MODULAR CONTRACTS

Jason Roderick Donaldson   Giorgia Piacentino   Zichen Zhao

FACTS

Current theory: Contract as incentive/sharing rule: outcome $\mapsto$ transfers

FACTS

Current theory: Contract as incentive/sharing rule: outcome $\mapsto$ transfers

Real contracts

FACTS

Current theory: Contract as incentive/sharing rule: outcome $\mapsto$ transfers

Real contracts: Written in language and divided into modular clauses

FACTS

Current theory: Contract as incentive/sharing rule: outcome $\mapsto$ transfers

Real contracts: Written in language and divided into modular clauses

Evaluated clause by clause

FACTS

Current theory: Contract as incentive/sharing rule: outcome $\mapsto$ transfers

Real contracts: Written in language and divided into modular clauses

Evaluated clause by clause $\Rightarrow$ phenomena not in standard model

FACTS

Current theory: Contract as incentive/sharing rule: outcome $\mapsto$ transfers

Real contracts: Written in language and divided into modular clauses

Evaluated clause by clause $\Rightarrow$ phenomena not in standard model:

Loopholes

FACTS

Current theory: Contract as incentive/sharing rule: outcome $\mapsto$ transfers

Real contracts: Written in language and divided into modular clauses

Evaluated clause by clause $\Rightarrow$ phenomena not in standard model:

Loopholes: Outcomes can depend on how clauses combined

FACTS

Current theory: Contract as incentive/sharing rule: outcome $\mapsto$ transfers

Real contracts: Written in language and divided into modular clauses

Evaluated clause by clause $\Rightarrow$ phenomena not in standard model:

- Loopholes: Outcomes can depend on how clauses combined
- Errors due to complexity

FACTS

Current theory: Contract as incentive/sharing rule: outcome $\mapsto$ transfers

Real contracts: Written in language and divided into modular clauses

Evaluated clause by clause $\Rightarrow$ phenomena not in standard model:

- $\underline{\text{Loopholes}}$: Outcomes can depend on how clauses combined
- $\underline{\text{Errors due to complexity}}$: Mistakes parsing long clauses

FACTS

Current theory: Contract as incentive/sharing rule: outcome $\mapsto$ transfers

Real contracts: Written in language and divided into modular clauses

Evaluated clause by clause $\Rightarrow$ phenomena not in standard model:

- Loopholes: Outcomes can depend on how clauses combined
- Errors due to complexity: Mistakes parsing long clauses
- Landmines

FACTS

Current theory: Contract as incentive/sharing rule: outcome $\mapsto$ transfers

Real contracts: Written in language and divided into modular clauses

Evaluated clause by clause $\Rightarrow$ phenomena not in standard model:

- Loopholes: Outcomes can depend on how clauses combined
- Errors due to complexity: Mistakes parsing long clauses
- Landmines: Redundant clauses contradict one another

FACTS

Current theory: Contract as incentive/sharing rule: outcome $\mapsto$ transfers

Real contracts: Written in language and divided into modular clauses

Evaluated clause by clause $\Rightarrow$ phenomena not in standard model:

- Loopholes: Outcomes can depend on how clauses combined
- Errors due to complexity: Mistakes parsing long clauses
- Landmines: Redundant clauses contradict one another
- Incompleteness

FACTS

Current theory: Contract as incentive/sharing rule: outcome $\mapsto$ transfers

Real contracts: Written in language and divided into modular clauses

Evaluated clause by clause $\Rightarrow$ phenomena not in standard model:

- Loopholes: Outcomes can depend on how clauses combined
- Errors due to complexity: Mistakes parsing long clauses
- Landmines: Redundant clauses contradict one another
- Incompleteness: Clauses omit relevant contingencies

FACTS

Current theory: Contract as incentive/sharing rule: outcome $\mapsto$ transfers

Real contracts: Written in language and divided into modular clauses

Evaluated clause by clause $\Rightarrow$ phenomena not in standard model:

Loopholes: Outcomes can depend on how clauses combined

Errors due to complexity: Mistakes parsing long clauses

Landmines: Redundant clauses contradict one another

Incompleteness: Clauses omit relevant contingencies

Conservatism

FACTS

Current theory: Contract as incentive/sharing rule: outcome $\mapsto$ transfers

Real contracts: Written in language and divided into modular clauses

Evaluated clause by clause $\Rightarrow$ phenomena not in standard model:

<u>Loopholes</u>: Outcomes can depend on how clauses combined

<u>Errors due to complexity</u>: Mistakes parsing long clauses

<u>Landmines</u>: Redundant clauses contradict one another

<u>Incompleteness</u>: Clauses omit relevant contingencies

<u>Conservatism</u>: High burden of truth to verify clause

QUESTIONS

Why are contracts modular, with clauses evaluated separately?

What explains loopholes, complexity, landmines, incompleteness,...?

THIS PAPER

Model of modular contracts

THIS PAPER

Model of modular contracts: Modules = clauses

THIS PAPER

Model of modular contracts: Modules = clauses

Clauses collections of atomic premises

THIS PAPER

Model of modular contracts: Modules = clauses

- Clauses collections of atomic premises
- Premises evaluated (1 or 0) based on threshold of observed data

THIS PAPER

Model of modular contracts: Modules = clauses

  Clauses collections of atomic premises

  Premises evaluated (1 or 0) based on threshold of observed data

Key assumption

THIS PAPER

Model of modular contracts: Modules = clauses

- Clauses collections of atomic premises
- Premises evaluated (1 or 0) based on threshold of observed data

Key assumption: Shuffled data

THIS PAPER

Model of modular contracts: Modules = clauses

- Clauses collections of atomic premises

- Premises evaluated (1 or 0) based on threshold of observed data

Key assumption: Shuffled data: Loss of labeling, alignment, or ordering

THIS PAPER

Model of modular contracts: Modules = clauses

Clauses collections of atomic premises

Premises evaluated (1 or 0) based on threshold of observed data

Key assumption: Shuffled data: Loss of labeling, alignment, or ordering

For each premise, observations permutations of true data

THIS PAPER

Model of modular contracts: Modules = clauses

  Clauses collections of atomic premises

  Premises evaluated (1 or 0) based on threshold of observed data

Key assumption: Shuffled data: Loss of labeling, alignment, or ordering

  For each premise, observations permutations of true data

E.g. time jitters, anonymized identity, fuzzy merge, binding failure (psych),...

THIS PAPER

Model of modular contracts: Modules = clauses

Clauses collections of atomic premises

Premises evaluated (1 or 0) based on threshold of observed data

Key assumption: Shuffled data: Loss of labeling, alignment, or ordering

For each premise, observations permutations of true data

E.g. time jitters, anonymized identity, fuzzy merge, binding failure (psych),...

E.g. true data: $\mathbf{a}_j = 110, \mathbf{a}_k = 101$; observation: $\hat{\mathbf{a}}_j = 110$, $\hat{\mathbf{a}}_k = 110$

THIS PAPER

Model of modular contracts: Modules = clauses

Clauses collections of atomic premises

Premises evaluated (1 or 0) based on threshold of observed data

Key assumption: Shuffled data: Loss of labeling, alignment, or ordering

For each premise, observations permutations of true data

E.g. time jitters, anonymized identity, fuzzy merge, binding failure (psych),...

E.g. true data: $\mathbf{a}_j = 110, \mathbf{a}_k = 101$; observation: $\hat{\mathbf{a}}_j = 110$, $\hat{\mathbf{a}}_k = 110$

Marginal unchanged $\implies$ evaluate $\mathbf{a}_j, \mathbf{a}_k$ correctly

THIS PAPER

Model of modular contracts: Modules = clauses

Clauses collections of atomic premises

Premises evaluated (1 or 0) based on threshold of observed data

Key assumption: Shuffled data: Loss of labeling, alignment, or ordering

For each premise, observations permutations of true data

E.g. time jitters, anonymized identity, fuzzy merge, binding failure (psych),...

E.g. true data: $\mathbf{a}_j = 110, \mathbf{a}_k = 101$; observation: $\hat{\mathbf{a}}_j = 110$, $\hat{\mathbf{a}}_k = 110$

Marginal unchanged $\implies$ evaluate $\mathbf{a}_j, \mathbf{a}_k$ correctly, not joint $\mathbf{a}_j \wedge \mathbf{a}_k$

RESULTS

RESULTS

Narrow clauses

RESULTS

Narrow clauses: Evaluate $\mathbf{a}_j, \mathbf{a}_k$ one by one to avoid shuffling error

RESULTS

Narrow clauses: Evaluate $\mathbf{a}_j, \mathbf{a}_k$ one by one to avoid shuffling error

Loopholes: Order of evaluation can change outcome

RESULTS

Narrow clauses: Evaluate $\mathbf{a}_j, \mathbf{a}_k$ one by one to avoid shuffling error

Loopholes: Order of evaluation can change outcome

Wide clauses

RESULTS

Narrow clauses: Evaluate $\mathbf{a}_j, \mathbf{a}_k$ one by one to avoid shuffling error

Loopholes: Order of evaluation can change outcome

Wide clauses: Keep track of dependencies among premises

RESULTS

Narrow clauses: Evaluate $\mathbf{a}_j, \mathbf{a}_k$ one by one to avoid shuffling error

Loopholes: Order of evaluation can change outcome

Wide clauses: Keep track of dependencies among premises

Errors due to complexity: Evaluation errors due to shuffled data

RESULTS

Narrow clauses: Evaluate $\mathbf{a}_j, \mathbf{a}_k$ one by one to avoid shuffling error

Loopholes: Order of evaluation can change outcome

Wide clauses: Keep track of dependencies among premises

Errors due to complexity: Evaluation errors due to shuffled data

Redundant clauses

RESULTS

Narrow clauses: Evaluate $\mathbf{a}_j, \mathbf{a}_k$ one by one to avoid shuffling error

Loopholes: Order of evaluation can change outcome

Wide clauses: Keep track of dependencies among premises

Errors due to complexity: Evaluation errors due to shuffled data

Redundant clauses: Repeat same premise $\mathbf{a}_j$ in multiple clauses

RESULTS

Narrow clauses: Evaluate $\mathbf{a}_j, \mathbf{a}_k$ one by one to avoid shuffling error

Loopholes: Order of evaluation can change outcome

Wide clauses: Keep track of dependencies among premises

Errors due to complexity: Evaluation errors due to shuffled data

Redundant clauses: Repeat same premise $\mathbf{a}_j$ in multiple clauses

Landmines: Internal inconsistencies if premise shuffled in one clause

RESULTS

Narrow clauses: Evaluate $\mathbf{a}_j, \mathbf{a}_k$ one by one to avoid shuffling error

Loopholes: Order of evaluation can change outcome

Wide clauses: Keep track of dependencies among premises

Errors due to complexity: Evaluation errors due to shuffled data

Redundant clauses: Repeat same premise $\mathbf{a}_j$ in multiple clauses

Landmines: Internal inconsistencies if premise shuffled in one clause

Omitted premises

RESULTS

Narrow clauses: Evaluate $\mathbf{a}_j, \mathbf{a}_k$ one by one to avoid shuffling error

Loopholes: Order of evaluation can change outcome

Wide clauses: Keep track of dependencies among premises

Errors due to complexity: Evaluation errors due to shuffled data

Redundant clauses: Repeat same premise $\mathbf{a}_j$ in multiple clauses

Landmines: Internal inconsistencies if premise shuffled in one clause

Omitted premises: Might omit relevant premises entirely

RESULTS

Narrow clauses: Evaluate $\mathbf{a}_j, \mathbf{a}_k$ one by one to avoid shuffling error

- Loopholes: Order of evaluation can change outcome

Wide clauses: Keep track of dependencies among premises

- Errors due to complexity: Evaluation errors due to shuffled data

Redundant clauses: Repeat same premise $\mathbf{a}_j$ in multiple clauses

- Landmines: Internal inconsistencies if premise shuffled in one clause

Omitted premises: Might omit relevant premises entirely

- Incompleteness: Cost of omission less than risk of shuffling

RESULTS

- Narrow clauses: Evaluate $\mathbf{a}_j, \mathbf{a}_k$ one by one to avoid shuffling error
  - Loopholes: Order of evaluation can change outcome
- Wide clauses: Keep track of dependencies among premises
  - Errors due to complexity: Evaluation errors due to shuffled data
- Redundant clauses: Repeat same premise $\mathbf{a}_j$ in multiple clauses
  - Landmines: Internal inconsistencies if premise shuffled in one clause
- Omitted premises: Might omit relevant premises entirely
  - Incompleteness: Cost of omission less than risk of shuffling
- Extension: High thresholds

RESULTS

Narrow clauses: Evaluate $\mathbf{a}_j, \mathbf{a}_k$ one by one to avoid shuffling error

Loopholes: Order of evaluation can change outcome

Wide clauses: Keep track of dependencies among premises

Errors due to complexity: Evaluation errors due to shuffled data

Redundant clauses: Repeat same premise $\mathbf{a}_j$ in multiple clauses

Landmines: Internal inconsistencies if premise shuffled in one clause

Omitted premises: Might omit relevant premises entirely

Incompleteness: Cost of omission less than risk of shuffling

Extension: High thresholds $\implies$ conservatism

MODEL

MODEL

MODEL

(Atomic) Premises

MODEL

(Atomic) Premises: Random vectors $\{\mathbf{a}_j\}_{j\in\mathcal{A}}$

MODEL

(Atomic) Premises: Random vectors $\{\mathbf{a}_j\}_{j\in\mathcal{A}}$

Data

MODEL

(Atomic) Premises: Random vectors $\{\mathbf{a}_j\}_{j \in \mathcal{A}}$

Data: $\{a_j^i\}_{i=1}^I \in \{0,1\}^I$ are realizations of $\mathbf{a}_j$

MODEL

(Atomic) Premises: Random vectors $\{\mathbf{a}_j\}_{j\in\mathcal{A}}$

Data: $\{a_j^i\}_{i=1}^I \in \{0,1\}^I$ are realizations of $\mathbf{a}_j$

Target policy

MODEL

(Atomic) Premises: Random vectors $\{\mathbf{a}_j\}_{j\in\mathcal{A}}$

Data: $\{a_j^i\}_{i=1}^I \in \{0,1\}^I$ are realizations of $\mathbf{a}_j$

Target policy: Conjunction of premises: For $\mathcal{T} \subset \mathcal{A}$, $\phi_{\mathcal{T}} = \bigwedge_{j\in\mathcal{T}} \mathbf{a}_j$

MODEL

(Atomic) Premises: Random vectors $\{\mathbf{a}_j\}_{j\in\mathcal{A}}$

Data: $\{a_j^i\}_{i=1}^I \in \{0,1\}^I$ are realizations of $\mathbf{a}_j$

Target policy: Conjunction of premises: For $\mathcal{T} \subset \mathcal{A}$, $\phi_{\mathcal{T}} = \bigwedge_{j\in\mathcal{T}} \mathbf{a}_j$

Observations = shuffled data

MODEL

(Atomic) Premises: Random vectors $\{\mathbf{a}_j\}_{j\in\mathcal{A}}$

Data: $\{a_j^i\}_{i=1}^I \in \{0,1\}^I$ are realizations of $\mathbf{a}_j$

Target policy: Conjunction of premises: For $\mathcal{T} \subset \mathcal{A}$, $\phi_{\mathcal{T}} = \bigwedge_{j\in\mathcal{T}} \mathbf{a}_j$

Observations = shuffled data: $\hat{\mathbf{a}}_j = \mathbf{a}_j$ w.p. $1 - e$

MODEL

(Atomic) Premises: Random vectors $\{\mathbf{a}_j\}_{j\in\mathcal{A}}$

Data: $\{a_j^i\}_{i=1}^I \in \{0,1\}^I$ are realizations of $\mathbf{a}_j$

Target policy: Conjunction of premises: For $\mathcal{T} \subset \mathcal{A}$, $\phi_{\mathcal{T}} = \bigwedge_{j\in\mathcal{T}} \mathbf{a}_j$

Observations = shuffled data: $\hat{\mathbf{a}}_j = \mathbf{a}_j$ w.p. $1-e$; permutation of $\mathbf{a}_j$ w.p. e

MODEL

(Atomic) Premises: Random vectors $\{\mathbf{a}_j\}_{j \in \mathcal{A}}$

Data: $\{a_j^i\}_{i=1}^I \in \{0,1\}^I$ are realizations of $\mathbf{a}_j$

Target policy: Conjunction of premises: For $\mathcal{T} \subset \mathcal{A}$, $\phi_{\mathcal{T}} = \bigwedge_{j \in \mathcal{T}} \mathbf{a}_j$

Observations = shuffled data: $\hat{\mathbf{a}}_j = \mathbf{a}_j$ w.p. $1 - e$; permutation of $\mathbf{a}_j$ w.p. e

Clause

MODEL

(Atomic) Premises: Random vectors $\{\mathbf{a}_j\}_{j \in \mathcal{A}}$

Data: $\{a_j^i\}_{i=1}^I \in \{0,1\}^I$ are realizations of $\mathbf{a}_j$

Target policy: Conjunction of premises: For $\mathcal{T} \subset \mathcal{A}$, $\phi_{\mathcal{T}} = \bigwedge_{j \in \mathcal{T}} \mathbf{a}_j$

Observations = shuffled data: $\hat{\mathbf{a}}_j = \mathbf{a}_j$ w.p. $1-e$; permutation of $\mathbf{a}_j$ w.p. e

Clause: $\mathcal{M}$ defines sentence $\hat{\phi}_{\mathcal{M}} = \bigwedge_{j \in \mathcal{M}} \hat{\mathbf{a}}_j$, componentwise

MODEL

(Atomic) Premises: Random vectors $\{\mathbf{a}_j\}_{j \in \mathcal{A}}$

Data: $\{a_j^i\}_{i=1}^I \in \{0,1\}^I$ are realizations of $\mathbf{a}_j$

Target policy: Conjunction of premises: For $\mathcal{T} \subset \mathcal{A}$, $\phi_{\mathcal{T}} = \bigwedge_{j \in \mathcal{T}} \mathbf{a}_j$

Observations = shuffled data: $\hat{\mathbf{a}}_j = \mathbf{a}_j$ w.p. $1 - e$; permutation of $\mathbf{a}_j$ w.p. e

Clause: $\mathcal{M}$ defines sentence $\hat{\phi}_{\mathcal{M}} = \bigwedge_{j \in \mathcal{M}} \hat{\mathbf{a}}_j$, componentwise

Evaluation

MODEL

(Atomic) Premises: Random vectors $\{\mathbf{a}_j\}_{j \in \mathcal{A}}$

Data: $\{a_j^i\}_{i=1}^I \in \{0,1\}^I$ are realizations of $\mathbf{a}_j$

Target policy: Conjunction of premises: For $\mathcal{T} \subset \mathcal{A}$, $\phi_{\mathcal{T}} = \bigwedge_{j \in \mathcal{T}} \mathbf{a}_j$

Observations = shuffled data: $\hat{\mathbf{a}}_j = \mathbf{a}_j$ w.p. $1-e$; permutation of $\mathbf{a}_j$ w.p. e

Clause: $\mathcal{M}$ defines sentence $\hat{\phi}_{\mathcal{M}} = \bigwedge_{j \in \mathcal{M}} \hat{\mathbf{a}}_j$, componentwise

Evaluation: $[\![\, \hat{\phi} \,]\!] = 1 \;\text{ if }\; \#\{\, i \;:\; \hat{\phi}^i = 1\} \geq \tau$

MODEL

(Atomic) Premises: Random vectors $\{\mathbf{a}_j\}_{j \in \mathcal{A}}$

Data: $\{a_j^i\}_{i=1}^I \in \{0,1\}^I$ are realizations of $\mathbf{a}_j$

Target policy: Conjunction of premises: For $\mathcal{T} \subset \mathcal{A}$, $\phi_{\mathcal{T}} = \bigwedge_{j \in \mathcal{T}} \mathbf{a}_j$

Observations = shuffled data: $\hat{\mathbf{a}}_j = \mathbf{a}_j$ w.p. $1 - e$; permutation of $\mathbf{a}_j$ w.p. e

Clause: $\mathcal{M}$ defines sentence $\hat{\phi}_{\mathcal{M}} = \bigwedge_{j \in \mathcal{M}} \hat{\mathbf{a}}_j$, componentwise

Evaluation: $[\![\, \hat{\phi} \,]\!] = 1$ if $\#\{ i \ : \ \hat{\phi}^i = 1\} \geq \tau$; $[\![\, \hat{\phi} \,]\!] = 0$ o.w.

MODEL

(Atomic) Premises: Random vectors $\{\mathbf{a}_j\}_{j \in \mathcal{A}}$

Data: $\{a_j^i\}_{i=1}^I \in \{0,1\}^I$ are realizations of $\mathbf{a}_j$

Target policy: Conjunction of premises: For $\mathcal{T} \subset \mathcal{A}$, $\phi_{\mathcal{T}} = \bigwedge_{j \in \mathcal{T}} \mathbf{a}_j$

Observations = shuffled data: $\hat{\mathbf{a}}_j = \mathbf{a}_j$ w.p. $1 - e$; permutation of $\mathbf{a}_j$ w.p. e

Clause: $\mathcal{M}$ defines sentence $\hat{\phi}_{\mathcal{M}} = \bigwedge_{j \in \mathcal{M}} \hat{\mathbf{a}}_j$, componentwise

Evaluation: $[\![\, \hat{\phi} \,]\!] = 1$ if $\#\{ i \,:\, \hat{\phi}^i = 1\} \geq \tau$; $[\![\, \hat{\phi} \,]\!] = 0$ o.w.

Contract Φ

MODEL

(Atomic) Premises: Random vectors $\{\mathbf{a}_j\}_{j\in\mathcal{A}}$

Data: $\{a_j^i\}_{i=1}^I \in \{0,1\}^I$ are realizations of $\mathbf{a}_j$

Target policy: Conjunction of premises: For $\mathcal{T} \subset \mathcal{A}$, $\phi_{\mathcal{T}} = \bigwedge_{j\in\mathcal{T}} \mathbf{a}_j$

Observations = shuffled data: $\hat{\mathbf{a}}_j = \mathbf{a}_j$ w.p. $1-e$; permutation of $\mathbf{a}_j$ w.p. e

Clause: $\mathcal{M}$ defines sentence $\hat{\phi}_{\mathcal{M}} = \bigwedge_{j\in\mathcal{M}} \hat{\mathbf{a}}_j$, componentwise

Evaluation: $[\![\, \hat{\phi} \,]\!] = 1$ if $\#\{ i \;:\; \hat{\phi}^i = 1\} \geq \tau$; $[\![\, \hat{\phi} \,]\!] = 0$ o.w.

Contract Φ: Set of clauses $\{\mathcal{M}_m\}_m$ evaluated & conjoined

MODEL

(Atomic) Premises: Random vectors $\{\mathbf{a}_j\}_{j\in\mathcal{A}}$

Data: $\{a_j^i\}_{i=1}^I \in \{0,1\}^I$ are realizations of $\mathbf{a}_j$

Target policy: Conjunction of premises: For $\mathcal{T} \subset \mathcal{A}$, $\phi_{\mathcal{T}} = \bigwedge_{j\in\mathcal{T}} \mathbf{a}_j$

Observations = shuffled data: $\hat{\mathbf{a}}_j = \mathbf{a}_j$ w.p. $1-e$; permutation of $\mathbf{a}_j$ w.p. e

Clause: $\mathcal{M}$ defines sentence $\hat{\phi}_{\mathcal{M}} = \bigwedge_{j\in\mathcal{M}} \hat{\mathbf{a}}_j$, componentwise

Evaluation: $[\![\, \hat{\phi} \,]\!] = 1$ if $\#\{\, i \ : \ \hat{\phi}^i = 1\} \geq \tau$; $[\![\, \hat{\phi} \,]\!] = 0$ o.w.

Contract Φ: Set of clauses $\{\mathcal{M}_m\}_m$ evaluated & conjoined: $[\![\Phi]\!] = \bigwedge_m [\![\hat{\phi}_{\mathcal{M}_m}]\!]$

MODEL

(Atomic) Premises: Random vectors $\{\mathbf{a}_j\}_{j\in\mathcal{A}}$

Data: $\{a_j^i\}_{i=1}^I \in \{0,1\}^I$ are realizations of $\mathbf{a}_j$

Target policy: Conjunction of premises: For $\mathcal{T} \subset \mathcal{A}$, $\phi_{\mathcal{T}} = \bigwedge_{j\in\mathcal{T}} \mathbf{a}_j$

Observations = shuffled data: $\hat{\mathbf{a}}_j = \mathbf{a}_j$ w.p. $1-e$; permutation of $\mathbf{a}_j$ w.p. e

Clause: $\mathcal{M}$ defines sentence $\hat{\phi}_{\mathcal{M}} = \bigwedge_{j\in\mathcal{M}} \hat{\mathbf{a}}_j$, componentwise

Evaluation: $[\![\, \hat{\phi} \,]\!] = 1$ if $\#\{\, i \;:\; \hat{\phi}^i = 1\} \geq \tau$; $[\![\, \hat{\phi} \,]\!] = 0$ o.w.

Contract Φ: Set of clauses $\{\mathcal{M}_m\}_m$ evaluated & conjoined: $[\![\Phi]\!] = \bigwedge_m [\![\hat{\phi}_{\mathcal{M}_m}]\!]$

Objective

MODEL

(Atomic) Premises: Random vectors $\{\mathbf{a}_j\}_{j\in\mathcal{A}}$

Data: $\{a_j^i\}_{i=1}^I \in \{0,1\}^I$ are realizations of $\mathbf{a}_j$

Target policy: Conjunction of premises: For $\mathcal{T} \subset \mathcal{A}$, $\phi_{\mathcal{T}} = \bigwedge_{j\in\mathcal{T}} \mathbf{a}_j$

Observations = shuffled data: $\hat{\mathbf{a}}_j = \mathbf{a}_j$ w.p. $1-e$; permutation of $\mathbf{a}_j$ w.p. e

Clause: $\mathcal{M}$ defines sentence $\hat{\phi}_{\mathcal{M}} = \bigwedge_{j\in\mathcal{M}} \hat{\mathbf{a}}_j$, componentwise

Evaluation: $[\![\, \hat{\phi} \,]\!] = 1 \;\text{ if }\; \#\{\, i \;:\; \hat{\phi}^i = 1\} \geq \tau$; $[\![\, \hat{\phi} \,]\!] = 0$ o.w.

Contract Φ: Set of clauses $\{\mathcal{M}_m\}_m$ evaluated & conjoined: $[\![\Phi]\!] = \bigwedge_m [\![\hat{\phi}_{\mathcal{M}_m}]\!]$

Objective: Minimize $C = c_{\mathrm{I}}\mathsf{Pr}[[\![\Phi]\!] > [\![\phi_{\mathcal{T}}]\!]] + c_{\mathrm{II}}\mathsf{Pr}\,[[\![\Phi]\!] < [\![\phi_{\mathcal{T}}]\!]]$

MODEL: EXAMPLE: SHOE DELIVERY

MODEL: EXAMPLE: SHOE DELIVERY

(Atomic) Premises

MODEL: EXAMPLE: SHOE DELIVERY

(Atomic) Premises: $\mathbf{a}_\ell$: left shoe $\mathbf{a}_r$: right; Data: size $i \in \{1, 2, 3\}$

MODEL: EXAMPLE: SHOE DELIVERY

(Atomic) Premises: $\mathbf{a}_\ell$: left shoe $\mathbf{a}_r$: right; Data: size $i \in \{1, 2, 3\}$

Target policy

MODEL: EXAMPLE: SHOE DELIVERY

(Atomic) Premises: $\mathbf{a}_\ell$: left shoe $\mathbf{a}_r$: right; Data: size $i \in \{1, 2, 3\}$

Target policy: $\phi_{\mathcal{T}} = \mathbf{a}_\ell \wedge \mathbf{a}_r$ (“[deliver] pairs”: $a_\ell^i \wedge a_r^i$)

MODEL: EXAMPLE: SHOE DELIVERY

(Atomic) Premises: $\mathbf{a}_\ell$: left shoe $\mathbf{a}_r$: right; Data: size $i \in \{1, 2, 3\}$

Target policy: $\phi_{\mathcal{T}} = \mathbf{a}_\ell \wedge \mathbf{a}_r$ ("[deliver] pairs": $a_\ell^i \wedge a_r^i$)

Observations = shuffled data

MODEL: EXAMPLE: SHOE DELIVERY

(Atomic) Premises: $\mathbf{a}_\ell$: left shoe $\mathbf{a}_r$: right; Data: size $i \in \{1, 2, 3\}$

Target policy: $\phi_{\mathcal{T}} = \mathbf{a}_\ell \wedge \mathbf{a}_r$ ("[deliver] pairs": $a_\ell^i \wedge a_r^i$)

Observations = shuffled data: $\hat{\mathbf{a}}_j$ sizes shuffled w.p. e

MODEL: EXAMPLE: SHOE DELIVERY

(Atomic) Premises: $\mathbf{a}_\ell$: left shoe $\mathbf{a}_r$: right; Data: size $i \in \{1, 2, 3\}$

Target policy: $\phi_{\mathcal{T}} = \mathbf{a}_\ell \wedge \mathbf{a}_r$ ("[deliver] pairs": $a_\ell^i \wedge a_r^i$)

Observations = shuffled data: $\hat{\mathbf{a}}_j$ sizes shuffled w.p. e

Clause

MODEL: EXAMPLE: SHOE DELIVERY

(Atomic) Premises: $\mathbf{a}_\ell$: left shoe $\mathbf{a}_r$: right; Data: size $i \in \{1, 2, 3\}$

Target policy: $\phi_{\mathcal{T}} = \mathbf{a}_\ell \wedge \mathbf{a}_r$ ("[deliver] pairs": $a_\ell^i \wedge a_r^i$)

Observations = shuffled data: $\hat{\mathbf{a}}_j$ sizes shuffled w.p. e

Clause: "right shoe," "left shoe," "pairs"

MODEL: EXAMPLE: SHOE DELIVERY

(Atomic) Premises: $\mathbf{a}_\ell$: left shoe $\mathbf{a}_r$: right; Data: size $i \in \{1, 2, 3\}$

Target policy: $\phi_{\mathcal{T}} = \mathbf{a}_\ell \wedge \mathbf{a}_r$ ("[deliver] pairs": $a_\ell^i \wedge a_r^i$)

Observations = shuffled data: $\hat{\mathbf{a}}_j$ sizes shuffled w.p. e

Clause: "right shoe," "left shoe," "pairs"

Evaluation

MODEL: EXAMPLE: SHOE DELIVERY

(Atomic) Premises: $\mathbf{a}_\ell$: left shoe $\mathbf{a}_r$: right; Data: size $i \in \{1, 2, 3\}$

Target policy: $\phi_{\mathcal{T}} = \mathbf{a}_\ell \wedge \mathbf{a}_r$ (“[deliver] pairs”: $a^i_\ell \wedge a^i_r$)

Observations = shuffled data: $\hat{\mathbf{a}}_j$ sizes shuffled w.p. e

Clause: “right shoe,” “left shoe,” “pairs”

Evaluation: $[\![\, \hat{\phi} \,]\!] = 1 \;$ if $\; \# \{\text{shoes/pairs}\} \geq 2$

MODEL: EXAMPLE: SHOE DELIVERY

(Atomic) Premises: $\mathbf{a}_\ell$: left shoe $\mathbf{a}_r$: right; Data: size $i \in \{1, 2, 3\}$

Target policy: $\phi_{\mathcal{T}} = \mathbf{a}_\ell \wedge \mathbf{a}_r$ ("[deliver] pairs": $a_\ell^i \wedge a_r^i$)

Observations = shuffled data: $\hat{\mathbf{a}}_j$ sizes shuffled w.p. e

Clause: "right shoe," "left shoe," "pairs"

Evaluation: $[\![\, \hat{\phi} \,]\!] = 1$ if $\# \{\text{shoes/pairs}\} \geq 2$

Contract Φ

MODEL: EXAMPLE: SHOE DELIVERY

(Atomic) Premises: $\mathbf{a}_\ell$: left shoe $\mathbf{a}_r$: right; Data: size $i \in \{1, 2, 3\}$

Target policy: $\phi_{\mathcal{T}} = \mathbf{a}_\ell \wedge \mathbf{a}_r$ ("[deliver] pairs": $a_\ell^i \wedge a_r^i$)

Observations = shuffled data: $\hat{\mathbf{a}}_j$ sizes shuffled w.p. e

Clause: "right shoe," "left shoe," "pairs"

Evaluation: $[\![\, \hat{\phi} \,]\!] = 1 \;\text{ if }\; \#\,\{\text{shoes/pairs}\} \geq 2$

Contract Φ: E.g. "right shoe" $\wedge$ "left shoe"; e.g. "pairs"

MODEL: EXAMPLE: SHOE DELIVERY

(Atomic) Premises: $\mathbf{a}_\ell$: left shoe $\mathbf{a}_r$: right; Data: size $i \in \{1, 2, 3\}$

Target policy: $\phi_{\mathcal{T}} = \mathbf{a}_\ell \wedge \mathbf{a}_r$ ("[deliver] pairs": $a_\ell^i \wedge a_r^i$)

Observations = shuffled data: $\hat{\mathbf{a}}_j$ sizes shuffled w.p. e

Clause: "right shoe," "left shoe," "pairs"

Evaluation: $[\![\, \hat{\phi} \,]\!] = 1 \;\text{ if }\; \#\{\text{shoes/pairs}\} \geq 2$

Contract Φ: E.g. "right shoe" $\wedge$ "left shoe"; e.g. "pairs"

Objective

MODEL: EXAMPLE: SHOE DELIVERY

(Atomic) Premises: $\mathbf{a}_\ell$: left shoe $\mathbf{a}_r$: right; Data: size $i \in \{1, 2, 3\}$

Target policy: $\phi_{\mathcal{T}} = \mathbf{a}_\ell \wedge \mathbf{a}_r$ ("[deliver] pairs": $a_\ell^i \wedge a_r^i$)

Observations = shuffled data: $\hat{\mathbf{a}}_j$ sizes shuffled w.p. e

Clause: "right shoe," "left shoe," "pairs"

Evaluation: $[\![\, \hat{\phi} \,]\!] = 1 \;\text{ if }\; \#\,\{\text{shoes/pairs}\} \geq 2$

Contract Φ: E.g. "right shoe" $\wedge$ "left shoe"; e.g. "pairs"

Objective: $C = c_{\mathrm{I}}\mathsf{Pr}[\text{accept} < \text{two pairs}] + c_{\mathrm{II}}\mathsf{Pr}[\text{reject} > \text{two pairs}]$

MODEL: EXAMPLE: COVENANTS

MODEL: EXAMPLE: COVENANTS

Premises

MODEL: EXAMPLE: COVENANTS

Premises: $\mathbf{a}_{\mathrm{IC}}$: “interest coverage”, $\mathbf{a}_{\$}$: “cash”; Data: date-$i$ statements

MODEL: EXAMPLE: COVENANTS

Premises: $\mathbf{a}_{\mathrm{IC}}$: "interest coverage", $\mathbf{a}_{\$}$: "cash"; Data: date-$i$ statements

Target policy

MODEL: EXAMPLE: COVENANTS

Premises: $\mathbf{a}_{\mathrm{IC}}$: “interest coverage”, $\mathbf{a}_{\$}$: “cash”; Data: date-$i$ statements

Target policy: $\phi_{\mathcal{T}} = \mathbf{a}_{\mathrm{IC}} \wedge \mathbf{a}_{\$}$: both ICR and cash above thresholds

MODEL: EXAMPLE: COVENANTS

Premises: $\mathbf{a}_{\mathrm{IC}}$: “interest coverage”, $\mathbf{a}_{\$}$: “cash”; Data: date-$i$ statements

Target policy: $\phi_{\mathcal{T}} = \mathbf{a}_{\mathrm{IC}} \wedge \mathbf{a}_{\$}$: both ICR and cash above thresholds

Observations = shuffled data

MODEL: EXAMPLE: COVENANTS

Premises: $\mathbf{a}_{\mathrm{IC}}$: “interest coverage”, $\mathbf{a}_{\$}$: “cash”; Data: date-$i$ statements

Target policy: $\phi_{\mathcal{T}} = \mathbf{a}_{\mathrm{IC}} \wedge \mathbf{a}_{\$}$: both ICR and cash above thresholds

Observations = shuffled data: asynchronous or time-aggregated reports...

MODEL: EXAMPLE: COVENANTS

Premises: $\mathbf{a}_{\mathrm{IC}}$: “interest coverage”, $\mathbf{a}_{\$}$: “cash”; Data: date-$i$ statements

Target policy: $\phi_{\mathcal{T}} = \mathbf{a}_{\mathrm{IC}} \wedge \mathbf{a}_{\$}$: both ICR and cash above thresholds

Observations = shuffled data: asynchronous or time-aggregated reports...

Clause

MODEL: EXAMPLE: COVENANTS

Premises: $\mathbf{a}_{\text{IC}}$: “interest coverage”, $\mathbf{a}_{\$}$: “cash”; Data: date-$i$ statements

Target policy: $\phi_{\mathcal{T}} = \mathbf{a}_{\text{IC}} \wedge \mathbf{a}_{\$}$: both ICR and cash above thresholds

Observations = shuffled data: asynchronous or time-aggregated reports...

Clause: “interest coverage,” “cash,” “interest coverage and cash”

MODEL: EXAMPLE: COVENANTS

Premises: $\mathbf{a}_{\mathrm{IC}}$: “interest coverage”, $\mathbf{a}_{\$}$: “cash”; Data: date-$i$ statements

Target policy: $\phi_{\mathcal{T}} = \mathbf{a}_{\mathrm{IC}} \wedge \mathbf{a}_{\$}$: both ICR and cash above thresholds

Observations = shuffled data: asynchronous or time-aggregated reports...

Clause: “interest coverage,” “cash,” “interest coverage and cash”

Evaluation

MODEL: EXAMPLE: COVENANTS

Premises: $\mathbf{a}_{\mathrm{IC}}$: “interest coverage”, $\mathbf{a}_{\$}$: “cash”; Data: date-$i$ statements

Target policy: $\phi_{\mathcal{T}} = \mathbf{a}_{\mathrm{IC}} \wedge \mathbf{a}_{\$}$: both ICR and cash above thresholds

Observations = shuffled data: asynchronous or time-aggregated reports...

Clause: “interest coverage,” “cash,” “interest coverage and cash”

Evaluation: $[\![\hat{\phi}]\!] = 1 \;\text{ if }\; \#\{\text{dates clause satisfied}\} \geq \tau$

MODEL: EXAMPLE: COVENANTS

Premises: $\mathbf{a}_{\mathrm{IC}}$: “interest coverage”, $\mathbf{a}_{\$}$: “cash”; Data: date-$i$ statements

Target policy: $\phi_{\mathcal{T}} = \mathbf{a}_{\mathrm{IC}} \wedge \mathbf{a}_{\$}$: both ICR and cash above thresholds

Observations = shuffled data: asynchronous or time-aggregated reports...

Clause: “interest coverage,” “cash,” “interest coverage and cash”

Evaluation: $[\![\, \hat{\phi} \,]\!] = 1$ if $\#\{\text{dates clause satisfied}\} \geq \tau$

Contract Φ

MODEL: EXAMPLE: COVENANTS

Premises: $\mathbf{a}_{\mathrm{IC}}$: “interest coverage”, $\mathbf{a}_{\$}$: “cash”; Data: date-$i$ statements

Target policy: $\phi_{\mathcal{T}} = \mathbf{a}_{\mathrm{IC}} \wedge \mathbf{a}_{\$}$: both ICR and cash above thresholds

Observations = shuffled data: asynchronous or time-aggregated reports...

Clause: “interest coverage,” “cash,” “interest coverage and cash”

Evaluation: $[\![\, \hat{\phi} \,]\!] = 1 \;\text{ if }\; \#\{\text{dates clause satisfied}\} \geq \tau$

Contract Φ: E.g. “interest coverage” $\wedge$ “cash”; e.g. “both IC and cash”

MODEL: EXAMPLE: COVENANTS

Premises: $\mathbf{a}_{\mathrm{IC}}$: "interest coverage", $\mathbf{a}_{\$}$: "cash"; Data: date-$i$ statements

Target policy: $\phi_{\mathcal{T}} = \mathbf{a}_{\mathrm{IC}} \wedge \mathbf{a}_{\$}$: both ICR and cash above thresholds

Observations = shuffled data: asynchronous or time-aggregated reports...

Clause: "interest coverage," "cash," "interest coverage and cash"

Evaluation: $[\![\, \hat{\phi} \,]\!] = 1 \;\text{ if }\; \#\{\text{dates clause satisfied}\} \geq \tau$

Contract Φ: E.g. "interest coverage" $\wedge$ "cash"; e.g. "both IC and cash"

Objective

MODEL: EXAMPLE: COVENANTS

Premises: $\mathbf{a}_{\text{IC}}$: “interest coverage”, $\mathbf{a}_{\$}$: “cash”; Data: date-$i$ statements

Target policy: $\phi_{\mathcal{T}} = \mathbf{a}_{\text{IC}} \wedge \mathbf{a}_{\$}$: both ICR and cash above thresholds

Observations = shuffled data: asynchronous or time-aggregated reports...

Clause: “interest coverage,” “cash,” “interest coverage and cash”

Evaluation: $[\![\, \hat{\phi} \,]\!] = 1 \ \text{ if } \ \#\{\text{dates clause satisfied}\} \geq \tau$

Contract Φ: E.g. “interest coverage” $\wedge$ “cash”; e.g. “both IC and cash”

Objective: $C = c_{\text{I}}\mathsf{Pr}[\text{accept if violated}] + c_{\text{II}}\mathsf{Pr}[\text{reject if satisfied}]$

DEFINITIONS: PREMISE- AND TARGET-BASED

DEFINITIONS: PREMISE- AND TARGET-BASED

DEFINITIONS: PREMISE- AND TARGET-BASED

$\Phi^{\mathcal{A}}$ called <u>premise-based</u> if there are $|\mathcal{T}|$ clauses s.t. $\{\mathcal{M}_m\}_m = \big\{\{\mathbf{a}_j\}\big\}_{j\in\mathcal{T}}$

DEFINITIONS: PREMISE- AND TARGET-BASED

$\Phi^{\mathcal{A}}$ called premise-based if there are $|\mathcal{T}|$ clauses s.t. $\{\mathcal{M}_m\}_m = \big\{\{\mathbf{a}_j\}\big\}_{j\in\mathcal{T}}$

$\Phi^{\mathcal{T}}$ called target-based if there is one clause s.t. $\mathcal{M} = \bigcup_{j\in\mathcal{T}} \{\mathbf{a}_j\}$

BENCHMARKS: $I = 1$ AND $e = 0$

BENCHMARK: $I = 1$

Each premise is just a truth value in $\{0, 1\}$, so $[\![\phi]\!] = \phi$

BENCHMARK: $I = 1$

Each premise is just a truth value in $\{0, 1\}$, so $[\![\phi]\!] = \phi$

Example: $\mathbf{a}_j = 1$ and $\mathbf{a}_k = 0$

BENCHMARK: $I = 1$

Each premise is just a truth value in $\{0, 1\}$, so $[\![\phi]\!] = \phi$

Example: $\mathbf{a}_j = 1$ and $\mathbf{a}_k = 0 \Rightarrow [\![\mathbf{a}_j]\!] \wedge [\![\mathbf{a}_k]\!] = 1 \wedge 0 = 0 = [\![\mathbf{a}_j \wedge \mathbf{a}_k]\!]$

BENCHMARK: $I = 1$

Each premise is just a truth value in $\{0, 1\}$, so $[\![\phi]\!] = \phi$

Example: $\mathbf{a}_j = 1$ and $\mathbf{a}_k = 0 \Rightarrow [\![\mathbf{a}_j]\!] \wedge [\![\mathbf{a}_k]\!] = 1 \wedge 0 = 0 = [\![\mathbf{a}_j \wedge \mathbf{a}_k]\!]$

$\Rightarrow [\![\Phi^{\mathcal{A}}]\!] = [\![\Phi^{\mathcal{T}}]\!]$: premise- and target-based coincide

BENCHMARK: $I = 1$

Each premise is just a truth value in $\{0, 1\}$, so $[\![\phi]\!] = \phi$

Example: $\mathbf{a}_j = 1$ and $\mathbf{a}_k = 0 \Rightarrow [\![\mathbf{a}_j]\!] \wedge [\![\mathbf{a}_k]\!] = 1 \wedge 0 = 0 = [\![\mathbf{a}_j \wedge \mathbf{a}_k]\!]$

$\Rightarrow [\![\Phi^{\mathcal{A}}]\!] = [\![\Phi^{\mathcal{T}}]\!]$: premise- and target-based coincide

Evaluation reduces to logic; grouping of premises irrelevant

BENCHMARK: $e = 0$

No shuffling, so $\hat{\mathbf{a}}_j = \mathbf{a}_j$ for all j

BENCHMARK: $e = 0$

No shuffling, so $\hat{\mathbf{a}}_j = \mathbf{a}_j$ for all j

Example: $\mathbf{a}_j = 110$ and $\mathbf{a}_k = 101$

BENCHMARK: $e = 0$

No shuffling, so $\hat{\mathbf{a}}_j = \mathbf{a}_j$ for all j

Example: $\mathbf{a}_j = 110$ and $\mathbf{a}_k = 101 \Rightarrow \llbracket \Phi^{\mathcal{T}} \rrbracket = \llbracket \hat{\mathbf{a}}_j \wedge \hat{\mathbf{a}}_k \rrbracket = \llbracket \mathbf{a}_j \wedge \mathbf{a}_k \rrbracket = \llbracket 100 \rrbracket$

BENCHMARK: $e = 0$

No shuffling, so $\hat{\mathbf{a}}_j = \mathbf{a}_j$ for all j

Example: $\mathbf{a}_j = 110$ and $\mathbf{a}_k = 101 \Rightarrow \llbracket \Phi^{\mathcal{T}} \rrbracket = \llbracket \hat{\mathbf{a}}_j \wedge \hat{\mathbf{a}}_k \rrbracket = \llbracket \mathbf{a}_j \wedge \mathbf{a}_k \rrbracket = \llbracket 100 \rrbracket$

$\Rightarrow$ target-based contract implements target policy: $\llbracket \Phi^{\mathcal{T}} \rrbracket = \llbracket \phi_{\mathcal{T}} \rrbracket$

LOOPHOLE AND SHUFFLE RISK SETS

DEFINITIONS: PREMISE- AND TARGET-BASED

Loophole states LH: Data realizations s.t. $[\![\Phi^{\mathcal{A}}]\!] \neq [\![\Phi^{\mathcal{T}}]\!]$ even if $e = 0$

DEFINITIONS: PREMISE- AND TARGET-BASED

Loophole states LH: Data realizations s.t. $[\![\Phi^{\mathcal{A}}]\!] \neq [\![\Phi^{\mathcal{T}}]\!]$ even if $e = 0$

Shuffle risk states SR: Data realizations s.t. $\mathsf{Pr}\left[[\![\Phi^{\mathcal{A}}]\!] \neq [\![\Phi^{\mathcal{T}}]\!] \,\middle|\, \{\mathbf{a}_j\}_j \right] > 0$

EXAMPLE

EXAMPLE

Two premises with three data each and $\tau = 2$:

EXAMPLE

Two premises with three data each and $\tau = 2$:

i
1
2
3

EXAMPLE

Two premises with three data each and $\tau = 2$:

i	$\mathbf{a}_j$
1	1
2	1
3	0

EXAMPLE

Two premises with three data each and $\tau = 2$:

i	$\mathbf{a}_j$
1	1
2	1
3	0
# of 1s $\geq \tau$	

EXAMPLE

Two premises with three data each and $\tau = 2$:

i	$\mathbf{a}_j$
1	1
2	1
3	0
# of 1s $\geq \tau$	**1**

EXAMPLE

Two premises with three data each and $\tau = 2$:

i	$\mathbf{a}_j$	$\mathbf{a}_k$
1	1	1
2	1	0
3	0	1
# of 1s $\geq \tau$	**1**	

EXAMPLE

Two premises with three data each and $\tau = 2$:

i	$\mathbf{a}_j$	$\mathbf{a}_k$
1	1	1
2	1	0
3	0	1
# of 1s $\geq \tau$	**1**	**1**

EXAMPLE

Two premises with three data each and $\tau = 2$:

i	$\mathbf{a}_j$	$\mathbf{a}_k$
1	1	1
2	1	0
3	0	1
# of 1s $\geq \tau$	**1**	**1**

Premise-based: $[\![\Phi^{\mathcal{A}}]\!]$

EXAMPLE

Two premises with three data each and $\tau = 2$:

i	$\mathbf{a}_j$	$\mathbf{a}_k$
1	1	1
2	1	0
3	0	1
# of 1s $\geq \tau$	**1**	**1**

Premise-based: $\llbracket \Phi^{\mathcal{A}} \rrbracket = \llbracket \mathbf{a}_j \rrbracket \wedge \llbracket \mathbf{a}_k \rrbracket$

EXAMPLE

Two premises with three data each and $\tau = 2$:

i	$\mathbf{a}_j$	$\mathbf{a}_k$
1	1	1
2	1	0
3	0	1
# of 1s $\geq \tau$	**1**	**1**

Premise-based: $[\![\Phi^{\mathcal{A}}]\!] = [\![\mathbf{a}_j]\!] \wedge [\![\mathbf{a}_k]\!] = [\![110]\!] \wedge [\![101]\!]$

EXAMPLE

Two premises with three data each and $\tau = 2$:

i	$\mathbf{a}_j$	$\mathbf{a}_k$
1	1	1
2	1	0
3	0	1
# of 1s $\geq \tau$	**1**	**1**

Premise-based: $[\![\Phi^{\mathcal{A}}]\!] = [\![\mathbf{a}_j]\!] \wedge [\![\mathbf{a}_k]\!] = [\![110]\!] \wedge [\![101]\!] = 1 \wedge 1 = 1$

EXAMPLE

Two premises with three data each and $\tau = 2$:

i	$\mathbf{a}_j$	$\mathbf{a}_k$	$\mathbf{a}_j \wedge \mathbf{a}_k$
1	1	1	1
2	1	0	0
3	0	1	0
# of 1s $\geq \tau$	**1**	**1**	

Premise-based: $[\![\Phi^{\mathcal{A}}]\!] = [\![\mathbf{a}_j]\!] \wedge [\![\mathbf{a}_k]\!] = [\![110]\!] \wedge [\![101]\!] = 1 \wedge 1 = 1$

Target-based:

EXAMPLE

Two premises with three data each and $\tau = 2$:

i	$\mathbf{a}_j$	$\mathbf{a}_k$	$\mathbf{a}_j \wedge \mathbf{a}_k$
1	1	1	1
2	1	0	0
3	0	1	0
# of 1s $\geq \tau$	**1**	**1**	

Premise-based: $[\![\Phi^{\mathcal{A}}]\!] = [\![\mathbf{a}_j]\!] \wedge [\![\mathbf{a}_k]\!] = [\![110]\!] \wedge [\![101]\!] = 1 \wedge 1 = 1$

Target-based: $[\![\Phi^{\mathcal{T}}]\!] = [\![\mathbf{a}_j \wedge \mathbf{a}_k]\!]$

EXAMPLE

Two premises with three data each and $\tau = 2$:

i	$\mathbf{a}_j$	$\mathbf{a}_k$	$\mathbf{a}_j \wedge \mathbf{a}_k$
1	1	1	1
2	1	0	0
3	0	1	0
# of 1s $\geq \tau$	**1**	**1**	

<u>Premise-based</u>: $\llbracket \Phi^{\mathcal{A}} \rrbracket = \llbracket \mathbf{a}_j \rrbracket \wedge \llbracket \mathbf{a}_k \rrbracket = \llbracket 110 \rrbracket \wedge \llbracket 101 \rrbracket = 1 \wedge 1 = 1$

<u>Target-based</u>: $\llbracket \Phi^{\mathcal{T}} \rrbracket = \llbracket \mathbf{a}_j \wedge \mathbf{a}_k \rrbracket = \llbracket 110 \wedge 101 \rrbracket$

EXAMPLE

Two premises with three data each and $\tau = 2$:

i	$\mathbf{a}_j$	$\mathbf{a}_k$	$\mathbf{a}_j \wedge \mathbf{a}_k$
1	1	1	1
2	1	0	0
3	0	1	0
# of 1s $\geq \tau$	**1**	**1**	

Premise-based: $[\![\Phi^{\mathcal{A}}]\!] = [\![\mathbf{a}_j]\!] \wedge [\![\mathbf{a}_k]\!] = [\![110]\!] \wedge [\![101]\!] = 1 \wedge 1 = 1$

Target-based: $[\![\Phi^{\mathcal{T}}]\!] = [\![\mathbf{a}_j \wedge \mathbf{a}_k]\!] = [\![110 \wedge 101]\!] = [\![100]\!]$

EXAMPLE

Two premises with three data each and $\tau = 2$:

i	$\mathbf{a}_j$	$\mathbf{a}_k$	$\mathbf{a}_j \wedge \mathbf{a}_k$
1	1	1	1
2	1	0	0
3	0	1	0
# of 1s $\geq \tau$	**1**	**1**	**0**

Premise-based: $[\![\Phi^{\mathcal{A}}]\!] = [\![\mathbf{a}_j]\!] \wedge [\![\mathbf{a}_k]\!] = [\![110]\!] \wedge [\![101]\!] = 1 \wedge 1 = 1$

Target-based: $[\![\Phi^{\mathcal{T}}]\!] = [\![\mathbf{a}_j \wedge \mathbf{a}_k]\!] = [\![110 \wedge 101]\!] = [\![100]\!] = 0$

EXAMPLE

Two premises with three data each and $\tau = 2$:

i	$\mathbf{a}_j$	$\mathbf{a}_k$	$\mathbf{a}_j \wedge \mathbf{a}_k$
1	1	1	1
2	1	0	0
3	0	1	0
# of 1s $\geq \tau$	**1**	**1**	**0**

Premise-based: $[\![\Phi^{\mathcal{A}}]\!] = [\![\mathbf{a}_j]\!] \wedge [\![\mathbf{a}_k]\!] = [\![110]\!] \wedge [\![101]\!] = 1 \wedge 1 = 1$

Target-based: $[\![\Phi^{\mathcal{T}}]\!] = [\![\mathbf{a}_j \wedge \mathbf{a}_k]\!] = [\![110 \wedge 101]\!] = [\![100]\!] = 0$

Loophole: Evaluation & conjunction don't commute: $[\![\mathbf{a}_j]\!] \wedge [\![\mathbf{a}_k]\!] \neq [\![\mathbf{a}_j \wedge \mathbf{a}_k]\!]$

EXAMPLE: SHUFFLE RISK

Observe permuted data

i	$\mathbf{a}_j$	$\mathbf{a}_k$	$\mathbf{a}_j \wedge \mathbf{a}_k$
1	1	1	1
2	1	0	0
3	0	1	0
# of 1s $\geq \tau$	**1**	**1**	**0**

EXAMPLE: SHUFFLE RISK

Observe permuted data E.g. $\hat{\mathbf{a}}_j = \mathbf{a}_j$ (no shuffle)

i	$\mathbf{a}_j$	$\mathbf{a}_k$	$\mathbf{a}_j \wedge \mathbf{a}_k$	$\hat{\mathbf{a}}_j$
1	1	1	1	1
2	1	0	0	1
3	0	1	0	0
# of 1s $\geq \tau$	**1**	**1**	**0**	**1**

EXAMPLE: SHUFFLE RISK

Observe permuted data E.g. $\hat{\mathbf{a}}_j = \mathbf{a}_j$ (no shuffle) ; $\hat{\mathbf{a}}_k = 110$ (shuffle of $\mathbf{a}_k$)

i	$\mathbf{a}_j$	$\mathbf{a}_k$	$\mathbf{a}_j \wedge \mathbf{a}_k$	$\hat{\mathbf{a}}_j$	$\hat{\mathbf{a}}_k$
1	1	1	1	1	1
2	1	0	0	1	1
3	0	1	0	0	0
# of 1s $\geq \tau$	**1**	**1**	**0**	**1**	**1**

EXAMPLE: SHUFFLE RISK

Observe permuted data E.g. $\hat{\mathbf{a}}_j = \mathbf{a}_j$ (no shuffle) ; $\hat{\mathbf{a}}_k = 110$ (shuffle of $\mathbf{a}_k$)

i	$\mathbf{a}_j$	$\mathbf{a}_k$	$\mathbf{a}_j \wedge \mathbf{a}_k$	$\hat{\mathbf{a}}_j$	$\hat{\mathbf{a}}_k$
1	1	1	1	1	1
2	1	0	0	1	1
3	0	1	0	0	0
# of 1s $\geq \tau$	**1**	**1**	**0**	**1**	**1**

Marginals preserved ⇒ premise-based evaluation unchanged

EXAMPLE: SHUFFLE RISK

Observe permuted data E.g. $\hat{\mathbf{a}}_j = \mathbf{a}_j$ (no shuffle) ; $\hat{\mathbf{a}}_k = 110$ (shuffle of $\mathbf{a}_k$)

i	$\mathbf{a}_j$	$\mathbf{a}_k$	$\mathbf{a}_j \wedge \mathbf{a}_k$	$\hat{\mathbf{a}}_j$	$\hat{\mathbf{a}}_k$	$\hat{\mathbf{a}}_j \wedge \hat{\mathbf{a}}_k$
1	1	1	1	1	1	1
2	1	0	0	1	1	1
3	0	1	0	0	0	0
# of 1s $\geq \tau$	**1**	**1**	**0**	**1**	**1**	**1**

Marginals preserved $\Rightarrow$ premise-based evaluation unchanged

Joint changed: 101 → 110 $\Rightarrow$ target-based evaluation flips

RESULTS

LEMMA

LEMMA: LOOPHOLES $\subsetneq$ SHUFFLE RISK

$\mathsf{LH} \subsetneq \mathsf{SR}$

LEMMA: LOOPHOLES $\subsetneq$ SHUFFLE RISK: PROOF

LH $\Rightarrow$ SR

LEMMA: LOOPHOLES $\subsetneq$ SHUFFLE RISK: PROOF

$\mathsf{LH} \Rightarrow \mathsf{SR}$: Loophole states: Premise-based accepts, target-based rejects

LEMMA: LOOPHOLES $\subsetneq$ SHUFFLE RISK: PROOF

$\mathsf{LH} \Rightarrow \mathsf{SR}$: Loophole states: Premise-based accepts, target-based rejects

Premise based accepts if $\min\limits_{j} \sum\limits_{i} a^i_j \geq \tau$

LEMMA: LOOPHOLES $\subsetneq$ SHUFFLE RISK: PROOF

LH $\Rightarrow$ SR: Loophole states: Premise-based accepts, target-based rejects

Premise based accepts if $\min_j \sum_i a_j^i \geq \tau$

Target-based accepts too if shuffle “aligns” all 1’s

LEMMA: LOOPHOLES $\subsetneq$ SHUFFLE RISK: PROOF

$\mathsf{LH} \Rightarrow \mathsf{SR}$: Loophole states: Premise-based accepts, target-based rejects

Premise based accepts if $\min_j \sum_i a_j^i \geq \tau$

Target-based accepts too if shuffle “aligns” all 1’s

$\mathsf{SR} \not\Rightarrow \mathsf{LH}$

LEMMA: LOOPHOLES $\subsetneq$ SHUFFLE RISK: PROOF

LH $\Rightarrow$ SR: Loophole states: Premise-based accepts, target-based rejects

Premise based accepts if $\min_j \sum_i a_j^i \geq \tau$

Target-based accepts too if shuffle "aligns" all 1's

SR $\not\Rightarrow$ LH: E.g. $\mathbf{a}_j = 110$, $\mathbf{a}_k = 110$

LEMMA: LOOPHOLES $\subsetneq$ SHUFFLE RISK: PROOF

$\mathsf{LH} \Rightarrow \mathsf{SR}$: Loophole states: Premise-based accepts, target-based rejects

Premise based accepts if $\min_j \sum_i a_j^i \geq \tau$

Target-based accepts too if shuffle "aligns" all 1's

$\mathsf{SR} \not\Rightarrow \mathsf{LH}$: E.g. $\mathbf{a}_j = 110$, $\mathbf{a}_k = 110$: $[\![\mathbf{a}_j \wedge \mathbf{a}_k]\!] = [\![110]\!] = 1$ if $\tau = 2$

LEMMA: LOOPHOLES $\subsetneq$ SHUFFLE RISK: PROOF

$\mathsf{LH} \Rightarrow \mathsf{SR}$: Loophole states: Premise-based accepts, target-based rejects

Premise based accepts if $\min_j \sum_i a^i_j \geq \tau$

Target-based accepts too if shuffle "aligns" all 1's

$\mathsf{SR} \not\Rightarrow \mathsf{LH}$: E.g. $\mathbf{a}_j = 110$, $\mathbf{a}_k = 110$: $[\![\mathbf{a}_j \wedge \mathbf{a}_k]\!] = [\![110]\!] = 1$ if $\tau = 2$ (not LH)

LEMMA: LOOPHOLES $\subsetneq$ SHUFFLE RISK: PROOF

$\mathsf{LH} \Rightarrow \mathsf{SR}$: Loophole states: Premise-based accepts, target-based rejects

Premise based accepts if $\min_j \sum_i a_j^i \geq \tau$

Target-based accepts too if shuffle "aligns" all 1's

$\mathsf{SR} \not\Rightarrow \mathsf{LH}$: E.g. $\mathbf{a}_j = 110$, $\mathbf{a}_k = 110$: $[\![\mathbf{a}_j \wedge \mathbf{a}_k]\!] = [\![110]\!] = 1$ if $\tau = 2$ (not LH)

But shuffle could be $\hat{\mathbf{a}}_j = 110, \hat{\mathbf{a}}_k = 101$

LEMMA: LOOPHOLES $\subsetneq$ SHUFFLE RISK: PROOF

$\mathsf{LH} \Rightarrow \mathsf{SR}$: Loophole states: Premise-based accepts, target-based rejects

Premise based accepts if $\min_j \sum_i a_j^i \geq \tau$

Target-based accepts too if shuffle “aligns” all 1’s

$\mathsf{SR} \not\Rightarrow \mathsf{LH}$: E.g. $\mathbf{a}_j = 110$, $\mathbf{a}_k = 110$: $[\![\mathbf{a}_j \wedge \mathbf{a}_k]\!] = [\![110]\!] = 1$ if $\tau = 2$ (not LH)

But shuffle could be $\hat{\mathbf{a}}_j = 110, \hat{\mathbf{a}}_k = 101$: $[\![\hat{\mathbf{a}}_j \wedge \hat{\mathbf{a}}_k]\!] = [\![100]\!] = 0$ (yes SR)

R1: LOOPHOLES

R1: LOOPHOLES

With $e > 0$, if $\mathsf{Pr}[\mathsf{SR} \setminus \mathsf{LH}]$ large, then $\Phi^{\mathcal{A}} \succ \Phi^{\mathcal{T}}$

R1: LOOPHOLES: INTUITION

Shuffled data ($e > 0$): Evaluation errors for all but not only loopholes

Accept loopholes to reduce shuffling error

LOOPHOLES IN PRACTICE: J. CREW

J. Crew was distressed and wanted to free up collateral

Credit agreement ostensibly prohibited moving it to unrestricted sub, but...

Clause $\mathbf{a}_j$: "Can move assets to restricted sub"

Clause $\mathbf{a}_k$: "Can move assets from restricted to unrestricted sub"

LOOPHOLES IN PRACTICE: J. CREW

J. Crew was distressed and wanted to free up collateral

Credit agreement ostensibly prohibited moving it to unrestricted sub, but...

Clause $\mathbf{a}_j$: "Can move assets to restricted sub"

Clause $\mathbf{a}_k$: "Can move assets from restricted to unrestricted sub"

$\Rightarrow \mathbf{a}_j \wedge \mathbf{a}_k$: "Can move assets to restricted sub [via unrestricted]"

LOOPHOLES IN PRACTICE: J. CREW

J. Crew was distressed and wanted to free up collateral

Credit agreement ostensibly prohibited moving it to unrestricted sub, but...

Clause $\mathbf{a}_j$: "Can move assets to restricted sub"

Clause $\mathbf{a}_k$: "Can move assets from restricted to unrestricted sub"

$\Rightarrow \mathbf{a}_j \wedge \mathbf{a}_k$: "Can move assets to restricted sub [via unrestricted]"

I.e. loophole: $[\![\mathbf{a}_j]\!] \wedge [\![\mathbf{a}_k]\!] \neq [\![\mathbf{a}_j \wedge \mathbf{a}_k]\!]$

LOOPHOLES IN THE LITERATURE

Katz 10: Many loopholes openly exploited after much litigation

LOOPHOLES IN THE LITERATURE

Katz 10: Many loopholes openly exploited after much litigation

  - Refutes form-vs-substance, intent-to-evade, spirit-vs-letter theories

LOOPHOLES IN THE LITERATURE

Katz 10: Many loopholes openly exploited after much litigation

- Refutes form-vs-substance, intent-to-evade, spirit-vs-letter theories
- Say loopholes akin to voting paradoxes (Arrow's Theorem; cf. Saari 01)

LOOPHOLES IN THE LITERATURE

Katz 10: Many loopholes openly exploited after much litigation

- Refutes form-vs-substance, intent-to-evade, spirit-vs-letter theories
- Say loopholes akin to voting paradoxes (Arrow's Theorem; cf. Saari 01)

Doctrinal Paradox (DP)

LOOPHOLES IN THE LITERATURE

Katz 10: Many loopholes openly exploited after much litigation

  Refutes form-vs-substance, intent-to-evade, spirit-vs-letter theories

  Say loopholes akin to voting paradoxes (Arrow's Theorem; cf. Saari 01)

Doctrinal Paradox (DP): Vote on premises $\neq$ on targets (List–Pettit 02)

LOOPHOLES IN THE LITERATURE

Katz 10: Many loopholes openly exploited after much litigation

- Refutes form-vs-substance, intent-to-evade, spirit-vs-letter theories
- Say loopholes akin to voting paradoxes (Arrow's Theorem; cf. Saari 01)

Doctrinal Paradox (DP): Vote on premises $\neq$ on targets (List–Pettit 02)

- Aggregation paradox about beliefs over logical statements (nests Arrow)

LOOPHOLES IN THE LITERATURE

Katz 10: Many loopholes openly exploited after much litigation

  Refutes form-vs-substance, intent-to-evade, spirit-vs-letter theories

  Say loopholes akin to voting paradoxes (Arrow's Theorem; cf. Saari 01)

Doctrinal Paradox (DP): Vote on premises $\neq$ on targets (List–Pettit 02)

  Aggregation paradox about beliefs over logical statements (nests Arrow)

We nest DP (observations $\leftrightarrow$ agent beliefs) and connect with loopholes

LOOPHOLES IN THE LITERATURE

Katz 10: Many loopholes openly exploited after much litigation

- Refutes form-vs-substance, intent-to-evade, spirit-vs-letter theories
- Say loopholes akin to voting paradoxes (Arrow's Theorem; cf. Saari 01)

Doctrinal Paradox (DP): Vote on premises $\neq$ on targets (List–Pettit 02)

- Aggregation paradox about beliefs over logical statements (nests Arrow)

We nest DP (observations $\leftrightarrow$ agent beliefs) and connect with loopholes

- Show why loopholes persist $e > 0$

R2: ERRORS DUE TO COMPLEXITY

R2: ERRORS DUE TO COMPLEXITY

If $c_{\mathrm{I}}\mathsf{Pr}[\mathsf{LH}]$ sufficiently high, $\Phi^{\mathcal{T}} \succ \Phi^{\mathcal{A}}$

R2: COMPLEXITY RISK: INTUITION

If loopholes likely or costly use complex target-based contract (long clauses)

Cost: With interacting premises, shuffle risk $\Rightarrow$ likely evaluated with error

R2: COMPLEXITY RISK: INTUITION

If loopholes likely or costly use complex target-based contract (long clauses)

Cost: With interacting premises, shuffle risk ⇒ likely evaluated with error

Empirically: Complexity leads to errors in taxes and regulator compliance
OECD, Benzarti 20, Zwick 21, Behn et al 22...

R3: REDUNDANCY

DEFINITION: REDUNDANCY

$\Phi^{\mathcal{R}}$ called redundant if $j \in \mathcal{M}_m \cap \mathcal{M}_{m'}$ for some clauses $j \in \mathcal{A},\ m \neq m'$

R3: REDUNDANCY

If $e > 0$ not too high and $c_{\mathrm{I}}\mathsf{Pr}[\mathsf{LH}]$ high enough, both $\Phi^{\mathcal{R}} \succ \Phi^{\mathcal{T}}$ & $\Phi^{\mathcal{R}} \succ \Phi^{\mathcal{A}}$

R3: REDUNDANCY: INTUITION

Benefit w.r.t. target based: Fewer false positives

Repeated clause ⇒ require multiple shuffling errors

Benefit w.r.t. premise-based: Fewer loopholes

Keeps joint information

R3: REDUNDANCY: INTUITION

Benefit w.r.t. target based: Fewer false positives

- Repeated clause ⇒ require multiple shuffling errors

Benefit w.r.t. premise-based: Fewer loopholes

- Keeps joint information

Cost: False negatives: Shuffles in any clause can cause rejection

R4: INCOMPLETENESS

DEFINITION: INCOMPLETENESS

$\Phi^{\mathcal{I}}$ called incomplete w.r.t. $\mathcal{T}$ if $j \notin \bigcup \mathcal{M}_m$ for some $j \in \mathcal{T}$

R4: INCOMPLETENESS

If e and c_{II} high and $\mathsf{Pr}\left[[\![\Phi^{\mathcal{I}}]\!] > [\![\phi^{\mathcal{T}}]\!]\right]$ low, then $\Phi^{\mathcal{I}} \succ \Phi^{\mathcal{T}}$

R4: INCOMPLETENESS: INTUITION

Long clauses induce shuffling risk ⇒ omit some premises to avoid it

Akin to mismeasurement in Holmström–Milgrom:

  Decrease power on distortionary tasks

R5: CONSERVATISM

EXTENSION

Clause-dependent thresholds $\tau_{\mathcal{M}}$

DEFINITION: CONSERVATISM

Let $\tau_{\mathcal{T}}$ be threshold of target

$\Phi^{\mathcal{C}}$ is $\underline{\text{conservative}}$ if $\tau_{\mathcal{M}_m} > \tau_{\mathcal{T}}$ for every m

R5: CONSERVATISM

Define $\mathsf{ACC}_\tau(\Phi) := \big\{\{\mathbf{a}_j\}_j \ : [\![\Phi]\!] = 1\big\}$

If $\mathsf{Pr}\left[[\![\phi_\mathcal{T}]\!] = 1 \mid \mathsf{ACC}_\tau(\Phi^\mathcal{A}) \setminus \mathsf{ACC}_{\tau+1}(\Phi^\mathcal{A})\right] < \dfrac{c_\mathrm{I}}{c_\mathrm{I} + c_\mathrm{II}}$, then $\Phi^\mathcal{C} \succ \Phi^\mathcal{A}$

R5: CONSERVATISM: INTUITION

Premise-based evaluations gives too many false positives

Conservatism decreases them via increased threshold

R5: CONSERVATISM: INTUITION

Premise-based evaluations gives too many false positives

Conservatism decreases them via increased threshold

Feature: Relies on only marginals ⇒ not subject to shuffle risk

R5: CONSERVATISM: APPLICATIONS

Law: “beyond reasonable doubt,” “reasonable suspicion,” “probable cause”
In re Winship, 397 U.S. 358 (1970), Terry v. Ohio, 392 U.S. 1 (1968)

Accounting & auditing: High standard for good news Basu 97; Watts 03

E.g. losses recognized faster than gains

CONCLUSION

CONCLUSION

Model of modular contracts in formal language s.t. shuffled data

- Loopholes
- Errors of complexity
- Landmines
- Incompleteness
- Conservatism

MODULAR CONTRACTS