



HAL
open science

Data Fusion in Information Retrieval Using Consensus Aggregation Operators

Julien Ah-Pine

► **To cite this version:**

Julien Ah-Pine. Data Fusion in Information Retrieval Using Consensus Aggregation Operators. 2008 IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT-08) , Dec 2008, Sydney, Australia. pp.662-668, 10.1109/WIIAT.2008.146 . hal-01504437

HAL Id: hal-01504437

<https://hal.science/hal-01504437>

Submitted on 10 Apr 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Data fusion in information retrieval using consensus aggregation operators

Julien Ah-Pine
Xerox Research Centre Europe
6, chemin de Maupertuis
38240 Meylan, France
julien.ah-pine@xrce.xerox.com

Abstract

In this paper, we address the problem of unsupervised rank aggregation in the context of meta-searching in information retrieval field. The first goal of this paper is to apply aggregation operators that are defined in information fusion domain to the particular issue mentioned beforehand. Triangular norms, conorms and quasi-arithmetic means, are such kind of operators. Then, the second goal of this work is to introduce a new aggregation function, its logical foundations and its combinatorial properties. Particularly, this operator allows to take into account the relationships between experts in a flexible way. Finally, we test these different aggregation operators on the LETOR dataset. The results of our experiments show that this kind of aggregation functions can lead to better results than baseline methods such as CombSUM and CombMNZ approaches.

1. Introduction

Data fusion problems arise when for a given task, there are several experts which give different opinions. These opinions can agree or not and we are faced with the problem of combining them in order to figure out a consensus decision. That kind of issue can be encountered in different contexts: in voting theory, we have to aggregate the votes or preferences of many individuals in order to elect a consensus candidate; in decision making we have several alternatives and we have to choose the most appropriate one according to several criteria. Aggregating preference relations is an issue that has been analyzed in social sciences [7], [2], [11] and in statistics [14], [19].

More recently, in information retrieval, tasks such as meta-search problems aim at combining the results given by different search engines into a single consensus list of web pages which is expected to be more robust and efficient than if we take into account only one search engine.

In this paper, we are concerned with the data fusion problem when the experts' opinions are ranking distributions (rank values or score values) over a set of objects. We are more particularly interested in addressing this issue in the context of information retrieval. In this situation, there are several specificities which are analyzed in [10]. For example, in information retrieval, unlike in multicriteria decision making, we rather aggregate partial lists and the number of partial rankings (the number of search engines) to be combined is lower than the number of objects.

In the information retrieval community, many methods have been proposed for "learning to rank" in order to tackle rank aggregation problems in a supervised manner. In our case, we assume that there is no training data available and we rather deal with unsupervised rank aggregation issues.

In that context, linear combination of experts' scores such as CombSUM or CombMNZ methods [12], often give robust results. A recent work detailed in [10] proposes to extend aggregation approaches developed in multicriteria decision making field for dealing with rank aggregation problems in information retrieval. One of the aims of the present work is similar to the one presented in this last paper: we want to apply aggregation operators already defined in other fields (information fusion field in our case [3], [5]), to the rank aggregation problem in information retrieval. The paper is organized as follows.

In section 2, we first recall some related works and we introduce the basic notations and the baseline approaches CombSUM and CombMNZ. Second, in section 3, we give some details about the scores' normalization step that is required when applying aggregation functions. Next, in section 4, we recall some well-known aggregation operators in information fusion field: triangular norms, triangular conorms and quasi-arithmetic means. Another goal of this paper is then to introduce, in section 5, a new consensus aggregation function. This method has logical foundations, it involves triangular norms and more generally conjunction operators. As we will see, despite its combinatorial aspect, we prove a property that allows to reduce the computational

cost of this new function. Finally, we experiment the different proposals using three datasets taken from the LETOR package [18]. The experiments results given in section 6, show that many aggregation functions and particularly, our method, give better results than baseline approaches.

2. Related work and notations

We suppose there are M experts (the search engines) and N objects. Each expert gives an ordered list of those N objects according to their own preferences (or scores). However, we will restrict ourselves to the top list of each expert (partial rankings). In other words, we don't consider the whole list but only the most relevant objects according to each expert. From these M ordered top lists, the problem we address is to find a single consensus list which combines the M rankings. There are different methods for addressing this issue and we can in fact, notice two families of approaches according to [23], [10]:

- *positional methods*: for each object, we consider the scores given by each expert (the object's profile), we then aggregate these scores using different techniques and we finally re-rank the objects using the aggregated scores. The first positional method was proposed by Borda [4] but linear combination such as the arithmetic mean is for instance, one of these methods [12].
- *majoritarian methods*: these methods use pairwise comparisons matrices of objects and are mostly based upon order relations aggregation using association criteria such as Condorcet's criterion [7], [20] or distance criteria such as Kendall distance [19], [9]. Other methods have also been proposed using Markov chain models [8]. Lastly, outranking methods which were introduced in multicriteria decision problems have been adapted for information retrieval tasks [24], [10].

The methods proposed in this paper belong to the *positional family*. Before introducing the most popular approaches used in this context, we first give some basic notations:

- $S_j(i)$ is the normalized score expert j assigns to object i .
- $\mathcal{L}(j)$ is the ordered list of expert j .
- $\mathcal{L}_k(j)$ is the first k objects of $\mathcal{L}(j)$ or the top- k list of expert j . We will also denote $\mathbf{L}_k = \{\mathcal{L}_k(j); j = 1, \dots, M\}$ the set of all top- k lists.
- $\mathcal{U} = \bigcup_{j=1}^M \mathcal{L}_k(j)$ is the union set of the M top- k lists. We will also denote $K = \#\mathcal{U}$, the number of elements in \mathcal{U} .

- $\forall i \in \mathcal{U} : \text{NZ}(i, \mathbf{L}_k)$ is the number of top- k lists in which object i appears.

Using these notations, we can define below the general linear combination which represents the baseline methods for rank aggregation issues in information retrieval [12], $\forall i \in \mathcal{U}$:

$$\text{Comb}_{v,w}(i) = \text{NZ}(i, \mathbf{L}_k)^v \sum_{j=1}^M w_j S_j(i) \quad (1)$$

where v is an integer and w a $(M \times 1)$ vector of weights.

In this kind of approach, the most usual particular cases employed are the following ones [12], $\forall i \in \mathcal{U}$:

$$\text{CombSUM}(i) = \sum_{j=1}^M S_j(i) \quad (2)$$

$$\text{CombMNZ}(i) = \text{NZ}(i, \mathbf{L}_k) \sum_{j=1}^M S_j(i) \quad (3)$$

Other methods have been proposed for learning the weight vector w in an unsupervised manner. In [16] for example, an unsupervised learning algorithm is suggested. However, this approach is a linear combinations of scores. In our case, we rather use non linear aggregation operators.

3. Normalization step

The normalization step is inherent to any positional methods. Indeed, each expert may have its own particular score distribution which belongs to a certain range of values. As a result, before applying any aggregation function, we have to normalize each distribution if we want them to be mutually comparable. It exists several normalization methods [22]. However, in our particular case, the aggregation operators we are going to introduce have a fuzzy logic flavor. In other words, they aggregate truth values that belong to $[0, 1]$. Thus, we propose to normalize the experts' scores as follows.

Let us denote by $V_j(i)$, the original score value the expert j assigns to object i . Then, its associated normalized score denoted by $S_j(i)$ is given by, $\forall j = 1, \dots, M; \forall i \in \mathcal{L}_k(j)$:

$$S_j(i) = \frac{V_j(i) - \min_{i' \in \mathcal{L}_k(j)} \{V_j(i')\}}{\max_{i' \in \mathcal{L}_k(j)} \{V_j(i')\} - \min_{i' \in \mathcal{L}_k(j)} \{V_j(i')\}} \quad (4)$$

With respect to expert j , the previous normalization formula is given for all objects belonging to the expert's top- k list, $\mathcal{L}_k(j)$. We still need to assign normalized scores to objects that belong to $\mathcal{U} - \mathcal{L}_k(j)$. The only thing we can assume about these last objects is that they are not preferred to any of the objects belonging to $\mathcal{L}_k(j)$. Consequently, $\forall j = 1, \dots, M; \forall i \in \mathcal{U} - \mathcal{L}_k(j)$, we put: $S_j(i) = 0$.

4. Some aggregation operators defined in information fusion

Many aggregation functions have been defined in the field of information fusion. We recall some of them namely triangular norms, triangular conorms and quasi-arithmetic means. These functions will be tested in our experiments in section 6.

4.1. Triangular norms and conorms

Triangular norms (t-norms in short) are binary operations $T : [0, 1]^2 \rightarrow [0, 1]$ which are associative, commutative, non decreasing in each argument and such that $T(a, 1) = x$, $\forall a \in [0, 1]$. They are special functions used in probabilistic metric spaces [25], and fuzzy logic [15].

In fuzzy logic particularly, these operations extend the conjunction from the binary case to the multi-valued case. Let \mathbb{A} and \mathbb{B} be two events¹ and $\mu(\mathbb{A}) = a$ and $\mu(\mathbb{B}) = b$ their associated truth values in $[0, 1]$. Then, $T(a, b)$ measures the truth value of the conjunction $\mathbb{A} \cap \mathbb{B}$. As t-norms are associative and commutative, they can easily be extended for measuring the conjunction of more than two events. We have for example: $T(a, b, c) = T(T(a, b), c)$ and so on.

When using t-norms as aggregation operators, we measure the conjunction of the events to be combined. In our case, we measure the truth value of “object i is relevant for all experts”. However, the disjunction of these events is also of interest. Then, departing from a t-norm T , we can define its related triangular conorm T^* (t-conorm in short) as follows, which allows to extend the disjunction to the multi-valued case:

$$T^*(a, b) = 1 - T(1 - a, 1 - b) \quad (5)$$

Thus, by using t-conorms as aggregation operators, we measure the truth value of the event, “object i is relevant for at least one expert”.

It exists non-parametric and parametric t-norms and t-conorms. The four fundamental non-parametric ones are:

- the minimum t-norm and its related t-conorm:

$$T_M(a, b) = \min(a, b) \quad (6)$$

$$T_M^*(a, b) = \max(a, b) \quad (7)$$

- the product t-norm and its related t-conorm:

$$T_P(a, b) = ab \quad (8)$$

$$T_P^*(a, b) = a + b - ab \quad (9)$$

- the Lukasiewicz t-norm and t-conorm:

$$T_W(a, b) = \max(a + b - 1, 0) \quad (10)$$

$$T_W^*(a, b) = \min(a + b, 1) \quad (11)$$

- the drastic product and its related t-conorm:

$$T_D(a, b) = \begin{cases} 0 & \text{if } (a, b) \in [0, 1]^2 \\ \min(a, b) & \text{otherwise} \end{cases} \quad (12)$$

$$T_D^*(a, b) = \begin{cases} 1 & \text{if } (a, b) \in]0, 1]^2 \\ \max(a, b) & \text{otherwise} \end{cases} \quad (13)$$

T_M and T_D are respectively, upper and lower bounds for t-norms. Thus, $\forall T$ as a t-norm; $\forall (a, b) \in [0, 1]^2$:

$$T_D(a, b) \leq T(a, b) \leq T_M(a, b) \quad (14)$$

T_M gives the highest value for the conjunction of two events: it assumes that one event is included in the other one and thus a positive dependence between them. T_P gives the conjunction value of two events assuming that these last ones are independent. On the contrary, T_W which is lower than T_P , supposes that there is a rather negative dependence between the events. Finally, T_D gives a zero value to the conjunction of two events unless one of the two has a truth value 1.

It also exists numerous parametric t-norms and t-conorms. However, we will introduce only one pair of them: the Schweizer-Sklar family for which the non-parametric t-norms and conorms are particular cases.

The Schweizer-Sklar t-norm is given by, $\forall \lambda \in [-\infty, \infty]$:

$$T_{SS}^\lambda(a, b) = \begin{cases} T_M(a, b) & \lambda = -\infty \\ T_P(a, b) & \lambda = 0 \\ T_D(a, b) & \lambda = \infty \end{cases} \quad (15)$$

and for $\lambda \in]-\infty, 0[\cup]0, \infty[$, we have:

$$T_{SS}^\lambda(a, b) = (\max(a^\lambda + b^\lambda - 1, 0))^{\frac{1}{\lambda}}$$

Its corresponding t-conorm is as follows:

$$T_{SS}^{*\lambda}(a, b) = \begin{cases} T_M^*(a, b) & \lambda = -\infty \\ T_P^*(a, b) & \lambda = 0 \\ T_D^*(a, b) & \lambda = \infty \end{cases} \quad (16)$$

and for $\lambda \in]-\infty, 0[\cup]0, \infty[$, we have:

$$T_{SS}^{*\lambda}(a, b) = 1 - (\max((1 - a)^\lambda + (1 - b)^\lambda - 1, 0))^{\frac{1}{\lambda}}$$

As mentioned beforehand, t-norms or t-conorms can be used as aggregation operators. In that case they represent two *extreme point of views* which are respectively the conjunction and the disjunction over all events. On the contrary,

¹Typically, in our case, events are for example: “object i is relevant for expert j ”

mean operators are more *compromise*. According to [3], we have the following classification:

$$T(a, b) \leq \min(a, b) \leq \mathbf{M}(a, b) \leq \max(a, b) \leq T^*(a, b)$$

where \mathbf{M} is a mean operator.

As we will see in our experiments, due to their limit behaviors, t-norms and t-conorms don't fit well to meta-search problems. T-norms are too strict and they perform very badly. T-conorms perform better than t-norms but still, they are not able to reach the baseline results. Nevertheless, we mention, in section 6, the results obtained using different t-conorms. As a result, in meta-search problems, we must use more compromise operators than t-norms or t-conorms. In that perspective, we introduce in the following paragraph, quasi-arithmetic means.

4.2. Quasi-arithmetic means

Quasi-arithmetic means, also called generalized f -means, are operators that generalize the classical arithmetic mean by using a transformation function f . In our case, we will consider $f : [0, 1] \rightarrow [-\infty, \infty]$. f must be continuous, strictly monotone, and it must also respects the following conditions: $\{f(0), f(1)\} \neq \{-\infty, \infty\}$. The quasi-arithmetic mean associated to f is denoted \mathbf{M}_f . Given a sequence of M values $\{a_1, \dots, a_M\} \in [0, 1]^M$, the quasi-arithmetic mean of this sequence is defined by:

$$\mathbf{M}_f(a_1, \dots, a_M) = f^{-1} \left(\frac{1}{M} \sum_{j=1}^M f(a_j) \right) \quad (17)$$

We will consider more particularly, the case of power functions $f(a) = a^p$ with $p \geq 1$ which are strictly monotonic functions. We remark that CombSUM is obviously equivalent to the arithmetic mean. To go further, we present in our experiments in section 6, the results obtained with $p = 2$ (quadratic mean). As we will see, this function outperforms the CombSUM results.

5. A new consensus aggregation operator

In addition to the previous aggregation functions, we introduce a new method which also presents interesting results on the LETOR datasets [18], compared to baseline methods. Unlike the t-norms and t-conorms methods recalled beforehand, which are extreme point of views, the proposed approach is more consensus. Thus, it is comparable to quasi-arithmetic means. However, unlike this last method, our approach allows to take into account different types of dependencies between experts using different t-norms.

5.1. Definition and logical foundations

Let us denote by $\mathbb{S}_j(i)$ the event: “object i is relevant for expert j ”. If we consider all M experts, we can define the following event: “object i is relevant for *at least m experts among M* ”. Formally, this last event that we will denote $\mathbb{E}_m^M(i)$ is given by the following formula, $\forall m = 1, \dots, M; \forall i = 1, \dots, K$:

$$\mathbb{E}_m^M(i) = \bigcup_{1 \leq j_1 < \dots < j_m \leq M} \left(\bigcap (\mathbb{S}_{j_1}(i), \dots, \mathbb{S}_{j_m}(i)) \right) \quad (18)$$

Given the truth value, $S_j(i) \in [0, 1]$, of event $\mathbb{S}_j(i)$, we are interested in computing the truth value of event $\mathbb{E}_m^M(i)$ which will be denoted $E_m^M(i)$.

This is a combinatorial problem which generalizes Poincaré's formula² also known as the “inclusion-exclusion principle”. It has been shown in [1] that E_m^M can be expressed as a linear combination of the following quantities, $\forall i = 1, \dots, K$:

- for $l = 1$:

$$\mathbf{S}_l^M(i) = \sum_{j=1}^M S_j(i) \quad (19)$$

- for $l = 2, \dots, M$:

$$\mathbf{S}_l^M(i) = \sum_{1 \leq j_1 < \dots < j_l \leq M} T(S_{j_1}(i), \dots, S_{j_l}(i)) \quad (20)$$

where T is a t-norm or a conjunction operation.

This linear combination is related to Jordan's combinatorial formulas [13], [6], and we have, $\forall m = 1, \dots, M; \forall i = 1, \dots, K$:

$$E_m^M(i) = \sum_{l=m}^M (-1)^{l-m} \binom{l-1}{m-1} \mathbf{S}_l^M(i) \quad (21)$$

where $\binom{n}{p} = \frac{n!}{p!(n-p)!}$, are binomial coefficients.

Now that we have introduced the consensus measures $E_m^M; m = 1, \dots, M$, we make a simple assumption on which our aggregation function is based: *the greater the number of experts who find object i relevant, the higher the aggregated score corresponding to object i must be³* [17].

In a literal formulation, our aggregation function can be defined as follows : *we weight by m the measure that object i is relevant for at least m experts then we sum over m from 1 to M* . As the proposed method gives increasing weights

²Poincaré's formula corresponds to the case $m = 1$.

³Notice that baseline methods given in (1) pursue the same goal by weighting by $NZ(i, \mathbf{L}_k)^v$ the experts' score sum. However, our proposal is a different approach which is rather based upon a logical definition.

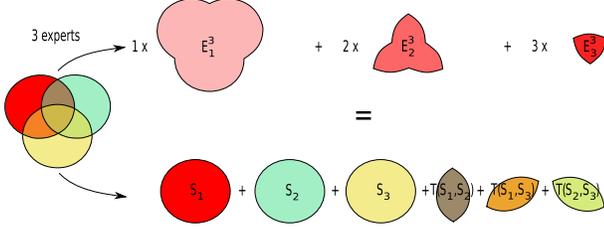


Figure 1. Illustration of the combinatorial “trick” with 3 experts

when m , the number of considered experts, grows, it is a method that respect the property we are looking for.

We denote our consensus aggregation function \mathbf{A} . It is formally defined by the following equation, $\forall i = 1, \dots, K$:

$$\mathbf{A}(i) = \sum_{m=1}^M m E_m^M(i) \quad (22)$$

We may also consider a normalized version by dividing (22) by $\frac{M(M+1)}{2}$.

In its original formulation given by equations (21) and (20), the computation cost of $\mathbf{A}(i)$ is very expensive. Indeed, in order to compute $S_i^M(i)$ in (20), we need to enumerate $\binom{M}{i}$ combinations which grows exponentially. But, as we will show in the following paragraph, using combinatorial properties, we can *reduce the computational cost* of $\mathbf{A}(i)$ to $O(M^2)$.

5.2. The combinatorial “trick”

From equation (22) which has an exponential computational cost, we can prove the following statement which allows to reduce significantly the computational cost of the proposed consensus aggregation operator, $\forall i = 1, \dots, K$:

$$\mathbf{A}(i) = \sum_{j=1}^M S_j(i) + \sum_{1 \leq j < j' \leq M} T(S_j(i), S_{j'}(i)) \quad (23)$$

Given equation (23), we can see that our proposal goes beyond a simple sum⁴ as the proposed aggregation function allows to take into account *pairwise conjunctions between experts*. Particularly, we can specify the type of relationship between each pair of experts by using different t-norms such as introduced in paragraph 4.1.

We give in figure 1, an illustration of the property for $M = 3$. We give the proof of the combinatorial “trick” which allows to obtain the reduced form (23) in appendix.

⁴As the first part is similar to CombSUM.

6. Experiments and results

In this section, using numerical examples, we show that the proposed aggregation operators can perform better than baseline approaches.

We tested the different methods introduced previously, on the three different datasets of the LETOR package [18]:

- OHSUMED dataset (subset of MEDLINE, a database of medical publications): 106 queries, 25 different ranking features (the experts)
- TREC 2003 web track (topic distillation task): 50 queries, 44 different ranking features (the experts)
- TREC 2004 web track (topic distillation task): 75 queries, 44 different ranking features (the experts)

For each datasets, we have different queries (topics) and for each ranking features, we are given a distribution of scores among a set of objects (medical publications or html files). The ranking features are either low-level or high-level content features, between each judged query-document pairs. The reader can consult [18], for a more detailed description of the characteristics of these datasets and for the definition of the features that were extracted.

The LETOR dataset was initially constituted for benchmarking “learning to rank” methods. In that context the goal is to learn, from a training set, how to combine the different experts. In our case, we are in an unsupervised context and we only use the relevance file for measuring a posteriori the performances of the proposed methods.

First, we normalized the data using the score normalization⁵ given by (4) with $k = 1000$. Then, we computed the following aggregation functions: CombSUM (equivalent to M_f with $f(a) = a$), CombMNZ, the quadratic mean (M_f with $f(a) = a^2$) and the new aggregation method \mathbf{A} using either non-parametric t-norms or the parametric t-norm T_{SS}^λ with $\lambda = 2$.

For measuring and comparing the performances of the different aggregation methods, we used the `trec_eval` tool⁶ and we retained the mean average precision (MAP) and the precision at 5 (P@5) measures. The results are given in tables 1, 2 and 3. In italic are the baseline results given by CombSUM. We put in bold the best result with respect to the MAP, among the aggregation operators we tested.

These results show that quadratic means and our aggregation function outperform the baseline results whereas the t-conorms perform very poorly except for the OHSUMED dataset. It is not clear if our approach is better than quasi-arithmetic means as there is little differences between both

⁵We found that it may happen that for some features, a unique value is assigned to all objects. In that case, we replaced this value by 0.

⁶cf. <http://trec.nist.gov/>

Method	MAP	P@5
CombSUM	39.52 %	47.36 %
CombMNZ	38.68 %	46.98 %
Quadratic mean	40.39 %	49.25 %
T_M^*	39.71 %	49.25 %
T_P^*	37.32 %	40.94 %
T_W^*	29.97 %	26.98 %
A with T_M	38.49 %	46.60 %
A with T_P	39.44 %	47.17 %
A with T_W	40.23 %	49.81 %
A with $T_{SS}, \lambda = 2$	40.74 %	50.19 %

Table 1. Results for the OHSUMED dataset

Method	MAP	P@5
CombSUM	16.83 %	16.80 %
CombMNZ	15.42 %	14.40 %
Quadratic mean	16.57 %	16.80 %
T_M^*	07.98 %	07.20 %
T_P^*	04.40 %	03.20 %
T_W^*	01.27 %	01.20 %
A with T_M	15.86 %	14.00 %
A with T_P	16.81 %	16.40 %
A with T_W	17.04 %	17.60 %
A with $T_{SS}, \lambda = 2$	17.37 %	17.60 %

Table 2. Results for the TREC 2003 dataset

Method	MAP	P@5
CombSUM	26.81 %	22.13 %
CombMNZ	23.09 %	20.53 %
Quadratic mean	29.41 %	21.87 %
T_M^*	07.93 %	04.80 %
T_P^*	04.59 %	02.40 %
T_W^*	07.60 %	00.00 %
A with T_M	22.63 %	20.80 %
A with T_P	26.78 %	22.13 %
A with T_W	29.00 %	21.60 %
A with $T_{SS}, \lambda = 2$	29.56 %	21.87 %

Table 3. Results for the TREC 2004 dataset

of them in terms of MAP or P@5. We could have tested M_f with $f(a) = a^p$ and $p \geq 3$. This may give better results than the one obtained using **A**. However, we could also tune the parameter λ when applying **A** with a parametric t-norm.

Finally, despite quite similar results, both methods are different. We argue in favor of our approach as it has deep logical foundations, and as it also allows to model the relationships between experts using different t-norms unlike quasi-arithmetic means. In our experiments, we assumed

that all pairs of experts had the same kind of relationships. In a more general perspective, a better expertise of the dependencies between the features, can be taken into account by choosing for each pair of features a different conjunction operation.

7. Conclusion

In this paper, we address data fusion problems. We are particularly concerned with unsupervised rank aggregation issues in information retrieval field. The general aim of this work is to investigate the use of some aggregation operators defined in information fusion field in the context mentioned beforehand. We have recalled and tested classic aggregation operators such as t-conorms and quasi-arithmetic means. We have moreover introduced a new aggregation method **A**, its logical foundations and its combinatorial properties.

Our experiments on the LETOR dataset show that quasi-arithmetic means and **A** operator can improve the baseline results. It also allows us to conclude that t-norms and t-conorms are not good aggregation operators for that kind of issues and datasets as they are not enough compromise.

The results we obtain in an unsupervised manner are interesting compared to the ones obtained in a supervised manner⁷. Consequently, we intend to embed the methods described in this paper and particularly, the new aggregation function, in a supervised context. In that perspective, the concept of expert’s reliability seems interesting to develop.

8. Acknowledgments

The author would like to thank Jean-Michel Renders for helpful comments.

A. Appendix: proof of the combinatorial “trick”

Proof. In order to prove the property given in equation (23), we will use the two following combinatorial identities⁸:

$$\sum_{p=0}^n (-1)^p \binom{n}{p} = \mathbf{1}_{\{n=0\}} \quad (24)$$

$$\sum_{p=0}^n (-1)^{p+1} \binom{n}{p} p = \mathbf{1}_{\{n=1\}} \quad (25)$$

⁷For example, the baseline results in a supervised context for the OHSUMED dataset reach a MAP of approximately 44.00 % [18]. We have seen previously that the best result we obtained in an unsupervised manner using **A** is 40.74 %.

⁸For identities and properties referring to binomial coefficients and Pascal’s triangle, a good reference is the following website [21].

where $\mathbf{1}_{\{\mathbb{A}\}}$ is the indicator function which equals 1 if proposition \mathbb{A} is true; 0 otherwise. We first have:

$$\begin{aligned}
\mathbf{A}(i) &= \sum_{m=1}^M m E_m^M(i) \\
&= \sum_{m=1}^M m \sum_{l=m}^M (-1)^{l-m} \binom{l-1}{m-1} \mathbf{S}_l^M(i) \\
&= \sum_{m=1}^M m \sum_{l=1}^M (-1)^{l-m} \binom{l-1}{m-1} \mathbf{S}_l^M(i) \mathbf{1}_{\{l \geq m\}} \\
&= \sum_{l=1}^M \sum_{m=1}^M m (-1)^{l-m} \binom{l-1}{m-1} \mathbf{S}_l^M(i) \mathbf{1}_{\{m \leq l\}} \\
&= \sum_{l=1}^M \sum_{m=1}^l m (-1)^{l-m} \binom{l-1}{m-1} \mathbf{S}_l^M(i)
\end{aligned}$$

Then we denote $m' = m - 1$; $l' = l - 1$ and $M' = M - 1$. Using these variables we obtain:

$$\begin{aligned}
\mathbf{A}(i) &= \sum_{l'=0}^{M'} \sum_{m'=0}^{l'} (m'+1) (-1)^{l'-m'} \binom{l'}{m'} \mathbf{S}_{l'+1}^{M'+1}(i) \\
&= \sum_{l'=0}^{M'} \mathbf{S}_{l'+1}^{M'+1}(i) \sum_{m'=0}^{l'} (m'+1) (-1)^{l'-m'} \binom{l'}{m'} \\
&= \sum_{l'=0}^{M'} \mathbf{S}_{l'+1}^{M'+1}(i) \left[\sum_{m'=0}^{l'} (-1)^{l'-m'} \binom{l'}{m'} \right. \\
&\quad \left. + \sum_{m'=0}^{l'} (-1)^{l'-m'} \binom{l'}{m'} m' \right]
\end{aligned}$$

Next, if we enumerate a bit this last expression with respect to l' , we can see that using the combinatorial identities given in (24) and (25): we obtain $\mathbf{S}_1^M(i)$ for $l' = 0$; $\mathbf{S}_2^M(i)$ for $l' = 1$ and a null quantity for $l' > 1$. This shows finally that $\mathbf{A}(i) = \mathbf{S}_1^M(i) + \mathbf{S}_2^M(i)$ which is the same as (23). \square

References

- [1] J. Ah-Pine. *Sur des aspects algébriques et combinatoires de l'Analyse Relationnelle*. PhD thesis, University of Pierre and Marie Curie (Paris 6), 2007.
- [2] K. J. Arrow. *Choix collectif et préférences individuelles*. Calmann-Lévy, 1974.
- [3] I. Bloch. Information Combination Operators for Data Fusion: A Comparative Review with Classification (invited conference). In *SPIE/EUROPTO Conference on Image and Signal Processing for Remote Sensing*, pages 148–159, Rome, Italy, 1994.
- [4] J. Borda. *Mémoire sur les élections au scrutin*. Histoire de l'Académie Royale des Sciences, 1781.
- [5] B. Bouchon Meunier, editor. *Aggregation and fusion of imperfect information*. Physica-Verlag, 1998.
- [6] L. Comtet. *Analyse Combinatoire, tome 1 et tome 2*. Presses Universitaires de France, 1970.
- [7] M. M. d. Condorcet. *Essai sur l'application de l'analyse à la probabilité des décisions rendues à la pluralité des voix*. Paris, 1785.
- [8] C. Dwork, R. Kumar, M. Naor, and D. Sivakumar. Rank aggregation methods for the web. In *WWW*, pages 613–622, 2001.
- [9] R. Fagin, R. Kumar, and D. Sivakumar. Comparing top k lists. *SIAM Journal on discrete mathematics*, 17:134–160, 2003.
- [10] M. Farah and D. Vanderpooten. An outranking approach for rank aggregation in information retrieval. In *Proceedings of the 30th annual international ACM SIGIR Conference*, 2007.
- [11] P. Fishburn. *Mathematics of decision theory*. Mouton Publishers, 1973.
- [12] E. Fox and J. Shaw. Combination of multiple searches. In *Proceedings of the 3rd NIST TREC Conference*, pages 105–109, 1994.
- [13] C. Jordan. Problèmes de la probabilité des épreuves répétées dans le cas général. *Bulletin de la S.M.F.*, 67:223–242, 1939.
- [14] M. Kendall. *Rank correlation methods*. Griffin, 1962.
- [15] E. Klement, R. Mesiar, and E. Pap. *Triangular norms*. Kluwer Academic Pub, 2000.
- [16] A. Klementiev, D. Roth, and K. Small. An unsupervised learning algorithm for rank aggregation. In *Proceedings of the European Conference on Machine Learning (ECML)*, September 2007.
- [17] J. H. Lee. Analyses of multiple evidence combination. *SIGIR Forum*, 31(SI):267–276, 1997.
- [18] T.-Y. Liu, T. Qin, J. Xu, W. Xiong, and H. Li. Letor: Benchmark dataset for research on learning to rank for information retrieval. In *LR4IR 2007 in conjunction with SIGIR 2007*, 2007.
- [19] J. Marcotorchino and P. Michaud. *Optimisation en analyse ordinale des données*. Masson, 1978.
- [20] P. Michaud and J. Marcotorchino. Modèles d'optimisation en analyse des données relationnelles. *Mathématiques et Sciences Humaines*, 17(67):7–38, 1979.
- [21] Pascal. Pascal's triangle from top to bottom. <http://binomial.csuhayward.edu/index.html>.
- [22] M. Renda and U. Straccia. Web metasearch : rank vs score based rank aggregation methods. In *Proceedings of the 2003 ACM symposium on applied computing*, pages 841–846, 2003.
- [23] W. Riker. *Liberalism against populism*. Waveland Press, 1982.
- [24] B. Roy. The outranking approach and the foundations of ELECTRE methods. *Theory and decisions*, 31:49–73, 1991.
- [25] B. Schweizer and A. Sklar. *Probabilistic metric spaces*. North-Holland, 1983.