

Cluster algorithm for social choice

Emanuele d’Ajello, Davide Formica, Elio Masciari, Gaia Mattia, Arianna Anniciello, Cristina Moscariello, Stefano Quintarelli and Davide Zaccarella

The 10th edition of the Workshop New Frontiers in Mining Complex Patterns (NFMCP) in conjunction with the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML-PKDD 2022) Grenoble, France.

Abstract. In order to overcome the classical methods of judgement, in the literature there is a lot of material about different methodology and their intrinsic limitations. One of the most relevant modern model to deal with votation system dynamics is the Majority Judgement.

It was created with the aim of reducing polarization of the electorate in modern democracies and not to alienate minorities, thanks to its use of a highest median rule, producing more informative results than the existing alternatives. Nonetheless, as shown in the literature, in the case of multiwinner elections it can lead to scenarios in which minorities, albeit numerous, are not adequately represented.

For this reason our aim is to implement a clustered version of this algorithm, in order to mitigate these disadvantages: it creates clusters taking into account the similarity between the expressed judgements and then for, each of these created groups, Majority Judgement rule is applied to return a ranking over the set of candidates. These traits make the algorithm available for applications in different areas of interest in which a decisional process is involved.

Keywords: Decision Making · Social Choice · Cluster · Majority Judgement · K-Medoids.

1 Introduction

Voting rules are different and behave differently according to their limitations or sometimes paradoxal traits. Asking for a more inclusive democracy also represents a modern citizens’ quest, but what does exactly it mean? First of all, we want to underline why a majority voting system embodies the best option between the classical judgement methods.

Consider three agents who express their binary judgement (“Yes” or “No”) for two statements A , B , $A \wedge B$ and $A \longleftrightarrow B$. Premised-based rule take majority decisions on A and B and then infers conclusions on the other two propositions. As shown in the table 1, results are quite different based on the used rule.

We now focus on Agent 2 case: he’s represented in just one of the single proposition (A), and his judgement doesn’t agree with the outcome, in the other cases.

So, a huge liability of this model could appear: Agent 2 could think about manipulating the outcome, pretending a disagreement for A. The premised model reacts by providing as final outcome on 3 agents' votation a "No" for both $A \wedge B$ and $A \longleftrightarrow B$, as originally expressed by Agent 2.

	A	B	$A \wedge B$	$A \longleftrightarrow B$
Agent 1	Yes	Yes	Yes	Yes
Agent 2	Yes	No	No	No
Agent 3	No	Yes	No	No
Premised rule	Yes	Yes	Yes	Yes
Majority	Yes	Yes	No	No

Table 1. Three agent case of voting

In such a way, a strategical approach on voting could lead to a deviation effect, providing as result the best tricker's choice.

Looking at the table, we can also highlight another paradoxal aspect: considering majority-based outcome, the latest two propositions are inconsistent with "Yes" value assigned to both A and B.

This is known as *discursive dilemma* and deals with inconsistency problem in judgement aggregation based on majority rule [10].

Both premised and majority rule present drawbacks, but the latter has one important feature: it doesn't suffer from deficiency shown by the first, so that, if an Agent care about the number of propositions agreeing with his own judgement, then it is always in his best interest to report his true preference. For this reason we focus our attention on majority rule as a transparent asset in decisional process, while trying to deal with its intrinsic problems related to judgement aggregation [11].

Our attempt is not aimed to solve above-mentioned dilemma, rather joining a more refined majority rule (*Majority Judgement*) with cluster approach's advantages in aggregating similar patterns.

2 Majority Judgement

2.1 Formal aspects

To introduce social choice theory formally, consider a simple decision problem: a collective choice between two alternatives. The first involves imposing some 'procedural' requirements on the relationship between individual votes and social decisions and showing that majority rule is the only aggregation rule satisfying them. May (1952) [9] [32] introduced four such requirements for majority voting rule must satisfies:

- **Universal domain:** the domain of admissible inputs of the aggregation rule consists of all logically possible profiles of votes $\langle v_1, v_2, \dots, v_n \rangle$, where each $v_i \in [-1, 1]$ (to cope with any level of ‘pluralism’ in its inputs);
- **Anonymity:** applying any kind of permutation on individual preferences does not affect the outcome (to treat all voters equally), i.e.,

$$f(v_1, v_2, \dots, v_n) = f(w_1, w_2, \dots, w_n) \quad (1)$$

- **Neutrality:** each alternative has the same weight and for any admissible profile $\langle v_1, v_2, \dots, v_n \rangle$, if the votes for the two alternatives are reversed, the social decision is reversed too (to treat all alternatives equally), i.e.

$$f(-v_1, -v_2, \dots, -v_n) = -f(v_1, v_2, \dots, v_n) \quad (2)$$

- **Positive responsiveness:** For any admissible profile $\langle v_1, v_2, \dots, v_n \rangle$, if some voters change their votes in favour of one alternative (say the first) and all other votes remain the same, the social decision does not change in the opposite direction; if the social decision was a tie prior to the change, the tie is broken in the direction of the change, i.e., if $[w_i > v_i \text{ for some } i \text{ and } w_j = v_j \text{ for all other } j]$ and $f(v_1, v_2, \dots, v_n) = 0$ or 1 , then $f(w_1, w_2, \dots, w_n) = 1$.

The May theorem (Theorem: “An aggregation rule satisfies universal domain, anonymity, neutrality, and positive responsiveness if and only if it is majority rule”) provides an argument for the majority rule based on four plausible procedural desires and the theorem helps us characterize other aggregation rules in terms of which desiderata they violate.

But that’s with regards to binary choice. Now, we consider a set $N = [1, 2, \dots, n]$ of individuals ($n \geq 2$). Let $X = [x, y, z, \dots]$ be a set of social alternatives, for example possible policy platforms, election candidates, or other[8]. Each individual $i \in N$ has a preference ordering R_i over these alternatives that represents a complete and transitive binary relation on X . For any $x, y \in X$, xR_iy means that individual i weakly prefers x to y . We write xP_iy if xR_iy and not yR_ix (‘individual i strictly prefers x to y ’), and xI_iy if xR_iy and yR_ix (‘individual i is indifferent between x and y ’). But we must specify that at the heart of social choice theory is the analysis of preference aggregation [33], understood as the aggregation of several individuals’ preference rankings of two or more social alternatives into a single, collective preference ranking (or choice) over these alternatives [7]. In case of many successful alternatives, we need a more sophisticated model to deal with preferences’ aggregation [6]. A multi-winner election (V,C,F,k) is defined by a set of voters V expressing preferences over a number of candidates C , and then a voting rule F returns a subset of size k winning candidates. A voting rule can perform its role on different types of ordered preferences, even though the most common refers to a pre-fixed linear order on the alternatives. In most of cases, these are chosen *a priori*.

Formally we denote set of judgements performed by the i -th voter as profile preferences P_i . Each profile contains information about the grade of candidates by voters. The voting rule F associates with every profile P a non-empty subset

of winning candidates.

In multi-winner elections more precise traits are required, compared to the ones stated in May's theory [12]. Indeed:

- **Representation:** for each partition of voters

$$V_i \in V(\text{with } |V_i| \geq \lfloor \frac{n}{k} \rfloor) \quad (3)$$

at least one successful candidate is elected from that partition;

- **Proportionality:** for each partition of voters

$$V_i \in V(\text{with } |V_i| \geq \lfloor \frac{n}{k} \rfloor) \quad (4)$$

number of elected candidate is proportional to the partition's size.

An implicit assumption so far has been that preferences are ordinal and not interpersonally comparable: preference orderings contain no information about each individual's strength or about how to compare different individuals' preferences with one another. Statements such as 'Individual 1 prefers alternative x more than Individual 2 prefers alternative y' or 'Individual 1 prefers a switch from x to y more than Individual 2 prefers a switch from x* to y*' are considered meaningless. In voting contexts, this assumption may be plausible, but in welfare-evaluation contexts—when a social planner seeks to rank different social alternatives in an order of social welfare—the use of richer information may be justified.

2.2 Single-winner Majority judgement

In order to describe the majority judgement, we need to use a table that refers to ranking for all the candidates C , by using tuples [5]. Suppose having six possible choices we may use the words: *excellent*, *very good*, *good*, *discrete*, *bad*, *very bad*. So each candidate is described by a bounded set of vote.

In general, letting $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_n)$ be a candidate A's set of n grades (written from highest to lowest, $\alpha_i \geq \alpha_{i+1}$ for all i), there is a majority of (at least) $n-k+1$ for n A's grade to beat most α_k and at least α_{n-k+1} , for all $1 \leq k \leq \frac{(n+1)}{2}$. We call this the $(n-k+1)$ - majority for $[\alpha_k, \alpha_{n-k+1}]$.

As already mentioned any possible ranking tuple that we choose to describe must follow ordering relations.

So the ranking should respect domination: namely, evaluate one candidate above another when that candidate's grades dominate the other's.

The described majority judgement is a single winner system, found comparing recursively median grade between candidates: first, grades are ordered in columns from the highest to the lowest according to the order relation, then the middle column (lower middle if number of grades are even) with the highest grade between candidates' row is selected. If there's a tie, algorithm keeps on discarding

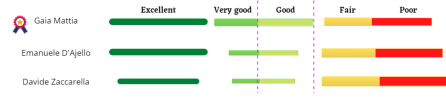


Fig. 1. Example with 5 grades, between the dashed lines it's reported the median grade. Highest occurrences in "Good" determines the winner.

grades equal in value to the shared median, until one of the tied candidate is found to have the highest median. Before describing how it's possible to generalize this single winner system to a multi winner strategy, thanks to the use of clusters, we focus our attention on how these works, analyzing in particular *K-medoids*.

3 Clustering approach

3.1 How clusters work

There's no precise definition of clustering, mostly due to the huge variety in different clustering algorithms. We can state that they share the ability to divide data into groups with some common features. According to some general traits, we can distinguish types of clustering:

1. **Connectivity models:** data points in a sample space exhibits similarity according to the distance between them. Two approaches are equally valid: *bottom-up* where each observation constitutes a group and then pairs of clusters are merged; *top-down*, where observation are included in one cluster and then it's segregated; but in both approaches is not included the possibility of modifying a cluster once created;
 2. **Distribution models:** once created a cluster, model check probabilities on observations following a particular distribution. Good performances are not always guaranteed since these models are prone to overfit data if no constraint on complexity is made;
 3. **Density models:** areas of higher density are identified and local cluster are there created, while remaining data can be grouped into arbitrary shaped region, with no assumption about data distribution; for their flexibility, these models are fit to handle noise better than organizing data on fixed required body.
- Since we would like to model clusters that satisfy requirements expressed before, based on pretty fixed structure with no assumption about distribution followed by data, it seems more accurate considering a different class of clustering algorithm known as *centroid models*.

3.2 K-Medoids

Clustering is the process of grouping a set of objects in order to have each similar object to each other in one cluster, that are dissimilar to objects in other clusters.

For our goal, namely selecting winners from a group of candidates, *K-medoids* clustering are used, because medoids are the representative objects that are considered, in order to have a result that belongs to the group of candidates: it is based on the most centrally located object in a cluster, so it is less sensitive to outliers in comparison with the K-means clustering, which is not the best model in our case since it could result in something that is not present in the candidate list due to the fact that is an average-based method rather than median. In fact, the medoid is a data point (unlike the centroid) which has the least total distance to the other members of its cluster.

Another advantage for this choice is that the mean of the data points is a measure that gets highly affected by the extreme points; so, in K-Means algorithm, the centroid may get shifted to a wrong position and hence result in incorrect clustering if the data has outliers because then other points will move away from. On the contrary, the K-Medoids algorithm is the most central element of the cluster, such that its distance from other points is minimum. Thus, K-Medoids algorithm is more robust to outliers and noise than K-Means algorithm.

The K-medoid we use is part of the python `sklearn` library [13], which is oriented to machine learning. This library supports *partitioning around medoids* (PAM) [2] proposed by Kaufman and Rousseeuw (1990), that is known to be most powerful. The workflow of PAM is described below [1].

The PAM procedure consists of two phases: *BUILD* and *SWAP*:

- In the BUILD phase, primary clustering is performed, during which k objects are successively selected as medoids.
- The SWAP phase is an iterative process in which the algorithm makes attempts to improve some of the medoids. At each iteration of the algorithm, a pair is selected (medoid and non-medoid) such that replacing the medoid with a non-medoid object gives the best value of the objective function (the sum of the distances from each object to the nearest medoid). The procedure for changing the set of medoids is repeated as long as there is a possibility of improving the value of the objective function.

Suppose that n objects having p variables each should be grouped into k ($k < n$) clusters, where k is known. Let us define j -th variable of object i as X_{ij} ($i = 1, \dots, n; j = 1, \dots, p$). As a dissimilarity measure is used the Euclidean distance, that is defined, between object i and object j , by:

$$d_{ij} = \sqrt{\sum_{a=1}^p (X_{ia} - X_{ja})^2} \quad (5)$$

where i and j range from 1 to n . The medoids is selected in this way:

- calculate the Euclidean distance between every pair of all objects;
- calculate $v_j = \sum_{i=1}^n \frac{d_{ij}}{\sum_{l=1}^n d_{il}}$;
- sort all v_j for $j = 1, \dots, n$ in ascending order and select the first k object that have smallest initial medoids value;

- from each object to the nearest medoid we can obtain the initial cluster result;
- calculate the sum of distances from all objects to their medoids;
- update the current medoid in each cluster by replacing with the new medoid, selected minimizing the total distance from a certain object to other objects in its cluster;
- assign each object to the nearest medoid and obtain the cluster result;
- calculate the sum of distance from all objects to their medoids, so if the sum is equal to the previous one, then stop the algorithm; otherwise, go back to the update step.

In our case, prior knowledge about the number of winners is required, and identified clusters are restricted in minimum size that is number of voters on the number of candidates $\binom{n}{k}$.

3.3 Clustered Majority Judgement

Multi winner majority judgement exploits clustering approach to apply to each group majority judgement [4]. Given k the number of candidates to be elected, algorithm seeks the optimal number of cluster to create.

This ranges from 1 to k and has to satisfy an important additional requirement: once selected a number of clusters, if a tie occurs and so k' vacant seats are left, algorithm is repeated k' times until tie's broken. In case there's no broken tie, fixed number of cluster is changed.

3.4 Algorithm

In order to explain how the algorithm deals with polarization problem, most relevant steps are described in pseudocode and in annotated strides:

1. set the number of winners as maximum number of clusters;
2. cluster are created decreasing the previous maximum number of clusters until the optimal number is not achieved. This number is bound by the size of cluster, that satisfies the following proportion: *number of voters : number of winners = number of voters in one cluster : one winner* (line 8 in pseudocode);
3. the function *winners* calculates the median for every created cluster (line 15 of pseudocode);
4. check that winners from cluster are different between each other (line 29 in pseudocode); in case it's not true (condition="ko" on pseudocode) algorithm goes back to step 2 with a maximum number of cluster equal to number of vacant seats and the proceedings are held until all seats have been filled.

Algorithm 1

Require: $k \geq 0$ **Ensure:** $n_winners = (n_1, \dots, n_k), k > 1$ $k \leftarrow number_winners$ $max_cluster \leftarrow k$ $condition \leftarrow "ko"$ **while** $condition = "ko"$ **do** $cluster_list \leftarrow cluster(vote_list)$ **for all** $list_cluster$ **do** $winners_per_cluster \leftarrow compute_winners(cluster)$ $all_winners \leftarrow list_of_all_winners(winners_per_cluster)$ **end for** $list_winner_distinct = list_of_all_distinct_winners(all_winners)$ $option_remaining \leftarrow number_winners - len(list_winner_distinct)$ **if** $option_remaining = 0$ **then** $condition = 'ok'$ **else** $k \leftarrow option_remaining$ $condition \leftarrow 'ko'$ **end if****end while**

3.5 Case Studies

In this section, we describe two interesting comparisons of majority judgement (MJ) and clustered majority judgement (CMJ).

Case study 1: President of the Republic election In order to test our algorithm, we asked an heterogeneous group of voters to express judgements on a pre-defined list of possible candidates as President of the Republic before the elections took place. This list has been created according to the rumours circulated on that period, creating a bias effect on our results, as it was excluded a possible reelection of Sergio Mattarella.

In spite of it, we focus on how the algorithm has worked in order to balance polarization, returning a subset of winners with size chosen *a priori*, that we may interpret as best solutions for majority of people who took part into the venture.

Input parameters of Clustered Majority Judgement test are *Excellent*, *Very Good*, *Good*, *Acceptable*, *Poor*, *To Reject*, *No Opinion* and the number of winners is set a priori equal to 3. 125 voters took part into this election and the algorithm form three clusters, exactly like the number of winners.

Testing our algorithm on the described election has shown how difference preference has a leverage on judgement aggregation: for example, voters in Cluster 1 are more bound to express "Good" judgement for candidates considered neutral in terms of political ideas, than the cluster 3 in which voters have a tendency

in judging neutral ones as "Fair" or "Poor". Cluster 2 has intermediate traits and no particular tendency is emphasized. We can compare CMJ results with

Cluster	Cluster size	Winner
Cluster 1	65	Mario Draghi
Cluster 2	35	Paolo Gentiloni
Cluster 3	25	Anna Finocchiaro

Table 2. CMJ results

Ranking MJ	Candidate
1	Mario Draghi
2	Paolo Gentiloni
3	Emma Bonino

Table 3. Top 3 of single-winner Majority Judgement applied to voters

single-winner MJ ranking, comparing the tables tab3 and tab2. The comparison shows different results for the third candidate, highlighting how clustering influences outcome, giving more weight to minorities' judgement.

3.6 Case Study 2: Working hours per week

The last case study is a good paradigm for deciding how to manage working hours in the office, given a fixed number of working hours to be done (18 hours). In this case, we asked 160 students of University Federico II of Naples to choose the best combination of working hours, in presence (P) or with online lectures (O). We used again the grades *Excellent*, *Very Good*, *Good*, *Acceptable*, *Poor*, *To Reject*, *No Opinion* and the five options are:

1. 6 hours (P) - 6 hours (O) - 6 hours (P or O)
2. 10 hours (P) - 4 hours (O) - 4 hours (P or O)
3. 8 hours (P) - 6 hours (O) - 4 hours (P or O)
4. 7 hours (P) - 9 hours (O) - 2 hours (P or O)
5. 5 hours (P) - 5 hours (O) - 8 hours (P or O)

The results of MJ method, with the traditional compute of medians takes back as winner the option 4 (7 hours (P) - 9 hours (O) - 2 hours (P or O)) that has the highest number of "Good" votes.

Instead the compute of winner with CMJ method takes back a different situation: we fixed 2 as number of winners (and number of clusters) and the first one is the option 3 (8 hours (P) - 6 hours (O) - 4 hours (P or O)) and the second one is the option 4, the same winner of MJ method.

As we can see, probably because the number of voters is quite high, the results are not the same like in case study 2. With CMJ, we take into account the wide spectrum of preferences, with special regards for the most polarising ones, which are the most influent in creating different clusters.

Especially for this reason, we may prefer CMJ to MJ for this case-study's look-alike situations, where a shared solution should be taken, considering the different impact it can have on the heterogeneous groups (*clusters*) the judgement is made by.

Conclusions

In section 1, we dealt with logical issues involved in voting rules and judgement aggregation, highlighting majority rule's resistance to strategical vote.

In section 2, a more fined model of majority rule, Majority Judgement, has been presented as an option to better estimate the most shared candidate.

In section 3, the related works have been shown and in section 4, all possible categories of clustering approach has been reported in order to choose the fittest one for our generalization of Majority Judgement as a multi-winner strategy. After that, three different case studies are reported, with a particular attention to the comparison between MJ and CMJ results.

In spite of non-deterministic nature of K-Medoids, Clustered Majority Judgement is thought to be used in high populated disputes. For these reasons, we feel confident about clustering's role of taking into account all different perspectives could be shown in such situation.

Moreover, our implementation is not strictly linked to political field, as is clearly shown in the case studies, mostly because it requires only some fixed parameters: number of winners, number of grades and grades themselves.

An important future challenge could be speeding up the algorithm or making a more flexible structure, even though all the constraints already explained in previous sections need to be satisfied.

References

1. Hae-Sang Park, Chi-Hyuck Jun: A simple and fast algorithm for K-medoids clustering. POSTECH
2. Leonard Kaufman, Peter J. Rousseeuw: Partitioning Around Medoids. In: Finding Groups in Data: An Introduction to Cluster Analysis. John Wiley & Sons (2015)
3. Michel Balinski, Rida Laraki: Majority judgement vs. majority rule. Social Choice and Welfare. HAL open science.(2016)
4. Andrea Loreggia, Nicholas Mattei, Stefano Quintarelli: Artificial Intelligence Research for Fighting. In: Political Polarisation: A Research Agenda, pp. 1–2. Publisher,(2020)
5. Michel Balinski: Fair Majority Voting (or How to Eliminate Gerrymandering). (2006)

6. Klaus Nehring, Marcus Pivato: Incoherent majorities: The McGarvey problem in judgement aggregation. In: Elsevier (2011)
7. Juan Carlos Garcia-Bermejo: A plea for the majority method in aggregating judgements, Oxford Journal (2011)
8. List Christian: Social Choice Theory, The Stanford Encyclopedia of Philosophy (2022)
9. Kenneth O. May: A set of independent necessary and sufficient conditions for simple majority decision (1952)
10. G. Bellec, F. Scherr, A. Subramoney: A solution to the learning dilemma for recurrent networks of spiking neurons (2020)
11. Jon Kleinberg: An impossibility theorem for clustering (2002)
12. Adrien Fabre: Tie-breaking the Highest Median: Alternatives to the Majority Judgment (2018)
13. Pedregosa F., Varoquaux G., Gramfort A., Michel V., Thirion B., Grisel O., Blondel M., Prettenhofer P., Weiss R., Dubourg V., Vanderplas J., Passos A., Cournapeau D., Brucher M., Perrot M., Duchesnay: Scikit-learn Machine Learning in Python (2011)
14. Streibel, Barbara J.: The manager's guide to effective meetings (2003)
15. Robert S. Tannenbaum and Warren H. Schmidt: How to Choose a Leadership Pattern (2009)
16. Blake, Robert Rogers and Mouton, Jane Srygley and Louis B.. Barnes and Larry E.. Greiner: Breakthrough in organization development (1964)
17. Vroom, Victor H and Yetton, Philip W: Leadership and decision-making (1973)
18. Verzuh, Eric and American Psychological Association and others: A Guide to the Project Management Body of Knowledge: PMBOK Guide (2021)
19. Balinski, Michel and Laraki, Rida: Election by majority judgment: experimental evidence (2011)
20. Balinski, Michel and Laraki, Rida: A theory of measuring, electing, and ranking (2007)
21. Brandt, Felix and Conitzer, Vincent and Endriss, Ulle and Lang, Jérôme and Procaccia, Ariel D: Handbook of computational social choice (2016)
22. Serafini, P.: La Matematica in Soccorso Della Democrazia: Cosa Significa Votare e Come Si Può Migliorare il Voto (2019)
23. Arrow, Kenneth J: Social choice and individual values (2012)
24. Costa, Bana E and Carlos, A and Vansnick, Jean-Claude: The MACBETH approach: Basic ideas, software, and an application (1999)
25. Silvia Angilella and Sebastiano Mazzù: The financing of innovative SMEs: A multicriteria credit rating model (2015)
26. Wang, Kun and Shao, Yun and Shu, Lei and Zhu, Chunsheng and Zhang, Yan: Mobile big data fault-tolerant processing for ehealth networks (2016)
27. Chi, Guotai and Zhang, Zhipeng: Multi Criteria Credit Rating Model for Small Enterprise Using a Nonparametric Method (2017)
28. Balinski, Michel and Laraki, Rida: Judge: Don't Vote! (2014)
29. Fazzinga, Bettina and Flesca, Sergio and Furfaro, Filippo and Masciari, Elio: RFID data compression for supporting aggregate queries.
30. Ceci, Michelangelo and Corizzo, Roberto and Fumarola, Fabio and Ianni, Michele and Malerba, Donato and Maria, Gaspare and Masciari, Elio and Oliverio, Marco and Rashkovska, Aleksandra: Big data techniques for supporting accurate predictions of energy production from renewable sources (2015)
31. Luciano Caroprese and Ester Zumpano: Aggregates and priorities in P2P data management systems (2011)

32. Luciano Caroprese and Ester Zumpano: Declarative Semantics for P2P Data Management System (2020)
33. Luciano Caroprese and Ester Zumpano : A Logic Framework for P2P Deductive Databases (2020)