

Probability theory can be fun and simple with dependent types (Yet another formal theory of probabilities in Coq)

Reynald Affeldt, Alessandro Bruni, Pierre Roux, Takafumi Saikawa

30th International Conference on Types for Proofs and Programs
10 - 14 June 2024

An overview of existing formalizations of probabilities in Coq ¹

InfoTheo (2009–ongoing)


- Formalizes *finite probabilities*; used for information theory [Affeldt et al., 2014], error-correcting codes [Affeldt et al., 2020a], robust statistics [Affeldt et al., 2024a]

coq-proba [Tassarotti, 2023]

- Used to verify a compiler for probabilistic programming languages [Tassarotti and Tristan, 2023]

FormalML [The FormalML development team, 2023]

- Contains *advanced theorems* in probability theory, e.g., a stochastic approximation theorem [Vajjha et al., 2022]

¹ISABELLE/HOL and MATHLIB have extensive libraries for probabilities, this talk focuses on Coq 

A proof engineering effort



Mathematical Components

29 followers <https://math-comp.github.io/math-...>

math-comp

Public

Mathematical Components

Coq ☆ 541 🍷 109 🕒 98 📄 34 Updated 7 minutes ago



analysis

Public

Mathematical Components compliant Analysis Library

Coq ☆ 176 🍷 40 🕒 72 (1 issue needs help) 📄 38 Updated 2 hours ago



real-closed

Public

Theorems for Real Closed Fields

Coq ☆ 12 🍷 10 🕒 5 📄 3 Updated 5 hours ago



hierarchy-builder

Public

High level commands to declare a hierarchy based on packed classes

Prolog ☆ 90 🎓 MIT 🍷 19 🕒 64 📄 13 Updated 10 hours ago



Applications of MathComp-Analysis to probabilities?

MathComp-Analysis timeline

- Asymptotic reasoning + Landau notations \rightarrow differentiability [Affeldt et al., 2018]
- Lebesgue integral [Affeldt and Cohen, 2023]
- Fundamental theorem of calculus [Affeldt and Stone, 2024]
- Probability theory (2023–ongoing)

Applications to probabilities

- Verified probabilistic programming languages [CPP 2023, APLAS 2023]
- Verified worst-case failure probability of real-time systems [Markovic et al., 2023]

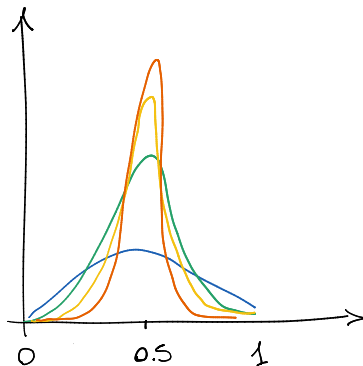
Other planned applications

- Verified robust statistics [Daukantas et al., 2021, Affeldt et al., 2024a]
- Verified machine learning [Affeldt et al., 2024b]

An example: Bernoulli sampling [Rajani, 2019]

Bernoulli sampling

Given n independent 0-1 random variables X_i , $p \in (0, 1]$, $\theta \in (0, p)$, $\delta \in (0, 1]$ with $Pr(X_i = 1) = p$, $X = \sum_{i=1}^n X_i$, and $\bar{X} = \frac{X}{n}$, then $Pr(|\bar{X} - p| \leq \theta) \geq 1 - \delta$ when $n \geq \frac{3}{\theta^2} \ln(\frac{2}{\delta})$.

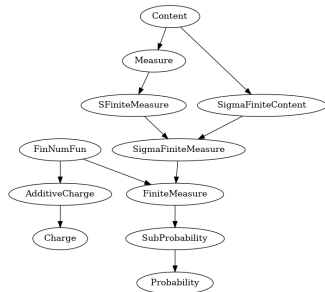


Simple and general: inherit from measure theory with Hierarchy Builder

Definition (Measure)

A **measure** $\mu : \mathcal{P}(T) \rightarrow \overline{\mathbb{R}}$ satisfies:

1. $\mu(\emptyset) = 0$ (measure-0)
2. $0 \leq \mu(A)$ for any set A (non-negativity)
3. $\mu(\bigcup_{i=1}^{\infty} A_i) = \sum_{i=1}^{\infty} \mu(A_i)$ (σ -additivity)

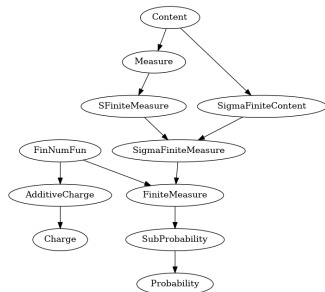


Simple and general: inherit from measure theory with Hierarchy Builder

Definition (Measure)

A **measure** $\mu : \mathcal{P}(T) \rightarrow \overline{\mathbb{R}}$ satisfies:

1. $\mu(\emptyset) = 0$ (measure-0)
2. $0 \leq \mu(A)$ for any set A (non-negativity)
3. $\mu(\bigcup_{i=1}^{\infty} A_i) = \sum_{i=1}^{\infty} \mu(A_i)$ (σ -additivity)



Definition (Probability measure)

A probability measure additionally implements the following interface:

```
HB.factory Record Measure_isProbability d (T : measurableType d)
  (R : realType) (P : set T -> \bar R) of isMeasure _ _ _ P :=
  { probability_setT : P setT = 1%E }.
```

...and fun: random variables and expectations



Context d (T : measurableType d) (R : realType) (P : probability T R).

Definition (Random variables)

A random variable is neither random, nor a variable. It's a measurable function from T to R.

Definition `random_variable` := {mfun T \rightarrow R}.

Notation "{ 'RV' P \rightarrow R }" := (@random_variable _ _ R P).

...and fun: random variables and expectations



Context d (T : measurableType d) (R : realType) (P : probability T R).

Definition (Random variables)

A random variable is neither random, nor a variable. It's a measurable function from T to R.

Definition `random_variable` := {mfun T >-> R}.

Notation "{ 'RV' P >-> R }" := (@random_variable _ _ R P).

Definition (Expectation)

Expectation of X with the measure P can be expressed as the Lebesgue integral $\int X dP$:

Definition `expectation` (X : {RV P >-> R}) := \int [P]_w (X w)%:E.

Recovering discreteness

Discrete (random) variables

Discrete random variables additionally implement the following interface:

```
HB.mixin Record MeasurableFun_isDiscrete d (T : measurableType d) (R : realType)
  (X : T -> R) of @MeasurableFun d T R X := { countable_range : countable (range X) }.
```

Recovering discreteness

Discrete (random) variables

Discrete random variables additionally implement the following interface:

```
HB.mixin Record MeasurableFun_isDiscrete d (T : measurableType d) (R : realType)
  (X : T -> R) of @MeasurableFun d T R X := { countable_range : countable (range X) }.
```

Discrete sums

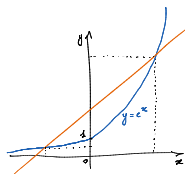
When $X : \{\text{dRV } P \rightarrow R\}$ (the type of discrete random variables), we build a function a_k to enumerate its values, and c_k to enumerate the probabilities, so that the distribution can be written as $\sum_k c_k \delta_{a_k}$:

```
Lemma distribution_dRV A : measurable A ->
  distribution P X A = \sum_(k <oo) c X k * \d_(a X k) A.
```

(More) formal adventures in convex spaces

[Affeldt et al., 2020b] shows that probability theory benefits from a theory of *convex spaces*.

We are porting it to MathComp-Analysis to define convex functions:



Convex function

```
Definition convex_function (R : realType) (D : set R) (f : R -> R) :=  
  forall t : {i01 R}, {in D &, forall (x y : R), f (x <| t |> y) <= f x <| t |> f y}.
```

Exponentials are convex

```
Lemma convex_expR : convex_function setT expR.
```

```
Lemma convex_powR p : 1 <= p -> convex_function `[0, +oo[ (fun x : R => powR x p).
```

Moments: exponential expectations

```
Definition mmt_gen_fun (X : {RV P -> R}) (t : R) := 'E_P[expR \o t \o* X].
```

Applications of convexity: Hölder and Minkowski and L_p -spaces

We are building a theory of L_p -spaces. For that purpose we prove Hölder's and Minkowski's inequalities, which are also generally applicable to probabilities:

Hölder

```
Lemma hoelder (f g : T -> R) (p q : R) : measurable_fun setT f -> measurable_fun setT g ->  
  0 < p -> 0 < q -> p^-1 + q^-1 = 1 (* Hoelder conjugates *) ->  
  'N_1 [f \* g] <= 'N_p [f] * 'N_q [g].
```

(Here $\backslash+$ and $\backslash*$ are pointwise addition and multiplication, and $N_p [f]$ is the p -norm of f)

Applications of convexity: Hölder and Minkowski and L_p -spaces

We are building a theory of L_p -spaces. For that purpose we prove Hölder's and Minkowski's inequalities, which are also generally applicable to probabilities:

Hölder

Lemma `hoelder` $(f\ g : T \rightarrow \mathbb{R})\ (p\ q : \mathbb{R}) : \text{measurable_fun setT } f \rightarrow \text{measurable_fun setT } g \rightarrow$
 $0 < p \rightarrow 0 < q \rightarrow p^{-1} + q^{-1} = 1$ (** Hölder conjugates **) \rightarrow
 $'N_p [f \ * \ g] \leq 'N_p [f] * 'N_q [g]$.

Minkowski

Lemma `minkowski` $f\ g\ p : \text{measurable_fun setT } f \rightarrow \text{measurable_fun setT } g \rightarrow 1 \leq p \rightarrow$
 $'N_p [f \ + \ g] \leq 'N_p [f] + 'N_p [g]$.

(Here $\backslash +$ and $\backslash *$ are pointwise addition and multiplication, and $N_p [f]$ is the p -norm of f)

More useful lemmas: Markov, Chernoff, Chebyshev and Cantelli

Lemma markov $(X : \{RV\ P \rightarrow R\}) (f : R \rightarrow R) (eps : R) : (0 < eps) \rightarrow$
measurable_fun [set: R] f \rightarrow (forall r, $0 \leq r \rightarrow 0 \leq f\ r$) \rightarrow
{in Num.nneg &, {homo f : x y / x \leq y}} \rightarrow
(f eps)%:E * P [set x | eps%:E \leq `| (X x)%:E |] \leq
'E_P[f \o (fun x => `| x |) \o X].

Lemma chernoff $(X : \{RV\ P \rightarrow R\}) (r a : R) : (0 < r) \rightarrow$
P [set x | X x \geq a] \leq mmt_gen_fun X r * (expR (- (r * a))):E.

Lemma chebyshev $(X : \{RV\ P \rightarrow R\}) (eps : R) : (0 < eps) \rightarrow$
P [set x | (eps \leq `| X x - fine ('E_P[X])|)] \leq (eps \wedge 2)%:E * 'V_P[X].

Lemma cantelli $(X : \{RV\ P \rightarrow R\}) (\lambda : R) :$
P.-integrable setT (EFin \o X) \rightarrow P.-integrable setT (EFin \o (X \wedge 2)) \rightarrow
(0 < lambda) \rightarrow
P [set x | lambda%:E \leq (X x)%:E - 'E_P[X]] \leq
(fine 'V_P[X] / (fine 'V_P[X] + lambda \wedge 2))%:E.

Our experiment (WIP): Bernoulli sampling [Rajani, 2019]

Theorem

Given n independent 0-1 random variables X_i , $p \in (0, 1]$, $\theta \in (0, p)$, $\delta \in (0, 1]$ with $Pr(X_i = 1) = p$, $X = \sum_{i=1}^n X_i$, and $\bar{X} = \frac{X}{n}$, then $Pr(|\bar{X} - p| \leq \theta) \geq 1 - \delta$ when $n \geq \frac{3}{\theta^2} \ln(\frac{2}{\delta})$.

becomes:

```
Theorem sampling (X : seq {RV P >-> R}) (theta delta p : R) :
  let n := size X in let X' x := ((\sum_(Xi in X) Xi) x) / n%:R in
  is_bernoulli_trial X n -> 0 < p <= 1 -> 0 < delta <= 1 ->
  0 < theta < p -> 0 < n -> 3 / theta^+2 * ln(2 / delta) <= n%:R
  -> P [set i | `| X' i - p | <= theta] >= 1 - delta%:E.
```


Conclusions

- We are generalizing Infotheo theories by porting them to MathComp-Analysis (future work: conditional probabilities, information theory, etc.)
- We are working on the verification of probabilistic programs by equational reasoning
- We aim to have a rich and general library that can be reused
- We are looking for contributors!

Bibliography I

- Reynald Affeldt and Cyril Cohen. Measure construction by extension in dependent type theory with application to integration. *J. Autom. Reason.*, 67(3):28, 2023. doi: 10.1007/s10817-023-09671-5. URL <https://doi.org/10.1007/s10817-023-09671-5>.
- Reynald Affeldt and Zachary Stone. A comprehensive overview of the lebesgue differentiation theorem in coq. 2024. To appear.
- Reynald Affeldt, Manabu Hagiwara, and Jonas Sénizergues. Formalization of Shannon’s theorems. *Journal of Automated Reasoning*, 53(1):63–103, 2014.
- Reynald Affeldt, Cyril Cohen, and Damien Rouhling. Formalization techniques for asymptotic reasoning in classical analysis. *Journal of Formalized Reasoning*, 11(1):43–76, 2018.
- Reynald Affeldt, Jacques Garrigue, and Takafumi Saikawa. A library for formalization of linear error-correcting codes. *J. Autom. Reason.*, 64(6):1123–1164, 2020a. doi: 10.1007/s10817-019-09538-8. URL <https://doi.org/10.1007/s10817-019-09538-8>.
- Reynald Affeldt, Jacques Garrigue, and Takafumi Saikawa. Formal adventures in convex and conical spaces. In *13th Conference on Intelligent Computer Mathematics (CICM 2020), Bertinoro, Forlì, Italy, July 26–31, 2020*, volume 12236 of *Lecture Notes in Artificial Intelligence*, pages 23–38. Springer, Jul 2020b. doi: 10.1007/978-3-030-53518-6_2.
- Reynald Affeldt, Alessandro Bruni, Clark Barrett, Ieva Daukantas, Harun Khan, Takafumi Saikawa, and Carsten Schürmann. Robust mean estimation by all means. 2024a. To appear.
- Reynald Affeldt, Alessandro Bruni, Ekaterina Komendantskaya, Natalia Ślusarz, and Kathrin Stark. Taming differentiable logics with coq formalisation. 2024b. To appear.

Bibliography II

- Ieva Daukantas, Alessandro Bruni, and Carsten Schürmann. Trimming data sets: a verified algorithm for robust mean estimation. In *23rd International Symposium on Principles and Practice of Declarative Programming (PPDP 2021), Tallinn, Estonia, September 6–8, 2021*, pages 17:1–17:9. ACM, 2021. doi: 10.1145/3479394.3479412. URL <https://doi.org/10.1145/3479394.3479412>.
- Filip Markovic, Pierre Roux, Sergey Bozhko, Alessandro V. Papadopoulos, and Björn B. Brandenburg. CTA: A correlation-tolerant analysis of the deadline-failure probability of dependent tasks. In *IEEE Real-Time Systems Symposium (RTSS 2023), Taipei, Taiwan, December 5–8, 2023*, pages 317–330. IEEE, 2023. doi: 10.1109/RTSS59052.2023.00035. URL <https://doi.org/10.1109/RTSS59052.2023.00035>.
- Samir Rajani. Applications of chernoff bounds.
<http://math.uchicago.edu/~may/REU2019/REUPapers/Rajani.pdf>, 2019. The University of Chicago Mathematics REU 2019.
- Joseph Tassarotti. A probability theory library for the Coq theorem prover.
<https://github.com/jtassarotti/coq-proba>, 2023. Since 2020.
- Joseph Tassarotti and Jean-Baptiste Tristan. Verified density compilation for a probabilistic programming language. *Proc. ACM Program. Lang.*, 7(PLDI):615–637, 2023. doi: 10.1145/3591245. URL <https://doi.org/10.1145/3591245>.
- The FormalML development team. FormalML: Formalization of machine learning theory with applications to program synthesis. <https://github.com/IBM/FormalML>, 2023. Since 2019.
- Koundinya Vajjha, Barry M. Trager, Avraham Shinnar, and Vasily Pestun. Formalization of a stochastic approximation theorem. In *ITP*, 2022. doi: 10.4230/LIPICS.ITP.2022.31.