Aggregating Correlated Estimations with (Almost) no Training

Théo Delemazure
LAMSADE, Université Paris Dauphine-PSL, CNRS

François Durand
Nokia Bell Labs France

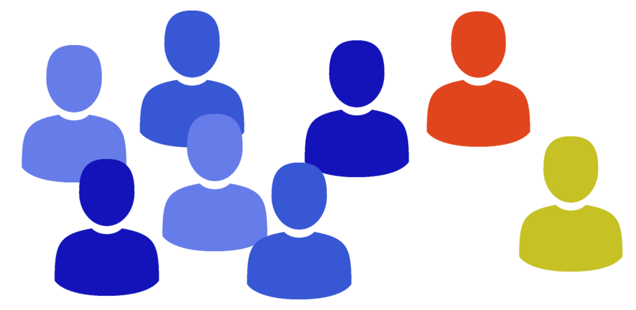
Fabien Mathieu
Swapcard

The problem

We want to choose an item **that maximizes utility** among a set of *candidates*.

However, we only have access to **noisy estimates of the items' utilities** by human or software *agents*.

Problem: What if agents are (heavily) correlated?



If all **blue** agents have similar estimates, **we should not** take the average.

The usual way to solve this problem is to **assume diversity** among agents.
...But what if we don't?

Question: What aggregation method should we use to avoid drawbacks due to correlations?

The challengers

Range voting (RV)
Select the candidate that maximizes the *sum* of estimates.

Approval voting (AV)
Select the candidate that maximizes the *number of agents* who estimate its utility greater than the average.

Nash product (NP)
Select the candidate that maximizes the *product* of estimates.

Model Aware (MA)
Maximum likelihood approach, knowing the *noise model* used to generate the estimates *and its parameters*.

Pseudo Likelihood (PL)
Maximum likelihood approach, where the parameters of the models are *evaluated* from observations of the estimates.

Pseudo Likelihood + (PL+)
With training on *1000 past observations*.

Our proposal: Embedded Voting (EV)

Using the **Singular Value Decomposition** (SVD) of the matrix of estimates, we can identify the different *"groups"* of voters. **The EV score** is the product of the estimates of each group (where the estimate is one of the singular values). For **EV+**, the estimates matrix contains 1000 past observations.

Experiments

Parameters of the noise model

$E = (e_{i,l})_{1 \le i \le n, 1 \le l \le k}$: *features* of the n agents.
$\sigma_f \in \mathbb{R}_{\ge 0}$: *feature* noise intensity.
$\sigma_d \in \mathbb{R}_{\ge 0}$: *distinct* noise intensity.

Error of agent i for candidate j

$$\varepsilon_i(c_j) \coloneqq \sigma_d d_{i,j} + \sigma_f \sum_{1 \le l \le k} e_{i,l} f_{l,j},$$

where $d_{i,j} \sim \mathcal{N}(0,1)$ and $f_{l,j} \sim \mathcal{N}(0,1)$.

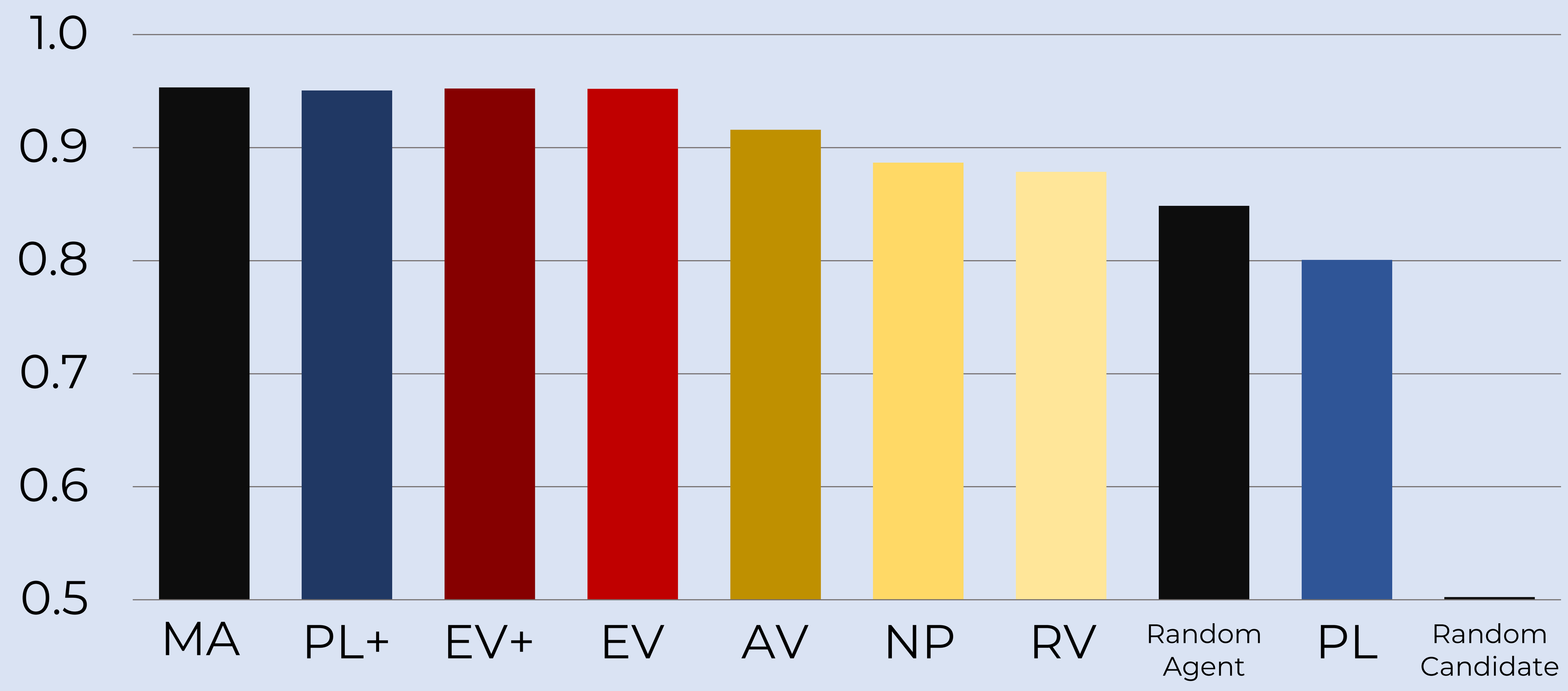
Reference scenario

- One group of 20 agents and 4 independent agents:

$$E = \begin{pmatrix} \mathbb{1}_{20\times 1} & 0 \\ 0 & I_4 \end{pmatrix}$$

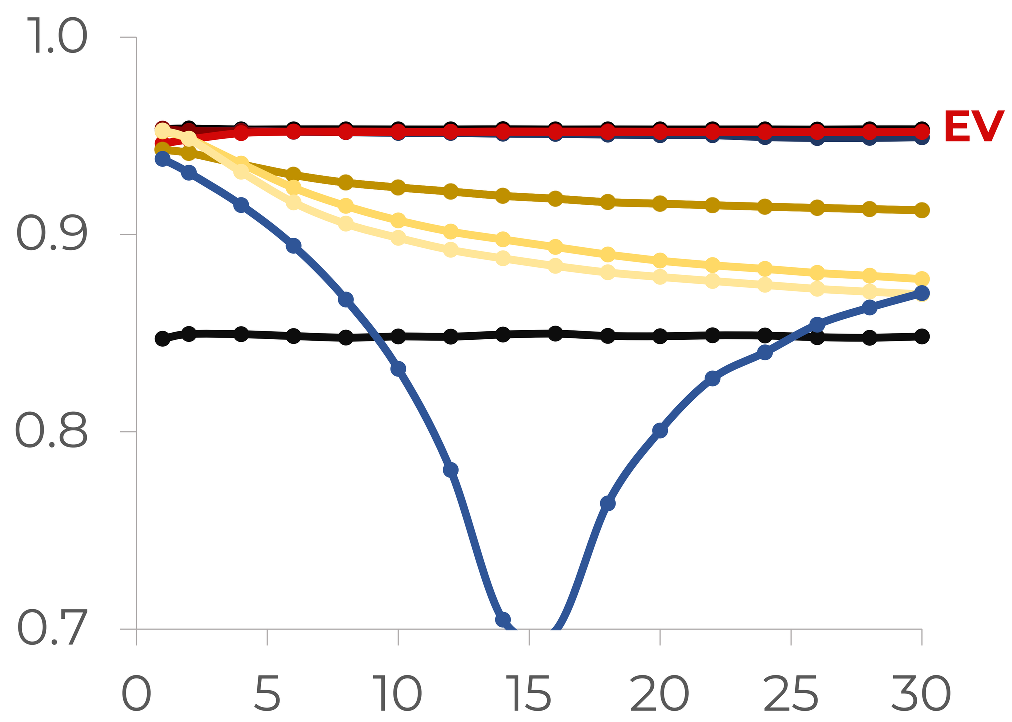
- $\sigma_f = 1$ and $\sigma_d = 0.1$.

We compute **the average relative utility** obtained by each rule over 10,000 choices.

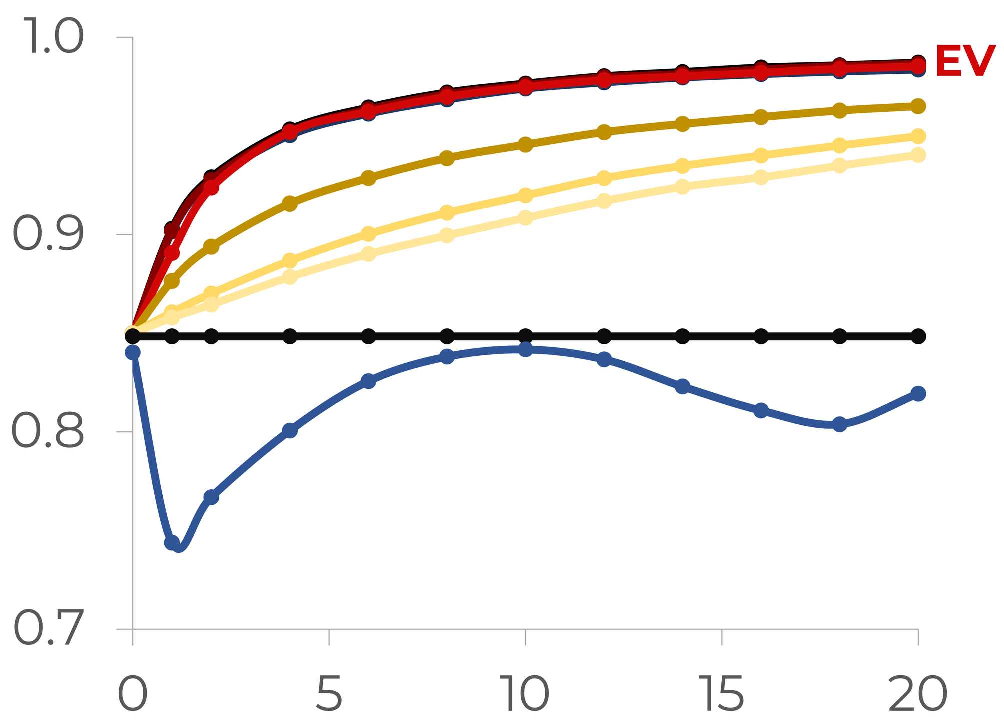


The performances of **E**mbedded **V**oting stay competitive **when we vary...**

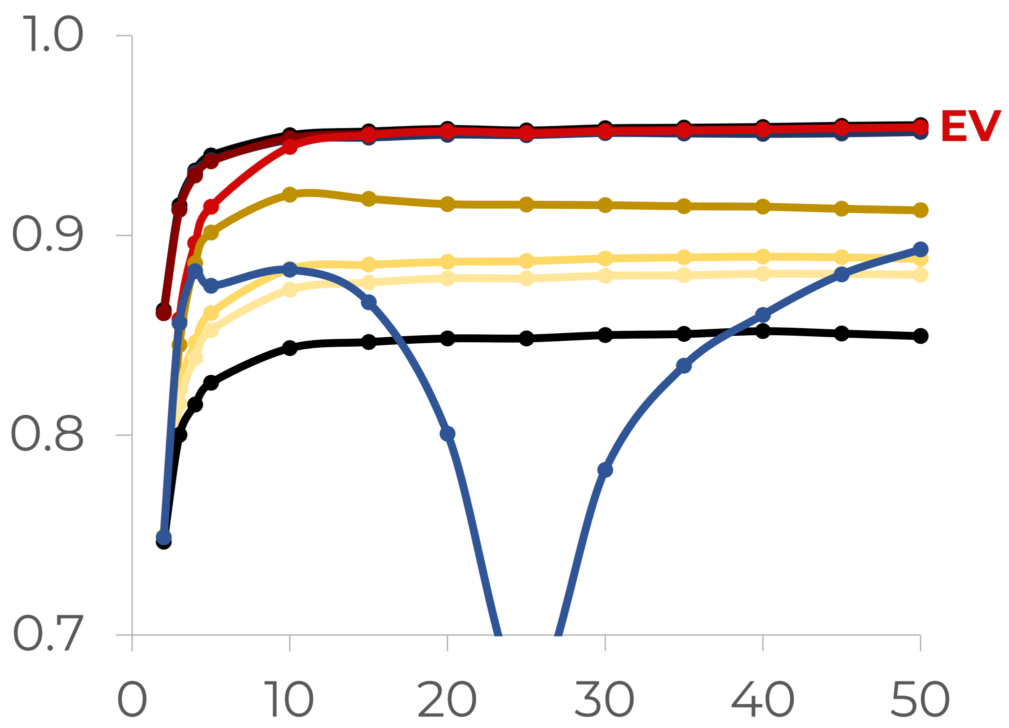
...the **number of agents in the large group**.



...the **number of independent agents**.



...the **number of candidates**.



...the feature and distinct **noise intensities**.

...the noise **distribution functions.**

...the **correlation degree** between the agents.

...the probability **distribution of utilities.**

Take-away

Context
Aggregating correlated agents in a choice problem.

Our proposal
Embedded Voting (EV), that uses SVD to embed the agents according to the estimates they produce.

Our results
1. Our method *outperforms* classical ones, particularly when agents are correlated.
2. When a training set is available, a *maximum likelihood* approach is the best option.
3. If there is no such training, *Embedded Voting* should be preferred.

Our paper
hal-04195384

Our Python Package
embedded_voting

The 26th **E**uropean **C**onference on **A**rtificial **I**ntelligence (**ECAI 2023**)