Automated Design of Voting Rules by Learning From Examples

Ariel D. Procaccia, Aviv Zohar, Jeffrey S. Rosenschein

Outline

- Scoring rules:
  - Definition
  - Advantages
- Our approach
  - Learning voting rules in the PAC model
  - Main theorem
- Limitations of our approach
- Conclusions

Scoring rules

Scoring rules	Learning	Limitations	Conclusions

- Election: set of voters $N=\{1,\ldots,n\}$, set of candidates/alternatives $A=\{x_1,\ldots,x_m\}$. Voters express linear preferences R^i over A.
- Winner determined according to a voting rule/social choice function.
- Scoring rules: defined by a vector $\alpha=\langle\alpha_1,\ldots,\alpha_m\rangle$, all $\alpha_i \geq \alpha_{i+1}$. Each candidate receives α_i points from every voter which ranks it in the i'th place.
- Examples:
  - Plurality: $\alpha=\langle 1,0,\ldots,0\rangle$
  - Veto: $\alpha=\langle 1,\ldots,1,0\rangle$
  - Borda: $\alpha=\langle m-1,m-2,\ldots,0\rangle$

On the diversity of scoring rules

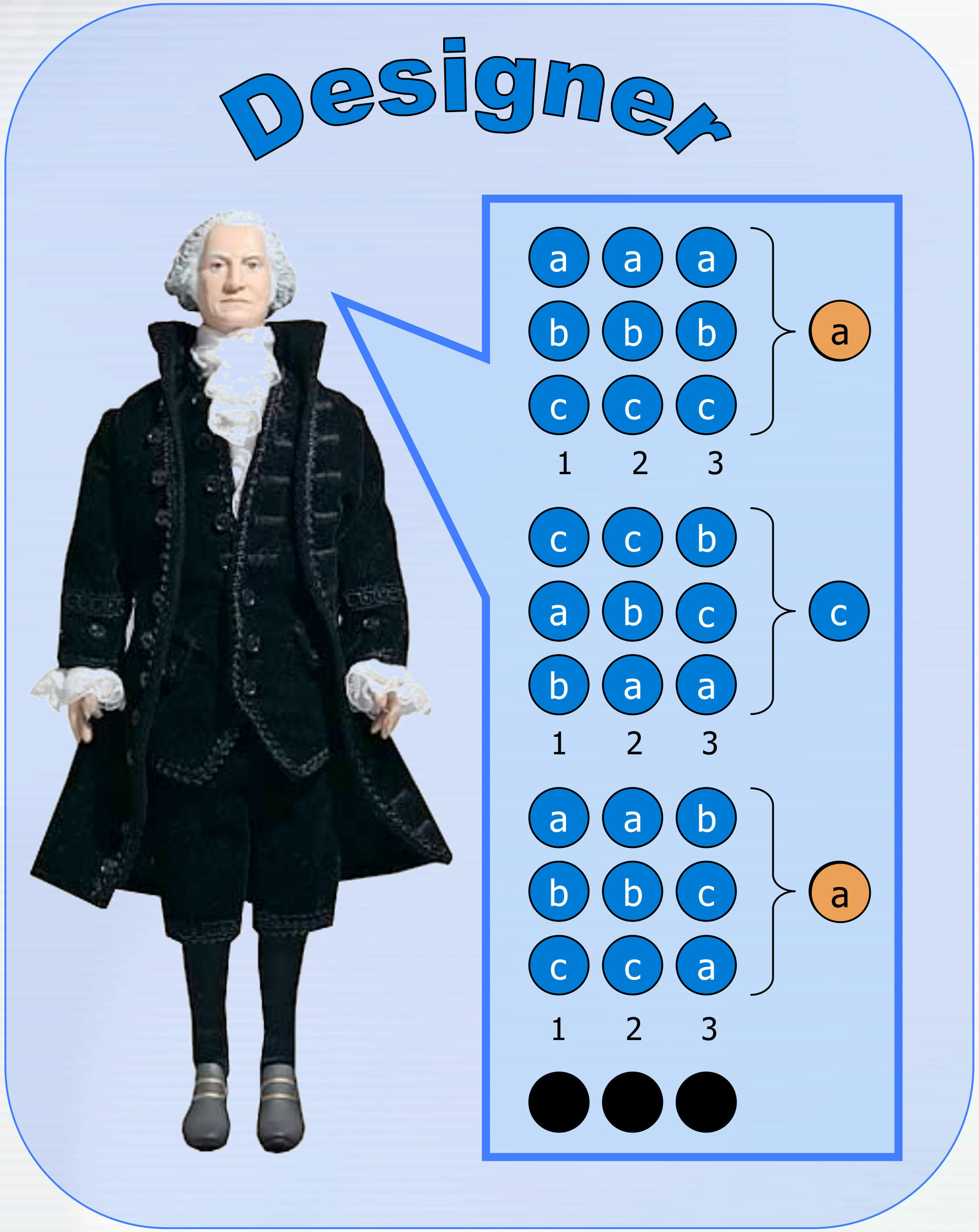
- Different choice of parameters result in different properties.
- Some properties:
  - Majority: candidate most preferred by majority is elected.
  - Robustness: worst-case prob. of the outcome not changing as a result of a fault.
  - Computational Complexity of coalitional manipulation.
  - Communication Complexity.

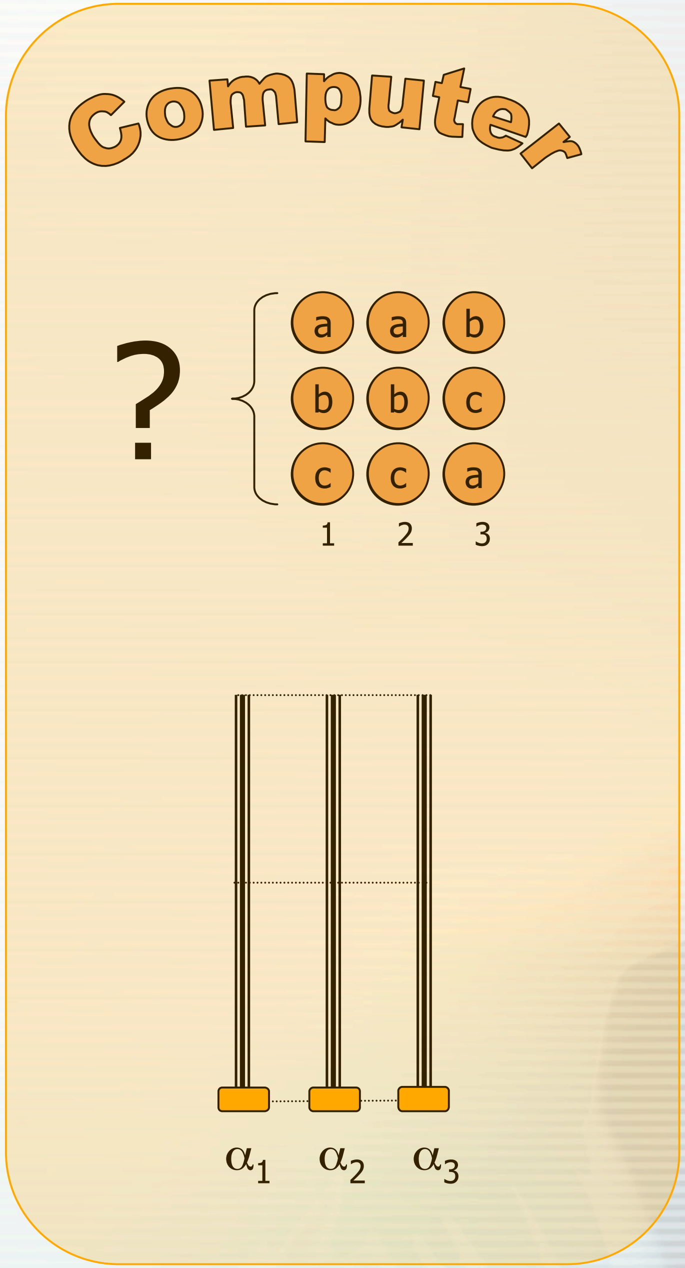
Rule	Majority	Robustness	Manipulation	Communication
Plurality	Yes	$\geq (m-2)/(m-1)$	P	$\Theta(n * \log m)$
Veto	No	$\geq (m-2)/(m-1)$	NP-complete	$O(n * \log m)$
Borda	No	$\leq 1/m$	NP-complete	$\Theta(n * m * \log m)$

Automated Design of voting rules

- Designer/teacher is presented with pref. profiles, and designates the winner in each.
- Philosophical justification.
- Practical justification: designer simply wants to find a concise representation.
- Assuming there exists a "target" scoring rule, the goal is to find a scoring rule which is "close".

An Illustration



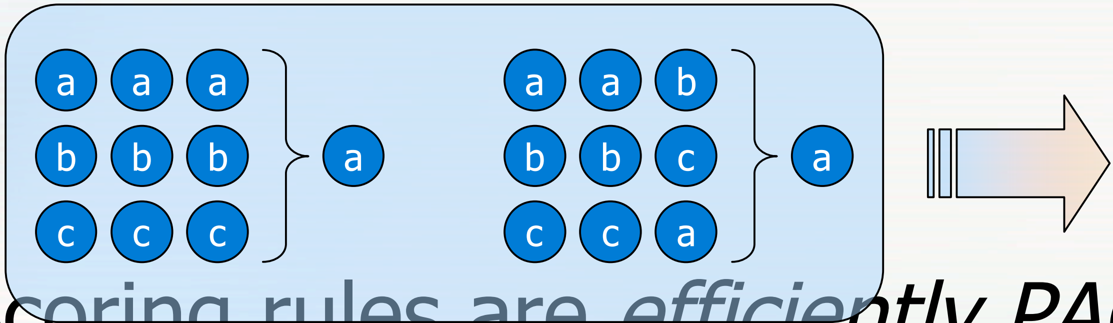


PAC Learning

- Training set consists of pairs of examples $(R_j, f(R_j))$.
- R_j are drawn from fixed dist. D.
- f = target scoring rule.
- Goal: given ε, find scoring rule g such that $Prob_D[f(R) \neq g(R)] \leq \varepsilon$.
- Q: How many examples are needed in order to guarantee that goal is achieved with prob. at least $1-\delta$?

PAC-learnability of scoring rules

- **Theorem:** If there are at least poly(n,m,1/ε,1/δ) examples in the training set, then any "consistent" scoring rule g achieves the goal.
- Such a rule can be efficiently found using LP.
- Example:



find $\alpha_1, \alpha_2, \alpha_3$ s.t.

$3\alpha_1 > 3\alpha_2$

$3\alpha_1 > 3\alpha_3$

$2\alpha_1 + \alpha_3 > \alpha_1 + 2\alpha_2$

$2\alpha_1 + \alpha_3 > \alpha_2 + 2\alpha_3$

$\alpha_1 \geq \alpha_2 \geq \alpha_3 \geq 0$

- Scoring rules are *efficiently PAC-learnable.*

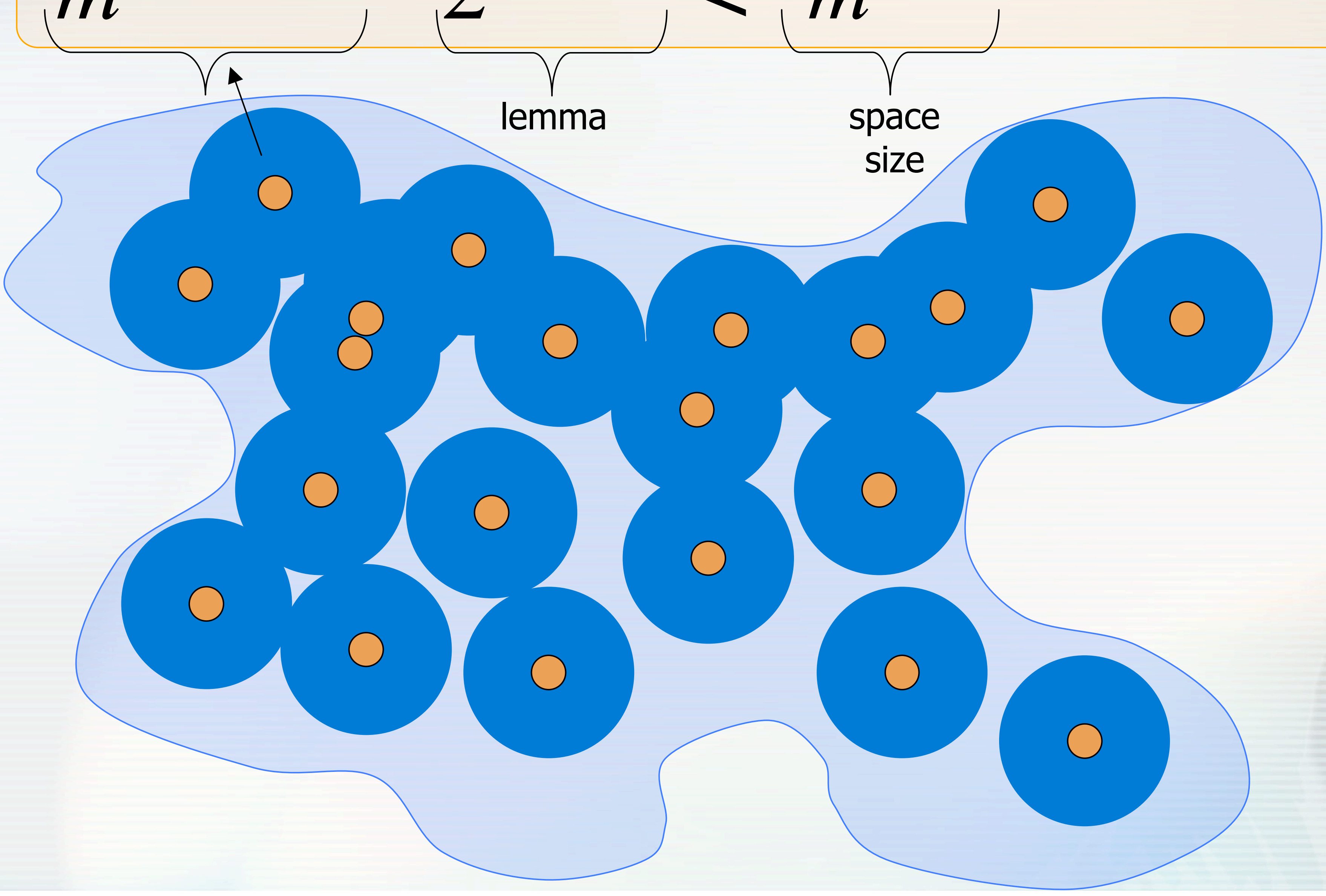
Limitations

Scoring rules	Learning	Limitations	Conclusions

- There are many different scoring rules.
- Can any voting rule be approximated by a scoring rule?
- **Definition:** g is a *c-approximation* of f iff f and g agree on a c-fraction of the possible preference profiles.
- Reformulation: given a voting rule f, how hard is it to learn a scoring rule which is a c-approximation, with c close to 1?
- **Theorem:** Let $\varepsilon>0$. For large enough n,m, $\exists$f such that no scoring rule is a $(1/2+\varepsilon)$-approximation of f.
- **Lemma:** $\exists$ polynomial p(n,m) s.t. the number of distinct scoring rules $\leq 2^{p(n,m)}$.

Proof of Theorem

$$m^{(1-\varepsilon)(m!)^n} \cdot 2^{p(n,m)} < m^{(m!)^n}$$



Conclusions

- If the designer can designate winners, then it can automatically design voting rule.
- Cumbersome representation $\rightarrow$ concise.
- Many voting rules cannot be approximated by scoring rules.
- Open questions:
  - Is there a broad class of rules which can be approximated by scoring?
  - Is there a broad class of rules which is efficiently learnable and concisely representable?