Simplifying Debiased Inference via Automatic Differentiation and Probabilistic Programming

Alex Luedtke

Department of Health Care Policy
Harvard University

Background and our objective


Deriving the EIF with **automatic differentiation**


Implementing an estimator through **probabilistic programming**

Examples and experiments

Discussion

Background

Long history of constructing **semiparametric efficient estimators**

- (Levit, 1975; Hasminskii et al., 1979; Bickel, Klaassen, et al., 1993)

Many methods allow flexible estimation of nuisances:

- **One-step estimation** (Pfanzagl, 1982)
- **Estimating equations** (Newey, 1994)
- **Targeted learning** (van der Laan et al., 2006)
- **Double/debiased machine learning** (Chernozhukov et al., 2017)

One-step estimation (Le Cam, 1956; Pfanzagl, 1982)

One-step estimation often used to construct efficient estimators:

initial estimator + **debiasing term**

One-step estimation (Le Cam, 1956; Pfanzagl, 1982)

One-step estimation often used to construct efficient estimators:

$$\textbf{initial estimator} \; + \; \textbf{debiasing term}$$

In parametric models: update $\widehat{\beta}$ with one step of Newton's method, yielding

$$\widehat{\beta} \; + \; \frac{1}{n} \sum_{i=1}^{n} I_{\widehat{\beta}}^{-1} \dot{\ell}_{\widehat{\beta}}(Z_i)$$

One-step estimation (Le Cam, 1956; Pfanzagl, 1982)

One-step estimation often used to construct efficient estimators:

$$\textbf{initial estimator} \; + \; \textbf{debiasing term}$$

In parametric models: update $\widehat{\beta}$ with one step of Newton's method, yielding

$$\widehat{\beta} \; + \; \frac{1}{n} \sum_{i=1}^{n} I_{\widehat{\beta}}^{-1} \dot{\ell}_{\widehat{\beta}}(Z_i)$$

Key object for debiasing: the **efficient influence function (EIF)**

One-step estimation (Le Cam, 1956; Pfanzagl, 1982)

One-step estimation often used to construct efficient estimators:

$$\textbf{initial estimator} \; + \; \textbf{debiasing term}$$

In infinite-dimensional models: a similar approach works, but must:

- Construct initial estimator more flexibly
- Use a semiparametric or nonparametric EIF

$$\psi(\widehat{P}) \; + \; \frac{1}{n}\sum_{i=1}^{n} \widehat{f}(Z_i)$$

Running example: nonparametric R^2 (Wright, 1921; Williamson et al., 2021)

Observe n iid draws of $(X, Y) \sim P$

- X: Feature
- Y: Outcome

Aim to estimate nonparametric R^2 criterion:

$$\psi(P) = 1 - \frac{\int \{y - E_P(Y \mid X = x)\}^2 \, dP(x, y)}{\mathrm{Var}_P(Y)}$$

Running example: nonparametric R^2 (Wright, 1921; Williamson et al., 2021)

Observe n iid draws of $(X, Y) \sim P$

- X: Feature
- Y: Outcome

Aim to estimate nonparametric R^2 criterion:

$$\psi(P) = 1 - \frac{\int \{y - E_P(Y \mid X = x)\}^2 \, dP(x, y)}{\mathrm{Var}_P(Y)}$$

Even though the parameter is simple, computing the EIF is tedious!

Even though the parameter is simple, computing the EIF is tedious!

Williamson et al. (2021)

Proof of Lemma 1

For a given distribution $P \in \mathcal{M}$, we denote by p the density of P with respect to some dominating measure ν. For bounded $h \in L_2(P)$, we can define the parametric submodel $p_\epsilon = (1 + \epsilon h)p$, which is valid for small enough ϵ and has score h for ϵ at $\epsilon = 0$. Every regular parametric submodel centered at P and with score h is either of this form or can be approximated arbitrarily well by a submodel of this form. Given that the statistical model $\mathcal{M}$ considered is nonparametric, and that $\varphi_{P,\nu} \in L_2(P)$ with $P\varphi_{P,\nu} = 0$, if we show that for any $P \in \mathcal{M}$

$$\frac{\partial}{\partial \epsilon}\Theta_\nu(P_\epsilon)\Big|_{\epsilon=0} = \int \varphi_{P,\nu}(o)h(o)dP(o) \ ,$$

we will have established that $\Theta_\nu(P)$ is pathwise differentiable with respect to $\mathcal{M}$ at P with efficient influence function $\varphi_{P,\nu}$.

The evaluation of Θ_ν on the distribution P_ϵ corresponding to p_ϵ equals

$$\begin{aligned}\Theta_\nu(P_\epsilon) &= \int\int \{\mu_{P_\epsilon}(x) - \mu_{P_\epsilon,\nu}(x)\}^2 dP_\epsilon(z) \ = \ \int\int \alpha_{P_\epsilon,\nu}(x) dP_\epsilon(z) \\ &= \int\int \alpha_{P_\epsilon,\nu}(x)\{1 + \epsilon h(x,y)\}p(x,y)\nu(dx,dy) \\ &= \int\int \alpha_{P_\epsilon,\nu}(x)p(x,y)\nu(dx,dy) + \epsilon \int\int \alpha_{P_\epsilon,\nu}(x)h(x,y)p(x,y)\nu(dx,dy) \ ,\end{aligned}$$

where $\alpha_{P,\nu}(x) := \{\mu_{P,\nu}(x) - \mu_P(x)\}^2$, and so,

$$\frac{\partial}{\partial \epsilon}\Theta_\nu(P_\epsilon)\Big|_{\epsilon=0} = \int\int \frac{\partial}{\partial \epsilon}\alpha_{P,\nu}(x)\Big|_{\epsilon=0} p(x,y)\nu(dx,dy) + \int\int \alpha_{P,\nu}(x)h(x,y)p(x,y)\nu(dx,dy) \ ,$$

where $\alpha_{P,\nu} = \alpha_{P_\epsilon,\nu}|_{\epsilon=0}$. With a slight abuse of notation, we can write $\alpha_{P_\epsilon,\nu}(x)$ as

$$\alpha_{P_\epsilon,\nu}(x) = \left\{\frac{\int y\{1 + \epsilon h(x,y)\}p(x,y)\nu(dy)}{\int\{1 + \epsilon h(x,y)\}p(x,y)\nu(dy)} - \frac{\iint y\{1 + \epsilon h(x,y)\}p(x,y)\nu(dx_\nu,dy)}{\iint\{1 + \epsilon h(x,y)\}p(x,y)\nu(dx_\nu,dy)}\right\}^2$$

and so, we find that $\frac{\partial}{\partial \epsilon}\alpha_{P_\epsilon,\nu}(x)\big|_{\epsilon=0}$ equals

$$2\{\mu_P(x) - \mu_{P,\nu}(x)\}\left\{\frac{\int\{y - \mu_P(x)\}h(x,y)p(x,y)\nu(dy)}{\int p(x,y)\nu(dy)} - \frac{\iint\{y - \mu_{P,\nu}(x)\}h(x,y)p(x,y)\nu(dx_\nu,dy)}{\iint p(x,y)\nu(dx_\nu,dy)}\right\}.$$

This allows us to write that

$$\begin{aligned}\frac{\partial}{\partial \epsilon}\Theta_\nu(P_\epsilon)\Big|_{\epsilon=0} &= \int\int 2\{\mu_P(x) - \mu_{P,\nu}(x)\}\{y - \mu_P(x)\} + \alpha_{P,\nu}(x)\big]h(x,y)p(x,y)\nu(dx,dy) \\ &= \int\int 2\{\mu_P(x) - \mu_{P,\nu}(x)\}\{y - \mu_P(x)\} + \alpha_{P,\nu}(x) - \Theta_\nu(P)\big]h(x,y)p(x,y)\nu(dx,dy) \ .\end{aligned}$$

Because Ψ_ν is the ratio of two parameters, namely Θ_ν and the population outcome variance parameter, both of which are pathwise differentiable and have known efficient influence functions relative to nonparametric models, it follows that Ψ_ν is itself pathwise differentiable at each $P \in \mathcal{M}$. Furthermore, its efficient influence function can readily be found using the delta method. We will use the fact that the parameter $P \mapsto var_P(Y)$ has nonparametric efficient influence function given by $z \mapsto \tilde{\varphi}_P(z) := \{y - E_P(Y)\}^2 - var_P(Y)$. It follows then that the nonparametric efficient influence function of Ψ_ν at P equals

$$z \mapsto \varphi^*_{P,\nu}(z) \ = \ \frac{\varphi_{P,\nu}(z)var_P(Y) - \tilde{\varphi}_P(z)\Theta_\nu(P)}{var_P^2(Y)} \ ,$$

which simplifies algebraically to the form provided in the Lemma.

Even though the parameter is simple, computing the EIF is tedious!

Williamson et al. (2021)

Proof of Lemma 1

For a given distribution $P \in \mathcal{M}$, we denote by p the density of P with respect to some dominating measure ν. For bounded $h \in L_2(P)$, we can define the parametric submodel $p_\epsilon = (1 + \epsilon h)p$, which is valid for small enough ϵ and has score h for ϵ at $\epsilon = 0$. Every regular parametric submodel centered at P and with score h is either of this form or can be approximated arbitrarily well by a submodel of this form. Given that the statistical model $\mathcal{M}$ considered is nonparametric, and that $\varphi_{P,\epsilon} \in L_2(P)$ with $P\varphi_{P,\epsilon} = 0$, if we show that for any $P \in \mathcal{M}$

$$\frac{\partial}{\partial \epsilon}\Theta_\epsilon(P_\epsilon)\Big|_{\epsilon=0} = \int \varphi_{P,\epsilon}(o)h(o)dP(o) \ ,$$

we will have established that $\Theta_\epsilon(P)$ is pathwise differentiable with respect to $\mathcal{M}$ at P with efficient influence function $\varphi_{P,\epsilon}$.

The evaluation of Θ_ϵ on the distribution P_ϵ corresponding to p_ϵ equals

$$\begin{aligned}\Theta_\epsilon(P_\epsilon) &= \iint \{\mu_{P_\epsilon}(x) - \mu_{P_\epsilon,x}(x)\}^2 dP_\epsilon(z) \ = \ \iint \alpha_{P_\epsilon,x}(x)dP_\epsilon(z) \\ &= \iint \alpha_{P_\epsilon,x}(x)\{1 + \epsilon h(x,y)\}p(x,y)\nu(dx,dy) \\ &= \iint \alpha_{P_\epsilon,x}(x)p(x,y)\nu(dx,dy) + \epsilon \iint \alpha_{P_\epsilon,x}(x)h(x,y)p(x,y)\nu(dx,dy) \ ,\end{aligned}$$

where $\alpha_{P_\epsilon,x}(x) := \{\mu_{P_\epsilon,x}(x) - \mu_{P_\epsilon}(x)\}^2$, and so,

$$\frac{\partial}{\partial \epsilon}\Theta_\epsilon(P_\epsilon)\Big|_{\epsilon=0} = \iint \frac{\partial}{\partial \epsilon}\alpha_{P_\epsilon,x}(x)\Big|_{\epsilon=0} p(x,y)\nu(dx,dy) + \iint \alpha_{P,x}(x)h(x,y)p(x,y)\nu(dx,dy) \ ,$$

where $\alpha_{P,x} = \alpha_{P_\epsilon,x}|_{\epsilon=0}$. With a slight abuse of notation, we can write $\alpha_{P_\epsilon,x}(x)$ as

$$\alpha_{P_\epsilon,x}(x) = \left[\frac{\int y\{1 + \epsilon h(x,y)\}p(x,y)\nu(dy)}{\int \{1 + \epsilon h(x,y)\}p(x,y)\nu(dy)} - \frac{\iint y\{1 + \epsilon h(x,y)\}p(x,y)\nu(dx_*,dy)}{\iint \{1 + \epsilon h(x,y)\}p(x,y)\nu(dx_*,dy)}\right]^2$$

and so, we find that $\frac{\partial}{\partial \epsilon}\alpha_{P_\epsilon,x}(x)\big|_{\epsilon=0}$ equals

$$2\{\mu_P(x) - \mu_{P,x}(x)\}\left[\frac{\int\{y - \mu_P(x)\}h(x,y)p(x,y)\nu(dy)}{\int p(x,y)\nu(dy)} - \frac{\iint\{y - \mu_{P,x}(x)\}h(x,y)p(x,y)\nu(dx_*,dy)}{\iint p(x,y)\nu(dx_*,dy)}\right].$$

This allows us to write that

$$\begin{aligned}\frac{\partial}{\partial \epsilon}\Theta_\epsilon(P_\epsilon)\Big|_{\epsilon=0} &= \iint 2\{\mu_P(x) - \mu_{P,x}(x)\}\{y - \mu_P(x) + \alpha_{P,x}(x)\}h(x,y)p(x,y)\nu(dx,dy) \\ &= \iint 2\{\mu_P(x) - \mu_{P,x}(x)\}\{y - \mu_P(x)\} + \alpha_{P,x}(x) - \Theta_\epsilon(P)\big]h(x,y)p(x,y)\nu(dx,dy) \ .\end{aligned}$$

Because Ψ_ϵ is the ratio of two parameters, namely Θ_ϵ and the population outcome variance parameter, both of which are pathwise differentiable and have known efficient influence functions relative to nonparametric models, it follows that Ψ_ϵ is itself pathwise differentiable at each $P \in \mathcal{M}$. Furthermore, its efficient influence function can readily be found using the delta method. We will use the fact that the parameter $P \mapsto var_P(Y)$ has nonparametric efficient influence function given by $z \mapsto \tilde{\varphi}_P(z) := \{y - E_P(Y)\}^2 - var_P(Y)$. It follows then that the nonparametric efficient influence function of Ψ_ϵ at P equals

$$z \mapsto \varphi^*_{P,\epsilon}(z) \ = \ \frac{\varphi_{P,\epsilon}(z)var_P(Y) - \tilde{\varphi}_P(z)\Theta_\epsilon(P)}{var_P(Y)} \ ,$$

which simplifies algebraically to the form provided in the Lemma.

Li et al. (2023)

Proof. We prove this lemma by directly applying the definition of a gradient. We consider the one-dimensional submodel $\{P_\epsilon : |\epsilon| \leq 1\}$ with density

$$p_\epsilon(w, y) = p(y|w)\{1 + \epsilon s_1(y|w)\}p(w)\{1 + \epsilon s_2(w)\},$$

where the range of s_1 and s_2 falls in $[-1, 1]$ and these functions satisfy $E_P[s_1(Y|W)|W] = 0$ P-almost surely and $E_P[s_2|W] = 0$. Let $f_P(w) = E_P[u_P(Y)|W = w]$ and $f_{P_\epsilon}(w) = E_{P_\epsilon}[u_{P_\epsilon}(Y)|W = w]$. We have that

$$\begin{aligned}\sigma^2_f(P_\epsilon) = &\int \{u_{P_\epsilon}(y) - f_{P_\epsilon}(w)\}^2 \{1 + \epsilon s_1(y|w)\}\{1 + \epsilon s_2(w)\}dP(y|w)dP(w) \\ &- \int \{u_{P_\epsilon}(y) - f_{P_\epsilon}(w)\}^2 dP(w, y) \\ &+ \int \{u_{P_\epsilon}(y) - f_{P_\epsilon}(w)\}^2 \{\epsilon s_1(y|w) + \epsilon s_2(w) + \epsilon^2 s_1(y|w)s_2(w)\}dP(w, y).\end{aligned}$$

The second term on the right has the following derivative with respect to ϵ at $\epsilon = 0$

$$\int \{u_P(y) - f_P(w)\}^2 \{s_1(y|w) + s_2(w)\}dP(w, y),$$

and so will contribute $\{u_P(Y) - f_P(W)\}^2 - \sigma^2_f$ to the gradient. We now focus on the first term, which re-writes as

$$\begin{aligned}&\int \left[\int h(x, y)\{1 + \epsilon s_1(x|w)\}\{1 + \epsilon s_2(w)\}dP(x|w)dP(w) - f_{P_\epsilon}(w)\right]^2 dP(w, y) \\ &- \int \left[\left\{\int h(x, y)dP(w, x) - f_{P_\epsilon}(w)\right\}^2 + c(\epsilon)^2 + 2c(\epsilon)\left\{\int h(x, y)dP(w, x) - f_{P_\epsilon}(w)\right\}\right]dP,\end{aligned}$$

where $c(\epsilon) = \int h(x, y)\{\epsilon s_1(x|w) + \epsilon s_2(w) + \epsilon^2 s_1 s_2\}dP(w, x)$. The first term in the above display has derivative 0 with respect to ϵ at $\epsilon = 0$, as f_P is the true conditional mean. The second term also has derivative 0 at $\epsilon = 0$, as it is quadratic in ϵ. At $\epsilon = 0$, the third term has derivative

$$\begin{aligned}&2\int\left(\left[\int h(x, y)\{s_1(x|w) + s_2(w)\}dP(w, x)\right]\left\{\int h(x, y)dP(w, x) - f_P(w)\right\}\right)dP(w, y) \\ &- 2\int\left(\left\{\int h(x, y)dP(w, x) - f_P(w)\right\}[h(x, y)\{s_1(x|\tilde{w}) + s_2(\tilde{w})\}]\right)dP(\tilde{w}, x)dP(w, y) \\ &- 2\int\left[\left\{u_P(y) - f_P(w)\right\}h(x, y)dP(w, y)\right]\{s_1(x|\tilde{w}) + s_2(\tilde{w})\}dP(\tilde{w}, x).\end{aligned}$$

The inner integral has mean

$$\int\int \{u_P(y) - f_P(w)\}h(x, y)dP(w, y)dP(\tilde{w}, x) = \int\{u_P(y) - f_P(w)\}u_P(y)dP(w, y) = \sigma^2_f.$$

Therefore the following is a gradient:

$$D_f(w, y) = \{u_P(y) - f_P(w)\}^2 + 2\int\{u_P(\tilde{y}) - f_P(\tilde{w})\}h(y, \tilde{y})dP(\tilde{w}, \tilde{y}) - 3\sigma^2_f.$$

Since we are working within a locally nonparametric model, the above is also the canonical gradient. □

Form of EIF

Nonparametric EIF is

$$f(x,y) = \frac{[(y - E_P[Y])^2 - \sigma_P^2]\gamma_P}{\sigma_P^4} - \frac{(y - E_P[Y \mid X = x])^2 - \gamma_P}{\sigma_P^2},$$

where $\sigma_P^2 = \mathrm{Var}_P(Y)$ and $\gamma_P = E_P[\mathrm{Var}_P(Y \mid X)]$

Implementing a one-step estimator

To use an efficient estimator, it must be implemented in R/Python/etc.

- Should **estimate nuisances flexibly**
- **Cross-fitting** and other best practices should be used
- **Testing** and **debugging** generally needed

```
.sgtmle = function(W,A,Y,Delta,txs,Q.est,blip.est,g.est,k
  require(Hmisc)
  n = nrow(W)
  if(is.null(obsWeights)) obsWeights = rep(1,length(Y))
  obsWeights = obsWeights/mean(obsWeights)

  if(family$family=='binomial') {
    for(i in seq(txs)){Q.est[,i] = pmin(pmax(Q.est[,i],tr
    offs = family$linkfun(Reduce("+",lapply(seq(txs),func
      (A==txs[i])*Q.est[,i]
    }))))
    offs[!(A %in% txs)] = 0
  } else if(family$family=='gaussian'){
    offs = Reduce("+",lapply(seq(txs),function(i){(A==txs
  } else {
```

Example: code for Luedtke et al. (2017)

```
.sgtmle = function(W,A,Y,Delta,txs,Q.est,blip.est,g.est,baseline.probs,family,kappa=1,sig.trunc=0.1,alpha=0.05,trunc.Q=c(0.005,0.995),obsWeights=N
  require(Hmisc)
  n = nrow(W)
  if(is.null(obsWeights)) obsWeights = rep(1,length(Y))
  obsWeights = obsWeights/mean(obsWeights)

  if(family$family=='binomial') {
    for(i in seq(txs)){Q.est[,i] = pmin(pmax(Q.est[,i],trunc.Q[1]),trunc.Q[2])}
    offs = family$linkfun(Reduce("+",lapply(seq(txs),function(i){
      (A==txs[i])*Q.est[,i]
    })))
    offs[!(A %in% txs)] = 0
  } else if(family$family=='gaussian'){
    offs = Reduce("+",lapply(seq(txs),function(i){(A==txs[i])*Q.est[,i]}))
  } else {
    stop("family must be either binomial or gaussian.")
  }

  # treatment effect of assigning first treatment in txs vs assigning the next best treatment
  # useful when kappa<1 (so that there is a constraint on this first treatment resource)
  one.vs.next.best = blip.est[,1] - apply(blip.est[,-1,drop=FALSE],1,max)
  tau = wtd.quantile(one.vs.next.best, obsWeights, type="i/n", probs=1-kappa)
  if(mean((one.vs.next.best>tau) * obsWeights)/mean(obsWeights)>kappa){
    tau = min(one.vs.next.best[one.vs.next.best>tau])
  }

  # if one.vs.next.best<tau, then the individual will not be treated with tx 1 in the presence of the resource constraint
  blip.est[one.vs.next.best<tau,1] = -Inf

  # probability of always treating someone with treatment 1 (one.vs.next.best>tau)
  alway.trt1.prob = mean((one.vs.next.best>tau) * obsWeights)
  # probability of sometimes treating someone with treatment 1 (one.vs.next.best==tau) -- prob determined by resource constraint
  sometimes.trt1.prob = mean((one.vs.next.best==tau) * obsWeights)

  # vector of optimal treatment probabilities
  # (allowed to be stochastic to deal with resource constraint. typically is deterministic)
  g.star = t(apply(blip.est,1,function(xx){
    wm = which.max(xx)
    out = rep(0,length(xx))
    if((wm==1) & (xx[1]-max(xx[-1])==tau)){
      # if the optimal treatment is 1 at the probability that is allowed by the resource constraint
      out[1] = (tau>0)*(kappa - alway.trt1.prob)/sometimes.trt1.prob
      # otherwise assign the second best treatment
      out[which.max(xx[-1])+1] = 1-out[1]
    } else {
      out[wm] = 1
    }
    return(out)
  }))

  H = Reduce("+",lapply(seq(txs),function(i){
    a = txs[i]
    Delta*(A==a)*(g.star[,i]-baseline.probs[i])/(g.est[,i]
```

Example: code for Luedtke et al. (2017)

Example: code for Luedtke et al. (2017)

Recap of the status quo

Implementing efficient estimators currently requires

- 📝 **Manually deriving the EIF**
- 💻 **Implementing an estimator in software**

These tasks are

- **time-consuming**
- **error-prone**

Our objective: develop an algorithm to construct an efficient estimator of any suitably ‘nice’ parameter

Desideratum for our example

Want to estimate

$$\psi(P) = 1 - \frac{\int \{y - E_P(Y \mid X = x)\}^2 \, dP(x, y)}{\mathrm{Var}_P(Y)}$$

with the following code:

```
P = Distribution(data=dat)
v = Var(P,'Y')
mu = E(P,'Y',indep_vars=['X1','X2'])
R2 = 1-E(P,(RV('Y')-mu)**2)/v
estimate(R2)
# Output:  {'est': 0.433,
#           'ci': [0.393, 0.474]}
```

Background and our objective


Deriving the EIF with **automatic differentiation**


Implementing an estimator through **probabilistic programming**

Examples and experiments

Discussion

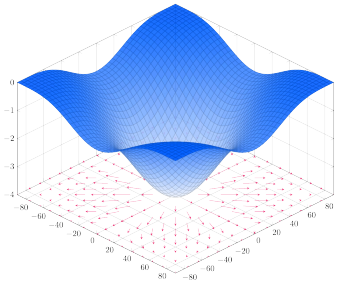
I'll focus on a **nonparametric model** today.

An EIF is a gradient (Pfanzagl, 1982; van der Vaart, 1991b)

Differentiability of function $h : \mathbb{R}^d \to \mathbb{R}$ at a:

$\exists$ gradient $\nabla h(a)$ s.t., for all $a_\epsilon = a + \epsilon t$,

$$\frac{h(a_\epsilon) - h(a)}{\epsilon} \xrightarrow{\epsilon \to 0} \langle \nabla h(a), t \rangle_{\mathbb{R}^d}.$$



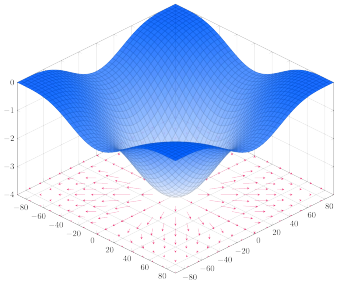
Wikimedia Commons

An EIF is a gradient (Pfanzagl, 1982; van der Vaart, 1991b)

Differentiability of function $h : \mathbb{R}^d \to \mathbb{R}$ at a:

$\exists$ gradient $\nabla h(a)$ s.t., for all $a_\epsilon = a + \epsilon t$,

$$\frac{h(a_\epsilon) - h(a)}{\epsilon} \quad \xrightarrow{\epsilon \to 0} \quad \langle \nabla h(a) \, , \, t \rangle_{\mathbb{R}^d}.$$



Wikimedia Commons

Pathwise differentiability of parameter $\psi : \mathcal{M} \to \mathbb{R}$ at P:

$\exists$ EIF f s.t., for all $\{P_\epsilon : \epsilon \in \mathbb{R}\}$ with $P_{\epsilon=0} = P$ and score s,[1]

$$\frac{\psi(P_\epsilon) - \psi(P)}{\epsilon} \quad \xrightarrow{\epsilon \to 0} \quad \langle f \, , \, s \rangle_{L^2(P)}.$$

[1] The EIF must also be P-mean zero and square integrable.

Chain rule

Differentiable functions $h : \mathbb{R}^d \to \mathbb{R}^c$ satisfy a **chain rule**

Chain rule

Differentiable functions $h : \mathbb{R}^d \to \mathbb{R}^c$ satisfy a **chain rule**

We use this fact all the time! E.g., to differentiate

$$h(x, y) = \frac{e^{xy}\cos(x) + \sin^2(y) + \log(x)}{2x}$$

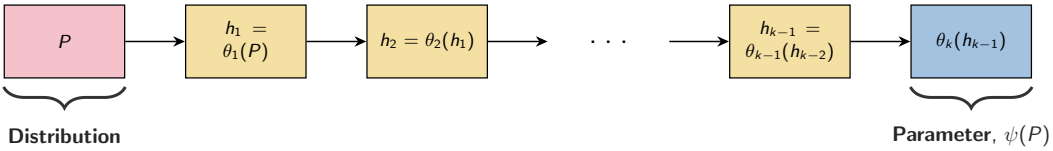
A chain rule for pathwise differentiability

Does pathwise differentiability satisfy a chain rule?

A chain rule for pathwise differentiability

Does pathwise differentiability satisfy a chain rule?

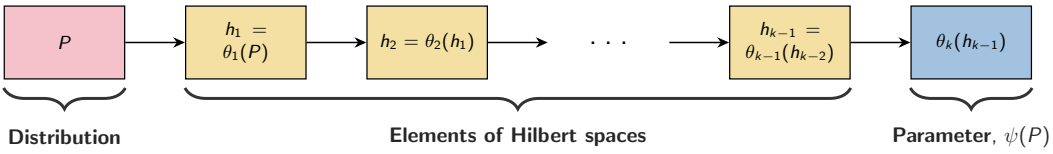
Yes!
Averbukh et al. (1967) and van der Vaart (1991) show it does for compositions like



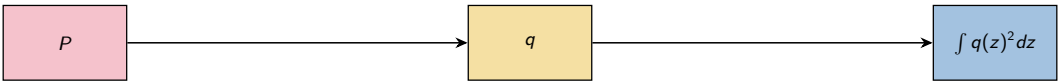
A chain rule for pathwise differentiability

Does pathwise differentiability satisfy a chain rule?

Yes!
Averbukh et al. (1967) and van der Vaart (1991) show it does for compositions like



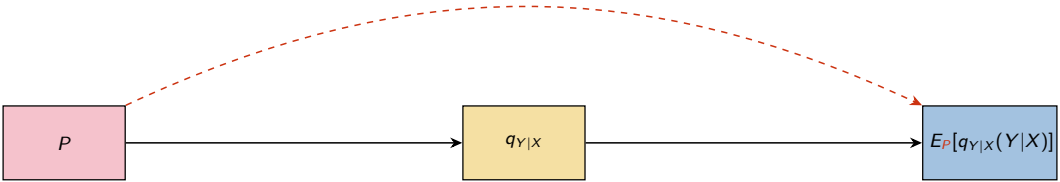
A chain rule for pathwise differentiability



Example: Integral of density squared, $\psi(P) = \int q(z)^2 dz$

$$\theta_1(P) = q$$
$$\theta_2(q) = \int q(z)^2 \, dz$$

A chain rule for pathwise differentiability



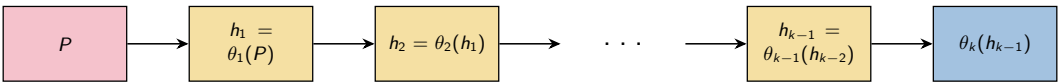
Non-example: Expected conditional density, $\psi(P) = E_P[q_{Y|X}(Y \mid X)]$

$$\theta_1(P) = q_{Y|X}$$

$$\theta_2(q_{Y|X}) \stackrel{?!}{=} E_P[q_{Y|X}(Y \mid X)]$$

A chain rule for pathwise differentiability

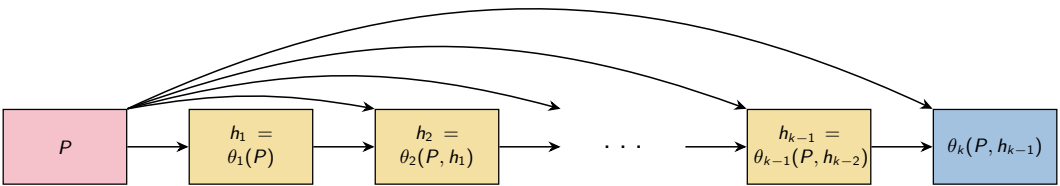
We let all maps in the composition potentially depend on P.



A chain rule for pathwise differentiability

We let all maps in the composition potentially depend on P.

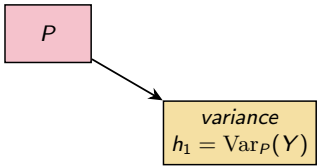
In its simplest form, this means letting:



Example: nonparametric R^2 parameter

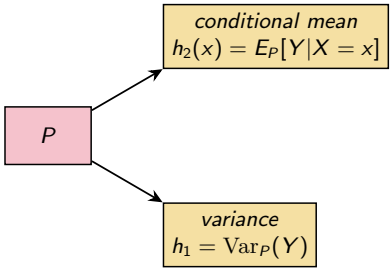

```
P = Distribution(data=dat)
```

Example: nonparametric R^2 parameter



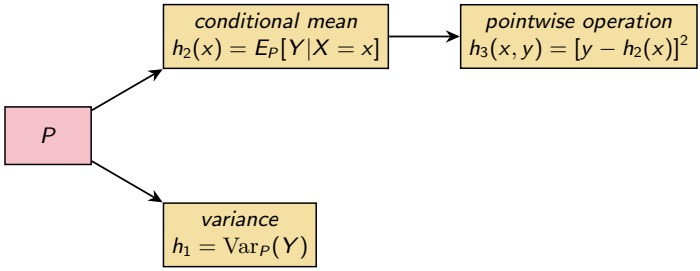

```
P = Distribution(data=dat)
v = Var(P,'Y')
```

Example: nonparametric R^2 parameter



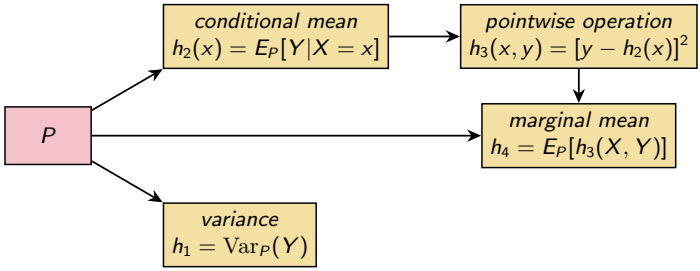

```
P = Distribution(data=dat)
v = Var(P,'Y')
mu = E(P,'Y',indep_vars=['X1','X2'])
```

Example: nonparametric R^2 parameter



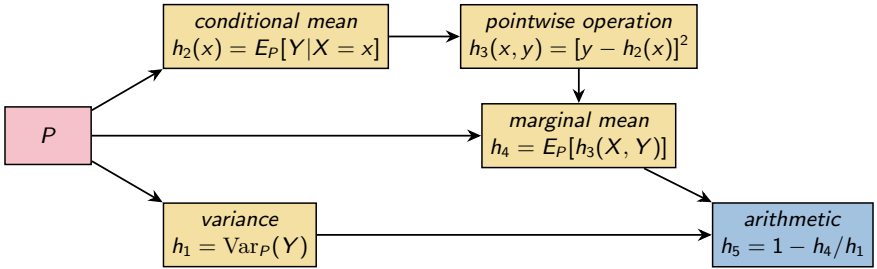

```
P = Distribution(data=dat)
v = Var(P,'Y')
mu = E(P,'Y',indep_vars=['X1','X2'])
sq_resid = (RV('Y')-mu)**2
```

Example: nonparametric R^2 parameter




```
P = Distribution(data=dat)
v = Var(P,'Y')
mu = E(P,'Y',indep_vars=['X1','X2'])
sq_resid = (RV('Y')-mu)**2
mean_sq_resid = E(P,sq_resid)
```

Example: nonparametric R^2 parameter




```
P = Distribution(data=dat)
v = Var(P,'Y')
mu = E(P,'Y',indep_vars=['X1','X2'])
sq_resid = (RV('Y')-mu)**2
mean_sq_resid = E(P,sq_resid)
R2 = 1-mean_sq_resid/v
```

We introduce a chain rule for differentiating any ψ expressible as follows:

Algorithm Takes as input P and returns $\psi(P)$

1: $h_1 = \theta_1(P, h_{\mathrm{pa}(1)})$

2: $h_2 = \theta_2(P, h_{\mathrm{pa}(2)})$

$\vdots$

k: $h_k = \theta_k(P, h_{\mathrm{pa}(k)})$

return $\psi(P) = h_k$

We introduce a chain rule for differentiating any ψ expressible as follows:

Algorithm Takes as input P and returns $\psi(P)$

1: $h_1 = \theta_1(P, h_{\mathrm{pa}(1)})$

2: $h_2 = \theta_2(P, h_{\mathrm{pa}(2)})$

$\vdots$

k: $h_k = \theta_k(P, h_{\mathrm{pa}(k)})$

return $\psi(P) = h_k$

Each **primitive** θ_j must be **totally pathwise differentiable**

Having a **chain rule**

enables

automatic differentiation

Having a **chain rule**

enables

automatic differentiation

Algorithm Automatic differentiation to find the EIF

Require: $h_1, h_2, \ldots, h_k$

1: **initialize** $f_0, f_1, f_2, \ldots, f_{k-1} = 0$ and $f_k = 1$

2: **for** $j = k, k-1, \ldots, 1$ **do**

3: **augment** $(f_0, f_{\mathrm{pa}(j)}) \mathrel{+}= \dot{\theta}^*_{j,P,h_{\mathrm{pa}(j)}}(f_j)$

4: **return** f_0

Examples of totally pathwise differentiable primitives

Conditional mean: $\theta(P, h)(\cdot) = E_P[\, h(Z) \mid X = \cdot\,]$

Conditional variance: $\theta(P, h)(\cdot) = \mathrm{Var}_P[\, h(Z) \mid X = \cdot\,]$

Conditional density: $\theta(P, h) = p_{Y|X}$

Squared L^2 norm: $\theta(P, h) = \int h(z)^2 \, dz$

Also: **differentiable functions**, **optimal values**, **pointwise operations**, ...

Background and our objective


Deriving the EIF with **automatic differentiation**


Implementing an estimator through **probabilistic programming**

Examples and experiments

Discussion

Probabilistic programming

Probabilistic programming unifies

- **traditional programming**, and
- **statistical inference**

Examples:

- **Stan** (Stan Development Team, 2024)
- **Edward** (Tran et al., 2017)
- **Pyro** (Bingham et al., 2019)

To date, there is no probabilistic programming language for constructing **efficient estimators** in **nonparametric models**.

High-level overview of our approach

Algorithm Takes as input P and returns $\psi(P)$

1: $h_1 = \theta_1(P, h_{\mathrm{pa}(1)})$

2: $h_2 = \theta_2(P, h_{\mathrm{pa}(2)})$

$\vdots$

k: $h_k = \theta_k(P, h_{\mathrm{pa}(k)})$

return $\psi(P) = h_k$

High-level overview of our approach

Algorithm Obtain initial estimate of $\psi(P)$ using data

1: $\widehat{h}_1 = \theta_1(\widehat{P}_1, \widehat{h}_{\mathrm{pa}(1)})$

2: $\widehat{h}_2 = \theta_2(\widehat{P}_2, \widehat{h}_{\mathrm{pa}(2)})$

$\vdots$

k: $\widehat{h}_k = \theta_k(\widehat{P}_k, \widehat{h}_{\mathrm{pa}(k)})$

return $\widehat{\psi} = \widehat{h}_k$

High-level overview of our approach

Algorithm Obtain initial estimate of $\psi(P)$ using data

1: $\widehat{h}_1 = \theta_1(\widehat{P}_1, \widehat{h}_{\mathrm{pa}(1)})$

2: $\widehat{h}_2 = \theta_2(\widehat{P}_2, \widehat{h}_{\mathrm{pa}(2)})$

$\vdots$

k: $\widehat{h}_k = \theta_k(\widehat{P}_k, \widehat{h}_{\mathrm{pa}(k)})$

return $\widehat{\psi} = \widehat{h}_k$

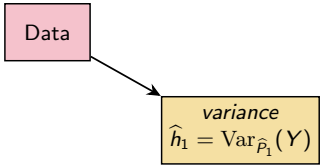
Nuisances are estimated in a **just-in-time** fashion

Example: initial estimate of the nonparametric R^2

Data

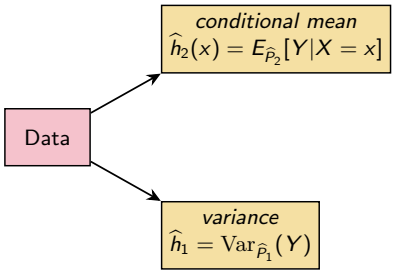
```
P = Distribution(data=dat)
```

Example: initial estimate of the nonparametric R^2



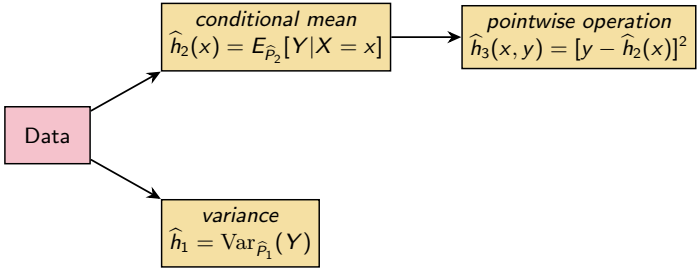

```
0  P = Distribution(data=dat)
1  v = Var(P,'Y')
```

Example: initial estimate of the nonparametric R^2



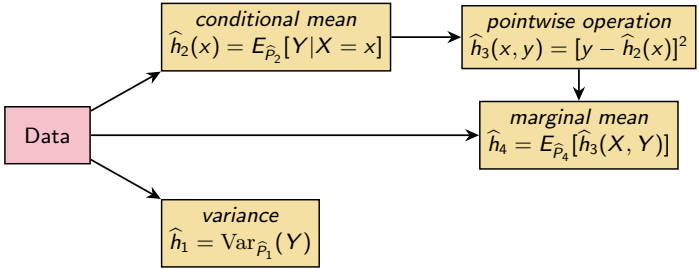

```
P = Distribution(data=dat)
v = Var(P,'Y')
mu = E(P,'Y',indep_vars=['X1','X2'])
```

Example: initial estimate of the nonparametric R^2



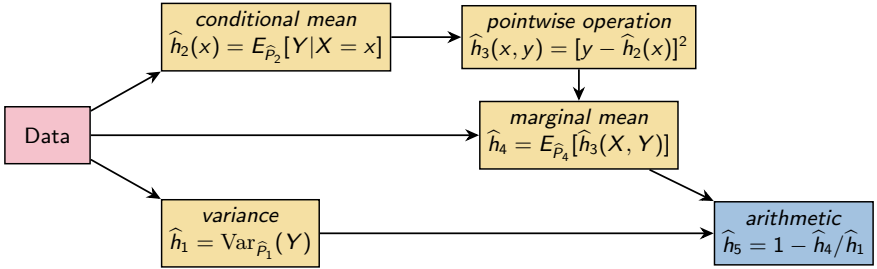

```
P = Distribution(data=dat)
v = Var(P,'Y')
mu = E(P,'Y',indep_vars=['X1','X2'])
sq_resid = (RV('Y')-mu)**2
```

Example: initial estimate of the nonparametric R^2




```
P = Distribution(data=dat)
v = Var(P,'Y')
mu = E(P,'Y',indep_vars=['X1','X2'])
sq_resid = (RV('Y')-mu)**2
mean_sq_resid = E(P,sq_resid)
```

Example: initial estimate of the nonparametric R^2




```
P = Distribution(data=dat)
v = Var(P,'Y')
mu = E(P,'Y',indep_vars=['X1','X2'])
sq_resid = (RV('Y')-mu)**2
mean_sq_resid = E(P,sq_resid)
R2 = 1-mean_sq_resid/v
```

Debiasing and confidence interval construction

A similar just-in-time approach is used to estimate the **EIF** $\widehat{f}$

Debiasing and confidence interval construction

A similar just-in-time approach is used to estimate the **EIF** $\widehat{f}$

Final estimator is a **one-step estimator**

$$\widehat{\psi}^{\mathrm{OS}} = \widehat{\psi} + \frac{1}{n}\sum_{i=1}^{n} \widehat{f}(Z_i)$$

with 95% CI

$$\widehat{\psi}^{\mathrm{OS}} \pm 1.96\, \frac{\widehat{\sigma}}{n^{1/2}}$$

Background and our objective


Deriving the EIF with **automatic differentiation**


Implementing an estimator through **probabilistic programming**

Examples and experiments

Discussion

Nonparametric R^2

Observe n iid draws of $(X, Y) \sim P$

- $X = (X_1, X_2)$ with $X_1, X_2 \overset{iid}{\sim} \text{Unif}[-1, 1]$
- $Y \mid X \sim N(\frac{25}{9} X_1^2, 1)$

```
P = Distribution(data=dat)
v = Var(P,'Y')
mu = E(P,'Y',indep_vars=['X1','X2'])
R2 = 1-E(P,(RV('Y')-mu)**2)/v
estimate(R2)
```

Nonparametric R^2 results

n	Coverage $\geq 95\%$	Rel. Width ≤ 1.00	Rel. Variance ≤ 1.00	Bias^2/MSE $= 0.00$
250	87%	0.89	1.12	0.10
1000	92%	0.94	1.05	0.05
4000	94%	0.97	1.03	0.00
16000	93%	0.98	1.10	0.01

Nonparametric R^2 results

n	Coverage $\geq 95\%$	Rel. Width ≤ 1.00	Rel. Variance ≤ 1.00	Bias^2/MSE $= 0.00$
250	87%	0.89	1.12	0.10
1000	92%	0.94	1.05	0.05
4000	94%	0.97	1.03	0.00
16000	93%	0.98	1.10	0.01

Nonparametric R^2 results

n	Coverage $\geq 95\%$	Rel. Width ≤ 1.00	Rel. Variance ≤ 1.00	Bias^2/MSE $= 0.00$
250	87%	0.89	1.12	0.10
1000	92%	0.94	1.05	0.05
4000	94%	0.97	1.03	0.00
16000	93%	0.98	1.10	0.01

Nonparametric R^2 results

n	Coverage $\geq 95\%$	Rel. Width ≤ 1.00	Rel. Variance ≤ 1.00	Bias^2/MSE $= 0.00$
250	87%	0.89	1.12	0.10
1000	92%	0.94	1.05	0.05
4000	94%	0.97	1.03	0.00
16000	93%	0.98	1.10	0.01

Nonparametric R^2 results

n	Coverage $\geq 95\%$	Rel. Width ≤ 1.00	Rel. Variance ≤ 1.00	Bias^2/MSE $= 0.00$
250	87%	0.89	1.12	0.10
1000	92%	0.94	1.05	0.05
4000	94%	0.97	1.03	0.00
16000	93%	0.98	1.10	0.01

Further examples

Integral of density squared (Bickel and Ritov, 1988)

Longitudinal G-formula (Robins, 1986; Bang et al., 2005)

Further examples

Integral of density squared (Bickel and Ritov, 1988)

```
P = Distribution(data=dat)
dens = Density(P, 'Z')
expected_density = E(P,dens)
estimate(expected_density)
```

Longitudinal G-formula (Robins, 1986; Bang et al., 2005)

```
P = Distribution(data=dat)
T = 3
mu = 'Y'
for t in reversed(range(T)):
    H = [f'X{j}' for j in range(t+1)]+[f'A{j}' for j in range(t)]
    mu = E(P,dep=mu,indep_vars=H, fixed_vars={f'A{t}==1'})
mu = E(P,dep=mu)
estimate(mu)
```

Background and our objective


Deriving the EIF with **automatic differentiation**


Implementing an estimator through **probabilistic programming**

Examples and experiments

Discussion

Parameter representation is not unique

The same parameter can be expressed in different ways.

Simple analogy: $f(x) = 2x^2 + 1$ can be a composition $g \circ h$ of

- $g(u) = 2u + 1$ and $h(x) = x^2$
- $g(u) = \cosh(u)$ and $h(x) = 2\sinh^{-1}(x)$
- $g(u) = 2u^{1/2} + 1$ and $h(x) = x^4$

Parameter representation is not unique

The same parameter can be expressed in different ways.

Simple analogy: $f(x) = 2x^2 + 1$ can be a composition $g \circ h$ of

- $g(u) = 2u + 1$ and $h(x) = x^2$
- $g(u) = \cosh(u)$ and $h(x) = 2\sinh^{-1}(x)$
- $g(u) = 2u^{1/2} + 1$ and $h(x) = x^4$

Some representations can be used to automatically compute $f'(0)$

Parameter representation is not unique

The same parameter can be expressed in different ways.

Simple analogy: $f(x) = 2x^2 + 1$ can be a composition $g \circ h$ of

- $g(u) = 2u + 1$ and $h(x) = x^2$
- $g(u) = \cosh(u)$ and $h(x) = 2\sinh^{-1}(x)$
- $g(u) = 2u^{1/2} + 1$ and $h(x) = x^4$

Some representations can be used to automatically compute $f'(0)$, but **some can't**.

Parameter representation is not unique

The same parameter can be expressed in different ways.

The way the parameter is expressed has statistical consequences:

1. It determines the **nuisances that must be estimated**.
   E.g., $\psi(P) = E_P[Z]^2$ can be expressed in a way that requires either
   - **estimating a density**, or
   - **estimating a mean**.
2. It changes the required **convergence rate conditions** for efficiency.

Parameter representation is not unique

The same parameter can be expressed in different ways.

The way the parameter is expressed has statistical consequences:

1. It determines the **nuisances that must be estimated**.
   E.g., $\psi(P) = E_P[Z]^2$ can be expressed in a way that requires either
   - **estimating a density**, or
   - **estimating a mean**.
2. It changes the required **convergence rate conditions** for efficiency.

Future work: *Is there an algorithm to find a representation that simplifies nuisance estimation?*

A word of caution

Any automated approach to inference has a potential for abuse

The validity of the proposed methods relies on regularity conditions

- Interpretable by experts, but software can be used without consulting them

Possible software improvement:

return a list of regularity conditions along with the estimate?

A new workflow for debiased inference

New approach suggests **refocusing research effort**:

- Current practice: derive the EIF for each new parameter
- Proposed alternative: establish differentiability of **novel, reusable primitives**.

Provides a path to rapidly translate new developments **from theory to practice**:

- Iterative debiasing, novel cross-fitting schemes, shape constraints, etc.

Python package

Proof-of-concept Python package available on PyPi:

`pydimple`

Acknowledgements

This research was supported by the:

- **NIH** through award numbers DP2-LM013340;
- **NSF** through award number DMS-2210216.

Thank you!

Questions?

References I

Averbukh, V. I. and O. G. Smolyanov (1967). "The theory of differentiation in linear topological spaces". In: *Russian Mathematical Surveys* 22.6, p. 201.

Bang, H. and J. M. Robins (2005). "Doubly robust estimation in missing data and causal inference models". In: *Biometrics* 61.4, pp. 962–973.

Bickel, P. J., C. A. Klaassen, et al. (1993). *Efficient and adaptive estimation for semiparametric models*. Vol. 4. Johns Hopkins University Press Baltimore.

Bickel, P. J. and Y. Ritov (1988). "Estimating integrated squared density derivatives: sharp best order of convergence estimates". In: *Sankhyā: The Indian Journal of Statistics, Series A*, pp. 381–393.

Bingham, E. et al. (2019). "Pyro: Deep universal probabilistic programming". In: *Journal of machine learning research* 20.28, pp. 1–6.

Chernozhukov, V. et al. (2017). "Double/debiased/Neyman machine learning of treatment effects". In: *American Economic Review* 107.5, pp. 261–265.

Hasminskii, R. Z. and I. A. Ibragimov (1979). "On the nonparametric estimation of functionals". In: *Proceedings of the Second Prague Symposium on Asymptotic Statistics*. Vol. 473. North-Holland Amsterdam, pp. 474–482.

Le Cam, L. (1956). "On the asymptotic theory of estimation and testing hypotheses". In: *Proceedings of the Third Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics*. Vol. 3. University of California Press, pp. 129–157.

Levit, B. Y. (1975). "On efficiency of a class of non-parametric estimates". In: *Teoriya Veroyatnostei i ee Primeneniya* 20.4, pp. 738–754.

Li, X., S. Li, and A. Luedtke (2023). "Estimating the efficiency gain of covariate-adjusted analyses in future clinical trials using external data". In: *Journal of the Royal Statistical Society Series B: Statistical Methodology* 85.2, pp. 356–377.

Luedtke, A. (2025). "Simplifying Debiased Inference via Automatic Differentiation and Probabilistic Programming". In: *Journal of the Royal Statistical Society - Series B*.

Newey, W. K. (1994). "The asymptotic variance of semiparametric estimators". In: *Econometrica: Journal of the Econometric Society*, pp. 1349–1382.

Pfanzagl, J. (1982). *Contributions to a general asymptotic statistical theory*. Springer.

Robins, J. (1986). "A new approach to causal inference in mortality studies with a sustained exposure period – application to control of the healthy worker survivor effect". In: *Mathematical modelling* 7.9-12, pp. 1393–1512.

Stan Development Team (2024). *Stan Modeling Language Users Guide and Reference Manual, Version 2.18.0*. URL: `http://mc-stan.org/`.

Tran, D. et al. (2017). "Deep probabilistic programming". In: *International Conference on Learning Representations*.

van der Laan, M. J. and D. Rubin (2006). "Targeted maximum likelihood learning". In: *The international journal of biostatistics* 2.1.

van der Vaart, A. (1991a). "Efficiency and Hadamard differentiability". In: *Scandinavian Journal of Statistics*, pp. 63–75.

van der Vaart, A. (1991b). "On differentiable functionals". In: *The Annals of Statistics*, pp. 178–204.

VanderWeele, T. J. et al. (2019). "Selecting optimal subgroups for treatment using many covariates". In: *Epidemiology* 30.3, pp. 334–341.

Williamson, B. D. et al. (2021). "Nonparametric variable importance assessment using machine learning techniques". In: *Biometrics* 77.1, pp. 9–22.

Wright, S. (1921). "Correlation and causation". In: *Journal of agricultural research* 20.7, p. 557.

Extra Slides

One-step estimation (extra details)

One-step estimation often used to construct efficient estimators:

$$\textbf{initial estimator} \; + \; \textbf{debiasing term}$$

In parametric models: update $\widehat{\beta}$ with one step of Newton's method to solve

$$\frac{1}{n}\sum_{i=1}^{n} \dot{\ell}_{\beta}(Z_i) = 0, \, ,$$

yielding

$$\widehat{\beta} \; + \; \frac{1}{n}\sum_{i=1}^{n} I_{\widehat{\beta}}^{-1} \dot{\ell}_{\widehat{\beta}}(Z_i)$$

Examples of totally pathwise differentiable primitives $\theta : \mathcal{U} \to \mathcal{W}$

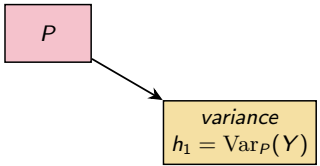
	$\mathcal{U}$	$\mathcal{W}$	$\theta(P, u)$
Conditional mean	$L^2(Q)$	$L^2(Q_X)$	$E_P[u(Z) \mid X = \cdot\,]$
Conditional variance	$L^2(Q)$	$L^2(Q_X)$	$\mathrm{Var}_P[u(Z) \mid X = \cdot\,]$
Conditional covariance	$L^2(Q)^{\oplus 2}$	$L^2(Q_X)$	$\mathrm{cov}_P[u_1(Z), u_2(Z) \mid X = \cdot\,]$
Optimal value	$\mathcal{U}$	$\mathbb{R}$	$\min_{y \in \mathcal{Y}} F_y(P, u)$
Optimal solution	$\mathcal{U}$	$\mathbb{R}^d$	$\mathrm{argmin}_{w \in \mathcal{W}} F_w(P, u)$
Conditional density	$\{0\}$	$L^2(Q)$	$p_{Y\mid X}$, with $Z = (X, Y)$
Counterfactual density	$\{0\}$	$L^2(\lambda)$	$y \mapsto E_P[p(y\mid A = 1, X)]$
Hadamard diff. map	$\mathcal{U}$	$\mathcal{W}$	$\zeta(u)$
↳ Inner product	$\mathcal{R} \oplus \mathcal{R}$	$\mathbb{R}$	$\langle u_1, u_2 \rangle_{\mathcal{R}}$, with $u = (u_1, u_2)$
↳ Squared norm	$\mathcal{U}$	$\mathbb{R}$	$\|u\|^2_{\mathcal{T}}$
↳ Differentiable function	$\mathbb{R}^d$	$\mathbb{R}$	$\zeta(u)$
↳ Pointwise operations	$L^2(\lambda)^{\oplus d}$	$L^2(\lambda)$	$x \mapsto f_x \circ \bar{u}(x)$, with $\bar{u}(x) = (u_j(x))_{j=1}^d$
↳ Constant map	$\{0\}$	$\mathcal{W}$	c, with $c \in \mathcal{W}$
↳ Coordinate projection	$\mathcal{R}_1 \oplus \mathcal{R}_2$	$\mathcal{R}_1$	u_1, with $u = (u_1, u_2)$
↳ Change of measure	$L^2(\lambda_1)$	$L^2(\lambda_2)$	u
↳ Lift to new domain	$L^2(Q_X)$	$L^2(Q)$	$z \mapsto u(x)$
↳ Fix binary argument	$L^2(Q_{A,X})$	$L^2(Q_X)$	$x \mapsto u(1, x)$

Example: nonparametric R^2 parameter (extra details)


```
P = Distribution(data=dat)
```

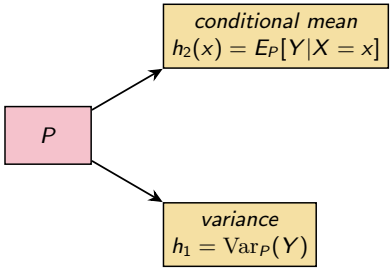
P

Example: nonparametric R^2 parameter (extra details)



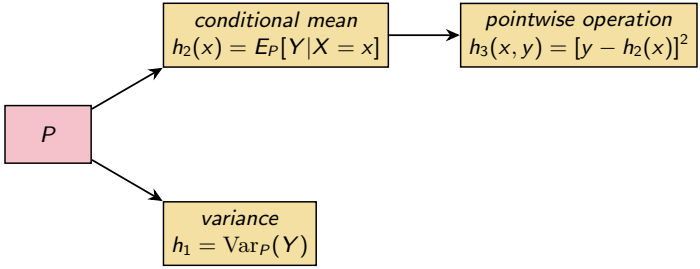

```
0  P = Distribution(data=dat)                    P
1  v = Var(P,'Y')                                Var_P(Y)
```

Example: nonparametric R^2 parameter (extra details)



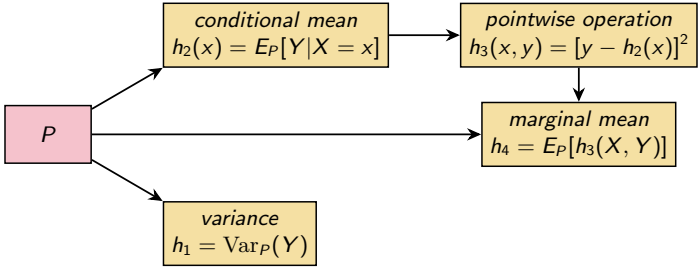

```
P = Distribution(data=dat)                          P
v = Var(P,'Y')                                      Var_P(Y)
mu = E(P,'Y',indep_vars=['X1','X2'])                E_P[Y | X = ·]
```

Example: nonparametric R^2 parameter (extra details)



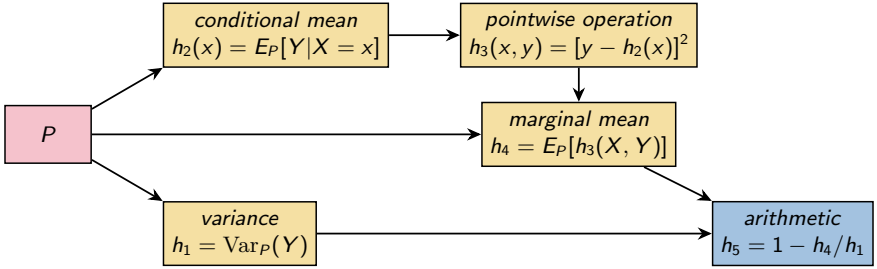

```
P = Distribution(data=dat)                          P
v = Var(P,'Y')                                      Var_P(Y)
mu = E(P,'Y',indep_vars=['X1','X2'])                E_P[Y | X = ·]
sq_resid = (RV('Y')-mu)**2                          (Y − E_P[Y | X])^2
```

Example: nonparametric R^2 parameter (extra details)




```
0  P = Distribution(data=dat)                          P
1  v = Var(P,'Y')                                      Var_P(Y)
2  mu = E(P,'Y',indep_vars=['X1','X2'])                E_P[Y | X = ·]
3  sq_resid = (RV('Y')-mu)**2                          (Y − E_P[Y | X])^2
4  mean_sq_resid = E(P,sq_resid)                       E_P[(Y − E_P[Y | X])^2]
```

Example: nonparametric R^2 parameter (extra details)




```
0  P = Distribution(data=dat)                          P
1  v = Var(P,'Y')                                      Var_P(Y)
2  mu = E(P,'Y',indep_vars=['X1','X2'])                E_P[Y | X = ·]
3  sq_resid = (RV('Y')-mu)**2                          (Y − E_P[Y | X])^2
4  mean_sq_resid = E(P,sq_resid)                       E_P[(Y − E_P[Y | X])^2]
5  R2 = 1-mean_sq_resid/v                              ψ(P)
```

Integral of density squared

Observe n iid draws of $Z \sim \mathrm{Beta}(3, 5)$

```
P = Distribution(data=dat)
dens = Density(P, 'Z')
expected_density = E(P,dens)
estimate(expected_density)
```

Integral of density squared results

n	Coverage $\geq 95\%$	Rel. Width ≤ 1.00	Rel. Variance ≤ 1.00	Bias^2/MSE $= 0.00$
250	91%	0.94	1.13	0.04
1000	92%	0.96	0.95	0.07
4000	93%	0.97	1.06	0.02
16000	95%	0.98	0.98	0.03

Integral of density squared results

n	Coverage $\geq 95\%$	Rel. Width ≤ 1.00	Rel. Variance ≤ 1.00	Bias^2/MSE $= 0.00$
250	91%	0.94	1.13	0.04
1000	92%	0.96	0.95	0.07
4000	93%	0.97	1.06	0.02
16000	95%	0.98	0.98	0.03

Integral of density squared results

n	Coverage $\geq 95\%$	Rel. Width ≤ 1.00	Rel. Variance ≤ 1.00	Bias^2/MSE $= 0.00$
250	91%	0.94	1.13	0.04
1000	92%	0.96	0.95	0.07
4000	93%	0.97	1.06	0.02
16000	95%	0.98	0.98	0.03

Integral of density squared results

n	Coverage $\geq 95\%$	Rel. Width ≤ 1.00	Rel. Variance ≤ 1.00	Bias^2/MSE $= 0.00$
250	91%	0.94	1.13	0.04
1000	92%	0.96	0.95	0.07
4000	93%	0.97	1.06	0.02
16000	95%	0.98	0.98	0.03

Integral of density squared results

n	Coverage $\geq 95\%$	Rel. Width ≤ 1.00	Rel. Variance ≤ 1.00	Bias^2/MSE $= 0.00$
250	91%	0.94	1.13	0.04
1000	92%	0.96	0.95	0.07
4000	93%	0.97	1.06	0.02
16000	95%	0.98	0.98	0.03

Longitudinal G-formula (Robins, 1986; Bang et al., 2005)

Observe n iid draws of $H_T = (X_0, A_0, X_1, A_1, \ldots, X_{T-1}, A_{T-1}, Y)$

- (X_t, A_t) a covariate-treatment pair at time t
- Y an outcome

$H_t = (X_0, A_0, \ldots, A_{t-1}, X_t)$ denotes history before treatment t

$\psi(P)$ defined recursively by letting

- $\mu_{P,T}(h_T) = y$
- $\mu_{P,t}(h_t) = E_P[\mu_{P,t+1}(H_{t+1}) \,|\, A_t = 1, H_t = h_t]$ for $t = T-1, T-2, \ldots, 0$
- $\psi(P) = E_P[\mu_{P,0}(H_0)]$

EIF takes the form

$$f(h_{T+1}) = \mu_{P,0}(h_0) - \psi(P) + \sum_{t=0}^{T-1} \left[\prod_{s=0}^{t} \frac{1(a_s = 1)}{P(A_s = 1 \,|\, H_s = h_s)} \right] [\mu_{P,t+1}(h_{t+1}) - \mu_{P,t}(h_t)] .$$

Longitudinal G-formula (Robins, 1986; Bang et al., 2005)

```
P = Distribution(data=dat)
T = 3
mu = 'Y'
for t in reversed(range(T)):
    H = [f'X{j}' for j in range(t+1)]+[f'A{j}' for j in range(t)]
    mu = E(P,dep=mu,indep_vars=H, fixed_vars={f'A{t}==1'})
mu = E(P,dep=mu)
estimate(mu)
```

Longitudinal G-formula results

n	Coverage $\geq 95\%$	Rel. Width ≤ 1.00	Rel. Variance ≤ 1.00	Bias^2/MSE $= 0.00$
250	79%	1.09	4.44	0.12
1000	93%	1.28	2.01	0.02
4000	94%	1.25	2.10	0.01
16000	94%	1.06	1.26	0.00

Longitudinal G-formula results

n	Coverage $\geq 95\%$	Rel. Width ≤ 1.00	Rel. Variance ≤ 1.00	Bias^2/MSE $= 0.00$
250	79%	1.09	4.44	0.12
1000	93%	1.28	2.01	0.02
4000	94%	1.25	2.10	0.01
16000	94%	1.06	1.26	0.00

Longitudinal G-formula results

n	Coverage $\geq 95\%$	Rel. Width ≤ 1.00	Rel. Variance ≤ 1.00	Bias^2/MSE $= 0.00$
250	79%	1.09	4.44	0.12
1000	93%	1.28	2.01	0.02
4000	94%	1.25	2.10	0.01
16000	94%	1.06	1.26	0.00

Longitudinal G-formula results

n	Coverage $\geq 95\%$	Rel. Width ≤ 1.00	Rel. Variance ≤ 1.00	Bias^2/MSE $= 0.00$
250	79%	1.09	4.44	0.12
1000	93%	1.28	2.01	0.02
4000	94%	1.25	2.10	0.01
16000	94%	1.06	1.26	0.00

Longitudinal G-formula results

n	Coverage $\geq 95\%$	Rel. Width ≤ 1.00	Rel. Variance ≤ 1.00	Bias^2/MSE $= 0.00$
250	79%	1.09	4.44	0.12
1000	93%	1.28	2.01	0.02
4000	94%	1.25	2.10	0.01
16000	94%	1.06	1.26	0.00