IIIS 2014 Spring: ATCS - Selected Topics in Optimization
Lecture date: Mar 5, 2013
Instructor: Jian Li **Scribe: Wei Cao**

1 Chernoff Bounds [1]

In probability theory, the Chernoff bound, gives exponentially decreasing bounds on tail distributions of **sums of independent random variables**. It is a sharper bound than the known first or second moment based tail bounds such as Markov's inequality or Chebyshev inequality, which only yields power-law bounds on tail decay. However, the Chernoff bound requires that the variates are independent, a condition that neither Markov nor Chebyshev inequalities requires.

Let $X_1, ..., X_n$ be n independent random variables in $[0, 1]$. Let $X = \sum_i X_i$ and $\mu = \mathbb{E}[X] = \sum_i \mathbb{E}[X_i]$. Then, the following inequalities hold:

1. $\Pr[X > (1+\delta)\mu] \leq e^{-\frac{\delta^2 \mu}{3}}, \delta \in (0, 1)$
2. $\Pr[X < (1-\delta)\mu] \leq e^{-\frac{\delta^2 \mu}{2}}, \delta \in (0, 1)$
3. $\Pr[X > (1+\delta)\mu] \leq e^{-\frac{(1+\delta)ln(1+\delta)}{4}\mu}, \delta > 1$

2 Independent Rounding

2.1 Coin Flipping

Let $X_i = 1$ if the i_{th} flip is head with probability $\frac{1}{2}$, $X = \sum_i X_i$. From the Chernoff bound, $\mu = \frac{n}{2}$,

$$\Pr[X \geq \mu + \lambda] = \Pr[X \geq \mu(1 + \frac{\lambda}{\mu})] \leq e^{-(\frac{\lambda}{\mu})^2 \frac{\mu}{3}} = e^{-\frac{\lambda^2}{3\mu}}$$

If $\lambda = O(\sqrt{n})$, we get that $\Pr[X \geq \mu(1 + \frac{\lambda}{\mu})] \leq e^{-O(1)}$. If $\lambda = O(\sqrt{n \log n})$, we get that $\Pr[X \geq \mu(1 + \frac{\lambda}{\mu})] \leq \frac{1}{n^C}$ for some constant C.

Hence, w.h.p. (with high probability), $X \in \mu \pm O(\sqrt{n \log n})$.

2.2 Discrepancy

Let U be a universe of n elements, $S_1, ..., S_n$ be the subsets of U. Each element is colored with $+1$ or -1. For a subset $S \subseteq U$, define $\mathsf{Disc}(S) = \sum_{v \in S} color(v)$. We want to find a way to color all the elements such that the discrepancy, defined as

$$\mathsf{Disc}(color) = \max_e \ |\mathsf{Disc}(e)|,$$

is minimized.

Again we try to utilize the random coloring method. In this case, the colors are not in $[0, 1]$, thus we can not use Chernoff bound directly. However, a simple linear transformation works as follows.

Consider any specific subset S. Let X_i denote the color of i_{th} vertex, applying a transformation $Y_i = (X_i + 1)/2$. We can easily see that $Y_i \in [0, 1]$ and we can use Chernoff bound now: It is straightforward to see that $\sum_{i \in S} Y_i \in \frac{|S|}{2} \pm O(\sqrt{n \log n})$ w.h.p. Then, w.h.p.,

$$\mathsf{Disc}(S) = \sum_{i \in S} X_i = 2 \sum_{i \in S} Y_i - |S| = 2(\frac{|S|}{2}) \pm O(\sqrt{n \log n}) - |S| = \pm O(\sqrt{n \log n}).$$

Therefore $\mathsf{Disc}(\text{random color}) = O(\sqrt{n \log n})$ w.h.p. (we use the union bound here over all subsets).

Remark: In fact, the best bound for the discrepancy problem is $O(\sqrt{n})$. Spencer proved that a coloring with discrepancy $O(\sqrt{n})$ always exists using a clever entropy argument. However, his proof does not give a polynomial time algorithm. Recently, Bansal employed the semi-definite programming method and a Brownian motion type of randomized rounding method and obtained a polynomial time algorithm for getting a coloring with discrepancy $O(\sqrt{n})$. Both papers are very beautiful and I highly recommend you to read them.

2.3 Congestion Minimization

Let $G = (V, E)$ be an undirected graph. We are given a set of vertex pairs $D = (s_i, t_i)_{i=1...n}$, and $P_i = \{\text{a collection of paths from } s_i \to t_i\}$. Let the congestion on edge e, denoted by $\mathsf{cong}(e)$, be the number of paths containing e. Our goal is to pick one path from each P_i such that

$$\max_e \mathsf{cong}(e)$$

is minimized.

This problem is NP-hard. Again, we will be using independent rounding method to give an approximation algorithm.

First we express this problem as an integer program. Let $f_p^i = 1$ denotes the event that if the i_{th} pair uses the path p. Let c denote the congestion of the graph. Then the problem is equivalent with :

$$\begin{array}{lll} & \min c & \\ s.t. & \sum_{p \in P_i} f_p^i = 1 & \forall i \\ & \sum_i \sum_{e \in p} f_p^i \le c & \forall e \\ & f_p^i \in \{0, 1\} & \forall i, p \end{array}$$

As usual, we relax the third constraint to $f_p^i \in [0, 1]$ to obtain a linear programming relaxation. Then, we use standard LP solver to get a fractional solution. Suppose the fractional solution is f_i^p. Now, we obtain an integral solution using randomized rounding, as follows. For each pair i, we do the following independently:

- Since $\sum_{p \in P_i} f_p^i = 1$, we can interpret $\{f_p^i\}_{p \in P_i}$ as a probability distribution over all paths in P_i. Then, we pick exactly one path $p \in P_i$ with corresponding probability f_p^i [1].

---

[1] It is very important to notice that this is different from rounding each path $p \in P_i$ independently (with probability f_p^i, we choose this path).

The algorithm is very simple. Now, we analyze the congestion induced by the above algorithm. Consider a particular edge, say e. Let $X_i^e = 1$ if the path picked in P_i use e. Let $X^e = \sum_i X_i^e$ be the congestion of edge e. Then,

$$\mathbb{E}[X^e] = \mathbb{E}[\sum_i X_i^e] = \sum_i \mathbb{E}[X_i^e] = \sum_i \sum_{p \in P_i, e \in p} f_p^i \leq c$$

If c in LP is greater or equal than 1, substitute it into Chernoff bound,

$$\Pr[X^e \geq kc] \leq e^{\frac{-k \log k \cdot c}{4}}.$$

Letting $k = O(\frac{\log n}{\log \log n})$, we can see $\Pr[X^e \geq kc] \leq \frac{1}{n^{const}}$. Using union bound over all edges, we can see the algorithm is a k-approximation w.h.p.

Otherwise if $c < 1$, $\Pr[X^e \geq kc]$ may not be smaller than $\frac{1}{n^{const}}$. However, notice that in the any non-trivial instance, we have $OPT \geq 1$. Hence, we only need to show $X^e \geq k$ w.h.p. in order to get a k-approximation. Indeed,

$$\Pr[X^e \geq k] = \Pr[X^e \geq c \cdot \frac{k}{c}] \leq e^{\frac{-c \cdot \frac{k}{c} \log \frac{k}{c}}{4}} \leq e^{\frac{-k \log k}{4}}$$

Therefore we still have a $O(\frac{\log n}{\log \log n})$-approximation.

3 Dependent Rounding[3]

In the congestion minimization problem, what if we change "choose one path from each set P_i" into "choose k different paths of each set P_i" (called *multi-path routing problem*)? Let us try to do the same thing as before. Now, the first LP constraint should be $\sum_{p \in P_i} f_p^i = k, \forall i$. The fractional solution $\{f_p^i\}_{p \in P_i}$ can not be interpreted as a probability distribution any more (since the sum is not one). For example, what is $\{f_p^i\}_{p \in P_i} = \{0.5, 0.4, 0.2, 0.9\}$ and we need to choose exactly two different paths? What should we do here? If we round each path $p \in P_i$ independently (with probability f_p^i, we choose this path), we may not have exactly k path chosen from P_i (for some P_i we choose more and some less).

Therefore, we need to introduce some new rounding procedure, the dependent rounding method. We first consider a general case of dependent rounding.

3.1 On Bipartite Graph

Suppose we are given a bipartite graph (A, B, E) with bipartition (A, B). We are also given a value $x_{i,j} \in [0, 1]$ for each edge $(i, j) \in E$. We provide a randomized polynomial-time scheme that rounds each $x_{i,j}$ to a random variable $X_{i,j} \in \{0, 1\}$ such that the following properties hold:

P1. Marginal distribution. For every edge (i, j), $\Pr[X_{i,j} = 1] = x_{i,j}$

P2. Degree-preservation. Consider any vertex $i \in A \cup B$. Define its fractional degree d_i to be $\sum_{j:(i,j) \in E} x_{i,j}$, and integral degree D_i to be the random variable $\sum_{j:(i,j) \in E} X_{i,j}$. Then we have $D_i \in [\lfloor d_i \rfloor, \lceil d_i \rceil]$.

P3. Negative correlation. If $f = (i, j)$ is an edge, let X_f denote $X_{i,j}$. For any vertex i and any subset S of the edges incident on i:

$$\forall b \in \{0, 1\}, \Pr[\bigwedge f \in S(X_f = b)] \leq \prod_{f \in S} \Pr[X_f = b]$$

A few remarks:

1. As before, we would like to interpret $x_{i,j}$ values as probabilities. This is exactly what P1 says.
2. In this multi-path routing problem, we want the cardinality of rounded-up elements to be exactly some value. Such cardinality constraints can be seen in many places. P2 can help us to achieve this.
3. P3 is very useful in many probabilistic analysis, especially when we want to show some concentration result. If the random variables are negative correlated, all versions of Chernoff bound we have covered before for independent random variables still hold (You should read the proof of Chernoff bound to see why the proof carries over to negatively correlated random variables).

Now, we describe how to achieve the above properties. Initially, let $y_{i,j} = x_{i,j}$ for all edges. The rounding algorithm will modify $y_{i,j}$ iteratively such that $\forall i, j, y_{i,j} \in \{0, 1\}$ at the end. The iteration will satisfy two invariants:

1. For all $(i, j) \in E, y_{i,j} \in [0, 1]$
2. Once $y_{i,j}$ rounds to $0/1$, it never changes.

The algorithm is described in **Algorithm 1:**

Algorithm 1: Dependent Rounding on Bipartite Graph

Initially, let $F = E$;
while $F \neq \emptyset$ **do**
- Find a simply cycle P (if there is no cycle, find a maximal path P) in the subgraph (A, B, F);
- Color the edges of P alternately with *black* and *white*;
- Let $M_1 \leftarrow$ all *black* edges, $M_2 \leftarrow$ all *white* edges;
- $\alpha = \min\{\epsilon > 0 : ((\exists (i, j) \in M_1 : y_{i,j} + \epsilon = 1) \bigvee (\exists (i, j) \in M_2 : y_{i,j} - \epsilon = 0)\}$;
- $\beta = \min\{\epsilon > 0 : ((\exists (i, j) \in M_1 : y_{i,j} - \epsilon = 0) \bigvee (\exists (i, j) \in M_2 : y_{i,j} + \epsilon = 1)\}$;
- With probability $\beta/(\alpha + \beta)$, set $y_{i,j} = y_{i,j} + \alpha$ for all $(i, j) \in M_1$ and $y_{i,j} = y_{i,j} - \alpha$ for all $(i, j) \in M_2$; with probability $\alpha/(\alpha + \beta)$, set $y_{i,j} = y_{i,j} + \beta$ for all $(i, j) \in M_1$ and $y_{i,j} = y_{i,j} + \beta$ for all $(i, j) \in M_2$;
- Remove all $(i, j) \in P$ satisfy $y_{i,j} = \{0, 1\}$ from F and let $X_{i,j} = y_{i,j}$

Return X;

It is easy to verify **marginal distribution** and **degree-preservation** holds in the algorithm. See [3] for the proof of **negative correlation** part.

An illustration is shown in Figure 1.

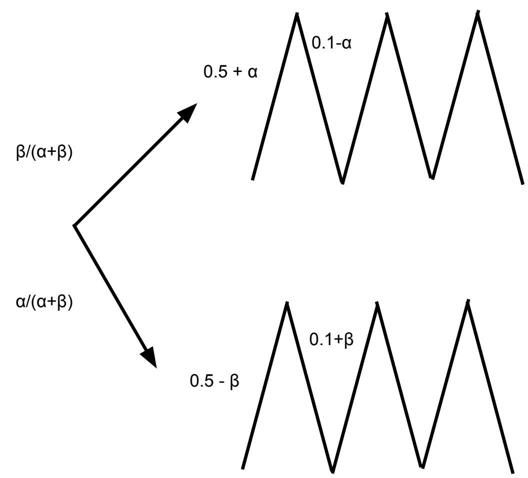


Figure 1: Example of dependent rounding

3.2 Low congestion multi-path routing

We now come back to the *low congestion multi-path routing* problem. We could construct a special bipartite graph for this problem.

Assume we are supposed to pick exactly k paths from each set P_i. And we are given a sequence $(x_1, ..., x_t)$ of t real numbers such that each $x_i \in [0, 1]$ and $\sum_i x_i = k$.

First add a special node "u", and for each x_i, add an edge (u, i) with value x_i. Obviously it is a bipartite graph with $|A| = 1$ and $|B| = t$ and contains no cycle. It is not hard to verify this case satisfy negative correlation. The first two edges are special case of the maximal path described in **Algorithm 1**, denoting as e_i, e_j. Thus, if $x_i + x_j > 1$, with probability $x_j/(x_i + x_j)$ set $x_i := 1, x_j := x_j - (1 - x_i)$ and with probability $x_i/(x_i + x_j)$ set $x_i := x_i - (1 - x_j), x_j = 1$. Similarly we could process the case when $x_i + x_j < 1$. See Figure 2 for illustration.

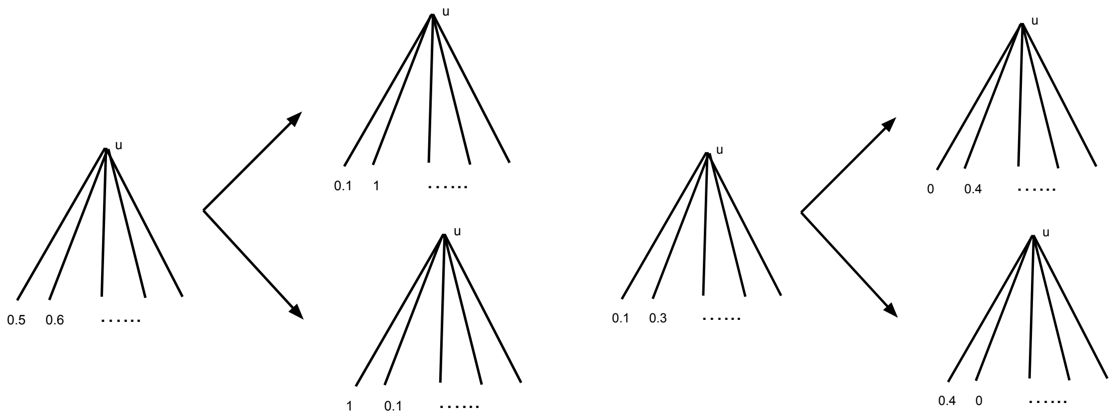
3.3 Throughput Maximization for Broadcast Scheduling

Definition 3.1 *There is a set of pages $P = \{1, 2, ..., n\}$ that can be broadcast by a broadcast server. Assume that time is discrete; for an integer t, the time-slot t is the window of time $(t-1, t]$. At each time slot, the broadcast server could broadcast exactly* one *page and all of the users could receive that page. There are several users, each user query for page p in a certain time interval. Once the user receives the page p from the broadcast server in corresponding time interval, the query will be satisfied.*

Design an algorithm to schedule the broadcast server in order to satisfy user queries as many as possible.

This problem has been proven to be NP-hard. We give an approximation algorithm using dependent rounding.

First we formulate it into an IP problem. Define $Y_p^t = 1$ if the broadcast server broadcast page p at time slot t, $X_{p,i} = 1$ if the i_{th} query for page p is satisfied and $\Gamma_{i,p}$ is the time slot interval of



(a) $x_i + x_j > 1$

(b) $x_i + x_j < 1$

Figure 2: Illustration for congestion problem

the i_{th} query for page p. Then the problem can be described as follows:

$$\begin{array}{lll} & \text{maximize} \quad \sum_{p,i} X_{p,i} & \\ s.t & \sum_t Y_p^t \geq X_{p,i} & \forall t \in \Gamma_{i,p} \\ & \sum_p Y_p^t = 1 & \forall t \\ & X_{p,i} \in \{0,1\} & \forall t, i \\ & Y_p^t \in \{0,1\} & \forall t, p \end{array}$$

We first show the approximation ratio by independent rounding.

Relax IP to LP, a trivial lower bound of fractional solution is

$$\Pr[query_{i,p} \text{ is satisfied}] = 1 - \prod_{t \in \Gamma_{i,p}} (1 - Y_p^t) \geq (1 - \frac{1}{e}) X_{p,i}$$

To prove this lower bound, assume $|\Gamma_{i,p}| = T$, then

$$\begin{aligned} 1 - \prod_{t \in \Gamma_{i,p}} (1 - Y_p^t) &\geq 1 - (1 - \frac{X_{p,i}}{T})^T \\ &= 1 - (1 - \frac{1}{T/X_{p,i}})^{T/X_{p,i} \cdot X_{p,i}} \\ &\geq 1 - (\frac{1}{e})^{X_{p,i}} \geq (1 - \frac{1}{e}) X_{p,i} \end{aligned}$$

The last inequality is because of the following: Since $X_{p,i} \in [0,1]$, let $g(X_{p,i}) = 1 - (\frac{1}{e})^{X_{p,i}} - (1 - \frac{1}{e}) X_{p,i}$. Observe that $g(0) = g(1) = 0$. Take the derivative of $g(X_{p,i})$, we find that g is always nonegtive in $[0,1]$ (increases monotonically from 0, reaches its maximum and then decrease down to 0, but never gets below 0). Therefore we have $1 - (\frac{1}{e})^{X_{p,i}} \geq (1 - \frac{1}{e}) X_{p,i}$.

Utilizing **random shifting** we could improve the lower bound to $\frac{3}{4}$.

First construct a bipartite graph $G = (U, V, E)$ as follows. $u_t \in U$ represents time slots t. For each page p, we will group all the time slots that p is broadcast fractionally into some number m_p

of windows and add m_p vertices in $v_p^1, ..., v_p^{m_p}$. Choose a random variable z uniformly in $[0, 1]$, then the weight of edges can be derived by such an algorithm(Algorithm 2).

Algorithm 2: Construct the Bipartite Graph for A Particular p

Initially, let $pu = 1, pv = 1, sum = 0$;
forall the Y_p^t **do**
    if $pv = 1$ **then**
        $upper = z$;
    else
        $upper = 1$;
    if $sum + Y_p^t \geq upper$ **then**
        add edge $(u_{pu}, v_{pv}, upper - Y_p^t)$;
        add edge $(u_{pu}, v_{pv+1}, Y_p^t - (upper - Y_p^t))$;
        $pv \leftarrow pv + 1$;
        $sum \leftarrow Y_p^t - (upper - Y_p^t)$;
    else
        add edge (u_{pu}, v_{pv}, Y_p^t);
        $sum \leftarrow sum + Y_p^t$
    $pu \leftarrow pu + 1$;

An instance of particular p is illustrated in Figure 3, in which $Y_p^t = [0.3, 0.3, 0.5, 0.8, 0.1, 0.3, 0.5]$.

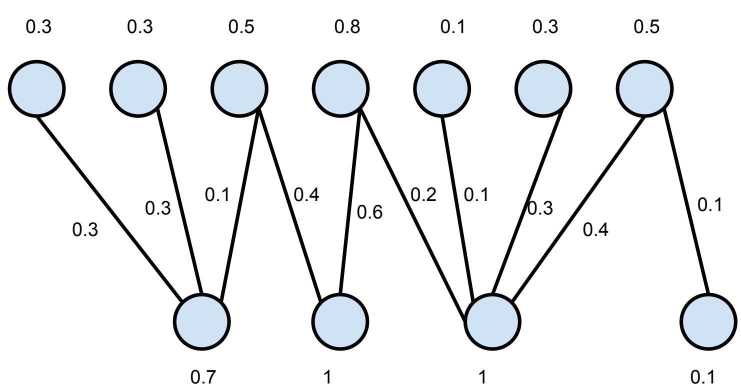


Figure 3: Construction of bipartite graph of particular p

Use dependent rounding in the bipartite graph. If an edge (u_t, v_p^i) is rounded to 1, then we broadcast page p at time t.

Since $\forall t, \sum_p Y_p^t = 1$, according to **Degree-preservation** property, at each time at most one page will be broadcast. Thus our algorithm return an available solution.

Now consider a certain query $q = (p, t)$, if $X_{p,i} = 1$, in the worst case, the fractional solution will span two adjacent windows. These two windows together fractionally provide one unit of page p.

Let random variable a denote the fraction of p provided by first windows. Actually, a is a uniform random variable in $[0, 1]$ which is "random shifting" by z we picked at first. From **Algorithm 2**,

$$\Pr[q \text{ is satisfied}] = \int_0^1 \max\{a, 1-a\} da = \frac{3}{4}$$

If $X_{p,i} < 1$, besides the above case, the time interval of query may lie in a certain window, therefore the query will be satisfied if and only if we round the edge of this window into the time interval. Thus,

$$\Pr[q \text{ is satisfied}] = \int_0^{X_{p,i}} \max\{a, X_{p,i} - a\} da + \int_{X_{p,i}}^1 X_{p,i} db \geq \frac{3}{4} X_{p,i}$$

4 More Geometry About LP

4.1 Standard Form

Every LP

$$Ax \leq b \tag{1}$$

could be rewritten as "standard form"

$$\begin{cases} Ax & = & b \\ x & \geq & 0 \end{cases} \tag{2}$$

where form (2) indicating an intersection of a hyper plane and nonnegative cone. Assume that there are m constrains in $Ax = b$ and n variables in (2). For any vertex solution of the LP, we have

$$\#zero \geq n - m,$$

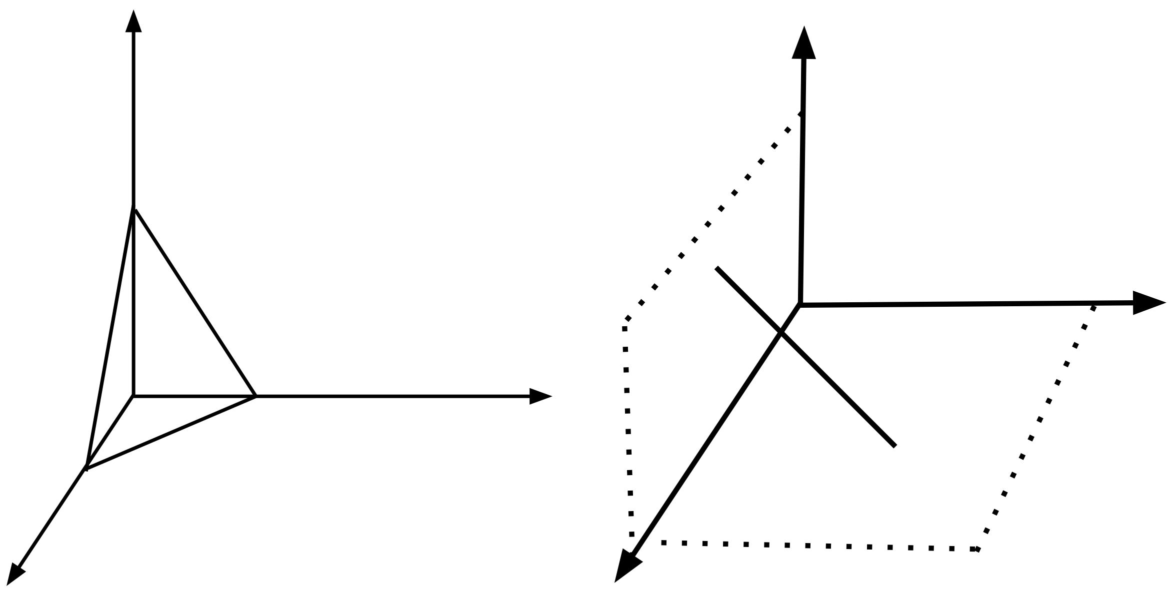
where $\#zero$ is the number of zero coordinates in the vertex solution. We do not prove this rigorously here. See Figure 4 as an intuitive example in $\mathbb{R}^3$.

References

[1] Chernoff Bound (Wikipedia) : `http://en.wikipedia.org/wiki/Chernoff_bound`

[2] Random Algorithms : `http://www.cs.cmu.edu/afs/cs/academic/class/15859-f04/www/scribes/lec9.pdf`

[3] Gandhi R, Khuller S, Parthasarathy S, et al. Dependent rounding and its applications to approximation algorithms[J]. Journal of the ACM (JACM), 2006, 53(3): 324-360. Random Algorithms : `http://www.cs.cmu.edu/afs/cs/academic/class/15859-f04/www/scribes/lec9.pdf`



(a) n=3, m=1, $\#zero = 2$

(b) n=3, m=2, $\#zero = 1$

Figure 4: An example in $\mathbb{R}^{\nvDash}$.