Principal Surrogate Evaluation with pseval

Basics

pseval is designed to analyze data from a **randomized clinical trial** in order to asses the **surrogate** value of a post-randomization measurement. Start by describing the study **design**, including augmentations.

design

```
p1 <- psdesign(data = data,
   Z = Z, Y = Y.obs, S = S.obs,
   BIP = BIP, CPV = CPV)
```

The counterfactual **surrogate** S.1 is missing for many subjects, thus we need to define a model to **integrate** over the missing values.

integration

```
p1 <- p1 +
integrate_parametric(S.1 ~ BIP)
```

The **risk model** describes the relationship between the **outcome** Y, the **surrogate** S.1, and the **treatment** Z. Use the risk model that is most appropriate for your outcome type, binary, count, or time-to-event.

risk model

```
p1 <- p1 +
   risk_binary(Y ~ S.1 * Z,
   D=500, risk.logit)
```

Estimation and **bootstrap** inference are done in separate steps. The main method is estimated maximum likelihood, but pseudo-score is available for a subset of models.

estimation

```
p1 <- p1 + ps_estimate() +
    ps_bootstrap()
```

Specifics combine model components together with the '+' sign

Study Design specification and mapping

psdesign controls the dataset that is being used, and how to map variables to their roles in the analysis. The "keys" to the left of "=" map to variables in data

data frame, treatment

```
a <- psdesign(data = data, Z = Z,
   Y = Y.obs, S = S.obs, BIP = W)
```

clinical outcome, surrogate, augmentation

survival outcome, other mappings

```
b <- psdesign(data = data, Z = Z,
   Y = Surv(time.obs, event.obs),
   tau = .25, S = S.obs,
   BIP = W, CPV = CPV, BSM = V1,
   weights = p, covariate = X)
```

Integration over the missing counterfactuals

integrate_* functions control how the missing counterfactual variables are handled

Parametric: Assumes normal distribution conditional on a BIP + other variables

```
a + integrate_parametric(S.1 ~ BIP)
```

Semiparametric: Assumes location and scale vary as functions of BIP + other variables, no assumption about distribution of S

```
a + integrate_semiparametric(
      formula.location = S.1 ~ BIP,
      formula.scale = S.1 ~ 1)
```

Nonparametric: Totally empirical, requires categorical S and W

```
a + integrate_nonparametric(S.1 ~ BIP)
```

Risk Model distribution of the outcome

risk_* functions define the assumed relationship between Y, S.1, and Z. The default formula is `Y ~ S.1 * Z`

binary outcome

```
a + risk_binary(risk = risk.logit)
a + risk_binary(risk = risk.probit)
```

time to event outcome

```
a + risk_exponential()
a + risk_weibull()
```

count outcome

```
a + risk_poisson()
```

Options

flexible spline

```
a + risk_binary(Y ~ bs(S.1, df = 2) * Z)
a + risk_exponential(D = 200)
a + risk_poisson(Y~ S.1 * Z + offset(t))
```

Estimation post-estimation and plotting

```
est <- a + ps_estimate(method = "BFGS")
a + ps_estimate(method = "pseudo-score")
boot <- est + ps_bootstrap(n.boots = 50,
   start = binary.est$estimates$par)
```

see ?optim for options

Post estimation

summary of parameters

```
summary(boot)
calc_risk(boot, contrast = "VE")
calc_STG(boot) # total gain statistic
```

plots of different CEP

```
plot(boot, contrast = "VE")
plot(boot, contrast = "logRR")
plot(boot, contrast = "RD",
     CI.type = "pointwise")
```

custom CEP

```
calc_risk(boot,
contrast = function(R0, R1) 1 - R1/R0)
```

 Michael C Sachs • sachsmc@gmail.com • sachsmc.github.io/pseval

Learn more at sachsmc.github.io/pseval • pseval version 1.2.1 • Updated: 6/16