Probability theory in Mathlib

Rémy Degenne (Univ. Lille, Inria, CNRS, Centrale Lille, CRIStAL)

10/12/2025

A result from Mathlib

Azuma-Hoeffding inequality for sub-Gaussian random variables.

Let $(Y_i)_{i\in\mathbb{N}}$ be a sequence of real random variables adapted to a filtration $(\mathcal{F}_i)_{i\in\mathbb{N}}$. Suppose that Y_0 is sub-Gaussian with variance proxy c_0 and that for all $i \geq 0$, Y_{i+1} is conditionally sub-Gaussian with variance proxy c_{i+1} given $\mathcal{F}_i$.
That is, for all $\lambda \in \mathbb{R}$, the moment generating functions satisfy

$$\mathbb{E}\left[e^{\lambda Y_0}\right] \leq \exp\left(\frac{\lambda^2 c_0}{2}\right), \qquad \mathbb{E}\left[e^{\lambda Y_{i+1}} \mid \mathcal{F}_i\right] \leq \exp\left(\frac{\lambda^2 c_{i+1}}{2}\right).$$

Then, for all $n \in \mathbb{N}$ and all $\varepsilon \geq 0$,

$$\mathbb{P}\left(\sum_{i=0}^{n-1} Y_i \geq \varepsilon\right) \leq \exp\left(-\frac{\varepsilon^2}{2\sum_{i=0}^{n-1} c_i}\right).$$

A result from Mathlib

Azuma-Hoeffding inequality for sub-Gaussian random variables.

```
lemma measure_sum_ge_le_of_HasCondSubgaussianMGF
  {𝒳 : Type*} {m : MeasurableSpace 𝒳} [StandardBorelSpace 𝒳]
  {P : Measure 𝒳} [IsZeroOrProbabilityMeasure P]
  {Y : ℕ → 𝒳 → ℝ} {cY : ℕ → ℝ≥0} {F : Filtration ℕ m}
  (h_adapted : Adapted F Y) (h0 : HasSubgaussianMGF (Y 0) (c 0) P) (n : ℕ)
  (h_subG : ∀ i < n - 1, HasCondSubgaussianMGF (F i) (F.le i) (Y (i + 1)) (c (i +
    1)) P)
  {ε : ℝ} (hε : 0 ≤ ε) :
  P.real {ω | ε ≤ ∑ i ∈ Finset.range n, Y i ω}
    ≤ exp (-ε ^ 2 / (2 * ∑ i ∈ Finset.range n, c i)) :=
```

Inria

Table of Contents

1 Probability measures and random variables

Probability spaces

First, a type with a sigma-algebra: { $\mathcal{X}$: `Type*`} [`MeasurableSpace` $\mathcal{X}$]

A measurable set: {`s` : `Set` $\mathcal{X}$} (`hs` : `MeasurableSet s`)

A random variable: {`X` : $\mathcal{X} \to$ `E`} (`hX` : `Measurable X`)

Probability measures

A probability measure on $\mathcal{X}$: {`P` : `Measure` $\mathcal{X}$} [`IsProbabilityMeasure P`]
Other measure classes: `IsFiniteMeasure`, `SigmaFinite`, `SFinite`.

Don't use the `ProbabilityMeasure` type unless you are working with its topology.

Probability measures

A probability measure on $\mathcal{X}$: {`P : Measure` $\mathcal{X}$} [`IsProbabilityMeasure P`]
Other measure classes: `IsFiniteMeasure`, `SigmaFinite`, `SFinite`.

Don't use the `ProbabilityMeasure` type unless you are working with its topology.

A measure sends sets to $[0, +\infty]$ (`ENNReal`, or $\mathbb{R}_{\geq 0\infty}$).
(even non-measurable sets. Lemmas will require some measurability)

ENNReal is not well behaved: subtraction truncates at zero, `ring` does not apply.

Probability measures

A probability measure on $\mathcal{X}$: {`P` : `Measure` $\mathcal{X}$} [`IsProbabilityMeasure P`]
Other measure classes: `IsFiniteMeasure`, `SigmaFinite`, `SFinite`.

Don't use the `ProbabilityMeasure` type unless you are working with its topology.

A measure sends sets to $[0, +\infty]$ (`ENNReal`, or $\mathbb{R}_{\geq 0}\infty$).
(even non-measurable sets. Lemmas will require some measurability)

ENNReal is not well behaved: subtraction truncates at zero, `ring` does not apply.

But a probability measure is never infinite!?
`simp` knows about this. It can prove `P s` < ∞ automatically.
`finiteness` is a tactic that proves that things are finite.

Use `P.real` for a real-valued function on sets.

Measurability

There are several notions of measurability in Mathlib:

- `Measurable f`: f is measurable
- `AeMeasurable f P`: f is almost everywhere equal to a measurable function
- `StronglyMeasurable f`: f is a limit of measurable simple functions
- `AEStronglyMeasurable f P`: f is almost everywhere equal to a strongly measurable function

Tactic to prove measurability of functions: `fun_prop`
Try it on any measurability goal!

If it fails, it's probably because one of two things:

- the measurability depends on a set being measurable $\rightarrow$ prove that first, with the `measurability` tactic
- there is something mathematically interesting going on

Discrete probability spaces

In a `DiscreteMeasurableSpace`, all sets are measurable.
All functions from discrete spaces are measurable.
Typically, finite or countable types.

- Mathlib has measure-theoretic probability, but not much about its specialization to discrete spaces.
- You will be asked to prove measurability of functions from/to discrete spaces.
- `fun_prop` is the answer.

Integrals

Two integrals in Mathlib:

- $\int^-$ `x`, `f x` ∂`P`: Lebesgue integral of `f` : $\mathcal{X} \to \mathbb{R}\geq 0\infty$ with respect to measure P
- $\int$ `x`, `f x` ∂`P` or `P[f]`: Bochner integral of `f` : $\mathcal{X} \to$ `E`, where E is a normed space. Default value 0 if f is not integrable. `Integrable f P`.

Strong measurability is important for Bochner integrals.

Tip: use the Lebesgue integral whenever possible. The Bochner integral generates many integrability side goals.

Integrable $=$ a.e. strongly measurable $+$ Lebesgue integral of extended norm (with value in $\mathbb{R}\geq 0\infty$) finite.

See also `MemLp` for L^p spaces.

Almost surely

$x \in S$ almost surely iff $P(S^c) = 0$.
In Mathlib, `∀ᵐ x ∂P, x ∈ S`
Other example: `∀ᵐ x ∂P, f x < g x`
Expressed with filters. Almost everywhere filter of a measure: filter of co-null sets.

`filter_upwards` tactic.

Context: `hp : ∀ᵐ x ∂P, p x`, `hpq : ∀ᵐ x ∂P, p x → q x`
Goal: `∀ᵐ x ∂P, q x`

After using the `filter_upwards [hp, hpq] with x hpx hpqx` tactic:

Context: `x`, `hpx : p x`, `hpqx : p x → q x`
Goal: `q x`

Inria

Table of Contents

2 Distributions, moments

Moments

Integral, variance.
Covariance(s):

- of two real random variables
- as a bilinear form in an inner product space
- as a bilinear form on the dual of a normed space

Characteristic function(s):

- charFunDual in a normed space: `charFunDual` μ `L` = $\int$ `v, exp (L v * I)` $\partial\mu$
- charFun in an inner product space

Moment generating function $t \mapsto P\left[e^{tX}\right]$, cumulant generating function (log of the MGF).
$\rightarrow$ defined on an interval, analytic.

Gaussian distributions

- In $\mathbb{R}$: `gaussianReal m v`, with mean `m` : $\mathbb{R}$ and variance `v` : $\mathbb{R}_{\geq 0}$
- In normed spaces: `IsGaussian` μ. All maps by continuous linear forms have a Gaussian distribution.
- No multivariate Gaussian in $\mathbb{R}^n$ yet (coming soon).

Some things Mathlib knows about Gaussians:

- Characteristic function
- Fernique's theorem: Gaussians in Banach spaces have exponential tails
- Moments are finite (consequence of Fernique's theorem)
- Many transforms of Gaussians are Gaussians

Gaussian distributions

- In $\mathbb{R}$: `gaussianReal m v`, with mean `m` : $\mathbb{R}$ and variance `v` : $\mathbb{R}_{\geq 0}$
- In normed spaces: `IsGaussian` μ. All maps by continuous linear forms have a Gaussian distribution.
- No multivariate Gaussian in $\mathbb{R}^n$ yet (coming soon).

Some things Mathlib knows about Gaussians:

- Characteristic function
- Fernique's theorem: Gaussians in Banach spaces have exponential tails
- Moments are finite (consequence of Fernique's theorem)
- Many transforms of Gaussians are Gaussians

Other distributions are defined, but barely used. So they might need work.
An unused definition is probably unusable

Table of Contents

3 Markov kernels and conditional distributions

Markov kernels

Probability transition kernel from $\mathcal{X}$ to $\mathcal{Y}$: κ : `Kernel` $\mathcal{X}$ $\mathcal{Y}$
A measurable map from $\mathcal{X}$ to measures on $\mathcal{Y}$.
That is: for all measurable sets B of $\mathcal{Y}$, the map $x \mapsto \kappa(x)(B)$ is measurable.

If all measures are probability measures: [`IsMarkovKernel` κ]

Markov kernels

Probability transition kernel from $\mathcal{X}$ to $\mathcal{Y}$: κ : `Kernel` $\mathcal{X}$ $\mathcal{Y}$
A measurable map from $\mathcal{X}$ to measures on $\mathcal{Y}$.
That is: for all measurable sets B of $\mathcal{Y}$, the map $x \mapsto \kappa(x)(B)$ is measurable.

If all measures are probability measures: [`IsMarkovKernel` κ]

Operations on kernels and measures:

- Composition of kernels, of a kernel and a measure
  $\int_{p \in \mathcal{X} \times \mathcal{Y}} f(p) d(\kappa \circ_m P) = \int_{x \in \mathcal{X}} \int_{y \in \mathcal{Y}} f(x, y) d\kappa(x) dP$
- Products of kernels
- "Composition-products" of kernels: getting a joint distribution from a marginal and a conditional distribution
  From κ : *Kernel* $\mathcal{X}$ $\mathcal{Y}$ and η : *Kernel* $(\mathcal{X} \times \mathcal{Y})$ $\mathcal{Z}$, get $\kappa \otimes_k \eta$: *Kernel* $\mathcal{X}$ $(\mathcal{Y} \times \mathcal{Z})$

Important theorem: disintegration of kernels

A kernel κ from $\mathcal{X}$ to $\mathcal{Y} \times \mathcal{Z}$ can be written as $\kappa_X \otimes_K \eta$, where κ_X is a kernel from $\mathcal{X}$ to $\mathcal{Y}$ and η is a kernel from $\mathcal{X} \times \mathcal{Y}$ to $\mathcal{Z}$.

Assumptions: $\mathcal{Z}$ standard Borel, and either $\mathcal{X}$ countable or $\mathcal{Y}$ countably generated.

Application to measures: a measure on $\mathcal{X} \times \mathcal{Y}$ can be disintegrated into its marginal on $\mathcal{X}$ and a kernel from $\mathcal{X}$ to $\mathcal{Y}$.

Conditionals and posteriors

`condDistrib Y X P` : `Kernel` $\mathcal{X}$ $\mathcal{Y}$
Conditional probability distribution of random variable `Y` given `X` under measure `P`. Obtained by disintegration of the joint distribution of `(X, Y)` on $\mathcal{X} \times \mathcal{Y}$.

Posterior kernel: `P` : `Measure` $\mathcal{X}$ and κ : `Kernel` $\mathcal{X}$ $\mathcal{Y}$ combine into `P` $\otimes_m$ κ : `Measure` ($\mathcal{X} \times \mathcal{Y}$). It disintegrates into a measure on $\mathcal{Y}$ and a kernel from $\mathcal{Y}$ to $\mathcal{X}$: the posterior kernel.

A form of Bayes' theorem

```
lemma posterior_eq_withDensity (h_ac : ∀ᵐ ω ∂P, κ ω ≪ κ ∘ₘ P) :
  ∀ᵐ x ∂(κ ∘ₘ P), (κ†P) x = P.withDensity (fun ω ↦ κ.rnDeriv (Kernel.const _ (κ
    ∘ₘ P)) ω x) := by
```

Independence

(Conditional) independence of random variables, sigma-algebras, sets.

A particularity: derived from independence with respect to a kernel and a measure.

Definition (Independence with respect to a kernel and a measure)

Let P be a measure on $\mathcal{X}$, and κ be a kernel from $\mathcal{X}$ to $\mathcal{Y}$. Let X and Y be two random variables on $\mathcal{Y}$. Then X and Y are independent with respect to κ and P if for all measurable sets s, t of $\mathcal{Y}$, for P-almost all x,

$$\kappa(x)(X^{-1}(s) \cap Y^{-1}(t)) = \kappa(x)(X^{-1}(s)) \cdot \kappa(x)(Y^{-1}(t)) \,.$$

- Apply to the conditional kernel to get conditional independence
- Apply to a constant kernel to get usual independence

$\rightarrow$ same trick used for sub-Gaussianity.

Table of Contents

4 Stochastic processes and martingales

Stochastic processes

A stochastic process: a function $\mathtt{X} : \mathtt{T} \to \mathcal{X} \to \mathtt{E}$.
Add measurability assumptions when needed.

We have definitions of

- Filtrations: `F : Filtration T m`
- Adapted processes: `Adapted F X`
- Stopping times: `IsStoppingTime F τ`
- Martingales: `Martingale X F P`

```
def Martingale (X : T → 𝒳 → E) (F : Filtration T m0) (P : Measure 𝒳) : Prop :=
 Adapted F X ∧ ∀ i j, i ≤ j → P[X j|F i] =ᵐ[P] X i
```

Stopping times take values in `WithTop T`.

Definitions are general. But theorems are mostly for discrete time.

(Sub)martingale results

All in discrete time:

- Optional stopping theorem
- Optional sampling theorem
- Doob decomposition
- Doob's maximal inequality
- Doob's L^p inequality
- Martingale convergence theorems (a.s. and in Lp)

[Kexing Ying, RD, A formalization of Doob's martingale convergence theorems in mathlib, CPP 2023]

"in discrete time" $=$ index set is precisely $\mathbb{N}$.
Very restrictive. We are generalizing now:
`[LinearOrder T] [OrderBot T] [LocallyFiniteOrder T]`

We are working on continuous time in the Brownian motion project (talk on Friday).

Inria

Table of Contents

5 Other results in and outside Mathlib

Other results in and outside Mathlib

Also in Mathlib:

- Convergence of probability measures. Portmanteau theorem
- Strong law of large numbers

Not in Mathlib yet, but in PRs or other projects:

- Central limit theorem
- Cameron-Martin theorem
- Some decision theory and information theory (divergences)
- Kolmogorov continuity theorem
- Brownian motion in $\mathbb{R}$
- Continuous time martingales

In progress: stochastic integrals (talk on Friday).

How you can contribute

- Mathlib is missing many facts about the existing definitions
- You can join the Brownian motion project and work on stochastic calculus

We have enough basic probability definitions to start formalizing other fields

- Statistics
- Stochastic optimization
- Bayesian probability
- Information theory

How you can contribute

- Mathlib is missing many facts about the existing definitions
- You can join the Brownian motion project and work on stochastic calculus

We have enough basic probability definitions to start formalizing other fields

- Statistics
- Stochastic optimization
- Bayesian probability
- Information theory

I'm looking for 1 postdoc (probably soon 2 postdocs) to work on formalization of machine learning theory.