Constrained minimization of Manhattan distance for voting on budgets

Jan Behrens

2024-11-06 10:10 UTC

1 Problem

We have a certain budget or resource, e. g. 1000 person hours or $ 1,000,000 to be assigned to d candidates (e.g. projects). We assume n voters, which each propose a relative budget allocation for each candidate, e.g. $(0.6, 0.3, 0.1)$ if he/she/they wants to assign 60 % of the total budget to the first, 30 % to the second, and 10 % to the third candidate. In the following, $X \in \mathbb{R}^{n \times d}$ is a $n \times d$ matrix containing all candidates as columns and all voters as rows. X_{ij} is voter i's desired relative budget for candidate j.

We assert certain constraints on the voters' ballots. Equations (1) below ensure that no voter can spend less than 0 % or more than 100 % on a particular candidate on their respective ballot, and equations (2) ensure that each voter must use 100 % of the total budget[1].

$$\forall i \, \forall j : 0 \leq X_{ij} \leq 1 \tag{1}$$

$$\forall i : \sum_{j=1}^{d} X_{ij} = 1 \tag{2}$$

We now search for a suitable voting algorithm $f : X \mapsto w$, which maps all votes X to a result w that assigns each candidate j a certain budget $w_j \in [0, 1]$. For the output w of the voting algorithm, we demand $\sum w_j = 1$ such that the total available budget is spent, but not more.

$$\forall j : 0 \leq w_j \leq 1 \tag{3}$$

$$\sum_{j=1}^{d} w_j = 1 \tag{4}$$

$$f : \left\{ X \in [0,1]^{n \times d} \mid \forall i : \sum_{j=1}^{d} X_{ij} = 1 \right\} \to \left\{ x \in [0,1]^d \mid \sum_{j=1}^{d} x_j = 1 \right\} \tag{5}$$

$$f : X \mapsto w \tag{6}$$

Moreover, the algorithm f shall be designed in such a way that each voter has little incentive for tactical voting.

[1] Note that in cases where there should be an option to not spend all resources, it's always possible to add another candidate labeled "save the money/resource", if desired.

2 Definitions and notation

In the following, X_i (without $_j$) shall denote the i-th row of matrix X, i. e. voter i's ballot, as a (column) vector; whereas X_{ij} or $X_{i,j}$ is the element in the i-th row and j-th column, i. e. voter i's assigned relative budget for candidate j.

Let further $\|x\|_1$ denote the Manhattan norm and $\|x\|_2$ the Euclidean norm of vector x, i.e.:

$$\|x\|_1 = \sum_j |x_j| \tag{7}$$

$$\|x\|_2 = \sqrt{\sum_j x_j^2} \tag{8}$$

3 1-dimensional case

If $d = 2$, there is a well-known solution to the problem because due to equation (2), the budgets on each ballot will lay on a line:

$$X_i \in \left\{ \begin{pmatrix} x_1 \\ 1 - x_1 \end{pmatrix} \in \mathbb{R}^2 \mid 0 \leq x_1 \leq 1 \right\} \tag{9}$$

Here, the 1-dimensional median can be used as a solution to the problem and there will be no incentive[2] for tactical voting:

$$w_1 = \text{median}\, X_{i,1} \tag{10}$$

$$w = \begin{pmatrix} w_1 \\ 1 - w_1 \end{pmatrix} \tag{11}$$

Unfortunately this method cannot be used for $d > 2$ because there exists no useful ordering for $\mathbb{R}^{d-1}$ if $d > 2$, which is a requirement for the median.

4 Proposed algorithm for higher dimensions

We first calculate the geometric median g:

$$g = \underset{x \in \mathbb{R}^d}{\arg\min} \sum_{i=1}^{n} \sqrt{\sum_{j=1}^{d} (x_j - X_{ij})^2} \tag{12}$$

$$= \underset{x \in \mathbb{R}^d}{\arg\min} \sum_{i=1}^{n} \|x - X_i\|_2 \tag{13}$$

Note that g may be undefined in the rare case of n being even *and* all votes being on a 1-dimensional line. In this case, we use the 1-dimensional median and take the arithmetic mean of the two middle values.[3]

[2] This holds for an odd number of voters. In case of an even number of voters, this depends on the tie-breaker.

[3] This value is then also the final result w of the algorithm.

Now the assigned budgets w_j are:

$$w = \underset{x \in \mathbb{R}^d, \sum_j x_j = 1}{\arg\min} \left[\frac{\sum_{j=1}^{d} |x_j - g_j|}{2} + \sum_{i=1}^{n} \sum_{j=1}^{d} |x_j - X_{ij}| \right] \tag{14}$$

$$= \underset{x \in \mathbb{R}^d, \sum_j x_j = 1}{\arg\min} \left[\frac{\|x - g\|_1}{2} + \sum_{i=1}^{n} \|x - X_i\|_1 \right] \tag{15}$$

Here, $\sum x_j = 1$ ensures that, according to (4), not more and not less than the total available budget is used and $\frac{\|x-g\|_1}{2}$ serves as a tie-breaker, which corresponds to adding half a vote for the geometric median.

5 Reasoning

Disregarding the tie-breaking and the constraint that the budget must be spent and not be exceeded, the definition of w corresponds to minimizing the sum of Manhattan distances between w and the votes X_i. This can be seen as an equillibrium process where each voter tries to increase the budgets for those candidates where they desire a higher budget and to reduce the budgets for those candidates where they desire a lower budget, and where each voter can transfer the same amount at the same time but not increase or reduce the total budget spent (i.e. each increase must have an accompanying reduction). Because it doesn't matter how much higher or lower a voter's wish for a particular candidate's budget is in this process, there is little incentive for tactical voting; i. e. there is little incentive to express polarized opinions where a voter, for example, assigns all resources to one candidate in the attempt to increase that candidate's budget further.

However, during this process, all candidates are treated equally. This means that for each voter, the algorithm doesn't distinguish from which candidate the budget is removed or to which candidate the budget is added. Assuming that some voters might have a strong opinion on the budget for one candidate but might not care for the budget on other candidates, this unfortunately still leaves room for strategic optimization of one's ballot. In particular, it may be advisable for a voter i to orient his/her/their budget assignments for those candidates they don't care much about toward the other voters in order to avoid spending their voting power on those candidates.

One solution to this problem could be to seek algorithms that allow each voter to express certain preferences, e. g *"I want candidate 1's budget to be $2,000, but I don't care much about candidate 2 and 3."* However, such an approach isn't complete as the preferences might be even more specific, such as *"I want candidate 1's budget to be $2,000, and candidate 2 and 3 should get a budget of $1,000 in total, but I don't care how much of those $1,000 is assigned to candidate 2 and how much to candidate 3."* Complexity for the voters when filling out their ballots might increase drastically with such an approach.

6 Comparison with component-wise median and geometric median

6.1 Component-wise median

The component-wise median may yield results where more than 100 % or less than 100 % of the total budget is spent, i. e. violation of equation (4). Attempting linear scaling of the result would violate the optimization performed in equation (14)/(15), however.

Another option would be to calculate the component-wise median for $d-1$ candidates and assign one candidate the remaining budget. Apart from the problem that this remaining budget might be negative, resulting in violation of (3), the choice of that candidate has an impact on the voting result as can easily be seen in example E_4 from the next section.

6.2 Geometric median

Using the geometric median not only as a tie-breaker but as a final result would correspond to voters being willing to reduce a candidate's budget even if they denoted a higher budget for that candidate on their ballot as long as the quadratic error is improved. This seems unreasonable unless the dimensions are somehow related to one another (e. g when deciding on a location on a two-dimensional plane), which is generally not true in case of budget allocation.

7 Examples

The following examples E_k have been calculated with the computer program given in the next section.

$$E_1 = \begin{pmatrix} 100\,\% & 0\,\% \\ 58\,\% & 42\,\% \\ 0\,\% & 100\,\% \end{pmatrix} \tag{16}$$

$$f(E_1) = \begin{pmatrix} 58\,\% & 42\,\% \end{pmatrix}^{\mathsf{T}} = g \tag{17}$$

$$E_2 = \begin{pmatrix} 60\,\% & 30\,\% & 10\,\% \\ 20\,\% & 60\,\% & 20\,\% \\ 25\,\% & 25\,\% & 50\,\% \end{pmatrix} \tag{18}$$

$$f(E_2) \approx \begin{pmatrix} 33.11\,\% & 39.60\,\% & 27.28\,\% \end{pmatrix}^{\mathsf{T}} = g \tag{19}$$

$$E_3 = \begin{pmatrix} 60\,\% & 30\,\% & 10\,\% \\ 20\,\% & 60\,\% & 20\,\% \\ 0\,\% & 0\,\% & 100\,\% \end{pmatrix} \tag{20}$$

$$f(E_3) \approx \begin{pmatrix} 30.28\,\% & 42.39\,\% & 27.34\,\% \end{pmatrix}^{\mathsf{T}} = g \tag{21}$$

$$E_4 = \begin{pmatrix} 60\,\% & 30\,\% & 10\,\% \\ 60\,\% & 30\,\% & 10\,\% \\ 60\,\% & 30\,\% & 10\,\% \\ 60\,\% & 30\,\% & 10\,\% \\ 60\,\% & 30\,\% & 10\,\% \\ 20\,\% & 60\,\% & 20\,\% \\ 20\,\% & 60\,\% & 20\,\% \\ 20\,\% & 60\,\% & 20\,\% \\ 0\,\% & 0\,\% & 100\,\% \\ 0\,\% & 0\,\% & 100\,\% \\ 0\,\% & 0\,\% & 100\,\% \\ 0\,\% & 0\,\% & 100\,\% \end{pmatrix} \tag{22}$$

$$f(E_3) = \begin{pmatrix} 50\,\% & 30\,\% & 20\,\% \end{pmatrix}^{\mathsf{T}} \neq g \tag{23}$$

In examples E_1 through E_3, tie-breaking is used and the results are also equivalent to the geometric median g. Example E_4 shows a case where tie-breaking is not needed and the geometric median has no influence on the result.

8 Implementation in Octave

```
function w = budget_voting(X)

  # usage: budget_voting(X)
  #
  # X is a matrix with voters as rows and candidates as columns; each entry
  # reflects a relative budget assigned by the voter to the candidate.
  #
  # The function returns a column vector of relative budget assignments for
  # each candidate such that the Manhattan distance is minimized and the
  # total budget is spent. The geometric median and, as a fallback, the
  # 1-dimensional median are used as tie-breaker.

  # Xt is X transposed, i.e. candidates as rows and voters as columns.
  Xt = transpose(X);

  # Xtn is Xn normalized such that each voter assigns a total budget of 1.
  Xtn = Xt ./ sum(Xt, 1);

  # Handle the 1-dimensional case:
  if rank(Xtn - mean(Xtn, 2)) <= 1
    w = median(Xtn, 2);
    return
  endif

  # eudists(x) contains the Euclidean distances between x and each vote.
  eudists = @(x) sqrt(sum((x-Xtn).^2, 1));

  # eudist(x) is the sum of all Euclidean distances between x and each vote.
  eudist = @(x) sum(eudists(x), 2);

  # eudist_grad(x) is the gradient of eudist(x) as a column vector,
  # used to aid optimization.
  eudist_grad = @(x) sum((x-Xtn) ./ eudists(x), 2);

  # The geometric median minimizes the sum of all Euclidean distances:
  g = sqp(mean(Xtn, 2), {eudist, eudist_grad});

  # mandist(x) contains the Manhattan distances between x and each vote
  # plus a tie-breaker (weighted 1/2) in favor of the geometric median.
  mandist = @(x) sum(sum(abs(x-Xtn), 1), 2) + sum(abs(x-g), 1)/2;

  # mandist_grad(x) is the gradient of mandist(x) as a column vector,
  # used to aid optimization.
  mandist_grad = @(x) sum(sign(x-Xtn), 2) + sign(x-g)/2;

  # excess(x) is the sum of the candidates' budgets minus 1.
  excess = @(x) sum(x, 1) - 1;

  # excess_grad_t(x) is the gradient of excess(x) as a row vector,
  # used to aid optimization.
  excess_grad_t = @(x) ones(1, rows(x));

  # The solution minimizes the Manhattan distance (including the tie-breaker)
  # under the condition that the sum of the candidates' budgets is 1.
  w = sqp(g, {mandist, mandist_grad}, {excess, excess_grad_t});
end
```