Information Sciences 278 (2014) 466–487

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Reaching a joint decision with minimal elicitation of voter preferences

Lihi Naamani Dery *, Meir Kalech, Lior Rokach, Bracha Shapira

Information Systems Engineering, Ben Gurion University of the Negev, Israel

ARTICLE INFO

Article history:
Received 5 August 2012
Received in revised form 18 February 2014
Accepted 5 March 2014
Available online 25 March 2014

Keywords:
Preference elicitation
Range voting
Social choice

ABSTRACT

Sometimes voters are required to reach a joint decision and find an item that best suits the group's preferences. Voters may wish to state preferences only when necessary, particularly in cases where there are many available options, therefore it is unpractical to assume that all voter preferences are known at all times. In order to elicit voter preferences at a minimal cost, a preference elicitation process is required. We introduce a general approach for reaching a joint decision with minimal elicitation of voter preferences. The approach is probabilistic and uses voting rules to find a necessary winning item which is presented to the group as their best option. We propose computing a voter-item probability distribution and developing methods based on this distribution that can then determine which voter-item pair to query. Computing the optimal minimal set of voter-item queries is computationally intractable; therefore we propose novel heuristic algorithms, named DIG and ES, which proceed iteratively until the identification of a winning item. The probabilistic voting distribution is updated as more information is revealed. Experiments on simulated data examine the benefits of each of the algorithms under different settings. Experiments with the real-world Netflix data show that the proposed algorithms reduce the required number of ratings for identifying the winning item by more than 50%.

1. Introduction

Joint decisions are often required in different aspects of everyday life. These decisions range from a group of friends who wish to choose which movie to watch to a faculty acceptance committee that need to choose a Ph.D. or a new faculty candidate to accept out of many applicants, as described in [33]. When voters state all of their preferences, a consensus can be reached by different voting protocols like Range voting, Majority, Borda, Copeland, etc. However, requiring the voters to state complete preferences may be viewed as disruptive by the voters as this can be difficult or time consuming. Furthermore, this requirement is sometimes completely impractical due to the communication burden [5,15] or to voter limitations. Consider, for example a meeting scheduling application whose purpose is to set a time for a conference [13], or an application selecting movies for a group to watch, such as Netflix (www.netflix.com). It is impractical to ask the voters for their preferences on all available options as there might be hundreds available.

Another scenario where decision making with partial information is necessary occurs when group members are required to make a choice, but due to the amount of possibilities and their limited resources, members cannot process all of the

* Corresponding author. Fax: +972 8 647936.
E-mail address: ln@bgu.ac.il (L. Naamani Dery).

http://dx.doi.org/10.1016/j.ins.2014.03.065
0020-0255/

information. For example, we expand the scenario described in [33] of the faculty acceptance committee.[1] A committee assembles in order to choose a fixed number of candidates to accept for a Ph.D. program out of hundreds of applicants. Ideally, a group decision requires each committee member to express his or her opinion about each of the applicants and then a joint decision is reached based on all opinions. As their time is limited, the committee members do not have the resources for such a process. Instead, each member reads a portion of the total requests and then grades her assigned applicant files on a scale from 1 to 5. Applicants in the high and low range of scores do not require any special care as they are immediately accepted or declined; it is those who receive an average rating which present a problem. Their files need to be reread by other members in order to receive additional opinions. The question is then, which committee member should be assigned which file to read so that a decision will be reached in optimal time. Assuming we know the member's previous grading pattern, e.g., members who tend to grade high/low as opposed to members who give all candidates average scores, we can define the member's grading probability distribution. Then *voter-item query pairs* can be selected: the voter being the committee member, the item being the applicant assigned to her for grading and the query being a request for the voters rating for this item. Therefore, the goal is to reduce the number of interactions with the voter needed in order to reach a verdict by selecting specific voter-item query pairs.

In order to reduce the amount of queries sent to the voter during the elicitation process, we propose an incremental elicitation process and determine a winning item based only on partial information from the voters. In this process, voters are queried only for some preferences, since as [15] showed, not all preferences are needed in order to reach a winning item. We assume the voters' preferences are unknown in advance, but can be acquired during the process, i.e., a voter queried for her preferences on an item is able to submit them. Uncertainty regarding the voter preferences is facilitated by creating and updating voter-item probability distributions. We focus specifically on Range voting [31], where voters are asked to assign a rating within a specified range for the items. Range voting is relevant to many already existing applications where voters are asked to rate items on a specified scale (e.g., recommender systems at Netflix and Amazon). The ratings for each item are added up and the item with the highest score is the winner. This item is presented as the joint decision.

Previous studies have reported the theoretical upper and lower bounds of the required communication with the voters [5]. However, to the best of our knowledge, only two studies propose practical algorithms for preference elicitation [13,20]. The first assumes each voter holds a predefined decreasing order of the preferences where the voters are requested to submit their highest preferences in an iterative process. Requiring the users to predefine their preferences can be inconvenient to the users. Therefore, we require voters to respond and rate a specific item only when necessary. Furthermore, the procedure in [13] does not necessarily reduce communication significantly. The reason is that while in our approach each request is for a rating of a specific item from a specific user, in Kalech et al. [13] the request is for the rating of one item from all the users. In the second study [20] a practical elicitation process is proposed for the Borda voting protocol. In an iterative process users are queried for their preference between a pair of items. The underlying assumption is that a voter necessarily prefers one item over the other, and voters are requested to hold a strict set of preferences where no two items are equally preferred. We relax this assumption and allow voters to submit the same rating for many items. We focus on the Range voting protocol rather than on the Borda protocol. The Range protocol is very common in Recommender systems settings; it requires users to submit a score. For example, the famous Netflix prize competition (http://www.netflixprize.com) provided explicit user ratings suitable for the range voting protocol. Users are familiar with applications that ask for their score on an item, another example being Amazon (www.amazon.com).

Since computing the optimal minimal set of queries that are required to determine a winner is computationally intractable due to the combinatorial space of queries' orders, this paper proposes two novel heuristic approaches for determining a winner. As we show, each approach has an advantage under different circumstances. Both approaches proceed iteratively, selecting a voter-item pair and querying the selected voter for her score on the item. To determine a voter-item query pair, the first algorithm, named DIG, heuristically computes the information gain of each potential query based on the entropy of the item's probability to win. The query that maximizes the information gain is selected. The second algorithm, named ES, uses the probability distribution of the voters' preferences to select the candidate most likely to win and the voter that is expected to maximize the score of that item. In both algorithms, voter-item probability distributions are computed and updated as new information is revealed. This is achieved by computing a non-parametric probability distribution for each voter's preferences of items, i.e., for voter-item pairs. The algorithms output a necessary winner item [15], i.e., a definite winner.

Experiments on a simulated meeting schedule domain as well as on two real-world datasets (Netflix contest dataset and Sushi dataset) highlight that both DIG and ES algorithms reduce communication time while guaranteeing that a winning candidate will be found. For example, in the real-world Netflix contest dataset, we show that the communication cost can be cut by more than 50%. Furthermore, the analysis of the heuristics under different settings provides some interesting insights. For example, DIG heuristic excels when the data is noisy and there is no typical skewness pattern, while ES runs faster and performs better when some assumptions can be made on the voter-item distribution patterns.

The current paper is an extension of the authors' previous paper [26]. The previous paper presented novel algorithms for group decision and voter querying. Here, we add a formal problem definition in Section 3. In our previous work we merely assumed existence of probability distributions. Now, in Section 6, we present and demonstrate a method for calculating

[1] This example is taken from a private discussion with Vincet Conitzer in 2011.