Yet another formal theory of probabilities (with an application to random sampling)

Reynald Affeldt[1], Alessandro Bruni[2], Pierre Roux[3], and Takafumi Saikawa[4]

[1] National Institute of Advanced Industrial Science and Technology (AIST), Tokyo, Japan
[2] IT University of Copenhagen, Denmark
[3] ONERA/DTIS, Université de Toulouse, France
[4] Nagoya University, Japan

Abstract

There are already several formalizations of probability theory in the Coq proof assistant with applications to mathematics, information theory, and programming languages. They have been developed independently, do not cover the same ground, and a substantial effort is required to make them inter-operate. In this presentation, we report about an on-going effort in Coq to port and generalize a library about finite probabilities to a more generic formalization of real analysis called MathComp-Analysis. This gives us an opportunity to generalize results about convexity and probability and to enrich the library of probability inequalities. We explain our process of formalization and apply the resulting library to an original formalization of random sampling.

An overview of formalization of probabilities in Coq We know of several formalizations of probabilities in Coq[1] . InfoTheo is a formalization of finite probabilities that has been used to formalize information theory, error-correcting codes, and robust statistics (e.g., [5, 9]). Discrete probabilities has been formalized in coq-proba [18] and used to reason about programs (e.g., [10]). FormalML contains advanced theorems on probability theory [19, 20]. On the other hand, the MathComp-Analysis library, built on top of the Mathematical Components library [14], provides a rich formalization of measure theory and Lebesgue integral [2, 13]. In particular, MathComp-Analysis has been used to formalize probabilistic programming [3, 17].

Porting convexity results from InfoTheo to MathComp-Analysis We learn from InfoTheo that dealing with probabilities benefits from having a theory of *convex spaces*, to represent, among others, convex functions [6, Sect. 3.3]. A convex space is a mathematical structure with an operator written `a <| p |> b` (where `p` is a real number between 0 and 1) that expresses convex combination and a few axioms about this operator (skewed commutativity, quasi-associativity, etc.). Convex spaces are advantageously formalized using Hierarchy-Builder [8], a tool to build hierarchies of mathematical structures, see [12, `convex.v`]. The operator for convex combination is better handled with a dedicated type for real numbers between 0 and 1 (to represent the `p` in `a <| p |> b`), and InfoTheo provides such a specific type. On the other hand, MathComp-Analysis also had theories for positive and non-negative real numbers (i.e., real numbers in $]0,+\infty[$ and $[0,+\infty[$). We figured out that real numbers in $[0,1]$ can be handled similarly, thus providing a type `{i01 R}` to write convexity statements, e.g., [1, `convex.v`]:

```
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}.
```

Using convex spaces and convex functions from MathComp-Analysis, we have been able to port results from InfoTheo such as the convexity of the exponential function [1, `hoelder.v`]:

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

We are also planning to port more related results from InfoTheo such as conical spaces [4, Sect. 4].

Basic definitions of probability theory in MathComp-Analysis Probability measures come from basic definitions about measure theory. A measure μ satisfies the following: $\mu(\emptyset)=0$, $0\leq\mu(A)$ for any A, and σ-additivity: $\mu(\bigcup_{i=1}^{\infty}A_i)=\sum_{i=1}^{\infty}A_i$ for countably many pairwise disjoint A_i's [1, `measure.v`]. A probability measure extends a measure with the following interface (giving rise to a type `probability T R`):

[1]It should be noted that other proof assistants also provide substantial accounts of probability theory (in particular in Isabelle/HOL [11, 7]).

```
HB.mixin Record isProbability d (T : measurableType d) (R : realType) (P : set T -> \bar R) :=
  { probability_setT : P setT = 1 }. (* setT is the full set *)
```

The Lebesgue integral (noted `\int[mu]_(x in A) f x` [2, Sect. 6.4]) is used to formalize the notions of expectation, covariance, and variance [1, `probability.v`], e.g., for the expectation (noted `'E_P[X]`):

```
Definition expectation d (T : measurableType d) (R : realType) (P : probability T R)
  (X : T -> R) := \int[P]_w (X w)%:E. (* %:E turns real numbers into extended real numbers *)
```

Random variables are essentially measurable functions (noted `{mfun T >-> R}`). Like in InfoTheo, the probability measure `P` of the underlying space is encoded as a phantom type:

```
Definition random_variable d (T : measurableType d) (R : realType) (P : probability T R) :=
  {mfun T >-> R}.
Notation "{ 'RV' P >-> R }" := (@random_variable _ _ R P).
```

This way, when we write `{RV P >-> R}` for the type of a random variable, we understand that the underlying sample space is the one corresponding to the probability measure `P`.

We use Hierarchy-Builder and the theory of cardinality of MathComp-Analysis [1, `cardinality.v`] to extend the mathematical structure of random variables to the one of discrete random variables:

```
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) }.
```

Let `{dRV P >-> R}` be the type of discrete random variables. From a discrete random variable `X` we can derive a function `dRV_enum` to enumerate the values a_k it takes and a function `enum_prob` to enumerate the weigths c_k so that the distribution $P_{\mathtt{X}}$ of `X` can be written as a countable sum of Dirac measures $\sum_k c_k \delta_{a_k}$, eventually recovering the fact that the expectation of `X` is $\sum_k c_k a_k$ (using the properties of the Lebesgue integral):

```
Lemma distribution_dRV A : measurable A ->
  distribution P X A = \sum_(k <oo) enum_prob X k * \d_(dRV_enum X k) A. (* \d_ is for δ *)
```

The last bit of our basic setting of probability theory in MathComp-Analysis consists of the definition of L^p spaces. For that purpose, we prove Hölder's inequality:

```
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 ->
    'N_1 [f \* g] <= 'N_p [f] * 'N_q [g]. (* \* is the pointwise multiplication *)
```

The notation `'N_p[f]` denotes the L^p norm of `f`. This theorem relies on the formalization of convexity mentioned above. Cauchy-Schwarz's inequality is widely used in probability theory and is just a special case of Hoelder's where $p = q = 2$. Furthermore, Hoelder's inequality can be used to prove Minkowski's inequality:

```
Lemma minkowski f g p : measurable_fun setT f -> measurable_fun setT g -> 1 <= p ->
  'N_p%:E[f \+ g] <= 'N_p%:E[f] + 'N_p%:E[g]. (* \+ is the pointwise addition *)
```

This lemma shows that L^p spaces are normed vector spaces.

Recent and current work We further extend the above setup with fundamental inequalities such as Markov's, Chernoff's, Chebyshev's, and Cantelli's, etc. We are now working on defining precisely L^p spaces with MathComp's generic quotients. Our development has already been used in the verification of worst-case failure probability of real-time systems [15]. We are tackling the formalization of a sampling theorem [16, Theorem 3.1] which requires formalizing notions of random trials, independent random variables, and makes use of Chernoff's bound:

```
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.
```

References

[1] R. Affeldt, Y. Bertot, A. Bruni, C. Cohen, M. Kerjean, A. Mahboubi, D. Rouhling, P. Roux, K. Sakaguchi, Z. Stone, P.-Y. Strub, and L. Théry. MathComp-Analysis: Mathematical components compliant analysis library. https://github.com/math-comp/analysis, 2024. Since 2017. Version 1.0.0.

[2] R. Affeldt and C. Cohen. Measure construction by extension in dependent type theory with application to integration. *J. Autom. Reason.*, 67(3):28, 2023.

[3] R. Affeldt, C. Cohen, and A. Saito. Semantics of probabilistic programs using s-finite kernels in Coq. In *12th ACM SIGPLAN International Conference on Certified Programs and Proofs (CPP 2023), Boston, MA, USA, January 16–17, 2023*, pages 3–16. ACM, 2023.

[4] R. Affeldt, J. Garrigue, and T. Saikawa. Formal adventures in convex and conical spaces. In *13th Conference on Intelligent Computer Mathematics (CICM 2020), Bertinoro, Forli, Italy, July 26–31, 2020*, volume 12236 of *Lecture Notes in Artificial Intelligence*, pages 23–38. Springer, Jul 2020.

[5] R. Affeldt, J. Garrigue, and T. Saikawa. A library for formalization of linear error-correcting codes. *J. Autom. Reason.*, 64(6):1123–1164, 2020.

[6] R. Affeldt, J. Garrigue, and T. Saikawa. Reasoning with conditional probabilities and joint distributions in Coq. *Computer Software*, 37(3):79–95, 2020.

[7] J. Avigad, J. Hölzl, and L. Serafin. A formally verified proof of the central limit theorem. *J. Autom. Reason.*, 59(4):389–423, 2017.

[8] C. Cohen, K. Sakaguchi, and E. Tassi. Hierarchy builder: Algebraic hierarchies made easy in Coq with Elpi (system description). In *5th International Conference on Formal Structures for Computation and Deduction (FSCD 2020), June 29–July 6, 2020, Paris, France (Virtual Conference)*, volume 167 of *LIPIcs*, pages 34:1–34:21. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020. https://hal.inria.fr/hal-02478907/document.

[9] I. Daukantas, A. Bruni, and C. 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.

[10] S. O. Gregersen, A. Aguirre, P. Haselwarter, J. Tassarotti, and L. Birkedal. Asynchronous probabilistic couplings in higher-order separation logic, 2023.

[11] J. Hölzl. *Construction and stochastic applications of measure spaces in higher-order logic*. PhD thesis, Technical University Munich, 2013.

[12] Infotheo. Infotheo: A Coq formalization of information theory and linear error-correcting codes. https://github.com/affeldt-aist/infotheo, 2018. Authors: Reynald Affeldt, Manabu Hagiwara, Jonas Sénizergues, Jacques Garrigue, Kazuhiko Sakaguchi, Taku Asai, Takafumi Saikawa, and Naruomi Obata. Last stable release: 0.6.1 (2023).

[13] Y. Ishiguro and R. Affeldt. The Radon-Nikodým theorem and the Lebesgue-Stieltjes measure in Coq. *Computer Software*, 2024. To appear.

[14] A. Mahboubi and E. Tassi. *Mathematical Components*. Zenodo, Jan 2021. https://zenodo.org/record/7118596.

[15] F. Markovic, P. Roux, S. Bozhko, A. V. Papadopoulos, and B. 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.

[16] S. Rajani. Applications of chernoff bounds. http://math.uchicago.edu/~may/REU2019/REUPapers/Rajani.pdf, 2019. The University of Chicago Mathematics REU 2019.

[17] A. Saito and R. Affeldt. Experimenting with an intrinsically-typed probabilistic programming language in coq. In *21st Asian Symposium on Programming Languages and Systems (APLAS 2023), Taipei, Taiwan, November 26–29, 2023*, volume 14405 of *Lecture Notes in Computer Science*, pages 182–202. Springer, 2023.

[18] J. Tassarotti. A probability theory library for the Coq theorem prover. https://github.com/jtassarotti/coq-proba, 2023. Since 2020.

[19] The FormalML development team. FormalML: Formalization of machine learning theory with applications to program synthesis. https://github.com/IBM/FormalML, 2023. Since 2019.

[20] K. Vajjha, A. Shinnar, B. M. Trager, V. Pestun, and N. Fulton. Certrl: formalizing convergence proofs for value and policy iteration in coq. In *10th ACM SIGPLAN International Conference on Certified Programs and Proofs (CPP '21), Virtual Event, Denmark, January 17–19, 2021*, pages 18–31. ACM, 2021.