

Aggregating Correlated Estimations with (Almost) no Training

Théo Delemazure^{a,*}, François Durand^b and Fabien Mathieu^c

^aLAMSADE, Paris Dauphine University, PSL, CNRS

^bNokia Bell Labs France

^cSwapcard

Abstract. Many choice problems cannot be solved exactly and use several estimation algorithms that assign scores to the different available options. The estimation errors can have various correlations, from low (e.g. between two very different approaches) to high (e.g. when using a given algorithm with different hyperparameters). Most aggregation rules would suffer from this diversity of correlations. In this article, we introduce *Embedded Voting (EV)*, an aggregation rule designed to take correlations into account, and we compare it to other aggregation rules in various experiments based on synthetic data. Our results show that when sufficient information about the correlations between errors is available, a maximum likelihood aggregation should be preferred. Otherwise, typically with limited training data, EV outperforms the other approaches.

1 Introduction

We consider the problem of score aggregation through the general choice problem of having to choose one item from a set of options called *candidates*. The actual utility of each candidate is unknown and impossible to compute, but we have access to human or software agents that can provide a noisy estimate of it. How to select the best possible candidate?

In the aggregated output, the errors of the individual agents should ideally mitigate one another and produce a collective decision that would outperform the one of a single agent, like in the seminal Condorcet Jury Theorem [9]. However, two issues can hamper aggregation. The first one is the varying accuracy of the agents. Although of great importance, this issue is outside the scope of this paper, where we focus on the second one: correlations between agents.

This problem can arise from various reasons. In the case of software agents, it is frequent to use several variants of an algorithm, or the same algorithm with different hyperparameters. In the case of human experts, some of them may share a common sociocultural background or prior deliberation on the topic at hand, which may give them a common bias.

These correlations, which may be unknown to the aggregation rule, may hinder its performance. Consider the following toy example: Three algorithms are used, two of them being exact clones (they always return the same estimate). Unaware of the clones, the aggregation rule uses the median score. As the median estimate is always the one of a clone, the aggregation behaves exactly as any of the clones, and the potential benefit from the independence of the third algorithm is lost.

1.1 Related Work

Our model is close to the one of *epistemic social choice*, in which voters have noisy estimates of the candidates' true values. Some works investigate the quality of the results of specific voting rules in this model [24, 8, 10].

Decisions based on the aggregation of scores appear in many contexts, not necessarily related to the human activity of *social choice*. In regression analysis, the quality of prediction can be improved by aggregating the results of different algorithms. For example, the methods of *boosting* and *bagging* achieve some form of aggregation [28].

When several estimations are available for the same problem, it is common for some of them to be correlated, for example when the same technique is used with different hyperparameters or different training datasets. Since these correlations can be harmful for aggregation, many of the main methods require the panel of algorithms to be *diversified* [14]. In particular, this avoids biasing the results in the direction of a potential “majority group”. Here, we do not assume this diversity constraint to be satisfied.

Our topic is also close to the Unsupervised Ensemble Learning problem, where the goal is to find the best way to aggregate the outputs of a family of classifiers without any feedback. Some methods in this area address the problem of correlated classifiers [6, 16, 18]. These methods are also useful in the crowd-sourcing framework in which agents might be correlated. However, in their setting, each choice problem always examines the same candidates (e.g. “cat” and “dog”) in different situations (e.g. images to classify), whereas in our setting, there is no obvious mapping between the candidates of any two distinct choice problems.

Finally, there is a vast literature in statistics applied to physics on the problem of aggregating correlated measures of the same quantity to reduce the noise [13, 15, 25, 26]. In particular, the method called BLUE is equivalent to the model-aware benchmark that we present in Section 3.2 [21, 22]. But our general problem differs as we do not assume any prior knowledge on the distributions and the correlations of the estimators. Moreover, we focus on the choice problem of selecting the best candidate, which is not the same as inferring one value from noisy estimates. We also differentiate ourselves by proposing a novel method based on spectral decomposition for this problem.

It is worth noting that this paper introduces a new choice rule where the identification of correlations between agents is based on the classic data preprocessing technique known as *Singular Value Decomposition* (SVD) [30]. The use of SVD is not uncommon in re-

* Corresponding Author. Email: theo.delemazure@dauphine.eu

lated fields such as Multi-armed Bandits [1] (to compute upper confidence bounds) and recommendation systems [3] (for collaborative filtering). However, to the best of our knowledge, this is the first time that SVD is applied to a choice problem based on noisy estimations by correlated agents.

1.2 Contribution

In this paper, we propose a new aggregation technique, Embedded Voting (EV), inspired by the notion of *Nash product* (cf. Section 2.3). EV uses spectral analysis to mitigate the impact of correlations. We validate the interest of this technique in a controlled model where agents only differ by the correlations between them, but not by their accuracy. We compare it to other approaches, using as upper bound a maximum likelihood technique based on a full knowledge of the model parameters.

Our main findings are that with a prior training, EV is on par with the upper bound, except when all noise parameters are high (cf. Section 4.3); without training, EV is the best of all the untrained aggregation rules of our benchmark throughout all our experiments, with a performance that is generally very close to its trained version, meaning that it can generally infer the relevant information about the correlations just by considering the options of the present choice problem.

The rest of the paper is organized as follows. Section 2 presents our proposal for score aggregation. Section 3 gives a first experimental validation. Section 4 explores how various parameters impact the performance of the proposed solutions. Section 5 concludes.

2 Score Aggregation

Our notations are inspired by the field of *computational social choice* [5]. We consider a choice problem with a given set $\mathcal{C} = \{c_1, \dots, c_m\}$ of m candidates. Each candidate $c_j \in \mathcal{C}$ has a utility $u(c_j) \in \mathbb{R}$. We want to select a candidate with the highest possible utility.

However, the utility values cannot be directly observed. Instead, we have access to a set $\mathcal{A} = \{a_1, \dots, a_n\}$ of n agents that provide utility estimates. Formally, each agent a_i assigns a *score* to each candidate c_j , denoted by $s_i(c_j) \in \mathbb{R}$. We also introduce $\varepsilon_i(c_j) := s_i(c_j) - u(c_j)$, the (unknown) noise of agent a_i for candidate c_j . We view $\varepsilon_i(c_j)$ as a random variable to enable the use of a probabilistic framework. All the scores can be represented in an $n \times m$ matrix $S = (s_i(c_j))_{1 \leq i \leq n, 1 \leq j \leq m}$. An *aggregation rule* is a function f that associates to S a candidate $f(S) \in \mathcal{C}$. Our goal is to maximize $\mathbb{E}(u(f(S)))$. The rest of this section presents different approaches to solve this problem.

2.1 Welfare-Based Approaches

In many contexts, the *utilitarian welfare* is a good metric to select a winner from a list of candidates. In our case, viewing the scores as subjective utility values, the corresponding aggregation rule is as follows: compute the welfare of each candidate as the sum of its scores, $w_{\text{util}}(c_j) := \sum_{i=1}^n s_i(c_j)$; then choose a candidate with maximal welfare. This rule is equivalent to selecting the candidate with the highest average score. It is usually called *Range Voting (RV)* in the literature of computational social choice [27].

If the noise functions ε_i are drawn from independent and identical Gaussian distributions, then RV is the optimal aggregation rule (see Theorem 4.1 of [19] or Section 3.2). However, we precisely focus on cases where they are not independent. For example, consider $n = 24$

agents such that 20 of them are strongly correlated and each of the remaining 4 is independent of all the others. In most cases, the utilitarian welfare will mostly depend on the scores given by the large group of 20 agents, and the information given by the 4 independent agents will mostly be ignored. More generally, the lack of diversity can significantly decrease the quality of the results given by the utilitarian welfare maximization [14, 23].

A variant of RV consists in processing the raw scores before summing them in order to mitigate the impact of large errors. For example, we can apply a Heaviside step function shifted by a given threshold. In that case, the welfare of a candidate is simply the number of agents that rate the candidate above the threshold, which is a form of *Approval Voting (AV)* [5, p. 53]. Note that AV increases the odds of having multiple candidates that maximize the welfare, especially if the number n of agents is small. Therefore, the choice of the tie-breaking rule may be important.

Another notion of welfare is the *Nash Product (NP)*, widely used in game theory [17, 20]. It has appealing fairness properties in numerous contexts, like collective decision making or project fundings [2, 4, 7, 12]. Assuming that all scores are positive, the Nash product of a candidate is defined as the product of its scores: $w_{\text{Nash}}(c_j) := \prod_{i=1}^n s_i(c_j)$. It also suffers in the case of varying correlations: for example, if one independent agent mistakenly doubles the score of a candidate, then the welfare of the candidate is doubled, but if a group of k correlated agents do the same, then the welfare of the candidate is multiplied by 2^k .

2.2 Maximum-Likelihood Approaches

Viewing the noises as random variables, it is natural to use a maximum likelihood estimator [11]. With perfect information about the joint distribution of the noises, it can give the best possible estimates for the utility values of the candidates (Section 3.2 gives an example of exact computation). However, it is impossible in general to determine the characteristics of these distributions from the observed data alone. For example, consider the following case with $n = m = 2$: $s_1(c_1) = s_2(c_1) = 10$ and $s_1(c_2) = s_2(c_2) = 15$, i.e. both agents estimate that the first candidate has utility 10 and the second has utility 15. It is possible that both agents are error-free and that 10 and 15 are the two utility values. But it is also possible that both agents are noisy and fully correlated, that both candidate have the same utility, and that the difference between the values 10 and 15 just comes from the shared error of the agents. The two cases are impossible to distinguish from observation alone.

2.3 Our Proposal: Embedded Voting

We now introduce our aggregation rule, whose goal is to exploit correlations between the agents without requiring prior knowledge on the ground truth or the noise model.

To start with, consider the following ideal case: the set of agents \mathcal{A} is partitioned into k perfectly identified groups A_1, \dots, A_k . The agents of the same group are fully correlated (they always yield the same estimates) while agents from different groups are independent. All the agents have the same accuracy. We focus on the extraction of some welfare value $w(c_j)$ associated to a given fixed candidate c_j , based on the estimates $s_i(c_j)$ as in Section 2.1.

Contrary to most classical voting rules, which are *anonymous*, in the sense that they treat voters symmetrically [5], here we want to use the additional information that we have about the agents in order to weigh them. In particular a desired property would be that the impact of a given group of identical agents does not depend on its

size. A natural way to achieve this is to assign the same weight to each group. If using the utilitarian welfare, this leads to the average of the average scores of each group, which results in the following welfare for candidate c_j :

$$w(c_j) := \frac{1}{k} \sum_{l=1}^k \frac{1}{|A_l|} \sum_{a_i \in A_l} s_i(c_j). \quad (1)$$

If the scores are positive, one can also consider the Nash product of the utilitarian welfare of each group, defined as:

$$w(c_j) := \prod_{l=1}^k \sum_{a_i \in A_l} s_i(c_j). \quad (2)$$

Equation 2 is affected by the size of the groups, like Equation 1, but with the same scaling factor for each candidate: for example, if we double the size of one group, it simply doubles the welfare of each candidate, which does not change the outcome of the choice problem.

However, in practice, groups cannot be precisely delimited: the correlations within the groups may be hard to identify, or some agents may share correlations with several groups. Yet, it is possible to use spectral analysis to infer the underlying groups and generalize Equation 2 to arbitrary cases.

But keep on considering the ideal case for the moment. We can represent the link between agents and groups by vectors: to each agent $a_i \in A_l$, associate the $1 \times k$ vector $\vec{e}_i := (\mathbf{1}_{l'=l})_{1 \leq l' \leq k}$ (1 at the l^{th} position, 0 everywhere else). We call \vec{e}_i the *embedding* of agent a_i .

Using the embeddings, define the $n \times k$ score matrix M_j of candidate c_j as follows:

$$M_j := \begin{pmatrix} \sqrt{s_1(c_j)} \vec{e}_1 \\ \vdots \\ \sqrt{s_n(c_j)} \vec{e}_n \end{pmatrix}. \quad (3)$$

Denote by $(\lambda_1, \dots, \lambda_k)$ the singular values of the matrix M_j , obtained by SVD [30]. Up to re-indexing, we have the relation $\lambda_l^2 = \sum_{i: a_i \in A_l} s_i(c_j)$ (the square root in (3) provides the benefit of this simple formula). Thus, each λ_l^2 represents the utilitarian welfare of the group A_l . In particular, the product $\prod_{l=1}^k \lambda_l^2$ is exactly the welfare value defined by Equation 2.

Using spectral analysis may seem excessive in the ideal case where the partition is perfect and known. However, its main advantage is that it is based on some embedding of the agents that can convey more information than a simple partition. In particular, we can extend Equation 2 to a more general case if we can infer an embedding of the agents.

To this end, we simply define the estimated embedding vectors $\hat{\vec{e}}_i$ of the agents as their score vectors (including the scores on a potential training set of candidates), normalized such that all the score vectors of the agents have the same mean and standard deviation. More specifically, we have the j^{th} component of agent i set as:

$$\hat{e}_{i,j} := \frac{s_i(c_j) - \text{avg}_{j'} s_i(c_{j'})}{\text{std}_{j'} s_i(c_{j'})}. \quad (4)$$

Then for each candidate c_j , we define \hat{M}_j similarly to Equation (3). Note that its size is $n \times m$, instead of $n \times k$ for M_j in the ideal case.

For a given candidate c_j , denote $(\lambda_1, \dots, \lambda_{\min(m,n)})$ the singular values of \hat{M}_j sorted in descending order. \hat{M}_j can contain redundant dimensions (in particular if the agents are correlated), so we would

like to compute the welfare $w(c_j)$ as the product $\prod_{l=1}^{\hat{k}} \lambda_l^2$ of the \hat{k} greatest singular values, where \hat{k} would estimate the number of relevant dimensions in the embedding of the agents.

To estimate \hat{k} , we define $\hat{S} = (\hat{e}_{i,j})_{i,j}$, the normalized full score matrix, and we set \hat{k} to the number of singular values above 0.95 times the average of all $\min(m, n)$ singular values of \hat{S} .¹

We call the resulting method *Embedded Voting (EV)*.

As an illustration, let us apply this method to the ideal case, in the limit of a large dataset (i.e. large m). By definition in Equation 4, all vectors \vec{e}_i have norm 1. When two agents are fully correlated, they have the same embedding vector. And when two agents i and i' are independent, then by definition of independence, their vectors \vec{e}_i and $\vec{e}_{i'}$ are orthogonal in the limit of a large dataset. So the estimated embedding $(\hat{\vec{e}}_i)_i$ will be the same as $(\vec{e}_i)_i$, up to an orthonormal change of basis. It is then easy to prove that the singular values of \hat{S} tend to $(|A_1|, \dots, |A_k|, 0, \dots, 0)$ in the limit of a large dataset. The average singular value is $(\sum_l |A_l|)/n = 1$, so the number \hat{k} of singular values higher than 0.95 the average is precisely k . Now consider a candidate j . Once again, \hat{M}_j is the same as M_j , up to an orthonormal change of basis and addition of irrelevant dimensions due to the statistical noise. So taking the $\hat{k} = k$ greatest singular values of \hat{M}_j will give the same result as relying on M_j .

3 Validation

We now evaluate the performance of the aggregation rules that we have described. We use artificial datasets to have a full control over the behavior of the candidates and the agents. Broadly speaking, one performance measure goes as follows:

- Draw for m candidates their utilities $u(c_j)$, $1 \leq j \leq m$.
- Use a noise model to draw the agent estimations $(s_i(c_j))_{1 \leq i \leq n, 1 \leq j \leq m}$.
- Feed different aggregation rules with the estimates (the rules never access the utility values).
- Measure with the utility values the quality of the returned candidate.
- Repeat the steps above enough times to have a good measurement of the expected performance.

In details, we draw the utility values of the candidates independently according to a normal distribution, which is chosen to be $\mathcal{N}(0, 1)$ without loss of generality. By default, for each choice problem, $m = 20$ candidates are considered.

The performance metric we use to compare the different aggregation rules is the *average relative utility*: for each choice, if a rule yields candidate c_j , then its relative utility is a value between 0 and 1 defined by $\frac{u(c_j) - u_{\min}}{u_{\max} - u_{\min}}$, where u_{\max} (resp. u_{\min}) is the best (resp. worst) utility among those of the candidates. For each rule, we compute the average relative utility over 10,000 choices to assert its performance.² The open source code developed for this paper is avail-

¹ Ideally, we would like to only keep the singular values that are above the average of the singular values. But if all the voters are independent, we would like to keep all the singular values, i.e. $\hat{k} = n$. However, in that case, all the singular values will be very similar but not perfectly equal due to noise and numerical errors, so some of them will be below the average, and thus we would have $\hat{k} < n$. The 0.95 discount enables these singular values that are slightly below the average to still be taken into account. Thus, we will get the correct \hat{k} in the symmetric case without impacting much the non-symmetric cases.

² In our preliminary experiments, we also compared the different rules with respect to the accuracy metric, i.e. the probability to choose the best candidate, and we observed similar trends than for utility and relative utility.

able on GitHub³.

The rest of this section is organized as follows. Section 3.1 details our noise model. Section 3.2 provides the exact maximum-likelihood solution, which requires the knowledge of the model and its parameters, along with an approximation, which does not. Section 3.3 recaps the aggregation rules that we consider and gives some implementation details. Section 3.4 presents a first representative experiment.

3.1 Noise Model

We want our noise model to have the following properties:

- All agents have the same accuracy (we focus here on correlations). Considering agents with different accuracies and weighting them accordingly is another complex problem, outside the scope of this paper.
- The correlations between agents can be tuned finely.

To achieve these objectives, we propose the following model, which combines correlated and uncorrelated errors. It relies on three parameters:

- An $n \times k$ matrix $E = (e_{i,l})_{1 \leq i \leq n, 1 \leq l \leq k}$ for some $k \in \mathbb{N}$ that represents the agents in a *feature* space.
- A *feature* noise intensity $\sigma_f \in \mathbb{R}_{\geq 0}$.
- A *distinct* noise intensity $\sigma_d \in \mathbb{R}_{\geq 0}$.

The matrix E generalizes the embedding introduced in Section 2.3. For convenience, we assume that each row of E has at least one non-zero component and that its Euclidean norm is 1: $\forall i \in \{1, \dots, n\}, \sum_{1 \leq l \leq k} e_{i,l}^2 = 1$. The parameter σ_f determines the part of the noise related to the underlying features of the agents, while σ_d determines some noise that is independent across all agents.

For a given candidate c_j , we draw the noises as follows: we first draw $n + k$ i.i.d. values $d_{1,j}, \dots, d_{n,j}, f_{1,j}, \dots, f_{k,j}$ following the standard normal distribution $\mathcal{N}(0, 1)$. The first n values determine noises that are specific to each agent, while the last k ones are specific to each feature. The estimation noise of agent a_i for c_j is then defined as:

$$\varepsilon_i(c_j) := \sigma_d d_{i,j} + \sigma_f \sum_{1 \leq l \leq k} e_{i,l} f_{l,j}. \quad (5)$$

By construction, the marginal distribution of each $\varepsilon_i(c_j)$ is the sum of two independent distributions $\mathcal{N}(0, \sigma_d)$ and $\mathcal{N}(0, \sigma_f)$, and two given agents have fully independent noises if they have no feature in common.

Equivalently, let $S(c_j)$ be the $n \times 1$ random vector of the scores provided by the agents, $\bar{E} := (\sigma_d I_n \quad \sigma_f E)$ be the $n \times (n + k)$ matrix that concatenates the characteristics of distinct noise and feature noise (where I_n denote the $n \times n$ identity matrix), and $X_j = (d_{1,j}, \dots, d_{n,j}, f_{1,j}, \dots, f_{k,j})^T$ be a random vector whose $n + k$ values follow i.i.d. standard normal distributions. Under our noise model, $S(c_j)$ can be written as:

$$S(c_j) = u(c_j) \mathbf{1}_{n \times 1} + \bar{E} X_j. \quad (6)$$

Thus, conditionally on the value of $u(c_j)$, $S(c_j)$ follows a multivariate normal distribution of mean $u(c_j) \mathbf{1}_{n \times 1}$ and covariance matrix $\Sigma := \bar{E} \bar{E}^T$.

Let us illustrate the noise model on a few toy examples: in the degenerate case where E is a $n \times 0$ matrix, i.e. $\bar{E} = I_n$, we have n

independent agents; the case $E = \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix}$ with $\sigma_d = 0$ models the example with two clones given in the introduction (more generally, the ideal case with k perfect groups introduced in Section 2.3 can be modeled with a $n \times k$ indicator matrix); the case $E = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1/\sqrt{2} & 1/\sqrt{2} \end{pmatrix}$ models two independent agents and a third one that is correlated with the other two.

3.2 Maximum Likelihood

With the knowledge of the noise model, i.e. (E, σ_d, σ_f) , it is possible to infer the maximum likelihood utility of a candidate c_j given the observed estimates. For the moment, let us ignore our knowledge on the distribution of utility values, and assume that $u(c_j)$ is a hidden parameter of the model. Hence the only probabilistic contribution comes from the noise.

Denoting $s_j := (s_1(c_j), \dots, s_n(c_j))^T$ a score vector and $\Delta(s_j, u(c_j)) = (s_j - u(c_j) \mathbf{1}_{n \times 1})$, Equation 6 implies:

$$\mathbb{P}(S(c_j) = s_j) \propto e^{-\frac{1}{2} \Delta(s_j, u(c_j))^T \Sigma^{-1} \Delta(s_j, u(c_j))}. \quad (7)$$

To find the maximum likelihood estimation u_j^* of the unknown value $u(c_j)$, we just need to maximize the above expression. Denoting $\omega = (\omega_1, \dots, \omega_n)$ the vector $\mathbf{1}_{1 \times n} \Sigma^{-1}$, logarithmic derivation gives the following solution:

$$u_j^* = \frac{\sum_{i=1}^n \omega_i s_i(c_j)}{\sum_{i=1}^n \omega_i}. \quad (8)$$

Thus, u_j^* is just a well-chosen weighted average of the observed scores. Intuitively, the weights are lower for correlated agents to account for their redundancy. One can check that in the case of fully correlated groups (each agent belongs to a single feature, no distinct noise), the weight of an agent is the inverse of the size of its group. In particular, we have the intuitive result that if all agents are independent (each belong to a group of size 1), the simple average of the observed score is the best estimator.

Now assume that we also know the distribution of the utility values. For simplicity, we also assume that its probability density function (PDF) can be written as $\exp(f(u))$, where f is twice differentiable. By a similar likelihood maximization computation as above, we find a new solution u_j^+ that meets:

$$0 = f'(u_j^+) + \Omega(u_j^* - u_j^+), \text{ with } \Omega := \sum_{i=1}^n \omega_i. \quad (9)$$

Implicitly, this defines u_j^+ as a function of u_j^* . By implicit differentiation, we obtain:

$$\frac{du_j^*}{du_j^+} = 1 - \frac{f''(u_j^+)}{\Omega}. \quad (10)$$

If f'' is always smaller than Ω (and in particular if it is nonpositive), then u_j^* is increasing wrt u_j^+ . This means that selecting the best candidate according to u^+ or according to u^* is equivalent. In other words, the selection of the best candidate is not improved by knowing the utility distribution, as long as its PDF is log-concave, which is the case for a normal distribution.

³ github.com/TheoDlmz/embedded_voting.

We call Model-Aware (MA) the rule that selects the candidate with (equivalently) the highest u_j^+ or u_j^* . Since it relies on the best possible estimate (in the sense of maximum likelihood), we argue that Model-Aware (MA) should be a near-perfect solution. In our particular use case, it happens to be equivalent to the method BLUE [21, 22], even if the rationale behind BLUE is based on minimizing the variance and not on maximizing likelihood.

That being said, our paper assumes that the noise model is unknown to the observer in practice. Hence, while MA gives a nice tool to estimate an upper bound of performance, it is not a usable rule as it cannot be computed only from the observed scores.

However, the covariance matrix Σ of the noise can be loosely approximated from the covariance $\hat{\Sigma}$ of the observed scores, which combines the noise of the agents and the utility variations of the candidates. The difference between Σ and $\hat{\Sigma}$ comes from two factors: Σ and $\hat{\Sigma}$ do not correspond to the same random variable; $\hat{\Sigma}$ is an estimate based on observation.

In details, the utility variations introduce a global correlation between agents, so the random variable measured by $\hat{\Sigma}$ is not the true noise: all correlations are over-estimated. If the noise dominates utility variation, it stays an accurate approximation. On the other hand, if the noise is small compared to utility variation, the over-estimation of agent correlations will make the weights ω in Equation 8 sub-optimal (they will be more uniform than if Σ was used), but we argue that if the noise is small the impact on performance should be limited.

Also note that if the number of candidates is low, $\hat{\Sigma}$ is noisy due to the lack of observations. Using $\hat{\Sigma}$ instead of Σ gives a rule that can be used in practice, which we call Pseudo-Likelihood (PL).

3.3 Summary of the Rules Considered

We can now summarize all the rules that we want to evaluate, and specify some implementation details.

Unless otherwise stated, each agent rescales its scores to have a null mean and a unit standard deviation, so that agents with different score scales can be compared. This is not necessary in our experimental setting where all agents have the same marginal distribution but it may be required in practical applications.

All the rules that we study aggregate the estimated scores in a welfare measure w in order to select the winning candidate.

For Range Voting (RV), the welfare is $w(c_j) := \sum_{a_i \in \mathcal{A}} s_i(c_j)$.

For Approval Voting (AV), the welfare is $w(c_j) := \sum_{a_i \in \mathcal{A}} \mathbf{1}_{s_i(c_j) \geq 0}$: each agent computes a threshold as its average score across candidates and approve of all the candidates with a higher score. AV often generates ties, so we investigated several reasonable tie-breaking rules, like the sum, the product of the positive scores, or a fixed arbitrary order over the candidates. The results, not displayed in this paper, are relatively similar with a slight advantage for the product of positive scores, which we will then use.

The Nash Product (NP) and Embedded Voting (EV) require non-negative estimates. To do so, for both rules, each agent first rescales its observations to have unit standard deviation and mean 2 (i.e. two standard deviations).

For NP, we need positive values so the estimates that are still below 0.1 are replaced with 0.1. Then $w(c_j) := \prod_{a_i \in \mathcal{A}} s_i(c_j)$.

For EV, we only need non-negative values so the threshold is set to 0 instead. EV then uses $w(c_j) := \prod_{l=1}^k \lambda_l^2$, where $\lambda_1, \dots, \lambda_k$ are the \hat{k} greatest singular values of the score matrix \hat{M}_j and \hat{k} is the estimated number of relevant dimensions, computed as described in Section 2.3. The embedding vectors $(\hat{e}_i)_i$ of the agents, used to

define \hat{S} and \hat{M}_j , are based on the matrix of scores of the agents on the candidates considered for the problem. If we have access to other candidates, like candidates considered in previous choices, we can increase the precision of the method by considering all of them (from past and current decisions) to build the embedding vector \hat{e}_i of each agent i . This defines a *trained* version of EV based on previous observations, which we call Embedded Voting + (EV+).

Model-Aware (MA) computes the welfare from Equation 8, where the weight vector ω is deduced from the noise model. As the underlying noise model cannot be known by a rule in general, MA is essentially an upper reference point. We also consider Pseudo-Likelihood (PL), which approximates the noise model from the observed scores. As for EV, it is possible to use past candidates to enhance the precision of the approximation. We call Pseudo-Likelihood + (PL+) the *trained* version of PL based on previous observations, which uses all candidates from previous and current decisions to build the estimate $\hat{\Sigma}$ of the covariance matrix.

Unless otherwise stated, EV+ and PL+ use a training set of 1,000 candidates (not including the 20 candidates of the current choice).

Additionally, we consider Single Agent (SA), a rule that uses the estimated score of a single agent (no gain from aggregation) and Random Winner (RW), a rule that selects a candidate randomly regardless of the estimations. These two rules serve as lower reference points for the other rules.

3.4 Study of a Reference Scenario

We start our evaluation with the same setting as in the example of Section 2.1: a correlated group of 20 agents and 4 additional independent agents. This case can be described with a 24×5 embedding matrix defined by blocks as $E = \begin{pmatrix} \mathbb{1}_{20 \times 1} & 0 \\ 0 & I_4 \end{pmatrix}$, where I_4 is the 4×4 identity matrix. We use a feature noise intensity $\sigma_f = 1$ (i.e. equal to the standard deviation of utility values between candidates) and a distinct noise intensity $\sigma_d = 0.1$. The avowed purpose of this setting is to demonstrate the problems caused by correlations, by considering a large correlated group and a high feature noise compared to the distinct noise.

The average relative utility for each rule is displayed in Figure 1. The lower reference point is RW with 50%. Clearly, as long as the distribution of utility values is symmetric, it will always be the case, and we will omit RW from the figures. As expected, the upper reference point is MA, with 95%.

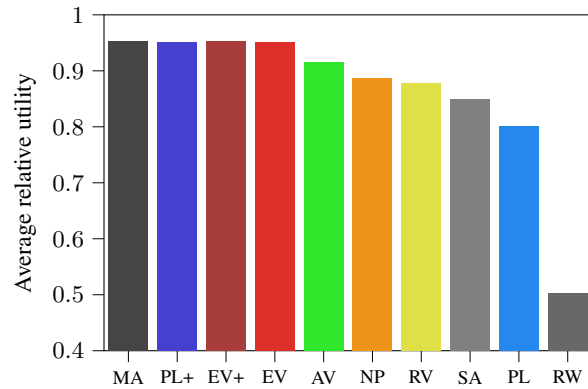


Figure 1: Performance in our reference scenario.

We first observe the relatively poor performance of RV (88%),

which is only slightly above the use of a single agent (85%). This is a case of *tyranny of the majority* [29]: the group of 20 agents concentrates most of the decision power but its *collective wisdom* is close to the one of a single agent because of the correlations. NP has similar performance (89%). Surprisingly, the approval variant of RV, AV, performs significantly better despite its simplicity (92%). Intuitively, as AV compresses the range of estimations, it prevents a large error from the group in favor of a candidate to dominate the decision.

Then, we see that the three rules EV, EV+, and PL+ have remarkable performance (95%). The gain compared to the next best rule, AV, is small in percentage (3%), but important as we are close to the optimal: in terms of *regret*, i.e. difference with the true optimal choice, we go from 8% to only 5%. Moreover, the difference with the upper reference MA is within the margin of error (which is of the order of 1%, because the sample size is 10,000).

On the other hand, the performance of PL is remarkably poor (80%): still much better than returning a random candidate (50%) but worse than only relying on the estimate of one single agent (85%).

4 Impact of the Parameters

In order to examine more broadly the performance of the rules considered, we now alter some parameters in isolation, while the others keep their values from the reference scenario.

4.1 Number of Agents

To start with, Figure 2 varies the size of the correlated group, which is 20 in the reference scenario, from 1 (i.e. 5 independent agents) to 30. Other things being equal, a larger correlated group should provide marginally better performance for optimal methods: the law of large numbers tend to cancel out the distinct noises inside the correlated group. On the other hand, a larger correlated group also pollutes the pool of scores with redundant information.

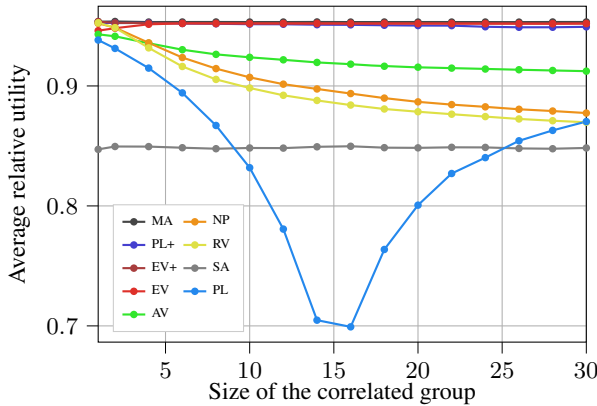


Figure 2: Changing the size of the correlated group.

We first observe that PL+, EV+, and EV are very stable at 95%, within the margin of error with MA: they manage to constantly ignore the redundant information. RV is optimal in the independent case but its performance decreases for a large group (87%), where it tends to ignore the scores of the independent agents. NP shows similar results (88% for a large group). AV is slightly sub-optimal in the independent case (94%), but it is more robust and becomes competitive for large groups (91%). Lastly, PL is insufficiently trained

with the candidates of the current choice only and behaves erratically, with minimal performance around 15 correlated agents due to ranks effects in the correlation matrix⁴.

Figure 3 shows the companion experiment investigating the impact of the number of independent agents (4 in the reference scenario). More independent agents give more information. The best rules PL+, EV+ and EV take advantage of the additional information, and so do AV, NP and RV to a lesser extent. PL is constantly worse than a single agent.



Figure 3: Changing the number of uncorrelated agents.

4.2 Number of Candidates

Figure 4 varies the number of candidates. More candidates means more “good” ones, hence a rule can afford to choose a good candidate even if it misses the best one. This explains the global increase of relative utility with the number of candidates. In the extreme case of 2 candidates, the non-trained EV is significantly less efficient (75%) than MA, PL+ and EV+ (86%), but with at least 10 candidates, the difference is within the margin of error: EV efficiently exploits the candidates of the current election as a training set to identify the correlations between agents. The performance of PL is minimal around $m = n$, for the same reason as in Figure 2.

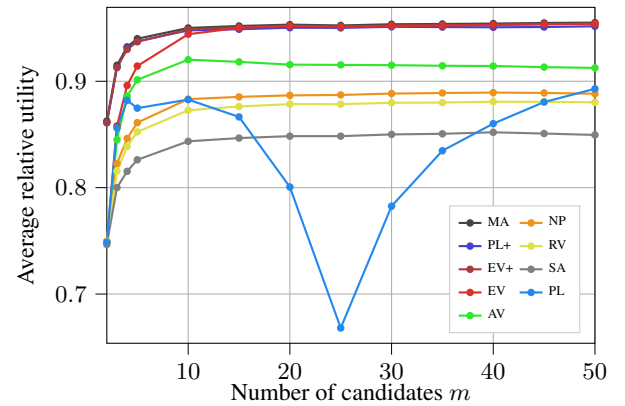


Figure 4: Changing the number of candidates.

⁴ When $m \approx n$, the conditioning of the estimated correlation matrix $\hat{\Sigma}$ is bad. With noise, its pseudo-inverse creates artifacts in the weight vector ω used in Equation 8: some components become negative so the result is out of the convex hull of the agents’ estimations. This explains why PL behaves worse than selecting a single method.

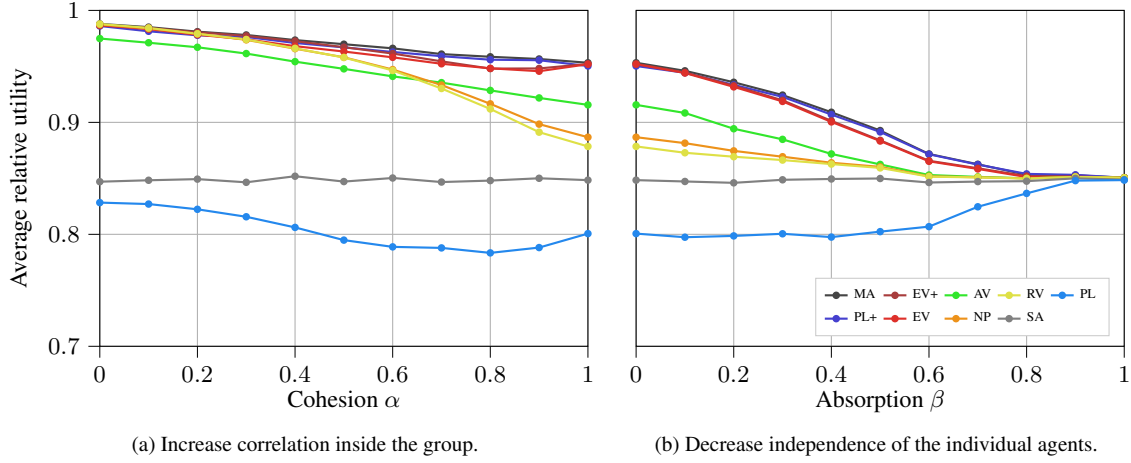


Figure 6: Modulating the correlations, from all agents being independent ($\alpha = 0$ on Figure 6a) to one unique correlated group ($\beta = 1$ on Figure 6b). The values $\alpha = 1$ on Figure 6a and $\beta = 0$ on Figure 6b correspond to the reference scenario (Figure 1).

4.3 Noise Intensity

In Figure 5, we change the intensities σ_d and σ_f , corresponding to the distinct noise and the feature noise respectively, from 0.1 to 10, while the variance of the true utility distribution stays equal to 1. Unsurprisingly, when the noise increases, the performance degrades for all rules. PL+ stays on par with MA in all settings. EV+ drops slightly only when both σ_d and σ_f are high. In addition to this case, the untrained version EV is also sub-optimal when $\sigma_d = \sigma_f = 1$. However, it remains the best of all untrained rules for all settings.

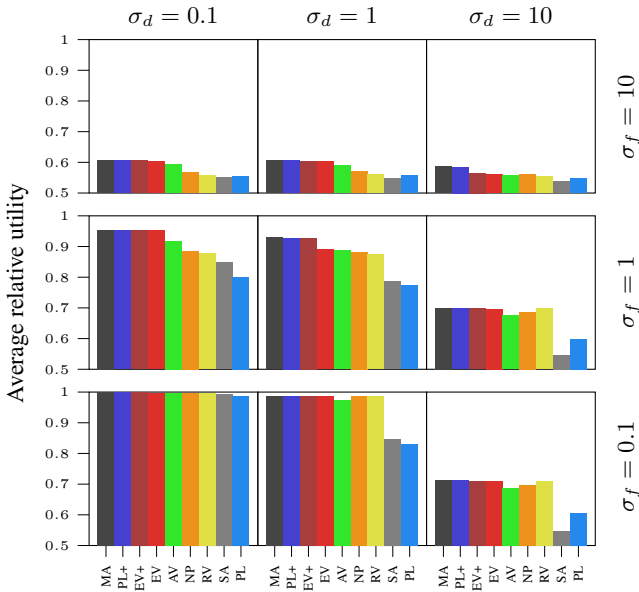


Figure 5: Changing the noise intensity

4.4 Soft Partition of the Agents

The reference scenario represents a clear partition of the agents into a group of 20 and 4 individuals. Here we modify the embedding E to account for more subtle correlations.

For example, the correlation between two agents a_i and $a_{i'}$ from the group may depend on i and i' (e.g. the agents are algorithms

and their indices represent a varying hyperparameter). For some $\alpha \in [0, 1]$, this can be modeled by a 24×24 embedding matrix defined by blocks as $E_\alpha = \begin{pmatrix} G_\alpha & 0 \\ 0 & I_4 \end{pmatrix}$, where $(G_\alpha)_{i,i'} \propto \alpha^{|i-i'|}$. In this model, the correlation between two agents of the group depends on the distance between their respective indices. The model parameter α , called *cohesion*, controls the group correlations: if $\alpha = 0$, all agents of the group are independent; if $\alpha = 1$, the model is equivalent to the reference scenario.

Another case that makes the partition less binary is if the independent agents can be influenced, to some degree, by the opinion of the group. For some $\beta \in [0, 1]$, this can be modeled by a 24×5 embedding matrix defined by blocks as $E_\beta = \begin{pmatrix} \mathbb{1}_{20 \times 1} & 0 \\ c\beta \mathbb{1}_{4 \times 1} & c(1-\beta)I_4 \end{pmatrix}$, with $c = \frac{1}{\sqrt{\beta^2 + (1-\beta)^2}}$. The model parameter β , which we call *absorption*, controls the influence of the group on the independent agents: if $\beta = 0$, which corresponds to the reference scenario, the influence is null; if $\beta = 1$, all 24 agents form a unique group.

Figure 6 show the evolution of the utility when α and β vary respectively. It can be read as one unique figure ranging from 24 independent agents ($\alpha = 0$) to one unique group ($\beta = 1$), with the reference scenario in the middle ($\alpha = 1$ or $\beta = 0$). The results are in line with what we have observed so far: if the agents are fully independent or correlated, any decent rule like RV is optimal, but in the presence of non-trivial correlations, the rules PL+ and EV+ are virtually indistinguishable from MA. EV has slightly lower performance (the difference is of the order of the margin of error) but clearly remains the best of all untrained rules. This remains true even if the correlations represent something more complex than a simple partition of the agents.

5 Conclusion

In the context of aggregating correlated noisy agents in a choice problem, we have proposed Embedded Voting, a method that embeds the agents according to the scores they produce, and we have compared its performance with a variety of other methods. Our main findings are that if a sufficient score history of the agents is available, then a maximum-likelihood approach is the best option. On the other hand, if the training set is limited, EV should be preferred, as it is robust and outperforms other untrained methods.

Acknowledgments

This work was done at LINCS (<https://www.lincs.fr/>).

Théo Delemazure was supported by the PRAIRIE 3IA Institute under grant ANR-19-P3IA-0001 (e).

References

- [1] Peter Auer, ‘Using confidence bounds for exploitation-exploration trade-offs’, *J. Mach. Learn. Res.*, **3**, 397–422, (mar 2003).
- [2] Haris Aziz, Anna Bogomolnaia, and Hervé Moulin, ‘Fair mixing: The case of dichotomous preferences’, in *Proceedings of the 2019 ACM Conference on Economics and Computation*, pp. 753–781, (2019).
- [3] Daniel Billsus and Michael J. Pazzani, ‘Learning collaborative information filters’, in *Proceedings of the Fifteenth International Conference on Machine Learning*, ICML ’98, p. 46–54, San Francisco, CA, USA, (1998). Morgan Kaufmann Publishers Inc.
- [4] Florian Brandl, Felix Brandt, Matthias Greger, Dominik Peters, Christian Stricker, and Warut Suksompong, ‘Funding public projects: A case for the nash product rule’, *Journal of Mathematical Economics*, **99**, 102585, (2022).
- [5] Felix Brandt, Vincent Conitzer, Ulle Endriss, Jérôme Lang, and Ariel D. Procaccia, *Handbook of Computational Social Choice*, Cambridge University Press, USA, 1st edn., 2016.
- [6] Margarita Cabrera-Bean, Alba Pagès-Zamora, and Carles Díaz-Vilor, ‘Unsupervised ensemble classification with correlated decision agents’, *IEEE Signal Processing Letters*, **26**(7), 1085–1089, (2019).
- [7] Ioannis Caragiannis, David Kurokawa, Hervé Moulin, Ariel D Procaccia, Nisarg Shah, and Junxing Wang, ‘The unreasonable fairness of maximum nash welfare’, *ACM Transactions on Economics and Computation (TEAC)*, **7**(3), 1–32, (2019).
- [8] Ioannis Caragiannis, Ariel D. Procaccia, and Nisarg Shah, ‘When do noisy votes reveal the truth?’, *ACM Trans. Econ. Comput.*, **4**(3), (mar 2016).
- [9] Marie Jean Antoine Nicolas de Caritat, marquis de Condorcet, *Essai sur l'application de l'analyse à la probabilité des décisions rendues à la pluralité des voix*, Imprimerie royale, 1785.
- [10] Vincent Conitzer and Tuomas Sandholm, ‘Common voting rules as maximum likelihood estimators’, *CoRR*, **abs/1207.1368**, (2012).
- [11] Scott R Eliason, *Maximum likelihood estimation logic and practice*, Quantitative applications in the social sciences; 96, SAGE, Newbury Park, [Calif.] ; London, 1993.
- [12] Till Fluschnik, Piotr Skowron, Mervin Triphaus, and Kai Wilker, ‘Fair knapsack’, *Proceedings of the AAAI Conference on Artificial Intelligence*, **33**(01), 1941–1948, (2019).
- [13] Seymour Geisser, ‘A Bayes approach for combining correlated estimates’, *Journal of the American Statistical Association*, **60**(310), 602–607, (1965).
- [14] Pablo M. Granitto, Pablo F. Verdes, and H. Alejandro Ceccatto, ‘Neural network ensembles: Evaluation of aggregation algorithms’, *Artificial Intelligence*, **163**(2), 139–162, (2005).
- [15] Otaviano Helene, Leandro Mariano, and Zwinglio Guimarães-Filho, ‘Useful and little-known applications of the least square method and some consequences of covariances’, *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, **833**, 82–87, (2016).
- [16] Ariel Jaffe, Ethan Fetaya, Boaz Nadler, Tingting Jiang, and Yuval Kluger, ‘Unsupervised ensemble learning with dependent classifiers’, in *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, eds., Arthur Gretton and Christian C. Robert, volume 51 of *Proceedings of Machine Learning Research*, pp. 351–360, Cadiz, Spain, (09–11 May 2016). PMLR.
- [17] Mamoru Kaneko and Kenjiro Nakamura, ‘The Nash social welfare function’, *Econometrica*, **47**(2), 423–435, (1979).
- [18] Matthaeus Kleindessner and Pranjal Awasthi, ‘Crowdsourcing with arbitrary adversaries’, in *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pp. 2708–2717. PMLR, (2018).
- [19] Ludmila Kuncheva, *Combining Pattern Classifiers: Methods and Algorithms*, Wiley Publishing, 2nd edn., 2014.
- [20] John F. Nash, ‘The bargaining problem’, *Econometrica*, **18**(2), 155–162, (1950).
- [21] Richard Nisius, ‘On the combination of correlated estimates of a physics observable’, *The European Physical Journal C*, **74**(8), (aug 2014).
- [22] Richard Nisius, ‘BLUE: Combining correlated estimates of physics observables within ROOT using the best linear unbiased estimate method’, *SoftwareX*, **11**, 100468, (jan 2020).
- [23] Robi Polikar, ‘Ensemble based systems in decision making’, *IEEE Circuits and Systems Magazine*, **6**(3), 21–45, (2006).
- [24] Ariel D Procaccia and Nisarg Shah, ‘Is approval voting optimal given approval votes?’, in *Advances in Neural Information Processing Systems*, eds., C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, volume 28. Curran Associates, Inc., (2015).
- [25] Michael Schmelling, ‘Averaging correlated data’, *Physica Scripta*, **51**(6), 676, (Jun 1995).
- [26] Michael Schmelling, ‘Averaging measurements with hidden correlations and asymmetric errors’, *arXiv preprint hep-ex/0006004*, (2000).
- [27] Warren D Smith, ‘Range voting’, (2000).
- [28] Clifton D Sutton, ‘Classification and regression trees, bagging, and boosting’, *Handbook of Statistics*, **24**, 303–329, (2005).
- [29] Alexis de Tocqueville, *De la démocratie en Amérique II*, Gosselin, 1840.
- [30] Michael E. Wall, Andreas Rechtsteiner, and Luis M. Rocha, *Singular Value Decomposition and Principal Component Analysis*, 91–109, Springer US, Boston, MA, 2003.