Use Effect Coding to Control Marginal Probability in Simulations

Boyi Guo

2022-05-02

Motivation

Recently, I came across Rudolph et al. (2021) who addresses the problem of controlling marginal probability of the outcome when constructing a simulation study. The authors proposed to calculate a concept called "balance intercept" to replace the "standard intercept" (i.e. the proportion of successes in the reference level). The balance intercept is roughly an adjustment to the reference level prevalence by the group sizes. This same problem is revisited by Robertson, Steingrimsson, and Dahabreh (2021) proposing to calculate the the balance intercept with a numeric approximation. Nevertheless, I think both methods are a bit complicated, due to the fact that both simulations are based on a specific coding scheme of categorical variables, reference coding. [TODO: add citation] I would suggest to use the effect coding, specifically deviation coding, to construct the simulation as a simpler alternative to control the marginal probability of a binary outcome, particularly when the design of the study is balanced (i.e. equal sample size with in each group). The basis of this proposal is that the intercept term of the effect coding model (regardless the link function or parametric assumption) is the mean of the group means, which coincides with the marginal probability of a binary outcome when the groups are balanced. Hence, less calculation is needed to derive the intercept term to reach the target marginal probability. In the case of unbalanced design, we can leverage the given information, e.g., effect size, target marginal probability, and group sizes, to derive the desired intercept, i.e. (the mean of the group mean).

The problem of controlling marginal probability in essence is to find the conditional probability for the reference group, as all the other simulation parameters and the model degrees of freedom are considered fixed.

In the rest of this report, we will show that the simulation procedures with balance and unbalanced design in _R_ (R Core Team 2021) with controlled marginal probability of the outcome for balanced and unbalanced design. We defer the readers who's unfamiliar with coding scheme to <https://stats.oarc.ucla.edu/r/library/r-library-contrast-coding-systems-for-categorical-variables/>. The simulation strategy translates to the programming languages other than _R_, e.g. *SAS*, *STATA*, whose implementation are shown in this article.

Examples

Balanced design

It is very simple to control the marginal probability for balanced design when simulation is based on the effect coding. There are minimum calculation required as the intercept in an effect coding model is the grand mean, in the binary outcome case the marginal probability. Here we illustrate the process following the simple additive probability example in Rudolph et al. (2021), where the target marginal probability is 0.3, the effect size as in risk difference is 0.2, and a balanced covariates with 2 levels.

```r
set.seed(123)

n <- 10000

# Marginal probabilities of each variable
p.y <- 0.3
p.x <- 0.5
rd <- 0.2

# Example 2: Generate L, X, and Y -------------------
X <- rbinom(n, 1, p.x)

# Generate X with marginal prob 0.5
dev_coding <- contr.sum(2)  # Deviation Coding with 2 levels
X_design_dev <- cbind( 1,   # Adding intercept column
                       dev_coding[X+1,]) # Construct the deisgn matrix

# The design matrix with effect coding can be more easily construct with model.matrix function

beta_vec <- c(p.x,   # Intercept term, the marginal probability for balanced desgin
                -rd/2) # Set up conditional prob for reference level
Y <- rbinom(n, 1, X_design_dev %*% beta_vec)
```

To validate the simulation, we can see the marginal probability 0.5041, and the conditional probability of the two levels are 0.3985 and 0.6122 respectively. Hence the simulation matches with the desired design.

```r
mean(Y);
```

```
## [1] 0.5041
```

```r
mean(Y[X==0]);
```

```
## [1] 0.3984576
```

```r
mean(Y[X==1]);
```

```
## [1] 0.6121788
```

For anyone who want to examine the model performance, they can use the following code.

```r
summary(glm(Y~X, family = binomial(link="identity"))) # Reference coding model
summary(glm(Y~X_design_dev-1, family = binomial(link="identity"))) # Effect coding model
```

Unbalanced design

When the groups are not balanced, the simulation with effect coding is less straightforward compared to the balanced case, mainly because the equality between intercept and grand mean doesn't hold. The intercept needs to be adjusted based on the conditional probability of one of the levels (default to the reference level in the reference coding scheme). This adjustment of intercept requires some arithmetic calculation; nevertheless, in author's biased view, the complexity is still manageable and requires less calculation than the previous proposals. We demonstrate the simulation procedure for unbalanced design with a toy example. The simulation settings are similar to the above except that we change the group ratio to 8:2 and the effect size to 0.4.

To calculate the new intercept, we need first to establish the conditional probability for one of the levels (by default the $X = 0$ level in this example). As we know that the marginal probability can be expressed

$$Pr(Y=1) = \frac{n_1 Pr(Y=1|X=0) + n_2 Pr(Y=1|X=1)}{n_1+n_2} = \frac{n_1 Pr(Y=1|X=0) + n_2(Pr(Y=1|X=0) + RD}{n_1+n_2}$$

where RD is the effect size in risk difference, n_1 and n_2 are the group sample size for $X = 0$ and $X = 1$ respectively. Given $Pr(Y = 1), n_1, n_2$ and RD, we can derive easily derive the conditional probability of X=0,

$$Pr(Y=1|X=0) = \frac{(n_1+n_2)Pr(Y=1) - n_2 RD}{n_1+n_2}.$$

The intercept, a_0, as the mean of the group means can be calculated with

$$a_0 = \frac{2Pr(Y=1|X=0) + RD}{2}.$$

The simulation procedure translates to the toy example as

```
set.seed(123)

n <- 10000

# Marginal probabilities of each variable
p.y <- 0.3
p.x <- 0.8      # Imbalanced design
rd <- 0.2

cond.p <- (n*p.y - n*(p.x)*rd)/n
a.0 <- cond.p + rd/2


# Example 2: Generate L, X, and Y -------------------
X <- rbinom(n, 1, p.x)

# Generate X with marginal prob 0.5
dev_coding <- contr.sum(2)  # Deviation Coding with 2 levels
X_design_dev <- cbind( 1,   # Adding intercept column
                       dev_coding[X+1,]) # Construct the deisgn matrix

# The design matrix with effect coding can be more easily construct with model.matrix function

beta_vec <- c(a.0,   # Intercept term, the calculated mean of group means
              -rd/2) # Set up conditional prob for reference level
eta <- X_design_dev %*% beta_vec
eta[eta<0] <- 0
eta[eta>1] <- 1
Y <- rbinom(n, 1, eta)
```

A quick examination shows the simulated data matches with the expectation.

```r
mean(Y);
```

```
## [1] 0.2976
```

```r
mean(Y[X==0]);
```

```
## [1] 0.1380195
```

```r
mean(Y[X==1]);
```

```
## [1] 0.3362315
```

```r
# summary(glm(Y~X, family = binomial(link="identity"))) # Reference coding model
# summary(glm(Y~X_design_dev-1, family = binomial(link="identity"))) # Effect coding model
```

To note, if we ignore the group ratio change and don't adjust the intercept (for example, use the balanced-design simulation procedure without any modification), the observed marginal probability deviates from the target marginal probability and the deviation is more obvious with more extreme value of effect size and unbalanced group ratio.

Conclusion

In this report, we propose to use effect coding scheme to address the problem of controlling marginal probability of binary outcomes. We provide preliminary evidence that the proposal works for both balanced and unbalanced designs via toy examples. Compared to the previous solutions that based on reference coding scheme, our proposed solution requires less calculation to derive analytic and numeric approximation of the balance point. Particularly, it requires modest calculation when the study design is balanced.

In this report, we only consider easy simulation scenarios, i.e. binary covariates, one covariate, identify link function, to demonstrate the feasibility of this simulation strategy. We anticipate with the levels of a covariate and the number of covariates grow, the complexity of current calculation would grow. Nevertheless, we will provide a more delicate equation, particularly for the unbalanced design, to generalize for those situation. We will also conduct larger scale of simulation studies to evaluate the efficacy of the proposed solution.

References

R Core Team. 2021. *R: A Language and Environment for Statistical Computing.* Vienna, Austria: R Foundation for Statistical Computing. https://www.R-project.org/.

Robertson, Sarah E, Jon A Steingrimsson, and Issa J Dahabreh. 2021. "Using Numerical Methods to Design Simulations: Revisiting the Balancing Intercept." *American Journal of Epidemiology*, November, kwab264. https://doi.org/10.1093/aje/kwab264.

Rudolph, Jacqueline E, Jessie K Edwards, Ashley I Naimi, and Daniel J Westreich. 2021. "SIMULATION IN PRACTICE: THE BALANCING INTERCEPT." *American Journal of Epidemiology* 190 (8): 1696–98. https://doi.org/10.1093/aje/kwab039.