pprincrt: A Package for Design and Analysis of Cluster Randomized Trials with Power Prior Method

Shan Xiao

2017-08-12

Introduction

Cluster randomized trials are widely used in clinical trials (Campbell 2000; Campbell, Donner, and Klar 2007). With this design, clusters of participants in a trial are randomly assigned to different treatment arms, and all subjects within each cluster receive the same treatment. Trials outcomes, however, are typically assessed at the subject level, as in individual randomized trials. Cluster randomized trials are usually chosen for practical reasons (A. Donner 1998), such as convenience of implementation, or minimization of contamination, etc. But the practical appeal of cluster randomization is retained at the expense of a much reduced analytical power. This is because, in cluster randomized trials, the subjects within each cluster are correlated, and therefore, the variance of the effect size is increased (Hemming et al. 2011). Methods for improving analytical power of cluster randomized trials becomes a study of great practical importance. One proposed method is to borrow strength from similar historical data.

The idea of borrowing information from historical data is not new. Using information from previous trials of similar interventions to boost the power of the current trial is intuitively an appealing idea (Viele et al. 2014). Philosophically, such an approach is no different from meta analysis, which seeks to quantify an unknown treatment effect by combining all existing trials on the same or similar interventions (A. Donner, Piaggio, and Villar 2001; A. Donner and Klar 2002). Alternatively, one could approach the problem from the perspective of hierarchical Bayes, by formulating the prior distributions based on historical data (D. J. Spiegelhalter 2001; Turner, Omar, and Thompson 2001; Clark and Bachmann 2010). Specifically, a frequently used method is to determine the prior parameters from historical data (Goodman and Sladky 2005; Turner, Thompson, and Spiegelhalter 2005; B. P. Hobbs and Carlin 2007; Schoenfeld, Zheng, and Finkelstein 2009; Hampson et al. 2014). To prevent an overwhelming influence of the historical data, several researchers developed the idea of discounting the historical information. Magnitude of the discount is quantified by a power parameter (also discounting parameter) associated with the prior density, and thus leading to the concept of power prior (Duan 2005; Zhang 2010; Ibrahim et al. 2015).

An essential question concerning the use of power prior is to determine the discounting paramater. In the absence of a generally accepted mechanism for determination of this parameter, a reasonable approach is to adopt a data-driven method that directly evaluates the resemblance of the distributions of the data sources. In this research, we propose to use the Kullback-Leibler (KL) divergence measure to quantify the distance between the current and historical data, and then use this distance measure to determine the amount of discounting of historical data. A greater KL divergence indicates a larger discrepancy, and thus less incentive to place great weight on the historical information. To implement, we propose to use the KL divergence as the discounting parameter in a likelihood framework.

In this vignette, we give a brief introduction to our method in Section 2. Then in Section 3, we describe the structure of the package **pprincrt**. For the illustrative purpose of this package, we introduce two examples in Section 4. In Section 5, we demonstrate the use of the package **pprincrt**.

Method

This method could work in very general situations of cluster randomized trials, with multiple arms, balanced or imbalanced in sample size and multiple types of data following the exponential family of distributions. In principle, we could implement it with two steps. In the first step, we determine the discounting parameter with KL divergence measure. In the second step, we construct the power prior with the pre-determined discounting parameter, and then use the current data likelihood to update it through Bayes formula to get the posterior for estimation.

Determination of discounting parameter

We assume the current trial is a cluster randomized trial with P arms, including one placebo arm and $P-1$ treatment arms. The current trial data come from each subject, and we denote them as $\boldsymbol{D} = \{(Y_{ij}, X_{1i}, X_{2i}, \cdots, X_{P-1i}) : i = 1, 2, \cdots, I, j = 1, 2, \cdots, J_i\}$, where i and j are the cluster and subject indicators, respectively. Y_{ij} is the outcome of subject j in cluster i. $X_{1i}, X_{2i}, \cdots, X_{P-1i}$ are the $P-1$ dummy variables for the treatment status of cluster i. If cluster i is in the placebo arm, then all dummy variables take value 0. If cluster i is in treatment arm $p, p = 1, 2, \cdots, P-1$, then all dummy variables take value 0 except X_{pi}, which takes value 1. We assume the outcome Y_{ij} follows an exponential family distribution as,

$$f(Y_{ij}|b_i) = \exp\{\frac{Y_{ij}\eta_i - d(\eta_i)}{a(\phi)} + c(Y_{ij}, \phi)\},$$

where η_i is the natural parameter, and ϕ is the nuisance parameter. We let μ_i be the conditional mean of Y_{ij} given b_i, and it is related to η_i through a monotone, invertible link function $k(\cdot)$. We consider the generalized linear mixed model (GLMM) as below,

$$\eta_i = k(\mu_i) = \beta_0 + \boldsymbol{X}_i^T \boldsymbol{\beta} + b_i,$$

where $\boldsymbol{X}_i^T = (X_{1i}, X_{2i}, \ldots, X_{P-1i})$, $\boldsymbol{\beta} = (\beta_1, \beta_2, \ldots, \beta_{P-1})^T$ is a $(P-1) \times 1$ coefficient vector, and b_i is the ith cluster-specific random effect. We fit the GLMM under non-informative prior within Bayesian framework, and therefore evaluate the posterior of the treatment effects $\boldsymbol{\beta}$, denoted by $f(\boldsymbol{\beta}|\boldsymbol{D})$.

On the other hand, we assume the historical trial is a simple randomized trial with the same number of arms and the same type of outcome. The historical data, however, is summary data at the treatment group level. We denote them as $\boldsymbol{D_0} = \{(Z_l, N_l, X'_{1l}, X'_{2l}, \cdots, X'_{P-1l}) : l = 1, 2, \cdots, P\}$, where l is the treatment group indicator, N_l is the number of subjects in arm l, and Z_l is the sum of outcome in arm l. $Z_l = \sum_{i=1}^{N_l} Z_{il}$, where $Z_{il}, i = 1, 2, \cdots, N_l$ are the (unobserved) individual outcome in arm l. The probability density function of Z_{il} could be written as,

$$g(Z_{il}) = \exp\{\frac{Z_{il}\eta'_l - d(\eta'_l)}{a(\phi')} + c(Z_{il}, \phi')\},$$

where η'_l is the natural parameter, and ϕ' is the nuisance parameter. Therefore, the probability density function of Z_l is,

$$g(Z_l) = \int \cdots \int \exp\{\frac{Z_l\eta'_l - N_l d(\eta'_l)}{a(\phi')}\}\exp\{\sum_{i=1}^{N_l-1} c(Z_{il}, \phi') + c(Z_l - \sum_{i=1}^{N_l-1} Z_{il}, \phi')\}\, dZ_{1l} \ldots dZ_{N_l-1l}.$$

If the individual outcome is normal, count or binary, Z_l follows $\mathrm{N}(N_l\eta'_l, N_l a(\phi'))$, $\mathrm{Poisson}(N_l \exp(\eta'_l))$ or $\mathrm{Binomial}(N_l, \mathrm{logit}^{-1}(\eta'_l))$, respectively. We let μ'_l be the mean of Z_{il}, or equivalently the mean of $\frac{Z_l}{N_l}$, and it is related to η'_l through the monotone, invertible link function $k(\cdot)$. We consider the generalized linear model (GLM) as below,

$$\eta'_l = k(\mu'_l) = \beta'_0 + \boldsymbol{X}'^{T}_l \boldsymbol{\beta'},$$

where $\boldsymbol{X}'^{T}_l = (X'_{1l}, X'_{2l}, \ldots, X'_{P-1l})$, $\boldsymbol{\beta'} = (\beta'_1, \beta'_2, \ldots, \beta'_{P-1})^T$ is a $(P-1) \times 1$ coefficient vector. We fit the GLM model under non-informative prior in the framework of Bayesian statistics, and therefore evaluate the posterior of the treatment effects $\boldsymbol{\beta'}$, denoted by $g(\boldsymbol{\beta'}|\boldsymbol{D_0})$.

Since the two posteriors are derived under non-informative priors, they are not influenced by external data other than the current and historical trial information. From this, we ascertain the symmetric and asymmetric KL divergence measures, and use them to quantify the similarity of the two data sources.

$$D^{\mathrm{sym}}_{KL}(f||g) = \mathrm{E}_f\{\log(\frac{f}{g})\} + \mathrm{E}_g\{\log(\frac{g}{f})\},$$

$$D^{\mathrm{asym}}_{KL}(f||g) = \mathrm{E}_f\{\log(\frac{f}{g})\}.$$

where f and g stand for $f(\boldsymbol{\beta}|\boldsymbol{D})$ and $g(\beta'|\boldsymbol{D}_0)$, and $E_f\{\cdot\}$ and $E_g\{\cdot\}$ are the expectations taken with respect to f and g, respectively.

Under such a setup, we propose the following discounting fractions

$$a^{\mathrm{sym}} = e^{-D^{\mathrm{sym}}_{KL}(f||g)} = e^{-D^{\mathrm{sym}}_{KL}(g||f)},$$

$$a^{\mathrm{asym}} = e^{-D^{\mathrm{asym}}_{KL}(f||g)}.$$

The two KL divergence measures are not easy to compute by definition, we therefore use the k-Nearest Neighbor (k-NN) algorithm for calculation (Wang, Kulkarni, and Verdu 2009; Beygelzimer et al. 2013).

Estimation of treatment effect

We assume the pth $, p = 1, 2, \ldots, P-1$ treatments in the two trials are the same or similar, so it is reasonable to allow the historical strength to be borrowed through β'_p in the estimation of β_p. We let $\boldsymbol{\beta}_p = (\beta_p, \beta'_p)$, and we assume it follows a bivariate normal distribution $h_p(\boldsymbol{\beta}_p|\boldsymbol{m}_p, \boldsymbol{\Lambda}_p)$. We assume β_p and β'_p are exchangeable, then the mean vector $\boldsymbol{m}_p = \mu_p(1,1)'$, and the 2×2 precision matrix $\boldsymbol{\Lambda_p} = \tau_p \begin{bmatrix} 1 & \rho_p \\ \rho_p & 1 \end{bmatrix}^{-1}$. Furthermore, we assume μ_p, τ_p and ρ_p follow the hyper-priors as below,

$$h_p(\mu_p|a_p, R_p) = \sqrt{\frac{R_p}{2\pi}} e^{-\frac{R_p}{2}(\mu_p - a_p)^2},$$

$$h_p(\tau_p|\kappa_p, \nu_p) = \frac{\nu_p^{\kappa_p}}{\Gamma(\kappa_p)} \tau_p^{\kappa_p - 1} e^{-\nu_p \tau_p},$$

$$h_p(\rho_p|c_p, d_p) = \frac{\rho_p^{c_p - 1}(1-\rho_p)^{d_p - 1}}{B(c_p, d_p)},$$

where $\Gamma(\cdot)$ and $B(\cdot, \cdot)$ are respectively Gamma and Beta functions. Therefore, we could write the power prior of $\boldsymbol{\beta}$ as below,

$$h(\boldsymbol{\beta}|\boldsymbol{D}_0, a) \propto \int \cdots \int \{ \int (L(\beta_0', \boldsymbol{\beta}'|\boldsymbol{D}_0)^a g(\beta_0')\, d\beta_0' \} \prod_{p=1}^{P-1} \{ h_p(\boldsymbol{\beta_p}|\boldsymbol{m}_p, \boldsymbol{\Lambda}_p)$$
$$h_p(\mu_p|a_p, R_p) h_p(\tau_p|\kappa_p, \nu_p) h_p(\rho_p|c_p, d_p) \}\, d\boldsymbol{\beta}'\, d\boldsymbol{\mu}\, d\boldsymbol{\tau}\, d\boldsymbol{\rho},$$

where $\boldsymbol{\mu} = (\mu_1, \mu_2, \ldots, \mu_{P-1})$, $\boldsymbol{\tau} = (\tau_1, \tau_2, \ldots, \tau_{P-1})$ and $\boldsymbol{\rho} = (\rho_1, \rho_2, \ldots, \rho_{P-1})$, $L(\beta_0', \boldsymbol{\beta}'|\boldsymbol{D}_0)$ and $g(\beta_0')$ are the likelihood and prior density function of β_0' and $\boldsymbol{\beta}'$ in the GLM model. With this power prior, we could derive the posterior of $\boldsymbol{\beta}$ as below,

$$h(\boldsymbol{\beta}|\boldsymbol{D}, \boldsymbol{D}_0, a) \propto \int \cdots \int \{ L(\beta_0, \boldsymbol{\beta}|\boldsymbol{D}, \boldsymbol{b}) f(\boldsymbol{b}|\tau_b) f(\beta_0) f(\tau_b)\, d\boldsymbol{b}\, d\beta_0\, d\tau_b \} h(\boldsymbol{\beta}|\boldsymbol{D}_0, a),$$

where $L(\beta_0, \boldsymbol{\beta}|\boldsymbol{D}, \boldsymbol{b})$, $f(\boldsymbol{b}|\tau_b)$, $f(\beta_0)$ and $f(\tau_b)$ are the likelihood function and priors in the GLMM model.

Since the three posteriors $f(\boldsymbol{\beta}|\boldsymbol{D})$, $g(\boldsymbol{\beta}'|\boldsymbol{D}_0)$, $h(\boldsymbol{\beta}|\boldsymbol{D}, \boldsymbol{D}_0, a)$ do not have closed forms, we use a Metropolis Hastings (MH) algorithm within Gibbs sampling to draw samples from the posteriors (Gamerman 1997; Chib and Carlin 1999; D. Lunn et al. 2009; Robert and Casella 2009). Treatment effect estimates are then obtained from the appropriate summary statistics of the posterior samples.

Package structure

The **pprincrt** package has two main functions, *pprmodelBUGS*() and *SimPower*(). The former implements the Bayesian power prior analysis for cluster randomized trials, and the latter implements power calculation through simulation. The functions are able to incorporate historical trial information in both data analysis and trial design through power prior analysis. Additionally, the package provides two utility functions: *print.pprMod*(), which prints model fitting results in an easy to read format, and *AniPlot*(), which provides an animated presentation of estimated power and type 1 error rates.

Function *pprmodelBUGS()*

Function *pprmodelBUGS()* executes a series of tasks "under the hood", outlined in order of implementation below,

- Write the file *'cmodelfile.txt'* into the file directory. It is the BUGS script of the GLMM model for the current trial data.
- Use the function *bugs()* in the **R2OpenBUGS** package to call OpenBUGS from R and to analyze the current trial data with the model in *'cmodelfile.txt'* through batch mode (D. Spiegelhalter et al. 2007).
- Write the file *'hmodelfile.txt'* into the file directory. It is the BUGS script of the GLM model for the historical trial data.
- Use the function *bugs()* in the **R2OpenBUGS** package to call OpenBUGS from R and to analyze the historical trial data with the model in *'hmodelfile.txt'* through batch mode.
- Write the file *'pprmodelfile.txt'* into the file directory. It is the BUGS script of the power prior model for both current and historical trial data.

In function *pprmodelBUGS()*, the file directory is set to be *'tempdir()'* by default, but the users may change this through the argument *file.dir*. All files are removed after OpenBUGS is done if the argument *file.rm* is set to *TRUE*. However, the default value is set to *FALSE*. For the current trial, the data are input through the argument *cData*, while the analytical model is specified by the argument *cForm*. *cData* is a data frame, and *cForm* is a formula object. The variable names in *cData* must be the same as those in *cForm*. Besides, *hData* and *hForm* are similar arguments to *cData* and *cForm*, but for data input and model specification of the historical trial.

The current package runs on the Windows platform. The OpenBUGS directory could be specified by the user through the argument *OpenBUGS.dir*, which is the directory where the OpenBUGS software is installed. By default, it is set to be *NULL*, which means that the most recent OpenBUGS version registered in the Windows registry will be used. The pseudo random number generator in OpenBUGS has 14 different internal states. Each state is 10^{12} draws apart to avoid overlap in the pseudo random number sequences (D. Spiegelhalter et al. 2007). The state could be pre-specified through the argument *OpenBUGS.seed* by setting its value to be an interger between 1 and 14. By default, it is set to be 1.

In function *pprmodelBUGS()*, one MCMC chain of length $2,000$ is generated. We discard the first half of the generated values and we do not use thinning. The user may change the default values for the number of chains, the number of iteration, the number of burn-in and the thinning parameter through arguments *nchain*, *niter*, *nburnin*, *nthin*, respectively. Two different discounting methods are allowed: by the symmetric and asymmetric KL divergence based discounting parameters, or by an expert proposed proportion (between 0 and 1). The discounting method is specified by the argument *weight*.

Function *pprmodelBUGS()* returns an S3 class object *pprMod* containing the posterior samples of the treatment effect, which can be further processed with the **coda** package. In addition, the object contains summaries of the posterior samples, including the posterior median, standard deviation, and the highest posterior density (HPD) interval. The credible level of the HPD interval is specified by argument *cover.level*. An estimate of the discounting parameter is also included in the object. Convergence diagnostic statistics are also presented, including trace plot, and the Gelman-Rubin statistic and related plots if two or more MCMC chains are run. All diagnostic plots are contained in file *'graphics.pdf'* under the directory specified by the argument *file.dir*.

Function *SimPower()*

Function *SimPower()* is used to determine the sample size in designing a new cluster randomized trial with the proposed method given a historical data or a specific historical trial parameter setting. It may also be used to assess the power of the proposed method under a pre-specified parameter setting for both current and historical trials. The desired result is specified by argument *to.do.option*. It takes value *real_SSD*, *sim_SSD* or *sim_power* for the three aforementioned aims, respectively. When argument *to.do.option* takes value *real_SSD*, function *SimPower()* executes a series of tasks "under the hood", outlined in order of implementation below,

- Identify the pre-specified historical trial data.
- Generate multiple curren trial data replicates under an experimental sample size.
- Analyze each current trial data with power prior method by borrowing information from that historical trial data, and output the HPD interval of the treatment effect.
- Use the HPD interval to make a decision whether to reject the null hypothesis of non-significant treatment effect.
- Report the proportion of rejection among the decisions as power.

If the power returned by function *SimPower()* is not the target one, then we should try another sample size and repeat the process above with function *SimPower()* until the target power is obtained, and then we could report the corresponding sample size to people. When argument *to.do.option* takes value *sim_SSD*, function *SimPower()* works similarly. The only difference is, in the first step, we should generate one historical trial data with the pre-specified historical trial parameter setting, instead. When argument *to.do.option* takes value *sim_power*, the first three steps are changed to the following two steps,

- Generate multiple pairs of current and historical data replicates under the pre-specified current and historical trial setting.
- For each pair of data replicates, analyze the current trial data with power prior method by borrowing information from the historical trial data, and output the HPD interval of the treatment effect.

In this case, the power returned by function *SimPower()* is reported to people, directly.

As shown above, multiple replicates of current trial data or historical trial data might be generated. The number of data replicates is specified by argument *Rep*. The arguments *cRdmSeed.init* and *hRdmSeed.init* are used to specify the random seeds for the generation of the first current trial data and the first historical trial data, respectively. Both random seeds increase by 1 for each of the following trial data. In concept, the computations on different data replicates are similar and independent, so they could be done on different cores of the same computer simultaneously, thereby reducing the computational time. Therefore, we allow parallel computing via the **foreach** package in this function.

Multiple files will be yielded in the file directory as a result of the computation on each data replicate, such as *'hmodelfile.txt'*, *'cmodelfile.txt'*, *'pprmodelfile.txt'* and *'graphics.pdf'*. For the concern of space, all these files are removed by default after the computation is done. This default value could be changed by setting the argument *file.rm=FALSE*. Similar to function *pprmodelBUGS()*, the same arguments are provided in function *SimPower()* to specify the OpenBUGS directory, the MCMC chain features, the discounting parameter and the HPD interval. Additionally, when *to.do.option* takes value *real_SSD*, the historical trial data should be specified in the format of data frame through the argument *hData*, and the variable names in the data frame should be the same as those specified in the arguments *hTrtVar*, *hOutcomeVar* and *hSumVar*. When *to.do.option* takes value *sim_SSD* or *sim_power*, the historical trial parameter setting should be specified in the format of list through the argument *hSet*, and the variable names in the list should be the same as those specified in the arguments *hNarmVar*, *hNpat.armVar*, *hWthVar*, *hTrtVecVar*. Across the three aims, the current trial parameter setting should be specified in the format of list through the argument *cSet*, and the variable names in the list should be the same as those specified in the arguments *cNarmVar*, *cNcluster.armVar*, *cNpat.clusterVar*, *cBtwVar*, *cWthVar*, *cTrtVecVar*.

Function *print.pprMod()*

Function *print.pprMod()* is a new print method for objects of class *pprMod*, which are returned by function *pprmodelBUGS()*. It has one new argument *Open.plot*. This argument determines whether to open the file '*graphics.pdf*', which contains all convergence diagnostic plots. By default, the argument *Open.plot* is set to be *TRUE*.

Function *AniPlot()*

Function *AniPlot()* uses the function *saveLatex()* in the **animation** package to create animation plot. The software *Adobe Acrobat* and *MiKTeX* must be installed to use this function. This function is useful in visualizing four dimensional data. The data to plot is specified in the format of data frame through argument *x*. It has four variables, including the X-axis variable, Y-axis variable, group variable and the variable defining different frames in an animation plot, which is called frame variable. The name of the four variables should be the same as those specified in the arguments *XVar*, *YVar*, *GroupVar* and *FrameVar*, respectively. The limits of the X axis and Y axis in the animation plot are determined by the corresponding data ranges of the X-axis variable and the Y-axis variable, respectively. The data must be ordered by the group variable and the frame variable before it is used by function *AniPlot()*. The animation plot is output to a pdf file. The file name and directory could be specified by the arguments *imagename* and *AniPlot.dir*, respectively. Additionally, the time interval to play each frame in the animation plot could be set by argument *play.int*.

This utility function is used to visualize the results returned by function *SimPower()*. For example, if we obtain the values of power of different power prior methods under different values of current and historical treatment effects by using function *SimPower()*, then we can treat the historical treatment effect as the X-axis variable, the value of power as the Y-axis variable, the method as the group variable and the current treatment effect as the frame variable, and order the values of power by the method and the current treatment effect, then apply function *AniPlot()* to the ordered data to create an animation plot. This plot illustrates how the historical treatment effect impacts the power of different methods under different current treatment effects.

Examples

For the illustrative purpose of the functions above, we introduce two examples in this section. The first example includes two real HPV vaccine reminder trials: the Merck HPV vaccine reminder trial and the Szilagyi HPV vaccine reminder trial. First, we use function *pprmodelBUGS()* to analyze Merck trial data by borrowing information from Szilagyi trial data, and then use function *print.pprMod()* to print the results returned by *pprmodelBUGS()*. Second, we use function *SimPower()* to design a cluster randomized trial with Szilagyi trial data. Finally, we apply function *AniPlot()* to a non real data to create an animation plot.

HPV vaccine reminder trials

The HPV vaccine reminder trial sponsored by Merck pharmaceuticals is a cluster randomized trial that evaluates the effect of two reminder interventions on the uptake of the first dose of HPV vaccine in adolescents. In this trial, 28 physicians are recruited. They are randomized to one of the three arms: the placebo arm and the two intervention arms. The electronic interventions are directly delivered to the physicians, and they remind the physicians that some patients in his/her clinic could take the first dose of HPV vaccine. All eligible patients are recruited from the clinics of these physicians. A patient is eligible if he/she is $11 - 14$ years old, and he/she has not received HPV vaccine before. The number of patients are different from physician to physician (i.e. from cluster to cluster). At the end of this trial, the arm status and the uptake status of the first dose of HPV vaccine are collected from each patient. The data are available in data frame *cdata*. For illustrative purposes, we collapse the two intervention arms into one arm, and compare it to the control arm. We aggregate the data by each physician, as shown in Table 1.

Table 1: Merck HPV vaccine reminder trial

Intervention status	Physician ID	Number of subjects	Number of subjects taking vaccine
0	5	14	10
0	13	22	5
0	15	50	22
0	17	11	4
0	19	20	16
0	21	15	8
0	24	10	4
0	25	23	16
0	26	3	1
0	27	55	14
0	28	2	1
1	1	7	7
1	2	24	19
1	3	16	12
1	4	9	2
1	6	1	1
1	7	12	7

Intervention status	Physician ID	Number of subjects	Number of subjects taking vaccine
1	8	11	9
1	9	16	12
1	10	45	11
1	11	37	33
1	12	12	8
1	14	2	1
1	16	22	10
1	18	2	2
1	20	7	3
1	22	19	13
1	23	8	6

The HPV vaccine reminder trial conducted by Szilagyi et al. (2011) is a simple randomized trial that evaluates the effect of one intervention on the uptake of the first dose of HPV vaccine in adolescents. In this trial, $2,139$ patients are recruited, and they are randomized to one of two arms: the placebo arm and the reminder intervention arm. A patient is eligible if she is $11 - 15$ years old, and she has not taken HPV vaccine before. The data from this trial come in a summary fashion. We only know the total number of patients and the number of patients taking the vaccine in each arm. The data are available in data frame *hdata*, and shown in Table 2.

Table 2: Szilagyi HPV vaccine reminder trial

Intervention status	Number of subjects	Number of subjects taking vaccine
0	1055	453
1	1084	634

Animation data

The animation data is a non-real data that has a total of 72 observations. It has four different variables, including the group variable *group*, frame variable *frame*, X-axis variable x and Y-axis variable y. The group and frame variables have 2 and 6 unique values, respectively. This data has been ordered by *group* and *frame* already. The data are available in data frame *anidata*, and could be accessed by the users with the statement *anidata*.

Package demonstration

Power prior analysis with function *pprmodelBUGS()*

We treat the Merck and Szilagyi HPV vaccine reminder trials as the current and historical trials, respectively, and apply function *pprmodelBUGS()* to them to perform a power prior analysis. We use the discounting parameter estimated with asymmetric KL divergence measure. This is done as follows,

```
my.pprobject <- pprincrt::pprmodelBUGS(cData = pprincrt::cdata, cForm = y ~
    x + (1 | cl), hData = pprincrt::hdata, hForm = y | n ~ x, weight = "asym",
    family = "binomial", niter = 2500, nburnin = 1500, nthin = 5, nchain = 2)
```

It returns *my.pprobject*, which is a realization of the S3 object *pprMod*. Then we print it out as follows,

```
print(my.pprobject, open.plot = T)
```

```
## Discounting parameter value:
## [1] 0.4341808
## Summay of mcmc chain:
##                 median        sd 95% HPD Lower 95% HPD Upper
## trt2 v.s. trt1 0.7581 0.3788875       0.08777          1.58
## Convergence diagnostic statistics for mcmc chain:
## Potential scale reduction factors:
##
##                Point est. Upper C.I.
## trt2 v.s. trt1       1.03       1.03
##
## Convergence diagnostic plots for mcmc chain:
## [1] "C:\\Users\\shanxiao\\AppData\\Local\\Temp\\RtmpymygXt\\graphics.pdf"
```

Power prior design with function *SimPower()*

We assume the Szilagyi HPV vaccine reminder trial is the historical trial, and we apply the function *SimPower()* to it to design a cluster randomized trial. In this power prior design, we use the discounting parameter estimated by asymmetric KL divergence measure. The current trial setting is specified by *my.curset*. The following code estimates power for detecting a difference between *trt1* and *trt2*:

```
my.curset <- list(narm = 2, ncluster.arm = rep(10, 2), npat.cluster = rep(20,
    20), trt1 = -0.1, `trt2 v.s. trt1` = log(4), sigma.b = 1)
mypower <- pprincrt::SimPower(family = "binomial", to.do.option = "real_SSD",
    cSet = my.curset, cTrtVecVar = c("trt1", "trt2 v.s. trt1"), hData = pprincrt::hdata,
    weight = "asym", nchain = 1, niter = 2500, nburnin = 1500, nthin = 5, file.rm = T,
    Rep = 100)
mypower
```

```
## trt2 v.s. trt1
##              0.7
```

It returns the power of the design under the specified trial setting. For the purpose of illustration, we set *Rep* to be 100 to get a quick result. A larger value must be specified, such as $1,000$, if we want to obtain an accurate estimate of the power. Assuming the number of subjects within each cluster is fixed, we must increase the number of clusters within each arm and rerun the above code in the new trial setting if the returned power is smaller than our target, such as 0.8. Otherwise, we must decrease it and repeat the process until the target power is reached.

Animation plot with function *AniPlot()*

We apply the function *AniPlot()* to the animation data as follows,

```
pprincrt::AniPlot(pprincrt::anidata,imagename="AnimationImage"
                  ,AniPlot.dir="C:/Users/shanxiao/Desktop/pprincrt/vignettes")
```

```
## LaTeX document created at: C:/Users/shanxiao/Desktop/pprincrt/vignettes/animation.tex
```

```
## successfully compiled: pdflatex animation.tex
```

It generates a tex file, which is compiled into a pdf file containing the animation plot. Furthermore, the animation plot could be incorporated into this vignette file as follows,

```
\begin{figure}
\centering
\resizebox{0.8\textwidth}{!}{\begin{minipage}{\textwidth}
\animategraphics[controls,width=\linewidth]{1}
{C:/Users/shanxiao/Desktop/pprincrt/vignettes/AnimationImage}{}{}
\caption{\small{A animation plot}}
\end{minipage}}
\end{figure}
```

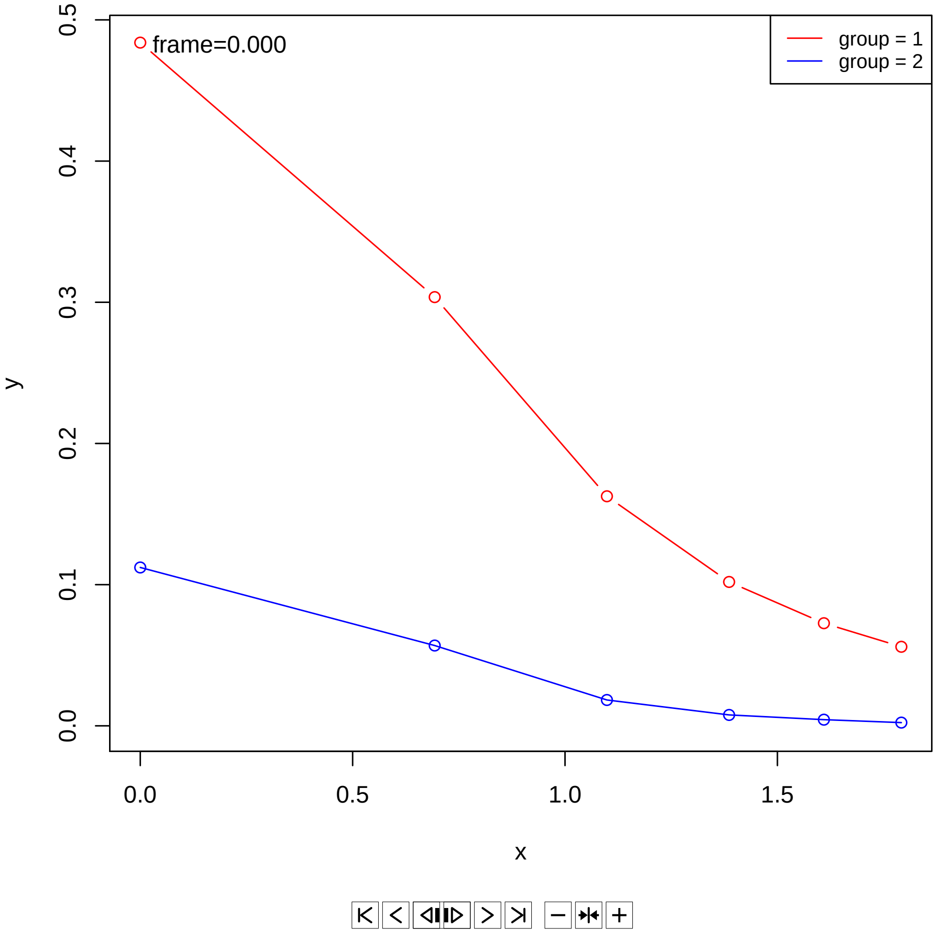

Figure 1: A animation plot

Ackowledgements

I thank Dr. Gregory D. Zimet for allowing me to use his HPV vaccine reminder trial data for the testing of our method.

References

Beygelzimer, A, S Kakadet, J Langford, S Arya, D Mount, and S Li. 2013. "FNN: Fast Nearest Neighbor Search Algorithms and Applications. R Package Version 1.1."

Campbell, MJ. 2000. "Cluster Randomized Trials in General (Family) Practice Research." *Statistical Methods in Medical Research* 9 (2): 81–94.

Campbell, MJ, A Donner, and N Klar. 2007. "Developments in Cluster Randomized Trials and Statistics in Medicine." *Statistics in Medicine* 26 (1): 2–19.

Chib, Siddhartha, and Bradley P Carlin. 1999. "On Mcmc Sampling in Hierarchical Longitudinal Models." *Statistics and Computing* 9 (1): 17–26.

Clark, Allan B, and Max O Bachmann. 2010. "Bayesian Methods of Analysis for Cluster Randomized Trials with Count Outcome Data." *Statistics in Medicine* 29 (2): 199–209.

Donner, Allan. 1998. "Some Aspects of the Design and Analysis of Cluster Randomization Trials." *Journal of the Royal Statistical Society: Series C (Applied Statistics)* 47 (1): 95–113.

Donner, Allan, and Neil Klar. 2002. "Issues in the Meta-Analysis of Cluster Randomized Trials." *Statistics in Medicine* 21 (19): 2971–80.

Donner, Allan, Gilda Piaggio, and Jose Villar. 2001. "Statistical Methods for the Meta-Analysis of Cluster Randomization Trials." *Statistical Methods in Medical Research* 10 (5): 325–38.

Duan, Yuyan. 2005. "A Modified Bayesian Power Prior Approach with Applications in Water Quality Evaluation." PhD thesis, Virginia Polytechnic Institute; State University.

Gamerman, Dani. 1997. "Sampling from the Posterior Distribution in Generalized Linear Mixed Models." *Statistics and Computing* 7 (1): 57–68.

Goodman, Steven N, and John T Sladky. 2005. "A Bayesian Approach to Randomized Controlled Trials in Children Utilizing Information from Adults: The Case of Guillain-Barre." *Clinical Trials* 2 (4): 305–10.

Hampson, Lisa V, John Whitehead, Despina Eleftheriou, and Paul Brogan. 2014. "Bayesian Methods for the Design and Interpretation of Clinical Trials in Very Rare Diseases." *Statistics in Medicine* 33 (24): 4186–4201.

Hemming, Karla, Alan J Girling, Alice J Sitch, Jennifer Marsh, and Richard J Lilford. 2011. "Sample Size Calculations for Cluster Randomised Controlled Trials with a Fixed Number of Clusters." *BMC Medical Research Methodology* 11 (1): 1.

Hobbs, Brian P, and Bradley P Carlin. 2007. "Practical Bayesian Design and Analysis for Drug and Device Clinical Trials." *Journal of Biopharmaceutical Statistics* 18 (1): 54–80.

Ibrahim, Joseph G, Ming-Hui Chen, Yeongjin Gwon, and Fang Chen. 2015. "The Power Prior: Theory and

Applications.” *Statistics in Medicine* 34 (28): 3724–49.

Lunn, David, David Spiegelhalter, Andrew Thomas, and Nicky Best. 2009. “The Bugs Project: Evolution, Critique and Future Directions.” *Statistics in Medicine* 28 (25): 3049–67.

Robert, Christian, and George Casella. 2009. *Introducing Monte Carlo Methods with R.* Springer Science & Business Media.

Schoenfeld, David A, Hui Zheng, and Dianne M Finkelstein. 2009. “Bayesian Design Using Adult Data to Augment Pediatric Trials.” *Clinical Trials* 6 (4): 297–304.

Spiegelhalter, D, A Thomas, N Best, and D Lunn. 2007. “OpenBUGS User Manual, Version 3.0. 2.” *MRC Biostatistics Unit, Cambridge.*

Spiegelhalter, David J. 2001. “Bayesian Methods for Cluster Randomized Trials with Continuous Responses.” *Statistics in Medicine* 20 (3): 435–52.

Szilagyi, Peter G, Sharon G Humiston, Sarah Gallivan, Christina Albertin, Martha Sandler, and Aaron Blumkin. 2011. “Effectiveness of a Citywide Patient Immunization Navigator Program on Improving Adolescent Immunizations and Preventive Care Visit Rates.” *Archives of Pediatrics & Adolescent Medicine* 165 (6): 547–53.

Turner, Rebecca M, Rumana Z Omar, and Simon G Thompson. 2001. “Bayesian Methods of Analysis for Cluster Randomized Trials with Binary Outcome Data.” *Statistics in Medicine* 20 (3): 453–72.

Turner, Rebecca M, Simon G Thompson, and David J Spiegelhalter. 2005. “Prior Distributions for the Intracluster Correlation Coefficient, Based on Multiple Previous Estimates, and Their Application in Cluster Randomized Trials.” *Clinical Trials* 2 (2): 108–18.

Viele, Kert, Scott Berry, Beat Neuenschwander, Billy Amzal, Fang Chen, Nathan Enas, Brian Hobbs, et al. 2014. “Use of Historical Control Data for Assessing Treatment Effects in Clinical Trials.” *Pharmaceutical Statistics* 13 (1): 41–54.

Wang, Qing, Sanjeev R Kulkarni, and Sergio Verdu. 2009. “Divergence Estimation for Multidimensional Densities via-Nearest-Neighbor Distances.” *IEEE Transactions on Information Theory* 55 (5): 2392–2405.

Zhang, Honglian. 2010. *Bayesian Power Prior Analysis and Its Application to Operational Risk and Rasch Model.* ERIC.