Optimizing positional scoring rules for rank aggregation

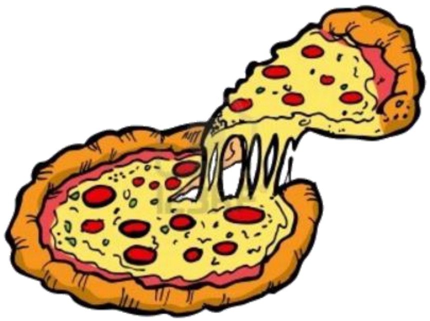
Ioannis Caragiannis
Xenophon Chatzigeorgiou
George Krimpas
Alexandros Voudouris

University of Patras

AAAI 2017

An application

- Find a pizza place at San Francisco to have lunch



- You can ask a local or … search in a recommendation website (like tripadvisor and yelp)
- Final decision based on *reviews* from previous visitors

An application

- Reviews: cardinal information
  - cost-effective? score: 4/5 stars
  - Tasty? score: 4.5/5 …

- Our study: reviews that contain ***ordinal*** information
- A user *compares* a small number of pizza places and comes up with a *ranking* of them
- The recommendation website *merges* the reviewing rankings and presents a ranking of all pizza places to the new user

The basic setting

- A set of m alternatives (= pizza places)
- A set of n agents (= previous visitors)
- Each agent provides a ranking of some alternatives
  - ***incomplete ranking*** of exactly d alternatives

Goal: merge the individual rankings into a single ranking of all alternatives

$\Rightarrow$ use ***positional scoring rules***

Example

agents
Iannis
Xenophon
George

pizza places
Montesacro
Victor’s
Del Popolo
Delarosa
Uncle Vito’s

Example

agents
Iannis
Xenophon
George

individual rankings		
Montesacro	Victor's	Del Popolo
Delarosa	Victor's	Uncle Vito's
Victor's	Montesacro	Uncle Vito's

pizza places
Montesacro
Victor's
Del Popolo
Delarosa
Uncle Vito's

Example

(positional) scoring rule		
4	2	1

agents	individual rankings		
Iannis	Montesacro	Victor’s	Del Popolo
Xenophon	Delarosa	Victor’s	Uncle Vito’s
George	Victor’s	Montesacro	Uncle Vito’s

pizza places
Montesacro
Victor’s
Del Popolo
Delarosa
Uncle Vito’s

Example

(positional) scoring rule		
4	2	1

agents	individual rankings		
Iannis	Montesacro	Victor’s	Del Popolo
Xenophon	Delarosa	Victor’s	Uncle Vito’s
George	Victor’s	Montesacro	Uncle Vito’s

pizza places	scores
Montesacro	
Victor’s	
Del Popolo	
Delarosa	
Uncle Vito’s	

Example

(positional) scoring rule		
4	2	1

agents	individual rankings		
Iannis	Montesacro	Victor's	Del Popolo
Xenophon	Delarosa	Victor's	Uncle Vito's
George	Victor's	Montesacro	Uncle Vito's

pizza places	scores
Montesacro	
Victor's	
Del Popolo	
Delarosa	
Uncle Vito's	

Example

(positional) scoring rule		
4	2	1

agents	individual rankings		
Iannis	Montesacro	Victor’s	Del Popolo
Xenophon	Delarosa	Victor’s	Uncle Vito’s
George	Victor’s	Montesacro	Uncle Vito’s

pizza places	scores
Montesacro	6
Victor’s	
Del Popolo	
Delarosa	
Uncle Vito’s	

Example

(positional) scoring rule		
4	2	1

agents	individual rankings		
Iannis	Montesacro	Victor’s	Del Popolo
Xenophon	Delarosa	Victor’s	Uncle Vito’s
George	Victor’s	Montesacro	Uncle Vito’s

pizza places	scores
Montesacro	6
Victor’s	7
Del Popolo	1
Delarosa	4
Uncle Vito’s	2

Example

(positional) scoring rule		
4	2	1

agents	individual rankings		
Iannis	Montesacro	Victor's	Del Popolo
Xenophon	Delarosa	Victor's	Uncle Vito's
George	Victor's	Montesacro	Uncle Vito's

pizza places	scores
Montesacro	6
Victor's	7
Del Popolo	1
Delarosa	4
Uncle Vito's	2

final ranking	
1st	Victor's
2nd	Montesacro
3rd	Delarosa
4th	Uncle Vito's
5th	Del Popolo

Example

(positional) scoring rule		
4	2	1

pizza places	scores
Montesacro	6
Victor’s	7
Del Popolo	1
Delarosa	4
Uncle Vito’s	2

final ranking	
1st	Victor’s
2nd	Montesacro
3rd	Delarosa
4th	Uncle Vito’s
5th	Del Popolo

Example

constraint	weight
Victor's ≻ Montesacro	3
Del Popolo ≻ Uncle Vito's	1
Delarosa ≻ Del Popolo	1

(positional) scoring rule		
4	2	1

pizza places	scores
Montesacro	6
Victor's	7
Del Popolo	1
Delarosa	4
Uncle Vito's	2

final ranking	
1st	Victor's
2nd	Montesacro
3rd	Delarosa
4th	Uncle Vito's
5th	Del Popolo

Example

constraint	weight
Victor's ≻ Montesacro	3
Del Popolo ≻ Uncle Vito's	1
Delarosa ≻ Del Popolo	1

(positional) scoring rule		
4	2	1

pizza places	scores
Montesacro	6
Victor's	7
Del Popolo	1
Delarosa	4
Uncle Vito's	2

final ranking	
1st	Victor's
2nd	Montesacro
3rd	Delarosa
4th	Uncle Vito's
5th	Del Popolo

Final ranking agrees with the constraint

Example

constraint	weight
Victor's ≻ Montesacro	3
Del Popolo ≻ Uncle Vito's	1
Delarosa ≻ Del Popolo	1

(positional) scoring rule		
4	2	1

pizza places	scores
Montesacro	6
Victor's	7
Del Popolo	1
Delarosa	4
Uncle Vito's	2

final ranking	
1st	Victor's
2nd	Montesacro
3rd	Delarosa
4th	Uncle Vito's
5th	Del Popolo

Final ranking DOES NOT agree with the constraint

Example

constraint	weight
Victor's ≻ Montesacro	3
Del Popolo ≻ Uncle Vito's	1
Delarosa ≻ Del Popolo	1

Total weight of satisfied constraints = 4

(positional) scoring rule		
4	2	1

pizza places	scores
Montesacro	6
Victor's	7
Del Popolo	1
Delarosa	4
Uncle Vito's	2

final ranking	
1st	Victor's
2nd	Montesacro
3rd	Delarosa
4th	Uncle Vito's
5th	Del Popolo

Example

constraint	weight
Victor's ≻ Montesacro	3
Del Popolo ≻ Uncle Vito's	1
Delarosa ≻ Del Popolo	1

(positional) scoring rule		
4	2	1

pizza places	scores
Montesacro	6
Victor's	7
Del Popolo	1
Delarosa	4
Uncle Vito's	2

final ranking	
1st	Victor's
2nd	Montesacro
3rd	Delarosa
4th	Uncle Vito's
5th	Del Popolo

Total weight of satisfied constraints = 4

Can we do better with a different scoring rule?

The OptPSR problem

- **Input**
  - A profile of incomplete rankings of size d
  - A set C of constraints
- **Output**
  - A positional scoring rule
- **Goal**
  - Maximize the total weight of the satisfied constraints

Theoretical results (overview)

✓ Exact algorithm that runs in time $O(|C|^d \cdot \text{poly}(|C|, d))$

⇒ poly-time when *d* is constant

✓ The **best** *t*-approval rule is *d*-approximate

- always produces a ranking that satisfies at least 1/d of the sum of all weights and there is an example where this is tight)

✓ Cannot approximate OptPSR within a constant factor

- reduce from the well-known APX-hard problem MAX-3-LIN2 [Hastad 2001]

Experiments

The *ppl* scenario

- A pool of **48** countries
- Assign (randomly) **6** countries per agent
- Each agent orders the countries w.r.t. **population**

The *col* scenario

- A pool of **36** cities
- Assign (randomly) **6** cities per agent
- Each agent orders the cities w.r.t. **cost of living**

Real-world data

Order the cities with respect to the cost of living from more expensive (1) to cheapest (6)

	Sydney, Australia
	Oslo, Norway
	Baghdad, Iraq
	Vienna, Austria
	Washington, USA
	London, UK

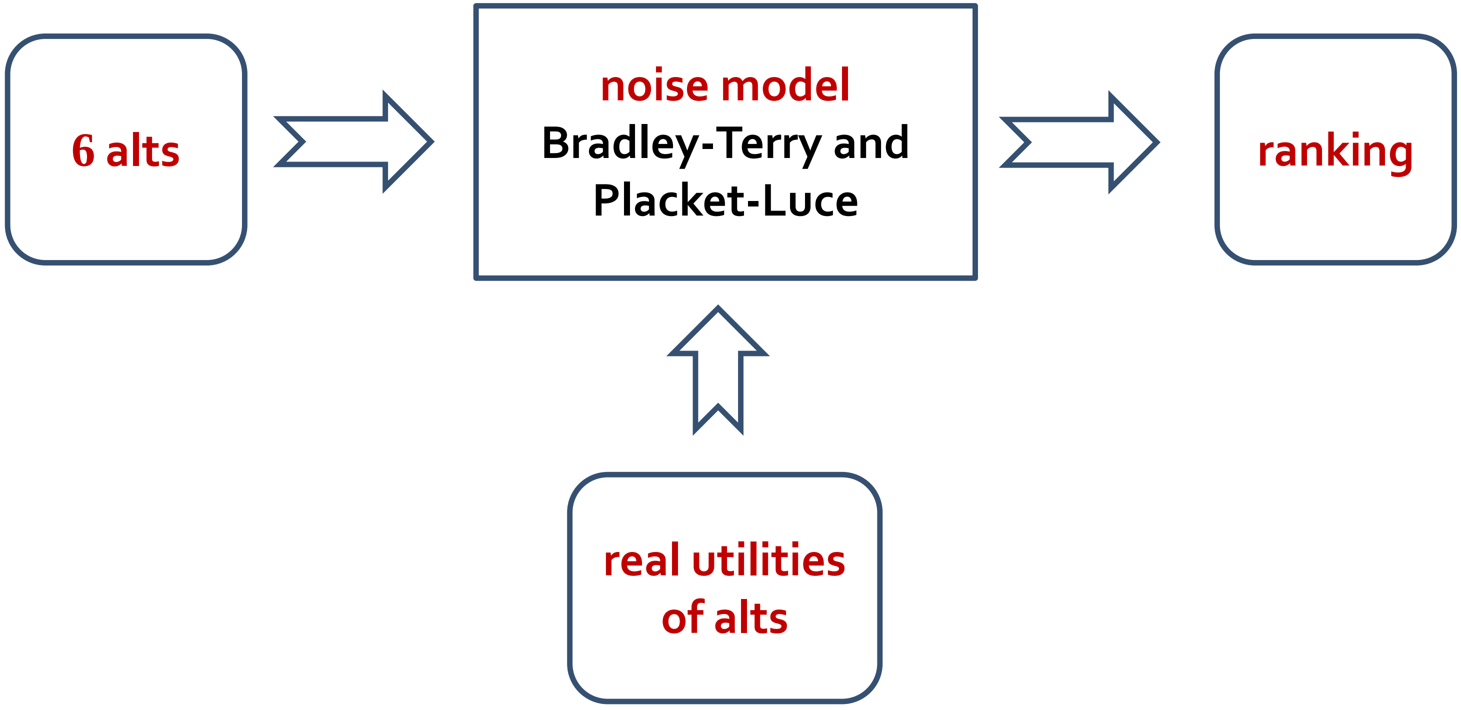
Order the countries with respect to population from the largest (1) to the smallest (6)

	Greece
	Israel
	Nigeria
	Thailand
	China
	Mexico

- Data collected from 392 individuals
- **Noisy rankings**

Synthetic data

- Simulate 392 agents that produce noisy ranking according to specific theoretical noise models



Results

scoring rule	real-world		BT model		PL model	
	ppl	*col*	*ppl*	*col*	*ppl*	*col*
opt	81.83	83.97	94.54	93.74	93.19	88.20
Borda	79.87	81.43	**93.16**	91.03	**91.59**	**84.67**
Harmonic	**80.94**	**82.54**	92.84	**91.34**	90.50	83.13
1-approval	77.75	78.09	83.69	87.89	83.90	75.72
2-approval	78.19	79.36	89.71	90.65	88.72	81.23
3-approval	79.43	80.48	91.69	90.93	90.30	83.73
4-approval	77.57	79.68	90.00	89.44	89.70	83.88
5-approval	73.14	72.86	81.08	84.45	82.83	80.66

$$w(x,y) = \begin{cases} 1, & \text{if } u_x > u_y \\ 0, & \text{otherwise} \end{cases}$$

Results

scoring rule	real-world		BT model		PL model	
	ppl	*col*	*ppl*	*col*	*ppl*	*col*
opt	95.98	92.93	99.66	98.69	99.48	96.02
Borda	94.56	91.57	**99.49**	97.66	**99.23**	**93.96**
Harmonic	**95.42**	**92.01**	99.42	**97.80**	99.02	92.58
1-approval	94.85	89.84	98.59	96.48	97.86	86.34
2-approval	95.24	90.40	99.29	97.67	98.86	91.50
3-approval	93.68	90.83	99.05	97.70	99.04	93.37
4-approval	92.63	90.06	97.46	96.91	98.38	93.51
5-approval	84.61	82.00	87.94	93.81	91.69	90.99

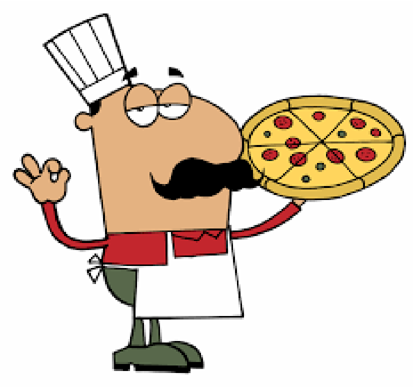
$$w(x,y) = \begin{cases} u_x - u_y, & \text{if } u_x > u_y \\ 0, & \text{otherwise} \end{cases}$$

Future work

- Can we solve optimally small instances, in reasonable time?
- Approximation algorithms for OptPSR
- Analysis of scoring rules with noisy models

Future work

- Can we solve optimally small instances, in reasonable time?
- Approximation algorithms for OptPSR
- Analysis of scoring rules with noisy models



Thank you!