Learning Binary Relations Using Weighted Majority Voting

Sally A. Goldman*
Manfred K. Warmuth†

UCSC-CRL-93-51
December 29, 1993

Baskin Center for
Computer Engineering & Information Sciences
University of California, Santa Cruz
Santa Cruz, CA 95064 USA

ABSTRACT

In this paper we apply a weighted majority voting algorithm to the problem of learning a binary relation between two sets of objects. When using exponentially many weights, the mistake bound of the algorithm is essentially optimal. We present a construction where a number of copies of our algorithm divide the problem amongst themselves and learn the relation cooperatively. In this construction the total number of weights is polynomial. The mistake bounds are non-optimal (at least when compared to the best bound obtainable when computational resources are ignored) but significantly improve previous mistake bound bounds achieved by polynomial algorithms. Moreover our method can handle noise, which widens the applicability of the results.

---

*Net address: sg@cs.wustl.edu. Mailing address: Department of Computer Science, Washington University, St. Louis, Missouri 63130. This author was supported in part by NSF grant CCR-91110108.

†Net address: manfred@cs.ucsc.edu. Mailing address: Department of Computer Science, University of California, Santa Cruz, California 95064. This author was supported by ONR grant NO0014-91-J-1162 and NSF grant IRI-9123692.

1 Introduction

In this paper we demonstrate how weighted majority voting can be applied to obtain robust algorithms for learning binary relations. Following Goldman, Rivest and Schapire [GRS93], a binary relation is defined between two sets of objects, one of cardinality n and the other of cardinality m. For all possible pairings of objects, there is a predicate relating the two sets of variables that is either true (1) or false (0). The relation is represented as an $n \times m$ matrix M of bits, whose (r, j) entry is 1 if and only if the relation holds between the corresponding elements of the two sets. Furthermore, there are a limited number of object types. Namely, the matrix M is restricted to have at most k distinct row types amongst its n rows. (Two rows are of the same type if they agree in all columns.) This restriction is satisfied whenever there are only k types of objects in the set of n objects being considered in the relation.

We shall study the problem of learning binary relations under the standard on-line (or incremental) learning model [Lit89, Lit88]. The *learning session* consists of a sequence of *trials*. In each trial, the learner must predict the value of some unknown matrix entry that has been selected by the adversary[1]. After predicting the learner receives the value of the matrix entry in question as *feedback*. If the prediction disagrees with the feedback, then we say the learner has made a *mistake*. The learning session continues until the learner has predicted each matrix entry. The goal of the learner is to make as few mistakes as possible.

Since the number of binary relations is at most $2^{km}k^n$ the standard halving algorithm [BF72, Lit88, Ang88] makes at most $km + n \lg k$ mistakes[2]. Observe that the halving algorithm can be viewed as keeping $2^{km}k^n$ weights, one weight per possible binary relation. Initially, all weights start at 1, and whenever a binary relation becomes inconsistent with the current partial matrix its weight is set to 0. To make a prediction for a given matrix entry, each binary relation votes according to its bit in that entry. Finally, the halving algorithm predicts according to the majority of consistent relations (i.e. those with weight 1), and thus each mistakes halves the total weight in the system. Since the initial weight is $2^{km}k^n$ and the final weight is at least 1, at most $km + n \lg k$ mistakes can occur.

Observe that the time used to make each prediction is linear in the number of weights. Thus we are interested in algorithms that use a small number of weights in representing their hypotheses. The algorithms we present update the weights according to a variant of the weighted majority algorithm of Littlestone and Warmuth [LW89] called WMG. We view WMG as a node that is connected to each of its inputs by a weighted edge. The inputs are in the interval $[0, 1]$. An input x of weight w votes with xw for 1 and $(1 - x)w$ for 0. The node "combines" the votes of the inputs by determining the total weight q_0 (respectively q_1) placed on 0 (respectively 1) and predicts with the bit corresponding to the larger of the two totals (and for the sake of discreteness with 1 in case of a tie). After receiving feedback of what the prediction should have been, then for each input the fraction of the weight placed on the wrong bit is multiplied by β, where $\beta \in [0, 1)$. Thus the weight w of an input x becomes $(1 - x + x\beta)w$ if the feedback is 0 and $((1 - x)\beta + x)w$ if the feedback is 1. If $\beta = 0$ then the total weight halves in each trial in which the node makes a mistake and we obtain an analysis like that of the halving algorithm.

---

[1] The adversary, who tries to maximize the learner's mistakes, knows the learner's algorithm and has unlimited computing power.

[2] Throughout this paper we let lg denote the base 2 logarithm and ln the natural logarithm.

The remainder of this paper is organized as follows. In Section 2 we present our two different constructions for applying WMG to the problem of learning a binary relation. In Section 3 we define an important generalization of the problem of learning binary relations with noisy data and show how the robust nature of WMG can be exploited to handle such noise. In Section 4 we present our main result. Namely, we provide a technique that enables us to prove an upper bound on the number of mistakes made by our polynomial-time algorithm to learn binary relations even when noise is present. Finally, in Section 5 we end with some concluding remarks.

2 Our Constructions For Applying WMG

In this section we describe two different methods for applying WMG to the problem of learning a binary relation. The first construction uses one node and k^n weights. Thus the number of weights is still exponential but significantly lower than the number of weights of the halving algorithm ($2^{km}k^n$). For this case, the analysis is straightforward and for the noise-free case the bounds achieved are essentially optimal with respect to the known information-theoretic lower bound. The purpose of this construction is to show what is possible when computational resources are cheap. In the second construction we use one node for each of the n rows and one weight for each pair of rows (i.e. $\binom{n}{2}$ weights). Proving bounds for the second construction is much more involved and is the focus of the paper. The bounds obtained are non-optimal when compared to those obtained when computation time is not a concern. However, our bounds are significantly better than previous bounds obtained by a polynomial algorithm.

In the first construction we use one weight per partition of the n rows into at most k row types (i.e. k^n weights). Initially all weights are set to 1. To make a prediction for a new matrix entry, each partition (with non-zero weight) votes as follows: If a column of the row type to which the new entry belongs has already been set then vote with the value of this bit, otherwise, vote with 1/2 causing the weight to be split between the votes of 0 and 1. Our algorithm predicts according to the weighted majority of these votes (see Figure 2.1). Recall that after receiving the feedback WMG multiplies the fractions of the weights that were placed on the wrong bit by β. By selecting $\beta = 0$, the weight of partitions that predict incorrectly (and are thus inconsistent with the partial matrix) are set to zero and the weights of all partitions that split their vote are halved. After all entries of the target matrix are known, the correct partition has weight at least 2^{-km} since it never predicted incorrectly, and split its vote at most km times. Since the initial weight is k^n, we obtain the mistake bound of $km + n \lg k$ just as for the halving algorithm. (Actually, one can show that when $\beta = 0$ then the first construction simulates the halving algorithm with k^n weights instead of $2^{km}k^n$ weights). Note that the mistake bound of this algorithm is essentially optimal since Goldman et al. [GRS93] prove an information-theoretic lower bound of $km + (n-k)\lfloor \lg k \rfloor$ mistakes. While this construction has assumed that an upper bound on k is known, if no such bound is provided the standard doubling trick can be applied.

In the second construction one weighted majority node is used per row of the matrix, and one edge between each pair of nodes. Thus, unlike the first construction, no knowledge of k is needed. Let $e_{rr'}$ (and $e_{r'r}$) denote the (undirected) edge between the node for row r and the node for row r', and let $w(e)$ to denote the weight of edge e. The node for row r dictates the predictions for all entries in row r. So the nodes, in some sense, partition the learning problem amongst themselves. Assume M_{rj} is the next value to predict. To make

Learn-Relation(γ)
For all r, r' such that $(r \neq r')$ initialize $w(e_{rr'}) = 1$
In each trial do the following four parts:
1. Receive a matrix entry M_{rj} for prediction
2. Produce a prediction as follows
   For each row $r' \neq r$ such that $w(e_{rr'}) > 0$
      If $M_{r'j}$ is not known then row r' predicts that $M_{rj} = 1/2$
      If $M_{r'j} = 1$ then row r' predicts that $M_{rj} = 1$
      If $M_{r'j} = 0$ then row r predicts that $M_{rj} = 0$
   Let R_0 contain rows that predict $M_{rj} = 0$
   Let R_1 contain the algorithms that predict $M_{rj} = 1$
   If $\sum_{r' \in R_1} w(e_{rr'}) \geq \sum_{r' \in R_0} w(e_{rr'})$ predict 1
   Else predict 0
3. Receive correct value for M_{rj}
4. If the prediction of the algorithm was wrong then update the weights as follows
   For each $r' \neq r$ such that $w(e_{rr'}) > 0$
      If row r' made a correct prediction then let $w(e_{rr'}) \leftarrow (2 - \gamma) w(e_{rr'})$
      Else if row r' predicted incorrectly then let $w(e_{rr'}) \leftarrow \gamma w(e_{rr'})$

Figure 2.3: Our polynomial prediction algorithm for learning binary relations.

3 A Generalization: Non-Pure Relations

A key contribution of this paper is showing how the robust nature of WMG enables us to solve an important generalization of the problem of learning binary relations with noisy data. To motivate this problem we briefly review the allergist example given by Goldman et al. [GRS93]. Consider an allergist with a set of patients to be tested for a given set of allergens. Each patient is either *highly allergic*, *mildly allergic*, or *not allergic* to any given allergen. The allergist may use either an *epicutaneous* (scratch) test in which the patient is given a fairly low dose of the allergen, or an *intradermal* (under the skin) test in which the patient is given a larger dose of the allergen. What options does the allergist have in testing a patient for a given allergen? He/she could just perform the intradermal test (option 0). Another option (option 1) is to perform an epicutaneous test, and if it is not conclusive, then perform an intradermal test. Which option is best? If the patient has no allergy or a mild allergy to the given allergen, then option 0 is best, since the patient need not return for the second test. However, if the patient is highly allergic to the given allergen, then option 1 is best, since the patient does not experience a bad reaction. The allergist's goal here is to minimize the number of prediction mistakes in choosing the option to test each patient for each allergen. Although Goldman et al. explore several possible methods for the selection of the presentation order, here we only consider the standard worst-case model in which an adversary determines the order in which the patient/allergen pairs are presented.

While this example is generally convincing, there is an assumption that is a clear oversimplification. Namely, they assume that there are a common set of "allergy types" that occur often and that most people fit into one of these allergy types. Thus the allergy types

become the row types of the matrix. However, while it is true that often people have very similar allergies, there are not really pure allergy types. In other words, it is unreasonable to assume that all rows of the same "type" are identical but rather they are just close to each other. Without this flexibility one may be required to have most patient's allergies correspond to a distinct allergy type. Henceforth, we shall refer to original formulation of the problem of learning binary relations in which all row types are "pure" as *learning pure relations*.

We propose the following generalization of this problem. For any set column c of bits and let $\mathcal{N}_0(c)$ be the number of zeros in c. Likewise, let $\mathcal{N}_1(c)$ be the number of ones in c. Suppose that the rows of the matrix are partitioned into a set of k clusters $p = \{S^1, \ldots, S^k\}$. Let S^i_j denote the jth column of the cluster (or submatrix) S^i. For each cluster we define a distance measure

$$d(S^i) = \sum_{j=1}^{m} min\{\mathcal{N}_0(S^i_j), \mathcal{N}_1(S^i_j)\}$$

In other words, think of defining a *center* point for partition S^i by letting the value of column j in this center be the majority vote of the entries in S^i_j. Then $d(S^i)$ is just the sum over all rows s in S^i of the Hamming distance between s and this center point.

For the whole partition p we define the *noise* α_p as $\sum_{S^i \in p} d(S^i)$, and the *size* k_p as the number of clusters in partition p. We refer to this problem as *learning non-pure relations*. Due to the robust nature of WMG, we can use both constructions to learn non-pure relations by simply using a non-zero update factor β.

We now discuss both constructions when applied to the problem of learning non-pure relations and give bounds for each. The key to our approach is to view minor discrepancies between the row templates and the actual rows as noise. This greatly reduces the mistake bounds that one can obtain when using the original formulation of Goldman et al. [GRS93] by reducing the number of row types. The robust nature of the weighted majority algorithm enables us to handle noise.

To demonstrate our basic approach, we now show that our first construction (i.e. the one using k^n weights) can learn a non-pure relation by making at most

$$\min\left\{k_p m + \alpha_p + \frac{n \ln k + \alpha_p \ln \frac{1}{\beta}}{\ln \frac{2}{1+\beta}}\right\} \tag{3.1}$$

mistakes in the worst case, where the minimum is taken over all partitions p of size at most k and k_p denotes the size and α_p the noise of partition p.

In the noisy case the first construction still uses a single copy of the weighted majority algorithm with one weight for each of the k^n partitions. Assume M_{rj} is the next value to predict. Then a particular partition predicts with the majority of all already set entries from column j whose rows have the same type as row r in the partition. In case of a tie the partition predicts with $1/2$. A partition with weight w and prediction x votes with xw for 1 and $(1-x)w$ for 0. The Algorithm WMG totals the votes for 0 and for 1 and predicts with the bit of the larger total (with 1 in case of a tie).

When a partition predicts incorrectly, its weight is multiplied by β. A partition that splits its vote has its weight multiplied by $(1+\beta)/2$. (Half of its weight remains unchanged and the other half is multiplied by β.) We claim that a partition p predicts incorrectly at most α_p times and splits its vote at most $k_p m + \alpha_p$ times. To see this consider the case

when the matrix consists of one column and we have just one row type. If α is the number of occurences of the minority bit in the column then the number of wrong predictions is at most α and the number of ties at most $\alpha + 1$. Now the arbitrary partition case follows by summing over all columns and row types.

From the above it follows that the final weight in the system is at least $\beta^{\alpha_p}\left(\frac{1+\beta}{2}\right)^{k_p m+\alpha_p}$. Since the initial weight in the system is k^n and for each trial in which a mistake occurs the total weight after the trial is at most $(1+\beta)/2$ times the total weight before the trial, we get the following inequality for the total number of mistake μ:

$$k^n\left(\frac{1+\beta}{2}\right)^{\mu} \geq \beta^{\alpha_p}\left(\frac{1+\beta}{2}\right)^{k_p m+\alpha_p}.$$

Solving for μ gives the above bound (3.1).

Finally, by applying the results of Cesa-Bianchi et al. [CBFH$^+$93]) we can tune β as a function of an upper bound α on the noise.

Lemma 1: *[CBFH$^+$93] For any real value $z \geq 0$*

$$\frac{1 + z\ln\frac{1}{g(z)}}{2\ln\frac{2}{1+g(z)}} \leq z + \sqrt{z} + \frac{1}{2\ln 2},$$

where $g(z) = 1 - 2\frac{\sqrt{1+z}-1}{z}$ and $g(0) = 0$.

Theorem 1: *For any positive integers k and α, the first construction with $\beta = g(\frac{\alpha}{n\ln k})$ makes at most*

$$\min\left\{k_p m + 3\alpha_p + 2\sqrt{\alpha n\ln k} + n\lg k\right\}$$

mistakes, where the minimum is taken over all partitions p whose size k_p is at most k and whose noise α_p is at most α.

Proof: Since $\alpha_p \leq \alpha$ and the function $\left(\ln\frac{1}{\beta}\right) / \left(\ln\frac{2}{1+\beta}\right)$ is decreasing over the range $0 \leq \beta < 1$ and approaches 2 as $\beta \to 1$, it follows that

$$\begin{aligned}
\frac{n\ln k + \alpha_p\ln\frac{1}{\beta}}{\ln\frac{2}{1+\beta}} &= \frac{n\ln k}{\ln\frac{2}{1+\beta}} + 2\alpha_p + \alpha_p\left(\frac{\ln\frac{1}{\beta}}{\ln\frac{2}{1+\beta}} - 2\right) \\
&\leq \frac{n\ln k}{\ln\frac{2}{1+\beta}} + 2\alpha_p + \alpha\left(\frac{\ln\frac{1}{\beta}}{\ln\frac{2}{1+\beta}} - 2\right) \\
&= 2n\ln k\left(\frac{1 + \frac{\alpha}{n\ln k}\ln\frac{1}{\beta}}{2\ln\frac{2}{1+\beta}}\right) + 2\alpha_p - 2\alpha.
\end{aligned}$$

So by applying Lemma 1 with $z = \frac{\alpha}{n\ln k}$ and $\beta = g(z)$ we obtain a worst-case mistake bound[4] of

$$\min\left\{k_p m + \alpha_p + 2\left(\alpha_p + \sqrt{\alpha n\ln k} + \frac{n\lg k}{2}\right)\right\} = \min\left\{k_p m + 3\alpha_p + 2\sqrt{\alpha n\ln k} + n\lg k\right\}$$

for our first construction, where the minimum is taken over all partitions p of size at most k with noise at most α and k_p denotes the size and α_p the noise of partition p. ■

[4] We also can get rid of the factor 2 in the first formulation of the bound by either letting the algorithm predict probabilistically in $\{0,1\}$ or deterministically in the interval $[0,1]$ [CBFH$^+$93, KW93].

For the above tuning we needed an upper bound for both the size and the noise of the partition. If an upper bound for only one the two is known, then a standard doubling trick can be used to guess the other. This causes only a slight increase in the mistake bound (See Cesa-Bianchi et al. [CBFH+93].) Note that in the above mistake bound there is a subtle tradeoff between the noise α_p and size k_p of a partition p.

We recall that when this first construction is applied in the noise-free case that it essentially matches the information-theoretic lower bound. An interesting question is whether or not it can be shown to be essentially optimal in the case of learning non-pure relations.

As we show in the next section (Theorem 3), when using the second construction for learning non-pure relations, we show that our algorithm makes at most

$$min\left\{k_p m + \sqrt{3mn^2 \lg k + 4\alpha_p mn\left(1-\frac{\alpha_p}{mn}\right) + mn\sqrt{24\alpha n\left(1-\frac{\alpha}{mn}\right)\ln k}} : p \in P\right\}$$

mistakes in the worst case, where the minimum is taken over all partitions p and k_p denotes the size and α_p the noise of partition p where $k_p \leq k$ and $\alpha_p \leq \alpha$.

4 Construction Two: A Polynomial-time Algorithm

In this section we discuss the algorithm *Learn-Relation* (see Figure 2.3) obtained by our second construction that uses one weighted majority node per row. The mistake bound of Theorem 2 obtained for this algorithm is larger than the mistake bound of the first construction. However this algorithm uses only $\binom{n}{2}$ weights as opposed to exponentially many.

We begin by giving an update that is equivalent to the one used in WMG [LW89]. Recall that in WMG if x is the prediction of an input with weight w, then if the feedback is the bit ρ then w is multiplied by $1-(1-\beta)|x-\rho|$ for $\beta \in [0,1)$. If $\beta = \gamma/(2-\gamma)$, then for our application the update of WMG can be summarized as follows

- If a node predicts correctly (so, $|x-\rho| = 0$) its weight is not changed.
- If a node makes a prediction of $1/2$ then its weight is multiplied by $1/(2-\gamma)$.
- If a node predicts incorrectly (so, $|x-\rho| = 1$) then its weight is multiplied by $\gamma/(2-\gamma)$.

In the new update all factors in the above construction are simply multiplied by $(2-\gamma)$. This update is used in our Algorithm *Learn-Relation(γ)* (see Figure 2.3) since it leads to simpler proofs. Because voting is performed by a weighted majority vote, the predictions made by the two schemes are identical. In order to use the analysis technique of Littlestone and Warmuth we must obtain a lower bound for the final weight in the system. However, using WMG the weight in the system is decreased by nodes that do not predict, and thus we would have to compute an upper bound on the total number of times that this occurs. Thus to simplify the analysis, we have modified the update scheme (i.e. at each step we multiplied all weights by $(2-\gamma)$) so that the weights of nodes that do not predict remain unchanged.

4.1 The Analysis

We now compute an upper bound on the number of mistakes made by *Learn-Relation*.

We begin with some preliminaries. A function $f : \Re \rightarrow \Re$ is concave (respectively convex) over an interval D of $\Re$ if for all $x \in D$, $f_{xx}(x) \leq 0$ ($f_{xx}(x) \geq 0$). In our analysis we repeatedly use the following variants of Jensen's inequality. Let f be a function from $\Re$ to $\Re$ that is concave over some interval D of $\Re$. Let $q \in \mathcal{N}$, and let $x_1, x_2, \ldots, x_q \in D$. Then

$$\sum_{i=1}^{q} x_i = U \;\Rightarrow\; \sum_{i=1}^{q} f(x_i) \leq q f(U/q).$$

Furthermore, if f is monotonically increasing over the interval D then the following holds:

$$\sum_{i=1}^{q} x_i \leq U \;\Rightarrow\; \sum_{i=1}^{q} f(x_i) \leq q f(U/q).$$

Likewise, let f be a function from $\Re$ to $\Re$ that is convex over some interval D of $\Re$. Let $q \in \mathcal{N}$, and let $x_1, x_2, \ldots, x_q \in D$. Then

$$\sum_{i=1}^{q} x_i = U \;\Rightarrow\; \sum_{i=1}^{q} f(x_i) \geq q f(U/q).$$

We also use Jensen's inequality when applied to a function over two variables. A function $f : \Re \times \Re \rightarrow \Re$ is concave over an interval $D_x \times D_y$ of $\Re \times \Re$ if for all $x \in D_x$ and $y \in D_y$, $f_{xx} \leq 0$, $f_{yy} \leq 0$, and $f_{xx} f_{yy} - (f_{xy})^2 \geq 0$. Let f be a function from $\Re \times \Re$ to $\Re$ that is concave over some interval $D_x \times D_y$ of $\Re \times \Re$. Let $q \in \mathcal{N}$, and let $x_1, x_2, \ldots, x_q \in D_x$ and $y_1, y_2, \ldots, y_q \in D_y$. Then

$$\sum_{i=1}^{q} x_i = U_x \text{ and } \sum_{i=1}^{q} y_i = U_y \;\Rightarrow\; \sum_{i=1}^{q} f(x_i, y_i) \leq q f(U_x/q, U_y/q).$$

We now give an overview of the proof of our main result along with several key lemmas that are used in the proof. Let p be *any* partition of size k_p and noise α_p. Let μ to denote the total number of mistakes made by the learner, and let μ_i will denote the number of mistakes that occur when the learner is predicting an entry in a row of cluster i. (Thus, $\sum_{i=1}^{k_p} \mu_i = \mu$.) Let n_i be the number of rows in cluster i. (So $n = \sum_{i=1}^{k_p} n_i$.) Let $\mathcal{A}$ be all $\binom{n}{2}$ edges and the set $\mathcal{E}$ contain all edges connecting two rows of the same cluster. We further decompose $\mathcal{E}$ into $\mathcal{E}_1, \ldots, \mathcal{E}_{k_p}$ where $\mathcal{E}_i$ contains all edges connecting two rows in the same cluster i of p. Observe that $|\mathcal{E}_i| = \frac{n_i(n_i-1)}{2}$. When making an erroneous prediction for M_{rj}, we define the *force* of the mistake to be the number of rows in the same cluster as row r for which column j was known when the mistake occurred. Let F_i be the sum of the forces of all mistakes made when predicting an entry in a row of cluster i.

Recall that the noise of a partition p is defined as

$$\alpha_p = \sum_{S^i \in p} d(S^i) = \sum_{S^i \in p} \min\{\mathcal{N}_0(S^i_j), \mathcal{N}_1(S^i_j)\}.$$

For each cluster i and column j, let $\delta_{i,j} = \min\{\mathcal{N}_0(S^i_j), \mathcal{N}_1(S^i_j)\}$, and let $\delta_i = \sum_{j=1}^{m} \delta_{i,j}$. Thus observe that $\alpha_p = \sum_{i=1}^{k_p} \delta_i$.

We now define J_i to be the number of times that a weight in $\mathcal{E}_i$ is multiplied by γ when making a prediction for an entry in cluster i. That is, J_i is the total number of times, over all trials in the learning session in which a mistake occurs, where an entry in cluster i incorrectly predicts the value of an entry in cluster i (voting with all its weight).

We now give the key lemma used in our main proof. For ease of exposition, let $a = \lg(2-\gamma) = \lg \frac{2}{1+\beta}$ and $b = \lg\left(\frac{2-\gamma}{\gamma}\right) = \lg \frac{1}{\beta}$.

Lemma 2: *For each* $1 \leq i \leq k_p$,

$$F_i = \frac{b}{a} J_i + \frac{1}{a} \sum_{e \in \mathcal{E}_i} \lg w(e).$$

Proof: We begin by noting that, when $\alpha_p = 0$, then $J_i = 0$ and the number of times some weight in $\mathcal{E}_i$ is multiplied by $(2-\gamma)$ equals F_i. Thus, in the noise-free case, it follows that $(2-\gamma)^{F_i} = \prod_{e \in \mathcal{E}_i} w(e)$. When $\alpha_p > 0$, then F_i is the number of times some weight in $\mathcal{E}_i$ is multiplied by either $(2-\gamma)$ or γ. Since the number of times some weight in $\mathcal{E}_i$ is multiplied by γ is J_i we have that

$$(2-\gamma)^{F_i - J_i}\, \gamma^{J_i} = (2-\gamma)^{F_i} \left(\frac{\gamma}{2-\gamma}\right)^{J_i} = \prod_{e \in \mathcal{E}_i} w(e).$$

Taking logarithms of both sides we obtain the stated result. ∎

Note that if $n_i = 1$ then $J_i = |\mathcal{E}_i| = F_i = 0$. The proof of our main theorem uses Lemma 2 as its starting point. We first obtain (Lemma 4) a lower bound for F_i that depends on the total number of mistakes, μ_i, made by our algorithm when making a prediction for an entry in cluster i. Next we must determine the maximum amount by which the "noisy" entries of the submatrix S^i cause the weights in $\mathcal{E}_i$ to be "weakened" (i.e. multiplied by γ) instead of being "strengthened" (i.e. multiplied by $(2-\gamma)$) as desired. In Lemma 5 we show how to J_i can be upper bounded in terms of δ_i, the noise within cluster i. Finally in Lemma 7 we obtain an upper bound for the sum of the logarithms of the weights. We do this by observing that the total weight in the system never increases and then use the convexity of the logarithm function. The proof of our main theorem essentially combines all the lemmas and uses and addition convexity argument for combining the contributions from all clusters.

We now obtain a lower bound for F_i. In order to obtain this lower bound, it is crucial to first obtain an upper bound on the number of mistakes for a given cluster and given force. This quantity characterizes the rate at which the weighted-majority nodes are gaining information.

Lemma 3: *For each row type* r *and force* f *there are at most* m *mistakes of force* f.

Proof: We use a proof by contradiction. Suppose that for cluster i the learner makes $m+1$ force f mistakes. Then there must be two mistakes that occur for the same column. Suppose the first of these mistakes occurs when predicting M_{rj} and the second occurs when predicting $M_{r'j}$ where both rows r and r' are in cluster i. However, after making a force f mistake when predicting M_{rj} that entry is known and thus the force of the $M_{r'j}$ mistake must be at least $f+1$ giving the desired contradiction. ∎

We now compute a lower bound for the force of the mistakes made when predicting entries in cluster i.

Lemma 4: *For any partition* p *and for any* $1 \leq i \leq k_p$,

$$F_i \geq \max\left\{\mu_i - m, \frac{\mu_i^2}{2m} - \frac{\mu_i}{2}\right\}.$$

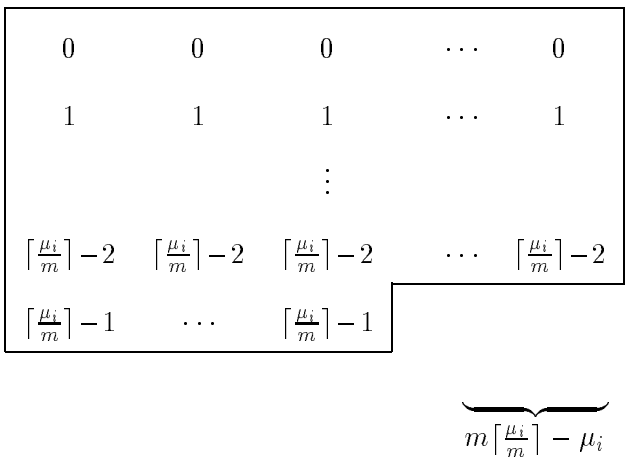

Figure 4.1: First μ_i elements of the sequence $\langle 0\rangle^m\langle 1\rangle^m\langle 2\rangle^m \ldots$.

Proof: We proceed by showing that both expressions above are lower bounds for F_i. Let σ_i denote the sum of the first μ_i elements of the sequence $\langle 0\rangle^m\langle 1\rangle^m\langle 2\rangle^m \cdots$. From Lemma 3 it follows that $F_i \geq \sigma_i$. Thus, clearly our first lower bound

$$F_i \geq \mu_i - m$$

follows since all but m mistakes have force at least one.

We now compute a more sophisticated lower bound on σ_i. Let $s(x) = \sum_{k=1}^{x} k = \frac{x(x+1)}{2}$. Using the structure illustrated in Figure 4.1 it is easily seen that

$$\begin{aligned} F_i &\geq m\, s\left(\left\lceil\frac{\mu_i}{m}\right\rceil - 1\right) - \left(m\left\lceil\frac{\mu_i}{m}\right\rceil - \mu_i\right)\left(\left\lceil\frac{\mu_i}{m}\right\rceil - 1\right) \\ &= \left(\left\lceil\frac{\mu_i}{m}\right\rceil - 1\right)\left(\mu_i - \frac{m}{2}\left\lceil\frac{\mu_i}{m}\right\rceil\right) \\ &\geq \left(\frac{\mu_i}{m} - 1\right)\left(\mu_i - \frac{m}{2}\frac{\mu_i}{m}\right) \\ &= \frac{\mu_i^2}{2m} - \frac{\mu_i}{2}. \end{aligned}$$

To see that the last inequality holds, first observe that if μ_i is a multiple of m then we have equality. Finally, it is easily seen that when $\mu_i = qm + r$ for quotient $q \geq 0$ and remainder $0 \leq r \leq m-1$ that $(\lceil\frac{\mu_i}{m}\rceil - 1)\,(\mu_i - \frac{m}{2}\lceil\frac{\mu_i}{m}\rceil) > \frac{\mu_i^2}{2m} - \frac{\mu_i}{2}$. This completes the proof of the lemma. ■

Observe that the simple linear bound is a better lower bound only for $m < \mu_i < 2m$.

Next, we capture the relationship between J_i and the noise within cluster i of the partition to obtain an upper bound for J_i.

Lemma 5: *For any partition p, and for $1 \leq i \leq k_p$,*

$$J_i \leq \delta_i n_i - \frac{\delta_i^2}{m}.$$

Proof: For ease of exposition, we assume that for each cluster i and column j, the majority of the entries in S_j^i, are 1. Thus $\delta_{i,j}$ is exactly the number of 0's in S_j^i. Observe that for every known 0 entry in S_j^i, the quantity J_i is increased by one whenever the learner makes a prediction error when predicting the value of an entry in S_j^i that is a 1. Thus, in the worst case, each of the $\delta_{i,j}$ entries in S_j^i that are 0 could cause J_i to be incremented for each of the $n_i - \delta_{i,j}$ entries in S_j^i that are 1. Thus,

$$J_i \leq \sum_{j=1}^{m} \delta_{i,j}(n_i - \delta_{i,j}) = \delta_i n_i - \sum_{j=1}^{m} \delta_{i,j}^2.$$

Since x^2 is convex, it follows that $\sum_{j=1}^{m} \delta_{i,j}^2 \geq \frac{\delta_i^2}{m}$. This completes the proof of the lemma. ■

Next we obtain an upper bound on the sum of the logarithms of the weights of a set of edges from $\mathcal{A}$. A key observation used to prove this upper bound is that the overall weight in the system never increases. Therefore, since the initial weight in the system is $n(n-1)/2$ we obtain the following lemma.

Lemma 6: *Throughout the learning session for any $\mathcal{A}' \subseteq \mathcal{A}$,*

$$\sum_{e \in \mathcal{A}'} w(e) \leq \frac{n(n-1)}{2}.$$

Proof: In trials where no mistake occurs the total weight of all edges $\sum_{e \in \mathcal{A}} w(e)$ clearly does not increase. Assume that a mistake occurs in the current trial. Ignore all weights that are not updated. Of the remaining total weight W that participates in the update let c be the fraction that was placed on the correct bit and $1-c$ be the fraction placed on the incorrect bit. The weight placed on the correct bit is multiplied by $2-\gamma$ and the weight placed on the incorrect bit by γ. Thus the total weight of all edges that participated in the update is $(c(2-\gamma) + (1-c)\gamma)W$ at the end of the trial. Since the latter is increasing in c and $c \leq 1/2$ whenever a mistake occurs, we have that the total of all weights updated in the current trial is at most $(\frac{2-\gamma}{2} + \frac{\gamma}{2})W = W$ at the end of the trial. We conclude that the total weight of all edges also does not increase in trials where a mistakes occurs. Finally since $\mathcal{A}' \subseteq \mathcal{A}$, the result follows. ■

Lemma 7: *Throughout the learning session for any $\mathcal{A}' \subseteq \mathcal{A}$,*

$$\sum_{e \in \mathcal{A}'} \lg w(e) \leq |\mathcal{A}'| \lg \frac{n(n-1)}{2|\mathcal{A}'|}.$$

Proof: This result immediately follows from Lemma 6 and the concavity of the log function. ■

We are now ready to prove our main result.

Theorem 2: *For all $\beta \in [0,1)$, Algorithm Learn-Relation when using the parameter $\gamma = \frac{2\beta}{1+\beta}$ makes at most*

$$\min\left\{ k_p m + \min\left\{ \frac{\frac{n^2}{2e}\lg e + \alpha_p(n - \frac{\alpha_p}{km})\lg\frac{1}{\beta}}{\lg\frac{2}{1+\beta}}, \sqrt{\frac{3mn^2 \lg k_p + 2\alpha_p(mn - \alpha_p)\lg\frac{1}{\beta}}{\lg\frac{2}{1+\beta}}} \right\} \right\}$$

mistakes in learning a binary-relation where the outside minimum is taken over all partitions p and k_p denotes the size and α_p the noise of partition p.

Proof: Let p be *any* partition of size k_p and noise α_p. We begin by noting that for any cluster i and column j in partition p, $\delta_{i,j} \leq n_i/2$ and thus $\delta_i = \sum_{j=1}^{m} \delta_{i,j} \leq n_i m/2$. Recall that in Lemma 4 we showed that $F_i \geq \mu_i - m$. Observe that if $n_i = 1$ then $F_i = 0$, and thus it follows that $\mu_i \leq m$. Thus, without loss of generality, we will assume throughout the remainder of this section that $n_i \geq 2$ for all i.

As we have discussed, the base of our proof is provided by Lemma 2. We then apply Lemmas 4, 5 and 7 to obtain an upper bound on the number of mistakes made by *Learn-Relation*.

We now proceed independently with the two lower bounds for F_i given in Lemma 4. Applying Lemma 2 with the first lower bound for F_i given in Lemma 4, summing over the clusters in p, and solving for μ yields,

$$\mu = \sum_{i=1}^{k_p} \mu_i \leq k_p m + \frac{b}{a} \sum_{i=1}^{k_p} J_i + \frac{1}{a} \sum_{e \in \mathcal{E}} \lg w(e). \tag{4.1}$$

From Lemma 5 we know that $J_i \leq \delta_i n_i - \frac{\delta_i^2}{m} \leq \delta_i n - \frac{\delta_i^2}{m}$. It is easily verified that the function $n\delta_i - \frac{\delta_i^2}{m}$ is concave. Thus, combining Jensen's inequality with the fact that $\sum_{i=1}^{k_p} \delta_i = \alpha_p$, we obtain:

$$\sum_{i=1}^{k_p} J_i \leq n\alpha_p - \frac{\alpha_p^2}{k_p m} = \alpha_p \left(n - \frac{\alpha_p}{k_p m} \right). \tag{4.2}$$

In addition, by applying Lemma 7 with $\mathcal{A}' = \mathcal{E}$ we obtain that:

$$\sum_{e \in \mathcal{E}} \lg w(e) \leq |\mathcal{E}| \lg \frac{n(n-1)}{2|\mathcal{E}|}.$$

Next observe that the function $x \lg \frac{n(n-1)}{2x}$ is concave and obtains its maximum value at $x = \frac{n(n-1)}{2e}$. Thus we obtain that

$$|\mathcal{E}| \lg \frac{n(n-1)}{2|\mathcal{E}|} \leq n(n-1)\frac{\lg e}{2e}. \tag{4.3}$$

Finally by combining Inequalities (4.1), (4.2), and (4.3) we obtain that:

$$\mu \leq k_p m + \frac{b}{a}\alpha_p \left(n - \frac{\alpha_p}{k_p m} \right) + \frac{n(n-1)}{2a}\frac{\lg e}{e} \leq k_p m + \frac{b}{a}\alpha_p \left(n - \frac{\alpha_p}{k_p m} \right) + \frac{n^2}{2a}\frac{\lg e}{e}$$

proving our first bound on μ.

We now proceed by combining Lemma 2 with the more sophisticated second lower bound for F_i given in Lemma 4 to obtain:

$$\frac{\mu_i^2}{2m} - \frac{\mu_i}{2} \leq \frac{b}{a} J_i + \frac{1}{a} \sum_{e \in \mathcal{E}_i} \lg w(e). \tag{4.4}$$

Next we apply Lemma 7 with $\mathcal{A}' = \mathcal{E}_i$ to obtain:

$$\sum_{e \in \mathcal{E}_i} \lg w(e) \leq |\mathcal{E}_i| \lg \frac{n(n-1)}{2|\mathcal{E}_i|}.$$

Applying this above inequality with Inequality (4.4) yields

$$\begin{aligned}\mu_i &\leq \frac{m}{2} + \sqrt{\frac{2bm}{a} J_i + \frac{2m}{a} \sum_{e \in \mathcal{E}_i} \lg w(e) + \frac{m^2}{4}} \\ &\leq m + \sqrt{\frac{2bm}{a}} \sqrt{J_i + \frac{1}{b}|\mathcal{E}_i| \lg \frac{n(n-1)}{2|\mathcal{E}_i|}}.\end{aligned}$$

Next we apply Lemma 5 and the fact that $|\mathcal{E}_i| = n_i(n_i-1)/2 \leq n_i^2/2$, and then sum over the k_p clusters in p to obtain:

$$\mu \leq k_p m + \sqrt{\frac{2bm}{a}} \sum_{i=1}^{k_p} \sqrt{\delta_i n_i - \frac{\delta_i^2}{m} + \frac{n_i^2}{2b} \lg \frac{n(n-1)}{n_i(n_i-1)}}.$$

As shown in the appendix, the function $f(\delta_i, n_i) = \sqrt{\delta_i n_i - \frac{\delta_i^2}{m} + \frac{n_i^2}{2b} \lg \frac{n(n-1)}{n_i(n_i-1)}}$ is concave for $n_i \geq 2$ and $\delta_i \leq n_i m/2$. Since $\sum_{i=1}^{k_p} n_i = n$ and $\sum_{i=1}^{k_p} \delta_i = \alpha_p$, we can thus apply Jensen's inequality to obtain:

$$\begin{aligned}\mu &\leq k_p m + k_p \sqrt{\frac{2bm}{a}} \sqrt{\frac{\alpha_p n}{k_p^2} - \frac{\alpha_p^2}{k_p^2 m} + \frac{n^2}{2k_p^2 b} \lg \frac{k_p^2(n-1)}{n-k_p}} \\ &= k_p m + \sqrt{\frac{2bm}{a} \alpha_p \left(n - \frac{\alpha_p}{m}\right) + \frac{n^2 m}{a} \lg \frac{k_p^2(n-1)}{n-k_p}} \qquad (4.5)\end{aligned}$$

Observe that $n \geq k_p$, and furthermore if $n = k_p$ then at most nm mistakes can occur and the upper bound of the theorem trivially holds. Thus without limiting the applicability of our result we can assume that $n \geq k_p + 1$ which in turn implies that $\frac{n-1}{n-k_p} \leq k_p$. Thus we can further simplify Inequality (4.5) to obtain:

$$\begin{aligned}\mu &\leq k_p m + \sqrt{\frac{3}{a} m n^2 \lg k_p + \frac{2bm}{a} \alpha_p (n - \frac{\alpha_p}{m})} \\ &= k_p m + \sqrt{\frac{3mn^2 \lg k_p + 2\alpha_p(mn - \alpha_p) \lg \frac{1}{\beta}}{\lg \frac{2}{1+\beta}}}\end{aligned}$$

thus giving us our second bound on μ.

The above analysis was performed for any partition $p \in P$. Thus taking the minimum over all partitions in P we get the desired result. ■

As when applying the weighted majority algorithm to a noise-free setting, notice that we obtain the best performance by selecting $\gamma = 0$ (respectively $\beta = 0$). Thus we obtain the following corollary for the case of learning pure relations.

Corollary 1: *For the case of learning pure relations where $\alpha_p = \alpha = 0$, the algorithm Learn-Relation with $\gamma = 0$ learns a pure k-binary-relation making at most*

$$km + \min\left\{\frac{n^2}{2e} \lg e, n\sqrt{3m \lg k}\right\}$$

mistakes in the worst-case.

We now apply the results of Cesa-Bianchi et al. [CBFH+93] to tune β for the more general case in which the relation is not pure.

Theorem 3: *For any positive integers k and α, Algorithm Learn-Relation when using $\gamma = \frac{2\beta}{1+\beta}$, where $\beta = g(z)$ and $z = \frac{2\alpha\left(1-\frac{\alpha}{mn}\right)}{3n\ln k}$, makes at most*

$$k_p m + \sqrt{3mn^2 \lg k + 4\alpha_p mn\left(1 - \frac{\alpha_p}{mn}\right) + mn\sqrt{24\alpha n\left(1 - \frac{\alpha}{mn}\right)\ln k}}$$

mistakes, where the minimum is taken over all partitions p whose size k_p is at most k and whose noise α_p is at most α.

Proof: From Theorem 2 we know that for all $\beta \in [0,1)$, our algorithm makes at most $\min\left\{k_p m + \sqrt{\frac{3mn^2 \lg k_p + 2\alpha_p(mn-\alpha_p)\lg\frac{1}{\beta}}{\lg\frac{2}{1+\beta}}}\right\}$ mistakes where the minimum is taken over all partitions p and k_p denotes the size and α_p the noise of partition p.

Assume that the partition p has the property that $k_p \le k$ and $\alpha_p \le \alpha$. Observe that the function $\left(\ln\frac{1}{\beta}\right) / \left(\ln\frac{2}{1+\beta}\right)$ is decreasing over the range $0 \le \beta < 1$ and approaches 2 as $\beta \to 1$. Furthermore, since $2\alpha_p(mn-\alpha_p) \le 2\alpha(mn-\alpha)$ for $\alpha_p \le \alpha \le \frac{mn}{2}$, it follows that

$$\begin{aligned}
&\frac{3mn^2 \lg k + 2\alpha_p(mn-\alpha_p)\lg\frac{1}{\beta}}{\lg\frac{2}{1+\beta}} \\
&\le \quad \frac{3mn^2 \ln k}{\ln\frac{2}{1+\beta}} + 4\alpha_p(mn-\alpha_p) + 2\alpha(mn-\alpha)\left(\frac{\ln\frac{1}{\beta}}{\ln\frac{2}{1+\beta}} - 2\right) \\
&= \quad 6mn^2 \ln k\left(\frac{1 + \frac{2\alpha\left(1-\frac{\alpha}{mn}\right)\ln\frac{1}{\beta}}{3n\ln k}}{2\ln\frac{2}{1+\beta}}\right) - 4\alpha(mn-\alpha) + 4\alpha_p(mn-\alpha_p).
\end{aligned}$$

So by applying Lemma 1 with $z = \frac{2\alpha\left(1-\frac{\alpha}{mn}\right)}{3n\ln k}$ and $\beta = g(z)$ we obtain a worst-case mistake bound[5] given in the theorem. ■

So for example if the optimal partition p is such that $\alpha_p = n$, then the number of mistakes is at most

$$k_p m + \sqrt{3mn^2 \lg k + 4mn^2 + mn^2\sqrt{24\ln k}}.$$

In addition to presenting their algorithm to make at most $km + n\sqrt{(k-1)m}$ mistakes, Goldman et al. [GRS93] present an information-theoretic lower bound for a class of algorithms that they call row-filter algorithms. They say that an algorithm A is a *row-filter algorithm* if A makes its prediction for M_{rj} strictly as a function of j and all entries in the set of rows consistent with row r and defined in column j. For this class of algorithms they show a lower bound to $\Omega(n\sqrt{m})$ for $m \ge n$ on the number of mistakes that any algorithm must make. Recently, William Chen [Che92] has extended their proof to obtain a lower bound of $\Omega(n\sqrt{m \lg k})$ for $m \ge n \lg k$. Observe that *Learn-Relation* is *not* a row-filter algorithm since the weights stored on the edges between the rows allows it to use the outcome of previous predictions to aid in its prediction for the current trial. Nevertheless, a simple modification of the projective geometry lower bound of Goldman et al. [GRS93] can be

[5] Again, we can get rid of the factor 2 in this mistake bound by either letting the algorithm predict probabilistically in $\{0,1\}$ or deterministically in the interval $[0,1]$ [CBFH+93, KW93].

used to show an $\Omega(n\sqrt{m})$ lower bound for $m \geq n$ on the number of prediction mistakes by our algorithm. Chen's extension of the projective geometry argument to incorporate k does not extend in such a straightforward manner, however, we conjecture that his lower bound can be generalized to prove that the mistake-bound we obtained for *Learn-Relation* is asymptotically tight. Thus to obtain a better algorithm, more than pairwise information between rows may be needed in making predictions.

5 Concluding Remarks

We have demonstrated that a weighted majority voting algorithm can be used to learn a binary relation even when there is noise present. Our first construction uses exponentially many weights. In the noise-free case this construction is essentially optimal. We believe that by proving lower bounds for the noisy case (possibly using the techniques developed in [CBFH+93]) one can show that the tuned version of the first contruction (Theorem 1) is close to optimal in the more general case as well.

The focus of our paper is the analysis of our second construction that uses a polynomial number of weights and thus can make predictions in polynomial time. In this construction a number of copies of our algorithm divide the problem amongst themselves and learn the relation cooperatively.

It is surprising that the parallel application of on-line algorithms using multiplicative weight updates can be used to do some non-trivial clustering with provable performance (Theorem 3). Are there other applications where the clustering capability can be exploited? For the problem of learning binary relations the mistake bound of the polynomial algorithm (second construction) which uses $\binom{n}{2}$ weights is still far away from the mistake bound of the exponential algorithm (first construction) which uses k^n weights. There seems to be a tradeoff between efficiency (number of weights) and the quality of the mistake bound. One of the most fascinating open problem regarding this research is the following: Is it possible to significantly improve our mistake bound (for either learning pure or non-pure relations) by using say $O(n^3)$ weights? Or can one prove, based on some reasonable complexity theoretic or cryptographic assumptions, that no polynomial-time algorithm can perform significantly better than our second construction.

Acknowledgements

We thank William Chen and David Helmbold for pointing out flaws in earlier versions of this paper. We also thank the anonymous referees for their comments.

References

[Ang88] Dana Angluin. Queries and concept learning. *Machine Learning*, 2(4):319–342, 1988.

[BF72] J. Barzdin and R. Freivald. On the prediction of general recursive functions. *Soviet Mathematics Doklady*, 13:1224–1228, 1972.

[CBFH+93] Nicolò Cesa-Bianchi, Yoav Freund, David P. Helmbold, David Haussler, Robert E. Schapire, and Manfred K. Warmuth. How to use expert advice. In *Proceedings of the Twenty Fifth Annual ACM Symposium on Theory of Computing*, pages 382–391, May 1993.

[Che92] William Chen, 1992. Personal communication.

[GRS93] Sally A. Goldman, Ronald L. Rivest, and Robert E. Schapire. Learning binary relations and total orders. *SIAM Journal of Computing*, 22:1006–1034, October 1993.

[KW93] Jyrki Kivinen and Manfred K. Warmuth. *Using Experts for Predicting Continuous Outcomes*. Eurocolt 1993.

[Lit88] Nick Littlestone. Learning when irrelevant attributes abound: A new linear-threshold algorithm. *Machine Learning*, 2:285–318, 1988.

[Lit89] Nicholas Littlestone. *Mistake Bounds and Logarithmic Linear-threshold Learning algorithms*. PhD thesis, U. C. Santa Cruz, March 1989.

[LLW91] Nicholas Littlestone, Philip M. Long, and Manfred K. Warmuth. On-line learning of linear functions. In *Proceedings of the Twenty Third Annual ACM Symposium on Theory of Computing*, pages 465–475, May 1991. To appear in *Journal of Computational Complexity*.

[LW89] Nick Littlestone and Manfred K. Warmuth. The weighted majority algorithm. In *30th Annual Symposium on Foundations of Computer Science*, pages 256–261, October 1989. To appear in *Information and Computation*.

[Vov90] Volodimir G. Vovk. Aggregating strategies. In *Proceedings of the Third Annual Workshop on Computational Learning Theory*, pages 371–383. Morgan Kaufmann, August 1990.

Appendix

We now demonstrate that the function $f(\delta_i, n_i) = \sqrt{\delta_i n_i - \frac{\delta_i^2}{m} + \frac{n_i^2}{2b} \lg \frac{n(n-1)}{n_i(n_i-1)}}$ is concave for $n_i \geq 2$ and $\delta_i \leq n_i m/2$.

For ease of exposition we shall let $x = \delta_i$ and $y = n_i$. We must now show that $f_{xx} \leq 0$, $f_{yy} \leq 0$, and $f_{xx} f_{yy} - (f_{xy})^2 \geq 0$.

It is easily verified that:

$$f_{xx} = \frac{-y^2}{(f(x,y))^3} \left(\frac{1}{4} + \frac{1}{2mb} \lg \frac{n(n-1)}{y(y-1)} \right) \leq 0.$$

It can also be verified that f_{yy} can be expressed such that the denominator of f_{yy} is

$$16b^2 (\ln 2)^2 (y-1)^2 (f(x,y))^3,$$

and the numerator is

$$\begin{aligned} f_{yy} \quad = \quad & -4y^3(y-1) - y^2 - 4b^2x^2(\ln 2)^2(y-1)^2 - \\ & 4bx \ln 2 \left(y(4y^2 - 7y + 2) - \frac{x}{m}(6y^2 - 10y + 3) \right) - \\ & \ln \frac{n(n-1)}{y(y-1)} \left(2y^2(2y^2 - 4y + 1) + \frac{8b \ln 2}{m} x^2 (y-1)^2 \right). \end{aligned}$$

It is easily shown that $2y^2 - 4y + 1 \geq 0$ for $y \geq 2$. Observe that $y(4y^2 - 7y + 2) - \frac{x}{m}(6y^2 - 10y + 3) \geq 0$ when

$$x \leq \frac{y}{m} \left(\frac{4y^2 - 7y + 2}{6y^2 - 10y + 3} \right).$$

Furthermore, for $y \geq 2$

$$\frac{4y^2 - 7y + 2}{6y^2 - 10y + 3} \geq \frac{4}{7} > \frac{1}{2}$$

and thus it suffices to have $x \leq ym/2$ which is the case.

Finally, it can be verified that $f_{xx} f_{yy} - (f_{xy})^2$ is

$$\frac{y^2 \left(4y(y-1) + 1 + bm \ln 2 + 2bmy \ln 2 (y-2) + 2 \ln \frac{n(n-1)}{y(y-1)} (2y^2 - 4y + 1) \right)}{16b^2 (\ln 2)^2 (y-1)^2 m (f(x,y))^4} \geq 0$$

for $y \geq 2$.

This completes the proof that $f(x,y)$ is concave over the desired interval.