Package ‘RobustBayesianCopas’

May 12, 2020

Type Package

Title Robust Bayesian Copas Selection Model

Version 1.0

Date 2020-05-08

Author Ray Bai

Maintainer Ray Bai <raybaistat@gmail.com>

Description

Implementation of the robust Bayesian Copas (RBC) selection model for correcting and quantifying publication bias, as introduced in Bai et al. (2020) <arXiv:2005.02930>. This package also implements standard random effects meta-analysis and the Copas-like selection model of Ning et al. (2017) <doi:10.1093/biostatistics/kxx004>.

License GPL-3

LazyData true

Depends R (>= 3.6.0)

Imports stats, statip, rjags

NeedsCompilation yes

Repository CRAN

Date/Publication 2020-05-12 09:40:06 UTC

R topics documented:

antidepressants 2
Barlow2014 3
BayesNonBiasCorrected 3
CopasLikeSelection 5
D.measure 7
Hackshaw1997 9
RobustBayesianCopas 10
StandardMetaAnalysis 13

Index **16**

`antidepressants` *A Meta-Analysis on the Efficacy of Antidepressants*

Description

This data set contains 73 studies with results on the effectiveness of antidepressants that were reported to the FDA. However, only 50 of these studies were subsequently published. Since studies reported their outcomes on different scales, effect sizes were all expressed as standardized mean differences by means of Hedges' *g* scores, accompanied by corresponding variances. This data set was originally analyzed by Turner et al. (2008).

Usage

```
data(antidepressants)
```

Format

A dataframe with 73 studies with the following seven variables.

`Drug`**:** antidepressant name.

`Study`**:** study identifier.

`Published`**:** a binary variable indicating whether the study was published: "1"=published, "0"=not published.

`Nonstandardized_effect_size`**:** estimated mean improvement in depression symptoms (nonstandardized).

`Nonstandardized_SE`**:** estimated standard error (nonstandardized).

`Standardized_effect_size` estimated mean improvement in depression symptoms (standardized). Note that the standardized values are only available for the *published* studies (`NA` if not published). The *non*-missing data in this column should be used as `y` in the selection model.

`Standardized_SE`**:** estimated standard error (standardized). Note that the standardized values are only available for the *published* studies (`NA` if not published). The *non*-missing data in this column should be used as `s` in the selection model.

Source

Turner, E. H., Matthews, A. M., Linardatos, E., Tell, R. A., and Rosenthal, R. (2008). "Selective publication of antidepressant trials and its influence on apparent efficacy." *New England Journal of Medicine*, **358**(3):252-260.

`Barlow2014` *A Meta-Analysis on the Effect of Parent Training Programs vs. Control for Improving Parental Psychosocial Health Within 4 Weeks After Intervention*

Description

This data set contains 26 studies with the standardized mean differences between the improvement in parental psychosocial health for subjects who were enrolled in parent training programs vs. those who were not. This data set is also available in the `R` package `altmeta`, along with many other useful data sets.

Usage

```
data(Barlow2014)
```

Format

A dataframe with 26 studies with the following five variables within each study.

`y`**:** standardized mean differences in improvement of parental psychosocial health.

`s`**:** sample standard errors of standardized mean differences.

`n1`**:** sample sizes in treatment group 1 (parent training programs).

`n2`**:** sample sizes in treatment group 2 (control).

`n`**:** total sample size.

Source

Barlow J, Smailagic N, Huband N, Roloff V, Bennett C (2014). "Group-based parent training programmes for improving parental psychosocial health." *Cochrane Database of Systematic Reviews*,**5**, Art. No.: CD002020. <doi: 10.1002/14651858.CD002020.pub4>

`BayesNonBiasCorrected` *Non-bias-corrected robust Bayesian meta-analysis model*

Description

This function implements the *non*-bias-corrected Robust Bayesian Copas selection model of Bai et al. (2020) when there is no publication bias (i.e. $\rho = 0$). If $\rho = 0$, then there is no publication bias and the Copas selection model reduces to the standard random effects meta-analysis model:

$$y_i = \theta + \tau u_i + s_i \epsilon_i,$$

where y_i is the reported treatment effect for the ith study, s_i is the reported standard error for the ith study, θ is the population treatment effect of interest, $\tau > 0$ is a heterogeneity parameter, ϵ_i is distributed as $N(0, 1)$, and u_i and ϵ_i are independent.

For the *non*-bias-corrected model, we place noninformative priors on (θ, τ^2) (see Bai et al. (2020) for details). For the heterogeneity $u_i, i = 1, \ldots, n$, the default RBC approach uses heavy-tailed standard Cauchy priors $C(0, 1)$. However, we also give the option for using $u_i \sim N(0, 1), i = 1, \ldots, n$, for "conventional" meta-analysis.

Usage

```
BayesNonBiasCorrected(y, s, init = NULL, het.dist=c("Cauchy","normal"),
                      burn=10000, nmc=10000)
```

Arguments

`y`	an $n \times 1$ vector of reported treatment effects.
`s`	an $n \times 1$ vector of reported within-study standard errors.
`init`	optional initialization values for (θ, τ). If specified, they must be provided in this exact order. If they are not provided, the program estimates initial values from the data.
`het.dist`	Distribution for the heterogeneity $u_i, i = 1, \ldots, n$. The user may specify either Cauchy or normal priors on the hetereogeneity. The default is `Cauchy`.
`burn`	Number of burn-in samples. Default is `burn=10000`.
`nmc`	Number of posterior samples to save. Default is `nmc=10000`.

Value

The function returns a list containing the following components:

`theta.hat`	posterior mean for θ.
`theta.samples`	MCMC samples for θ after burn-in. Used for plotting the posterior for θ and performing inference of θ.
`tau.hat`	posterior mean for τ.
`tau.samples`	MCMC samples for τ after burn-in. Used for plotting the posterior for τ and performing inference of τ.

References

Bai, R., Lin, L., Boland, M. R., and Chen, Y. (2020). "A robust Bayesian Copas selection model for quantifying and correcting publication bias." *arXiv preprint arXiv:2005.02930.*

Examples

```
######################################
# Example on the Barlow2014 data set #
######################################
data(Barlow2014)
```

```
attach(Barlow2014)
# Observed treatment effect
y.obs = Barlow2014[,1]
# Observed standard error
s.obs = Barlow2014[,2]

####################################
# Fit the non-bias-corrected model #
####################################

# NOTE: Use default burn-in (burn=10000) and post-burn-in samples (nmc=10000)
RBCNoBias.mod = BayesNonBiasCorrected(y=y.obs, s=s.obs, burn=500, nmc=500)

# Point estimate for theta
theta.hat.RBCNoBias = RBCNoBias.mod$theta.hat
# Standard error for theta
theta.se.RBCNoBias = sd(RBCNoBias.mod$theta.samples)
# 95% posterior credible interval for theta
theta.cred.int = quantile(RBCNoBias.mod$theta.samples, probs=c(0.025,0.975))

# Display results
theta.hat.RBCNoBias
theta.se.RBCNoBias
theta.cred.int

# Plot the posterior for theta
hist(RBCNoBias.mod$theta.samples)
```

`CopasLikeSelection` *Copas-like selection model*

Description

This function performs maximum likelihood estimation (MLE) of $(\theta, \tau, \rho, \gamma_0, \gamma_1)$ using the EM algorithm of Ning et al. (2017) for the Copas selection model,

$$y_i | (z_i > 0) = \theta + \tau u_i + s_i \epsilon_i,$$

$$z_i = \gamma_0 + \gamma_1 / s_i + \delta_i,$$

$$corr(\epsilon_i, \delta_i) = \rho,$$

where y_i is the reported treatment effect for the ith study, s_i is the reported standard error for the ith study, θ is the population treatment effect of interest, $\tau > 0$ is a heterogeneity parameter, and u_i, ϵ_i, and δ_i are marginally distributed as $N(0,1)$, and u_i and ϵ_i are independent.

In the Copas selection model, y_i is published (selected) if and only if the corresponding propensity score z_i (or the propensity to publish) is greater than zero. The propensity score z_i contains two parameters: γ_0 controls the overall probability of publication, and γ_1 controls how the chance of publication depends on study sample size. The reported treatment effects and propensity scores are correlated through ρ. If $\rho = 0$, then there is no publication bias and the Copas selection model reduces to the standard random effects meta-analysis model.

This is called the "Copas-like selection model" because to find the MLE, the EM algorithm utilizes a latent variable m that is treated as missing data. See Ning et al. (2017) for more details. An alternative funtion for implementing the Copas selection model using a grid search for (γ_0, γ_1) is available in the R package metasens.

Usage

```
CopasLikeSelection(y, s, init = NULL, tol=1e-20, maxit=1000)
```

Arguments

y	an $n \times 1$ vector of reported treatment effects.
s	an $n \times 1$ vector of reported within-study standard errors.
init	optional initialization values for $(\theta, \tau, \rho, \gamma_0, \gamma_1)$. If specified, they must be provided in this exact order. If they are not provided, the program estimates initial values from the data.
tol	Convergence criterion for the Copas-like EM algorithm for finding the MLE. Default is tol=1e-20.
maxit	Maximum number of iterations for the Copas-like EM algorithm for finding the MLE. Default is maxit=1000.

Value

The function returns a list containing the following components:

theta.hat	MLE of θ.
tau.hat	MLE of τ.
rho.hat	MLE of ρ.
gamma0.hat	MLE of γ_0.
gamma1.hat	MLE of γ_1.
H	5×5 Hessian matrix for the estimates of $(\theta, \tau, \rho, \gamma_0, \gamma_1)$. The square root of the diagonal entries of H can be used to estimate the standard errors for $(\theta, \tau, \rho, \gamma_0, \gamma_1)$.
conv	"1" if the optimization algorithm converged, "0" if algorithm did not converge. If conv=0, then using H to estimate the standard errors may not be reliable.

References

Ning, J., Chen, Y., and Piao, J. (2017). "Maximum likelihood estimation and EM algorithm of Copas-like selection model for publication bias correction." *Biostatistics*, **18**(3):495-504.

Examples

```
#######################################
# Example on the Barlow2014 data set #
#######################################
data(Barlow2014)
attach(Barlow2014)
# Observed treatment effect
y.obs = Barlow2014[,1]
# Observed standard error
s.obs = Barlow2014[,2]

###################################
# Fit Copas-like selection model #
###################################
CLS.mod = CopasLikeSelection(y=y.obs, s=s.obs)

# Point estimate for theta
CLS.theta.hat = CLS.mod$theta.hat

# Use Hessian to estimate standard error for theta
CLS.Hessian = CLS.mod$H
# Standard error estimate for theta
CLS.theta.se = sqrt(CLS.Hessian[1,1]) # 0.02235201

# 95 percent confidence interval (0.3656911, 0.4533110)
CLS.interval = c(CLS.theta.hat-1.96*CLS.theta.se, CLS.theta.hat+1.96*CLS.theta.se)

# Display results
CLS.theta.hat
CLS.theta.se
CLS.interval

# Other parameters controlling the publication bias
CLS.mod$rho.hat
CLS.mod$gamma0.hat
CLS.mod$gamma1.hat
```

`D.measure` *D Measure for Quantifying Publication Bias*

Description

This function computes Bai's D measure for quantifying publication bias based on the robust Bayesian Copas (RBC) selection model. Let $\pi_{rbc}(\theta|y)$ be the posterior distribution for θ under the full RBC (bias-corrected) model, and let $\pi_{\rho=0}(\theta|y)$ be the posterior distribution for θ under the *non*-bias corrected model (when ρ is fixed at $\rho = 0$). The D measure is the Hellinger distance H between the bias-corrected and *non*-bias-corrected posteriors.

$$D = H(\pi_{rbc}(\theta|y), \pi_{\rho=0}(\theta|y)).$$

D is always between 0 and 1, with $D \approx 0$ indicating negligible publication bias and $D \approx 1$ indicating a very high magnitude of publication bias.

The posterior densities for $\pi_{rbc}(\theta|y)$ and $\pi_{\rho=0}(\theta|y)$ are approximated using MCMC samples. Numerical integration is used to compute the Hellinger distance between them.

Usage

```
D.measure(samples.RBCmodel, samples.nobiasmodel)
```

Arguments

`samples.RBCmodel`
: a vector of the MCMC samples from the RBC model. These can be obtained from the output of the function `RobustBayesianCopas`.

`samples.nobiasmodel`
: a vector of the MCMC samples from the non-bias-corrected model. These can be obtained from the output of the function `BayesNonBiasCorrected`.

Value

The function returns Bai's D measure, a value between 0 and 1. $D \approx 0$ means negligible publication bias, and $D \approx 1$ means a very high magnitude of publication bias.

References

Bai, R., Lin, L., Boland, M. R., and Chen, Y. (2020). "A robust Bayesian Copas selection model for quantifying and correcting publication bias." *arXiv preprint arXiv:2005.02930.*

Examples

```
######################################
# Example on the Barlow2014 data set #
######################################
data(Barlow2014)
attach(Barlow2014)
# Observed treatment effect
y.obs = Barlow2014[,1]
# Observed standard error
s.obs = Barlow2014[,2]

#############################################
# Compute D measure using posterior samples #
#############################################
# Fit RBC (bias-corrected) model
# NOTE: Use default burn-in (burn=10000) and post-burn-in samples (nmc=10000)
RBC.mod = RobustBayesianCopas(y=y.obs, s=s.obs, burn=500, nmc=500)
```

```
# Fit non-bias-corrected model
# NOTE: Use default burn-in (burn=10000) and post-burn-in samples (nmc=10000)
RBCNoBias.mod = BayesNonBiasCorrected(y=y.obs, s=s.obs, burn=500, nmc=500)

# Compute the D measure based on posterior samples
D = D.measure(RBC.mod$theta.samples, RBCNoBias.mod$theta.samples)


############################################
# Example on the antidepressants data set. #
# This is from Section 7.2 of the paper    #
# by Bai et al. (2020).                    #
############################################

# Load the full data
data(antidepressants)
attach(antidepressants)

# Extract the 50 published studies
published.data = antidepressants[which(antidepressants$Published==1),]

# Observed treatment effect
y.obs = published.data$Standardized_effect_size

# Observed standard error
s.obs = published.data$Standardized_SE


################################
# Compute the D measure for    #
# quantifying publication bias #
################################

# Fit RBC (bias-corrected) model
RBC.mod = RobustBayesianCopas(y=y.obs, s=s.obs)

# Fit non-biased-corrected model
RBCNoBias.mod = BayesNonBiasCorrected(y=y.obs, s=s.obs)

# Compute D measure using posterior samples
D = D.measure(RBC.mod$theta.samples, RBCNoBias.mod$theta.samples)
```

`Hackshaw1997` *A Meta-Analysis on the Relationship Between Second-hand Tobacco Smoke and Lung Cancer*

Description

This data set contains 37 studies analyzed by Hackshaw et al. (1997). Hackshaw et al. (1997) evaluated the risk of developing lung cancer in women who were lifelong nonsmokers but whose husbands smoked, compared to women whose husbands had never smoked.

Usage

```
data(Hackshaw1997)
```

Format

A dataframe with 37 studies with the following four variables within each study.

`Study`**:** study identifier.

`log_OR`**:** the reported log-odds ratio. Use this as the treatment effect in meta-analysis.

`SE`**:** the reported standard error.

`weight`**:** the percent weight that the study contributes to the pooled log-odds ratio.

Source

Hackshaw, A. K., Law, M. R., and Wald, N. J. (1997). "The accumulated evidence on lung cancer and environmental tobacco smoke." *BMJ*, **315**(7114):980-988.

`RobustBayesianCopas` *Robust Bayesian Copas selection model*

Description

This function implements the Robust Bayesian Copas selection model of Bai et al. (2020) for the Copas selection model,

$$y_i | (z_i > 0) = \theta + \tau u_i + s_i \epsilon_i,$$
$$z_i = \gamma_0 + \gamma_1 / s_i + \delta_i,$$
$$corr(\epsilon_i, \delta_i) = \rho,$$

where y_i is the reported treatment effect for the ith study, s_i is the reported standard error for the ith study, θ is the population treatment effect of interest, $\tau > 0$ is a heterogeneity parameter, and ϵ_i, and δ_i are marginally distributed as $N(0, 1)$ and u_i, and ϵ_i are independent.

In the Copas selection model, y_i is published (selected) if and only if the corresponding propensity score z_i (or the propensity to publish) is greater than zero. The propensity score z_i contains two parameters: γ_0 controls the overall probability of publication, and γ_1 controls how the probability of publication depends on study sample size. The reported treatment effects and propensity scores are correlated through ρ. If $\rho = 0$, then there is no publication bias and the Copas selection model reduces to the standard random effects meta-analysis model.

The RBC model places noninformative priors on $(\theta, \tau^2, \rho, \gamma_0, \gamma_1)$ (see Bai et al. (2020) for details). For the heterogeneity $u_i, i = 1, \ldots, n$, the default RBC approach uses heavy-tailed standard Cauchy priors $C(0, 1)$. However, we also give the option for using $u_i \sim N(0, 1), i = 1, \ldots, n$, for "conventional" meta-analysis.

Usage

```
RobustBayesianCopas(y, s, init=NULL, het.dist=c("Cauchy","normal"),
                    burn=10000, nmc=10000)
```

Arguments

y	an $n \times 1$ vector of reported treatment effects.
s	an $n \times 1$ vector of reported within-study standard errors.
init	optional initialization values for $(\theta, \tau, \rho, \gamma_0, \gamma_1)$. If specified, they must be provided in this exact order. If they are not provided, the program estimates initial values from the data.
het.dist	Distribution for the heterogeneity $u_i, i = 1, \ldots, n$. The user may specify either Cauchy or normal priors on the hetereogeneity. The default is `Cauchy`.
burn	Number of burn-in samples. Default is `burn=10000`.
nmc	Number of posterior samples to save. Default is `nmc=10000`.

Value

The function returns a list containing the following components:

theta.hat	posterior mean for θ.
theta.samples	MCMC samples for θ after burn-in. Used for plotting the posterior for θ and performing inference of θ.
tau.hat	posterior mean for τ.
tau.samples	MCMC samples for τ after burn-in. Used for plotting the posterior for τ and performing inference of τ.
rho.hat	posterior median for ρ.
rho.samples	MCMC samples for ρ after burn-in. Used for plotting the posterior for ρ and performing inference of ρ.
gamma0.hat	posterior median for γ_0.
gamma0.samples	MCMC samples for γ_0 after burn-in. Used for plotting the posterior for γ_0 and performing inference of γ_0.
gamma1.hat	posterior median for γ_1.
gamma1.samples	MCMC samples for γ_1 after burn-in. Used for plotting the posterior for γ_1 and performing inference of γ_1.

References

Bai, R., Lin, L., Boland, M. R., and Chen, Y. (2020). "A robust Bayesian Copas selection model for quantifying and correcting publication bias." *arXiv preprint arXiv:2005.02930.*

Examples

```
######################################
# Example on the Barlow2014 data set #
######################################
data(Barlow2014)
attach(Barlow2014)
# Observed treatment effect
y.obs = Barlow2014[,1]
# Observed standard error
s.obs = Barlow2014[,2]

#####################
# Fit the RBC model #
#####################
# NOTE: Use default burn-in (burn=10000) and post-burn-in samples (nmc=10000)
RBC.mod = RobustBayesianCopas(y=y.obs, s=s.obs, burn=500, nmc=500)

# Point estimate for rho
rho.hat.RBC = RBC.mod$rho.hat

# Plot posterior for rho
hist(RBC.mod$rho.samples)

# Point estimate for theta
theta.hat.RBC = RBC.mod$theta.hat
# Standard error for theta
theta.se.RBC = sd(RBC.mod$theta.samples)
# 95% posterior credible interval for theta
theta.cred.int = quantile(RBC.mod$theta.samples, probs=c(0.025,0.975))

# Display results
theta.hat.RBC
theta.se.RBC
theta.cred.int

# Plot the posterior for theta
hist(RBC.mod$theta.samples)


############################################
# Example on the antidepressants data set. #
# This is from Section 7.2 of the paper by #
# Bai et al. (2020).                       #
############################################
# Load the full data
data(antidepressants)
attach(antidepressants)
# Extract the 50 published studies
published.data = antidepressants[which(antidepressants$Published==1),]
# Observed treatment effect
y.obs = published.data$Standardized_effect_size
# Observed standard error
```

```
s.obs = published.data$Standardized_SE

#####################
# Fit the RBC model #
######################
RBC.mod = RobustBayesianCopas(y=y.obs, s=s.obs)

# Point estimate for rho
rho.hat.RBC = RBC.mod$rho.hat

# Plot posterior for rho
hist(RBC.mod$rho.samples)

# Point estimate for theta
theta.hat.RBC = RBC.mod$theta.hat
# 95% posterior credible interval for theta
theta.cred.int = quantile(RBC.mod$theta.samples, probs=c(0.025,0.975))
# Plot the posterior for theta
hist(RBC.mod$theta.samples)
```

StandardMetaAnalysis *Standard meta-analysis*

Description

This function performs maximum likelihood estimation (MLE) of (θ, τ) for the standard random effects meta-analysis model,

$$y_i = \theta + \tau u_i + s_i \epsilon_i,$$

where y_i is the reported treatment effect for the ith study, s_i is the reported standard error for the ith study, θ is the population treatment effect of interest, $\tau > 0$ is a heterogeneity parameter, and u_i and ϵ_i are independent and distributed as $N(0, 1)$.

Usage

```
StandardMetaAnalysis(y, s, init = NULL, tol=1e-10, maxit=1000)
```

Arguments

y	an $n \times 1$ vector of reported treatment effects.
s	an $n \times 1$ vector of reported within-study standard errors.
init	optional initialization values for (θ, τ). If specified, they must be provided in this order. If they are not provided, the program estimates initial values from the data.

`tol` Convergence criterion for the optimization algorithm for finding the MLE. Default is `tol=1e-10`.

`maxit` Maximum number of iterations for the optimization algorithm for finding the MLE. Default is `maxit=1000`.

Value

The function returns a list containing the following components:

`theta.hat` MLE of θ.

`tau.hat` MLE of τ.

`H` 2×2 Hessian matrix for the estimates of (θ, τ). The square root of the diagonal entries of H can be used to estimate the standard errors for (θ, τ).

`conv` "1" if the optimization algorithm converged, "0" if algorithm did not converge. If `conv=0`, then using H to estimate the standard errors may not be reliable.

References

Bai, R., Lin, L., Boland, M. R., and Chen, Y. (2020). "A robust Bayesian Copas selection model for quantifying and correcting publication bias." *arXiv preprint arXiv:2005.02930.*

Ning, J., Chen, Y., and Piao, J. (2017). "Maximum likelihood estimation and EM algorithm of Copas-like selection model for publication bias correction." *Biostatistics*, **18**(3):495-504.

Examples

```
############################################
# Example on the antidepressants data set. #
# This is from Section 7.2 of the paper by #
# Bai et al. (2020).                       #
############################################
# Load the full data
data(antidepressants)
attach(antidepressants)

# Extract the 50 published studies
published.data = antidepressants[which(antidepressants$Published==1),]
# Observed treatment effect
y.obs = published.data$Standardized_effect_size
# Observed standard error
s.obs = published.data$Standardized_SE

#################################
# Fit a standard meta-analysis  #
# that ignores publication bias #
#################################
SMA.mod = StandardMetaAnalysis(y=y.obs, s=s.obs)

# Point estimate for theta
SMA.theta.hat = SMA.mod$theta.hat
```

```
# Use Hessian to estimate standard error for theta
SMA.Hessian = SMA.mod$H
# Standard error estimate for theta
SMA.theta.se = sqrt(SMA.Hessian[1,1])
# 95 percent confidence interval
SMA.interval = c(SMA.theta.hat-1.96*SMA.theta.se, SMA.theta.hat+1.96*SMA.theta.se)

# Display results
SMA.theta.hat
SMA.theta.se
SMA.interval


######################################
# Example on the Barlow2014 data set #
######################################
data(Barlow2014)
attach(Barlow2014)
# Observed treatment effect
y.obs = Barlow2014[,1]
# Observed standard error
s.obs = Barlow2014[,2]

#################################
# Fit a standard meta-analysis  #
# that ignores publication bias #
#################################
SMA.mod = StandardMetaAnalysis(y=y.obs, s=s.obs)

# Point estimate for theta
SMA.theta.hat = SMA.mod$theta.hat

# Use Hessian to estimate standard error for theta
SMA.Hessian = SMA.mod$H
# Standard error estimate for theta
SMA.theta.se = sqrt(SMA.Hessian[1,1])

# 95 percent confidence interval
SMA.interval = c(SMA.theta.hat-1.96*SMA.theta.se, SMA.theta.hat+1.96*SMA.theta.se)

# Display results
SMA.theta.hat
SMA.theta.se
SMA.interval
```

Index

antidepressants, 2

Barlow2014, 3
BayesNonBiasCorrected, 3

CopasLikeSelection, 5

D.measure, 7

Hackshaw1997, 9

RobustBayesianCopas, 10

StandardMetaAnalysis, 13