Recent advances in the Matroid secretary domain

Shubham Anand Jain
Under the guidance of Professor Rohit Gurjar

March 2020

1 Introduction & brief history

The secretary problem is a classical problem in optimal stopping theory. Also known as the "game of googol" or "marriage problem", it was first introduced in print by Martin Gardner in 1960. The law of best choice, giving the approximation ratio $\frac{1}{e}$ was given by Thomas Bruss in 1984. This problem spanned many variants, such as multiple choice secretary, best secretary in expectation and second best secretary.

A huge interest was sparked in this domain after 2007, due to the seminal work of Babaioff, Immorlica and Kleinberg [8] in their paper "Matroids, secretary problems, and online mechanisms". This paper defined the secretary problem on matroids; the main open problem, which is still open, is as follows - Does there exist an $O(1)$ approximation ratio algorithm for all classes of matroids?

In this survey, we talk about the various variants of matroid secretary that have cropped up from 2007 to 2019, and discuss the approaches used to get the state of the art results in these variants. Thus, the goal of this manuscript is to sketch and highlight the various technique used to approach this problem.

2 Literature Review

The literature for the matroid secretary problem has a rich amount of results, from the matroid theory domain, from the theory of optimal stopping problems and from a mix of these classes. We examine all these previous results and refresh our memories by collecting them in one place. We also go through a high-level overview of the proofs, so as to see how solutions to these different problems tie in with one another.

2.1 Secretary problem

The original secretary problem is stated as follows [14] : A known number of applicants N for a single position arrive in a random order to an employer, who observers the rank of the present applicant relative to those preceding him/her. At each instant, the employer must decide whether to employ the present applicant, or to continue interviewing further applicants. An applicant once passed over, cannot be recalled.

2.1.1 The odds-algorithm

In 2000, a general solution algorithm for optimal stopping problems involving independent indicator functions was presented in [1]. The following theorem was shown:

Theorem 1 (Odds-theorem). Let $I_1, I_2, ..I_n$ be a sequence of independent indicator functions with $p_j = E(I_j)$. Let $q_j = 1 - p_j$ and $r_j = \frac{p_j}{q_j}$. Then an optimal rule τ_n for stopping on the last success exists, and it is to stop on the first index (if any) k with $I_k = 1$ and $k \geq s$, where

$$s = sup\{1, sup\{1 \leq k \leq n : \sum_{j=k}^{n} r_j \geq 1\}\} \tag{1}$$

with $sup\{\phi\} := -\infty$. The optimal win probability is given by $V(n) = \prod_{j=s}^{n} q_j \sum_{j=s}^{n} r_j$.

Since the proof of this theorem is not directly applicable to the matroid secretary variant, we shall omit it here.

2.1.2 Other variants of secretary problem

2.2 General Matroid secretary problem

The general matroid secretary problem was first stated in [8]. In this problem, elements of a matroid are presented to an online algorithm in some order. When an element arrives, the algorithm must make an irrevocable decision to choose this element. Elements can be accepted as long as they form an independent set, and the objective is to maximize the weight of the subset, which is an additive function. A useful characterization of the various problems studied in this area is given in [13].

The various proposed models are broken down into classes which are based on ordering of the matroid elements (Random order/ Adversarial order), assignment of weights (Random/ Adversarial assignment) and the prior information known (the matroid is known, the size of the matroid (n) is known, or the matroid is unknown). With this characterization, we see that the problem considered by [8] is RO-AA-MN, since we assume elements arrive in a random order, but they can be assigned in whatever way. For general matroid classes in the RO-AA-MN model, we first had a $O(\log r)$ approximation given in [8], which was then improved to $O(\sqrt{\log r})$ and recently to $O(\log \log r)$, where r is the rank of the matroid [12], [3], [7].

In 2015, the submodular matroid secretary problem (i.e, over a submodular weight function) was reduced to the modular matroid secretary problem with a constant factor [5].

Similarly, results have been found for matroid secretary problems on the intersection of multiple matroids [6], which reduce this problem to constant competitive if constant competitive algorithms are known for each of the matroid domains.

The original paper [8] gives a constant competitive algorithm for the Truncation of a matroid. The paper [10] solves the secretary problem on general downward-closed set systems, and is almost optimal - they achieve a $O(\log n)$ ratio, against the lower bound of $O(\log n/ \log \log n)$.

2.2.1 Original $\mathbf{O}(\log n)$ algorithm

The original O($\log n$) algorithm for the matroid secretary given in [8] is as follows:

1. Observe $s = \lfloor \frac{n}{2} \rfloor$ elements without picking any element. This set of elements, denoted S, is called the sample.
2. Let $l^* \in S$ be the element with maximum weight. Pick a random number j between 0 and $\lfloor \log k \rfloor$. The threshold price is chosen to be $\frac{w(l^*)}{2^j}$, where $w(l^*)$ is the weight of l^*.
3. Initialize the set of empty elements B to be the empty set.
4. Let l_t be the element in $U\S$ observed at time $t = s+1, ..., n$. If $w(l_t) \geq w(l*)/2^j$ and $l_t \cup B$ is an independent set, then select l_t.

This algorithm is titled the "Threshold price" algorithm, and formed the basis for all matroid secretary algorithms henceforth. All the algorithms to our knowledge use this sampling scheme.

2.2.2 Uniform Matroids

2.3 Transversal Matroids

For transversal matroids, the original algorithm in [8], the Price-sampling algorithm gives a $4d$ approximation, where d is the maximum left degree of the bipartite graph.

2.4 Laminar Matroids

The first constant competitive algorithm for laminar matroids was given by Im & Wang [15], with a constant competitive ratio of $\frac{16000}{3}$ due to a very involved analysis. A considerably simpler algorithm was presented in [11], with a ratio of $3\sqrt{3}e$. The current best known result is the ratio of 9.6, presented in [16]. Here follows a short overview of their proposed algorithms.

A laminar matroid is a generalization of a uniform/partition/truncated matroid. Their definition is as follows (taken from [15]):

Let $\mathcal{F}$ be a laminar family of sets defined over U, i.e. for any two sets $B_1, B_2 \in \mathcal{F}$ we must have $B_1 \subseteq B_2$, $B_2 \subseteq B_1$ or $B_1 \cap B_2 = \emptyset$. Each $B \in \mathcal{F}$ is associated with a capacity $\mu(B)$. A set $S \in \mathcal{U}$ if and only if $\forall B \in \mathcal{F}, |B \cap S| \leq \mu(B)$. Important to note here is that one element may belong to multiple sets in the family, and hence one must ensure that none of them is violated.

An interesting representation of laminar matroids is the tree form [2], where the elements of U are the leaves, and the inner nodes contain the $\mu(B)$ values, signifying the constraints. A set of leaves S is independent if and only if for any inner node v, not more than $\mu(v)$ values of its descendents are in S.

From this tree form, the laminar matroids can be shown to be the special case of the gammoid matroids. First, a few observations:

1. We can consider that $U \in \mathcal{F}$ without loss of generality.
2. If $\exists A \subseteq B$, then $\mu(A) \leq \mu(B)$. This is true, because otherwise the condition on A is redundant, and we can remove the node from the graph.

Now, using these observations, let us prove that any laminar matroid is a gammoid. Using the laminar tree form, multiply each inner node according to the label, i.e. $\mu(v)$. This then forms a gammoid, with the source edges as U and the target vertices as the vertices with no parents (i.e, U.)

Let us now discuss the various approaches used in literature to get constant-competitive algorithms for laminar matroid secretary.

2.4.1 The Kick Next algorithm

Add material here

2.4.2 Reduction to unitary partition matroids

In [11], an algorithm was proposed to reduce laminar matroids to unitary partition matroids, losing a factor of $3\sqrt{3}$ in the process. This leads to a $3\sqrt{3}e$ approximation algorithm for the laminar matroid secretary problem. The algorithm is described as follows.

3 Current results

Type of matroid	Known competitive ratio
Graphic Matroids	$2e$ [9]
Tranversal Matroids	e [17]
Laminar Matroids	9.6 [16]
Regular Matroids	$9e$ [4]
Decomposable Matroids	[4]

Table 1: Matroid secretary advancements

Type of matroid	Known competitive ratio
Unitary partition matroids	261
Tranversal matroids	2496
k-sparse linear matroids	$24ke(3k+1)$
Laminar matroids	585

Table 2: Submodular matroid secretary advancements

4 Our work

4.1 Why matroid secretary algorithms work

The following is a short proof of why we do not generally need to prove variance bounds when providing an algorithm for the matroid secretary problem and it's variants. We know that

$$E[A] \geq \frac{OPT}{e} \tag{2}$$

$$Pr[A \geq \frac{OPT}{10e}] = p \implies E[A] \leq p.OPT + (1-p).\frac{OPT}{10e} \tag{3}$$

Thus,

$$p \geq \frac{9}{10e-1} \geq \frac{1}{3} \tag{4}$$

4.2 Extension of Kesselheim's algorithm to layered graphs

4.3 Extension of Kesselheim's algorithm to the multiple choice secretary problem

Let us consider an extension of Kesselheim's algorithm to the multiple choice secretary problem. To this end, the algorithm we use is as follows:

1. We assume n, k are known in advance
2. We first sample the first $\frac{n}{x}$ values, and maintain a sorted list of these values, called L
3. When a new value v_j comes in, we add it to the sorted list, and then check if it is in the top k elements of the sorted list
4. If we have chosen less than k items, and it was in the top k, we choose this item. Else, we pass.

The following is an analysis of this algorithm:

Case-1: We assume that $k \geq \frac{n}{x}$. This means that, $\frac{n}{x}+1, \ldots . k$ are chosen with probability 1, and then $k+1, \ldots . . n$ are chosen with some probability.

First, we note that, the probability of v_j being in the top k when $j > k$ is $\frac{k}{j}$. This can be seen as all the permutations of the first j elements are equiprobable.

Second, we note that, for any element v_j, it is trivially true that $E[v_j] \geq \frac{OPT}{n}$.

Let I_j be the indicator variable which is 1 when v_j is in the top k when we are at item j. We note that, by Markov's Inequality,

$$Pr\left(\sum_{l=k+1}^{j} I_l \geq \frac{n}{x}\right) \leq \frac{E[\sum_{l=k+1}^{j} I_l]}{\frac{n}{x}} = \frac{\sum_{l=k+1}^{j} \frac{k}{l}}{\frac{n}{x}} \tag{5}$$

Thus,

$$Pr\left(\sum_{l=k+1}^{j} I_l < \frac{n}{x}\right) \geq 1 - \frac{\sum_{l=k+1}^{j} \frac{k}{l}}{\frac{n}{x}} \tag{6}$$

Now, we calculate the expectation of the items that we choose. This is equal to

$$\sum_{j=\frac{n}{x}+1}^{k} E[v_j] + \sum_{j=k+1}^{n} E[v_j|I_j] I_j Pr\left(\sum_{l=k+1}^{j} I_l < \frac{n}{x}\right) \tag{7}$$

First, we note that, when $j > k$, the expected weight of the top k elements is $\geq \frac{j}{n} OPT$, and the expected weight of the j^{th} element, given that it is in the top k, is $\geq \frac{1}{k}\frac{j}{n} OPT$.

Thus, $E[v_j|I_j]I_j \geq \frac{1}{n} OPT$. Thus, we get that, the expectation of items we choose, W_{alg}, is

$$W_{alg} \geq OPT\left(1 - \frac{1}{x}\right) - OPT\frac{kx \sum_{j=k+1}^{n} \sum_{l=k+1}^{j} \frac{1}{l}}{n^2} \tag{8}$$

Note that,

$$\sum_{l=k+1}^{j} \frac{1}{l} \leq ln\left(\frac{j}{k}\right) \tag{9}$$

Also,

$$\sum_{j=k+1}^{n} ln\left(\frac{j}{k}\right) \leq (n-1) ln\left(\frac{n}{k}\right) \leq n ln\left(\frac{n}{k}\right) \tag{10}$$

Now, we use this in the inequality of W_{alg} to get

$$W_{alg} \geq OPT\left(1 - \frac{1}{x}\right) - OPT\frac{kx ln(\frac{n}{k})}{n} \tag{11}$$

Differentiating this to get the maxima wrt x, we get the optimal x as

$$x = \sqrt{\frac{n}{k ln(\frac{n}{k})}} \tag{12}$$

Going back to the original assumption, this case holds when

$$k \geq \frac{n}{x} \implies k \geq nln(\frac{n}{k}) \tag{13}$$

By using numerical solvers, this region is equivalent to

$$\frac{n}{1.76} \leq k \leq n \tag{14}$$

Thus, when $\frac{n}{1.76} \leq k \leq n$, our algorithm is atleast a $1 - 2\sqrt{\frac{kln(\frac{n}{k}}{n}}$ approximation.

5 Interesting open problems

1. Extending Kesselheim's algorithm to multiple secretary problem
2. Gammoid matroid secretary (with 2 vertices to be selected)
3. Graphic matroids - $O(e)$ constant competitive
4. Stochastic departures in matroid secretary - material for secretary problem - Kleinberg'19
5. Submodular matroid secretary - getting a constant or hardness bounds
6. Duals of matroids - can these be shown to be constant competitive?
7. Unions, minors, truncations, contractions - any relation to matroid constant competitiveness?
8. Equivalence of matroid classes under intersection
9. Is there an equivalence between matroid secretary problems and linear programs, similar to [10]?

References

[1] F. Thomas Bruss. "Sum the odds to one and stop". In: *The Annals of Probability* (2000).

[2] Liri Finkelstein. "Two Algorithms for the Matroid Secretary Problem". In: *Research Thesis, Technion - Israel Institute of Technology* (2011).

[3] Oded Lachish. "O(log log rank) Competitive-Ratio for the Matroid Secretary Problem". In: *Foundations of Computer Science (FOCS)* (2014).

[4] Guy Kortsarz Michael Dinitz. "Matroid Secretary for Regular and Decomposable Matroids". In: *European Symposium on Algorithms* (2012).

[5] Rico Zenklusen Moran Feldman. "The submodular secretary problem goes linear". In: *SIAM Journal on Computing* (2015).

[6] Rico Zenklusen Moran Feldman Ola Svensson. "A Framework for the Secretary Problem on the Intersection of Matroids". In: *ACM-SIAM Symposium on Discrete Algorithms (SODA)* (2018).

[7] Rico Zenklusen Moran Feldman Ola Svensson. "A Simple O(log log(rank))-Competitive Algorithm for the Matroid Secretary Problem". In: *ACM-SIAM symposium on Discrete algorithms (SODA)* (2014).

[8] Robert Kleinberg Moshe Babaioff Nicole Immorlica. "Matroids, Secretary Problems, and Online Mechanisms". In: *ACM-SIAM symposium on Discrete algorithms (SODA)* (2007).

[9] Martin Pal Nitish Korula. "Algorithms for Secretary Problems on Graphs and Hypergraphs". In: *Proceedings of the 36th Internatilonal Collogquium on Automata, Languages and Programming: Part II* (2008).

[10] Mohit Singh Niv Buchbinder Kamal Jain. "Secretary Problems via Linear Programming". In: *Integer Programming and Combinatorial Optimization (IPCO)* (2010).

[11] Rico Zenklusen Patrick Jaillet Jose A. Soto. "Advances on Matroid Secretary Problems: Free Order Model and Laminar Case". In: *Integer Programming and Combinatorial Optimization* (2013).

[12] Jan Vondrak Shayan Oveis Gharan. "Improved competitive ratio for the matroid secretary problem". In: *ACM-SIAM symposium on Discrete algorithms (SODA)* (2007).

[13] Jan Vondrak Shayan Oveis Gharan. "On Variants of the Matroid Secretary Problem". In: *Algorithmica* (2013).

[14] M.H Smith. "A secretary problem with uncertain employment". In: *Journal of Applied Probability* (1975).

[15] Yajun Wang Sungjin Im. “Secretary Problems: Laminar Matroid and Interval Scheduling”. In: *ACM-SIAM Symposium on Discrete Algorithms* (2011).

[16] Yajun Wang Tengyu Ma Bo Tang. “The Simulated Greedy Algorithm for Several Submodular Matroid Secretary Problems”. In: *Theory of Computing Systems, Volume 58 Issue 4* (2016).

[17] et al Thomas Kesselheim Klaus Radke. “An Optimal Online Algorithm for Weighted Bipartite Matching and Extensions to Combinatorial Auctions”. In: *European Symposium on Algorithms* (2013).