CS880: Approximation and Online Algorithms	**Scribe:** Taylor Kemp, Mikhail Nedbai
Lecture 26: Matroid Formulation of Secretary Problem	**Date:** Nov. 22 2019

26.1 Introduction to Matroids

A matroid is an extension of linear independence to general sets. The following is a definition for Matroids that will be used throughout the rest of this document.

Definition 26.1.1 *We say a tuple $M = (E, I)$ is a Matroid if it satisfies the following statements*

- $\emptyset \in I$
- $\forall S' \subset S \subset E$ *with* $S \in I$*, we also have* $S' \in I$
- *if* $A, B \in I$ *with* $|A| > |B|$*, then* $\exists x \in A \backslash B$ *with* $\{B \cup x\} \in I$

We now provide a couple examples of Matroids to better understand how they capture and extend linear independence.

Example: Let us first consider the case of linear vectors. Let $M = (E, I)$ where E is the set of vectors in $\mathbb{R}^n$ and I is the set of sets of linear vectors $v_1, v_2, \cdots v_k$ s.t. $v_1, v_2, \cdots v_k$ are linearly independent. We will now show that the following conditions are met. $\emptyset \in I$ by definition of linear independence. Let us consider a set $v_1, v_2, \cdots v_k$ in E with $v_1, \cdots v_k \in I$, then we also have any subset of these vectors also in I. Lastly, if we consider two sets of vectors $v_1, \cdots v_k$ denoted A and $y_1, \cdots y_m$ denoted B, then if $m < k$ and $A, B \in I$, if we look at A, then as the rank of A is larger than the rank of B, there must be some element x in A that is independent from every element in B, hence, $B \cup x \in I$. As a result we have that the set of vectors in $\mathbb{R}^n$ under I forms a Matroid. ■

26.2 Secretary Problem and Matroid Formulation

We consider the secretary problem where we have a set of candidates randomly ordered $x_1, x_2, \cdots x_n$ with unknown corresponding values $v_1, v_2, \cdots v_n$. At step t we must make a snap decision to either hire or not hire candidate x_t. If we do not hire candidate x_t we cannot go back at a later step to hire that candidate. The goal is then to hire the candidate with the highest value v_t. One such algorithm we can consider is a thresholding algorithm where we observe the first half of candidates, then determine a threshold of $max_{t:t<\frac{n}{2}} v_t$. Then in the second half of candidates we hire the first candidate that exceeds this threshold. We note that with probability $\frac{1}{4}$ the second largest value is in the sample and the largest value is not in the sample, hence $\mathbb{E}[ALG] \geq \frac{1}{4} Opt$. This implies a $4-$approximation in expectation to this problem, and also forms a motivation for the algorithm we will use for the more general matroid domain.

We will now consider the more general Matroid formulation of the secretary problem. In this case we have an independent set $\mathcal{I}$ and a universe Ω. We again observe elements $x_1, x_2, \cdots x_n$ with corresponding values $v_1, v_2, \cdots v_n$. Now however, instead of only being able to hire one candidate, we can hire as many candidates as long as the set of hired candidates $B \in \mathcal{I}$. We will assume that the rank of the universe Ω is k. The following result is due to [1]

26.3 Logarithmically Competitive Algorithm for Matroid Domains

Algorithm 1 Threshold Price Algorithm for Matroid Domains

Observe $s = \lceil \frac{n}{2} \rceil$, and denote this sample $S \subset \Omega$ where Ω is the universe.
Let $l* = \arg\max_{l \in S}(w(l))$ denote the element observed of maximum weight.
Pick $j \in \{0, 1, 2, \cdots \lceil log(k) \rceil\}$ uniformly at random.
Let $t = \frac{w(l*)}{2^j}$ denote the threshold.
Initialize B to be an empty set.
for all i in $\{s+1, s+2, \cdots n\}$ **do**
    if $w(l_t) \geq t$ and $l_t \cup B$ is an independent set **then**
        $B = B \cup l_t$
return B

We will now show that the above threshold algorithm obtains a $32log(k)-$competitive ratio. Let us consider $B^* = \{x_1, \cdots x_k\}$ with values $v_1, \cdots v_k$ as the optimal solution. Let us order this set such that $v_i \geq v_{i+1} \forall i$ We will now consider bounding a subset of the overall value of the optimal set B^*. This will prove important in providing a bound on the value of the set B

Claim 26.3.1 *Either* $v_k \geq \frac{v_1}{k}$ *or* $\exists q < k$ *s.t.* $v_{q+1} < \frac{v_1}{k}$

Proof: Since v_i is monotonically decreasing, the above claim follows. ■

Claim 26.3.2 $\sum_{i=1}^{q} v_i > \frac{1}{2}\sum_{i=1}^{k} v_i$

Proof: If $q = k$ then we are done. Let us assume $q < k$. Then we have the following relationship

$$\frac{1}{2}\sum_{i=1}^{k} v_i = \frac{1}{2}(\sum_{i=1}^{q} v_i + \sum_{i=q+1}^{k} v_i) \tag{26.3.1}$$

$$\leq \frac{1}{2}(\sum_{i=1}^{q} v_i + \sum_{i=q+1}^{k} v_{q+1}) \tag{26.3.2}$$

$$< \frac{1}{2}(\sum_{i=1}^{q} v_i + \sum_{i=q+1}^{k} \frac{v_1}{k}) \tag{26.3.3}$$

$$< \frac{1}{2}(\sum_{i=1}^{q} v_i + v_1) \tag{26.3.4}$$

$$\leq \frac{1}{2}(2\sum_{i=1}^{q} v_i) = \sum_{i=1}^{q} v_i \tag{26.3.5}$$

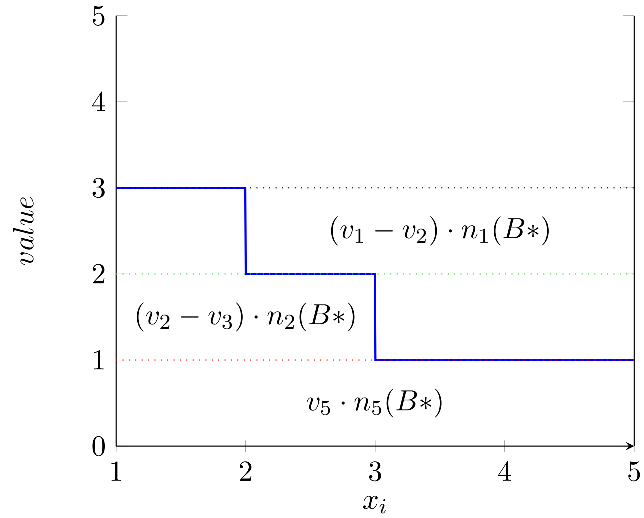
■

We will now look at the number of elements at certain values in the set B and B^*. Let us consider

$A \subset \Omega$, and denote $n_i(A) = |\{j : v_j \in A, v_j \geq v_i\}|$, and $m_i(A) = |\{j : v_j \in A, 2v_j \geq v_i\}|$. We will now take a closer look at the value of B^*.

Claim 26.3.3 $\sum_{i=1}^{q} v_i = v_q n_q(B*) + \sum_{i=1}^{q-1}(v_i - v_{i+1})n_i(B*)$.

Proof:



This illustrative proof shows that $B*$ is equal to the sum of each cross sectional area eg $(v_i - v_{i+1})n_i(B*)$. Summing over all i we obtain the above claim. ■

This gave us a way of looking at the value of B^*, we will now perform a similar computation on showing the value of B, and then show that in expectation this is at most a $32log(k)$ factor off.

Claim 26.3.4 *value of B is at least* $\frac{1}{2}(v_q m_q(B) + \sum_{i=1}^{q-1}(v_i - v_{i+1})m_i(B))$

Proof: In 26.3.1, we can consider the value the sum contributes for the i^{th} element of B^*. Let us first look at the index with half of v_j, v_{j+1} above and below the i^{th} value of B respectively. The amount of value lost for this case is the i^{th} value of B minus $\frac{1}{2}(v_j - v_{j+1})$. Considering all values v_k, $k > j$, we have that the sum of $\frac{1}{2}(v_k - v_{k+1})$ plus $\frac{1}{2}(v_k - v_{k+1})$ is equal to the i^{th} value of B. Looking at all values of B, we have that the value of B is at least $\frac{1}{2}(v_q m_i(B) + \sum_{1}^{q+1}(v_i - v_{i+1})m_i(B))$. ■

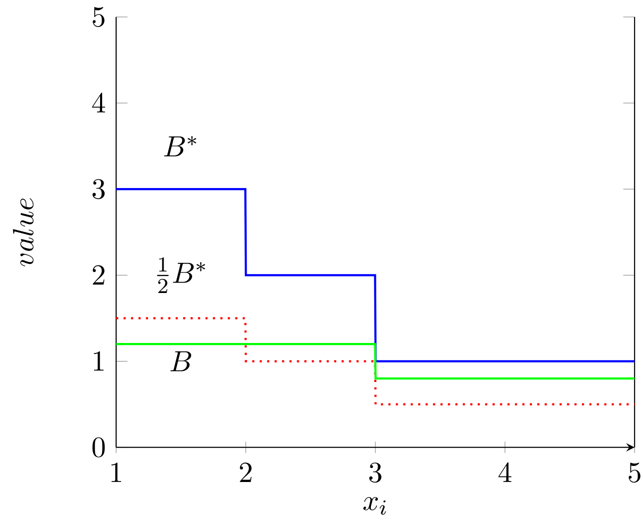
Claim 26.3.5 $\mathbb{E}[m_i(B)] \geq \frac{n_i(B^*)}{8log(k)}$

Proof: First let us consider the case where $i = 1$. Then $w.p. \frac{1}{4}$, the element $x_2 \in S \subset U$ and $x_1 \notin S$. If this is the case then with probability at least $\frac{1}{4}$ we choose x_2 for our threshold, and $w.p. \frac{1}{log(k)}$ we choose our threshold to be v_2 or the second largest weight. From this we have the following relationship as if the second largest weight is the threshold and the largest value is yet unpicked then we are guaranteed to pick the largest value.

$$\mathbb{E}[m_1(B)] \geq \frac{1}{4log(k)} = \frac{n_i(B*)}{4log(k)} \tag{26.3.6}$$

Now we consider the more general case where $i > 1$. We note that $n_i(B*) = i$ by construction of the ordering on $B*$. We then consider the event E that x_1 is in the observed set, and that

Figure 26.3.1: Area of B vs $\frac{1}{2}B^*$



$j* = \arg\max_{j:w(x_1)\leq 2^j w(x_i)} \frac{w(x_1)}{2^j}$ be the smallest j such that our threshold does not exclude any elements greater than or equal to $w(x_i)2^j$. We note that this event E must be possible do to the ordering assumption on $B*$, furthermore, the probability of such an event is $\frac{1}{2log(k)}$. If we consider that there are at least i elements $\{x_1, \cdots x_i\}$ at least the value of $\frac{w(x_1)}{2^{j*}}$, we have that in expectation at least $\frac{i-1}{2} \geq \frac{i}{4}$ of these elements are not in the observed set S when we condition on the event E. From this we have the following relationship.

$$\mathbb{E}[m_i(B)|E] \geq \frac{i-1}{2} \geq \frac{i}{4} = \frac{n_i(B*)}{4} \tag{26.3.7}$$

But this also implies that the following also holds.

$$\mathbb{E}[m_i(B)] \geq \mathbb{E}[m_i(B)|E]\Pr[E] \tag{26.3.8}$$

$$\geq \frac{n_i(B*)}{4} \cdot \frac{1}{2log(k)} = \frac{n_i(B*)}{8log(k)} \tag{26.3.9}$$

As a result we have that $\mathbb{E}[\text{value of B}] \geq \frac{1}{2}\frac{\sum_{i=1}^{q} v_i}{8log(k)} \geq \frac{\text{value of B*}}{32log(k)}$. As a result we have that this threshold algorithm provides a $32log(k)-$competitive ratio. ■

26.4 Submodular Secretary problem

We now consider the version of the Secretary problem on a matroid. Given the set of elements Ω, we consider a matroid $\mathcal{M}(\Omega, \mathcal{I})$ of rank k. The solution S to the problem must be in $\mathcal{I}$, and the objective is to maximize some submodular $f : 2^{\Omega} \to \mathbb{R}$. Formally we can define submodular functions with

$$f(A) + f(B) \geq f(A \cup B) + f(A \cap B) \quad \forall A, B \subseteq \Omega$$

A more intuitive equivalent definition of a submodular function can be provided through the *marginal value*. Let $f_S(A) = f(S \cup A) - f(S), A, S \subseteq 2^{\Omega}$. Then for A such that $A = \{e\}$ the value of $f_S(A)$ is the *marginal* value of f at S. From the definition of submodularity above it is easy to see that for a submodular f, and $A, B \subseteq \Omega$ such that $|B| > |A|$,

$$f_A(\{e\}) \geq f_B(\{e\}) \quad \forall e \in \Omega$$

This property of submodular f can be thought of as decreasing marginal value, that is, the addition of an element to a large set does not increase it value by as much, as it would if it were added to a smaller set. Submodularity of f will allow us to guarantee a good result in this proof by showing that an algorithm that collects enough elements with high marginal weights does not suffer from not including more elements with lower weights too much. Let $w_1 = \max_{e \in \Omega} f(\{e\})$, and let $e_1 = \arg\max_{e \in \Omega} f(\{e\})$. Additionally let $S + e, S \in 2^{\Omega}, e \in \Omega$ denote $S \cup \{e\}$.

The following proof is due to [2], section 4.3. At a high level, we proceed in three steps. We fist show that there exists an algorithm that can collect an independent set of high value, relative to a given threshold τ, over the entire universe Ω. We then show a way to properly choose a value for τ, such that the algoritm can accumulate a value close to the optimal. Finally, we show how translate the algorithm that collects an independent set over the entire universe to the Secretary setting.

Definition 26.4.1 ***Threshold Algorithm***
*Given a threshold τ, initialize $S_1, S_2 \leftarrow \emptyset$. Traverse Ω in an **arbitrary** order. For each element e, if $f(S_1 + e) \geq \epsilon\tau$ and $(S_1 + e) \in \mathcal{I}$, add e to S_1. Otherwise if $f(S_2 + e) \geq \epsilon\tau$ and $(S_2 + e) \in \mathcal{I}$, add e to S_2. Otherwise discard e. Output S_1 or S_2 uniformly at random.*

Throughout this proof let C^* be the optimal set with $f(C^*) = \mathbf{OPT}$. Order the elements of C^* based on their marginal values, or otherwise $f(S + e) - f(S), e \in \Omega, s \in 2^{\Omega}$. Given $\tau > 0$, let $C^*_{\tau} \subseteq C^*$ denote the subset of the optimal solution, where each element added a marginal cost of at least τ when it was added in this order. We first proove a mathematical lemma that will help us along the way. As in [2], special case of claim 2.7 in [3].

Lemma 26.4.2 *Given sets C, $S_1 \subseteq U$, let $C' = C \setminus S_1$, and $S_2 \subseteq U \setminus S_1$.*
Then $f(S_1 \cup C) + f(S_1 \cap C) + f(S_2 \cup C') \geq f(C)$.

Proof:
Observe that by submodularity of f:

$$f(S_1 \cup C) + f(S_2 \cup C') \geq f(S_1 \cup C \cup S_2 \cup C') + f((S_1 \cup C) \cap (S_2 \cup C'))$$
$$f(S_1 \cup C) + f(S_2 \cup C') \geq f(S_1 \cup S_2 \cup C) + f(C')$$

Similarly by submodularity of f:

$$f(S_1 \cap C) + f(C') \geq f(C) + f(\emptyset)$$

Putting the two inequalities together yields:

$$f(S_1 \cap C) + f(C') + f(S_1 \cup C) + f(S_2 \cup C') \geq f(C) + f(\emptyset) + f(S_1 \cup S_2 \cup C) + f(C')$$
$$f(S_1 \cup C) + f(S_1 \cap C) + f(S_2 \cup C') \geq f(C)$$

■

We now proceed to show that **Threshold** algorithm performs well.

Lemma 26.4.3 *For $\epsilon = \frac{2}{5}$, the set produced by the algorithm has the expected value of at least $\frac{\tau|C_\tau^*|}{10}$*

Proof: Suppose $|S_1| \geq \frac{|C_\tau^*|}{4}$ or $|S_2| \geq \frac{|C_\tau^*|}{4}$. Then since our algorithm only adds elements of marginal cost no less than $\frac{2}{5}\tau$, we get a value of at least $\frac{\tau|C_\tau^*|}{20}$.

Otherwise, we can show that the marginal value of elements in S_1 and S_2 is large enough, due to their small size and the submodularity of f, and the addition of elements from C_τ^* would not contribute much.

Lemma 26.4.4 *If $|S_1| \leq \frac{|C_\tau^*|}{4}$ and $|S_2| \leq \frac{|C_\tau^*|}{4}$, then*

$$\exists\ A \subseteq C_\tau^*\ :\ |A| \geq \frac{|C_\tau^*|}{2}, A \cup S_1 \in \mathcal{I}, A \cup S_2 \in \mathcal{I}$$

Proof: Let

$$K_1 = C_\tau^* \cap S_1\ ,\ K_2 = C_\tau^* \cap S_2$$
$$M_1 = S_1 \setminus C_\tau^*\ ,\ M_2 = S_2 \setminus C_\tau^*$$
$$B = C_\tau^* \setminus S_1 \setminus S_2\ ,\ n = |C_\tau^*|$$

Observe that $B \cup K_1$ is independant by the hereditary property. We know that $|M_1| \leq \frac{n}{4}$. By the exchange property, there exists some $E_1 : |E_1| \leq \frac{n}{4} - |K_1|$, such that $B \setminus E_1 \cup K_1 \cup M_1 \in \mathcal{I}$. Similarly for S_2, $\exists E_2\ : |E_2| \leq \frac{n}{4} - |K_2|$, and $B \setminus E_2 \cup K_2 \cup M_2 \in \mathcal{I}$. Thus if we let $A = B \setminus E_1 \setminus E_2$, then

$$|A| \geq |B| - |E_1| - |E_2| \geq (n - |K_1| - |K_2|) - (\frac{n}{4} - |K_1|) - (\frac{n}{4} - |K_2|)$$
$$= \frac{n}{2}$$

■

We claim that $f(S_1) \geq f(S_1 \cup A) - f(A)\epsilon\tau$. Consider $e \in A$. Since $S_1 \cup A \in \mathcal{I}$, e was not added to S_1 not because it would violate independance, but because the marginal value of its' addition was less than $\epsilon\tau$. Therefore

$$f(S_1 \cup A) - f(S_1) = f_{S_1}(A) \leq \sum_{e \in A} f_{S_1}(\{e\}) < |A|\epsilon\tau$$

By the same argument, $f(S_2) \geq f(S_2 \cup A) - f(A)\epsilon\tau$. Furthermore note that since no element in A was picked, $f(S_1 \cap A) = f(S_2 \cap A) = f(\emptyset) = 0$. Then applying lemma 0.2, we obtain

$$f(S_1) + f(S_2) \geq f(S_1 \cup A) + f(S_2 \cup A) + f(S_1 \cap A) - 2\epsilon\tau|A| \geq f(A) - 2\epsilon\tau|A|$$

Since every element of C_τ^* had a value of at least τ when added, and $A \subseteq C_\tau^*$, then $f(A) \geq |A|\tau$. This with the result above implies

$$f(S_1) + f(S_2) \geq |A|\tau(1 - 2\epsilon) \geq \tau(1 - 2\epsilon)\frac{|C_\tau^*|}{2}$$

Thus since $\epsilon = \frac{2}{5}$, the claim is proven. ∎

We will now show, that an excpected value of a solution that only considers elements above a particular threshold will not be too much smaller than the optimal solution, which will allow us to derive the compretitve ratio of **Threshold**.

Lemma 26.4.5 $\sum_{i=0}^{2+\log(2k)} |C^*_{\frac{w_1}{2^i}}| \frac{w_1}{2^i} \geq \frac{f(C^*)}{4} = \frac{\boldsymbol{OPT}}{4}$

Proof: Consider a greedy ordering of C^*, $e_1, e_2, \ldots, e_t$, let $w_j = f_{e_1,e_2,\ldots,e_j}(\{e_j\})$. Consider $\mathcal{S}_\infty = \sum_{i=0}^{\infty} |C^*_{\frac{w_1}{2^i}}| \frac{w_1}{2^i}$. Note that each term in this summation is an upper bound on a cost of $C^*_{\frac{w_1}{2^i}}$ for a given i. For each $e_j \in C^*$, the contribution of e_j to $\mathcal{S}_\infty$ is at least $\frac{w_j}{2}$. To see why, suppose that for some $i, i+1$, $\frac{w_1}{2^{i+1}} \leq w_j \leq \frac{w_1}{2^i}$. In this case the element will fall into $C^*_{2^{i+1}}$, and thus it will contribute the value of $\frac{w_1}{2^{i+1}} > \frac{w_j}{2}$ to the summation. Thus, $\mathcal{S}_\infty \geq \sum_{j=1}^{t} \frac{w_j}{2}$. On the other hand $f(C^*) = \sum_{j=1}^{t} w_j$, which would imply that $\mathcal{S}_\infty \geq \frac{\mathbf{OPT}}{2}$.

Now consider $\mathcal{S} = \sum_{i=0}^{2+\log(2k)} |C^*_{\frac{w_1}{2^i}}| \frac{w_1}{2^i}$. Since the rank of the matroid is k, we know that $|C^*| \leq k$. Consider all of the elements of summation covered by $\mathcal{S}_\infty$, but not $\mathcal{S}$. By the bound on the size of C^* there are at most k such elements. Furthermore since the last index in $\mathcal{S}$ is $2 + \log(2k)$, each element not in $\mathcal{S}$ contributes at the most

$$\sum_{i=2+\log(k)}^{\infty} \frac{w_1}{2^i} \leq \frac{w_1}{4k}$$

Finally, surely $OPT \geq w_1$. Thus $\mathcal{S}_\infty - \mathcal{S} \leq k \cdot \frac{w_1}{4k} = \frac{w_1}{4} \leq \frac{\mathbf{OPT}}{4}$. Therefore since $\mathcal{S}_\infty \geq \frac{\mathbf{OPT}}{2}$,then $\mathcal{S} \geq \frac{\mathbf{OPT}}{4}$, and thus the theorem is proved. ∎

We can use the result above as an upper bound on performance of a thresholding algorithm. Then given $\tau = \frac{w_1}{2^i}$, **Threshold** achieves value of

$$E[\textbf{Threshold}|\tau = \frac{w_1}{2^i}] = |C^*_{\frac{w_1}{2^i}}| \frac{w_1}{20 \cdot 2^i}$$

Suppose we choose τ uniformly at random from $\{w_1, \frac{w_1}{2}, \frac{w_1}{4}, \ldots, \frac{w_1}{8k}\}$. By lemmas 26.4.3,26.4.5 the expected cost of the algorithm is then

$$\frac{1}{3+\log(2k)} \sum_{i=0}^{2+\log(2k)} |C^*_{\frac{w_1}{2^i}}| \frac{w_1}{20 \cdot 2^i} \geq \frac{1}{3+\log(2k)} \frac{\mathbf{OPT}}{80}$$

Which achieves the desired logarithmic compretitve ratio. We now show how to adapt the new Threshold Algorithm to the Secretary setting.

Definition 26.4.6 ***Secretary Algorithm***
Sample have of the elements in the request sequence, and let $W = \max_{e \in \sigma[n/2]} f(\{e\})$. *Choose* i *uniformly at random from* $\{0, 1, \ldots, 2 + \log(2k)\}$. *Run the Threshold algorithm with* $\frac{W}{2^i}$ *as the threshold.*

Lemma 26.4.7 *The Secretary algorithm is* $O(\log(k))$ *compretitve on the Matroid domain in the Secretary setting.*

Proof: Consider two cases

Case 1

Suppose $w_1 \geq \frac{\mathbf{OPT}}{2}$. Then with probability $\Theta(\log(k))$ we choose $i = 0$. With probability $\frac{n/2}{n} \cdot \frac{n/2}{n-1} > \frac{1}{4}$ the second highest singleton weight element is in the first half, and the highest singleton weight element is in the second half of the request sequence. Thus the expected value is at least w_1, which implies that $E[\mathbf{Secretary}] = O(\log(n))$

26.4.1 Case 2

Suppose $w_1 \leq \frac{\mathbf{OPT}}{2}$. Then with high probability $i \geq 2$. In this case with probability $\frac{1}{2}$, e_1 will arrive in the first half of the request sequence, and thus $W = w_1$. For each $e \in C^* \setminus \{e_1\}$, with probability at least $\frac{1}{2}$, e will end up in the second half of the request sequence. Thus since $w_1 \leq \frac{1}{2}\mathbf{OPT}$, the value of the elements in the second half is at least $\frac{\mathbf{OPT}}{4}$ in expectation:

$$E[\mathbf{Secretary}] = \sum_{j=2}^{|C^*|} \frac{1}{2} w_j \geq \frac{\mathbf{OPT}}{4}$$

Therefore, using the result of lemma 3.3, we can conclude that we obtain the cost of $\Omega(\frac{\mathbf{OPT}}{\log(k)})$. ■

References

[1] M. Babaioff, N. Immorlica, and R. Kleinberg, "Matroids, secretary problems, and online mechanisms," in *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '07, (Philadelphia, PA, USA), pp. 434–443, Society for Industrial and Applied Mathematics, 2007.

[2] A. Gupta, A. Roth, G. Schoenebeck, and K. Talwar, "Constrained non-monotone submodular maximization: Offline and secretary algorithms," 2010.

[3] J. Lee, V. Mirrokni, V. Nagarjan, and M. Sviridenko, "Non-monotone submodular maximization under matroid and knapsack constraints," 2009.