# Proving Expected Sensitivity of Probabilistic Programs

Gilles Barthe
Thomas Espitau
Benjamin Grégoire
Justin Hsu
Pierre-Yves Strub

# Program Sensitivity

## Similar inputs $\rightarrow$ similar outputs

- Given: distances $d_{in}$ on inputs, $d_{out}$ on outputs
- Want: for all inputs $in_1, in_2$,

$$d_{out}(P(in_1), P(in_2)) \leq d_{in}(in_1, in_2)$$

# Program Sensitivity

## Similar inputs $\rightarrow$ similar outputs

- Given: distances $d_{in}$ on inputs, $d_{out}$ on outputs
- Want: for all inputs $in_1, in_2$,

$$d_{out}(P(in_1), P(in_2)) \leq d_{in}(in_1, in_2)$$

If $P$ is sensitive and $Q$ is sensitive, then $Q \circ P$ is sensitive

# Probabilistic Program Sensitivity?

Similar inputs $\to$ similar output distributions

- Given: distances $d_{in}$ on inputs, $d_{out}$ on output distributions
- Want: for all inputs $in_1, in_2$,

$$d_{out}(P(in_1), P(in_2)) \leq d_{in}(in_1, in_2)$$

# Probabilistic Program Sensitivity?

Similar inputs $\to$ similar output distributions

- Given: distances $d_{in}$ on inputs, $d_{out}$ on output distributions
- Want: for all inputs $in_1, in_2$,

$$d_{out}(P(in_1), P(in_2)) \leq d_{in}(in_1, in_2)$$

What distance $d_{out}$ should we take?

## Our contributions

- Coupling-based definition of probabilistic sensitivity
- Relational program logic $\mathbb{E}$PRHL
- Formalized examples: stability and convergence

# What is a good definition of probabilistic sensitivity?

# One possible definition: output distributions close

For two distributions $\mu_1, \mu_2$ over a set $A$:

$$d_{out}(\mu_1, \mu_2) \triangleq k \cdot \max_{E \subseteq A} |\mu_1(E) - \mu_2(E)|$$

# One possible definition: output distributions close

For two distributions $\mu_1, \mu_2$ over a set $A$:

$$d_{out}(\mu_1, \mu_2) \triangleq k \cdot \max_{E \subseteq A} |\mu_1(E) - \mu_2(E)|$$

$k$-Uniform sensitivity

- Larger $k \rightarrow$ closer output distributions
- Strong guarantee: probabilities close for all sets of outputs

# Application: probabilistic convergence/mixing

Probabilistic program forgets initial state

- Given: probabilistic loop, two different input states
- Want: state distributions converge to same distribution

# Application: probabilistic convergence/mixing

## Probabilistic program forgets initial state

- Given: probabilistic loop, two different input states
- Want: state distributions converge to same distribution

## Consequence of $k$-uniform sensitivity

- As number of iterations $T$ increases, prove $k$-uniform sensitivity for larger and larger $k(T)$
- Relation between $k$ and $T$ describes speed of convergence

# Another possible definition: average outputs close

For two distributions $\mu_1, \mu_2$ over real numbers:

$$d_{out}(\mu_1, \mu_2) \triangleq k \cdot |\mathbb{E}[\mu_1] - \mathbb{E}[\mu_2]|$$

# Another possible definition: average outputs close

For two distributions $\mu_1, \mu_2$ over real numbers:

$$d_{out}(\mu_1, \mu_2) \triangleq k \cdot |\mathbb{E}[\mu_1] - \mathbb{E}[\mu_2]|$$

$k$-Mean sensitivity

- Larger $k \to$ closer averages
- Weaker guarantee than uniform sensitivity

# Application: algorithmic stability

Machine learning algorithm $A$

- Input: set $S$ of training examples
- Output: list of numeric parameters (randomized)

Danger: overfitting

- Output parameters depend too much on training set $S$
- Low error on training set, high error on new examples

# Application: algorithmic stability

## One way to prevent overfitting

- $L$ maps $S$ to average error of randomized learning algorithm $A$
- If $|L(S) - L(S')|$ is small for all training sets $S, S'$ differing in a single example, then $A$ does not overfit too much

# Application: algorithmic stability

## One way to prevent overfitting

- $L$ maps $S$ to average error of randomized learning algorithm $A$
- If $|L(S) - L(S')|$ is small for all training sets $S, S'$ differing in a single example, then $A$ does not overfit too much

$L$ should be mean sensitive

## Wanted: a general definition that is ...

- Expressive
- Easy to reason about

# Ingredient #1: Probabilistic coupling

A coupling models two distributions with one distribution

Given two distributions $\mu_1, \mu_2 \in \mathsf{Distr}(A)$, a joint distribution $\mu \in \mathsf{Distr}(A \times A)$ is a coupling if

$$\pi_1(\mu) = \mu_1 \quad \text{and} \quad \pi_2(\mu) = \mu_2$$

# Ingredient #1: Probabilistic coupling

**A coupling models two distributions with one distribution**

Given two distributions $\mu_1, \mu_2 \in \mathsf{Distr}(A)$, a joint distribution $\mu \in \mathsf{Distr}(A \times A)$ is a coupling if

$$\pi_1(\mu) = \mu_1 \quad \text{and} \quad \pi_2(\mu) = \mu_2$$

**Typical pattern**

Prove property about two (output) distributions by constructing a coupling with certain properties

# Ingredient $\#2$: Lift distance on outputs

Given:

- Two distributions $\mu_1, \mu_2 \in \mathsf{Distr}(A)$
- Ground distance $d : A \times A \to \mathbb{R}^+$

# Ingredient $\#2$: Lift distance on outputs

Given:

- Two distributions $\mu_1, \mu_2 \in$ Distr$(A)$
- Ground distance $d : A \times A \to \mathbb{R}^+$

Define distance on distributions:

$$d^\#(\mu_1, \mu_2) \triangleq \min_{\mu \in C(\mu_1, \mu_2)} \mathbb{E}_\mu[d]$$

# Ingredient $\#2$: Lift distance on outputs

Given:

- Two distributions $\mu_1, \mu_2 \in$ Distr$(A)$
- Ground distance $d : A \times A \to \mathbb{R}^+$

Define distance on distributions:

$$d^{\#}(\mu_1, \mu_2) \triangleq \min_{\mu \in C(\mu_1, \mu_2)} \mathbb{E}_\mu[d]$$

set of all couplings

# Ingredient $\#2$: Lift distance on outputs

## Given:

- Two distributions $\mu_1, \mu_2 \in$ Distr$(A)$
- Ground distance $d : A \times A \to \mathbb{R}^+$

## Define distance on distributions:

$$d^\#(\mu_1, \mu_2) \triangleq \min_{\boxed{\mu \in C(\mu_1, \mu_2)}} \mathbb{E}_\mu[d]$$

set of all couplings

## Typical pattern

Bound distance $d^\#$ between two (output) distributions by constructing a coupling with small average distance $d$

# Putting it together: Expected sensitivity

Given:

- A function $f : A \to \text{Distr}(B)$ (think: probabilistic program)
- Distances $d_{in}$ and $d_{out}$ on $A$ and $B$

# Putting it together: Expected sensitivity

Given:

- A function $f : A \to \text{Distr}(B)$ (think: probabilistic program)
- Distances $d_{in}$ and $d_{out}$ on $A$ and $B$

We say $f$ is $(d_{in}, d_{out})$-expected sensitive if:

$$d_{out}^{\#}(f(a_1), f(a_2)) \leq d_{in}(a_1, a_2)$$

for all inputs $a_1, a_2 \in A$.

# Benefits: Expressive

If $d_{out}(b_1, b_2) > k$ for all distinct $b_1, b_2$:

$(d_{in}, d_{out})$-expected sensitive $\implies$ $k$-uniform sensitive

# Benefits: Expressive

If $d_{out}(b_1, b_2) > k$ for all distinct $b_1, b_2$:

$(d_{in}, d_{out})$-expected sensitive $\implies$ $k$-uniform sensitive

If outputs are real-valued and $d_{out}(b_1, b_2) = k \cdot |b_1 - b_2|$:

$(d_{in}, d_{out})$-expected sensitive $\implies$ $k$-mean sensitive

# Benefits: Easy to reason about

# Benefits: Easy to reason about

$f : A \to \text{Distr}(B)$ is $(d_A, d_B)$-expected sensitive

# Benefits: Easy to reason about

$f : A \to \mathsf{Distr}(B)$ is $(d_A, d_B)$-expected sensitive
$g : B \to \mathsf{Distr}(C)$ is $(d_B, d_C)$-expected sensitive

# Benefits: Easy to reason about

$$\frac{\begin{array}{c} f : A \to \text{Distr}(B) \text{ is } (d_A, d_B)\text{-expected sensitive} \\ g : B \to \text{Distr}(C) \text{ is } (d_B, d_C)\text{-expected sensitive} \end{array}}{g \,\tilde{\circ}\, f : A \to \text{Distr}(C) \text{ is } (d_A, d_C)\text{-expected sensitive}}$$

# Benefits: Easy to reason about

$$\frac{\begin{array}{c} f : A \to \mathsf{Distr}(B) \text{ is } (d_A, d_B)\text{-expected sensitive} \\ g : B \to \mathsf{Distr}(C) \text{ is } (d_B, d_C)\text{-expected sensitive} \end{array}}{g \,\tilde{\circ}\, f : A \to \mathsf{Distr}(C) \text{ is } (d_A, d_C)\text{-expected sensitive}}$$

Abstract away distributions

- Work in terms of distances on ground sets
- No need to work with complex distances over distributions

# How to verify this property?

## The program logic $\mathbb{E}$pRHL

# A relational program logic $\mathbb{E}$pRHL

## The pWhile imperative language

$$c ::= x \leftarrow e \mid x \xleftarrow{\$} d \mid \text{if } e \text{ then } c \text{ else } c \mid \text{while } e \text{ do } c \mid \text{skip} \mid c;\ c$$

# A relational program logic $\mathbb{E}$pRHL

## The pWhile imperative language

$$c ::= x \leftarrow e \mid \boxed{x \xleftarrow{\$} d} \mid \text{if } e \text{ then } c \text{ else } c \mid \text{while } e \text{ do } c \mid \text{skip} \mid c;\, c$$

# A relational program logic $\mathbb{E}$pRHL

## The pWhile imperative language

$$c ::= x \leftarrow e \mid x \xleftarrow{\$} d \mid \text{if } e \text{ then } c \text{ else } c \mid \text{while } e \text{ do } c \mid \text{skip} \mid c; c$$

## Judgments

$$\vdash \{P; d_{in}\} \; c_1 \sim c_2 \; \{Q; d_{out}\}$$

- Tagged program variables: $x\langle 1\rangle$, $x\langle 2\rangle$
- $P$ and $Q$: boolean predicates over tagged variables
- $d_{in}$ and $d_{out}$: real-valued expressions over tagged variables

# $\mathbb{E}$pRHL judgments model expected sensitivity

A judgment

$$\vdash \{P; d_{in}\}\ \ c_1 \sim c_2\ \ \{Q; d_{out}\}$$

is valid if:

for all input memories $(m_1, m_2)$ satisfying pre-condition $P$, there exists a coupling of outputs $(\llbracket c_1 \rrbracket m_1, \llbracket c_2 \rrbracket m_2)$ with

- support satisfying post-condition $Q$
- $\mathbb{E}[d_{out}] \leq d_{in}(m_1, m_2)$

# One proof rule: Sequential composition

$$\frac{\vdash \{P; d_A\}\ \ c_1 \sim c_2\ \ \{Q; d_B\} \qquad \vdash \{Q; d_B\}\ \ c_1' \sim c_2'\ \ \{R; d_C\}}{\vdash \{P; d_A\}\ \ c_1; c_1' \sim c_2; c_2'\ \ \{R; d_C\}}$$

## One proof rule: Sequential composition

$$\frac{\vdash \{P; d_A\}\ c_1 \sim c_2\ \{Q; d_B\} \qquad \vdash \{Q; d_B\}\ c_1' \sim c_2'\ \{R; d_C\}}{\vdash \{P; d_A\}\ c_1; c_1' \sim c_2; c_2'\ \{R; d_C\}}$$

# One proof rule: Sequential composition

$$\frac{\vdash \{P; d_A\}\ c_1 \sim c_2\ \{Q; d_B\} \qquad \vdash \{Q; d_B\}\ c_1' \sim c_2'\ \{R; d_C\}}{\vdash \{P; d_A\}\ c_1; c_1' \sim c_2; c_2'\ \{R; d_C\}}$$

# One proof rule: Sequential composition

$$\frac{\vdash \{P; d_A\}\ c_1 \sim c_2\ \{Q; d_B\} \qquad \vdash \{Q; d_B\}\ c_1' \sim c_2'\ \{R; d_C\}}{\vdash \{P; d_A\}\ c_1; c_1' \sim c_2; c_2'\ \{R; d_C\}}$$

## One proof rule: Sequential composition

$$\frac{\vdash \{P; d_A\} \;\; c_1 \sim c_2 \;\; \{Q; d_B\} \qquad \vdash \{Q; d_B\} \;\; c_1' \sim c_2' \;\; \{R; d_C\}}{\vdash \{P; d_A\} \;\; c_1; c_1' \sim c_2; c_2' \;\; \{R; d_C\}}$$

$\boxed{\text{Expected sensitivity composes}}$

# Wrapping up

# More in the paper

## Theoretical results

- Full proof system (sampling, conditionals, loops, etc.)
- Transitivity principle (internalizes path coupling)

## Implementation in EasyCrypt, formalizations of:

- Stability for the Stochastic Gradient Method
- Convergence for the RSM population dynamics
- Mixing for the Glauber dynamics

# Looking forward

## Possible directions

- Other useful consequences of expected sensitivity?
- Formal verification systems beyond program logics?
- How to automate this proof technique?

# Looking forward

## Possible directions

- Other useful consequences of expected sensitivity?
- Formal verification systems beyond program logics?
- How to automate this proof technique?

Shameless plug: Looking for students at UWisconsin!

# Proving Expected Sensitivity of Probabilistic Programs

Gilles Barthe
Thomas Espitau
Benjamin Grégoire
Justin Hsu
Pierre-Yves Strub

# Our contributions

- Coupling-based definition of probabilistic sensitivity
- Relational program logic $\mathbb{E}$PRHL
- Formalized examples: stability and convergence