Notes on Jesse's δs

A Local Cow — 2021-02

Introduction

Sampling-based evaluation of integrals is common and useful, but naïve sampling suffers high variances when integrands have small support. An extreme instance occurs when we attempt to evaluate:

$$\frac{\partial}{\partial t}\int_{x\in(-\infty,t]} f(x)dx \tag{1}$$

The answer is $f(1)$; can we see this by sampling?

To evaluate (1) by sampling, we re-express it as an integral. Our strategy is to move the integral sign's t-dependency into the integrand and thereupon to swap the derivative and integral. Indeed, when f is sufficiently smooth and bounded, the physicist's formalism of step "functions" Θ and dirac "functions" δ permits this strategy:

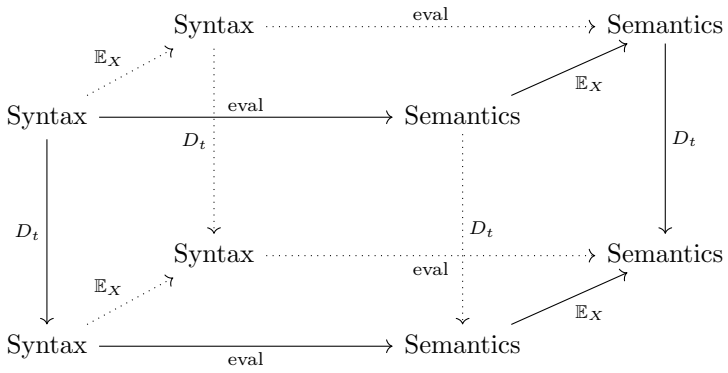
$$\begin{aligned}
&\frac{\partial}{\partial t}\int_{x\in(\infty,t]} f(x)dx \\
&= \frac{\partial}{\partial t}\int_{x\in\mathbb{R}} \Theta(x-0)\Theta(t-x)f(x)dx \\
&= \int_{x\in\mathbb{R}} \frac{\partial}{\partial t}[\Theta(x-0)\Theta(t-x)f(x)]dx \\
&= \int_{x\in\mathbb{R}} \Theta(x-0)\delta(t-x)f(x)dx
\end{aligned}$$

Intuitively, $\Theta(x-0)\delta(t-x)f(x)$ is infinite at $x=t$ and 0 elsewhere. Thus, a naïve sampling approach will fail. We'll show how the same δ function syntax that helps us translate (1) to a sampling problem also helps us us target our sampler to achieve finite variances.

Syntax and Semantics of Distributions

Here is an appropriate map from syntax to semantics for Jesse's variational-inference-through-integrals project. The hope is for automatic differentiation to

be correct, i.e. for the following cube to commute:



In what follows, we will name the set of syntax trees $\mathcal{S}$ and the semantic set (into which those trees are evaluated) $\mathfrak{D}$. The diagram makes clear that symbols such as D_t are overloaded: they act both on $\mathcal{S}$ and on $\mathfrak{D}$. In both cases, the intuition is that D_t differentiates with respect to t and that $\mathbb{E}_X$ integrates with respect to some probability density ρ on $(x, y, \cdots z) \in \mathbb{R}^d = X$. For concision, we assume that $\{t\} \sqcup \{x, y, \cdots z\}$ partitions the set of used variables, that we differentiate always with respect to t, and that we integrate always over X. To focus on applications, we restrict our attention to the non-dotted rectangle. We shall discuss its nodes, edges, and faces, i.e. define the worlds $\mathcal{S}$ of syntax and $\mathfrak{D}$ of semantics, construct the maps D_t and $\mathbb{E}_X$ and eval, and check that the rectangle commutes.

Syntax

Grammar — Because we focus on the non-dotted rectangle, it suffices to describe the syntax of integrands. Our integrand language includes arithmetic and (differentiated) step functions. For concreteness, say we have ring operations, real numbers, real-type variable names, and for each natural d the dth derivative $\Theta^{(d)}$ of the step function. The grammar has base types VAR, SMOOTH (for smooth expressions), and DISTR with natural "unit inclusions" VAR $\hookrightarrow$ SMOOTH and SMOOTH $\hookrightarrow$ DISTR; we regard the distributions as forming an algebra over the ring of smooth expressions, and we overload the ring operation symbols accordingly. Syntactically, $\Theta^{(d)}$ has the type SMOOTH $\to$ DISTR.

An example smooth expression is $y^2 - tx^3 - t$. An example distribution is:

$$\Theta^{(0)}(tx - y) \cdot \Theta^{(0)}(1 - tx - ty) \cdot \Theta^{(0)}(y) \cdot \Theta^{(0)}(1 - t/2 - y) \cdot x^2 y$$

Other standards conventions write $[f < g]$ for $\Theta^{(0)}(g - f)$ and δ for $\Theta^{(1)}$.

Algebraic Structure — Are $(x + y)$ and $(y + x)$ the same syntax trees? With the structure presented so far, the two differ. However, it is conceptually useful to ignore such distinctions. We thus redefine the set of syntax trees as the set of productions of the aforementioned grammar, *mod the axioms for*

real vector spaces. This makes the set of syntax trees into a real vector space. Going further, letting $\mathfrak{P}$ denote the ring of polynomials on $t, x, y, \cdots, z$ with real coefficients, we consider the syntax trees as forming a $\mathfrak{P}$-module called $\mathcal{S}$

Differentiation — The operator D_t transforms syntax trees to syntax trees by application of the product rule, of linearity, of the relation

$$D_t[\mathcal{V}] \triangleq 1 \text{ if } \mathcal{V} = t \text{ else } 0$$

for VAR node $\mathcal{V}$, and of the chain rule for $\Theta^{(d)}$ with SMOOTH tree $\mathcal{E}$:

$$D_t[\Theta^{(d)}](\mathcal{E})] \triangleq \Theta^{(d+1)}(\mathcal{E}) \cdot D_t[\mathcal{E}]$$

By induction, D_t's action on integrand syntax trees is thus fully determined. Moreover, it is a standard check (in the algebraic theory of *derivations*) that this map is well-defined despite the aforementioned module structure.

Semantics

Construction — How are we to interpret our integrand language? We are familiar with the ring $\mathfrak{R}$ of smooth functions on variables $t, x, y, \cdots, z$ whose every dth derivative ($0 \leq d$) is polynomially bounded. The interpretation of syntax trees of type SMOOTH into $\mathfrak{R}$ is standard. We now construct an $\mathfrak{R}$-module $\mathfrak{D}$ of *distributions* into which we shall interpret syntax trees of type DISTR.

Intuitively, evaluation sends a DISTR $\mathcal{D}$ to the integration-over-X operator

$$\text{eval}(\mathcal{D}) = (g \mapsto \mathbb{E}_X \mathcal{D} \cdot g)$$

To make sense of this, we formally define $\mathfrak{D}$ as the real vector space of continuous $\mathbb{R}$-linear functionals on $\mathfrak{G} \subseteq \mathfrak{R}$. Here, $\mathfrak{G}$ contains the smooth functions of $x, y, \cdots z$ whose every dth derivative ($0 \leq d$) is sub-gaussian.[1]

Algebraic Structure — The relation $(r \cdot d)(f) = d(r \cdot f)$ for $r, d, g \in \mathfrak{R}, \mathfrak{D}, \mathfrak{G}$ furnishes $\mathfrak{D}$ with the additional structure of an $\mathfrak{R}$-module. In particular, since $\mathfrak{R}$ contains the polynomials $\mathfrak{P}$, we may view $\mathfrak{D}$ as a $\mathfrak{P}$-module.

Differentiation and Integration — We define D_t's action on $\mathfrak{D}$ pointwise:

$$(D_t d)(g) \triangleq \frac{\partial d(g)}{\partial t}$$

We define $\mathbb{E}_X$'s action on $\mathfrak{D}$ by the simple rule:[2]

$$(\mathbb{E}_X d)(g) = d(\rho) \int_X g$$

[1] The specifics of sub-gaussianity are unimportant: we just want to avoid divergent integrals. We topologize $\mathfrak{G}$ by insisting that every ball $\{g \in G : \sup_X L(g) < r\}$ be open, where L ranges over polynomials of $x, y, \cdots, ; d_x, d_y, \cdots, d_z$ and r is positive.

[2] An annoying possibility is that $\rho \notin \mathfrak{G}$, in which case the previous equation is undefined. We may treat this general case by convolving with ever-narrowing gaussians and masking by ever-widening gaussians. We choose not to discuss this case.

Evaluation

Definition — We now define evaluation of syntactic integrands, i.e. the map $\mathrm{eval} : \mathcal{S} \to \mathfrak{D}$. Up to applications of the $\mathfrak{P}$-module axioms, the general form of a syntax tree is a polynomial of $\Theta^{(d)}$s with SMOOTH coefficients and arguments. As we know already how to evaluate SMOOTH expressions into $\mathfrak{P}$ and as we insist that evaluation be $\mathfrak{P}$-linear, it suffices to define evaluation on a generic monomial of $\Theta^{(d)}$s.

Well, we define[3]

$$\mathrm{eval}\left(\prod_i \Theta^{(d_i)}(f_i)\right) \triangleq \left(g \mapsto \lim_{\epsilon \to 0^+} \int_X g \cdot \prod_i \varphi^{(d_i)}(f_i/\epsilon)\right)$$

Here, φ is any smooth monotonic function equal to 0 for inputs below -1, equal to 1 for inputs above $+1$, and symmetrical in that $\phi(-x) + \phi(+x) = 1$.[4]

Examples — For convenience, we will write $\varphi_\epsilon(\cdot)$ for $\varphi(\cdot/\epsilon)$ and $\varphi_0(\cdot)$ for an actual step function from $\mathbb{R}$ to $\mathbb{R}$. Then one may check that

$$\mathrm{eval}\left(\Theta^{(0)}(f_0)\Theta^{(0)}(f_1)\Theta^{(0)}(f_d)\right) = \left(g \mapsto \int_X g \cdot \varphi_0(f_0)\varphi_0(f_1)\cdots\varphi_0(f_d)\right)$$

Likewise, for $0 \leq d$ one may check that

$$\mathrm{eval}\left(\left(\Theta^{(0)}(f)\right)^{d+1}\Theta^{(1)}(f)\right) = \mathrm{eval}\left(\Theta^{(0)}(f)\right)/(d+1)$$

As a final example, for coprime polynomials f_0, f_1 one may check that

$$\mathrm{eval}\left(\Theta^{(1)}(f_0 \cdot f_1)\Theta^{(0)}(f_0)\right) = \mathrm{eval}\left(\Theta^{(1)}(f_1)\Theta^{(0)}(f_0)\right) + \mathrm{eval}\left(\Theta^{(0)}(f_0)\right)/2$$

Correctness

of Automatic Differentiation — We check that evaluation and differentiation commute (front square of the cubical diagram), i.e. that:

$$\mathrm{eval} \circ D_t = D_t \circ \mathrm{eval}$$

Syntax trees are generated by: real linear combinations, VAR nodes, $\Theta^{(d)}(\cdot)$, and multiplication. To check commutativity, we induct on syntax trees so that we need only check commutativity for each of these generators.

The check for real linearity follows by eval's $\mathfrak{P}$-linearity. The check for VAR nodes follows from Leibniz's rule for indefinite integrals of smooth functions. The check for $\Theta^{(d)}(f)$ follows formally from the definitions of D_t and eval.

[3] This limit exists by the regularity conditions that define the space $\mathfrak{G}$ that g inhabits.

[4] The specific choice of φ doesn't matter; in fact, if we seek only first derivatives (so $D_t\mathbb{E}_X \cdots$ instead of $D_t^2\mathbb{E}_X \cdots$), φ doesn't even have to be smooth: it may linearly interpolate between $(-1, 0)$ and $(+1, 1)$. Still, it's important not to mix up φ_ϵ with $\Theta^{(0)}$: though $\lim_\epsilon \mathbb{E}_X \varphi(x/\epsilon)^2 g = \lim_\epsilon \mathbb{E}_X \varphi(x/\epsilon) g$ for $g \in \mathfrak{G}$, we have $\left(\Theta^{(0)}(x)\right)^2 \neq \Theta^{(0)}(x)$ as syntax trees.

Finally, we check commutativity on products. It suffices by D_t and eval's real linearity to check this on any real linear basis, for instance on the monomial products of fs and $\Theta^{(d)}(f)$s for polynomials f. We finish by writing out definitions. In order to avoid annoying notation, we illustrate this for a specific product; no new ideas are needed for the generic case.

$$\begin{aligned}
&(\mathrm{eval} \circ D_t) \left(\Theta^{(2)}(f) \cdot \Theta^{(3)}(h)\right)(g) \\
= &\lim_{\epsilon \to 0^+} \int_X g \cdot \varphi_\epsilon^{(3)}(f) f' \varphi_\epsilon^{(3)}(h) + \varphi_\epsilon^{(2)}(f) \varphi_\epsilon^{(4)}(h) h' \\
= &-\lim_{\epsilon \to 0^+} \int_X g' \cdot \Theta^{(2)}(f) \cdot \Theta^{(3)}(h) \\
= &\, (D_t \circ \mathrm{eval}) \left(\Theta^{(2)}(f) \cdot \Theta^{(3)}(h)\right)(g)
\end{aligned}$$

The middle equality applies integration by parts; the flanking ones are definitions.

of Switching Limits — We check that integration and differentiation commute. This follows formally:[5]

$$\begin{aligned}
&(D_t \circ \mathbb{E}_X)(d)(g) \\
= &\frac{\partial}{\partial t} \left(\int_X \rho \cdot \prod_i \varphi^{d_i}(f_i) \cdot \int_X g \right) \\
= &\left(\frac{\partial}{\partial t} \int_X \rho \prod_i \varphi^{d_i}(f_i) \right) \cdot \int_X g \\
= &\, (\mathbb{E}_X \circ D_t)(d)(g)
\end{aligned}$$

We used that $g \in \mathfrak{G}$ does not depend on t.

Unbiased Evaluation of Distributions

Example

[5] Remarkably, we need not invoke any theorems of calculus! In fact, this subsection follows for free since all the hard work was dome in defining things and in checking that D_t and eval commute.