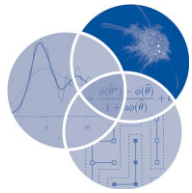
Overview



# Rank aggregation methods

Shili Lin*

This article provides an overview of rank aggregation methods and algorithms, with an emphasis on modern biological applications. Rank aggregation methods have traditionally been used extensively in marketing and advertisement research, and in applied psychology in general. In recent years, rank aggregation methods have emerged as an important tool for combining information from different Internet search engines or from different omics-scale biological studies. We discuss three classes of methods, namely distributional based, heuristic, and stochastic search. The original Thurstone's scaling and its extensions represent the first class of methods that are most appropriate for aggregating many short ranked lists. Aggregating results from consumer rankings of products falls into this category of problems. Its application to biological problems is also being explored. On the other hand, heuristic algorithms and stochastic search methods are applicable to the situation of aggregating a small number of long lists, the so-called 'high-level' meta-analysis scenario. Combining results from different search engines/criteria and a number of omics-scale biological applications fall into this category. Heuristic algorithms are deterministic in nature, ranging from simple arithmetic averages of ranks to Markov chains and stationary distributions. Stochastic search algorithms, on the other hand, aim at maximizing a particular criterion such as that following the Kemeny guideline. Several examples will be provided to illustrate, compare, and contrast the methods and algorithms. The examples range from simple and contrive to representing realistic scenarios. In particular, an application to aggregating results from gene expression microarray studies is provided to demonstrate applications of the methods to modern biological problems.  *WIREs Comp Stat* 2010 2 555–570 DOI: 10.1002/wics.111

**Keywords:** Borda's method; cross entropy Monte Carlo; Markov chain; stochastic search; Thurstone's scaling

## INTRODUCTION

Rank-based methods are frequently used in statistical analysis, especially in non-parametric approaches. The Mann–Whitney rank–sum statistic is a well-known non-parametric method based on the rankings of data. In recent years, there have been considerable interest in adopting rank-based methods for analyzing high-throughput, omics-scale, biological data. For analyzing data from different biological experiments, an advantage of using a rank-based method is that rankings are invariant to transformation and normalization (as long as the relative orderings are preserved); in fact, transformation and normalization are usually not needed for rank-based methods.[1–8] Further, rank-based methods are robust to outliers, although some information is inevitably lost compared to effect-size based methods.[9]

For studies from different platforms/data types that have some commonality, rank-based methods offer the opportunity to integrate individual results to arrive at some consensus that is more 'reliable' than any of the individual studies. For example, suppose data are available for copy number variation from SNP arrays, gene expression from cDNA arrays, and protein binding data from ChIP-seq, the goal is to find putative genes whose functionality may have been altered in a particular disease stage. Because the data come from different technological platforms measuring different aspects of the biological system, it would be hard (if not impossible given the limited understanding of the whole biological system to date) to model the interplay of the raw data to make use of all information available. On the other hand, a rank-based method could be used to synthesize information from the individual studies, each of which contains information on the ranking of a set of important genes. Another scenario may be dealing with multiple independent studies of the same kind. For example, multiple studies may be carried out to identify genes

*Correspondence to: shili@stat.osu.edu

Department of Statistics, The Ohio State University, Columbus, OH 43210, USA

DOI: 10.1002/wics.111

that are differentially expressed between prostate cancers and the normal controls. Some of the studies may be based on the microarray technologies (cDNA or Affymetric gene chips), whereas more recent studies may be next-generation-sequencing based. Even though all the studies aim at finding genes that are differentially expressed, and as such containing common information, the results are likely to vary from study to study. Synthesizing information from all the studies would increase the power to identify true biomarkers while keeping the false positive rate low.[3,10] Combining the results of ranked lists from individual studies, the so-called 'high-level' meta-analysis,[5] would be a viable way of doing so as one does not have to deal with the underlying raw data that come from different platforms.

The rest of the article is organized as follows. We first introduce the concepts of rankings and top-$k$ lists, the latter being the predominant form to be used throughout this article. Although the underlying spaces of the top-$k$ lists need not be the same, we assume that they are in this article to streamline our discussion. We then present three types of rank aggregation methods and discuss their applicability to two broad categories of ranking problems. As part of the description of the optimization methods, Kendall's tau and Spearman's footrule distances and their extensions to top-$k$ lists with potentially variable lengths will be discussed. We offer a couple of examples to demonstrate the methods. We finally conclude with some further remarks.

## COMPLETE AND INCOMPLETE RANKINGS AND TOP-$k$ LISTS

Given a discrete space $T$ (e.g., genes in the human genome) that contains $|T|$ elements, we associate each one with a unique label such that $T$ can be viewed as a list: $T = \{1, 2, \ldots, |T|\}$. Suppose the rank of element $t$ in $T$ is denoted by $R(t)$, then a permutation of $T, \tau(T) = (t_1, t_2, \ldots, t_{|T|})$, such that $R(t_i) < R(t_j)$ for any $i < j$, is a complete ranking of the elements in $T$ without allowing for ties. We refer to $\tau(T)$ as a full ranked list, and we further refer to $R_\tau(t)$ as the rank of element $t$ under the ranking mechanism (function) $\tau$ to distinguish it from the rank of $t$ under a different ranking mechanism. Note that we use curly brackets for a set of elements whose ordering within { } is arbitrary, whereas parentheses are used for an ordered list such that elements in ( ) are ranked from the highest to the lowest.

In many situations, including the biological problems being mentioned earlier in this article, a full ranked list is typically not desirable (nor available). Instead, one is only interested in a partial list $S \subset T$. Without loss of generality, we assume that the partial list $S = (s_1, s_2, \ldots, s_{|S|})$ is ordered according to their ranks such that $R(s_i) < R(s_j)$ for $i < j$. In particular, the situation that is of particular interest is when $S$ is composed of the top $k = |S|$ elements in $T$ according to results from a particular study, $l$. In this case, $R_{\tau_l}(s_j) = j$, where $\tau_l$ is the ranking mechanism associated with study $l$. We refer to such a ranked list as the top-$k$ list. It is implicitly assumed that all the elements that are in $T$ but not in $S$ are ranked lower than $k$. Note that for full ranked lists, the ranking mechanism $\tau$ is simply a permutation of all the elements in the space and thus the terms *ranking mechanism* and *ranked lists* may be used interchangeably without causing any ambiguity. However, this is not the case for top-$k$ lists, and the two terms need to be distinguished.

In the context of the problem being addressed in this article, we have $L$ top-$k$ lists, $S_l$ with $|S_l| = k_l$, $l = 1, \ldots, L$. Although the lengths of the lists may not necessarily be the same, we assume that they all come from the same underlying space $T$ for the sake of concise exposition. Our objective is then to aggregate these $L$ lists to arrive at a new ranking of the elements in $S = \bigcup_{l=1}^{L} S_l$, or a top-$k$ list of $S$, that synthesizes the information contained in all the individual lists.

To facilitate better understanding of the goal and the materials, we borrow a toy example from Lin and Ding[5] to illustrate the problem and the objective. Suppose there are five ice cream flavors, 1 = chocolate, 2 = vanilla, 3 = strawberry, 4 = butter pecan, and 5 = coffee, available for tasting. Three tasters were asked to rank their favorite top three for a market research project. Suppose the rankings from the three tasters are: $S_1 = (1, 2, 3), S_2 = (3, 5, 1)$, and $S_3 = (1, 3, 5)$. Then the aggregate favorite set is $S = \{1, 2, 3, 5\}$. Our goal is to use rank aggregation methods to order these four items by making use of all ranking information from the three tasters.

## CLASSIFICATION OF RANK AGGREGATION METHODS

Rank aggregation methods can be broadly divided into three categories: distributional based, heuristic, and stochastic optimization algorithms. Thurstone's method[11] and its extensions fall into the first category. Optimization algorithms are usually distance measure dependent, and Kemeny optimal aggregation (which optimizes the average Kendall's distances between a candidate aggregate list and each of the input lists) is an example.[12] However, it is well recognized that computing the Kemeny optimal aggregate is

NP-hard even when the number of ranked lists to be aggregated is small. A stochastic search algorithm provides an alternative for finding an optimal solution while circumventing the combinatorial nature of the problem.[5] Such a stochastic search algorithm is with respect to a criterion that we term the generalized Kemeny guidelines, which is defined as the weighted sum of distances between the aggregate list and each of the input lists with respect to a distance measure that is not necessarily Kendall's nor even a metric. More details are given in the Stochastic Optimization section below. Another alternative to direct optimization is a group of algorithms that provide approximate solutions. Such algorithms do not aim at optimizing any criterion, and thus they are heuristic in nature with unknown properties. Although they can be quite effective in some applications,[2,12] they may not perform as well compared to the stochastic search algorithm with respect to criteria conforming to the generalized Kemeny guidelines.

On the other hand, the problem of rank aggregation can be broadly divided into two categories: aggregating many short lists or aggregating a few long lists. Methods for rank aggregation first emerged for solving the former. Consumer rankings of products falls into this category; Thurstone's method was proposed for tackling such kind of problems. The problem of aggregating a few long lists was first presented in aggregating results from Internet search engines. This problem has been approached mainly from the computer science perspective, and a number of Markov chain based heuristic algorithms have been proposed[12] to alleviate the computational burden associated with the NP-hard nature of the problem. In more recent years, similar problems emerged in bioinformatics, where results from different omics experiments (on the same or different technological platforms and on the same or different aspects of the biological system) need to be reconciled and consolidated. The Markov chain based algorithms of Dwork et al.[12] were borrowed for such kind of new applications.[2] Meanwhile, the cross entropy Monte Carlo method (CEMC) that optimizes an objective criterion observing the generalized Kemeny guidelines[5] and the stochastic procedure that tests for random degeneration of paired rank information[4] were designed to solve such problems specifically. The latter is a method that tackles rank aggregation and inference simultaneously, which is not within the scope of this overview and will not be discussed further.

The following three sections describe each of the three broad categories of rank aggregation methods.

## THURSTONE'S MODEL

Thurstone's model, also known as Thurstone scaling[11,13–15] has been used extensively in marketing and advertisement research (e.g., consumer rankings of products) and in applied psychology in general.[16–18] Statistical treatment of Thurstone scaling was pioneered by Mosteller[19] and Daniels,[20] and many extensions have been proposed.[21–24]

### Full Lists

Thurstone's model assumes an underlying continuum of values for each item. Specifically, the vector of underlying values $\boldsymbol{X} = (X_1, \ldots, X_{|T|})$ is assumed to follow a multivariate normal distribution with mean vector $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_{|T|})$ and a $|T| \times |T|$ variance–covariance matrix $\Sigma$. Thus, each pair follows a bivariate normal distribution, that is,

$$(X_u, X_v) \sim \mathrm{BVN}(\mu_u, \mu_v, \sigma_u^2, \sigma_v^2, \rho_{uv}), 0 \le u < v \le |T|.$$

Then

$$X_u - X_v \sim N(\mu_u - \mu_v, \sigma_u^2 + \sigma_v^2 - 2\rho_{uv}\sigma_u\sigma_v),$$

and

$$P(X_u > X_v) = \Phi\left(\frac{\mu_u - \mu_v}{\sqrt{\sigma_u^2 + \sigma_v^2 - 2\rho_{uv}\sigma_u\sigma_v}}\right),$$

where $\Phi$ is the cumulative distribution function of $N(0,1)$. Let $\tau_1, \ldots, \tau_L$ be $L$ independent full rankings of the elements in $T$, then the underlying unobserved data $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_L$ that give rise to these rankings can be considered as independent realizations from $\boldsymbol{X}$. We define Thurstone's score for a pair of elements $u$ and $v$ in the ranked list $l$ as

$$H_{\tau_l}(u, v) = I(R_{\tau_l}(u) < R_{\tau_l}(v)),$$

where $I(\cdot)$ is the usual indicator function that is equal to 1 if the condition inside the parentheses is satisfied; otherwise it is equal to zero. Hence, we may approximate $P(X_u > X_v)$ by the frequency that item $u$ is ranked higher than item $v$:

$$P(X_u > X_v) \approx \frac{1}{L}\sum_{l=1}^{L} H_{\tau_l}(u, v).$$

Considering all pairwise comparisons leads to $|T|(|T|-1)/2$ equations with $|T|(|T|+3)/2$ unknown parameters, a problem in which the parameters are unidentifiable. In practice, the $\sigma_u$'s are often set to 1 and the $\rho_{uv}$ to zero to reduce the number of

parameters.[2] The non-linear least squares approach or other methods are used to estimate $\boldsymbol{\mu}$.[25,26] Care also needs to be taken to address the technical issues in estimating these parameters.[2] The aggregate rankings of the elements are set to be the rankings of the corresponding estimated mean parameters in $\boldsymbol{\mu}$.

### Top-$k$ Lists

For the problem where only partial rankings (e.g. top-$k$ lists) based on the same underlying space are available, Thurstone's method may still be used with the following modification. Let $S_1, \ldots, S_L$ be $L$ top-$k$ rankings of the elements in $T$ with the corresponding underlying ranking mechanisms $\tau_1, \ldots, \tau_L$. For each $u \in S_l^c$, let $R_{\tau_l}(u) = k_l + 1$, where $S_l^c$ is the complement of set $S_l$. In consumer ranking of products, the $S_l$'s are often top-$k$ lists with the same length, but some consumers may wish to only provide top-$h$ rankings with $h < k$. For $u, v \in S = \bigcup_{l=1}^{L} S_l$ define the Thurstone's score

$$H_{\tau_l}(u,v) = \begin{cases} p & \text{if } R_{\tau_l}(u) = R_{\tau_l}(v) = k_l + 1 \\ I(R_{\tau_l}(u) < R_{\tau_l}(v)) & \text{otherwise,} \end{cases}$$

where $I(\cdot)$ is the usual indicator function as defined before and $p$ is a parameter between 0 and 1. More specifically, $p$ can be regarded as the probability that one item is ranked above the other when their rankings are unknown. As such, a reasonable choice of $p$ is 1/2, which corresponds to the scenario of identical distribution for $X_u$ and $X_v$.

From the description of the procedure, one can see that information from ranked lists are being used in a pairwise fashion, leading to potential loss of information. The assumption of normality, independence, and univariate variance may also be unrealistic. Further, if $L$ is too small, it will not be a reasonable method due to poor estimates of probabilities from the frequencies. So Thurston's method is more reasonable for the problem with many short lists as it was originally designed for. In addition to consumer rankings of products problem, there are situations in bioinformatics in which Thurstone's method might find its applications. The problem of finding reproducible true biomarkers addressed by Fishel et al.[3] could potentially be an example of such, but further research is needed to explore this direction.

## HEURISTIC ALGORITHMS

Heuristic algorithms are neither distributional based nor intended to optimize any particular criterion. Instead, they are approaches that are computationally simple and some are intuitive. There has been some evidence showing that they can be effective in aggregating search engine results, although rigorous evaluation of their performance for biological applications is limited. In this section, we discuss two types of heuristic algorithms, namely Borda's methods and Markov chain based methods.

### Borda's Methods

The collection of Borda-inspired methods are intuitive and easy to understand. In the original method proposed by Borda,[27] aggregate ranks were computed based on arithmetic average for full ranked lists. Many other aggregation functions and modifications have been proposed and used, and are applicable to top-$k$ lists. It is interesting to note that variations of Borda's methods are still currently being used for elections in some countries[28,29] (also see www.minelres.lv/NationalLegislation/Slovenia/Slovenia_ElecParl_excerpts_English.htm).

#### *Full Lists*

Borda's method is intuitive and simple to compute for the case of full lists. Given full ordered lists $\tau_1, \ldots, \tau_L$, each a permutation of the underlying space $T$, we let $R_{\tau_l}(u)$ be the rank of element $u \in T$ in list $\tau_l$. We let $B_l(u)$ denote the Borda's score in general, with $B_l(u) = R_{\tau_l}(u)$ being a special case. Let $Bu = f(B_1(u), B_2(u), \ldots, B_L(u))$ be an aggregate function of the Borda scores. Then one sorts the $Bu$'s to obtain an aggregate ranked list $\tau(T)$. Frequently suggested aggregation functions include

$$\begin{aligned} f(x_1, \ldots, x_L) &= \text{median}\{|x_1|, \ldots, |x_L|\} \text{ (median)} \\ f(x_1, \ldots, x_L) &= \left( \prod_{l=1}^{L} |x_l| \right)^{1/L} \text{ (geometric mean)} \\ f(x_1, \ldots, x_L) &= \sum_{l=1}^{L} |x_l|^p / L \ (p - \text{norm}). \end{aligned} \quad (1)$$

Note that the method proposed by Borda[27] is a special case of $p$-norm when $p = 1$ (arithmetic mean) and $B_l(u) = R_{\tau_l}(u)$, apart from a scaling factor. Although the most frequently used Borda score is rank, in situations where other information, beyond the rankings, are available, Borda's score may be defined accordingly to take into account of such additional information.

#### *Top-k Lists and Ice Cream Flavors*

Let $S_1, \ldots, S_L$ be $L$ top-$k$ lists on the same space $T$ with $\tau_1, \ldots, \tau_L$ as the associated ranking mechanisms.

**TABLE 1** | Ice cream flavors: individual's rankings and Borda's aggregates.

| | Taster's rankings | | | ARM | | MED | | GEM | | L2N | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Flavor | L1 | L2 | L3 | Score | Rank | Score | Rank | Score | Rank | Score | Rank |
| 1 | 1 | 3 | 1 | 1.67 | 1 | 1 | 1 | 1.44 | 1 | 3.67 | 1 |
| 2 | 2 | 4 | 4 | 3.33 | 4 | 4 | 4 | 3.17 | 4 | 12.00 | 4 |
| 3 | 3 | 1 | 2 | 2.00 | 2 | 2 | 2 | 1.82 | 2 | 4.67 | 2 |
| 5 | 4 | 2 | 3 | 3.00 | 3 | 3 | 3 | 2.88 | 3 | 9.67 | 3 |

Rankings: modified rankings of tasters with respect to space $S = \{1, 2, 3, 5\}$; ARM = arithmetic average; GEM = geometric mean; L2N = l–2 norm; MED = median.

Denote $S = \cup_{l=1}^{L} S_l$. Then for $u \in S \cap S_l^c$ we may define $B_l(u) = R_{\tau_l}(u) = k_l + 1$. If other information besides rankings are available, we can define $B_l(u)$ accordingly as well. Then one can proceed as for the case of full ranked lists.

The ice cream flavors example is of the problem of aggregating top-$k$ lists from the same underlying space. The modified rankings from the three tasters with respect to the new space $S = S_1 \cup S_2 \cup S_3 = \{1, 2, 3, 5\}$ are given in Table 1 as L1, L2, and L3. For example, taster 3's top-3 flavors are $(1, 3, 5)$, and as such, flavors 1, 3, and 5 are getting the ranks of 1, 2, and 3, respectively. Flavor 2 (vanilla) is in $S$, but not ranked by taster 3, and hence the ranking of $k_3 + 1 = 3 + 1 = 4$ is assigned to flavor 2. The scores computed from four Borda algorithms, namely ARM (arithmetic average), MED (median), GEM (geometric mean), and L2N (l-2 norm), as defined in (1), and their corresponding aggregate rankings are given in four sets of two columns each. As can be seen from the table, all aggregates produce the same result, ranking 1 (chocolate), 3 (strawberry), and 5 (coffee) as the top-3 ice cream flavors.

## Markov Chain Methods

Markov chain methods provide a more elegant but less intuitive alternative to Borda's methods, especially for finding aggregate ranks from top-$k$ lists. On the other hand, Borda's methods treat all ranked lists in their totality, whereas Markov chain based methods use only pairwise ranking information. For the understanding of the Markov chain methods, it is important to note that the assumption about the underlying space(s) from which the top-$k$ come is crucial, as apparently same algorithms for constructing the Markov chain may lead to different results under different assumptions.

Let $S_1, \ldots, S_L$ be $L$ ranked lists (input lists), which could be full or partial ranked lists from the same space. Let $\tau_1, \ldots, \tau_L$ be the $L$ corresponding underlying ranking mechanisms of the lists. Let $S = \bigcup_{l=1}^{L} S_l$, which is a list consisting of all items to be ranked, and is treated as the state space of the Markov chain. As in Borda's method, for each $u \in S \cap S_l^c$, define $R_{\tau_l}(u) = k_l + 1$. Note that if the top-$k$ lists are all of the same length, then the specific ranks assigned to those $u \in S \cap S_l^c$ are immaterial as long as they are the same for all the lists and the common value is larger than the lengths of the lists. The idea is to construct the transition matrix of an ergodic Markov chain such that its stationary distribution will assign a larger probability to a state that is ranked higher. Hence, the stationary distribution will determine the aggregate rankings of the items. Information on pairwise rankings, that is, state $v$ is ranked higher than state $u$ in the same list, will be used to construct the transition probability from state $u$ to $v$. A number of ways of assigning such probabilities are possible depending on one's objectives.[2,12] We discuss three of them below, which may be viewed as variations of algorithms in Dwork et al.[12] and DeConde et al.[2] in that the underlying spaces are explicitly assumed to be the same here.

### *MC 1*

For each $u \in S$, let

$$P(u \to v) = \begin{cases} 1/|S| & \text{if } R_{\tau_l}(u) > R_{\tau_l}(v) \text{ for at least one of the input lists} \\ 0 & \text{otherwise,} \end{cases}$$

for each $v\ (\neq u) \in S$. Then define $P(u \to u) = 1 - \sum_{v \neq u} P(u \to v)$. The general idea in this construction of the transition matrix is that the chain will either move to a state with better ranking in at least one of the lists or stay at the same state. This algorithm is a slight variation of that defined in Dwork et al.,[12] and accommodates the situation of top-$k$ lists.

### *MC 2*

While MC1 may move to any state with a higher ranking in at least one of the input lists with equal probability, MC2 is a majority-rule algorithm.

Specifically, for each $u \in S$, let

$$P(u \to v) = \begin{cases} 1/|S| & \text{if } R_{\tau_l}(u) > R_{\tau_l}(v) \text{ for a majority of the input lists} \\ 0 & \text{otherwise,} \end{cases}$$

for each $v\ (\neq u) \in S$. Then define $P(u \to u) = 1 - \sum_{v \neq u} P(u \to v)$. The general idea in this construction of the transition matrix is that the chain will move to a state with better rankings in at least half of the input lists. It is also an extension of the corresponding algorithm by Dwork et al.[12] and DeConde et al.[2]; this current formulation accommodates the situation of top-$k$ lists from the same space.

### MC 3

While the original MC2 was proposed by Dwork et al.[12] as a spam fighting algorithm due to its majority-rule nature, MC3 may be more appropriate for the multi-platform omics problems given the unique feature of each data type. Specifically, for $u, v \in S$ and $u \neq v$, let

$$P(u \to v) = \sum_{l=1}^{L} I(R_{\tau_l}(u) > R_{\tau_l}(v))/(L|S|),$$

where $I(\cdot)$ is the usual indicator function as defined earlier. Then define $P(u \to u) = 1 - \sum_{v \neq u} P(u \to v)$. The general idea in this construction of the transition matrix is that the chain will move to a state with probability proportional to the number of lists that rank the new state higher than the current one. This is a modification of the MCT algorithm of DeConde et al.[2] to accommodate full ranked lists and top-$k$ lists with the same underlying space.

$$\text{MC1} = \begin{bmatrix} 0.50 & 0.00 & 0.25 & 0.25 \\ 0.25 & 0.25 & 0.25 & 0.25 \\ 0.25 & 0.25 & 0.50 & 0.00 \\ 0.25 & 0.25 & 0.25 & 0.25 \end{bmatrix}$$

**FIGURE 1** | Transition matrix built from heuristic algorithm MC1.

### Technical Note and Ice Cream Flavors

In all three MC procedures, the transition probability may be modified as follows to ensure the existence of a unique stationary distribution as suggested by DeConde et al.:

$$P'(u \to v) = (1 - a)P(u \to v) + a/|S|,$$

where $a$ is a tuning parameter and is usually set to be small.

The ice cream flavors example is simple enough to illustrate and compare the transition matrices constructed from the different MC algorithms. The transition matrix for flavor set $S = \{1, 2, 3, 5\}$ constructed using MC1 is given in Figure 1. Each of the non-zero entries $P(u \to v)$ where $u \neq v$ signifies that there is at least one of the three tasters who ranked flavor $v$ higher than flavor $u$. For example, taster 2 ranked flavor 5 above 1, and as such the last entry on the first row is equal to 1/4 ($\neq 0$). As can be seen from Figure 1, the transition matrix leads to an ergodic Markov chain, that is, any state in $S$ can be reached from any other state aperiodically.

Because of the more stringent requirement for MC2, there are more zero entries in the transition matrix (left matrix of Figure 2(a)). For example, among the three tasters, only taster 2 ranked flavor 5 above 1, which is a minority opinion, and thus the last

**(a)**

$$\text{MC2} = \begin{bmatrix} 1.00 & 0.00 & 0.00 & 0.00 \\ 0.25 & 0.25 & 0.25 & 0.25 \\ 0.25 & 0.00 & 0.75 & 0.00 \\ 0.25 & 0.00 & 0.25 & 0.50 \end{bmatrix} \xrightarrow{a = 0.05} \begin{bmatrix} 0.9625 & 0.0125 & 0.0125 & 0.0125 \\ 0.2500 & 0.2500 & 0.2500 & 0.2500 \\ 0.2500 & 0.0125 & 0.7250 & 0.0125 \\ 0.2500 & 0.0125 & 0.2500 & 0.4875 \end{bmatrix}$$

**(b)**

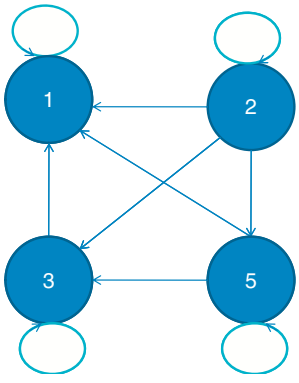



**FIGURE 2** | Dissection of heuristic algorithm MC2. (a) Transition matrix built from the algorithm (left matrix) and its ergodic counter part with tuning parameter $a = 0.05$ (right matrix). (b) Graphical representation of the transition matrix built from MC2.

$$\text{MC3} = \begin{bmatrix} 0.833 & 0.000 & 0.083 & 0.083 \\ 0.250 & 0.417 & 0.167 & 0.167 \\ 0.167 & 0.083 & 0.750 & 0.000 \\ 0.167 & 0.083 & 0.250 & 0.500 \end{bmatrix}$$

**FIGURE 3** | Transition matrix built from heuristic algorithm MC3.

entry on the first row now equals to zero. The corresponding Markov chain is no longer ergodic. As can be seen from Figure 2(b), state 2 can never be reached from any of the other states, and state 1 is an absorbent state (the Markov chain will stay at state 1 as soon as it reaches there). Nevertheless, as discussed in the technical note, the transition matrix can be easily modified to make it ergodic (right matrix of Figure 2(a)).

The transition matrix constructed from MC3 (Figure 3) has non-zero entries in the same positions as that from MC1, albeit with potentially smaller probabilities. Hence, the corresponding Markov chain is also ergodic. All three MC algorithms lead to the aggregate top-3 flavors ranking of $(1, 3, 5)$ (for a wide range of $a$, say from 0.01 to 0.15), the same as any of the Borda's algorithms.

## STOCHASTIC OPTIMIZATION METHODS

Optimization methods are with respect to some optimization criterion. Optimization criteria for aggregating top-$k$ lists are usually based on some measure of disagreement between the input lists and the aggregate rankings. One particular desirable criterion is that it conforms to the generalized Kemeny guidelines. One broad formulation that conforms to the generalized Kemeny guidelines is the weighted sum of distances between the aggregate rankings and the input lists. Thus, whether a particular aggregate list is better than another depends on the distance measure chosen. In the following, we elaborate on a class of optimization criteria and distance measures before presenting a stochastic optimization algorithm.

### Optimization Criteria

Optimization criteria following the generalized Kemeny guidelines are distance dependent. Suppose $\tau_1, \ldots, \tau_L$ are $L$ full ranked lists of space $T$, and our objective is to find the underlying 'correct' ordering of $T$, $\tau$. Suppose we assume that each $\tau_l$ is a 'noisy' realization of $\tau$ in that two of the elements in $\tau$ are swapped with some probability $p < 1/2$, then it can be shown[30] that the maximum likelihood estimate of $\tau$ is

$$\hat{\tau} = \arg\min_{\tau} \frac{1}{L} \sum_{l=1}^{L} K(\tau, \tau_l),$$

where $K(\tau, \tau_l)$ is the Kendall's tau distance between the underlying correct ranking $\tau$ and input list $\tau_l$. The estimated $\hat{\tau}$ is referred to as the Kemeny optimal aggregation.[12]

In reality, the observed $\tau_i$'s are not simply a swap of two elements of the correct underlying rankings, and the input lists are unlikely to be of full-rank. However, this special case does lend itself to the following general optimization criterion, which we call generalized Kemeny guidelines. Let $S_1, \ldots, S_L$ be $L$ top-$k$ lists with respect to the same space $T$ but different underlying ranking mechanisms $\tau_1, \ldots, \tau_L$. Denote $S = \bigcup_{l=1}^{L} S_l$. Let $A$ be a full or top-$k$ (with a pre-specified fixed size) list with respect to $S$, which is now treated as the new common space. For ease of reference, we (re)label the elements in $S$ as $1, 2, \ldots, n$ such that $S = \{1, \ldots, n\}$. Our goal is to find an estimate of $A$, an ordered subset of $S$, that minimizes the weighted sum of distances between $A$ and each of the input list $S_l$.[5] That is, we seek $A^*$ such that

$$\begin{aligned} A^* &= \arg\min_{A} \{\Phi(A), A \subset S = (1, \ldots, n)\} \\ &= \arg\min_{A} \left\{ \sum_{l=1}^{L} w_l d(A, S_l), \tau \subset S \right\}, \end{aligned} \tag{2}$$

where $\boldsymbol{w} = (w_1, \ldots, w_L)$ is a weight vector that can be used to specify prior information on the relative importance or reliability of the input lists, and $d$ is a distance measure between two lists that may be of different lengths. When $w_l = 1$ for all $l$ and $d(A, S_l) = K(A, S_l)$, (2) reduces to the Kemeny optimal aggregation. In the following subsection, we discuss two specific distance measures, the Kendall's tau distance and the Spearman's footrule distance. The optimization criterion as defined in (2) with respect to the Kendall's tau or the Spearman's distance will be referred to as the Kendall's criterion or the Spearman's criterion, respectively, hereafter.

### Distance Measures

A distance measure is essential in aggregating results from ranked lists that is based on an optimization criterion. Distance measures are of importance in non-parametric rank-based methods in statistics in general, and the topic has been treated extensively.[31] To fully specify (2), we need to specify the distance between two top-$k$ lists that are not necessarily of the same length. Here we discuss two distance measures, the Kendall's tau distance[32] and the Spearman's footrule distance,[33] and their modifications for top-$k$ lists.

### *Kendall's Tau*

#### *Full Lists*

For two full ranked lists $\tau_1$ and $\tau_2$ defined on the space $T$, Kendall's tau distance $K(\tau_1, \tau_2)$ is essentially counting the number of pairwise discordances between the two lists. Let

$$d_k(i,j) = I[(R_{\tau_1}(i) - R_{\tau_1}(j))(R_{\tau_2}(i) - R_{\tau_2}(j)) < 0], \quad i,j = 1,\ldots,|T|$$

where $I(\cdot)$ is the usual indicator function as defined before. Then

$$K(\tau_1, \tau_2) = \sum_{i,j} d_k(i,j)$$

is defined as the Kendall's tau distance.

#### *Top-k Lists*

To deal with the situation of partial lists, the Kendall's tau distance is modified following Fagin et al.[34] Let $S_1$ be the top-$k$ list of length $|S_1| = k_1$ from study 1 and $S_2$ be the top-$k$ list of length $|S_2| = k_2$ from study 2 with respect to the space $T$. Let $\tau_1$ and $\tau_2$ be the underlying associated ranking mechanisms over the space $T$. Let $S = S_1 \cup S_2$, which contains all the elements that are present in either $S_1$ or $S_2$. For each $u \in S \cap S_1^c$, define $R_{\tau_1}(u) = k_1 + 1$, and for each $u \in S \cap S_2^c$, define $R_{\tau_2}(u) = k_2 + 1$.

For each pair of elements $u, v \in S$, define

$$B = \{(u,v) : R_{\tau_1}(u) = R_{\tau_1}(v) = k_1 + 1 \text{ or } R_{\tau_2}(u) = R_{\tau_2}(v) = k_2 + 1\},$$

which is the collection of pairs of elements with the following property: both elements of each pair belong to one of the two lists but neither belongs to the other list. Let

$$d_k(u,v) = \begin{cases} I\left\{[R_{\tau_1}(u) - R_{\tau_1}(v)] \right. & \text{if } (u,v) \in B^c \\ \quad \left. [R_{\tau_2}(u) - R_{\tau_2}(v)] < 0\right\} & \\ p & \text{otherwise}, \end{cases} \quad (3)$$

where $p$ is a parameter between 0 and 1. Then the modified Kendall's tau distance is defined as before

$$K(S_1, S_2) = \sum_{u,v \in S} d_K(u,v).$$

For aggregating top-$k$ lists, the distances to be computed are those between the candidate aggregate list $A$ and each of the input list $S_l$. Suppose the lengths of the input lists, $k_l, l = 1,\ldots,L$, are not all the same, then the scaled Kendall's distance would be more appropriate. For example, $k_l(k_l - 1)/2$, the largest Kendall's distance between two full lists of length $k_l$, would be a reasonable choice as a scaling factor.

It remains to set the parameter $p$ in (3) to fully define the modified Kendall's distance. Following the discussion in Fagin, setting $p = 0$ is an 'optimistic approach' because it does not assign any penalty to a pair when it is uncertain whether the pair are in opposite order in the two lists. On the other hand, setting $p = 1/2$ gives an intuitively 'neutral approach'. In fact, this neutral approach has a statistical interpretation in the case when $k_1 = k_2$: $K(S_1, S_2)$ is the expected value of Kendall's tau defined on two permutations of $S = S_1 \cup S_2$ each preserving one of the two top-$k$ lists. Also note that in the original definition of Kendall's distance for full ranked lists, it is a bona fide distance metric. However, the modified Kendall's tau as defined in (3) is no longer a metric.[34]

### *Spearman's Footrule Distance*

#### *Full Lists*

For full ranked lists $\tau_1$ and $\tau_2$, the Spearman's footrule distance is the sum of the absolute differences between the ranks of the two lists over all elements in $T$:

$$S(\tau_1, \tau_2) = \sum_{u \in T} |R_{\tau_1}(u) - R_{\tau_2}(u)|.$$

From the definition, one can see that Spearman's footrule takes the actual rankings of the elements into consideration, whereas in Kendall's tau, only relative rankings matter. For full ranked lists, the maximum Spearman's distance is $|T|^2/2$ for an even $|T|$ and $(|T| + 1)(|T| - 1)/2$ for an odd $|T|$, which corresponds to the situation in which the two lists are exactly the reverse of one another.

#### *Top-k Lists*

To modify Spearman's footrule distance to accommodate partial lists $S_1$ and $S_2$, we consider $S = S_1 \cup S_2$ as the new space. We define $R_{\tau_1}(u)$ and $R_{\tau_2}(u)$ as before for the modified Kendall's distance for each $u \in S \cap S_1^c$ and for each $u \in S \cap S_2^c$. With such modified rankings, we can then compute the Spearman's distance as for the full ranked lists. It turns out that, for the case when $k_1 = k_2$, the above definition of Spearman's distance for top-$k$ lists has the same statistical interpretation as in the Kendall's tau's case.

As discussed earlier for Kendall's distance, it would be more sensible to use the scaled Spearman's distance in the optimization criterion for aggregating

top-$k$ lists if not all the top-$k$'s are of the same length. A reasonable scaling factor for distance $S(A, S_l)$ between the aggregate list $A$ and input list $S_l$ would be the maximum Spearman's distance for two full lists of length $k_l = |S_l|$.

## Stochastic Optimization

To optimize criterion (2), an exhaustive search will not be tractable even for small size top-$k$ lists. Instead, a stochastic search method that can efficiently explore the space of $A$ has been proposed[5] by adopting the cross entropy Monte Carlo (CEMC) approach.[35]

The general idea of CEMC may lead to multiple algorithms for performing the same task. Here we present the Order Explicit Algorithm (OEA) of Lin and Ding,[5] in which the orders of the elements in the optimal list are explicitly specified in the probability matrix as follows. Let $\boldsymbol{X} = (X_{jr})_{n\times k}$ be a random matrix with each component variable $X$ taking values of 0 or 1, and with the constraints that each column sums to 1 and each row sums to at most 1. Note that the lengths of the input lists are not necessarily the same, and the length $k$ of the aggregate top-$k$ list can be any number not exceeding the size of the union of the input lists, $n$. Let $\boldsymbol{v} = (p_{jr})_{n\times k}$ denote the corresponding probability matrix, with the constraints that each column sums to 1. Then each column variable, $\boldsymbol{X}_r = (X_{1r}, X_{2r}, \ldots, X_{nr})$, follows a multinomial distribution with a sample size of 1 and the probability vector $\boldsymbol{v}_r = (p_{1r}, \ldots, p_{nr})$, and with the above stated constraints on the joint column variables.

With such a specification, a realization of $\boldsymbol{X}$, $\boldsymbol{x}$, uniquely determines the corresponding candidate top-$k$ list without the need to reference the probability matrix. That is,

$$A = f(\boldsymbol{x}) = (x_{jr} | x_{jr} = 1, j = 1, \ldots, n, r = 1, \ldots, k).$$

In words, the '1' entries in each of the $k$ columns make up the top-$k$ list, in that order. Given the 1–1 correspondence between $A$ and $\boldsymbol{x}$, finding $A^*$ is equivalent to finding $\boldsymbol{x}^*$ that minimizes $\Phi\{f(\boldsymbol{x})\}$.

The idea of finding $\boldsymbol{x}^*$ using CEMC is to iteratively update the parameter matrix $\boldsymbol{v}$ such that, iteration by iteration, $P_{\boldsymbol{v}}(\boldsymbol{x})$ will place more and more of its probability mass on the $\boldsymbol{x}$'s that are in the 'neighborhood' of $\boldsymbol{x}^*$. Loosely speaking, $\boldsymbol{x}$ is called a neighbor of $\boldsymbol{x}^*$ if the corresponding value of the objective function, $y = \Phi\{f(\boldsymbol{x}; \boldsymbol{v})\}$, is close to the minimum $y^*$. Suppose $\boldsymbol{v}$ is the current estimate of the parameter matrix. We choose the next parameter update $\boldsymbol{v}'$ to minimize the cross entropy (i.e., Kullback–Leibler measure) $CE(Q^*, P_{\boldsymbol{v}'})$ between the distributions $P_{\boldsymbol{v}'} = P_{\boldsymbol{v}'}(\boldsymbol{x})$ and $Q^*$, where $Q^*$, given in the following, is the ideal (but unobtainable) importance sampling distribution for estimating rare probability $B = P_{\boldsymbol{v}}[\Phi\{f(\boldsymbol{x}; \boldsymbol{v})\} \le y]$:

$$Q^*(\boldsymbol{x}) = \frac{I[\Phi\{f(\boldsymbol{x}; \boldsymbol{v})\} \le y] P_{\boldsymbol{v}}(\boldsymbol{x})}{B}.$$

One can show that, after some simple algebra, minimizing $CE(Q^*, P_{\boldsymbol{v}'})$ is equivalent to maximizing

$$\sum_{\boldsymbol{x}} \{I[\Phi\{f(\boldsymbol{x}; \boldsymbol{v})\} \le y] \log P_{\boldsymbol{v}'}(\boldsymbol{x})\} P_{\boldsymbol{v}}(\boldsymbol{x}) = E_{\boldsymbol{v}}\left[I[\Phi\{f(\boldsymbol{x}; \boldsymbol{v})\} \le y] \log P_{\boldsymbol{v}'}(\boldsymbol{x})\right],$$

which is now free of the probability, $B$, to be estimated.

Suppose $\boldsymbol{x}_i = (x_{ijr})_{n\times k}, i = 1, \ldots, N$, is a sample drawn from $P_{\boldsymbol{v}}(\boldsymbol{x})$ with the current parameter specification $\boldsymbol{v}$, with their corresponding candidate top-$k$ lists denoted as $\tau_i = f(\boldsymbol{x}_i), i = 1, \ldots, N$. Then

$$\boldsymbol{v}_{\text{new}} = \arg\max_{\boldsymbol{v}'} \left\{ \frac{1}{N} \sum_{i=1}^{N} I[\Phi\{f(\boldsymbol{x}_t; \boldsymbol{v})\} \le y] \log P_{\boldsymbol{v}'}(\boldsymbol{x}_t) \right\} = \left[ \frac{\sum_{i=1}^{N} I\{\Phi(\tau_i) \le y\} x_{ijr}}{\sum_{i=1}^{N} I\{\Phi(\tau_i) \le y\}} \right]_{j=1,\ldots,n; r=1,\ldots,k,}$$

can be used in the update for the next parameter matrix $\boldsymbol{v}'$. In practice, a weighted average of $\boldsymbol{v}$ (with $N_1$ realizations) and $\boldsymbol{v}_{\text{new}}$ (with $N - N_1$ realizations) as $\boldsymbol{v}'$ can better balance the rate of convergence and the chance of not being trapped in a local minimum. Furthermore, the threshold value $y$ will also be updated iteratively by setting it to be the $\rho$-quantile of the values of the objective function defined in (2).[5] This exercise will lead to the construction of a sequence, $y_0, y_1, \ldots$, which converges to a value $(y_\infty)$ close to $y^*$.[36] Similarly, $\boldsymbol{v}_0, \boldsymbol{v}_1, \ldots$, will converge to $\boldsymbol{v}_\infty$, with the corresponding $P_{\boldsymbol{v}_\infty}(\boldsymbol{x})$ placing most of its probability mass on the $\boldsymbol{x}$'s that satisfy $\Phi\{f(\boldsymbol{x}; \boldsymbol{v})\} \le y_\infty$.

### *Technical Note and Ice Cream Flavors*

There are a number of tuning parameters that need to be set to run the OEA efficiently. They include $N, N_1, \rho$ as well as the initial starting matrix. Some recommendations on how to set these parameters can be found in the works of Liu et al.[37] and Lin and Ding,[5] but it is always a good idea to run OEA with multiple sets of tuning parameters to increase the chance of finding the global maximum.

The ice cream flavors example also offers a good opportunity to dissect the OEA algorithm

$$\upsilon^0 = \begin{bmatrix} 0.250 & 0.250 & 0.250 \\ 0.250 & 0.250 & 0.250 \\ 0.250 & 0.250 & 0.250 \\ 0.250 & 0.250 & 0.250 \end{bmatrix} \rightarrow \upsilon^1 = \begin{bmatrix} 0.425 & 0.225 & 0.225 \\ 0.125 & 0.175 & 0.375 \\ 0.325 & 0.375 & 0.125 \\ 0.125 & 0.225 & 0.275 \end{bmatrix} \rightarrow \upsilon^2 = \begin{bmatrix} 0.613 & 0.213 & 0.113 \\ 0.063 & 0.088 & 0.388 \\ 0.263 & 0.588 & 0.063 \\ 0.063 & 0.113 & 0.438 \end{bmatrix} \rightarrow$$

$$\upsilon^3 = \begin{bmatrix} 0.756 & 0.156 & 0.056 \\ 0.031 & 0.044 & 0.294 \\ 0.181 & 0.744 & 0.031 \\ 0.031 & 0.056 & 0.619 \end{bmatrix} \rightarrow \upsilon^4 = \begin{bmatrix} 0.878 & 0.078 & 0.028 \\ 0.016 & 0.022 & 0.147 \\ 0.091 & 0.872 & 0.016 \\ 0.016 & 0.028 & 0.809 \end{bmatrix} \rightarrow \upsilon^5 = \begin{bmatrix} 0.939 & 0.039 & 0.014 \\ 0.008 & 0.011 & 0.073 \\ 0.045 & 0.936 & 0.008 \\ 0.008 & 0.014 & 0.905 \end{bmatrix} \rightarrow$$

$$\upsilon^6 = \begin{bmatrix} 0.970 & 0.020 & 0.007 \\ 0.004 & 0.006 & 0.037 \\ 0.023 & 0.968 & 0.004 \\ 0.004 & 0.007 & 0.952 \end{bmatrix} \rightarrow \upsilon^7 = \begin{bmatrix} 0.985 & 0.010 & 0.004 \\ 0.002 & 0.003 & 0.018 \\ 0.011 & 0.984 & 0.002 \\ 0.002 & 0.004 & 0.976 \end{bmatrix} \rightarrow \upsilon^8 = \begin{bmatrix} 0.992 & 0.005 & 0.002 \\ 0.001 & 0.001 & 0.009 \\ 0.006 & 0.992 & 0.001 \\ 0.001 & 0.002 & 0.988 \end{bmatrix} \rightarrow$$

$$\upsilon^9 = \begin{bmatrix} 0.996 & 0.002 & 0.001 \\ 0.001 & 0.001 & 0.005 \\ 0.003 & 0.996 & 0.001 \\ 0.001 & 0.001 & 0.994 \end{bmatrix}$$

**FIGURE 4 |** Path of probability matrices - from starting ($\upsilon^0$) to convergence ($\upsilon^9$). The starting probability matrix is completely uninformative. The matrix at convergence leads to the top-3 list (1, 3, 5) by selecting the ice cream flavors corresponding to the largest probability in each column. The columns may not sum to exactly 1 due to rounding.

in terms of how its underlying probability matrix converges. This is a small problem where complete enumeration is possible so that the true answer is known. Consistent with the illustrations for the heuristic algorithms, suppose we wish to find the top-3 list $A$ according to the generalized Kemeny guidelines. In other words, we wish to find $A$ that minimizes $\Phi(A) = w_1 d(A, S_1) + w_2 d(A, S_2) + w_3 d(A, S_3)$. Since we do not have any prior information to suggest how reliable a taster's choice may be in representing the opinion of the population, we set $w_1 = w_2 = w_3 = 1$. In all, there are 24 possible top-3 rankings. For both the Kendall's tau and the Spearman's footrule distances, $A = (1, 3, 5)$ achieves the minimum value of the objective function: $\Phi(A = (1, 3, 5))$ equals to 4 and 8, respectively. Note that this optimal ranking was obtained by all of the heuristic algorithms. With no exception, OEA also obtains this optimal aggregate top-3 list using both the Kendall's and Spearman's criteria. Figure 4 shows the sequence of the probability matrices from the starting value to the one at convergence after nine iterations.

## EXAMPLES

Two examples are presented here to illustrate, compare, and contrast the performances of the algorithms. The first is a contrived but realistic example (compared to the ice cream flavors one) in term of the size of the top-$k$ lists. Although enumeration is no longer feasible, the data are created in such a way that the 'answer' can be guessed to a large extend. The second is an application to aggregating results from five gene expression studies. Since both examples are of the type of a few long lists, Thurstone's method is not used in the analyses.

### Three Long Lists

Consider the problem of obtaining an integrated top-40 list from three individual ranked lists, L1, L2, and L3, given in Table 2. As can be seen from the table, there are 65 items in the aggregate candidate set $S$, with 20 of them (items 1–20) appear in all three lists, five is common in each of the three pairs of the lists, and 30 of them only present in one of the studies. Since the number of potential aggregate top-40 lists is more than $5 \times 10^{65}$, it is impossible to evaluate all of them to find the aggregate list $A$ that optimizes the objective function. However, it would be quite reasonable to guess that the 35 items that are present in at least two of the individual lists should be included in the optimal integrated top-40 list, with their ordering being (1–30, 46–50). It is uncertain, though, as to what the last five targets in the optimal top-40 list may be. Throughout this demonstration of methods, we assume that all three lists come from the same space $T$, and therefore, any element in $S$ but not being ranked by a particular list will get the (implicit) ranking of 41 in that list. Note that this is an essential assumption, as different results may emerge if the underlying spaces of the ranked lists are assumed to be different.

#### *Borda*

For Borda's method, the score is taken to be the rank, and four aggregation functions, arithmetic mean (ARM), median (MED), geometric mean (GEM), and l2-norm (L2N), are used to obtain an aggregate top-40 list. The results and their corresponding Kendall's and Spearman's criteria are given in Table 2. As we can see from the table, all four aggregate lists select elements 1–20 as the first 20 ranked items, in that order, which match up with the information contained in the input lists. All four lists also contain elements 21–30,

**TABLE 2** | Input lists and aggregation results from the heuristic and the stochastic algorithms.

| | Input lists | | | Borda | | | | Markov chain | | | Stochastic | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Rank | L1 | L2 | L3 | ARM | MED | GEM | L2N | MC1 | MC2 | MC3 | CEK | CES |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |
| 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 |
| 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 |
| 7 | 7 | 7 | 7 | 7 | 7 | 7 | 7 | 7 | 7 | 7 | 7 | 7 |
| 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 |
| 9 | 9 | 9 | 9 | 9 | 9 | 9 | 9 | 9 | 9 | 9 | 9 | 9 |
| 10 | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 10 |
| 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 |
| 12 | 12 | 12 | 12 | 12 | 12 | 12 | 12 | 12 | 12 | 12 | 12 | 12 |
| 13 | 13 | 13 | 13 | 13 | 13 | 13 | 13 | 13 | 13 | 13 | 13 | 13 |
| 14 | 14 | 14 | 14 | 14 | 14 | 14 | 14 | 14 | 14 | 14 | 14 | 14 |
| 15 | 15 | 15 | 15 | 15 | 15 | 15 | 15 | 15 | 15 | 15 | 15 | 15 |
| 16 | 16 | 16 | 16 | 16 | 16 | 16 | 16 | 16 | 16 | 16 | 16 | 16 |
| 17 | 17 | 17 | 17 | 17 | 17 | 17 | 17 | 17 | 17 | 17 | 17 | 17 |
| 18 | 18 | 18 | 18 | 18 | 18 | 18 | 18 | 18 | 18 | 18 | 18 | 18 |
| 19 | 19 | 19 | 19 | 19 | 19 | 19 | 19 | 19 | 19 | 19 | 19 | 19 |
| 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 |
| 21 | 21 | 21 | 56 | 21 | 21 | 21 | 21 | 21 | 21 | 21 | 21 | 21 |
| 22 | 22 | 22 | 57 | 22 | 22 | 22 | 22 | 22 | 22 | 22 | 22 | 22 |
| 23 | 23 | 23 | 58 | 23 | 23 | 23 | 23 | 23 | 23 | 23 | 23 | 23 |
| 24 | 24 | 24 | 59 | 24 | 24 | 24 | 24 | 26 | 24 | 24 | 24 | 24 |
| 25 | 25 | 25 | 60 | 25 | 25 | 25 | 25 | 24 | 25 | 25 | 25 | 25 |
| 26 | 26 | 41 | 26 | 26 | 26 | 26 | 26 | 27 | 26 | 26 | 26 | 26 |
| 27 | 27 | 42 | 27 | 27 | 27 | 27 | 27 | 25 | 27 | 27 | 27 | 27 |
| 28 | 28 | 43 | 28 | 28 | 28 | 28 | 28 | 28 | 28 | 28 | 28 | 28 |
| 29 | 29 | 44 | 29 | 29 | 29 | 29 | 29 | 46 | 29 | 29 | 29 | 29 |
| 30 | 30 | 45 | 30 | 30 | 30 | 56 | 30 | 29 | 30 | 30 | 30 | 30 |
| 31 | 31 | 46 | 46 | 46 | 46 | 30 | 46 | 47 | 46 | 46 | 46 | 46 |
| 32 | 32 | 47 | 47 | 56 | 47 | 57 | 47 | 30 | 47 | 47 | 47 | 47 |
| 33 | 33 | 48 | 48 | 57 | 48 | 58 | 56 | 48 | 48 | 48 | 48 | 48 |
| 34 | 34 | 49 | 49 | 47 | 49 | 46 | 57 | 56 | 49 | 49 | 49 | 49 |
| 35 | 35 | 50 | 50 | 58 | 50 | 59 | 48 | 49 | 50 | 50 | 50 | 50 |
| 36 | 36 | 51 | 61 | 59 | 31 | 47 | 58 | 57 | 57 | 56 | 56 | 56 |
| 37 | 37 | 52 | 62 | 48 | 32 | 60 | 59 | 50 | 56 | 57 | 57 | 57 |
| 38 | 38 | 53 | 63 | 60 | 33 | 41 | 60 | 58 | 58 | 58 | 31 | 58 |
| 39 | 39 | 54 | 64 | 41 | 34 | 48 | 49 | 59 | 60 | 59 | 41 | 59 |
| 40 | 40 | 55 | 65 | 42 | 35 | 42 | 41 | 41 | 59 | 60 | 42 | 60 |
| Kendall | | | | 404 | 400 | 404 | 401 | 402 | 402 | 400 | 392 | 400 |
| Spearman | | | | 488 | 450 | 504 | 476 | 480 | 450 | 450 | 450 | 450 |

**TABLE 3** | Individual top-25 ranked genes from five prostate cancer studies.

| Rank | Luo | Welsh | Dhanasekaran | True | Singh |
|---|---|---|---|---|---|
| 1 | HPN | HPN | OGT | AMACR | HPN |
| 2 | AMACR | AMACR | AMACR | HPN | SLC25A6 |
| 3 | CYP1B1 | OACT2 | FASN | NME2 | EEF2 |
| 4 | ATF5 | GDF15 | HPN | CBX3 | SAT |
| 5 | BRCA1 | FASN | UAP1 | GDF15 | NME2 |
| 6 | LGALS3 | ANK3 | GUCY1A3 | MTHFD2 | LDHA |
| 7 | MYC | KRT18 | OACT2 | MRPL3 | CANX |
| 8 | PCDHGC3 | UAP1 | SLC19A1 | SLC25A6 | NACA |
| 9 | WT1 | GRP58 | KRT18 | NME1 | FASN |
| 10 | TFF3 | PPIB | EEF2 | COX6C | SND1 |
| 11 | MARCKS | KRT7 | STRA13 | JTV1 | KRT18 |
| 12 | OS-9 | NME1 | ALCAM | CCNG2 | RPL15 |
| 13 | CCND2 | STRA13 | GDF15 | AP3S1 | TNFSF10 |
| 14 | NME1 | DAPK1 | NME1 | EEF2 | SERP1 |
| 15 | DRRK1A | TMEM4 | CALR | RAN | GRP58 |
| 16 | TRAP1 | CANX | SND1 | PRKACA | ALCAM |
| 17 | FMO5 | TRA1 | STAT6 | RAD23B | GDF15 |
| 18 | ZHX2 | PRSS8 | TCEB3 | PSAP | TMEM4 |
| 19 | RPL36AL | EMTPD6 | EIF4A1 | CCT2 | CCT2 |
| 20 | ITPR3 | PPP1CA | LMAN1 | G3BP | SLC39A6 |
| 21 | GCSH | ACADSB | MAOA | EPRS | RPL5 |
| 22 | DDB2 | PTPLB | ATP6V0B | CKAP1 | RPS13 |
| 23 | TFCP2 | TMEM23 | PPIB | LIG3 | MTHFD2 |
| 24 | TRAM1 | MRPL3 | FMO5 | SNX4 | G3BP2 |
| 25 | YTHDF3 | SLC19A1 | SLC7A5 | NSMAF | UAP1 |

which appear in two of the three lists, although the GEM list does not have element 30 in the expected ranking order. It is further noted that only the median aggregate list contains all the other 5 two-appearance elements, 46–50, and have them in the same order as in the input lists. Indeed, the median aggregate list achieves both the minimum Kendall's and Spearman's criteria, among the four lists from Borda's methods. Further, we can see that Kendall's criterion appears to be less sensitive to the reordering of the elements, while Spearman's criterion leads to quite large differences among the lists.

### *Markov Chain*

The Markov chain algorithm MC1 does not perform as well as some of the simple Borda's method according to the two evaluation criteria. Although elements 21–30 and 46–50 are contained in the aggregate top-40, their relative orders are all jumbled. Aggregate lists from the other two Markov chain methods, MC2 and MC3, produce results that are much closer to the MED aggregate list, the best from Borda's methods. Both lists contain all the 35 elements expected to be in the optimum aggregate list in the correct order. However, the last five elements of MC2 lead to a slight increase over MC3 for Kendall's criterion due to the switching of the orders of two pairs: $(56, 57)$ and $(59, 60)$. On the other hand, the value of Spearman's criterion remains the same even with the switching.

In all three Markov chain procedures, multiple values of the tuning parameter $a$ in the range from 0.01 to 0.15 have been explored, and the results appear to be insensitive to the choice of $a$. Apart from exploring the sensitivity of the Markov chain methods to the specification of the tuning parameter, the use of multiple tuning parameter is to increase the chance of finding the optimum solution, which is strongly recommended. The results reported in Table 2 are the best obtained from the range of the $a$ values explored.

**TABLE 4** | Aggregation results from the heiristic and the stochastic algorithms.

| | Borda | | | | Markov chain | | | Stochastic | |
|---|---|---|---|---|---|---|---|---|---|
| Rank | ARM | MED | GEM | L2N | MC1 | MC2 | MC3 | CEK | CES |
| 1 | HPN | HPN | HPN | HPN | HPN | HPN | HPN | HPN | HPN |
| 2 | AMACR | AMACR | AMACR | AMACR | AMACR | AMACR | AMACR | AMACR | AMACR |
| 3 | GDF15 | FASN | FASN | GDF15 | GDF15 | GDF15 | GDF15 | FASN | FASN |
| 4 | FASN | KRT18 | GDF15 | NME1 | NME1 | FASN | NME1 | GDF15 | GDF15 |
| 5 | NME1 | GDF15 | NME2 | FASN | FASN | KRT18 | FASN | UAP1 | NME2 |
| 6 | KRT18 | NME1 | SLC25A6 | KRT18 | EEF2 | NME1 | EEF2 | OACT2 | UAP1 |
| 7 | EEF2 | EEF2 | EEF2 | EEF2 | KRT18 | EEF2 | KRT18 | KRT18 | OACT2 |
| 8 | NME2 | UAP1 | OACT2 | NME2 | NME2 | UAP1 | UAP1 | SLC25A6 | SLC25A6 |
| 9 | OACT2 | CYP1B1 | OGT | UAP1 | SLC25A6 | OACT2 | NME2 | NME1 | KRT18 |
| 10 | SLC25A6 | ATF5 | KRT18 | OACT2 | UAP1 | NME2 | SLC25A6 | EEF2 | EEF2 |
| 11 | UAP1 | BRCA1 | NME1 | SLC25A6 | OACT2 | SLC25A6 | OACT2 | STRA13 | STRA13 |
| 12 | CANX | LGALS3 | UAP1 | STRA13 | CYP1B1 | OGT | CANX | NME2 | NME1 |
| 13 | GRP58 | MYC | CYP1B1 | CANX | ATF5 | SAT | GRP58 | CANX | CANX |
| 14 | STRA13 | PCDHGC3 | ATF5 | GRP58 | CANX | NACA | STRA13 | SND1 | ALCAM |
| 15 | SND1 | WT1 | CBX3 | SND1 | OGT | LDHA | SND1 | GRP58 | GRP58 |
| 16 | OGT | TFF3 | SAT | ALCAM | BRCA1 | CANX | ALCAM | ALCAM | SND1 |
| 17 | ALCAM | MARCKS | CANX | TMEM4 | MTHFD2 | SLC19A1 | MTHFD2 | PPIB | FMO5 |
| 18 | CYP1B1 | OS-9 | BRCA1 | MTHFD2 | LGALS3 | ANK3 | MRPL3 | TMEM4 | TMEM4 |
| 19 | MTHFD2 | CCND2 | GRP58 | MRPL3 | GRP58 | ALCAM | SLC19A1 | CYP1B1 | CCT2 |
| 20 | ATF5 | DYRK1A | MTHFD2 | PPIB | SND1 | GUCY1A3 | TMEM4 | MTHFD2 | PRKACA |
| 21 | CBX3 | TRAP1 | STRA13 | OGT | CBX3 | SND1 | PPIB | ATF5 | MTHFD2 |
| 22 | SAT | FMO5 | LGALS3 | CYP1B1 | MRPL3 | MTHFD2 | CCT2 | MRPL3 | PTPLB |
| 23 | BRCA1 | ZHX2 | ANK3 | SLC19A1 | MYC | TMEM4 | FMO5 | BRCA1 | PPIB |
| 24 | MRPL3 | RPL36AL | GUCY1A3 | ATF5 | STRA13 | MRPL3 | OGT | LGALS3 | MRPL3 |
| 25 | LGALS3 | ITPR3 | LDHA | CBX3 | ALCAM | GRP58 | CYP1B1 | MYC | SLC19A1 |

### *Cross Entropy Monte Carlo*

For the two stochastic search algorithms based on CEMC that directly minimizes the generalized Kemeny criteria either based on the Kendall's distance (CEK) or the Superman's distance (CES), the results are at least as good as the best achieved by the Borda's or the Markov chain algorithms. It is interesting to see that CEK has a smaller Kendall's distance than any of the other aggregate lists. In fact, various reordering of the last few elements of CEK will lead to the same value of Kendall's distance. There are a number of tuning parameters in the OEA algorithm as discussed earlier; it would be a good idea to run the algorithm with multiple sets of tuning parameters just as recommended for running the Markov chain methods. In this example, multiple runs have led to different aggregate lists with the same value of the optimization criteria, signifying the existence of multiple modes. The results reported in Table 2 are one of those that minimizes the criteria.

## Prostate Cancer Gene Expression Studies

We apply the heuristic and the CEMC optimization algorithms to aggregate results from five microarray studies which aimed at finding genes that are differentially expressed between prostate cancers and normal prostate tissues.[38–42] Despite their common goal, these five studies differ in the laboratories in which the experiments were carried out, in the materials and in the microarray platforms used, which make it difficult to perform a 'low-level' data integration analysis given that the raw measures are not directly comparable. Instead, we apply the 'high-level' meta-analytic method presented in this section to combine the five lists of top-25 up-regulated genes

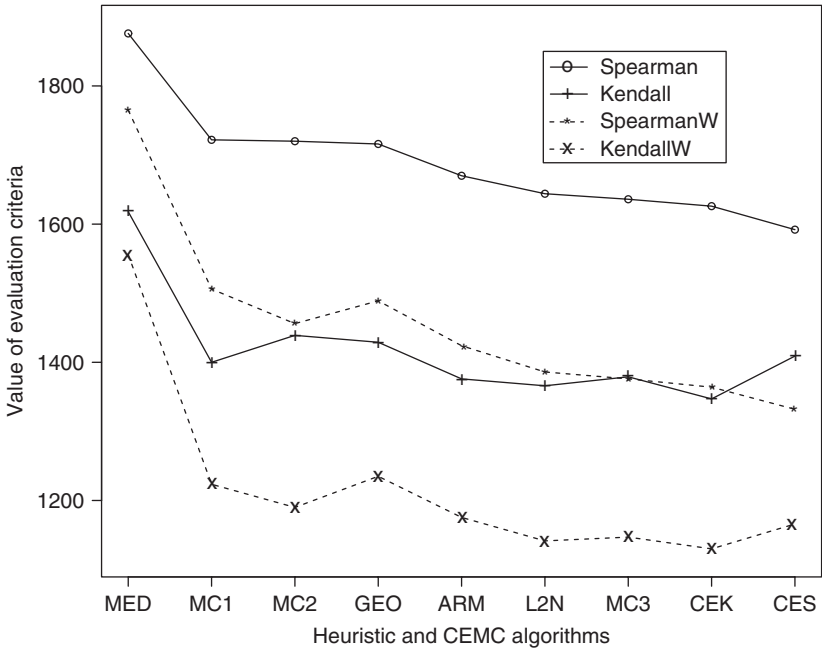



**FIGURE 5** | Evaluations of four optimization criteria for the aggregate lists: Spearman or SpearmanW: Spearman's criterion with all studies assigned the same weight of 1 or with the Luo study downweighted to half, respective; Kendall or KendallW: Kendall's criterion with all studies assigned the same weight of 1 or with the Luo study downweighted to half, respective.

identified from each of the individual studies. This data set was used by DeConde et al.[2] to study the Markov chain methods (variations of MC2 and MC3 algorithms presented in this article) and by Lin and Ding[5] to study the CEMC methods. Table 3 present the rankings and the top-25 ranked genes found to be up-regulated in prostate tumors compared to normal prostate tissues from five studies labeled in the column headings. As can be seen from the table, the results are quite discrepant in the genes selected to be included in the top-25's among the five studies, with the gene list from Luo et al.[39] being the least common with the other four studies. As such, one might wish to down weigh the contribution of the Luo study in forming the integrated list. This can be done in the stochastic optimization algorithm and the results presented in Figure 5 include some with the weight for the Luo study being half compared to the weights for the other studies.

The results from the heuristic and the stochastic algorithms are given in Table 4. There are general agreements among many of the lists. For example, all nine algorithms rank HPN and AMACR as the top-2 genes, where HPN is present in all five individual top-25 lists while AMACR is present in four of them. However, it can be easily seen that discrepancies among the aggregate lists exist. Figure 5, which plots the results for all nine algorithms in terms of four evaluation criteria (two Kendall's and two Spearman's, one with equal weights and one with the Luo study downweighted to half for each of the Kendall's and the Spearman's criteria) show that CEK performed better than any of the deterministic algorithms. CES, on the other hand, performs the best among all methods when evaluated under the Spearman's criterion, but was outperformed by some of the deterministic algorithms in addition to CEK under the Kendall's criterion. However, it is reassuring to see that CES and CEK do give the smallest value of the Spearman's and Kendall's criterion, respectively, indicating that the stochastic search algorithm is indeed performing reasonably well. Finally, we note that when evaluated under the Kendall weighted or the Spearman weighted criteria, the CEK or CES aggregates under the weighted criterion was used. On the other hand, since there is no obvious way to incorporate weighting into Borda's nor Markov chain based algorithms, the results shown are the results from the unweighted aggregates evaluated under the weighted criteria.

## CONCLUSION

This article presents three broad categories of rank aggregation algorithms: distribution based, heuristic, and stochastic optimization. The two examples (three long lists and prostate cancer studies) are both of the type of a few long lists and therefore only the heuristic and the stochastic algorithms were used in the analyses to demonstrate the performances of the methods. This is because the distribution based algorithm presented, the Thurstone's method, is more suitable for the problem of aggregating many short lists. As an aside, we note that although none of our

examples is on search engine aggregation, the number of search engines or evaluation criteria are typically much smaller compared to the 'matching' results, and as such, the heuristic and the stochastic optimization algorithms are also the relevant methods for this type of applications. In both examples considered, the two stochastic optimization algorithms based on CEMC generally perform well compared to the heuristic algorithms. Further, some of the simple scoring methods based on Borda's formulation performed surprisingly well compared to the Markov chain based algorithms. Given that the heuristic algorithms are all computationally very efficient, we recommend that all algorithms be run to increase one's chance of finding the global optimum. We further recommend that multiple sets of tuning parameters (for the Markov chain and the CEMC algorithms) be used, similar to the recommendation for running the EM algorithm.[43]

It is worth re-emphasizing the importance of the assumption about the underlying spaces from which the top-*k* lists come. This point is rarely explicitly discussed in most papers, but the implicit assumption can have a profound impact on the outcomes. A case in point is the two Markov chain based algorithms presented by DeConde et al.[2] DeConde's algorithms as presented there implicitly assume that the underlying spaces are all different to the extreme that each top-*k* list is in fact treated as a 'full-list' (but note that more information was used for the prostate cancer example in the DeConde paper). This would be a reasonable assumption if the ranked lists to be aggregated were treated as partial ranked lists (not top-*k* lists) that ranked all the elements considered. In other words, if each study only considered a subset of elements in the underlying space and ranked all the elements considered, then each list may be deemed as a full ranked list. However, for aggregating top-*k* lists, this implicit assumption wipes out the information implied by the virtue of 'top k': an element that is not being ranked in the top-*k* list but is contained in the space from which the top-*k* elements come is implicitly ranked lower than *k*. Such valuable information is completely ignored in DeConde's algorithms. Ignoring this information can be costly: running both datasets using DeConde's algorithms lead to much inflated values of the evaluation criteria. In fact, even the results from DeConde et al.[2] that used more information than the five top-25 lists lead to larger values of the Kendall and Spearman's criteria,[5] indicative of sub-optimal solutions.

## REFERENCES


1. Conlon EM, Song JJ, Liu A. Bayesian meta-analysis models for microarray data: a comparative study. *BMC Bioinformatics* 2007, 8:80.
2. DeConde RP, Hawley S, Falcon S, Clegg N, Knudsen B, Etzioni R. Combining results of microarray experiments: a rank aggregation approach. *Stat Appl Genet Mol Biol* 2006, 5:15.
3. Fishel I, Kaufman A, Ruppin E. Meta-analysis of gene expression data: a predictor-based approach. *Bioinformatics* 2007, 23:1599–1606.
4. Hall P, Schimek MG. *Inference for the top-k rank list problem*, Technical Report; 2009.
5. Lin S, Ding J. Integration of ranked lists via cross entropy Monte Carlo with applications to mRNA and microRNA studies. *Biometrics* 2009, 65:9–18.
6. Liu HC, Chen CY, Liu YT, Chu CB, Liang DC, Shih LY, Lin CJ. Cross-generation and cross-laboratory predictions of Affymetrix microarrays by rank-based methods. *J Biomed Inform* 2008, 41:570–579.
7. Tan AC, Naiman DQ, Xu L, Winslow RL, Geman D. Simple decision rules for classifying human cancers from gene expression profiles. *Bioinformatics* 2005, 21:3896–3904.
8. Yuen T, Wurmbach E, Pfeffer RL, Ebersole BJ, Sealfon SC. Accuracy and calibration of commercial oligonucleotide and custom cDNA microarrays. *Nucleic Acids Res* 2002, 30:e48.
9. Choi JK, Yu U, Kim S, Yo OJ. Combining multiple microarray studies and modeling interstudy variation. *Bioinformatics* 2003, 19:i84–i90.
10. Xu L, Tan AC, Naiman DQ, Geman D, Winslow RL. Robust cancer marker genes emerge from direct integration of inter-study microarray data. *Bioinformatics* 2005, 21:3905–3911.
11. Thurstone LL. A law of comparative judgment. *Psychol Rev* 1927, 34:273–286.
12. Dwork C, Kumar R, Naor M, Sivakumar D. Rank aggregation methods for the web. In: *Proceedings of the 10th International World Wide Web Conference, New York*; 2001, 613–622.
13. Thurstone LL. Rank order as a psychophysical method. *J Exp Psychol* 1931, 14:187–201.
14. Thurstone LL. *The Measurements of Values*. Chicago, IL: University of Chicago Press; 1959.

15. Thurstone LL, Jones LV. The rational origin for measuring subjective values. *J Am Stat Assoc* 1957, 52:458–471.
16. Bock RD, Jones LV. *The Measurement and Prediction of Judgment and Choice*. San Francisco: Holden-Day; 1968.
17. Green PE, Tull DS. *Research for Marketing Decisions*. New Jersey: Prentice-Hall; 1978.
18. Conklin M, Lipovetsky S. *Efficient assessment of self-explicated importance using latent class Thurstone scaling*. In: The 10th Annual Advanced Research Techniques Forum. Santa Fe, New Mexico: American Marketing Association; 1999.
19. Mosteller F. Remarks on the method of paired comparisons. *Psychometrika* 1951, 16:203–218.
20. Daniels HE. Rank correlation and population models. *J R Stat Soc Ser B* 1950, 12:171–181.
21. Glenn WA, David HA. Ties in paired-comparisons experiments using a modified Thurstone-Mosteller model. *Biometrics* 1960, 16:86–109.
22. David HA. *The Method of Paired Comparisons*. 2nd ed. London: Griffin; 1988.
23. Stern H. Models for distributions on permutations. *J Am Stat Assoc* 1990, 85:558–564.
24. Ennis DM, Johnson NL. Thurstone-Shepard similarity models as special cases of moment generating functions. *Math Psychol* 1993, 37:104–110.
25. Maydeu-Olivares A. Thursonian modeling of ranking data via mean and covariance structure analysis. *Psychometrika* 1999, 64:325–340.
26. Maydeu-Olivares A, Bockenholt U. Structural equation modeling of paired comparison and ranking data. *Psychol Methods* 2005, 10:285–304.
27. Borda JC. *Memoire sur les elections au scrutin*. Histoire de l'Academie des Sciences; 1781.
28. Emerson P. *Designing an All-Inclusive Democracy. Part 1: Collective Decision-making: The Modified Borda Count, MBC*. Berlin: Springer Verlag; 2007, 15–38. ISBN 978-3-540-33163-6 (Print) 978-3-540-33164-3 (Online).
29. Saari DG. *Chaotic Elections! A Mathematician Looks at Voting*. Providence, RI: American Mathematical Society; 2001.
30. Young HP. Condorcet's theory of voting. *Am Polit Sci Rev* 1988, 82:1231–1244.
31. Marden JI. *Analyzing and modeling rank data*. Monographs on Statistics and Applied Probability, vol. 64. London: Chapman & Hall; 1995.
32. Kendall M. *Rank Correlation Methods*. 4th ed. London: Griffin; 1970.
33. Diaconis P, Graham RL. Spearman's footrule as a measure of disarray. *J R Stat Soc B* 1977, 39:262–268.
34. Fagin R, Kumar R, Sivakumar D. Comparing top k lists. *SIAM J Disc Math* 2003, 17:134–160.
35. Rubinstein RY, Kroese DP. *The Cross-Entropy Method. A Unified Approach to Combinatorial Optimization, Monte-Carlo Simulation and Machine Learning*. UK: Springer; 2004.
36. Margolin L. On the convergence of the cross-entropy method. *Ann Oper Res* 2005, 134:201–214.
37. Liu Z, Lin S, Tan M. Genome-wide tagging SNPs with entropy-based Monte Carlo methods. *J Comput Biol* 2006, 13:1606–1614.
38. Dhanasekaran SM, Barrette TR, Ghosh D, Shah R, Varambally S, Kurachi K, Pienta KJ, Rubin MS, Chinnaiyan AM. Delineation of prognostic biomarkers in prostate cancer. *Nature* 2001, 412:822–826.
39. Luo J, Duggan DJ, Chen Y, Sauvageot J, Ewing M, Bittner ML, Trent JM, Isaacs WB. Human prostate cancer and benign prostatic hyperplasia: molecular dissection by gene expression profiling. *Cancer Res* 2001, 61:4683–4688.
40. Singh D, Febbo PG, Ross K, Jackson DG, Manola J, Ladd C, Tamayo P, Renshaw AA, Amico AV, Richie JP, et al. Gene expression correlates of clinical prostate cancer behavior. *Cancer Cell* 2002, 1:203–209.
41. True L, Coleman I, Hawley S, Huang A, Gifford D, Coleman R, Beer T, Gelman E, Datta M, Mostaghel E, et al. A molecular correlate to the gleason grading system for prostate adenocarcinoma. *Proc Natl Acad Sci USA* 2006, 103:10991–10996.
42. Welsh JB, Sapinoso LM, Su AI, Kern SG, Wang-Rodriguez J, Moskaluk CA, Frierson HF Jr., Hampton GM. Analysis of gene expression identifies candidate markers and pharmacological targets in prostate cancer. *Cancer Res* 2001, 61:5974–5978.
43. McLachlan G, Peel D. *Finite Mixture Models*. New York: John Wiley & Sons; 2000.