

Package ‘GRANDpriv’

May 7, 2026

Type Package

Title Graph Release with Assured Node Differential Privacy

Version 0.1.3

Description Implements a novel method for privatizing network data using differential privacy. Provides functions for generating synthetic networks based on LSM (Latent Space Model), applying differential privacy to network latent positions to achieve overall network privatization, and evaluating the utility of privatized networks through various network statistics. The privatize and evaluate functions support both LSM and RDPG (Random Dot Product Graph). For generating RDPG networks, users are encouraged to use the 'randnet' package <<https://CRAN.R-project.org/package=randnet>>. For more details, see the "proposed method" section of Liu, Bi, and Li (2025) <[doi:10.48550/arXiv.2507.00402](https://doi.org/10.48550/arXiv.2507.00402)>.

License GPL (>= 3)

URL <https://github.com/lisqu0000/GRANDpriv>

BugReports <https://github.com/lisqu0000/GRANDpriv/issues>

Encoding UTF-8

RoxygenNote 7.3.2

Imports EnvStats, rmutil, RSpectra, diffpriv, truncnorm, randnet, igraph, HCD, transport

NeedsCompilation no

Author Suqing Liu [aut, cre],
Xuan Bi [aut],
Tianxi Li [aut]

Maintainer Suqing Liu <liusuqing0123@uchicago.edu>

Repository CRAN

Date/Publication 2025-08-22 17:50:07 UTC

Contents

| | |
|-----------------------------|---|
| GRANDpriv-package | 2 |
| GRAND.evaluate | 3 |
| GRAND.privatize | 5 |
| LSM.Gen | 7 |

Description

GRANDpriv (**Graph Release with Assured Node Differential privacy**) is an R package that implements a novel method for privatizing network data using differential privacy. The package provides functions for generating synthetic networks based on LSM (Latent Space Model), applying differential privacy to network latent positions to achieve overall network privatization, and evaluating the utility of privatized networks through various network statistics. The `privatize` and `evaluate` functions support both LSM and RDPG (Random Dot Product Graph). For generating RDPG networks, users are encouraged to use the **randnet** package.

Details

The package implements a two-step approach:

1. **Latent Position Estimation:** Estimates latent positions from the network structure using either LSM.PGD (Projected Gradient Descent, for LSM) or ASE (Adjacency Spectral Embedding, for RDPG)
2. **Multivariate Differential Privacy:** Applies DIP (Distribution-Invariant differential Privacy) mechanism to protect latent positions while preserving network utility

Main Functions:

- `LSM.Gen`: Generate Latent Space Model Network
- `GRAND.privatize`: GRAND Privatization of Network Data
- `GRAND.evaluate`: GRAND Evaluation of Network Data

Key Features:

- Network Generation: Generate synthetic networks using Latent Space Models
- Differential Privacy: Apply node-level differential privacy to network data
- Multiple Models: Support for both LSM and RDPG models
- Comprehensive Evaluation: Evaluate utility through multiple network statistics
- Flexible Privacy Budgets: Support for multiple privacy levels in a single run

The package is designed for researchers and practitioners working with sensitive network data who need to balance privacy protection with data utility.

Author(s)

Suqing Liu [aut, cre], Xuan Bi [aut], Tianxi Li [aut]

Maintainer: Suqing Liu <<liusuqing0123@uchicago.edu>>

References

- P. D. Hoff, A. E. Raftery, and M. S. Handcock. Latent space approaches to social network analysis. *Journal of the American Statistical Association*, 97(460):1090–1098, 2002.
- S. J. Young and E. R. Scheinerman. Random dot product graph models for social networks. In *International Workshop on Algorithms and Models for the Web-Graph*, pages 138–149. Springer, 2007.
- Z. Ma, Z. Ma, and H. Yuan. Universal latent space model fitting for large networks with edge covariates. *Journal of Machine Learning Research*, 21(4):1–67, 2020.
- A. Athreya, D. E. Fishkind, M. Tang, C. E. Priebe, Y. Park, J. T. Vogelstein, K. Levin, V. Lyzinski, Y. Qin, and D. L. Sussman. Statistical inference on random dot product graphs: a survey. *Journal of Machine Learning Research*, 18(226):1–92, 2018.
- P. Rubin-Delanchy, J. Cape, M. Tang, and C. E. Priebe. A statistical interpretation of spectral embedding: The generalised random dot product graph. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 84(4):1446–1473, 2022.
- X. Bi and X. Shen. Distribution-invariant differential privacy. *Journal of Econometrics*, 235(2):444–453, 2023.
- S. Liu, X. Bi, and T. Li. GRAND: Graph Release with Assured Node Differential Privacy. arXiv preprint arXiv:2507.00402, 2025.

GRAND.evaluate

GRAND Evaluation of Network Data

Description

Evaluates the quality of GRAND privatization results by comparing various network statistics between the original and privatized networks using Wasserstein distance.

Usage

```
GRAND.evaluate(  
  result,  
  statistics = c("degree", "vshape", "triangle", "eigen", "harmonic")  
)
```

Arguments

| | |
|-------------------------|--|
| <code>result</code> | List. Output from GRAND.privatize function containing privatization results. |
| <code>statistics</code> | Character vector. Network statistics to evaluate. Options include: "degree" (Node Degree), "vshape" (V-Shape Count), "triangle" (Triangle Count), "eigen" (Eigen Centrality), "harmonic" (Harmonic Centrality). Default is all statistics. |

Details

GRAND Evaluation of Network Data

This function evaluates privatization quality by comparing network statistics between original and privatized networks using Wasserstein-1 distance. The evaluation covers:

- **Hat**: Performance on release set (without privacy)
- **Hat2**: Performance on holdout set (without privacy)
- **GRAND**: Performance of GRAND privatization method
- **Laplace**: Performance of standard Laplace mechanism

Lower Wasserstein distances indicate better utility preservation.

For more details, see the "simulation experiments" section of Liu, Bi, and Li (2025).

Value

A data frame containing evaluation results with columns:

- **stat**: Type of network statistic(s) evaluated
- **eps**: Privacy budget parameter(s) used
- **Hat**: Wasserstein distance for release set estimation
- **Hat2**: Wasserstein distance for holdout set estimation
- **GRAND**: Wasserstein distance for GRAND privatization method
- **Laplace**: Wasserstein distance for standard Laplace mechanism

References

S. Liu, X. Bi, and T. Li. GRAND: Graph Release with Assured Node Differential Privacy. arXiv preprint arXiv:2507.00402, 2025.

Examples

```
# Generate and privatize a network
network <- LSM.Gen(n = 400, k = 2, K = 3, avg.d = 40)
result <- GRAND.privatize(A = network$A, K = 2, idx = 1:200, eps = c(1, 2, 5, 10), model = "LSM")
# Evaluate results for all statistics
evaluation <- GRAND.evaluate(result)
# Evaluate only degree and triangle statistics
evaluation_subset <- GRAND.evaluate(result, statistics = c("degree", "triangle"))
```

GRAND.privatize *GRAND Privatization of Network Data*

Description

Applies the GRAND (Graph Release with Assured Node Differential privacy) method to privatize network data using differential privacy. The method estimates latent positions from the network and applies multivariate differential privacy to protect sensitive information.

Usage

```
GRAND.privatize(
  A,
  K,
  idx,
  eps = 1,
  model = c("LSM", "RDPG"),
  niter = 500,
  rho = 0.05,
  verbose = TRUE
)
```

Arguments

| | |
|---------|--|
| A | Matrix. Adjacency matrix of the input network. |
| K | Integer. Dimension of the latent space for network embedding. |
| idx | Integer vector. Indices of nodes to be privatized. |
| eps | Numeric or numeric vector. Privacy budget parameter(s) for differential privacy. Default is 1. |
| model | Character. Model type, either "LSM" (Latent Space Model) or "RDPG" (Random Dot Product Graph). Default is "LSM". |
| niter | Integer. Number of iterations for the optimization algorithm. Default is 500. |
| rho | Numeric. Parameter controlling the neighborhood size for conditional distributions. Default is 0.05. |
| verbose | Logical. Whether to print progress messages. Default is TRUE. |

Details

GRAND Privatization of Network Data

The GRAND privatization algorithm consists of the following steps:

1. **Network partitioning:** Split nodes into release set (`idx`) and holdout set
2. **Latent position estimation:** Use LSM.PGD (Projected Gradient Descent) or ASE (Adjacency Spectral Embedding) to estimate latent positions

3. **Differential privacy:** Apply multivariate DIP (Distribution-Invariant differential Privacy) mechanism to protect latent positions
4. **Network reconstruction:** Generate privatized networks from perturbed latent positions

The method also computes standard Laplace mechanism results for comparison.

For more details, see the "proposed method" section of Liu, Bi, and Li (2025).

Value

A list containing:

- `non.private.result`: Results without privacy, including the original and estimated data
- `GRAND.result`: List with one element per epsilon value. Each element contains GRAND privatization results for that specific epsilon
- `Laplace.result`: List with one element per epsilon value. Each element contains baseline Laplace mechanism results for that specific epsilon
- `eps`: Vector of privacy budget parameter(s) used

References

- P. D. Hoff, A. E. Raftery, and M. S. Handcock. Latent space approaches to social network analysis. *Journal of the American Statistical Association*, 97(460):1090–1098, 2002.
- S. J. Young and E. R. Scheinerman. Random dot product graph models for social networks. In *International Workshop on Algorithms and Models for the Web-Graph*, pages 138–149. Springer, 2007.
- Z. Ma, Z. Ma, and H. Yuan. Universal latent space model fitting for large networks with edge covariates. *Journal of Machine Learning Research*, 21(4):1–67, 2020.
- A. Athreya, D. E. Fishkind, M. Tang, C. E. Priebe, Y. Park, J. T. Vogelstein, K. Levin, V. Lyzinski, Y. Qin, and D. L. Sussman. Statistical inference on random dot product graphs: a survey. *Journal of Machine Learning Research*, 18(226):1–92, 2018.
- P. Rubin-Delanchy, J. Cape, M. Tang, and C. E. Priebe. A statistical interpretation of spectral embedding: The generalised random dot product graph. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 84(4):1446–1473, 2022.
- X. Bi and X. Shen. Distribution-invariant differential privacy. *Journal of Econometrics*, 235(2):444–453, 2023.
- S. Liu, X. Bi, and T. Li. GRAND: Graph Release with Assured Node Differential Privacy. arXiv preprint arXiv:2507.00402, 2025.

Examples

```
# Generate a sample network
network <- LSM.Gen(n = 400, k = 2, K = 3, avg.d = 40)
# Privatize the first 200 nodes with epsilon = 1, 2, 5, 10
result <- GRAND.privatize(A = network$A, K = 2, idx = 1:200, eps = c(1, 2, 5, 10), model = "LSM")
```

LSM.Gen

*Generate Latent Space Model Network***Description**

Generates a random network following LSM (Latent Space Model) with specified parameters. LSM assumes that each node has a latent position in a k -dimensional space, and edge probabilities depend on the inner products between these latent positions.

Usage

LSM.Gen(n , k , K , avg.d = NULL)

Arguments

| | |
|-------|---|
| n | Integer. Number of nodes in the network. |
| k | Integer. Dimension of the latent space. |
| K | Integer. Number of communities/groups. |
| avg.d | Numeric. Target average node degree. If NULL, no degree adjustment is performed. Default is "NULL". |

Details

Generate Latent Space Model Network

The Latent Space Model generates networks based on the following process:

1. Node-specific intercept parameters: $\alpha_i \sim \text{Unif}(1, 3)$, then transformed as $\alpha_i = -\alpha_i/2$
2. Community assignments: Each node is randomly assigned to one of K communities with equal probability
3. Community centers: μ_g sampled uniformly from $[-1, 1]^k$ hypercube
4. Latent positions: $Z_i \sim \mathcal{N}_{[-2,2]}(\mu_{g_i}, I_k)$ where μ_{g_i} is the community center for node i 's community
5. Edge probabilities:

$$P_{ij} = \text{logit}^{-1}(\alpha_i + \alpha_j + Z_i^\top Z_j) = \frac{1}{1 + \exp(-(\alpha_i + \alpha_j + Z_i^\top Z_j))}$$

6. Adjacency matrix: $A_{ij} \sim \text{Bern}(P_{ij})$ for $i < j$, with $A_{ii} = 0$ and $A_{ji} = A_{ij}$

If avg.d is specified, the edge probabilities are scaled to achieve the target average node degree.

For more details, see the "simulation studies" section of Ma, Ma, and Yuan (2020), noting that our implementation has slight differences.

Value

A list containing:

- A: Adjacency matrix of the generated network
- g: Graph object of the generated network
- P: Probability matrix of the generated network
- alpha: Node-specific intercept parameters
- Z: Latent positions in k-dimensional space
- idx: Community assignments for each node

References

P. D. Hoff, A. E. Raftery, and M. S. Handcock. Latent space approaches to social network analysis. *Journal of the American Statistical Association*, 97(460):1090–1098, 2002.

Z. Ma, Z. Ma, and H. Yuan. Universal latent space model fitting for large networks with edge covariates. *Journal of Machine Learning Research*, 21(4):1–67, 2020.

S. Liu, X. Bi, and T. Li. GRAND: Graph Release with Assured Node Differential Privacy. *arXiv preprint arXiv:2507.00402*, 2025.

Examples

```
# Generate a network with 400 nodes, 2D latent space, 3 communities
network <- LSM.Gen(n = 400, k = 2, K = 3)
# Generate with target average degree of 40
network2 <- LSM.Gen(n = 400, k = 2, K = 3, avg.d = 40)
```

Index

* **package**

GRANDpriv-package, [2](#)
_PACKAGE (GRANDpriv-package), [2](#)

GRAND.evaluate, [2](#), [3](#)
GRAND.privatize, [2](#), [5](#)
GRANDpriv (GRANDpriv-package), [2](#)
GRANDpriv-package, [2](#)

LSM.Gen, [2](#), [7](#)