Boyer-Moore and Greedy Algorithms to Display Personalized Contents on Social Media Feed

Rhea Elka Pandumpi 13519047
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
13519047@std.stei.itb.ac.id

***Abstract*—Social media's usage is not limited only to connection establishment, but also to promote, inform, and share news to the users. To keep the user engaged in the social media, a form of personalization may be required. This paper discusses an application of Boyer Moore and Greedy algorithms to perform simple personalization of contents on social media feed.**

Keywords—social media; personalization; content; boyer-moore; greedy

I. Introduction

An increase of the usage of social media is present in today's living. The initial purpose to connect with people have broaden into many other functionalities like promoting products, informing, and sharing news to the users. Maintaining user's interest in what social media has to offer is very crucial, because spams and unnecessary informations has taken a large place in the platforms. To answer the problem, an enhancement of quality and personalization in the posts on the social media must be made.

Most social media platforms have utilized the data access that they gained from the users. Product advertisements that are displayed for user's consumption are not rarely what the user has already been interested in. Most assume that they are a result of heavy and deep analysis performed on the data collected from the users' activity on the platform, such as the search or visits history of the users. However, the platforms never went as far as to uncover the algorithm that lies beneath their means of personalization and leaves the wonder open for all the users.

This paper, inspired by the personalization that has been heavily used by the social media platforms, attempts to apply two of the algorithm strategies to solve the problem of personalization, namely Boyer-Moore and Greedy algorithms in a more simplified approach. The method heavily depends on user's choice of what topics to display on their feed (a technique some social media platforms have already used). The user's choice will then be used as a pattern that the Boyer-Moore algorithm searches for the keyword in the available posts. After applying the Boyer-Moore Algorithm, the key part of personalization lies on the Greedy algorithm, used to optimize the personalization in the kind of approach that shall be further discussed in the paper.

II. Theory

A. Greedy Algorithm

A greedy algorithm is an algorithm that constructs an object X one step at a time, at each step choosing the locally best option. In some cases, greedy algorithms construct the globally best object by repeatedly choosing the locally best option.

Greedy algorithms have several advantages over other algorithmic approaches:

1. Simplicity: Greedy algorithms are often easier to describe and code up than other algorithms.
2. Efficiency: Greedy algorithms can often be implemented more efficiently than other algorithms

Greedy algorithms have several drawbacks:

1. Hard to design: Once you have found the right greedy approach, designing greedy algorithms can be easy. However, finding the right approach can be hard.
2. Hard to verify: Showing a greedy algorithm is correct often requires a nuanced argument

Greedy algorithm elements:

1. Candidate set, C: the candidates that will be chosen in each step (example: edge/node in graph, job, task, coin, item, character, etc.)
2. Solution set, S: the candidates that have been chosen
3. Solution function: determining whether the chosen candidates equate to solution or not
4. Selection function: choosing the candidate based on a specific, heuristic greedy strategy
5. Feasibility function: checking the feasibility of the chosen candidate to be inserted into the solution set
6. Objective function: maximizing or minimizing

Greedy algorithm involves searching of a subset, S, from the candidate set C, in which S must satisfy the specified criteria, that is, S gives a solution and S is being optimized by an objective function.

B. Integer Knapsack Problem With Greedy Approach

Consider n items, each available in unlimited quantities, and a knapsack which can hold a maximum weight of W. Given that each item i weighs w_i units and has a benefit b_i points, what is the optimal way to fill the sack so as to maximize the overall benefit? We can't pick part of an item. Either we take it or don't. That is why it is integer knapsack.

Maximize $\Sigma_i\ a_i b_i$ such that $\Sigma_i\ a_i w_i <= W$

To apply greedy strategy we need to first check if the problem exhibits (i) optimal substructure property and (ii) greedy choice property.

Optimal substructure: The recursive fomulation above reveals the optimal substructure. The problem of optimally filling the knapsack of capacity W contains within itself n subproblems of optimally filling the knapsak of capacities $W - w_i$.

Greedy choice property: In the above recursive solution, we reduced the problem of size W to n subproblems, each of size $W - w_i$. Solution to exactly one of these n subproblems ultimately leads to the optimal solution for the problem. Is there a way to determine which of the n would lead to optimal solution and eliminate the other n-1 subproblems from computation? In other words, can we make a sequence of locally optimal choices that leads to globally optimal solution?

One approach would be to pick the item with highest benefit per unit possible with the remaining capacity of the knapsack. Let bpu_i denote the benefit per unit for each item i.e. $bpu_i = b_i/w_i$.

$bpu = (bpu_1, bpu_2, bpu_3) = (5/2, 8/3, 14/5) = (2.5, 2.67, 2.8)$

Item3 seems to be more valuable followed by item2 and item3. Will this greedy strategy result in the optimal solution? For the above problem it seems to provide the optimal solution.

W = 7. Pick item 3. This reduces the capacity to $W - w_3 = 7 - 5 = 2$. Now W = 2. Items 3 and 2 cannot be picked now, since it would exceed knapsack's capacity. So pick item1. This reduces the capacity to $W - w_1 = 2 - 2 = 0$. Now W = 0. The knapsack is full.

The overall benefit achieved is 14 + 5 = 19 which is optimal.

Lets check for a different W. W = 6.

W = 6. Pick item 3. This reduces the capacity to $W - w_3 = 6 - 5 = 1$. Now W = 1. We can't pick any item now since $W < w_1, w_2, w_3$

The overall benefit achieved is 14 which is not optimal. The optimal solution is to pick item2 twice that gives the overall benefit of 2 * 8 = 16. It is safe to conclude that greedy does not guarantee optimal solution for integer knapsack.

C. Boyer-Moore

The Boyer-Moore algorithm is consider the most efficient string-matching algorithm in usual applications, for example, in text editors and commands substitutions. The reason is that it woks the fastest when the alphabet is moderately sized and the pattern is relatively long.

The algorithm scans the characters of the pattern from right to left beginning with the rightmost character. During the testing of a possible placement of pattern P against text T, a mismatch of text character T[i] = c with the corresponding pattern character P[j] is handled as follows: If c is not contained anywhere in P, then shift the pattern P completely past T[i]. Otherwise, shift P until an occurrence of character c in P gets aligned with T[i].

This technique likely to avoid lots of needless comparisons by significantly shifting pattern relative to text.

Last Function

We define a function last(c) that takes a character c from the alphabet and specifies how far may shift the pattern P if a character equal to c is found in the text that does not match the pattern.

last(c) = index of the last occurrence of c in pattern P if c is in P, otherwise -1.

For example consider the table below.

Table 1. Boyer-Moore Algorithm Last Occurence Example (a)

T:	0	1	2	3		4	5	6	7	8	9
	a	b	a	c		a	a	B	a	c	c
P:	a	b	a	c		a	b				
	0	1	2	3		4	5				

last(a) is the index of the last (rightmost) occurrence of 'a' in P, which is 4. last(c) is the index of the last occurrence of c in P, which is 3. 'd' does not exist in the pattern there we have last (d) = -1. Therefore, last(b) is the index of last occurrence of b in P, which is 5.

Table 2. Boyer-Moore Algorithm Last Occurence Example (b)

c	a	b	c	d
last(c)	4	5	3	-1

The figure below is a pseudocode of how Boyer Moore algorithm works.

BOYER_MOORE_MATCHER (T, P)

Input: Text with n characters and Pattern with m characters
Output: Index of the first substring of T matching P

```
Compute function last
i ← m-1
j ← m-1
Repeat
   If P[j] = T[i] then
      if j=0 then
         return i        // we have a match
      else
         i ← i -1
         j ← j -1
   else
      i ← i + m - Min(j, 1 + last[T[i]])
      j ← m -1
until i > n -1
Return "no match"
```

Fig 1. Boyer Moore Algorithm Example

D. Social Media

Social media help share information and build relationships with various audiences. From recruiting new students and faculty, to raising money, to diffusing incidents and situations—leveraging Twitter, Facebook, LinkedIn and other networks are powerful tools in the hands of communicators.

A few social media statistics:

1. Internet users have an average of 7 social media accounts (Global Web Index)
2. Nearly 80% of social media time is spent on mobile devices (Marketing Land)
3. 66% percent of Facebook users get “news or news headlines” from the social network (Pew Research Center)
4. 67% of consumers tap networks like Twitter and Facebook for customer service (JD Power)
5. Facebook users spend an average of 50 minutes a day on its multiple platforms (New York Times)

Social media is different from traditional approaches to marketing and communications because it's all about engagement. Instead of broadcasting information to an audience, social media enables us to connect and converse. This is a medium in which traditional approaches to "telling" people won't work or be accepted. Informing people about events, programs and news can be used, but that is just part of how these tools are used. The rest is about having a conversation. That's the "social" in social media. However, social media cannot stand apart from marketing and communications strategies, but should be incorporated as part of a holistic communications approach.

E. Impact of personalized marketing on brands and users

Marketers are more capable of gathering data and analyzing it which clearly indicates that in the coming future, hyper-personalized marketing campaigns will be the norm. Personalized marketing can be utilized in the following ways:

1. Personalized mailers
2. Retargeted ads
3. Personalized newsletters
4. Personalized content recommendations
5. Relevant product searches
6. Personalized customer support, etc.

Being a mutually beneficial strategy, this kind of marketing has a deeper impact on the brands, the way they communicate with their audiences, and the consumers themselves. Some of these benefits are:

1. Boosted social media interaction is one benefit that 56% of marketers have clearly observed as a consequence.
2. Increased rate of conversions and lead generations
3. This boosts Facebook relevance score because of the warm and positive engagement with target audience. This score impacts the price of Facebook ads and boosts ROI.

III. Implementation Of The Algorithm

To proceed, the user must fill what are their preferred topics. The system has saved the related keywords for those topics. For example, it is stored in a text file with a structure as follows.

```
technology tech advances science research
art paint draw beauty design color
music song listen stream beats rhythm
romance love flirt couple relationship dating
literature novel book poem
computer artificial intelligence informatics algorithm data
```

Fig 2. Topic Keywords Example

The system will search for the contents on the users that the users follow that contain those keywords using the Boyer-Moore Algorithm on Fig 1. The pseudocode to further explain the flow of the strategy is as follows:

```
// Array of topics
topics <- [technology, art, music, romance, literature, computer]

// An empty array of all the keywords searched

keywordSearched <- []

// Display topics to be checklisted by the user

for topic[0] in topics
    checkInput <- input()
    if(checkInput=checked) then // Checklisted by user
        for keywords in topic
            keywordSearched.append(keywords)

// Store all the keywords based on the topics

sumOfKeyword <- {} // dictionary

// Initialize as not having keywords

for i in posts
    sumOfKeyword[i] <- 0

for post in AllPosts
    for key in keywordSearched
        if BOYER_MOORE_MATCHER[post,key]!="no match" then
            postWithKeyword.append(post)
            sumOfKeyword[post]++
```

Fig 3. Application of Boyer-Moore to Match the Keyword in User's Chosen Topics and The Available Posts

Using the analytics of the social media, the system can see the average time the user is online in a day. A different approach to this is the platform may ask for user's input to how many minutes they are planning to be spending on the social media (which may not be preferable to the users).

With the average reading speed, 200 words per minute, the system can use greedy to optimize most posts with least words and most likes, most various keywords, and most interactions per minute that equates to user's online time.

The table below will show dummy example of 6 available posts, with the elements of likes, words, user's interactions with account, and keywords. Keywords are obtained from the value of the post in the dictionary sumOfKeyword.

Table 3. Dummy Example of Available Posts

Post	Likes	Words	User's Interactions with Account	Keywords
A	1000	1190	0.8	10
B	377	1235	0.9	4
C	381	756	0.1	8
D	598	394	0.2	6
E	134	1596	0.7	12
F	413	2134	0.6	7
G	860	1278	0.3	4

In this paper, we will ignore the factors of likes and user's interactions, and only use keywords to determine the posts to be showed, as the continuation of the Boyer-Moore algorithm that have been applied. This paper uses a very micro example, so say an user spends 30 minutes a day on the platform, then the goal is to optimize 30 minute multiplied by 200 words/minute (the average reading speed), that equals to 6000 words. This means that the 6000 words will be the limit of the words that the user can be exposed to.

Using a strategy equivalent to *greedy by profit* that's applied to solve the Integer Knapsack Problem, we prioritize taking the post with the most keywords/words, as long as the words limit is not exceeded.

With the strategy written in above paragraph, we can map the problem into Greedy elements:

1. Candidate set, C: posts (A to G)
2. Solution set, S: the candidates that have been chosen as a solution, that has been optimize with objective function.
3. Solution function: -
4. Selection function: choosing the maximum keywords/words from the posts that have not been chosen.
5. Feasibility function: checking whether the total words exceed the maximum (6000) or not.
6. Objective function: maximizing keywords/words

Table 4. Calculation of Keywords/Words

Post	Words	Keywords	Keywords/words
A	1190	10	0,00840336
B	1235	4	0,00323887
C	756	8	0,01058201
D	394	6	0,01522843
E	1596	12	0,0075188
F	2134	7	0,00328022
G	1278	4	0,00312989

The final calculation after applying the greedy by keywords/words strategy is as follows:

Table 5. Application of Greedy Algorithm on the Posts

Post	Words	Keywords	Keywords/Words	Greedy by Keywords/words
A	1190	10	0,00840336	1
B	1235	4	0,00323887	1
C	756	8	0,01058201	1
D	394	6	0,01522843	1
E	1596	12	0,0075188	1
F	2134	7	0,00328022	0
G	1278	4	0,00312989	0
Total words				5171
Total keywords				40

IV. Analysis Of the Solution

The use of the solution is fairly limited and simplistic, as to when an user is scrolling more than the usual time they spend, the use of the algorithm must be adjusted. The choice to merely sort the posts based on the keywords/words without limiting it to the time being spent by the user can be an alternative way to choose.

A different approach can also be used based on the prioritized goal. For example, if the social media prioritized user's familiarity, a greedy by user's interactions per words approach may be preferred. Similarly, if the prioritized goal is the popularity of the post, greedy by likes per words can be another alternative.

The usage of the greedy strategy can also be developed with other techniques like Dynamic Programming that may offer more optimum solution to Integer Knapsack Problem.

The strategy also lacks the ability to count for keywords more than once (for example, even if the keyword 'technology' occurs more than once in a post, it will be counted as 1 because the Boyer-Moore algorithm looks for the first solution). Because only different keywords are being counted, this approach heavily relies on the variation of the keywords. A modification of the algorithm, to suit this case better may be implemented.

V. Conclusion

The human civilization has always used words to express their needs or emotions. The usage of words is not limited to the social media problem that is being proposed in this paper, but any other human living aspect as well. The Boyer-Moore algorithm stays relevant as the usage of words lives on. That being said, human needs to optimization always applies, due to the limits of every fulfilling things despite the huge necessities in every aspect. Social media personalization is only one example of such optimization, and Greedy algorithm proves worthy to be applied for the problem. Despite of the weaknesses and micro scale of the strategy, this solution can be used as a simple and eased way of social media optimization.

Video Link at YouTube

The video for this paper is available on www.youtube.com/watch?v=YOksW4oI6o0.

Acknowledgment

I offer my deepest gratitudes to God for giving me the strength to endure everything in life. He has gifted me everyone I love and admire to keep me motivated, those whom I want to thank as well. Last, but not least, my thanking also goes to my lecturer Ir. Rila Mandala, M.Eng., Ph.D. for his guidance this semester.

References

[1] https://web.stanford.edu/class/archive/cs/cs161/cs161.1138/lectures/13/Small13.pdf Accessed on 8 May 2021, 14.36 WIB.

[2] https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2019-2020/Algoritma-Greedy-(2020).pdf. Accessed on 8 May 2021, 14.38 WIB.

[3] https://mention.com/en/blog/personalization-social-media-marketing. Accessed on 8 May 2021, 14.53 WIB.

[4] https://ucpa.ucsd.edu/resources/social/social-101. Accessed on 8 May 2021, 14.59 WIB.

[5] http://personal.kent.edu/~rmuhamma/Algorithms/MyAlgorithms/StringMatch/boyerMoore.htm. Accessed on 8 May 2021, 15.06 WIB.

[6] https://swaminathanj.github.io/algo/12_Integer_Knapsack.html. Accessed on 8 May 2021, 21.01 WIB.

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Palembang, 8 Mei 2021

Rhea Elka Pandumpi 13519047