THE LIGHT BULB PROBLEM [1]

Ramamohan Paturi[2] **Sanguthevar Rajasekaran**[3] **John Reif**[3]
Univ. of California, San Diego Univ. of Pennsylvania Duke University

[1]A preliminary version of this paper was presented in the second annual workshop on Computational Learning Theory held in Santa Cruz, CA, in 1989.

[2]This work was done while the author was at Harvard University in 1985-86 and was supported by the NSF grant MCS-83-02385.

[3]Supported by NSF-DCR-85-03251 and ONR contract N00014-87-K-0310 and AFOSR-87-0386.

Running Title: The Light Bulb Problem

Corresponding Author:

Ramamohan Paturi
Department of Computer Science
Mail Code C-014
University of California, San Diego
La Jolla, CA 92093

ABSTRACT

In this paper, we consider the problem of correlational learning and present efficient algorithms to determine correlated objects.

1. INTRODUCTION

Correlational learning, a subclass of unsupervised learning, aims to identify statistically correlated groups of attributes. In this paper, we consider the following correlational learning problem due to L. G. Valiant, 1985 and 1988: We have a sequence of n random light bulbs each of which is either on or off with equal probability at each time step. Further, we know that a certain pair of bulbs is positively correlated. The problem is to find efficient algorithms for recognizing the pair of light bulbs with the maximum correlation.

It is experimentally observed that humans can detect such a correlated pair *quickly* after being exposed to a 'small' number of samples. The motivation for the present problem is the desire to provide an algorithmic explanation for this phenomenon Valiant, 1984 and 1985. Some preliminary results are reported in Paturi, 1988.

In this paper, we consider a more general version of the basic light bulb problem. In the general version, we assume that the behavior of the bulbs is governed by some unknown probability distribution. Our goal would be to find the pair of bulbs with the largest pairwise correlation. We then consider the more general problem of k–way correlations.

Mathematically, we can regard each light bulb l_i at time step t as a random variable X_i^t which takes the values ± 1. We call $(X_1^t, X_2^t, \ldots, X_n^t)$ the t–th *sample.* We also assume that the behavior of the light bulbs is independent of their past behavior. In other words, the samples are independent of each other. We would like to find the desired object (k–tuples with the maximum correlation) *with high probability.* The complexity measures of interest are sample size and the number of operations to determine the desired object.

Before we proceed further, we introduce some definitions and facts from probability theory.

We define the correlation of a pair of light bulbs l_i and l_j as $\mathbf{P}[X_i = X_j]$. In general, for any $k \geq 2$, the correlation coefficient of the k–tuple $(l_{i_1}, l_{i_2}, \ldots, l_{i_k})$ of light bulbs is defined as $\mathbf{P}[X_{i_1} = X_{i_2} = \cdots = X_{i_k}]$.

Let Y be a random event with probability of success p. Consider the probability distribution of the number of successes in k independent occurrences of the event Y. The following gives bounds for the probability of the tails of this distribution.

$$\mathbf{P}[\text{No. of Successes} < (1-\delta)pk] \leq e^{-\delta^2 pk/2} \tag{1}$$

$$\mathbf{P}[\text{ No. of Successes } > (1+\delta)pk] \leq \left[\frac{e^{\delta}}{(1+\delta)^{(1+\delta)}}\right]^{pk} \tag{2}$$

where pk is the expected number of successes, and $\delta > 0$.

We say that a statement holds *with high probability* , if it holds with probability $1 - n^{-\alpha}$ for some $\alpha > 0$.

2. A QUADRATIC–TIME ALGORITHM

We now a present an algorithm called algorithm Q which samples each pair of bulbs for $O(\ln n)$ time to determine the pair with the largest correlation.

Let $S_{ij}^t = |\{1 \leq u \leq t | X_i^u = X_j^u\}|$. In other words, S_{ij}^t is the number of times the bulbs l_i and l_j have identical output when t samples are considered. Let p_1 be the largest pairwise correlation and p_2 be the second largest correlation. Let $p_2 = p_1(1 - 1/\gamma)$ where $\gamma > 1$. We characterize the performance of our algorithms in terms of the parameter γ. We can see that the value of S_{ij}^t tends to be larger if the pair (i, j) is more correlated. Hence, if we take a sufficiently large t, we can guarantee that the pair (i, j) with the largest S_{ij}^t has the maximum correlation provided p_1 and p_2 are sufficiently separated. It can easily be seen that the value of t depends on the correlation p_1 and p_2 which we may not know *a priori.* To overcome this problem, we check if the largest S_{ij}^t is greater than certain threshold $T' = O(\gamma^2 \ln n)$. If the largest S_{ij} can be so separated, we declare (i, j) as the correlated pair. Otherwise, we look at the random variables X_i^{t+1} at step $t + 1$ and update S_{ij}^{t+1} and repeat the above computation until we succeed.

Each step of this computation takes $O(n^2)$ operations. The following theorem gives us number of the iterations required.

Theorem 1 *Let $\alpha > 0$. Algorithm Q terminates when $t = O(\gamma^2 \ln n/p_1)$ and finds the pair with the maximum correlation with probability $1 - O(n^{-\alpha})$.*

Proof: Let $\alpha' = \alpha + \epsilon$ for for any $\epsilon > 0$, and let $T = 2c\gamma^2 \ln n/p_1$ and $T' = c\gamma^2 \ln n$ for some constant c which will be determined later. We will now claim that the following holds for a suitably chosen constant c: If the pair (i, j) is the pair with the maximum correlation p_1, then,

$$\mathbf{P}[S_{ij}^T < T'] < n^{-\alpha'},$$

otherwise, for $t \leq T$,

$$\mathbf{P}[S_{ij}^t \geq T'] < n^{-2-\alpha'}.$$

Since the number of the events $[S_{ij}^t \geq T']$ is at most n^2T, we can bound the probability than an undesirable event happens by $O(n^{-\alpha})$.

We select $c \geq 2 + 4\alpha'$. It is easy to verify, using the bounds on the probability of large deviations, that the above inequalities will be satisfied. This proves the theorem. □

It follows that the above algorithm takes $O(n^2 \ln n)$, where n^2 is the number of pairs considered. More generally, given m pairs of random variables, the above algorithm finds the pair with the largest correlation in time $O(m \ln n)$. Can we do better? We now present two different algorithms which take only *subquadratic* time.

3. ALGORITHM A

For each i and t, we will consider the string $s_i^t = X_i^1 X_i^2 \ldots X_i^t$ as an element of the hypercube $\{-1, 1\}^t$. Consider the sphere of radius εt with center s_i^t. For $\varepsilon < 1/2$, the volume $V(\varepsilon t)$ of this sphere is at most $\varepsilon t e^{h(\varepsilon)t}$ where $h(\varepsilon) = \varepsilon \ln(1/\varepsilon) + (1-\varepsilon) \ln(1/(1-\varepsilon))$ is the entropy function. If ε and t are selected appropriately, it is unlikely that the spheres corresponding to two different i and j would intersect if the light bulbs l_i and l_j do not have the maximum correlation. On the other hand, the spheres corresponding to the light bulbs with maximum correlation would intersect with high probability. This idea can be implemented as an algorithm with $O(nV(\varepsilon t))$ operations.

We consider the random variables $X_i^1, X_i^2, \ldots X_i^t$ for each i. We find a pair with the shortest distance in the t–dimensional hypercube by exploring the sphere around each $X_i^1 X_i^2 \ldots X_i^t$. Let (i,j) be such a pair. We check if $S_{ij}^t > O(\ln n)$. If so, we declare (i,j) as the pair with the maximum correlation. Otherwise, we consider the $t+1$–dimensional hypercube and repeat.

Such a strategy would result in a subquadratic time algorithm in the case p_1 and p_2 are well-separated. For example, in the special case when all but one pair of bulbs are independent and output 0 or 1 with equal probability, we get a subquadratic time algorithm. The following theorem gives the precise condition under which we can get a subquadratic time algorithm.

Theorem 2 *Let $p_2 = 1/2$. For any $p_1 > 0.92$, there exists an $\alpha > 0$ such that, with probability $1-O(n^{-\alpha})$, Algorithm A finds the maximally correlated pair in time $O(n^{1+\mu} \ln n)$ for some $\mu < 1$.*

Proof: Let $T = c \ln n$ and $T' = d \ln n$ for some constant c and d to be determined later. For $t \le T$, let εt be the radius of the sphere to be explored in the t–dimensional hypercube. We want to find c, d and ε that satisfy the following criteria.

We want the volume of the sphere to be less than $n^{1-\mu}$ for some $\mu > 0$. This gives the condition

$$c = T/\ln n < 1/h(\varepsilon).$$

We then want to ensure that the maximum correlated pair (i,j) has its $S_{ij}^T \le (1-2\varepsilon)T$ and $S_{ij}^T > T'$ with high probability. For any other pair (i,j), we want to have that, for $t \le T$, $S_{ij}^t < T'$ with high probability. More precisely, we want to show that there exists an $\alpha > 0$ such that

$$\mathbf{P}[S_{ij}^T \le (1-2\varepsilon)T] > 1 - n^{-\alpha}$$

and

$$\mathbf{P}[S_{ij}^T > T'] > 1 - n^{-\alpha}$$

for the pair (i,j) with the maximum correlation, and, for all other pairs,

$$\mathbf{P}[S_{ij}^t < T'] > 1 - n^{-2-\alpha}.$$

By using the large deviation theorems, it can easily be verified that when $p_2 = 1/2$ and $p_1 > 0.92$, we can determine the constants c, d and ε satisfying all the conditions mentioned above. This proves the theorem.

Algorithm A gives a subquadratic algorithm by looking at $O(\ln n)$ samples, only if the maximum correlation is sufficiently high. We now give an algorithm which does not suffer from this disadvantage but uses $O(n^{\varepsilon})$ ($\varepsilon < 1$) samples. We later use bootstrap technique to reduce the number of samples to $O(\ln n)$.

4. ALGORITHM B

To understand this algorithm, consider the special case with $p_1 = 1$. In this case, the problem is reduced to sorting. We want to determine the pair that produced identical outputs. This can be done by sorting the strings $s_i^t = X_i^1 X_i^2 \ldots X_i^t$. With $t = O(\ln n)$, we can identify the desired pair with high probability. The constant involved depends on p_2. The total number of operations in this special case is $O(n \ln n)$.

Even in the more general case, we can use the above idea to reduce the number of pairs to be considered. We classify the random variables based on their s_i^t. We say that two bulbs i and j fall into the same bucket if $s_i^t = s_j^t$. We consider all the pairs (i, j) for i and j in the same bucket. We select t such that no more than c variables X_i have the same s_i^t for some large enough constant c. This ensures that number of pairs to be considered is $O(n)$. On the other hand, if t is not too large, maximally correlated pair falls into same bucket with a sufficiently large probability. If this is repeated for a sufficient number of times, we can find the maximally correlated pair with high probability. In the following, we give the algorithm and its analysis.

Algorithm B:

Let $PAIRS$ be the empty list $\emptyset$;
for $i = 1$ *to* $O(n^{\frac{\ln p_1}{\ln p_2}} \ln n)$ *do*

(step1) Take $t = c \ln n$ (for some constant c to be determined later) and obtain the sample vectors $(X_1^j, X_2^j, \ldots, X_n^j)$ for $j = 1, 2, \ldots, t$.

(step2) Sort the n strings (of length $c \ln n$ each) $s_i^t = X_i^1 X_i^2 \cdots X_i^t$ for $i = 1, \ldots, n$.

(step3) For each bucket of bulbs obtained in step 2, consider all possible pairs of bulbs from the bucket. (A bucket is a group of all bulbs with equal value in all the t sample steps). Add any new pairs to the list $PAIRS$.

Using the algorithm Q that checks for correlation of each of the pairs, output the pair with the maximum correlation from among the pairs in PAIRS.

The following theorem gives the time and sample complexity of algorithm B. Let $p_2 = (1 - 1/\gamma)p_1$.

Theorem 3 *Algorithm B finds the pair with the largest correlation in expected time* $O(c' n^{1+\frac{\ln p_1}{\ln p_2}} \ln^2 n)$ *with probability* $1 - O(n^{-\alpha})$.

Proof: For any i, the probability that $s_i^t = s_j^t$ is at most $p_2^{(c \ln n)}$ if (i, j) is not the pair with the maximum correlation. Hence, if we select $c = 1/\ln(1/p_2)$, the expected number of pairs added to the list PAIRS in each iteration of the outer loop will be $O(n)$.

With this value of c, the probability that the maximum correlated pair falls into the same bucket is $(p_1)^{\ln n/\ln(1/p_2)} = n^{\ln p_1/\ln(1/p_2)}$. Therefore, if the outer loop of the algorithm B is executed $c' n^{\ln p_1/\ln p_2} \ln n$ times, then the maximum correlated pair will be included in the list PAIRS with probability $1 - O(n^{-\alpha})$. The constant c' depends on α.

Since the expected length of the list PAIRS is $O(n^{1+\ln p_1/\ln p_2} \ln n)$, we find the correct pair with high probability in expected time $O(\gamma^2 \alpha n^{1+\ln p_1/\ln p_2} \ln^2 n/p_1)$ using the algorithm Q. □

The previous theorem gives a bound on the expected run time of the algorithm. With an additional log factor, we can show that the time bound holds with high probability. We use the following fact for this purpose.

Lemma 1 *If a randomized algorithm $\mathcal{A}$ runs in expected time T and produces the correct answer with a high probability, then there is another random algorithm that runs in time $O(T \log_2 n)$ and produces the correct answer with high probability.*

Proof: Run $c \log_2 n$ copies of the algorithm independently for a sufficiently large c. Terminate each copy after $2T$ steps. We will show that at least one of the runs must have terminated with the correct answer with high probability.

Using Markov's inequality, the time needed for $\mathcal{A}$ to terminate with the correct answer will exceed $2T$ with probability at most $1/2$. Thus, the probability that none of the $c \log_2 n$ runs finds the correct answer is at most $(1/2)^{c \log_2 n} = n^{-c}$. Therefore, the modified algorithm computes the correct result with high probability and runs in time $O(T \log_2 n)$. □

An application of the above lemma to algorithm B implies that we can find the maximum correlated pair with high probability in time $O(\gamma^2 \alpha n^{1+\ln p_1/\ln p_2} \ln^3 n/p_1)$.

5. BOOTSTRAP TECHNIQUE

Algorithm B uses a large number ($O(n^{\ln p_1/\ln p_2})$) of samples. We can reduce the sample size to $O(\log n)$ using the bootstrap technique (Diaconis and Efron, 1980, Efron, 1979 and 1982).

Assume that we are given a data set (i.e., a random sample of size d) $D = \{x_1, x_2, \ldots, x_d\}$ from an unknown distribution, and we want to estimate some statistic, say θ. The idea of bootstrap is to generate a large number of new data sets from D and estimate θ on each one of the generated data sets to obtain a better estimate of θ. A data set is generated by drawing samples independently with replacement from D with each element in D being equally likely.

We can use this idea of bootstrap in algorithm B. We first make $d \log n$ observations (for some constant $d \geq \gamma$, where $p_2 = p_1(1-1/\gamma)$). This sample is then used to generate data sets for step 1 of the algorithm.

The reason why bootstrap works in our case can be explained as follows. The separation between the maximum correlated pair and the second largest correlated pair is preserved in the sample data (with high probability). More precisely, using the large deviation bounds, , we can show that the **sample** correlation of the pair with the largest correlation is at least $(1-\epsilon)p_1$ (for any $\epsilon > \sqrt{\frac{2\alpha}{dp_1}}$) with probability $\geq (1-n^{-\alpha})$ (for any $\alpha > 1$). Also the sample correlation of the pair with the second largest correlation is at the most $(1+\delta)p_2$ (for any $\delta > \sqrt{\frac{2(2+\alpha)}{dp_2}}$) with probability $\geq (1-n^{-\alpha})$. Therefore the expected run time of the modified algorithm is $O(n^{1+\frac{\log((1-\epsilon)p_1)}{\log((1+\delta)p_2)}} \ln^2 n)$. The constant will depend on γ and α.

6. k–WAY CORRELATION

We can modify the algorithm B to detect a k–tuple with the largest k–way correlation. The run time of the algorithm would then be $O(n^{k\frac{\log p_1}{\log p_2}+1} \ln^2 n)$ where p_1 is the largest k–way correlation and p_2 is the second largest k–way correlation. Let $p_2 = p_1(1-1/\gamma)$.

Let $t = c \log n$. Algorithm B_k is the same as Algorithm B except that c will be chosen as $k/\ln(1/p_2)$. Also, instead of obtaining pairs in step3, we obtain k–tuples here.

Algorithm $\mathbf{B}_k$:

$TUPLES = \emptyset$;

for $i = 1$ to $O(n^{k \log p_1/\log p_2} \ln n)$ *do*

(step1) Observe the light bulbs for $t = c \log n$ time and obtain the sample vectors $(X_1^j, X_2^j, \ldots, X_n^j)$ for $j = 1, 2, \ldots, t$.

(step2) Sort the n numbers ($c \log n$ bits each) $(X_1^1 X_1^2 \ldots X_1^t), (X_2^1 X_2^2 \ldots X_2^t), \ldots, (X_n^1 X_n^2 \ldots X_n^t)$ using radix sort.

(step3) Obtain all possible k-tuples from out of the bulbs in each bucket of step2. Add these k-tuples to $TUPLES$.

Find the k–tuple with the maximum correlation using a variant of Algorithm Q.

Theorem 4 *Algorithm* B_k *finds the* k*-tuple with the largest correlation with probability* $\geq 1 - n^{-\alpha}$ *in time* $O(n^{k\frac{\log p_1}{\log p_2}+1} \ln^3 n)$.

Proof. The analysis of Algorithm B_k is similar to that of Algorithm B and hence is omitted. □

7. k–WAY CORRELATION FOR ARBITRARY k

The algorithm in the previous section has a run time exponential in k. Is there a polynomial time algorithm for k-WAY CORRELATION for arbitrary k? The answer is yes if $O(\ln n)$ sample size suffices to solve the problem. This implies that p_1 and p_2 should be separated by a constant.

Given that $O(\log n)$ sample size suffices, the problem can be restated as follows. We define the observation matrix M of size n by t to be $M_{ij} = X_i^j$. A row corresponds to a light bulb and a column corresponds to a sample time step. The correlation of any k rows (k–tuple) is defined to be the number of identical columns in M when restricted to these k rows. The problem is to find the k rows with the maximum correlation.

The maximum correlated k-tuple will have i identical columns and no other k-tuple will have greater than i identical columns for some $1 \leq i \leq t$. The algorithm for k–WAY CORRELATION for an arbitrary k would exhaustively check if there is a k-tuple with j identical columns for $j = 1, 2, \ldots, t$.

Algorithm ARBITRARY k-WAY:

for $j = 1, 2, \ldots, t$ *do*

for each possible choice of j columns (there are $\binom{t}{j}$ choices in all) *do*

Find if there is a k-tuple with identical entries in these j columns. This can be done using radix sort of the n (j-bit) integers (one corresponding to each row).

If there is a bucket with $\geq k$ bulbs then it means there are k rows with j identical columns. If so, set $MAXCORR = j$ and register the corresponding bucket in $BUCKET$.

Output $MAXCORR$ and a k-tuple from $BUCKET$

Theorem 5 *The above algorithm correctly finds the maximum correlated k-tuple with high probability provided the sample size is sufficient to separate the maximum correlated k–tuple from the rest with high probability. Furthermore, if the sample size is $O(\log n)$, then the algorithm takes $n^{O(1)}$ steps.*

Proof. The correctness of the algorithm is obvious. If $t = c \log n$, clearly, the innerloop will be executed $\sum_{j=1}^{c \log n} \binom{c \log n}{j} = n^c - 1$ times and each execution takes $O(n)$ time. □

Observe that the total run time is independent of k. If t were polynomial in n, then this algorithm runs in exponential time.

OPEN PROBLEM: Is there an $O(n \ln n)$ algorithm for determining the pair with the maximum correlation?

Acknowledgments: We would like to thank Prof. Les Valiant for suggesting this problem and for listening to our solutions.

References

[1] DIACONIS, P., AND EFRON, B. (1980), Computer-Intensive Methods in Statistics. *Annals of Statistics*, pp 116–130.

[2] EFRON, B. (1979), Bootstrap Methods: Another Look at the Jackknife, *Annals of Statistics* **7**, pp 1–26.

[3] EFRON, B. (1982), “The Jacknife, the Bootstrap and Other Resampling Plans”, SIAM, Philadelphia, Pennsylvania.

[4] KEARNS, M., AND LI, M. (1988), Learning in the Presence of Malicious Errors. *in* "Proceedings of the 20th Symposium on Theory of Computing", pp 267–280.

[5] PATURI, R. (1988), The Light Bulb Problem. Technical Report CS88–129, University of California, San Diego.

[6] PATURI, R., RAJASEKARAN, S., and REIF, J.H. (1989), The Light Bulb Problem. *in* "Second Work shop on Computational Learning Theory," pp. 261–268.

[7] VALIANT. L.G. (1984), A Theory of the Learnable. *Communications of the ACM* **27**, pp 1134–1142.

[8] VALIANT, L.G. (1985), Learning Disjunctions of Conjunctions. *in* "Proceedings of the 9th International Joint Conference on Artificial Intelligence", Los Angeles, pp 560–566.

[9] VALIANT, L.G. (1985), Private Communication.

[10] VALIANT, L.G. (1988), Functionality in Neural Nets. *in* "First Work shop on Computational Learning Theory", pp 28–39.