processing or web page labelling where a human teacher can be queried, but at a cost.

The problem of selecting the optimal examples for learning is closely related to the problem of *experimental design* in statistics. Experimental design is the analysis of methods for selecting sets of experiments, which correspond to membership queries in the context of learning theory. The goal of a good design is to select experiments in a way that their outcomes, which correspond to labels, give sufficient information for constructing a hypothesis that maximizes some criterion of accuracy.

In today's lecture we will describe and analyze a very simple and natural algorithm for the selective sampling problem. The algorithm is called QBC (query by committee). The results we present today are restricted to a rather limited set of learning problems: concepts are assumed to be deterministic and noiseless

1 Setup

We work in a Bayesian model of concept learning. As in the PAC model, we denote by X an arbitrary sample space over which a distribution $\mathcal{D}$ is defined. For simplicity we assume that X is a Euclidean space $\mathbb{R}^d$. Each concept is a mapping $c : X \to \{0, 1\}$ and a concept class C is a set of concepts.

Bayesian model The Bayesian model differs from the PAC model in that we assume that the target concept is chosen according to a prior distribution $\mathcal{P}$ over C and that this distribution is known to the learner. We shall use the notation $Pr_{x\in D}(\cdot)$ to denote the probability of an event when x is chosen at random from X according to $\mathcal{D}$.

We assume that the learning algorithm has access to two oracles: **Sample** and **Label**. A call to **Sample** returns an unlabelled example $x \in X$, chosen according to (the unknown) $\mathcal{D}$. A call to **Label** with input x, returns $c(x)$, the label of x according to the target concept.

After making some calls to the two oracles, the learning algorithm is required to output a hypothesis $h : X \to \{0, 1\}$. We define the expected error of the learning algorithm as the probability that $h(x) \neq c(x)$, where the probability is taken with respect to:

1. $\mathcal{D}$ over the choice of x,
2. the distribution $\mathcal{P}$ over c; and
3. any random choices made as part of the learning algorithm or of the calculation of the hypothesis h.

We shall usually denote the number of calls that the algorithm makes to **Sample** by m and the number of calls to **Label** by n. Our goal is to give algorithms that achieve accuracy ϵ after making $O(1/\epsilon)$ calls to **Sample** and $O(\log 1/\epsilon)$ calls to **Label**.

bf Some notations: We denote the sequence of unlabeled examples by $X = \{x_1, x_2, \ldots\}$, and use $\langle X, c(X)\rangle = \{x_1, c(x_1), x_2, c(x_2), \ldots\}$ to denote the sequence of labeled examples that is generated by applying c to each $x \in X$. We use $X_{1\ldots m}$ to denote the sequence of the first m elements in X. The *version space* generated by the sequence of labeled examples $\langle X_{1\ldots m}, c(X_{1\ldots m})\rangle$ is the set of concepts $c \in C$ that are consistent with c on X (i.e., $c(x_i) = c'(x_i)$ for all $1 \le i \le m$). We denote the version space that corresponds to the first i labeled examples by $V_i = V(\langle X_{1\ldots i}, c(X_{\ldots i})\rangle)$. The initial version space, $V_0 = V(\emptyset)$, is equal to C. The version space is a representation of the information contained in the set of labeled examples observed by the learning algorithm. A natural measure of the progress of the learning process is the rate at which the size of the version space decreases.

Information: The instantaneous information gain from the i-th labeled example in a particular sequence of examples is defined to $-\log \Pr_{\mathcal{P}} P(V_i)/\Pr_{\mathcal{P}}(V_{i-1})$. Summing the instantaneous information gains over a complete sequence of examples we get the cumulative information gain, which is defined as

$$\mathcal{I}(\langle (x_1, c(x_1)), \ldots (x_m, c(x_m))\rangle) := -\sum_{i=1}^{m} \log \frac{\Pr_{\mathcal{P}} P(V_i)}{\Pr_{\mathcal{P}}(V_{i-1})} = -\log \Pr{}_{\mathcal{P}} P(V_m).$$

A natural measure of the information that we expect to gain from the label of an unlabeled example is the expected instantaneous information gain taken with respect to the probability that each one of the two labels occurs. Let p_0 be the probability that the label of x_m is 0, given that $c \in V_{m-1}$ and let V_m^0 be the version space that results from the label x_m being 0. Let p_1 and V_m^1 be similar for the case $c(x_m) = 1$. The expected information gain of x_i, given V_{i-1} is:

$$\mathcal{G}(x_i|V_{i-1}) = -p_0 \log \frac{\Pr_{\mathcal{P}} P(V_i^0)}{\Pr_{\mathcal{P}}(V_{i-1})} - p_1 \log \frac{\Pr_{\mathcal{P}} P(V_i^1)}{\Pr_{\mathcal{P}}(V_{i-1})} \overset{a}{=} -p_0 \log p_0 - (1-p_0)\log(1-p_0) = \mathcal{H}(p_0),$$

where a follows by the Bayesian estimate and where $\mathcal{H}$ is the Shannon Information for binary variables (we will use $\log$ to denote $\log$ base 2). The information gain is a reasonable measure of the gain that can be expected from asking **Label** for the label of an example. But it does not tell the whole story as we show below.

Gibbs prediction: The "Gibbs prediction rule is to predict rule is to predict the label of a new example x by picking a hypothesis h at random from the version space and labelling x according to it. The random choice of h is made according to the prior distribution P restricted to the version space. (Exercise: The expected error of this prediction error is at most twice larger than the expected error of the optimal prediction rule which is the Bayes rule.) We assume that our learning algorithm has access to an oracle, denoted **Gibbs**, that can compute the Gibbs prediction for a given example $x \in X$ and version space $V \subset C$. Note that two calls to Gibbs with the same V and x can result in different predictions.

2 Two simple examples

Consider the following concept class. Let $X = [0, 1]$, and let the associated probability distribution $\mathcal{D}$ be the uniform distribution. Let the concept class C, consist of all functions of the form $c_w(x) = 1_{\{w \le x\}}$ (we let 1 denote the indicator function) where $w \in [0, 1]$. We define the prior distribution of concepts, $\mathcal{P}$ to be the one generated by choosing w uniformly from $[0, 1]$.

The version space defined by the examples $\langle (x_1, c(x_1)), \ldots (x_m, c(x_m))\rangle$ is (essentially) the segment $V_i = [max(x_i|c(x_i) = 0), min(x_j|c(x_j) = 1)]$. Let us denote by ξ_i the ratio of the probabilities of the version space before and after observing the ith example, i.e., $\xi_i = \Pr_{\mathcal{P}} Vi/\Pr_{\mathcal{P}} V_{i-1}$. The instantaneous information gain of the example $(x_i, c(x_i))$ is $-\log \xi_i$. Given an unlabeled example, the expected instantaneous information gain from x_i is $\mathcal{H}(\xi_i)$. Examples that fall outside the segment have zero expected information gain, while the example that divides the segment into two equal parts obtains the highest possible expected information gain of one bit. This agrees with our intuition because the labels of examples that fall outside the segment are already completely determined by previous labeled examples, while the label of the example that falls in the middle of the version space interval is least predictable. It is easy to show that the probability of a prediction error for the Gibbs prediction rule is equal to the length of the segment divided by three. Thus, if the learner asks for the label of the example located in the middle of the segment, it is guaranteed to halve the error of the Gibbs prediction rule. In this case we see that asking the **Label** to label the example that maximizes the expected information gain guarantees an exponentially fast decrease in the error of the Gibbs prediction rule. In contrast, the expected prediction error after asking for the labels of n randomly chosen examples is $O(1/n)$.

The question is whether constructing queries according to their expected information gain is a good method in general, i.e. whether it always guarantees that the prediction error decreases exponentially fast to zero. The answer to this question is *negative* leading to the conclusion that expected information gain of an unlabeled example is not a very useful criterion for constructing good queries. The essential problem is that the distribution over the examples is completely ignored by this criterion, as the following example demonstrates.

Let the sample space be the set of pairs in which the first element, i, is either 1 or 2, and the second element, z, is a real number in the range $[0, 1]$, i.e., $x \in X = \{1, 2\} \times [0, 1]$. Let $\mathcal{D}$ be the distribution defined by picking both i and z independently and uniformly at random. Let the concept class be the set of functions of the form $c_w(i, z) = 1_{\{w_i \leq z\}}$, where $w \in [0, 1]^2$. The prior distribution over the concepts is the one generated by choosing w uniformly at random from $[0, 1]^2$. Each example corresponds to either a horizontal or a vertical half plane, and the version space, at each stage of learning, is a rectangle. There are always two examples that achieve maximal information gain, one horizontal and the other vertical. Labeling each one of those examples reduces the volume of the version space by a factor of two. However, the probability that the Gibbs rule makes an incorrect prediction is proportional to the *perimeter* of the rectangular version space, and not to its volume. Thus, if the learner always constructs queries of the same type (say, horizontal), only one of the dimensions of the rectangle is reduced, and the perimeter length stays larger than a constant implying that the prediction error also stays larger than a constant.

3 Query by Committee

The algorithm proceeds in iterations. In each iteration it calls **Sample** to get a random instance x. It then calls Gibbs twice, and compares the two predictions for the label of x. If the two predictions are equal, it proceeds to the next iteration. If the two predictions differ, it calls **Label** with input x, and adds the labeled example to the set of labeled examples that define the version space. It then proceeds to the next iteration. We consider a batch learning scenario, in which the learning algorithm is tested only after it has finished observing all of the training examples and has fixed its prediction hypothesis. (Alternative online learning algorithms are possible too.)

Termination rule: To avoid infinite processing of the above rule, we define a termination condition on the iterative process described above. The termination condition is satisfied if a large number of consecutive instances supplied by **Sample** have all equal predictions.

Success criterion: Since we work in a Bayesian setting we cannot use a PAC learning setup and instead use its Bayesian counterpart. We say that the learning algorithm is "successful" if its expected error is small, when trained on a typical sequence of instances. More precisely, we define two parameters, an accuracy parameter $1 > \epsilon > 0$ and a confidence parameter $1 > \delta > 0$. For each choice of the hidden concept, we allow a set of training histories that has probability δ to be marked as atypical training histories. Our requirement is that the expected error over the set of typical training histories is smaller than ϵ. The parameters ϵ and δ are provided to the learning algorithm as input and are used to define the termination criterion.

The QBC algorithm

Input :

- $\epsilon > 0$ – the maximal tolerable prediction error.
- $\delta > 0$ – the desired reliability.

- **Gibbs**– an oracle that computes Gibbs predictions.
- **Sample**– an oracle that generates unlabeled examples.
- **Label**– an oracle that generates the correct label of an example.

Initialize n - the counter of calls to **Label** to 0, and set the initial version space, V_0, to be the complete concept class C.
Repeat until more than t_n consecutive examples are rejected. Where

$$t_n = \frac{1}{\epsilon} \ln \frac{\pi^2(n+1)^2}{3\delta},$$

and n is the number of examples that have been used as queries so far.

1 Call **Sample** to get an unlabeled example $x \in X$ drawn at random according to $\mathcal{D}$.

2 Call **Gibbs**(V_n, x) twice, to get two predictions for the label of x.

3 If the two predictions are equal then reject the example and return to the beginning of the loop. (step 1)

4 Else call **Label**(x) to get $c(x)$, increase n by 1, and set V_n to be all concepts $c' \in V_{n-1}$ such that $c'(x) = c(x)$.

Output as the prediction hypothesis **Gibbs**(Vn, x).

Remark: It is important to notice that the termination condition depends only on ϵ and δ, and not of any properties of the concept class. While the performance of the algorithm does depend on such properties, the algorithm can be used without prior knowledge of these properties.

It is easy to show that if QBC ever stops, then the error of the resulting hypothesis is small with high probability. That is because it is very unlikely that the algorithm stops if the probability of error is larger than ϵ. The harder question is whether QBC ever stops, and if it does, how many calls to **Sample** and to **Label** does it make before stopping? As we shall show in the following two sections, there is a large class of learning problems for which the algorithm will stop, with high probability, after $O(1/\epsilon \log 1/\delta\epsilon)$ calls to **Sample**, and $O(\log 1/\epsilon)$ calls to **Label**.

It is instructive to consider the QBC algorithm from an information perspective. Let us normalize the probability of the version space to one and assume that an example x partitions the version space into two parts with probabilities F and $1-F$, respectively. The probability of accepting the example x as a query is $2F(1-F)$ and the information gain from an example is $\mathcal{H}(F)$. Both of these functions are maximized at $F = 0.5$ and decrease symmetrically to zero when F is increased to one or decreased to zero. Thus, the queries of QBC have a higher expected information gain than random examples. However, it is not true in general that the expected information gain of the queries will always be larger than a constant. The proof of the performance of QBC consists of two parts. In the first part we show that a lower bound on the information gain of the queries does guarantee a fast decrease in the prediction error of QBC. In the second part we show that the expected information gain of the queries of QBC is guaranteed to be higher than a constant in some important cases.

4 Analysis of the Query by Committee Algorithm

More notations: We treat runs of the algorithm as initial segments of infinite runs that would have been generated had there been no termination criterion on the execution of the main loop in QBC. We denote by X the infinite sequence of unlabeled examples that would have been generated by calls to Sample. We use an infinite sequence of integer numbers $I = \{1 \leq i_1 < i_2 < \cdots\}$ to refer to the sequence of indices of those examples that are used as queries to **Label**. This set of examples is denoted X_I. We denote by M the sequence of integers from 1 to m, and use X_M to denote the first m examples in X. We use I_n to denote the first n elements of I. Finally, X_{I_n} indicates the first n examples that are used as queries, and $X_{I \cap M}$ indicates the queries that are chosen from the first m unlabeled examples.

Probabilistic structure: We now present the probabilistic structure underlying the query process. A point in the sample space is a triple c, X, I. The probability distribution over this space is defined as follows:

1. The target concept c is chosen according to $\mathcal{P}$.
2. Each component in the infinite sequence X is chosen independently according to $\mathcal{D}$.
3. Fixing c and X , we define the distribution of the first n elements of I according to the probability that algorithm QBC calls **Label** on the iterations indexed by I_n. The distribution on I is the limiting distribution for $n \to \infty$ (Exercise: why does it exist?)

We denote the distribution we have defined on the triplets c, X, I by Δ and use $\Pr_\Delta$ and $\mathbb{E}_\Delta$ to indicate the probability and the expectation taken with respect to this distribution.

We say that the expected information gain of queries made by QBC for the learning problem of concept class C,concept distribution P, and input distribution $\mathcal{D}$, is uniformly lower bounded by $g > 0$ if the distribution generated by QBC satisfies that:

$$\Pr{}_\Delta \left(\mathbb{E} \left[\mathcal{G}x_{i_{n+1}} | V(\langle X_{I_n}, c(X_{I_n}) \rangle) \big| X_{I_n}, c(X_{I_n}) \right] > g \right) = 1.$$

(In words: For any version space that can be reached by QBC with non-zero probability, the expected information gain from the next query of QBC is larger than g.)

The following is the main theorem concerning the analysis of QBC.

Theorem 1 *If a concept class C has VC-dimension $0 < d < \infty$ and the expected information gain of queries made by QBC is uniformly lower bounded by $g > 0$ bits, then the following holds with probability larger than $1 - \delta$ over the random choice of the target concept, the sequence of examples, and the choices made by QBC:*

- *The number of calls to* **Sample** *that QBC makes is smaller than*

$$m_0 = \max \left\{ \frac{4d}{e\delta}, \frac{160(d+1)}{ge} \max \left(6, \ln \frac{80(d+1)}{\epsilon \delta^2 g} \right)^2 \right\}.$$

- *The number of calls to* **Label** *that QBC makes is smaller than*

$$n_0 = \frac{10(d+1)}{g} \ln \frac{4m_0}{\delta}.$$

- *The probability that the Gibbs prediction algorithm that uses the final version space of QBC makes a mistake in its prediction is smaller than ϵ.*

A conclusion from the theorem is that an exponentially small fraction of the number of calls to **Sample** are used for **Label**.

We now provide an outline of the proof which we make formal later.

QBC terminates. Two contradicting trends: 1. After observing many labeled examples the conditional distribution of the labels of new examples is highly biased to one of the two labels— information gained from knowing the label of a random example is small. This, in turn, means that the increase in the cumulative information from a sequence of random examples becomes slower and slower as the sequence gets longer. 2. Since information gained from the queries of QBC is lower bounded by a constant, then the cumulative information gain from the sequence of queries increases linearly with the number of queries. The only way in which both rates of increase can hold without violating this simple inequality is if the number of examples that are rejected between consecutive queries increases with the number of queries. As a result the termination criterion of QBC will hold, and the algorithm will output its final prediction rule after a reasonably small number of queries.

QBC returns a "good" rule. The prediction rule that is output is the Gibbs prediction rule, using the final version space that is defined by all the labeled examples seen so far. The probability of making a prediction error using this rule is, by definition, equal to the probability of a disagreement between a hypothesis that is randomly chosen according to the prior distribution restricted to the version space and a concept that is independently chosen according to the same distribution. This probability is also equal to the probability of accepting a random example as a query when using this version space. The termination condition is fulfilled only if a large number of random examples are not accepted as queries, which implies that the probability of accepting a query or making a prediction mistake when using the final version space is small.

The proof follows from the following lemmata:

Lemma 1 *If the expected instantaneous information gain of the query algorithm is uniformly lower bounded by $g > 0$ bits, then*

$$\Pr_{\Delta}\left(\mathcal{I}(\langle X_{I_n}, c(X_{I_n})\rangle) < g\frac{n}{2}\right) < \exp\{-gn/10\}.$$

Proof Define

$$Y_i = \mathcal{I}(\langle X_{I_i}, c(X_{I_i})\rangle) - \mathcal{I}(\langle X_{I_{i-1}}, c(X_{I_{i-1}})\rangle) - g.$$

It follows that $-g \leq Y_i \leq 1 - g$ (since information is between 0 and 1) and that Y_i is a sub-martingale. From Hoeffdings bound on the tails of bounded step sub-martingales from which we know that for any $\epsilon > 0$

$$\Pr\left(\sum_{i=1}^{n} Y_i \leq -\epsilon n\right) \leq \left[\left(\frac{g}{g+\epsilon}\right)^{g+\epsilon}\left(\frac{1-g}{1-g-\epsilon}\right)^{1-g-\epsilon}\right]^n$$

Taking $\epsilon = \lambda g$ and then $\lambda = 1/2$ and some algebra yields the bound.

Lemma 2 *The probability that the predictions made by QBC are wrong (after its main loop has terminated) is smaller than ϵ with probability larger than $1 - \delta/2$.*

Proof Assume that the probability of a wrong prediction is larger than ϵ. This implies that the probability of accepting a random example as a query with the final version space, is also larger than ϵ. It thus remains to show that the probability that QBC stops when the probability of accepting a query is larger than ϵ is smaller than $\delta/2$. The termination condition of QBC is that all t_n examples tested after the nth query are rejected. If the probability of accepting a random example is larger than ϵ, then this probability is smaller than $(1?\epsilon)^{t_n}$. From the definition of t_n we get that:

$$(1-\epsilon)^{\frac{1}{\epsilon}\ln(\pi^2\frac{(n+1)^2}{3\delta})} \leq e^{-\ln(\pi^2\frac{(n+1)^2}{3\delta})} = \frac{3\delta}{\pi^2(n+1)^2}.$$

Summing this probability over all possible values of n from zero to infinity we get the statement of the lemma.

Lemma 3 *Assume a concept c is chosen at random from a concept class with VC dimension d. Fix a sequence of examples X and recall that X_m denotes the first m examples. Then*

$$\Pr_{\mathcal{P}}\left(\mathcal{I}\left(\langle X_m, c(X_m)\rangle\right) \geq (d+1)\log(em/d)\right) \leq \frac{d}{em}.$$

Proof Recall from From Sauers Lemma the number of different labelings created by m examples is at most $(em/d)^d$ the expected cumulative information gain equals the base 2 entropy of the distribution of the labels and is maximized when all the possible labelings have equal probability. This gives an upper bound of $d\log(em/d)$ on the expected cumulative information gain. Labelings that have cumulative information gain larger than a this expected value, must have probability that is smaller by 2^a than the labels in the equipartition case. As the number of possible labelings remains the same, the total probability of all concepts that give rise to such labelings is at most 2^{-a}. Choosing $a = \log(em/d)$ yields get the bound.

Proof of Theorem 1

We consider a randomly chosen element of the event space $c, X, \mathcal{I}$. We consider the first m_0 random samples presented to QBC and the first n_0 samples it queries **Label**. We denote the number of queries that QBC makes during the first m_0 examples by n. The claim of the theorem is that, with probability at least $1-\delta$, the algorithm halts before testing the $m+1$ example, the number of queries it makes, n, is smaller than n_0, and the hypothesis it outputs upon halting has error smaller than ϵ. We shall enumerate a list of conditions that guarantee that all of these events occur for a particular random choice of examples and of internal randomization in QBC. By showing that the probability of each of those conditions to fail is small we get the statement of the theorem. The conditions are:

1. The cumulative information content of the first n_0 queries is at least $gn_0/2$. From Lemma 1 we get that in order for this condition to hold with probability larger than $1?\delta/4$ it is sufficient to require that $n_0 \geq 10/g \ln(4/\delta)$.

2. The cumulative information content from the first m_0 examples is at most $(d+1)(\log(em_0/d))$. From Lemma 3 we get that in order for this condition to hold with probability larger than $1-\delta/4$ it is sufficient to require that $m_0 \geq \frac{4d}{e\delta}$.

3. The number of queries made during the first m_0 examples, n, is smaller than n_0. The condition follows from conditions 1 and 2 if

$$\mathcal{I}(\langle X_{I_{n_0}}, c(X_{I_{n_0}})\rangle) \geq \mathcal{I}(\langle X_{I_{n_0}}, c(X_{I_{n_0}})\rangle)$$

   (When $n \geq n_0$ the information gained from the queries asked during the first m_0 examples is larger than the total information gained from the m_0 examples, which is impossible. We need $n_0 > 2(d+1)/g \log(em_0/d)$ for the information inequality to hold.

4. The number of consecutive rejected examples guarantees that the algorithm stops before testing the m_0+1 example. There are $m_n - n$ rejected examples so that the length of the shortest run of rejected examples is at least $(m_0-n)/(n+1)$. Requiring that this expression is larger than t_n and using the fact that $n < n_0$ we need that

$$m_0 \geq \frac{2(n_0+1)}{\epsilon} \ln\left[\frac{\pi^2}{3\delta}(n_0+1)^2\right].$$

5. The Gibbs prediction hypothesis that is output by the QBC has probability smaller than ϵ of making a mistaken prediction. This is straightforward from Lemma 2 and happens with probability smaller than $\delta/2$.

The result follows from combining the above and some algebra. Note that we accumulate the error of the above conditions.

5 So when does QBC work?

The main issue with Theorem 1 is that we require high information gain by queries. It is by no means trivial that one can guarantee high information gain by queries. The following paragraph describes a geometric case where, surprisingly, information gain can be guaranteed.

Define the domain, X, to be the set of all pairs of the form (x, t), where x is a vector in $\mathbb{R}^d$ whose length is 1, which we refer to as the "direction of the example, and t is a real number in the range $[?1, +1]$, to which we refer as the offset. Assume that $\mathcal{D}$ is uniform. The concept class, C, is defined to be a set of binary functions over X, parameterized by vectors w in $\mathbb{R}^d$ that are defined as follows:

$$c_w(x,t) = 1_{\{w \cdot x \geq t\}}.$$

Note that t is also an input to the concept so this is not a standard linear classifier. The information gain from random examples vanishes as the dimensions grows. The reason for this is that in high dimension, the volume of the sphere is concentrated near the equator. A typical random example will cut the sphere some distance away from the equator, in which case the sphere will fall into two pieces of very unequal volume. The piece containing the equator will contain almost all of the volume. Query by committee solves this problem by choosing two random points in the sphere. Since these two points are likely to be near the equator, an example that separates them is likely to be near the equator. For this reason, query by committee can attain an information gain that remains bounded away from 0 in high dimensions.

It is possible to prove that the information gain from every query to QBC is bounded from below by a constant. The proof is based on variational analysis and is not provided here.

For perceptrons we can obtain a similar bound under certain conditions on the prior and the distribution. Recall that a perception can be defined by:

$$c_w(x) = 1_{\{w \cdot x \geq 0\}},$$

where x, w in $\mathbb{R}^d$ and $\|w\| \leq 1$.

In order to obtain a lower bound on the information gain we need to require some regularity properties on the $\mathcal{D}$ and $\mathcal{P}$. We say that a density D is within λ of D' if for every measurable set A, we have that

$$\lambda \leq \Pr{}_D(A)/\Pr{}_{D'}(A) \leq 1/\lambda.$$

It is possible to prove that the information gain is bounded from below when the prior distribution is within λ_P of uniform and the input distribution is within λ_D of uniform, is at least bounded from below (and depends on λ_P and λ_D).

To prove that there exists a lower bound on the information gain of the queries of QBC we need to assume that the initial version space is not the complete unit sphere, but is restricted to be within a cone. That is, there has to exist a unit vector w_0 such that for any $w \in V_0$, the dot product $w \cdots w_0$is larger than some constant $\alpha > 0$. To guarantee that this condition holds we can use an initial learning phase, prior to the use of QBC, that does not use filtering but rather queries on all the random instances supplied by **Sample**. Standard results provide the number of training examples that are needed to guarantee that the prediction error of an arbitrary consistent hypothesis is small (with high probability). As the distribution of the instances is close to uniform, a small prediction error implies that the hypothesis vector is within a small angle of the vector that corresponds to the target concept. We conclude with the formal result for perceptrons:

Theorem 2 *For any $\alpha > 0$, let C_α be the d dimensional perceptron concept class restricted to those concepts c_w, such that $w_0 \cdot w > \alpha$ for some unit vector w_0. Let the prior distribution over C_α be within λ_P of uniform and the input distribution be within λ_D from uniform. Then the expected information gain of the queries of QBC is larger than $0.672\alpha^{5d}\lambda_P^4\lambda_D^4$.*