FAKULTÄT FÜR !NFORMATIK

Faculty of Informatics

ALGORITHMS AND COMPLEXITY GROUP

What Complexity Theory can tell us about Judgment Aggregation

Ronald de Haan

Algorithms & Complexity Group
Vienna University of Technology

What can complexity theory tell us about judgment aggregation?

It helps us make choices.

What do I mean with judgment aggregation?

What do I mean with judgment aggregation?

Judgment aggregation:

> *The formal and mathematical study of the process of combining the opinions of a group of individuals – on a set of logically related issues – into a combined group opinion.*

In this talk, we will see:

- two formal frameworks
- a few examples of aggregation procedures

What is complexity theory?

What is complexity theory?

Complexity theory (in a nutshell):

> *The mathematical study of what amount of resources (e.g., time) are needed to solve computational problems.*

Computational problems:

- Decision problems (input string, yes-no answer)
- Search problems (input string, output string)

Time:

- Measured as number of steps taken by a computer

Complexity theory

Time measured in terms of the input size (n)

Multiplicative constants are left-out

- $O(f(n))$ is written for $c \cdot f(n)$, where c is a constant

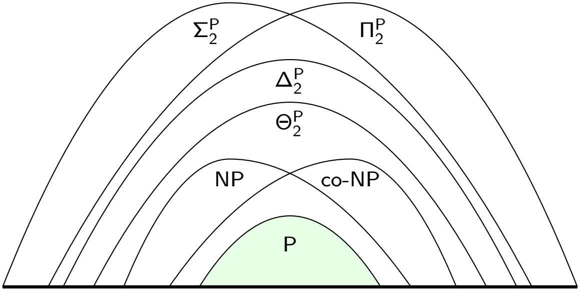
Worst-case analysis: count the maximum amount of time needed to solve any input of length n

Examples:

- 2SAT
  - input: a propositional formula φ in 2CNF
  - question: is φ satisfiable?
  - solvable in time $O(n)$
- SAT
  - input: a propositional formula φ
  - question: is φ satisfiable?
  - *apparently* needs time $\sim 2^n$

Complexity classes

Group problems into different classes:



Tractable:

- polynomial-time solvable problems (P)

Intractable:

- NP, co-NP, etc.
- (believed not polynomial-time solvable; but not proven!)

Complexity Theory

Indication of the difference between polynomial and exponential (for 10.000 steps per second):

n	n^2 time	2^n time
2	0.02 msec	0.02 msec
5	0.15 msec	0.19 msec
10	0.01 sec	0.10 sec
20	0.04 sec	1.75 min
50	0.25 sec	8.4 centuries
100	1.00 sec	9.4×10^{17} years
1000	1.67 min	7.9×10^{288} years

Complexity theory as an algorithmic guide

Use complexity results to determine how to solve a problem:

- P: direct algorithm works well in general
- intractable: not efficiently solvable in all cases

- NP, co-NP: encoding into SAT, use SAT solver
- Θ_2^P: encoding & MaxSAT solver
- Δ_2^P: iterative SAT solving
- Σ_2^P, Π_2^P: encoding & ASP solver / QBF solver

Complexity as a selection criterion for aggregation procedures

Formula-based judgment aggregation framework

- agenda: set Φ of propositional formulas $\varphi_1, \dots, \varphi_m$ and their negations $\neg\varphi_1, \dots, \neg\varphi_m$
- n individuals
- judgment set: subset J of the agenda Φ
  - *consistent* if there exists an assignment that satisfies all $\varphi \in J$
  - *complete* if for each φ_i, either $\varphi_i \in J$ or $\neg\varphi_i \in J$
  - all complete and consistent judgment sets for Φ: $\mathcal{J}(\Phi)$
- profile: a sequence $\mathbf{J} = (J_1, \dots, J_n) \in \mathcal{J}(\Phi)^n$ of n complete and consistent judgment sets for Φ
- judgment aggregation procedure: a function $F : \mathcal{J}(\Phi)^n \to 2^\Phi$
  - *consistent* if all $J \in F(\mathbf{J})$ are consistent for each $\Phi, \mathbf{J}$
  - *complete* if all $J \in F(\mathbf{J})$ are complete for each $\Phi, \mathbf{J}$

Formula-based JA framework (examples)

Agenda: $\{p, q, p \wedge q, \quad \neg p, \neg q, \neg(p \wedge q)\}$

Profile:

	p	q	$p \wedge q$
individual 1	1	0	0
individual 2	0	1	0
individual 3	1	1	1
majority	1	1	0

Majority rule:

- take the (possibly inconsistent) majority opinion
  - (1,1,0)

Slater's rule:

- take complete, consistent judgment sets that are *closest* to the majority opinion
  - ~~(0,0,0)~~ (0,1,0) (1,0,0) (1,1,1)

The winner determination problem

Winner determination (for procedure F):

- input: an agenda Φ, a profile $\mathbf{J}$, and a formula $\varphi \in \Phi$.
- question: is there some outcome $J \in F(\mathbf{J})$ with $\varphi \in J$?

Some complexity results (see [6, 11]):

- majority: in P
- quota: in P
- *premise-based*: in P
- Kemeny: Θ_2^P-complete
- Slater: Θ_2^P-complete
- Young: Θ_2^P-complete
- Tideman (ranked-agenda): Δ_2^P-/Σ_2^P-complete
- Duddy-Piggins: Θ_3^P-complete

The winner determination problem

Winner determination (for procedure F):

- input: an agenda Φ, a profile $\mathbf{J}$, and an integrity constraint Γ.
- output: some outcome $J \in F(\mathbf{J})$ that satisfies Γ

Some complexity results (see [6, 11]):

- majority: in FP
- quota: in FP
- *premise-based*: in FP
- Kemeny: $\mathrm{F}\Theta_2^{\mathrm{P}}$-complete
- Slater: $\mathrm{F}\Theta_2^{\mathrm{P}}$-complete
- Young: $\mathrm{F}\Theta_2^{\mathrm{P}}$-complete
- Tideman (ranked-agenda): $\mathrm{F}\Delta_2^{\mathrm{P}}$-/$\mathrm{F}\Sigma_2^{\mathrm{P}}$-complete
- Duddy-Piggins: $\mathrm{F}\Theta_3^{\mathrm{P}}$-complete

Complexity as a selection criterion for judgment aggregation frameworks

Constraint-based judgment aggregation framework

- agenda: set of propositional variables $X = \{x_1, \dots, x_m\}$ and an integrity constraint Γ in the form of a propositional formula over X
- n individuals
- judgments: truth assignments α to X that satisfy Γ
  - $\mathcal{J}(X, \Gamma)$: set of all judgments for X, Γ
- profile: a sequence $\mathbf{J} = (\alpha_1, \dots, \alpha_n) \in \mathcal{J}(X, \Gamma)^n$ of judgments
- judgment aggregation procedure: a function $F : \mathcal{J}(X, \Gamma)^n \to 2^{2^X}$
  - *consistent* if all $\alpha \in F(\mathbf{J})$ satisfy Γ, for each $X, \Gamma, \mathbf{J}$

Constraint-based JA framework (examples)

Agenda: $X = \{x_1, x_2, x_3\}$, $\Gamma = (x_1 \wedge x_2) \leftrightarrow x_3$

Profile:

	x_1	x_2	x_3
individual 1	1	0	0
individual 2	0	1	0
individual 3	1	1	1
majority	1	1	0

Majority rule:

- take the majority opinion (possibly inconsistent with Γ)
  - (1,1,0)

Slater's rule:

- take judgments (consistent with Γ) that are closest to the majority opinion
  - ~~(0,0,0)~~ (0,1,0) (1,0,0) (1,1,1)

How do the frameworks compare?

Burden on the individuals: choosing a consistent judgment

- formula-based: in FP
- constraint-based: FNP-complete

Succinctness (see [7]):

- for each constraint-based agenda (X, Γ) there is a "small" (poly-size) formula-based agenda Φ that is equivalent
  - but finding it is FNP-complete
- vice versa, not for each formula-based agenda Φ there is a "small" equivalent constraint-based agenda Φ
  - (under some complexity-theoretic assumptions)

Complexity of winner determination might differ:

- (for many rules it is the same)

How do the frameworks compare? (cont'd)

The two frameworks have different complexity properties:

- choosing a (consistent) judgment is easier in the formula-based framework
- agendas can be more succinct in the formula-based framework
- transforming agendas from the constraint-based to the formula-based framework has high complexity
- the complexity of winner determination for aggregation procedures can be different in different frameworks

Complexity as a selection criterion for aggregation procedures

The other side of the coin

High complexity as a good property

Computational problems related to cheating.

- Manipulation: can an individual report a dishonest judgment to improve the outcome?
- Control: can individuals be added/deleted/bundled to improve the outcome?
- Bribery: can the judgment of few individuals be changed to improve the outcome?

High computational complexity for these problems is an advantage for aggregation procedures. (See, e.g., [9])

For the *premise-based procedure*, these forms of cheating are NP-hard. [1, 2, 8]

High complexity as a good property (but beware!)

Beware!

- High worst-case complexity does not mean that cheating is impossible.
- Just not easy in all cases.

More refined complexity analysis is needed to improve the evidence that aggregation procedures are resistant to cheating.

- (More about this in a second.)

What future results should we look forward to?

Research direction one

Answer the complexity questions that are in front of us:

- determine the complexity of the winner determination problem of the different judgment aggregation procedures (in different frameworks)
- determine the complexity of 'cheating problems'

These results will give a **more complete picture** of the consequences of various choices (in terms of complexity).

Research direction two: parameterized complexity

Worst-case complexity analysis has its drawbacks

- maybe there are only a few (untypical) inputs that cause the high complexity

Parameterized complexity [3, 4, 5, 10, 12] is one way to refine this 'classical' analysis

- Measure complexity in terms of input size n and a parameter k
- The parameter captures structure in the input
  (smaller value $\rightsquigarrow$ more structure)
- Examples of parameters for JA:
  - # individuals
  - # issues in the agenda
  - # size of formulas in the agenda (formula-based)
  - degree of variables
  - treewidth

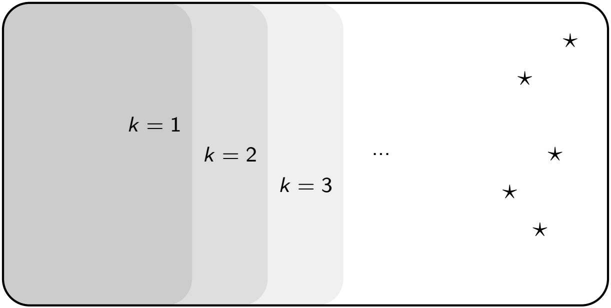
Research direction two: parameterized complexity

Inputs of size n:

$\star$ hard inputs

Research direction two: parameterized complexity

Inputs of size n: $\star$ hard inputs



Research direction two: parameterized complexity

Research direction: answer the various complexity questions using parameterized complexity analysis.

(People are already doing this. See, e.g., [2])

These results will give a **more detailed picture** of the consequences of various choices (in terms of complexity).

What can complexity theory tell us about judgment aggregation?

It gives us another collection of properties to distinguish aggregation frameworks and procedures.

References I

[1] Dorothea Baumeister, Gábor Erdélyi, Olivia Johanna Erdélyi, and Jörg Rothe.
Computational aspects of manipulation and control in judgment aggregation.
In *Algorithmic Decision Theory*, volume 8176 of *Lecture Notes in Computer Science*, pages 71–85. Springer Verlag, 2013.

[2] Dorothea Baumeister, Gábor Erdélyi, and Jörg Rothe.
How hard is it to bribe the judges? a study of the complexity of bribery in judgment aggregation.
In *Algorithmic Decision Theory*, volume 8176 of *Lecture Notes in Computer Science*, pages 1–15. Springer Verlag, 2011.

[3] Hans L. Bodlaender, Rod Downey, Fedor V. Fomin, and Dániel Marx, editors.
The Multivariate Algorithmic Revolution and Beyond – Essays Dedicated to Michael R. Fellows on the Occasion of His 60th Birthday, volume 7370 of *Lecture Notes in Computer Science*. Springer Verlag, 2012.

[4] Rodney G. Downey and Michael R. Fellows.
Parameterized Complexity.
Monographs in Computer Science. Springer Verlag, New York, 1999.

[5] Rodney G. Downey and Michael R. Fellows.
Fundamentals of Parameterized Complexity.
Texts in Computer Science. Springer Verlag, 2013.

References II

[6] Ulle Endriss and Ronald de Haan.
Complexity of the winner determination problem in judgment aggregation: Kemeny, Slater, Tideman, Young.
In *Proceedings of AAMAS 2015, the 14th International Conference on Autonomous Agents and Multiagent Systems*. IFAAMAS/ACM, 2015.

[7] Ulle Endriss, Umberto Grandi, Ronald de Haan, and Jérôme Lang.
Succinctness of languages for judgment aggregation.
Unpublished manuscript, 2015.

[8] Ulle Endriss, Umberto Grandi, and Daniele Porello.
Complexity of judgment aggregation.
J. Artif. Intell. Res., 45:481–514, 2012.

[9] Piotr Faliszewski, Edith Hemaspaandra, and Lane A Hemaspaandra.
Using complexity to protect elections.
Communications of the ACM, 53(11):74–82, 2010.

[10] Jörg Flum and Martin Grohe.
Parameterized Complexity Theory, volume XIV of *Texts in Theoretical Computer Science. An EATCS Series*.
Springer Verlag, Berlin, 2006.

References III

[11] Jérôme Lang and Marija Slavkovik.
How hard is it to compute majority-preserving judgment aggregation rules?
In *21st European Conference on Artificial Intelligence (ECAI 2014)*. IOS Press, 2014.

[12] Rolf Niedermeier.
Invitation to Fixed-Parameter Algorithms.
Oxford Lecture Series in Mathematics and its Applications. Oxford University Press, Oxford, 2006.