Decision Making with Clustered Majority Judgment

Emanuele D'ajello[1], Davide Formica[2], Elio Masciari[1], Gaia Mattia[1], Arianna Anniciello[1], Cristina Moscariello[1], Stefano Quintarelli[1] and Davide Zaccarella[1]
[1]*University of Napoli Federico II, Napoli, Italy*
[2]*Copernicani, Milano, Italy*

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

Abstract: In order to make a decision process that can best represent the will of a group of people who express themselves about something, like the election of a president or any other situation where people make judgements about more than two possibilities, this paper wants to propose the usage of unsupervised learning techniques, in particular cluster techniques, to extend a single-winner voting system *Majority Judgement* to a multi-winner system which aggregate the preferences of subsets of voters. After an introduction about Majority Judgement, the algorithm used for its clustered version is presented. In the end, a case study will be reported to highlight the differences with the classic Majority Judgment, since sometimes it could be preferable based on the contingencies of the particular election, especially when there is a desire not to neglect minority groups with the same preferences.

1 INTRODUCTION

In this work we first describe general behiavour and advantages of Majority Judgement.

Limits of this model are shown in the case of multiwinner elections as 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 preferences 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. Different voting rules provides different results. Their use depends on the main characteristics we would like to have during a decision process. For example, we could be more interested in avoiding tactical strategy, while accepting some limits about how representative judgements are. We want to explore an example of this trade-off and then describe a 'more inclusive' strategy, using clustering applied to majority judgement. Consider three agents who express their binary judgement ("Yes" or "No") for four statements A, B, $A \wedge B$ and $A \longleftrightarrow B$, comparing outcomes from two different rules. Premised-based rule first 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 considering the used role.

We now focus on Agent 2 case: he's represented in just one of the propositions (A), and his judgement doesn't agree with the outcome, in all the other cases. Here appears clear that Agent 2 could think about manipulating the outcome, pretending a disagreement for A. As consequence, 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.

Table 1: Three agent case of voting.

	A	B	A ∧ B	A ⟷ 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

In such a way, by strategically voting, Agent 2 could manipulate final results. This is the major drawback of using premised voting as rule.

On the other hand, we can highlight a paradoxal aspect if we consider the majority rule: outcomes of the latest two propositions are inconsistent with "Yes" value assigned to both A and B.

This is known as *discursive dilemma* and deals

D'ajello, E., Formica, D., Masciari, E., Mattia, G., Anniciello, A., Moscariello, C., Quintarelli, S. and Zaccarella, D.
Decision Making with Clustered Majority Judgment.
DOI: 10.5220/0011524600003335
In *Proceedings of the 14th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2022) - Volume 3: KMIS*, pages 134-140
ISBN: 978-989-758-614-9; ISSN: 2184-3228

with inconsistency problem in judgement aggregation based on majority rule (G. Bellec, 2020).

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 (Kleinberg, 2002).

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

2 STRATEGIES OF DECISION MAKING

2.1 Collective Decision Process and Majority Judgement

Business meetings are often perceived as useless and unproductive. Moreover, strong difficulties arise when there is an important decision to take: when complexity and effort increase, it's more likely to create 'clusters' representing opposite opinions. More often, final decision becomes leader's task. The biggest difficulty is to decide for the best alternative: in many contexts, more than maximizing the number of people in agreement with the taken decision, it's about making all different groups of people feel included in the decision process. This can be achieved both with leadership, that makes subordinates feel important, and with an inclusive criterion we identify in this paper as the clustered version of Majority Judgement.

Social choice theory studies methods to consolidate the different views of many individuals into a single outcome. The main applications of social choice theory are voting and jury decisions(Brandt et al., 2016).

During voting, electors in a democracy choose one candidate among a list of many candidates, while in a jury decision the individual judges evaluate competitors in a competition, ranking them. Social choice theory's fundamental problem is to find a social decision function that elaborates the preference of judges or voters converging into a jury or electoral decisions while adhering to the main principles of fair voting procedures such as non-dictatorship, universality, independence of irrelevant alternative. Arrow's impossibility theorem shows that the fundamental problem has no acceptable solution in the traditional model (Arrow, 2012). In (Serafini, 2019) Condorcet and Borda methods and limits, Arrow's impossibility theorem and Majority Judgement are illustrated. The results of the general elections have shown that voting systems can run into the Arrow's paradox. A famous example is the 2000 US presidential election. The presence of a minor candidate, Ralph Nader, who had no chance of winning, made Bush the winner in Gore's place. Given the political positions of Nader and Gore, it is very likely that Nader votes they would have gone to Gore if Nader hadn't shown up. It is also likely that Nader supporters preferred Gore to Bush. The American first past the post electoral system, whereby only one vote can be cast and the candidate who gets the most votes wins, has not allowed voters to fully express their preferences.

Majority Judgement (MJ) is a voting technique proposed by two mathematicians in 2007, Michel Balinski and Rida Laraki, aiming to overcome traditional voting methods' paradoxes and inconsistencies. In (Balinski and Laraki, 2007), Balinski and Laraki briefly describe MJ, moving from a social choice theory analysis which highlights traditional voting methods failures. Arrow's impossibility theorem shows that the fundamental problem has no acceptable solution in the traditional model. Practice, otherwise, suggests a different input formulation, traditionally expressed as a preference ranking. Measuring and voting is used during sports competitions such as ice-skating, gymnastics, or wine competitions.

Measuring occurs when a common language is defined, either if this is a quantitative or qualitative language. In this perspective, Arrow's theorem can be interpreted as follows: in absence of a common language a coherent collective decision cannot be made. Hence the need for a voting method where voters evaluate candidates in terms of a common language rather than simply ranking them. MJ makes it possible, since this method asks for electors/judges to express a judgment on all the candidates/competitors, using a known common language. Theorems and experiments confirm that, while there is no method which can completely overcome strategic voting, majority judgment strongly resists manipulation. Balinski and Laraki present MJ as a method both for evaluation and ranking of competitors, candidates or alternatives. In (Balinski and Laraki, 2014) authors underline that the assumption of traditional methods that electors don't really make a personal ranking of candidates, is false and the reason behind the inadequacy of traditional voting models. Forcing electors to rank candidates leads to incoherence, impossibility and incompatibility. Balinski and Laraki (Balinski and Laraki, 2011) present the case of the French presidential elections of 2002 and the results experiment related to the MJ conducted on the occasion of the French presidential elections of 2007, that is a perfect example of Arrow's paradox: the winner depends on the presence or absence of candidates, including those who have absolutely no chance of winning.

2.2 Social Theory's Requirements

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) (May, 1952) (Caroprese and Zumpano, 2020) 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 $< v_1, v_2, ..., v_n >$, where each $v_i \in [-1, 1]$ (to cope with any level of 'pluralism' in its inputs);
- **Anonimity:** applying any kind of permutation on individual preferences does not affect the outcome (to treat all voters equally), i.e.,

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

- **Neutrality:** each alternative has the same weight and for any admissible profile $< v-1, v_2, ..., v_n >$, if the votes for the two alternatives are reversed, the social decision is reversed too (to treat all alternatives equally), i.e.

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

- **Positive Responsiveness:** For any admissible profile $< v_1, v_2, ..., v_n >$, 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 $w_i > v_i$ for some i and $w_j = v_j$ for all other j] and $f(v_1, v_2, ..., v_n) = 0$ or 1, then $f(w_1, w_2, ..., w_n) = 1$.

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 prefixed 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 (Fabre, 2018). Indeed:

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

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

  at least one successful candidate is elected from that partition;
- **Proportionality**: for each subset of voters

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

  number of elected candidate is proportional to the subset'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 l 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.

2.3 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 (Balinski, 2006). 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.
It 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 grades equal in value to the shared median, until one of the tied candidate is found to have the highest median.

It's possible to generalize this system to a multi-winner strategy. The use of a particular cluster in such a contest is crucial, so we first describe the possible options and then we motivate our choice, explaining how K-medoids work.

3 CLUSTERS

3.1 Categories of Clusters

We can state that different types of cluster share the ability to divide data into groups with some common features. We can distinguish:
1. **Connectivity Models:** distance between data points is computed and according to this they show similarity.

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; this kind of model is not flexible as there is no chance to modify cluster once created;
2. **Distribution Models:** in this case, probabilities are computed. They refer to the belonging of a particular distribution once the cluster is created. Applying distribution methods sometimes can be risky as they are prone to overfit data if a precise constraint on complexity is given;
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 da ta 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

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 (Fazzinga et al., 2013).

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 (Ceci et al., 2015).

The used K-medoid algorithm is part of the python `sklearn` library (Pedregosa et al., 2011), which is oriented to machine learning. This library supports *partitioning around medoids* (PAM) (Leonard Kaufman, 2015) proposed by Kaufman and Rousseeuw (1990). The workflow of PAM is described below (Hae-Sang Park, 2008).
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} ($i = 1,...,n$; $j = 1,...,p$). 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 $j = 1, ..., n$ 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 ($\frac{n}{k}$).

3.3 Clustered Majority Judgement

For each cluster majority judgement is applied, and a final ranking of candidates is returned (Andrea Loreggia, 2020). 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.

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 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*;
3. the function *winners* calculates the median for every created cluster;
4. check that winners from cluster are different between each other ; 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 ≥ 0
Ensure: n_winners = (n_1, ..., n_k), k > 1
  k ← number_winners
  max_cluster ← k
  condition ← "ko"
  while condition = "ko" do
      cluster_list ← cluster(vote_list)
      for all list_cluster do
          winners_per_cluster ←
  compute_winners(cluster)
          all_winners ←
  list_of_all_winners(winners_per_cluster)
      end for
      list_winner_distinct =
  list_of_all_distinct_winners(all_winners)
      option_remaining ← number_winners −
  len(list_winner_distinct)
      if option_remaining = 0 then
          condition =' ok'
      else
          k ← option_remaining
          condition ←' ko'
      end if
  end while
```

3.4 Case Study: Using Clustered Majority Judgement to Maximize Agreement

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

In order to test our algorithm, we asked a group of people about their preferences on food. If the presented dish is not considered at least acceptable by people, it would be discarded. So the aim is to maximize the number of not discarded dishes, with two common choices for all the participants. 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 2. 63 voters took part into this experiment and the algorithm form two clusters, exactly like the number of winners.

Table 2: CMJ results.

Cluster	Cluster size	Winner
Cluster 1	33	Bovine meat
Cluster 2	30	Tuna

Table 3: Top 2 of single-winner Majority Judgement applied to voters.

Ranking MJ	Candidate
1	Bovine meat
2	Chicken

We can compare CMJ results with single-winner MJ ranking, comparing Table 3 3 and Table 2 2. Expressed judgements are very polarizing and the two formed cluster seems in opposition between each other, since the most preferred dishes for one are the most negatively judged by the other. For this reason, we notice both for Majority Judgement and Clustered Majority Judgement the tendency to avoid the favourite dishes, focusing on the moderate ones.

In case of Majority Judgement, the solution is *Bovine meat* and *Chicken*, where both alternatives are considered not acceptable for cluster 2 and for this reason, 29 dishes would be discarded. With clustered approach, the solution takes into account cluster 2's preferences, and provides a lower number of discarded dishes.

4 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, a case study is reported, with a particular attention to the comparison between MJ and CMJ results.

The CMJ, as shown, represents the optimal compromise in case of polarized groups (clusters), so, in this situations, this method could be the preferable choice in order to take into account minorities judgements.

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 (except the first one), 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

Andrea Loreggia, Nicholas Mattei, S. Q. (2020). Artificial intelligence research for fighting. In *Political Polarisation: A Research Agenda*.

Arrow, K. J. (2012). Social choice and individual values. Yale university press.

Balinski, M. (2006). Fair majority voting (or how to eliminate gerrymandering). In *The American Mathematical Monthly, Vol. 115, No. 2*. Mathematical Association of America.

Balinski, M. and Laraki, R. (2007). A theory of measuring, electing, and ranking. National Acad Sciences.

Balinski, M. and Laraki, R. (2011). Election by majority judgment: experimental evidence. In *In Situ and Laboratory Experiments on Electoral Law Reform*. Springer.

Balinski, M. and Laraki, R. (2014). Judge: Don't vote! volume 62, pages 483–511. INFORMS.

Brandt, F., Conitzer, V., Endriss, U., Lang, J., and Procaccia, A. D. (2016). Handbook of computational social choice. Cambridge University Press.

Caroprese, L. and Zumpano, E. (2020). Declarative semantics for P2P data management system. *J. Data Semant.*, 9(4):101–122.

Ceci, M., Corizzo, R., Fumarola, F., Ianni, M., Malerba, D., Maria, G., Masciari, E., Oliverio, M., and Rashkovska, A. (2015). Big data techniques for supporting accurate predictions of energy production from renewable sources. In Desai, B. C. and Toyama, M., editors, *Proceedings of the 19th International Database Engineering & Applications Symposium, Yokohama, Japan, July 13-15, 2015*, pages 62–71. ACM.

Fabre, A. (2018). Tie-breaking the highest median: Alternatives to the majority judgment. In *Paris School of Economics*.

Fazzinga, B., Flesca, S., Furfaro, F., and Masciari, E. (2013). Rfid-data compression for supporting aggregate queries. *ACM Trans. Database Syst.*, 38(2):11.

G. Bellec, F. Scherr, A. S. (2020). A solution to the learning dilemma for recurrent networks of spiking neurons. In *Nat Commun*.

Hae-Sang Park, C.-H. J. (2008). A simple and fast algorithm for k-medoids clustering. In *POSTECH*. Elsevier.

Kleinberg, J. (2002). An impossibility theorem for clustering. In *Cornell University*.

Leonard Kaufman, P. J. R. (2015). Partitioning around medoids. In *Finding Groups in Data: An Introduction to Cluster Analysis*. John Wiley & Sons.

May, K. O. (1952). A set of indipendent necessary and sufficient conditions for simple majority decision. In *Carleton College*.

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., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. In *Journal of Machine Learning Research*.

Serafini, P. (2019). La matematica in soccorso della democrazia: Cosa significa votare e come si può migliorare il voto. Independently Published.