Negligence, Strict Liability, and Complexity

Joshua C. Teitelbaum
Georgetown University

Law and Ambiguity Workshop
University of Exeter
21 April 2017

Introduction

- Strict liability versus negligence.
- Equivalence result in the basic accident model of Shavell.
  - negligence "puzzle"
- Literature explores many departures from the basic model.
- I and several others have explored ambiguity.
- David, Surajeet, and I are exploring unawareness.
- In this paper I will explore **complexity**.

The Model

- Unilateral care model with fixed activity level.
- Two risk neutral agents: injurer and victim.
- Injurer engages in a risky activity.
- In the event of an accident the victim incurs a loss.
- Only the injurer can take precautions against an accident.

The Model

Standard Continuous/Divisible Care Model

- Continuous set of precautions, $\mathbb{R}_+$.
- Injurer chooses level of care $x \in \mathbb{R}_+$ having cost $c(x)$.
  - $c : \mathbb{R}_+ \to \mathbb{R}_+$ is differentiable, increasing, and either linear or convex.
- Expected accident loss is $l(x)$.
  - $l : \mathbb{R}_+ \to \mathbb{R}_+$ is differentiable, decreasing, and convex.

The Model

Standard Continuous/Divisible Care Model

- The **social goal** is to

$$\underset{x \in \mathbb{R}_+}{\text{minimize}} \quad c(x) + l(x).$$

The solution x^* is given implicitly by $c'(x^*) = -l'(x^*)$.

- Under **strict liability**, the injurer's problem is

$$\underset{x \in \mathbb{R}_+}{\text{minimize}} \quad c(x) + l(x),$$

which is the social goal. Hence, the injurer takes socially optimal care x^*.

- Under **negligence**, the injurer's problem is

$$\underset{x \in \mathbb{R}_+}{\text{minimize}} \quad \begin{cases} c(x) & \text{if } x \geq \overline{x} \\ c(x) + l(x) & \text{otherwise} \end{cases},$$

where $\overline{x}$ is the due care standard set by the court. If $\overline{x} = x^*$, then the injurer takes socially optimal care.

The Model

Discrete/Indivisible Care Model

- Discrete set of precautions, N.
  - $|N| = n$.
- Injurer chooses subset $X \subseteq N$ having cost $c(X)$.
  - $c : 2^N \rightarrow \mathbb{R}_+$ is monotone increasing.
  - monotone increasing: $c(A) \leq c(B)$ for all $A \subseteq B \subseteq N$.
- Assume c is either modular or supermodular.
  - modular: $c(A) = \sum_{i \in A} c_i$ for all $A \subseteq N$.
  - supermodular: $c(B \cup \{i\}) - c(B) \geq c(A \cup \{i\}) - c(A)$ for all $A \subseteq B \subseteq N$ and $i \in N \backslash B$.
- Expected accident loss is $l(X)$.
  - $l : 2^N \rightarrow \mathbb{R}_+$ is monotone decreasing.
- Assume initially that l is submodular.
  - submodular: $l(B \cup \{i\}) - l(B) \leq l(A \cup \{i\}) - l(A)$ for all $A \subseteq B \subseteq N$ and $i \in N \backslash B$.

The Model

Discrete/Indivisible Care Model

- The **social goal** is to

$$\underset{X \subseteq N}{\text{minimize}} \quad c(X) + l(X). \tag{Social Goal}$$

  Let X^* denote the solution to Social Goal.

- Under **strict liability**, the injurer's problem is

$$\underset{X \subseteq N}{\text{minimize}} \quad c(X) + l(X). \tag{Strict Liability}$$

  which is identical to Social Goal.

- Under **negligence**, the injurer's problem is

$$\underset{X \subseteq N}{\text{minimize}} \quad \begin{cases} c(X) & \text{if } X \supseteq \overline{X} \\ c(X) + l(X) & \text{otherwise} \end{cases} . \tag{Negligence}$$

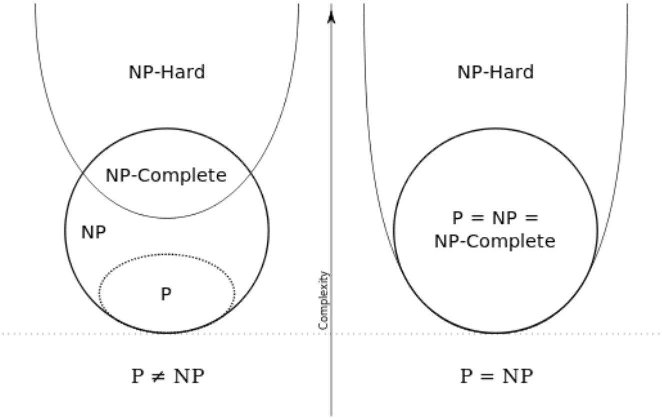
  where $\overline{X} \subseteq N$ is the due care standard set by the court.

Complexity

Four important computational complexity classes:

- $\mathcal{P}$ = problems that can be solved computationally in polynomial time.
- $\mathcal{NP}$ = problems for which a given solution can be verified in polynomial time.
  - $\mathcal{P} = \mathcal{NP}$? is a Millennium Prize Problem. It is conjectured that $\mathcal{P} \neq \mathcal{NP}$.
- $\mathcal{NP}$-hard = problems with no known polynomial-time solutions.
- $\mathcal{NP}$-complete = problems in $\mathcal{NP}$ that are $\mathcal{NP}$-hard.

Complexity


NP-Hard
NP-Complete
NP
P
Complexity
P ≠ NP
NP-Hard
P = NP =
NP-Complete
P = NP

Results

Theorem

If c is modular, then Strict Liability *and* Negligence *are computable in polynomial time.*

- Idea of proof:
- If c is modular then $f(X) = c(X) + l(X)$ is submodular.
- Observe that $f : 2^N \to \mathbb{R}_+ \Leftrightarrow f : \{0,1\}^n \to \mathbb{R}_+$.
- Construct Lovász extension $f^L : [0,1]^n \to \mathbb{R}_+$.
- Invoke Lovász theorem: f^L is convex and $\min_{X \subseteq N} f(X) = \min_{x \in [0,1]^n} f(x)$.
- Construct X^* from x^*.

Results

Theorem

If c is supermodular, then STRICT LIABILITY *and* NEGLIGENCE *are $\mathcal{NP}$-hard.*

- Proof is by reduction from MAX CUT, a well-known $\mathcal{NP}$-hard problem.
- MAX CUT is the problem of finding a subset of the vertex set of a graph that maximizes the number of edges between the subset and the complementary subset.

Results

- Many other reasonable variations result in computational intractability.
- For instance:
  - l is supermodular (reduction from MAX CUT).
  - cardinality constraint: $|X| \geq k$ (reduction from MINKP).
  - "knapsack" constraint: $c(X) \geq m$ (reduction from MINKP).

Results

- Upshot is that neither strict liability nor negligence can achieve socially optimal deterrence, if the precaution set is too large (i.e., if n is too large).
- For strict liability to achieve socially optimal deterrence, the injurer must solve STRICT LIABILITY.
- For negligence to achieve socially optimal deterrence:
  - The court must set $\overline{X} = X^*$, which requires solving STRICT LIABILITY.
  - The injurer must solve NEGLIGENCE.
- But in many cases neither the injurer nor the court, even if armed with a non-deterministic Turing machine (i.e., a computer that can perform an unbounded number of parallel computations), can solve STRICT LIABILITY or NEGLIGENCE.

Results

Theorem

If c and l are computable in polynomial time, then the decision problem forms of Strict Liability *and* Negligence *are $\mathcal{NP}$-complete.*

- Idea of proof:
- Transform each problem into a decision problem (i.e., a yes-no problem).
- For example, Strict Liability becomes: Given a threshold value t,

$$\begin{array}{ll} \text{there exists} & X \subseteq N \\ \text{such that} & c(X) + l(X) \leq t. \end{array}$$

- Show that, given any threshold value t and any proposed solution $\widehat{X} \subseteq N$, a polynomial-time algorithm exists that verifies whether $\widehat{X}$ is a solution.
- Algorithm is trivial: It simply checks whether $c(\widehat{X}) + l(\widehat{X}) \leq t$. If c and p are computable in polynomial time, then the algorithm runs in polynomial time.

Results

- Upshot is that negligence can move the law in the direction of optimal deterrence far more rapidly than strict liability.
- Imagine that the circumstances of an accident suggest to the parties a subset of precautions $\widehat{X}$ that would have prevented the accident.
- Suppose we have a negligence regime where the due care standard is $\overline{X}$.
  - For instance, suppose the court can compute $\overline{X}$ using a polynomial-time approximation algorithm for STRICT LIABILITY.
- Because STRICT LIABILITY is $\mathcal{NP}$-complete, the court can check whether $c(\widehat{X}) + l(\widehat{X}) \leq c(\overline{X}) + l(\overline{X})$.
- If yes, the court resets the due care standard to $\widehat{X}$. **This leads all injurers to move in the direction of optimal care.**
- If we have a strict liability regime, the injurer in question learns that $\widehat{X}$ is superior to $\overline{X}$. **But this knowledge is not disseminated to all injurers.**

Results

- In some cases, it may be possible to approximate solutions to STRICT LIABILITY and NEGLIGENCE in polynomial time.
  - No known approximation for the cases where c or l is supermodular.
- But the approximations often have poor performance guarantees.
  - For the case of a cardinality constraint, it appears there may be a $\sqrt{n}$-approximation.
  - It is unclear to me at this point if there is an approximation for the case of a knapsack constraint.

Conclusion

- Paper is a work in progress!
- It will develop the ideas outlined above.
- I welcome your comments, ideas, and suggestions!