ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 3, Issue 4, July 2014

Rank Aggregation of Search Engine Results

Sanchit Sapra

Abstract*— Rank Aggregation is the problem of taking some number of ranking lists as input and then combining them to generate the resultant ranking list according to some algorithm. When the aggregate of two numbers is found by using their arithmetic mean this computed mean number is equidistant from the two numbers. Similarly the aggregated list is a permutation of the members of the list such that it is 'equidistant' from the input lists. To calculate the distance between two lists the Spearman Footrule Distance (SFD) has been used in this work. The method proposed in this work uses the value of the arithmetic mean and the standard deviation (SD) of the position of all web pages occurring in the list and tries to optimize the result by using Bisection Method. Five search engines have been used for the aggregation and the results are compared with some of the earlier methods of aggregation.

***Index Terms* —Bisection Method, Fuzzy Membership Function, Mean-By-Variance Technique, NP Hard, Partial List.**

I. INTRODUCTION

Rank Aggregation is the problem of combining or collating a set of lists to form a resulting list which is in 'consensuses with a set of input lists. The consensus is achieved with the help of some algorithms which are to be applied on the input lists. Like for calculating the aggregate percentage of marks obtained by a student we require the marks obtained in the individual subjects and the corresponding credits associated with them and then the weighted mean is computed to give the result. So analogously in the problem of *rank aggregation* we are given a set of lists which are different permutations of some members of a set and the aim is to come up with a list which is a new permutation such that it satisfies some criterion with respect to the Spearman Footrule Distance (SFD) between the new list and the set of input lists. The different algorithms mentioned in the further sections of this paper give different weightage or 'importance' to the input lists.

II. BACKGROUND AND RELATED WORK

A. Definitions

Definition 1 Given a universal list U and T $\subseteq$ U, an ordered list (or simply, a list) l with respect to U is given as $l=[d_1,d_2,d_3\ldots d_{|T|}]$ with each d_i belonging to T and let l(i) denote the position or rank of element i, with a higher rank having a lower numbered position in the list.

Definition 2 Full List: If a list L contains all the elements present in the universal list U then it is said to be a full list.

Definition 3 Partial List: A list l containing elements, which are a strict subset of U, is called a partial list. We have a strict inequality $|l|<|U|$.

Definition 4 Kendall Tau distance: The *Kendall tau distance* between two full lists l_1 and l_2, each of cardinality $|l|$, is given as follows.

$$K(l_1, l_2) = \frac{\left|\left\{(i, j) \mid \forall l_1(i) < l_1(j), l_2(i) > l_2(j)\right\}\right|}{(1/2)|l|(|l|-1)} \quad (1)$$

Definition 5 Spearman foot rule distance**:** The *Spearman footrule distance* (SFD) between two full lists l_1 and l_2, each of cardinality $|l|$, is given as follows.

ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 3, Issue 4, July 2014

$$F(l_1, l_2) = \frac{\sum_{\forall i} |l_1(i) - l_2(i)|}{\left\lfloor \left(\frac{1}{2}\right) l|^2 \right\rfloor} \quad (2)$$

Definition 6 Given a set of k full lists as $L = \{l_1, l_2 \ldots l_k\}$, the *normalized aggregated Kendall distance* of a full list l to the set of full lists L is given as

$K(l, L) = \frac{\sum_{i=1}^{k} K(l, l_i)}{k}$, while the *normalized aggregated spearman footrule distance* of l to L is given as

$$F(l, L) = \frac{\sum_{i=1}^{k} F(l, l_i)}{k} \quad (3)$$

Definition 7 Rank Aggregation**:** Given a set of lists $L = \{l_1, l_2 \ldots l_k\}$, *Rank Aggregation* is the task of coming up with a list l such that either $K(l, L)$ or $F(l, L)$ is minimized.

The rank aggregation obtained by optimizing the Kendall distance is called Kemeny optimal aggregation (KOA), and it has been shown in [2] that KOA is NP-hard even when k=4.Therefore spearman footrule distance(SFD) has been used in this work as a parameter for rank aggregation.

B. Borda's Method of Rank Aggregation

Given k lists $l_1, l_2 \ldots l_k$, for each candidate c_j in list l_i, we assign a score $S_i(c_j) = |c_p : l_i(c_p) > l_i(c_j)|$. The candidates are then sorted in a decreasing order of the total Borda score $S(c_j) = \sum_{i=1}^{k} S_i(c_j)$.

Example Given lists $l_1 = [c, d, b, a, e]$ and $l_2 = [b, d, e, c, a]$, the aggregated rank using Borda's method may be obtained as follows.

$S_1(a) = |e| = 1$, as $l_1(e) = 5 > l_1(a) = 4$.

Similarly,

$S_1(b) = |a, e| = 2$, as $l_1(e) = 5 > l_1(b) = 3$ and $l_1(a) = 4 > l_1(b) = 3$.

Proceeding this way, we get

$S_1(c) = |a, b, d, e| = 4$,
$S_1(d) = |a, b, e| = 3$,
$S_1(e) = | | = 0$,
$S_2(a) = | | = 0$,
$S_2(b) = |a, c, d, e| = 4$,
$S_2(c) = |a| = 1$,
$S_2(d) = |a, c, e| = 3$,
$S_2(e) = |a, c| = 2$,
$S(a) = S_1(a) + S_2(a) = 1+0 = 1$,
$S(b) = S_1(b) + S_2(b) = 2+4 = 6$,
$S(c) = S_1(c) + S_2(c) = 4+1 = 5$,
$S(d) = S_1(d) + S_2(d) = 3+3 = 6$,
$S(e) = S_1(e) + S_2(e) = 0+2 = 2$.

Now, sorting the elements based on their total scores, we get the combined ranking as $b \approx d \succ c \succ e \succ a$. The '$\approx$' symbol indicates a tie.

ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 3, Issue 4, July 2014

C. Shimura technique of rank aggregation

For variables x_i and x_j defined on universe X, a relativity function $f(x_i|x_j)$ is taken to be the membership of preferring x_i over x_j. This function is given as

$$f\left(x_i \middle| x_j\right) = \frac{f_{x_j}(x_i)}{\max\left(f_{x_j}(x_i), f_{x_i}(x_j)\right)}$$

where, $f_{x_j}(x_i)$ is the membership function of x_i with respect to x_j, and $f_{x_i}(x_j)$ is the membership function of x_j with respect to x_i.

For $X = [x_1, x_2, ..., x_n]$, $f_{x_i}(x_i) = 1$.

$C_i = \min_{j=1}^{n} f\left(x_i \middle| x_j\right)$ is the membership ranking value for the i^{th} element. Now if a descending sort on C_i (i=1 to n) is carried out, the sequence of i's thus obtained would constitute the aggregated rank.

For the lists l_1, l_2… l_N from the N participating search engines, we can have

$$f_{x_j}(x_i) = \frac{\left|k \in [1, N] \vee l_k(x_i) < l_k(x_j)\right|}{N}$$

Example Given l_1=[3,4,2,1], l_2=[2,4,3,1] and l_3=[4,2,1,3]

$$f_{x_i}(x_j) = \begin{array}{c} \\ i \downarrow \end{array} \begin{array}{c} \begin{array}{cccc} j & \rightarrow & & \end{array} \\ \begin{bmatrix} 1 & 0 & 0.33 & 0 \\ 1 & 1 & 0.67 & 0.33 \\ 0.67 & 0.33 & 1 & 0.33 \\ 1 & 0.67 & 0.67 & 1 \end{bmatrix} \end{array}$$

$$f\left(x_i \middle| x_j\right) = \begin{array}{c} \\ i \downarrow \end{array} \begin{array}{c} \begin{array}{cccc} j & \rightarrow & & \end{array} \\ \begin{bmatrix} 1 & 0 & 0.5 & 0 \\ 1 & 1 & 1 & 0.5 \\ 1 & 0.5 & 1 & 0.5 \\ 1 & 1 & 1 & 1 \end{bmatrix} \end{array} \Rightarrow C_i = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{bmatrix} 0 \\ 0.5 \\ 0.5 \\ 1 \end{bmatrix}$$

A descending sort on C_i gives the aggregated rank as either l=[4,3,2,1] or l=[4,2,3,1].

D. Membership Function Ordering (MFO) Technique of Rank Aggregation

In this method we first find out the value of the mean position of each document calculated over all the lists. We also take the variance of the document positions into account. With the mean $\left(\bar{x}_{d_i}\right)$ and variance $\left(\bar{\sigma}_{d_i}^{\,2}\right)$ of the position of document d_i known, a Gaussian sub-normal fuzzy membership function is obtained as

$$\underset{\sim}{\mu}_{d_i}(x) = \frac{1}{\sqrt{2\pi\sigma_{d_i}^2}} \exp\left(-\frac{1}{2}\left[\frac{\left(x - \bar{x}_{d_i}\right)^2}{\sigma_{d_i}^2}\right]\right)$$

With the membership function of each document obtained as a function of positions, the amplitude of the membership function of each document at each position is evaluated. The document having the highest membership value at a given position is assigned to that position. This way, the documents are arranged in the

ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 3, Issue 4, July 2014

aggregated ranking. We name this technique as membership function ordering (MFO). For example from Fig.1 given below, we can see that out of the three documents whose membership functions are sketched, the first position has the maximum amplitude of document 1, and so document 1 must occur ahead of the rest two. Out of the remaining documents 2 and 3, the second position has the maximum amplitude of document 2, and so document 2 must be preferred over document 3 to give the ranking as

document 1 > document 2 > document 3.

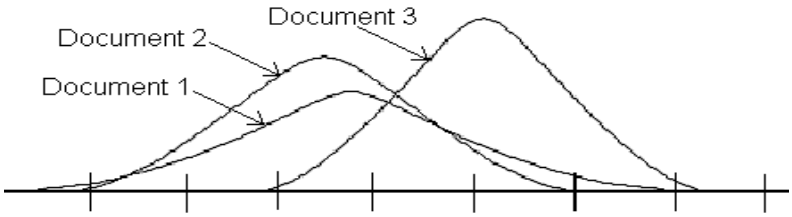

Fig.1 Logic behind MFO technique

E. Mean-By-Variance (MBV) technique of Rank Aggregation

It may be noted that if the variance of the position of two documents is same, then the document having a lesser mean position must be ranked first (see Fig.2.1). Conversely, if the mean position of two documents is same, then the document having a larger variance of position must be ranked first (see Fig.2.2). With this logic, the Mean-By-Variance (MBV) heuristic is proposed for rank aggregation. First, the ratio $mbv(i) = \left(\bar{x}_{d_i} \Big/ \bar{\sigma}_{d_i}{}^2 \right)$ is calculated for all the members of the list. Where X_d denotes mean and $\bar{\sigma}_{d_i}{}^2$ denotes the variance. An ascending sort on the set of these fractions would give the aggregated list *l*.

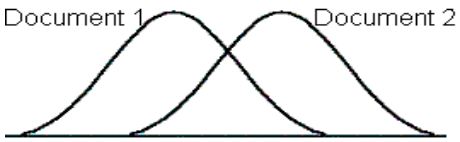

Fig 2.1 Different Mean, Same Variance

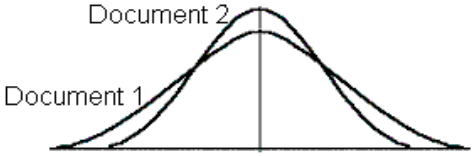

Fig.2.2 Same Mean,Different Variance

F. Modified Shimura Technique of Rank Aggregation

Since $C_i = \min_{j=1}^{n} f\left(x_i \middle| x_j\right)$ was used to find the final value of an element in the list the "min" function results in many ties, when a descending order sort is applied on C_i. There is no method suggested by Shimura to resolve these ties. So when resolved arbitrarily, these ties result in deterioration of the aggregated result. So therefore M. M. S. Beg, N. Ahmad in [4] proposed to replace this "min" function by an OWA operator [5]. The OWA operators provide a parameterized family of aggregation operators, which include many of the well-known operators such as the maximum, the minimum, arithmetic mean.

G. Genetic Algorithm for Rank Aggregation

Beg and Ahmad in [3] proposed a genetic algorithm based rank aggregation method with the following steps

1 Select some chromosomes.
2 Test the fitness of all chromosomes and discard the worse half of them.
3 Crossover ¼ of chromosome with other ¼ chromosome to get half of new chromosomes.
4 Repeat step 1 to 3 iteratively.

'Objective Function' used is the *normalized aggregated footrule distance* $F(l, L) = \dfrac{\sum_{i=1}^{k} F(l, l_i)}{k}$ .Crossover is done by multiplication of two permutations as illustrated by the following two instances:

ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 3, Issue 4, July 2014

{3, 1, 2} × {2, 3, 1} = {1, 2, 3},
{5, 2, 4, 3} × {2, 4, 3, 1} = {2, 3, 4, 5}
Mutation is carried out at a rate of 0.005/digit/generation. The digit to be mutated has to be changed with the restriction that the resulting chromosome remains a valid permutation. For this, the to-be-mutated digit is exchanged with any other randomly selected digit in that very permutation.

III. BISECTION METHOD

The Bisection method is a method used for finding the root of a non linear equation in the field of numerical computing. It repeatedly bisects an interval and then returns a subinterval in which a root must lie for further processing. It is a very simple and robust method. This method is based on the application of Intermediate Value Theorem.

Let the function f(x) be continuous between points a and b such that
f(a).f(b)<0 which means that the value of f(a) and f(b) should be of opposite signs. Then the first approximation of the root is X_1 =1/2(a+b) as shown in Fig.3.

If $f(x_1)$ =0 then x_1 is a root of f(x) =0.Otherwise the root lies between a and x_1 or x_1 and b depending on whether $f(x_1)$ is positive or negative. Then we bisect the interval as before and continue the process until the root is found to desired accuracy.

Since the new interval containing the root is exactly half the length of the previous one, the interval width is reduced by a factor of ½ at each step and therefore the method converges towards the result we want.

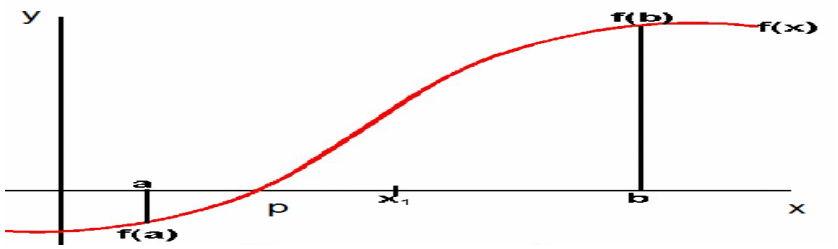

Fig.3 Bisection method

IV. PROPOSED METHOD

A. Motivation

The Mean-By-Variance (MBV) technique of Rank Aggregation is the main motivation behind my work. As it is mentioned in this method that the rank of a page in the aggregated list is proportional to the value of the arithmetic mean of the positions of that page in the set of input lists and inversely related to the variance of the positions of that page in the input lists so my main idea is that since variance is a very large numerical value so standard deviation should be used in place of that. Since variance is a very large value therefore it has a big negative impact on the $\left(\bar{x}_{d_i} / \bar{\sigma}_{d_i}^{\,2}\right)$ ratio. I personally feel that by using a denominator (which is variance) of very high value we are penalizing the rank of a page too much.

With this point in mind my aim was to find a function of mean and standard deviation according to which the aggregated list would be found and this list must have minimum value of normalized aggregated SFD (3).

Intuitively I decided that the general formula of the function to be found would be M-x*SD,
where M denotes mean and SD denotes Standard Deviation.
There are various other forms of general formula which can be used like
$M-(SD)^X$ or $M/(SD)^X$.

ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 3, Issue 4, July 2014

It can be easily seen that MBV is a special case of $M/(SD)^X$ where X=2.

The next question is how to find the value of variable X which gives the optimal result according to the normalized aggregated spearman footrule distance. Since there are infinite values of X and we cannot find the optimal value in a haphazard way so here we use the *Bisection Method* mentioned earlier.

For finding the root of an equation using Bisection method we start with two values of x, a and b. Similarly for finding the value of X that gives us the required function of mean and standard deviation I have started with values of 1 and 2 for X. So therefore my function initially becomes M-SD and M-2*SD. On the basis of these two functions the aggregated list is formed and the normalized SFD using (3) is calculated. On comparison two cases can exist

1) If SFD using X=1 < SFD using X=2, therefore next value of X is 1.5
2) If SFD using X=1 > SFD using X=2 therefore check SFD for X=3.

The interval between the values of X gets reduced by a factor of ½ in each iteration so I have stopped the iterations when the interval becomes 0.01.Another point to mention is that after the iterations have stopped and an optimum value of X has been obtained I have checked the results for those values of X which are in its neighbourhood.I have assumed neighbourhood to be ± 0.02.
In this way different values of X are used and the optimum value of X is found. After finding the optimum value of X the aggregation is performed according to that heuristic formula which is M-x*SD. The normalized SFD calculated on this list is compared with that of the MBV technique to draw a conclusion. So in short the role of bisection method is to converge towards the optimum value of X

B. Algorithm

1. Collect the top 20 results of five search engines for a keyword.

2. Take the union operation of all these web pages to form a universal set U.

3. Assign integer identifier 1, 2, 3… to each unique web page present in the set U.

4. Convert the partial list of the result of all individual search engines into full lists.

5. Determine the position of each web page in the list of 5 search engines. For web page not present in a particular list assign a specially calculated value to its position depending on the size of set U.

6. Calculate mean, variance and standard deviation of the values of the position of each web page.

7. Calculate Borda score of each page in set U and then rank them according to it. Let the resulting aggregated list be denoted by l_b.

8. Calculate the SFD using (2) between list of each search engine rankings and l_b. Calculate the normalized aggregated SFD (3).

9 Perform rank aggregation using the concept of MBV and calculate the average SFD (3) associated with the resulting aggregated list.

10. Find the optimum value of X and perform rank aggregation according to that list. Calculate the normalized SFD (3) and compare its result with that of Borda's and MBV technique.

Points 4 and 5 have been explained in the sections C and D respectively.

C. Partial list to Full list conversion

The biggest challenge in the task of rank aggregation is to solve the problem of partial lists. In [1] it is mentioned that the problem of rank aggregation of partial list is NP hard in nature. In this work I have taken the top 20 results

ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 3, Issue 4, July 2014

of 5 search engines for aggregation and since all of them use different algorithm for ranking the pages, the results of the 5 search engines are different due to which my input for aggregation is a set of 5 partial lists. So my first concern is to convert all these partial lists into a full list. There are various methods to convert a partial list to a full list used by different authors worldwide but the method that I have used is motivated and quite similar to that used by Beg and Ahmad [1].

I have taken the top 20 results of 5 search engines which mean that there are a total of 100 web pages. Out of these 100 some web pages may be present in the list of 2 or more search engines so total number of unique web pages out of these 100 would be less than 100.For converting the partial list to full my logic is suppose if there are 40 unique pages so the full list formed by taking union of 5 lists has 40 pages.

Each search engine gives 20 results and to convert it into full list we have to add 20 more pages to make it 40.To do so I have found the arithmetic mean of the identifiers of the pages not present in the result of the given search engine and denote it by 'a'. To complete the given partial list into full list I have augmented this value 'a' 20 times to the original list to make it a full list of 40 pages.

While calculating the numerator of the formula of SFD (2) between an aggregated list and the list returned by a search engine the above concept is used in the following manner. For calculating the difference between the position of pages which are present in the list of search engine result its position is found from the list and for calculating the difference for those pages which are not present in the search engine list its position in the list is taken as value 'a' calculated earlier. To illustrate my logic let me use an example.

Suppose a full list A is {1, 4, 6, 7, 9, 2, 12}. Therefore |A|=7

And a partial list B is {12, 7, 4} so |B|=3

To convert B into a full list
Arithmetic mean of identifiers of missing pages = (1+6+9+2)/4=4.5='a'

So B is {12, 7, 4, 4.5, 4.5, 4.5, 4.5}.And the position of pages 1,6,9,2 in list B is 4.5.

D. Computing Value of Position of pages not present in a list

As mentioned in step 5 we need to find the position of all web pages present in the Universal set U in the lists returned by the search engines employed. So the question is that if a page with identifier say 5 is not present in the list returned by the first search engine what value should be given to it. For this I have used the concept of probability. Suppose there are 70 different pages in the universal set U formed by taking the union of result of all search engines and if page 5 is not present in the top 20 results returned by a search engine then I have assumed that the probability of occurrence of this page 5 is equal at all positions from 21 to 70. So equivalently we can say that the position of page 5 in this list is the arithmetic mean of the numbers in the list 21, 22, 23, 24…….70.
We know that the mean of the above list of numbers is 45.5 and for avoiding complicated calculations I have approximated it to 45. This value of position of pages is used for calculating the mean, standard deviation and variance in the next step of the algorithm.

V. EXPERIMENTS AND RESULTS

The five search engines that I have used in this work are Google, Yahoo, Lycos, Excite and Exalead. As mentioned previously the top 20 results of each search engine are used. The keywords used in this paper are part of the list of words that were used in [1] so that it is easy to compare the results. First keyword used is "parallel architecture" The results for the method used and the value of X in the formula proposed are shown in table I.

Table I parallel architecture result

Method Used	Normalized Aggregated SFD
Arithmetic mean of position	0.438894
MBV	0.446659

ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 3, Issue 4, July 2014

Mean/standard deviation	0.437851
Mean-sd	0.435537
Mean-(1.5*sd)	0.436272
Mean-(1.25*sd)	0.435721
Mean-(1.13*sd)	0.435721
Mean-(1.07*sd)	0.435354
Mean-(0.9*sd)	0.435537
Mean-(0.7*sd)	0.435966
Mean-(0.95*sd)	0.435537
Mean-(1.03*sd)	0.435537
Mean-(1.04*sd)	0.435354(best case)
Mean-(1.05*sd)	0.435354
Mean-(1.06*sd)	0.435354
Mean-(1.10*sd)	0.435721
Mean-(1.08*sd)	0.435721
Mean-(1.09*sd)	0.435721
Mean-(1.11*sd)	0.435721
Mean-(1.12*sd)	0.435721

So the minimum value of normalized SFD is 0.435354 given by mean-(x*standard deviation) for x belonging to interval [1.04, 1.07]. The progress of the algorithm in iterations can be seen with the graph in Fig.4 shown on next page.

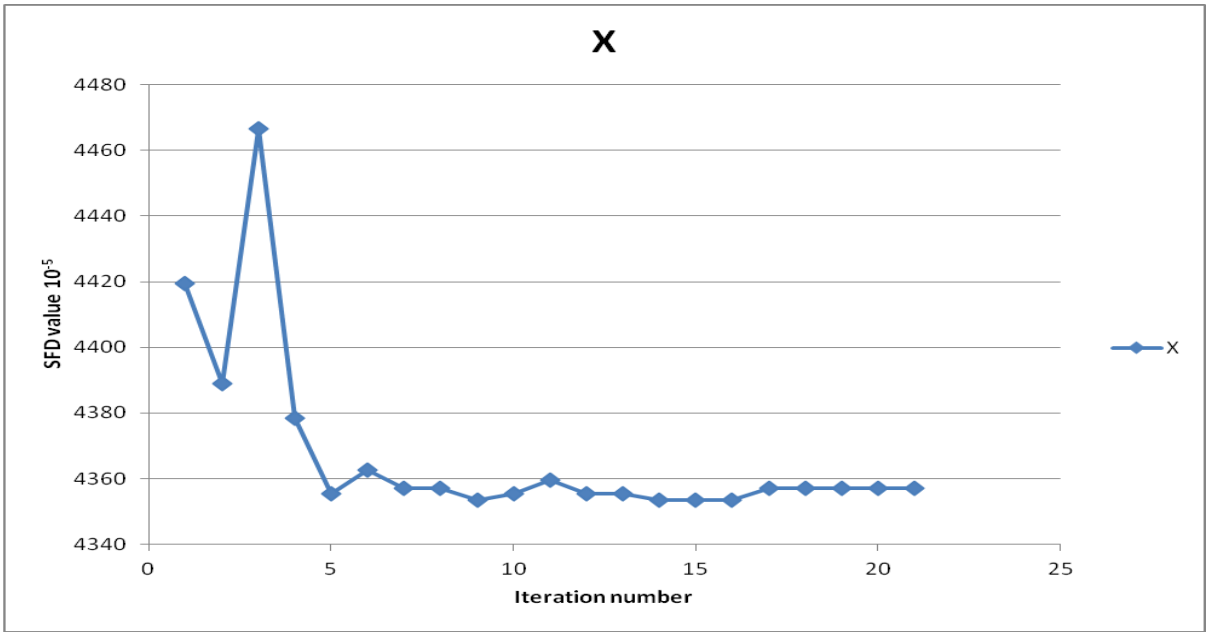

Fig.4 Results for "parallel architecture"

The top few results for the query "parallel architecture corresponding to the optimal value of coefficient of SD are

1 en.wikipedia.org/wiki/Parallel_computing

2 parallel-architecture.com

3 www.cs.cmu.edu/afs/cs/academics/class/15740-f03/..../lect08.4up.pdf

4 aparallel.com

ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 3, Issue 4, July 2014

5 books.google.co.in/books/about/Parallel_computer

6 computing.llnl.gov/tutorials/parallel_comp

7 williams.comp.ncat.edu/COMP375/Parallel Arch.pdf

8 www.cs.berkeley.edu/~culler/cs258-s99/

9 parallelarchitecture.com

10 public.callutheran.edu/~reinhart/CS521MSCS/…./FlynnTaxonomies.pdf

The second keyword used was "Citrus Groves". Its results are shown in table II.

Table II Citrus groves result

Method used	Normalized aggregated SFD
Borda's Score	0.4661214
Modified Borda's Score	0.4614264
Mean of position in list	0.4470662
MBV	0.4519972
Mean/sd	0.4456708
Mean – sd	0.445547
Mean - (2*sd)	0.4452272
Mean - (3*sd)	0.451713
Mean - (1.5*sd)	0.4447308
Mean - (0.5*sd)	0.4446442
Mean - (0.25*sd)	0.4454228
Mean - (0.75*sd)	0.4443846
Mean - (0.87*sd)	0.444159
Mean - (0.93*sd)	0.4442492
Mean - (0.81*sd)	0.4435196
Mean - (0.78*sd)	0.4440386
Mean - (0.795*sd)	0.4435196
Mean - (0.8025*sd)	0.4435196
Mean - (1.25*sd)	0.4449412
Mean - (1.37*sd)	0.4402326 (best case)
Mean - (1.44*sd)	0.4404056
Mean - (1.40*sd)	0.4445576
Mean - (1.42*sd)	0.4445576
Mean - (1.385*sd)	0.4445576
Mean - (1.41*sd)	0.4445576
Mean - (1.38*sd)	0.4445576
Mean - (1.39*sd)	0.4445576

So the minimum value of normalized SFD is 0.4402326 given by mean-(1.37*standard deviation)

The graph corresponding to the above table is shown in Fig.5

ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 3, Issue 4, July 2014

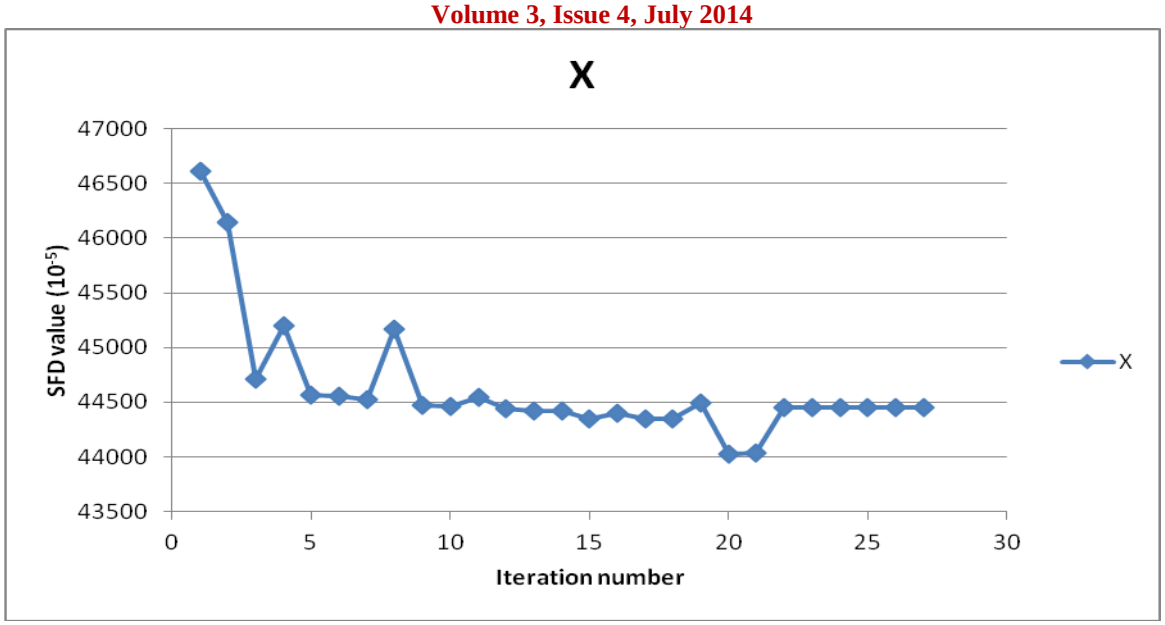

Fig. 5 Results for "citrus groves"

The top few result of the query "citrus groves" for the optimum value of M-1.37*SD are

1. http://www.visitflorida.com/en-us/articles/2007/november/747-visit-a-florida-citrus-grove.html

2 http://en.wikipedia.org/wiki/Citrus

3 http://www.floridajuice.com/visit-a-grove

4 http://articles.orlandosentinel.com/keyword/citrus-groves

5 http://www.citrusgroveapthomes.com/

6 http://www.pbchistoryonline.org/page/citrus-groves

7 citrusgroves.net

8 http://www.yellowpages.com/orlando-fl/citrus-groves

9 www.halegroves.com

10 http://www.floridajuice.com/florida-citrus

The third keyword used was "affirmative action"Its result are shown in table III

Table III Results of "affirmative action"

Method used	Normalized Aggregated SFD
Borda's method	0.456379
Modified Borda's method	0.459023

ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 3, Issue 4, July 2014

Mean - (1.0*sd)	0.4593104
Mean - (2.0*sd)	0.4593104
Mean - (3.0*sd)	0.4595976
Mean - (2.5*sd)	0.4590804
Mean - (1.5*sd)	0.4602298
Mean - (0.5*sd)	0.4628354
Mean - (2.25*sd)	0.4595404
Mean - (2.37*sd)	0.4593102
Mean - (2.44*sd)	0.4593102
Mean - (2.40*sd)	0.4593102
Mean - (2.31*sd)	0.4593102
Mean	0.4667432
Mean - (2.75*sd)	0.4600574
Mean - (2.62*sd)	0.4591378
Mean - (2,56*sd)	0.458908 (best case)
Mean - (2.59*sd)	0.458908
Mean - (2.53*sd)	0.4590804
Mean - (2.16*sd)	0.4595404
Mean - (2.08*sd)	0.4600002
Mean - (2.28*sd)	0.4595404
Mean - (2.04*sd)	0.4593104
Mean - (2.20*sd)	0.4595404
Mean - (2.12*sd)	0.4597704
MBV	0.463965

The graph corresponding to the above values is shown in Fig.6

ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 3, Issue 4, July 2014

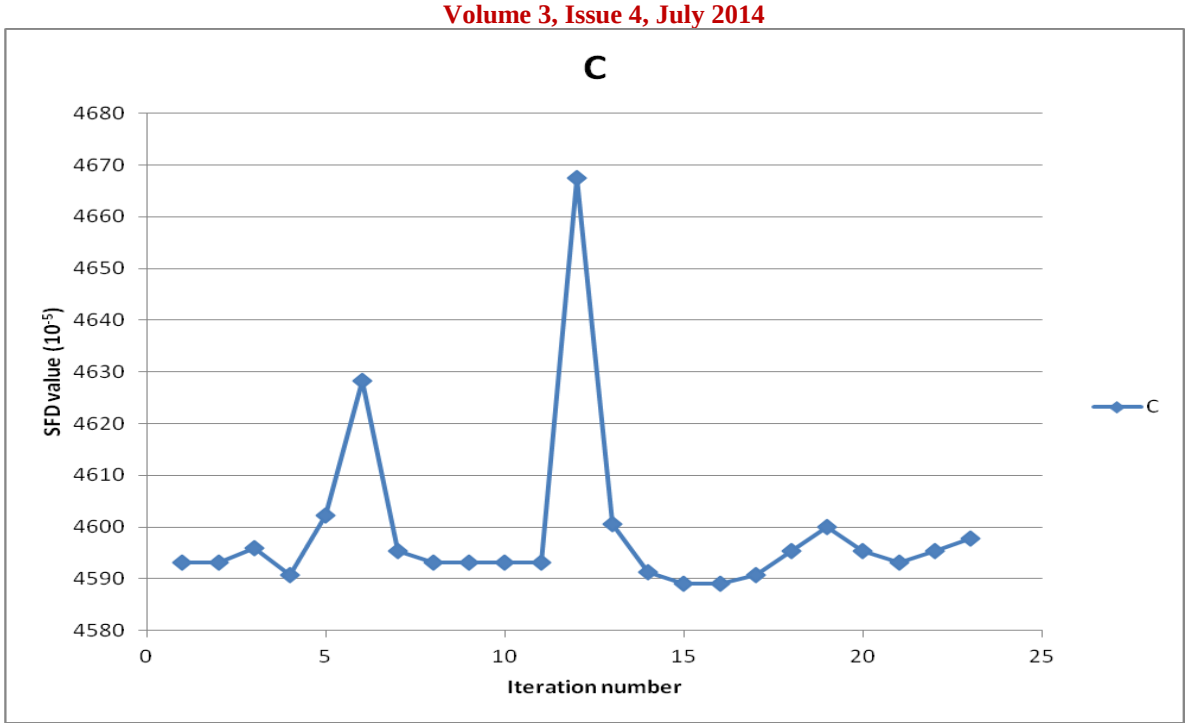

Fig.6 Results for "affirmative action"

It can be observed from the readings of table III that the best value of SFD (3), which is for mean – (2.56*standard deviation) is 0.458908 but it is still worse than the SFD by Borda's method. So therefore I tried another approach to minimize the value of SFD by using general equation $M/(SD)^X$ in place of M-(x*SD) with the aim to get a result which is better than that of Borda's method. The results of this approach are shown in table IV.

Table IV Second method for "affirmative action"

Value of X for mean/$(sd)^X$	Normalized aggregated SFD value
2	0.463965
1	0.4588508
0	0.4667432
0.5	0.4593104
1.5	0.4591378
1.25	0.4593678
1.12	0.4586208 (best case)
1.06	0.4588508
1.09	0.4586208
1.10	0.4586208
1.11	0.4586208

ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 3, Issue 4, July 2014

1.03	0.4588508
1.01	0.4588508
1.02	0.4588508

The graph corresponding to the above table is shown in Fig.7 on next page

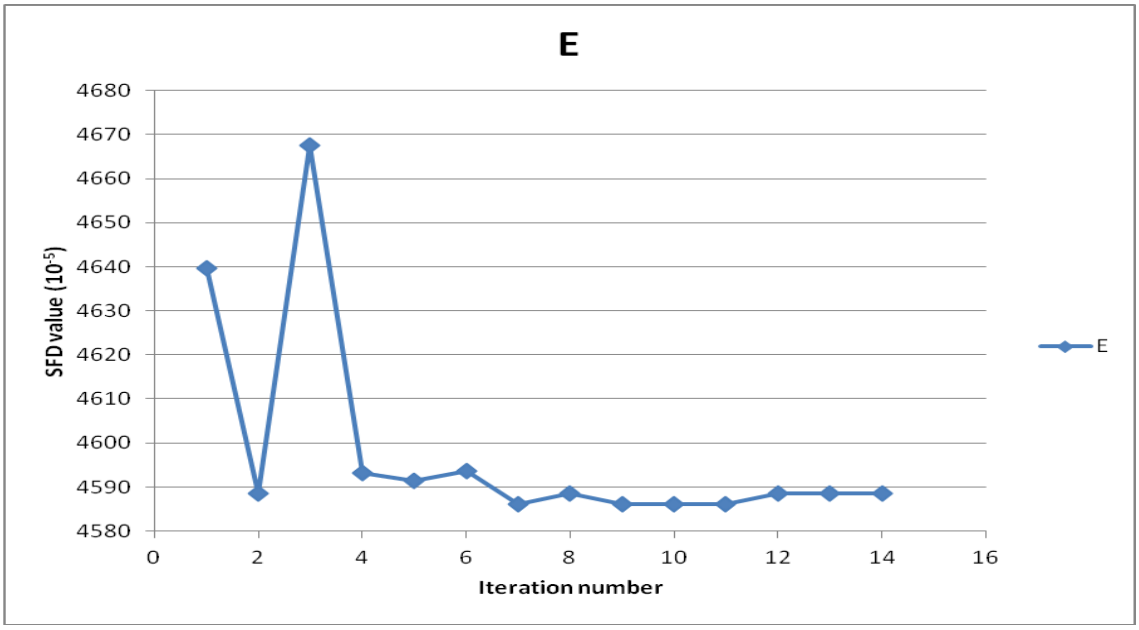

Fig.7 Results for "affirmative action" using second method

To compare the results of the 2 approaches used the values are plotted on the same graph as shown in Fig.8

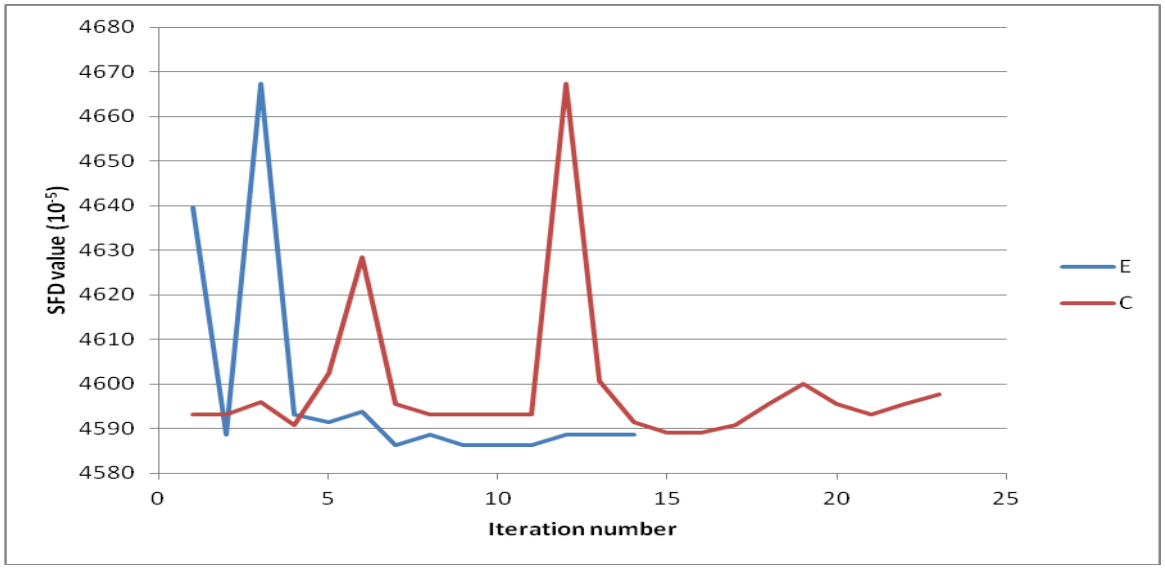

Fig.8 Comparison of method 1 and 2 for "affirmative action"

ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 3, Issue 4, July 2014

So after comparison the minimum value of normalized SFD (3) is 0.4586208 given by the second approach i.e. $M/(SD)^X$ for x=1.12.

The top few results for query "affirmative action" corresponding to the minimum value of SFD (3) are

1 http://en.wikipedia.org/wiki/Affirmative_action

2 http://plato.stanford.edu/entries/affirmative-action

3 http://dictionary.reference.com/browse/affirmative+action

4 http://legal-dictionary.thefreedictionary.com/Affirmative+Action

5 http://en.wikipedia.org/wiki/Affirmative_action_in_the_United_States

6 http://www.u-s-history.com/pages/h1970.html

7 http://www.affirmativeaction.org

8 http://www.newworldencyclopedia.org/entry/Affirmative_action

9 http://www.ciiaffirmativeaction.in

10 http://www.merriam-webster.com/dictionary/affirmative%20action

VI. CONCLUSION

The normalized SFD for rank aggregation performed on the result of 5 general purpose search engines was computed using Borda's method, MBV method and according to the values calculated using a function of arithmetic mean and standard deviation of the position of web pages. On comparison of the values of SFD it can be concluded that by using standard deviation instead of variance the aggregation performed is better. Hence the proposal to replace variance by Standard deviation in the MBV is correct and it leads us to a new direction for performing rank aggregation combining the field of Numerical Analysis with Web Mining.

At times the Bisection method can fail to converge towards the optimum value of variable X and also it has a slow rate of convergence. In future work the aim should be to use some other method which converges towards the optimum value more efficiently and at a faster rate.

REFERENCES

[1] M. M. S. Beg, N. Ahmad, "Fuzzy Logic and Rank Aggregation for the World Wide Web " In "Fuzzy Logic and the Internet, Studies in Fuzziness and Soft Computing" 2002

[2] Dwork C, Kumar R, Naor M, Sivakumar D (2001) "Rank aggregation methods for the web". Proceedings of the Tenth World Wide Web Conference. Hong Kong.

[3] M. M. S. Beg, N. Ahmad, "Soft Computing Techniques for Rank Aggregation on the World Wide Web", World Wide Web – An International Journal, Kluwer, vol. 6, issue 1, March 2003, pp. 5-22.

[4] M. M. S. Beg, N. Ahmad, "Improved Shimura technique for Rank Aggregation on the World Wide Web," Proc. 5th International Conference on Information Technology (CIT 2002), Bhubaneswar, India, December, 21-24, 2002.

[5] R. R. Yager, "*On Ordered Weighted Averaging Aggregation Operators in Multicriteria Decision Making*," IEEE Trans. Systems, Man and Cybernetics, vol. 18, no. 1, January/February 1988.

[6] Nir Ailon, "Aggregation of Partial Rankings, *p*-Ratings and Top-*m* Lists"Algorithmica June 2010, Volume 57, Issue 2, pp 284-300

[7] Dwork C, Kumar R, Naor M, Sivakumar D (2001) "Rank aggregation methods revisited ".

ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 3, Issue 4, July 2014

[8] M. M. S. Beg, "A Subjective Measure of Web Search Quality," International Journal of Information Sciences, Elsevier, vol. 169, no. 3-4, 2005, pp. 365-381.

[9] https://www.google.co.in

[10] http://in.search.yahoo.com

[11] http://www.lycos.in

[12] http://msxml.excite.com

[13] http://www.exalead.com/search

AUTHOR BIOGRAPHY

Sanchit Sapra obtained his B.Tech (Computer Engineering) from Jamia Millia Islamia, New Delhi. Currently he is pursuing M.Tech (Computer Science and Engineering) from Guru Gobind Singh Indraprastha University, New Delhi. His research interests are Web Mining, Analysis of Algorithms and Optimization Techniques.