Document Summarization

Matthieu Cordier
Jipeng Chen

Course : Discrete Optimization (IEOR 4008)

05/10/2019

Abstract

Abstract : Document summarization can be modelised as a maximization of a function reesenting the covering of the summary. Generally, the functions studied by the past are submodular. Based on this, a class of submodular functions is designed for document sumamrization from Lin and Bilmes (2011), which enforces fidelity, and rewards diversity. This function is monotone and submodular, and simple greedy algorithms guarentees a near-optimal solution

In this study, we will compare the performance of this function with a classic one, f_{MMR} submodular but non-monotone, with a new dataset (Opinosis).

1 Introduction

Summarization is a classic text processing problem. Broadly speaking, given one or more documents, the goal is to obtain a concise piece of text that contains the most important information of the document. Summarization has been studied for the past years in various settings—multi-document summarization, summarization using concept spaces in machine learning, Determinantal point processes, submodular optimization. Each domain throws up its own set of improvement, which can be combined together.

While there have been many approaches to automatic summarization, our work is directly inspired by the framework of Lin and Bilmes (2011) []. In this framework, each of the constraints (relevance, redundancy, etc.) is captured as a submodular function and the objective is to maximize their sum. A simple greedy algorithm is guaranteed to produce an approximately optimal summary. This framework obtains the best results on the DUC 2004 dataset. In this this project, we aim at comparing this approach with a more classical one (MMR) using an accelerated modified greedy algorithm on the Opinosis dataset.

2 Problem formulation

We introduce first the following notations :

- The ground set V corresponds to all the sentences in a document.
- Extractive document summarization: select a small subset S that accurately represents the document (ground set V).
- The summary is required to be length-limited :
  - c(S) : cost for sentences S (e.g., the number of words in all sentences of S)
  - b: the budget (e.g., the largest length allowed)
- A set function $f : 2^V \to \mathbf{R}$ measures the quality of the summary S,

Thus, we can formulate the problem as follow

Definition 2.1. *Document Summarization Optimization Problem*

$$S^* \in \underset{S \subset V}{\operatorname{argmax}} f(S) \qquad \textit{subject to} \quad c(S) \leq B \tag{1}$$

This problem is NP-hard and we are looking for some methods to compute near-optimal strategies.

3 Submodular optimization

3.1 Definition

There exists two equivalent definitions of submodularity. The first one is the following :

Definition 3.1. *Let V be a finite set, and denote by 2^V the power set of V, i.e., the family of all subsets of V. A function $f : 2^V \to \mathbf{R}$ is called submodular if, for each $A, B \in 2^V$, we have:*

$$f(A) + f(B) \geq f(A \cap B) + f(A \cup B). \tag{2}$$

Another one, more intuitive, is the following :

Theorem 3.1 (Law of diminishing returns)**.** *Let $f : 2^V \to \mathbf{R}$ be a function.f is submodular if and only if it satisfies the law of diminishing returns, i.e.*

$$f(A \cup \{e\}) - f(A) \geq f(B \cup \{e\}) - f(B) \textit{ for all } A \subseteq B \in 2^V \textit{ and } e \in V \backslash B \tag{3}$$

3.2 Properties

3.2.1 First Properties

We introduce two simple properties useful for the next parts

Property 3.1 (Combination of submodular functions)**.** *If a collection of functions $\{f_i\}_i$ is submodular, then any non negative weighted sum $g = \sum_i \alpha_i f_i$ is submodular, where $\forall i, \alpha_i \geq 0$.*

Property 3.2 (Composition)**.** *Given functions $\mathcal{F} : 2^V \to \mathbb{R}$ and $f : \mathbb{R} \to \mathbb{R}$, the composition $\mathcal{G} = f \circ \mathcal{F} : 2^V \to \mathbb{R}$ is non decreasing submodular if f is non-decreasing concave and $\mathcal{F}$ is nondecreasing submodular.*

This properties will be useful later.

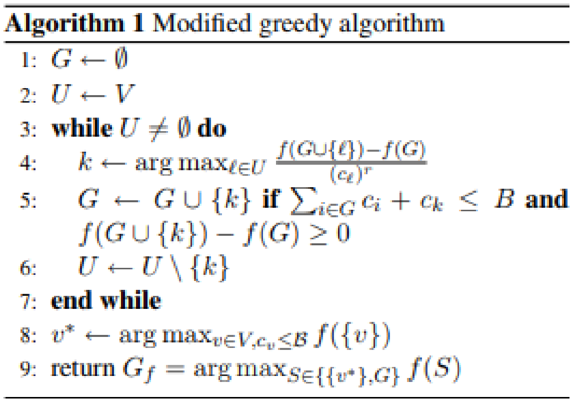
3.2.2 Modified Greedy Algorithm

In discrete optimization, submodularity is a very interesting function property since, even a submodular maximization problem is NP-hard, some greedy algorithms guaranties a near-optimal solution for any submodular function. In this study, we will work with the Algorithm 1 (Figure 1) [].

Theorem 3.2 (Submodular maximization with cost function and budget)**.** *f a normalized monotone submodular function. Let $\hat{S}$ be the set output by Algorithm 1 with $r = 1$ (enhanced greedy algorithm, Lin and Bilmes [2]) and let S^* be the optimal solution. Then $f(\hat{S}) \geq (1 - \frac{1}{\sqrt{e)}})f(S^*)$*

In practice, the near-optimality is better, which makes submodularity a very interesting property for the document summarization problem.

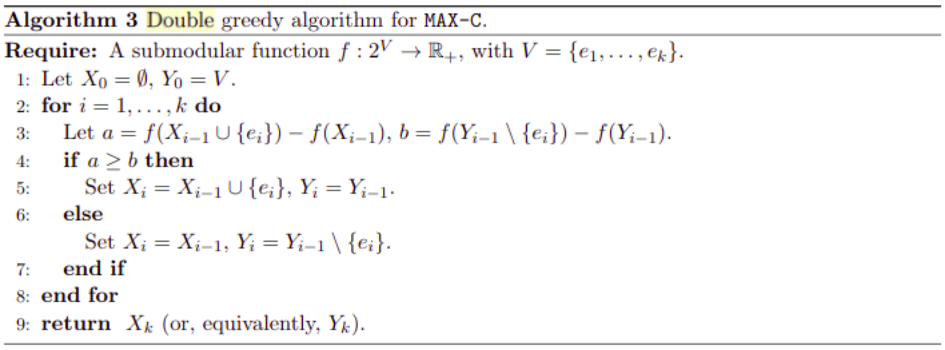
Finally, we can a little bit speed up the algorithm with the **accelerated version** of this greedy algorithm ("lazy greedy") [] and uses a trick to diminish the number of evaluations of f for the "*argmax*".



Algorithm 1 Modified greedy algorithm

```
1: $G \leftarrow \emptyset$
2: $U \leftarrow V$
3: while $U \neq \emptyset$ do
4:     $k \leftarrow \arg\max_{\ell \in U} \frac{f(G \cup \{\ell\}) - f(G)}{(c_\ell)^r}$
5:     $G \leftarrow G \cup \{k\}$ if $\sum_{i \in G} c_i + c_k \leq B$ and
       $f(G \cup \{k\}) - f(G) \geq 0$
6:     $U \leftarrow U \setminus \{k\}$
7: end while
8: $v^* \leftarrow \arg\max_{v \in V, c_v \leq B} f(\{v\})$
9: return $G_f = \arg\max_{S \in \{\{v^*\}, G\}} f(S)$
```

Figure 1: Modified greedy algorithm for submodular function (Bilmes and Line, 2010) [2]



Algorithm 3 Double greedy algorithm for MAX-C.

```
Require: A submodular function $f : 2^V \to \mathbb{R}_+$, with $V = \{e_1, \ldots, e_k\}$.
1: Let $X_0 = \emptyset$, $Y_0 = V$.
2: for $i = 1, \ldots, k$ do
3:     Let $a = f(X_{i-1} \cup \{e_i\}) - f(X_{i-1})$, $b = f(Y_{i-1} \setminus \{e_i\}) - f(Y_{i-1})$.
4:     if $a \geq b$ then
5:         Set $X_i = X_{i-1} \cup \{e_i\}$, $Y_i = Y_{i-1}$.
6:     else
           Set $X_i = X_{i-1}$, $Y_i = Y_{i-1} \setminus \{e_i\}$.
7:     end if
8: end for
9: return  $X_k$ (or, equivalently, $Y_k$).
```

Figure 2: Double Greedy algorithm without a budget ([1])

3.2.3 Double Greedy algorithm

When f is only submodular, but non monotone, a double greedy algorithm guaranties also a lower bound for the near-optimal subset found.

Theorem 3.3 (Submodular maximization with double greedy (from class))**.** *For f a submodular function, double greedy guarantees at least a $\frac{1}{3}$-approximation of the optimal solution.*

Hoever, this algorithm des not take into account the constraints of the problem. We need to have a feasible solution, so modify the line 4 (1) such as a new selected item for X_i needs to respect the budget. However, this modification is important because it breaks the symmetry of the algorithm. Still, we want to know how this modified double greedy algorithm performs in practice.

4 Monotone submodular objective functions for the document summarization problem

4.1 Rouge-N : a submodular function measuring the performance of the summarization

We define $\mathcal{F}_{ROUGE-2}$ by :

Definition 4.1. *ROUGE-2 (recall) For a document with K reference summaries, Rouge-2 is defined by ;*

$$\mathcal{F}_{Rouge-N} = \frac{\sum_{i=1}^{K} \sum_{e \in R_i} \min(c_e(S), r_{e,i})}{\sum_{i=1}^{K} \sum_{e \in R_i} r_{e,i}} \tag{4}$$

where $r_{e,i}$ is the number of times 2-gram e occurs in reference summary (gold summary) i. $c_e : 2^V \rightarrow \mathbb{N}$ is the number of times 2-gram e occurs in summary S, and R_i is the set of 2-gram contained in the reference summary. **It can be seen as the number of common bigrams between the gold summaries and the summary S over the number of bigrams in the gold summaries.**

$\mathcal{F}_{Rouge-N}$ is **monotone submodular** (intuition : min is a concave function). It has been shown that it is highly correlated with human summaries, so it a good metric to measure the performance of summarization task.

4.2 A non monotone submodular function : f_{MMR} function

We define f_{MMR} as the following :

Definition 4.2. f_{MMR}

$$\forall S \subseteq V, f_{MMR}(S) = \sum_{i \in V} \sum_{j \in S} w_{i,j} - \lambda \sum_{i,j \in S : i \neq j} w_{i,j}, \lambda \geq 0 \tag{5}$$

where $w_{i,j} = cosinus\_similarity(X_i^{tfid}, X_j^{tfid})$

This **function is submodular, but not monotone**.

Lin and Bilmes : Algorithm 1 (modified greedy) gives a good solution with high probability for f_{MMR}, even if the function is not monotone [2]. We will also compute a near-optimal solution with the double greedy alorithm (suitable for non-monotone functions).

4.3 A monotone submodular function : $\mathcal{F}_{divcov}$ function

Definition 4.3 (Coverage-Diversity submodular function (Lin and Bilmes, 2010 [])). *We say that* $\mathcal{F} : 2^V \rightarrow \mathbf{R}$ *is coverage-diversity submodular function if it* *has the following form :*

$$\forall S \in 2^V, \mathcal{F}(S) = \mathcal{L}(S) + \lambda \mathcal{R}(S) \tag{6}$$

$$= \sum_{i \in V} \min\{\mathcal{C}_i(S), \alpha \mathcal{C}_i(V)\} + \lambda \sum_{i}^{K} \sqrt{\sum_{j \in P_i \cap S} r_j} \tag{7}$$

Where $\lambda \geq 0$, $\mathcal{C}_i(S) : 2^V \rightarrow \mathbf{R}$ is a monotone submodular function, and r_i indicates a singleton reward, which represents the importance of i in the summary. $(P_i)_{i=1\ldots K}$ is partition of the set V. Thus, $\mathcal{L}$ measures the coverage ("fidelity") and $\mathcal{R}$ rewards diversity.

In [3], $\forall i, C_i(S) = \sum_{j \in S} w_{i,j}$ and $\forall j, r_j = \frac{1}{N} \sum_{i \in V} w_{i,j}$ where $w_{i,j}$ is the cosinus similarity between i and j.

Thus, **we reuse this formulation and call $\mathcal{F}_{covdiv}$ the following function** :

$$\mathcal{F}_{covdiv}(S) = \sum_{i \in V} \min\{\sum_{j \in S} w_{i,j}, \alpha \sum_{j \in V} w_{i,j}\} + \lambda \sum_{i}^{K} \sqrt{\sum_{j \in P_i \cap S} \frac{1}{N} \sum_{i \in V} w_{i,j}} \tag{8}$$

where $w_{i,j}$ **is the cosinus similarity between X_i^{Tfid}, X_j^{Tfid}** ([])

To define and optimise a coverage-diversity function, we need to compute a clustering to create a partition. In the experiment, we will use a k-mean algorithm to compute clusters over the sentences.

Let's now focus a little bit on each term $\mathcal{L}$ and $\mathcal{R}$, and prove that they are both monotone and submodular.

4.3.1 Coverage function

Property 4.1. *The function $\mathcal{L} : 2^V \rightarrow \mathbb{R}$ defined by*

$$\forall S \subseteq V, \mathcal{L}(S) = \sum_{i \in V} \min\{\sum_{j \in S} w_{i,j}, \alpha \sum_{j \in V} w_{i,j}\}$$

is non-decreasing submodular.

Proof. The non-decreasing property is trivial.

- (Sub)modularity of $\sum_{j \in S} w_{i,j}$: Let's take $A \subseteq B \subset V$ and an element v not in A. $\sum_{j \in (A+v)} w_{i,j} - \sum_{j \in (A)} w_{i,j} = \sum_{j \in (B+v)} w_{i,j} - \sum_{j \in (B)} w_{i,j}$, so this is "modular" (equality).
- The min operator is concav. Thus, the *property 3.2* shows that $\min\{\sum_{j \in S} w_{i,j}, \alpha \sum_{j \in V}\}$ is submodular.
- Since $\mathcal{L}$ is a sum of positive weighted submodular function, $\mathcal{L}$ is submodular.

□

4.3.2 Diversity function

Property 4.2. *For a partition $\{P_k\}_k$, the function $\mathcal{R} : 2^V \to \mathbb{R}$ defined by*

$$\forall S \subseteq V, \mathcal{L}(S) = \sum_i^K \sqrt{\sum_{j \in P_i \cap S} \frac{1}{N} \sum_{i \in V} w_{i,j}}$$

is non-decreasing submodular.

Proof. Again, the non-decreasing property is trivial.

- It is easy to see that $\sum_{j \in P_i \cap S} \frac{1}{N} \sum_{i \in V} w_{i,j}$ is modular
- Since the square function is concave, from the *property 3.2*,

  $$\sqrt{\sum_{j \in P_i \cap S} \frac{1}{N} \sum_{i \in V} w_{i,j}}$$

  is submodular.
- Thus, since this a sum of submodular function, positively weighted. This concludes the proof (*property 3.1*)

□

5 Experiment and results

5.1 The Dataset

The dataset is called **Opinosis**. This dataset contains sentences extracted from user reviews on a given topic (“performance of Toyota Camry” and “sound quality of ipod nano”, etc.) from various sources – Tripadvisor (hotels), Edmunds.com (cars) and Amazon.com (various electronics).

In total there are 51 topics with each topic having approximately 100 sentences (on average).

The dataset file also comes with gold standard summaries used for the summarization paper listed above, which is very useful to compute the metric.

5.2 Process of Experiment

In the data, the documents are given. For each document, there is also a set of corresponding reference summaries which can be used to construct the function Rouge-N for evaluating the quality of the summaries.

As for the document, 2 functions based on it for evaluating the quality of the summaries are used. They are f_{MMR} and $\mathcal{F}_{covdiv}$. Three optimization methods like greedy algorithm, lazy greedy algorithm are used to maximize the f_{MMR} and $\mathcal{F}_{covdiv}$. When they are maximized, the corresponding summaries S_{MMR} and S_{covdiv} are treated as the output of the 2 evaluation functions. The double greedy algorithm is also used to maximize f_{MMR} where the output summaries is denoted by $S_{MMR-double}$.

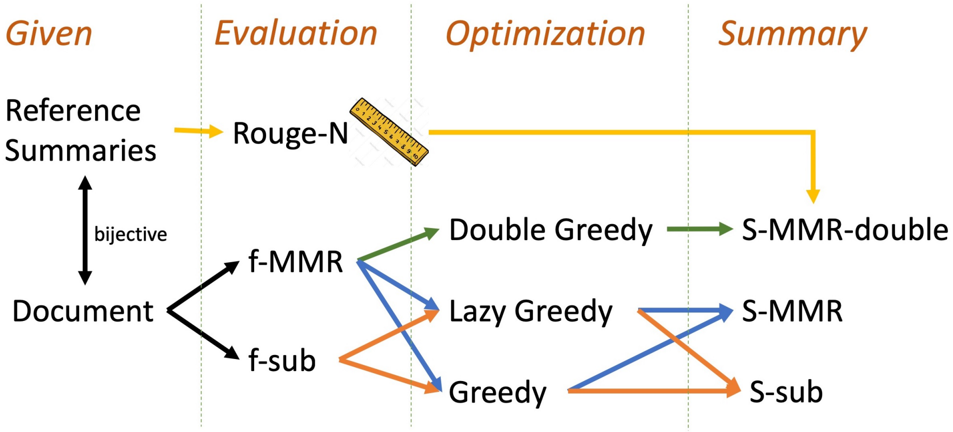


Figure 3: Process of the Experiments ($f_{sub} = \mathcal{F}_{covdiv}$, $S_{sub} = S_{covdiv}$)

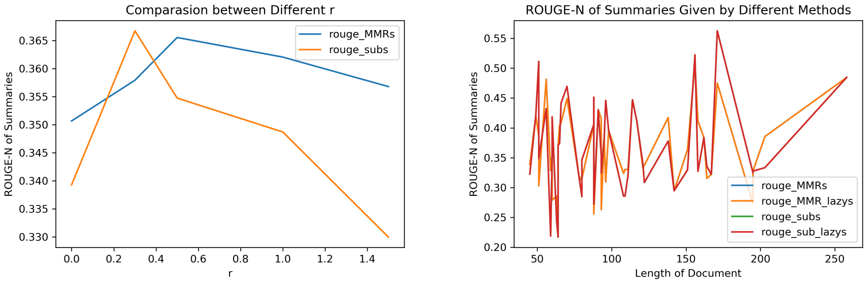


Figure 4: Rouge-N of Output Summary $(sub\, for\, covdiv)$

5.3 Results

Our main goal is to compare the different evaluation functions based on the documents. The comparison are done from 2 aspects, quality of the summary and optimization time. Double greedy is also applied to optimize f_{MMR} which is proved to be a bad idea.

5.3.1 Comparison of Summaries

From the left part in figure 3, which plotted the average Rouge-N value of the 51 documents against different r. When r equals 0.3, the summaries given by $\mathcal{F}_{covdiv}$ is better than that given by f_{MMR}.

For the right part in figure 3, which plotted Rouge-N value against length of the documents when r equals 0.3, the $\mathcal{F}_{covdiv}$ outperforms f_{MMR} when the documents length is around 160, while summaries of the rest documents are very similar to each other. Another thing is that, if we use the same evaluation function, the summaries given by lazy greedy algorithm are same to the summaries given by greedy algorithm. In the right part of figure 3, we can see the Rouge-N value of the summaries given by lazy greedy algorithm overlap with that given by greedy algorithm.

5.3.2 Comparison of Optimization Time

As for the optimization time, if the greedy algorithm is used, the $\mathcal{F}_{covdiv}$ is much longer to optimizer compared to f_{MMR}, especially when the document is longer than 100 sentences, which is shown in the figure 4.

If we use the lazy greedy algorithm, the optimization time can be reduced a lot for both $\mathcal{F}_{covdiv}$ and f_{MMR}, especially the $\mathcal{F}_{covdiv}$.

However, Lin's stated that monotone submodular function has a good scalability that the argmax in the algorithm can solved $O(log(n))$ calls of $\mathcal{F}_{covdiv}$ [3]. But in practice, the optimization time for $\mathcal{F}_{covdiv}$ is much longer than f_{MMR}. This should be blamed to the high computation cost for $\mathcal{F}_{covdiv}$ itself, which is shown in the figure 5.

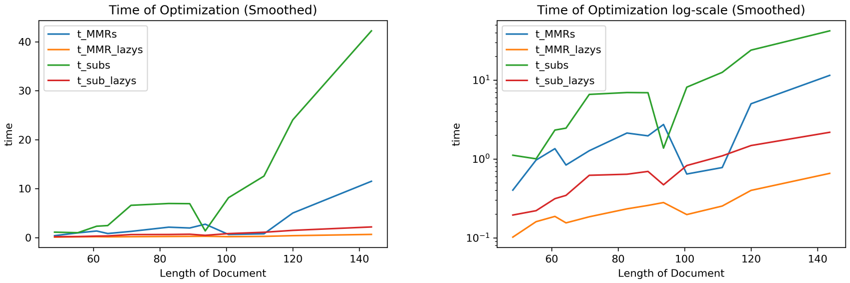


Figure 5: Optimization Time $(sub\, for\, covdiv)$

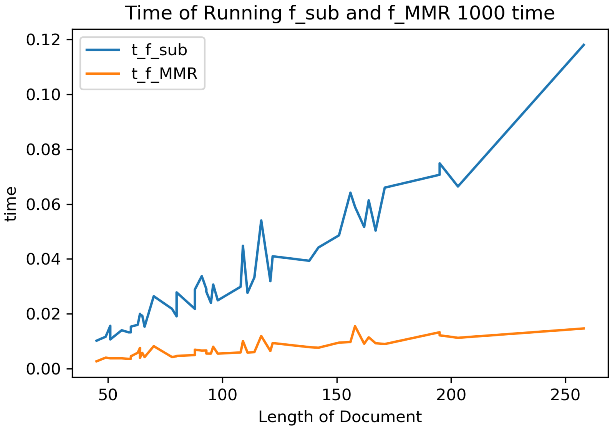


Figure 6: Computation Time for f_{MMR} and $\mathcal{F}_{covdiv}$ $(sub\, for\, covdiv)$

As a result, applying the lazy greedy algorithm to $\mathcal{F}_{covdiv}$ gives a better reduction in time since it reduces the time s the optimization calls $\mathcal{F}_{covdiv}$ which is very computationally expensive.

5.3.3 Comments on Double Greedy

Since f_{MMR} is submodular but not monotone, it is also interesting to see if there is any chance to have a better result if f_{MMR} is optimized by double greedy.

The document summarization is a knapsack problem essentially. So, a modified version of double greedy algorithm is applied. Every time when we want to add something into X_0 in Algorithm 3, it should be checked whether adding it will break the knapsack constraint. As a result, we get poor Rouge-N values of summaries which is represented by the green in the left part of figure 6.

The original version of double greedy algorithm is also applied to the f_{MMR}, which gives really high Rouge-N values represented by the green line in the right

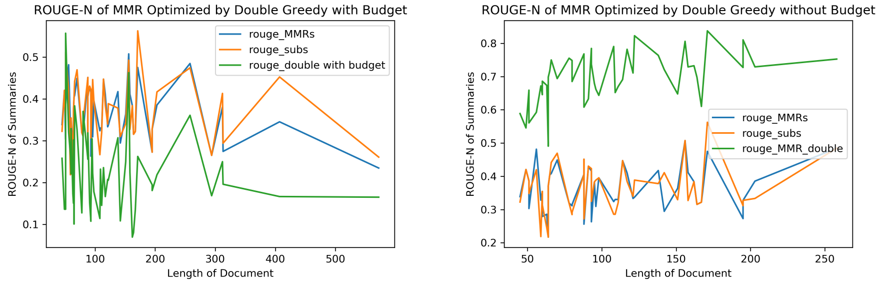


Figure 7: Results Given by Double Greedy $(subforcovdiv)$

part of figure 6. However, even though the summaries given by double greedy algorithm are really good, the cost of them can be of 700 words in average, which is not acceptable for the limit length of the summaries.

So, it is not a good idea to apply double greedy algorithm to optimize the f_{MMR}. Double greedy algorithm is not suitable for knapsack problem.

6 Conclusion and discussion

In document summarization task, the submodularity naturally exists in many models. Not only do many existing automatic summarization methods corresponding to submodular function optimization, but also the classic used Rouge-N evaluation is proved to be monotone submodular.

In order to design a submodular objective that best models the document summarization, a powerful class of monotone submodular functions are introduced. The quality of a summary is modeled into two part, fidelity and diversity. While more complicated machine learning techniques could be incoporated into the functions (use a different similarity measure, by learning a concept kernel) the current function do achieved good results compared to the classic model MMR if the parameters are chosen carefully.

The submodular optimization time can be solved efficiently and effectively since the time of calls to the optimized function is bounded in each step. Even though the computation cost for each call can be very high, the time can be much lower if lazy greedy algorithm is applied.

As for the double greedy algorithm in document summarization, considering the knapsack constraint in optimization will result in a poor result while direct optimization will lead to a result which is unreasonable for the knapsack constraint. So, double greedy is not a good choice for optimization in such knapsack problem.

Next, we should compare those results with Determinantal Point Processs, another technique where each subset has a defined probability by a specific kernel (which can be learned). In this very elegant theory, taking the subset of maximum probability is also a submodular optimization problem.

References

[1] N. Buchbinder et al. "A Tight Linear Time (1/2)-Approximation for Unconstrained Submodular Maximization". In: *2012 IEEE 53rd Annual Symposium on Foundations of Computer Science*. 2012, pp. 649–658. DOI: `10.1109/FOCS.2012.73`.

[2] Hui Lin and Jeff Bilmes. "Multi-document Summarization via Budgeted Maximization of Submodular Functions". In: *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. HLT '10. Los Angeles, California: Association for Computational Linguistics, 2010, pp. 912–920. ISBN: 1-932432-65-5. URL: `http://dl.acm.org/citation.cfm?id=1857999.1858133`.

[3] Hui Lin and Jeff A. Bilmes. "A Class of Submodular Functions for Document Summarization". In: *ACL*. 2011.