ANALYSIS OF DECENTRALIZED ALGORITHM FOR ONLINE SECRETARY PROBLEM

ABSTRACT

This is an abstract.

Keywords: Secretary Problem, Online Algorithm, Decentralized Algorithm, Competitive Analysis

Contents

Chapter 1 Introduction

Variants of online secretary problem have be

emphTODO: Need revision

TODO: connection and distinction with online matching

Consider a complete weighted bipartite graph $G = (U, V, w), w : U \times V \to R^+$. $U = \{u_1, u_2, \ldots, u_m\}$ and $V = \{v_1, v_2, \ldots, v_n\}$ ($m \leq n$) are commonly known as firms and applicants in the market, respectively. To avoid ties, we assume that no two edges have the same weight.

In an online setting, all applicants arrive one by one in a random order. Each firm decides whether to send an offer immediately or not, and then each applicant can choose an offer from what she has got. And the edge weight $w(u, v)$ is considered to be the benefit u and v will get if the applicant v accepts the offer from the firm u, in other words, $w(u, v)$ is both the value of u for v and the value of v for u. Note that each firm can hire at most one applicant as its secretary. All decisions can not be revoked. Firms and applicants can only see weights of edges which are incident to them. Our goal is to design decentralized algorithms for each firm such that (i) the resulting overall social welfare is nearly optimal and (ii) each firm by adopting the proposed algorithm can get the nearly optimal applicant. By "decentralized" it means there is no supervisor doing the assignments, and each firm runs its own algorithm independently with no communication.

Chapter 2 Preliminaries

2.1 Graph Matching

2.1.1 Bipartite Matching

A graph is called *bipartite* if its vertex set can be partitioned into two disjoint set U and V such that every edge connects a vertex in U with a vertex in V. Such a graph is often written as $G = (U, V, E)$ where U and V are two disjoint vertex set and E is the edge set.

A *matching* in a graph $G = (U, V, E)$ is a set of edges M such that no two edges shared a common vertex. And a matching M is called *maximal* if for every edge $e \in E \setminus M$ it satisfies that e shares some common vertices with some edges from E. Normally we are interested in the *maximum* matching, i.e. a matching containing the largest possible number of edges. And the size of the maximum matching is called the *matching number* of this graph. Figure **??** shows an example of a maximum matching in a bipartite graph. Note that the problem of finding a maximum matching can be solved in polynomial time by *Hungarian method*.

People show their great interests in finding the maximum matching since there are deep connections between the matching number and many other interesting properties in a given graph. For example, the *König's theorem* proved by Dénes Kőnig in 1931 states that, in bipartite graph, it is equivalent to find a maximum matching that to find a minimum set cover. Independently in the same year by Jenő Egerváry the same result was shown in the more general case of weighted bipartite graphs. Thus a lot of problems was shown to be NP-hard in general graphs have a polynomial time algorithm in bipartite graphs (e.g. minimum set cover).

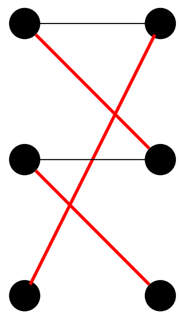

Figure 2.1 The red edges forms a maximum matching in graph

	potato	orange	banana
alice	**5**	3	1
bob	2	4	**7**
celine	3	**6**	4

Figure 2.2 An example of maximum weighted bipartite matching

2.1.2 Weighted Bipartite Matching

To generalize the maximum matching problem we could assign weights to those edges. And the goal is then changed to find a matching with the largest/smallest possible weight. This gives us the *maximum/minimum weighted bipartite matching problem*.

Formally speaking, given a weighted complete bipartite graph $G = (U, V, w)$ where U and V are two disjoint vertex sets and the edge set $E = U \times V$. w is the weight function maps every edge in E to its own weight in $\mathbb{R}$. We have to find a matching M such that its weight $w(M) \triangleq \sum_{e \in M} w(e)$ is maximized/minimized.

This maximum weighted bipartite matching problem characterized a lot of problems in our daily lives. For example we are selling some items (e.g. potato, orange and banana) and the customers (e.g. alice, bob and celine) provides their prices on each of these items. Every customer wants only one item and a item could only be sold to one of the customers. The problem is how are we going to sell these items so that we can earn the most money. Figure **??** shows an example of such a problem, and it is clear that we should sell potato to alice, orange to celine and banana to bob so that we can earn $5 + 6 + 7 = 18$ in total.

To solve the maximum/minimum weighted bipartite matching problem, a polynomial-time algorithm was proposed by Harold Kuhn (1955), who gives the name "*Hungarian method*" as this algorithm was largely based on the works done by two Hungarian mathematicians Dénes Kőnig and Jenő Egerváry.

2.2 Online Algorithms

But most of time in our real lives, our information about the graph is usually incomplete. And we have to make our decisions before the whole graph is shown to us.

For example in figure **??**, when Alice comes and wants to buy one item, we can not let her wait until Bob and Celine provide their prices. We have to make our decision right now on which one to sell before Alice gets angry. This scenario raises a new sort of problems in an online fashion – the problems we have to output our answers before the whole input was shown to us.

2.2.1 Secretary Problem

One of the most classical problem is the *online secretary problem*.

Suppose you are hiring a secretary for your firm, and there are n applicants who come to apply for this job. You have to interview these applicants one by one in a random order until you choose one of them as the secretary. After an interview you have to make your decision whether to offer her this secretarial position. Once the decision is made, it can not be revoked. Each applicant has a score on how good they can handle this job and of course you want the best one. The difficulty is, before a applicant is interviewed you can not know her score. How can you make a decision so that the probability of choosing the best applicant is maximized?

People have found out that the best strategy for this problem is *stopping rule*:

Step 1: interview the first $r-1$ applicants and reject them, set t be the best score among them.

Step 2: choose the first subsequent applicant who has a score better than t.

For example, suppose that 6 applicants are waiting for the interview and their score is $\{3, 4, 2, 5, 1, 6\}$ (in the coming order). We choose the secretary using stopping rule with $r = 3$. First we interview the first 2 applicants and reject them, record the best score as $t = 4$. When the third applicant comes in we found out that her score is no better than t, hence we reject her. Then the fourth applicant comes in, luckily she has a score 5 greater than t, so we offer her the secretarial position and terminate the protocol. Unfortuately we failed to pick the best one in the sixth place, but a score of 5 is not so bad to our real needs.

Now what I'm going to show that if the coming order of applicants is uniformly at random, then the probability to get the best applicant using stopping rule is high.

With a parameter r, the probability of choosing the best applicant can be easily calculated:

$$
\begin{aligned}
P(r) &= \sum_{i=1}^{n} \Pr(i\text{-th applicant is the best and chosen}) \\
&= \sum_{i=1}^{n} \Pr(i\text{-th applicant is chosen}|i\text{-th applicant is the best}) \times \frac{1}{n} \\
&= \sum_{i=r}^{n} \Pr(\text{no one is chosen before } i|i\text{-th applicant is the best}) \times \frac{1}{n} \\
&= \sum_{i=r}^{n} \frac{r-1}{i-1} \times \frac{1}{n}
\end{aligned}
$$

Note that in the previous equations, the event that no one is chosen before i means the second best among the first i applicant appears before r. Therefore $\Pr(\text{no one is chosen before } i|i\text{-th applicant is the best}) = \frac{r-1}{i-1}$.

When n goes to infinity, $P(r)$ is approximately the integral $\frac{r}{n}\int_{\frac{r}{n}}^{1} \frac{1}{x}\mathrm{d}x = \frac{r}{n}\ln(\frac{n}{r})$.

In this problem we can choose $r \approx \frac{n}{e}$ which maximize the integral above $P(\frac{n}{e}) \approx \frac{1}{e} \approx 0.368$.

2.2.2 Online Matching

Another perspective of this paper's work originates from online bipartite matching. Given a weighted bipartite graph $G = (U, V, w)$, but at first you don't know any information about this graph. Each time one vertex in V is revealed and you can see the weights of all edges incident to it. Then you have to decide which vertex in U should be matched to it immediately before the next vertex in V is revealed. When all vertex are revealed you should grantee that the decisions you made form a matching and the weight of this matching is maximized. Note that the unweighted version of online bipartite matching can be viewed as the weighted one where the edge weight are limited in $\{0, 1\}$.

In fact, the online secretary problem is a special case of weighted bipartite matching. Where the firm is the only one element in U and all applicants form V.

2.2.3 Competitive Analysis

2.2.4 Decentralized Algorithm

Chapter 3 General Model

3.1 Formal Definition

First of all, we need to formally define the objective function in our problem. Generally speaking, for a given instance $G = (U, V, w)$, considering every permutation of the applicants, we have the value of optimal assignment (denoted by OPT), and the value of the algorithm's outcome (denoted by ALG). By saying "competitive ratio α" in the general case, we mean that for any instance G, $\frac{E[ALG]}{E[OPT]} \geq \alpha$. In the following sections, we will present more specific settings, and will define the objectives accordingly.

3.2 Simultaneous stopping rule

Then we turn to the algorithms. For the applicants, everyone adopts the same simple strategy: select the most valuable offer among those she have got. And for firms, we propose two algorithms based on classical online secretary problem's approaches.

One direct algorithm is **simultaneous stopping rule**:

1. Each firm u observes and rejects the first $(r-1)$ applicants. Denote the value of the best applicant among them by t_u which is the threshold. Typically we set r to be a constant fraction of n such as $\lfloor \frac{n}{e} \rfloor + 1$.

2. For each applicant coming in later, each firm u who hasn't been matched before sends her an offer if and only if its value exceeds t_u.

But this algorithm does not work very well in general case if we let $r = \lfloor \beta n \rfloor + 1$ for some constant $\beta < 1$.

Example 3.1. *Let $m = \Theta(n)$. There are two kinds of applicants. First we have $a = \Theta(\log(n))$ 'good' applicants with weight $2 + \epsilon_{u,v}$ for all edges incident to them. The rest applicants are marked as 'bad' with weight $1 + \epsilon_{u,v}$ for all edges incident to them ($|\epsilon_{u,v}| < \frac{1}{2}$ is only used to avoid ties).*

Note that all $\epsilon_{u,v}$ could be arbitrarily small. So in the following discussion, we can ignore them. For any permutation of the applicants, $OPT = m + a$, thus $E[OPT] = m + a$. Now it's time to give a upper bound for $E[ALG]$.

In the first $(r-1)$ rounds, the probability that no firm sees a 'good' applicant is:

$$\prod_{i=0}^{\lfloor \beta n \rfloor} \frac{n-a-i}{n-i} \le (1-\frac{a}{n})^{\lfloor \beta n \rfloor} \le (1-\frac{a}{n})^{\beta n - 1}$$

And if they did observe a 'good' applicant, no 'bad' applicant can be matched since threshold for each firm is set to be near 2.

Therefore

$$\begin{aligned} E[ALG] &\le (1-\frac{a}{n})^{\beta n-1} \times (m+a) + (1 - \prod_{i=0}^{\lfloor \beta n \rfloor} \frac{n-a-i}{n-i}) \times 2a \\ &\le \frac{n}{n-a} e^{-\beta a} \times (m+a) + 2a, \\ \frac{E[ALG]}{E[OPT]} &\le \frac{n}{n-a} e^{-\beta a} + \frac{2a}{m+a} = \frac{1}{n^{\Theta(1)}} + \Theta(\frac{\log n}{n}). \end{aligned}$$

Corollary 3.2. *In the general case, simultaneous stopping rule cannot get better competitive ratio than $\Theta(\frac{\log n}{n})$ if we set $r = \lfloor \beta n \rfloor + 1$ for some constant $\beta < 1$.*

The problem encountered here is that, without global information, firms are competing over a small set of applicants and left all the others aside.

3.3 Simultaneous stopping rule with m slots

To tackle this problem, we slightly modify the algorithm above. Instead of only one threshold, we set m thresholds. Like the algorithm for multiple choice secretary problem proposed in *TODO*, we call it **simultaneous stopping rule with m thresholds**:

1. Each firm u observes and rejects the first $(r-1)$ applicants. Choose the best m applicants to form a threshold set T_u. Denote their values by $t_{u,1}, t_{u,2}, \ldots, t_{u,m}$ in decreasing order. (Some of them are set to be $-\infty$ if not enough).

2. For each applicant v arriving later and each firm u, if the applicant v's value

$w(u, v)$ exceed $t_{u,m}$, then add it to T_u and the least valuable one in T_u is removed. That is, firm u always keeps the best m values in T_u. And v will get an offer from u if and only if $w(u, v)$ is added to T_u and the one removed from T_u is $-\infty$ or has an arrival time less than r.

TODO: more comments

Later on, we are going to propose more specific models to see how well the two algorithms could work. In Section **??** and **??**, simultaneous topping rule can achieve constant competitive ratio under certain conditions; while in Section **??**, the algorithm with m thresholds works well.

Chapter 4 Ranking Model

Consider the following model: all of the weights of edges are independently sampled from the same distribution D which is unknown to us. The concept "competitive ratio α" in this particular model means that for any instance G and distribution D, by taking the expectation over all the possible weights and all the permutations, we have $\frac{E[ALG]}{E[OPT]} \geq \alpha$.

Theorem 4.1. *Simultaneous stopping rule achieves a constant competitive ratio.*

Proof. WLOG we may assume that applicants comes according to the order $\{v_1, v_2, \ldots, v_n\}$ since there is no difference between them.

Firstly let's introduce some notations:

- Event $B(u_i, v_j)$ means that applicant v_j values most to firm u_i.
- Event $P(u_i, v_j)$ means that firm u_i sends an offer to v_j.
- Event $A(u_i, v_j)$ means that firm u_i sends an offer to v_j and v_j accepts it.

Fix a particular firm $u \in U$, first bound the probability that it will get the best applicant according to its preference.

$$\begin{aligned}\Pr(u \text{ gets the best}) &= \sum_{i=1}^{n} \Pr(B(u, v_i)) \times \Pr(A(u, v_i)|B(u, v_i)) \\ &= \sum_{i=r}^{n} \frac{1}{n} \times \Pr(P(u, v_i)|B(u, v_i)) \\ &\quad \times \Pr(A(u, v_i)|B(u, v_i), P(u, v_i))\end{aligned}$$

Given v_i is the best applicant for u, once we ensure that the previous $(i - r)$ applicants are no better than the threshold, u will be free when v_i comes and will send her an offer. That is, we need to ensure that over the last $(i - 1)$ applicants, the best is among the first $(r-1)$ ones. Since all the applicants arrive in a random order, this probability would be $\frac{r-1}{i-1}$. Thus

$$\Pr(P(u,v_i)|B(u,v_i)) \geq \frac{r-1}{i-1}$$

If v_i receives only one offer which is from u, v_i can only choose u. And the condition holds when the value of v_i for each $u' \neq u$ does not exceed the threshold set by u'. In other words, for each $u' \neq u$, over the first $r-1$ applicants together with v_i, the best applicant for u' is not v_i. These $m-1$ events (each with probability $\frac{r-1}{r}$) are independent from each other, and are all independent from $B(u,v_i)$ and $P(u,v_i)$. Thus

$$\Pr(A(u,v_i)|B(u,v_i),P(u,v_i)) \geq (\frac{r-1}{r})^{m-1}$$

To sum up:

$$\Pr(u \text{ gets the best}) \geq \sum_{i=r}^{n} \frac{1}{n} \times \frac{r-1}{i-1} \times (\frac{r-1}{r})^{m-1}$$

Let $p = \frac{r}{n}$ be a constant, and assume that $m \leq \alpha n$ where $\alpha \in (0,1]$ is a parameter. Then $(\frac{r-1}{r})^{m-1} \geq (1-\frac{1}{r})^{\alpha n} = ((1-\frac{1}{r})^r)^{\frac{\alpha}{p}}$ and

$$\Pr(u \text{ gets the best}) \geq ((1-\frac{1}{r})^r)^{\frac{\alpha}{p}} \sum_{i=r}^{n} \frac{1}{n}\frac{r-1}{i-1}$$

Now let n goes to infinity and it approaches the integral:

$$f(p) = e^{-\frac{\alpha}{p}} \int_p^1 \frac{p}{x} dx = -p\ln(p)e^{-\frac{\alpha}{p}}$$

which is a constant depending on the choice of p. Thus, for each firm u, it has at least a constant probability $f(p)$ to get its best applicant, then we have

$$\begin{aligned} E[\text{value of the applicant } u \text{ gets in } ALG] &\geq f(p) \times E[\text{value of the best applicant for } u] \\ &\geq f(p) \times E[\text{value of the applicant } u \text{ gets in } OPT] \end{aligned}$$

Therefore, it is clear to see that

$$\frac{E[ALG]}{E[OPT]} \geq f(p),$$

and we achieves a constant competitive ratio. □

Note that, in the analysis above, it is not necessary that all the weights of edges are independent and identically distributed. All we need is that firms' preference lists are independent from each other. That is, the rank of an applicant v for a firm u_i has nothing to do with the rank of v for another firm u_j.

To generalize the random rank model above, we weaken the restriction by adding correlations between the weights of edges incident to the same applicant, and introduce a new model as follows.

Chapter 5 Gaussian Model

5.1 Normal distribution direction 1

TODO: Give it a name: generative model?

Assume that each applicant has a quality q_i, and the weights of edges incident to a certain applicant v_i are generated independently from a distribution D_i with mean q_i. As we can see, if all qualities are equal and all the distributions are the same, then it is equivalent to the random rank model discussed previously.

As before, we formally define "competitive ratio α" in this model — for any given G, $\{q_i\}_{i=1}^n$, and $\{D_i\}_{i=1}^n$, by taking the expectation over all the possible weights and all the permutations, we have $\frac{E[ALG]}{E[OPT]} \geq \alpha$.

Typically, we consider normal distributions. Assume D_i is a normal distribution $N(q_i, \sigma^2)$ where q_i is the quality of applicant v_i and σ is a fixed constant. Denote that $\delta_{max} = \max_{i \neq j} |q_i - q_j|$ and $\varphi = \frac{\delta_{max}}{\sigma}$. The following theorem holds.

Theorem 5.1. *Simultaneous stopping rule acheives a constant competetive ratio when $\varphi \leq O(\frac{1}{n^2})$.*

WLOG we assume that applicants arrive in the order of $\{v_1, v_2, \ldots, v_n\}$, and that for given qualities $\{q_i'\}_{i=1}^n$, $\{q_i\}_{i=1}^n$ is a random permutation of $\{q_i'\}_{i=1}^n$. This is equivalent with the model where applicants arrive in a random order.

For $i \geq r$, let $D(u, v_i)$ be the event that $w(u, v_i)$ does not exceed the threshold set by u.

Proposition 5.2. *For each $u \in U$ and $i \geq r$,*

$$\Pr(D(u, v_i) | \{q_i\}_{i=1}^n) \geq \Pr(D(u, v_i) | \forall 1 \leq j \leq r-1, q_j = q_i - \delta_{max}).$$

Proof. For convience, let $x_i = w(u, v_i)$ sampled from $N(q_i, \sigma^2)$. WLOG let $q_i = 0$.

$D(u, v_i)$ means that there exists some x_j such that $1 \leq j \leq r-1$ and $x_i < x_j$.

$$
\begin{aligned}
\Pr(D(u,v_i)|\{q_i\}_{i=1}^n) &= 1 - \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x_i^2}{2\sigma^2}} \prod_{j=1}^{r-1} (\int_{-\infty}^{x_i} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x_j-q_j)^2}{2\sigma^2}} \mathrm{d}x_j) \mathrm{d}x_i \\
&= 1 - \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x_i^2}{2\sigma^2}} \prod_{j=1}^{r-1} (\frac{1}{2} + \frac{1}{2}\mathrm{erf}(\frac{x_i - q_j}{\sqrt{2}\sigma})) \mathrm{d}x_i,
\end{aligned}
$$

where $\mathrm{erf}(\cdot)$ is the error function:

$$
\mathrm{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} \mathrm{d}t.
$$

Now take derivative with respect to q_j:

$$
\begin{aligned}
&\frac{\partial \Pr(D(u,v_i)|\{q_i\}_{i=1}^n)}{\partial q_j} \\
=&\int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x_i^2}{2\sigma^2}} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x_i-q_j)^2}{2\sigma^2}} \prod_{k<r,k\neq j} (\frac{1}{2} + \frac{1}{2}\mathrm{erf}(\frac{x_i - q_k}{\sqrt{2}\sigma})) \mathrm{d}x_i \geq 0,
\end{aligned}
$$

which concludes the result. □

Proposition 5.3. *For each $u \in U$ and $i \geq r$,*

$$
\Pr(D(u,v_i)|\forall 1 \leq j \leq r-1, q_j = q_i - \delta_{max}) \geq 1 - \frac{1}{r} - \frac{r-1}{\sqrt{2\pi}}\varphi.
$$

Proof. First we have

$$
\begin{aligned}
&\Pr(D(u,v_i)|\forall 1 \leq j \leq r-1, q_j = q_i - \delta_{max}) \\
=&1 - \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x_i^2}{2\sigma^2}} (\frac{1}{2} + \frac{1}{2}\mathrm{erf}(\frac{x_i + \delta_{max}}{\sqrt{2}\sigma}))^{r-1} \mathrm{d}x_i.
\end{aligned}
$$

Knowing that

$$\begin{aligned}&\frac{\mathrm{d}(\frac{1}{2}+\frac{1}{2}\mathrm{erf}(\frac{x}{\sqrt{2}\sigma}))^{r-1}}{\mathrm{d}x}\\=&\frac{r-1}{\sqrt{2\pi}\sigma}e^{-\frac{x^2}{2\sigma^2}}(\frac{1}{2}+\frac{1}{2}\mathrm{erf}(\frac{x}{\sqrt{2}\sigma}))^{r-2}\\\le&\frac{r-1}{\sqrt{2\pi}\sigma},\end{aligned}$$

by Mean Value Theorem, we have

$$(\frac{1}{2}+\frac{1}{2}\mathrm{erf}(\frac{x_i+\delta_{max}}{\sqrt{2}\sigma}))^{r-1}\le(\frac{1}{2}+\frac{1}{2}\mathrm{erf}(\frac{x_i}{\sqrt{2}\sigma}))^{r-1}+\frac{r-1}{\sqrt{2\pi}\sigma}\times\delta_{max}.$$

Therefore

$$\begin{aligned}&\int_{-\infty}^{\infty}\frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{x_i^2}{2\sigma^2}}(\frac{1}{2}+\frac{1}{2}\mathrm{erf}(\frac{x_i+\delta_{max}}{\sqrt{2}\sigma}))^{r-1}\mathrm{d}x_i\\\le&\int_{-\infty}^{\infty}\frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{x_i^2}{2\sigma^2}}(\frac{1}{2}+\frac{1}{2}\mathrm{erf}(\frac{x_i}{\sqrt{2}\sigma}))^{r-1}\mathrm{d}x_i\\&+\int_{-\infty}^{\infty}\frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{x_i^2}{2\sigma^2}}\frac{(r-1)\delta_{max}}{\sqrt{2\pi}\sigma}\mathrm{d}x_i.\end{aligned}$$

Note that the first term is exactly the probability that x_i is the highest value among $\{x_i\}_{j=1}^{r-1}\cup\{x_i\}$, which equals $\frac{1}{r}$, and the second term

$$\int_{-\infty}^{\infty}\frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{x_i^2}{2\sigma^2}}\frac{(r-1)\delta_{max}}{\sqrt{2\pi}\sigma}\mathrm{d}x_i=\frac{(r-1)\delta_{max}}{\sqrt{2\pi}\sigma}\int_{-\infty}^{\infty}\frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{x_i^2}{2\sigma^2}}\mathrm{d}x_i=\frac{(r-1)\delta_{max}}{\sqrt{2\pi}\sigma},$$

thus

$$\begin{aligned}&\int_{-\infty}^{\infty}\frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{x_i^2}{2\sigma^2}}(\frac{1}{2}+\frac{1}{2}\mathrm{erf}(\frac{x_i+\delta_{max}}{\sqrt{2}\sigma}))^{r-1}\mathrm{d}x_i\\=&\frac{1}{r}+\frac{r-1}{\sqrt{2\pi}}\frac{\delta_{max}}{\sigma}\\=&\frac{1}{r}+\frac{r-1}{\sqrt{2\pi}}\varphi.\end{aligned}$$

Hence concludes the result.

□

Lemma 5.4. *If every firm adopts simultaneous stopping rule, each of them can get its best applicant with constant probability.*

Proof. This is almost the same as Theorem 4.1. Fixing a particular firm $u \in U$, we estimate the probability that u gets its best applicant.

$$\Pr(u \text{ gets its best}|\{q_i\}_{i=1}^n) \geq \sum_{i=r}^{n} \frac{1}{n} \times \frac{r-1}{i-1} \times \Pr(A(u,v_i)|B(u,v_i),P(u,v_i),\{q_i\}_{i=1}^n)$$

Again if no other firm would send an offer to v_i, $A(u,v_i)$ must be true if $P(u,v_i)$ holds. For any $u' \neq u$, $P(u',v_i)$ only depends on the value of $\{q_i\}_{i=1}^n$. Thus, given $\{q_i\}_{i=1}^n$, all the events $\{P(u',v_i)|u' \neq u\}$ are independent from each other, and are all independent from $B(u,v_i)$ and $P(u,v_i)$. Therefore

$$\begin{aligned}
&\Pr(A(u,v_i)|B(u,v_i),P(u,v_i),\{q_i\}_{i=1}^n)\\
\geq &\Pr(\bigcap_{u'\neq u} \overline{P(u',v_i)}|B(u,v_i),P(u,v_i),\{q_i\}_{i=1}^n)\\
= &\prod_{u'\neq u} \Pr(\overline{P(u',v_i)}|\{q_i\}_{i=1}^n)\\
\geq &\prod_{u'\neq u} \Pr(D(u',v_i)|\{q_i\}_{i=1}^n)
\end{aligned}$$

Let $p = \frac{r}{n}$ be a constant, $\varphi \leq \frac{c}{r(r-1)}$ for some constant c since $\varphi \leq O(\frac{1}{n^2})$. Then by proposition 5.2 and 5.3 we have

$$\begin{aligned}
\Pr(A(u,v_i)|B(u,v_i),P(u,v_i),\{q_i\}_{i=1}^n) &\geq (1 - \frac{1}{r} - \frac{r-1}{\sqrt{2\pi}}\varphi)^{m-1}\\
&\geq (1 - (1 + \frac{c}{\sqrt{2\pi}})\frac{1}{r})^{m-1}
\end{aligned}$$

Denote that $c' = 1 + \frac{c}{\sqrt{2\pi}}$. $m \leq \alpha n$ where $\alpha \in (0,1]$ is a parameter. To sum up, we have

$$\begin{aligned}\Pr(u \text{ gets its best}|\{q_i\}_{i=1}^n) &\geq \sum_{i=r}^{n} \frac{1}{n} \times \frac{r-1}{i-1} \times (1-\frac{c'}{r})^{m-1}\\ &\geq ((1-\frac{c'}{r})^r)^{\frac{\alpha}{p}} \sum_{i=r}^{n} \frac{1}{n} \times \frac{r-1}{i-1}\end{aligned}$$

When n goes to infinity, the summation above can be approximated by integral:

$$f(p) = e^{-\frac{c'\alpha}{p}} \int_p^1 \frac{p}{x} \mathrm{d}x = -p\ln(p)e^{-\frac{c'\alpha}{p}}.$$

which is a constant.

Go back to the very beginning, where we are given a quality sequence $\{q_i'\}_{i=1}^n$, and $\{q_i\}_{i=1}^n$ is a random permutation of it. Thus we have

$$\begin{aligned}&\Pr(u \text{ gets its best}|\{q_i'\}_{i=1}^n)\\ =&\sum \Pr(\{q_i\}_{i=1}^n|\{q_i'\}_{i=1}^n) \times \Pr(u \text{ gets its best}|\{q_i\}_{i=1}^n)\\ \geq&\sum \Pr(\{q_i\}_{i=1}^n|\{q_i'\}_{i=1}^n) \times f(p)\\ =&f(p)\end{aligned}$$

when n goes to infinity. □

With Lemma 5.4 we could know that simultaneous stopping rule is almost an optimal strategy for firms. It grantees that each of them could have a very good response with constant probability. Now using the same analysis in Theorem 4.1, it's sufficient to complete the proof of our main theorem.

5.2 Normal distribution direction 2

In the previous sections, we have got a rough idea that simultaneous stopping rule works well when the preference lists of all firms are "different enough" from each other. Recall that in Example 3.1, this algorithm can get no better than $\Theta(\frac{\log n}{n})$−competitive ratio when the firms' view on the applicants are nearly identical to each other. In this situation, we claim that simultaneous stopping rule with m slots can solve the problem.

In this section, we consider the same model as Section **??**. And correspondingly, we define that $\delta_{min} = \min_{i \neq j} |q_i - q_j|$, and $\psi = \frac{\delta_{min}}{\sigma}$. We claim that

Theorem 5.5. *Simultaneous stopping rule with m slots achieves a constant competitive ratio when $\psi \geq \omega(n)$ with large probability.*

Here by saying "achieves a constant competitive ratio with large probability" we mean: with probability approaching 1 over all possible weights, $\frac{E[ALG]}{E[OPT]} \geq c$ for some constant $c > 0$ where the expectation is taken over all possible coming order of applicants. This theorem follows directly from the following two lemmas.

Lemma 5.6. *When $\psi \geq \omega(n)$, for a given sequence $\{q_i\}_{i=1}^n$, with probability approaching 1 that each firm will have the same preference list of applicant as $\{q_i\}_{i=1}^n$*

Proof. For a particular firm u, denote $w(u, v_i)$ by x_i. Note that x_i is sampled from $N(q_i, \sigma^2)$. What we hope to calculate is the probability that for any pair of x_i and x_j where $i \neq j$, $x_i < x_j$ iff $q_i < q_j$.

WLOG, we assume $q_1 > q_2 > ... > q_n$. And we are going to give a lowerbound for $\Pr(x_1 > x_2 > \cdots > x_n)$.

$$\begin{aligned}\Pr(Pr(x_1 > x_2 > \cdots > x_n) &= \Pr(\bigcap_{i=2}^{n}(x_{i-1} > x_i)) \\ &= 1 - \Pr(\bigcup_{i=2}^{n}(x_{i-1} > x_i)) \\ &\geq 1 - \sum_{i=2}^{n} \Pr(x_{i-1} \leq x_i)\end{aligned}$$

Given that $q_i - q_{i-1} \geq \delta_{min}$, we have

$$\begin{aligned}\Pr(x_{i-1} \leq x_i) =& \Pr(x_i - x_{i-1} \geq 0) \\ =& \Pr((x_i - q_i) - (x_{i-1} - q_{i-1}) \geq q_{i-1} - q_i) \\ =& Pr(a - b \geq q_{i-1} - q_i) \\ \leq& \Pr(a - b \geq \delta_{min}),\end{aligned}$$

where a and b are sampled independently from $N(0, \sigma^2)$.

Considering the random variable $a - b$, it's the same as the random variable $a + b$, i.e., the sum of two identical normal distributions. Thus $a - b$ follows another normal distribution $N(0, 2\sigma^2)$. By Chebyshev's inequality:

$$\Pr(x_i \leq x_{i-1}) \leq \Pr(b - a \geq \delta_{min}) = \frac{1}{2}\Pr(|b - a| \geq \delta_{min}) \leq \frac{\sigma^2}{\delta_{min}^2} = \frac{1}{\psi^2}$$

To sum up we have

$$\Pr(x_1 > x_2 > \cdots > x_n) \geq 1 - \frac{n-1}{\psi^2}$$

Thus the probability that each firm has the same preference list as $\{q_i\}_{i=1}^n$ is no less than $(1 - \frac{n-1}{\psi^2})^m$. Given that $m \leq n$ and $\psi \geq \omega(n)$, this probability approaches 1 when n goes to infinity. $\square$

In the following part we assume that each firms has the same preference list as $\{q_j\}_{i=1}^n$.

For convenience, we allow each firm to send offers even after it has been matched. That is to say, it can still send virtual offers (although virtual offers will be rejected all the time). By our assumption, if an applicant receives an offer from a firm, every other firm would also send her an offer which might be virtual. Note that, in the algorithm every firm sends out offers at most m times, thus no more than m applicants would receive offers. This follows that once receiving offers, the applicant can always see "real" offers, and choose the best from them.

Denote the set of all the applicants who receive offers by S. In *TODO* paper it is proved that for each applicant v who is among the best m applicants, $\Pr(v \in S) \geq \frac{r}{n}\ln(\frac{n}{r})$ which is a constant.

Lemma 5.7. *If all firms have the same preference list as $\{q_i\}_{i=1}^n$, with constant probability, each of the best m applicants (with highest qualities) will be matched to her best firm.*

Proof. Assume the coming order of applicants is τ, let $s_{\tau,i}$ be the i-th applicant who receives offers. Fix an applicant v who is one of the best m applicants.

First, for every τ where $s_{\tau,j} = v$ and $j > 1$, by swapping the position between $s_{\tau,j-1}$ and v we can obtain a new order τ'. In the new coming order, v becomes the $(j-1)$-th to receive offers, i.e., by algorithm $s_{\tau',j-1} = v$. Clearly, for two different coming order τ_1 and τ_2 with $s_{\tau_1,j} = s_{\tau_2,j} = v$, the corresponding new orders τ_1' and τ_2' are also different. Thus $|\{\tau|s_{\tau,j-1} = v\}| \geq |\{\tau|s_{\tau,j} = v\}|$. Therefore $\Pr_\tau(s_{\tau,j-1} = v|v \in S) \geq \Pr_\tau(s_{\tau,j} = v|v \in S)$ for all $j > 1$.

Now given that $s_{\tau,j} = v$, among m offers v has received, $j-1$ of them are virtual and must be rejected. If the best offer for v is among the left $m-j+1$ ones, then v will get her best offer. Since all the weights of edges incident to v are generated independently from the same distribution, this event occurs with probability $\frac{m-j+1}{m}$ and it is decreasing by the growth of j, therefore

$$\Pr(v_i \text{ gets her best}|v_i \in S) = \sum_{j=1}^{m} \Pr(s_{\tau,j} = v_i|v_i \in S) \times \frac{m-j+1}{m}$$

We know that $\sum_{j=1}^{m} \Pr(s_{\tau,j} = v_i|v_i \in S) = 1$, by Chebyshev's sum inequality:

$$\Pr(v_i \text{ gets her best}|v_i \in S) \geq \frac{1}{m} \sum_{j=1}^{m} \frac{m-j+1}{m} \geq \frac{1}{2}$$

Combine this with $\Pr(v \in S) \geq \frac{r}{n}\ln(\frac{n}{r})$, and it's done. □

Proof of Theorem 5.5. With Lemma 5.6, what we need to show is that given all firms have the same preference list as $\{q_i\}_{i=1}^{n}$, $\frac{E[ALG]}{E[OPT]} \geq c$ for some constant $c > 0$.

Denote the set of the best m applicant by T. In this situation, $E[OPT] \leq \sum_{v_i \in T} \max_{u \in U} w(u, v_i)$.

According to Lemma 5.7, the algorithm grantees that for every $v_i \in T$, she will be matched to her best firm with constant probability. Which means $E[ALG] \geq \sum_{v_i \in T} c \times \max_{u \in U} w(u, v_i)$ for some constant $c > 0$.

Which concludes the result. □

Corollary 5.8. *Each firm has a probability of $\Omega(\frac{1}{m})$ to obtain the best applicant.*

Proof. By Lemma 5.6, with probability approaching 1 that all firms consider the same applicant as the best. Denote the best applicant by v. By the fact that $Pr(v \in S) \geq \frac{r}{n}\ln(\frac{n}{r})$ is a constant, v would be matched to some firm with constant probability. Since there is no difference between the firms, each firm has a probability of $\frac{1}{m}$ to be chosen. □

Note that in this setting all firms are facing nearly the same situation. Because every firm wants good applicants and here 'good' means almost the same for them. When competing with other firms, the result relies more on applicant's choice instead of their own strategies.

Chapter 6 Conclusion

REFERENCE

[1] KUHN H W. The Hungarian method for the assignment problem[J]. Naval research logistics quarterly, 1955, 2(1-2):83–97.

Acknowledgements

Thx