Readme

This online supplement contains computer codes used in the paper:

Xu, P.F., Guo, J.H. and Tang, ML. (2013). A localized implementation of the iterative proportional scaling procedure for Gaussian graphical models. *Journal of Computational and Graphical Statistics*.

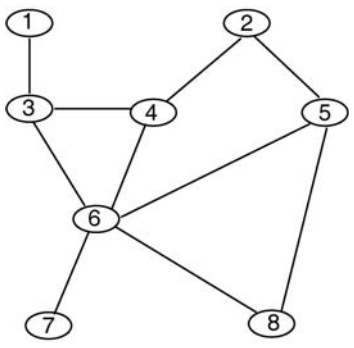
First, we introduce the representation of a graph in C++ codes. Then we will give a brief description of each file.

(1) Representation of graph

A graph is represented by its adjacent lists. For example, the moral graph of Bayesian network Asia (Lauritzen and Spiegelhalter, 1988) is in the file "MoralGraphAsia.txt". The first two numbers are the number of vertices and the number of edges. Then we list each vertex and its neighbors, and 0. In our file, 0 represents an end of a list.

```
8 10
1 3 0
2 4 5 0
3 1 4 6 0
4 2 3 6 0
5 2 6 8 0
6 3 4 5 7 8 0
7 6 0
8 5 6 0
```

MoralGraphAsia.txt



The moral graph of Asia network.

(2) Description of each file

(a) The file “graph.h” is a C++ Header file that gives some programs for graph operations, for example, finding all cliques of a given graph, constructing a minimal fill-in triangulation by MCS-M, constructing a junction tree of the triangulation, etc.

(b) In the folder “graphs” are some graphs generated randomly.

(c) “iips.m” is the IIPS procedure for adjusting the covariance matrix

```
function [Sigma, iter] = iips(GraphName, samples, threshold)
%IIPS Compute the mle of Gaussian graphical model proposed by Xu, Guo and
%    He (2011) An Improved Iterative Proportional Scaling Procedure for
%    Gaussian Graphical Models, 20, 417--431.
%[Sigma, iter] = iips(GraphName, samples, threshold)
%
%Input:
%   GraphName: a string which is the name of the file containing the graph
```

```
%   samples: numeric matrix. Columns are for variables and rows are for
%           samples.
%   threshold: threshold for convergence. Default value is 1e-5.
%             Iterations stop when max(max(abs(Sigma^{t+1}-Sigma^{t})))
%             is less than threshold
%Output:
%   Sigma: the mle of covariance matrix of the model
%   iter:  the total number of interactions
```

(d) “iht.m” is the IHT procedure for adjusting the inverse covariance matrix

```
function [omega, iter] = iht(GraphName, samples, threshold)
%IHT Compute the mle of Gaussian graphical model proposed by Xu, Guo and
%    Tang (2012) An improved Hara-Takamura procedure by sharing
%    computations on junction tree in Gaussian graphical models,
%    Statistics and Computing, 22(5), 1125--1133.
%[omega, iter] = iht(GraphName, samples, threshold)
%
%Input:
%   GraphName: a string which is the name of the file containing the graph
%   samples: numeric matrix. Columns are for variables and rows are for
%           samples.
%   threshold: threshold for convergence. Default value is 1e-5.
%             Iterations stop when max(max(abs(Omega^{t+1}-Omega^{t})))
%             is less than threshold
%Output:
%   Omega: the mle of inverse of covariance matrix of the model
%   iter:  the total number of interactions
```

(e) The file “RandomSigmaSamples.m” is used to construct normal distributions that satisfy Markov property with respect to a given graph.

(f) “ipsp1_rec.m” is the IPSP1 procedure for adjusting the covariance matrix

```
function [sigma, iter, timeSA, newmap, newnparts, newmul, newpartMul oldmul,
oldpartMul] = ipsp1_rec(GraphName, samples, initnparts, sa_input,
threshold)
% IPSP1 Compute the mle of Gaussian graphical model proposed by Xu, Guo
%    and Tang (2013) A localized implementation of the iterative
%    proportional scaling procedure for Gaussian graphical models
% [sigma, iter, timeSA, newmap, newnparts, newmul, newpartMul oldmul,
% oldpartMul] = ipsp1_rec(GraphName, samples, initnparts, sa_input,
% threshold)
%
%Input:
```

```
%   GraphName: a string which is the name of the file containing the graph
%   samples: numeric matrix. Columns are for variables and rows are for
%           samples.
%   initnparts: The number of blocks to partition the graph.
%   sa_input: the parameter for simulated annealing algorithm
%   threshold: threshold for convergence. Default value is 1e-5.
%             Iterations stop when max(max(abs(sigma^{t+1}-sigma^{t})))
%             is less than threshold
%Output:
%   sigma: the mle of covariance matrix of the model
%   iter:  the total number of interactions
```

(g) “ipsp2_rec.m” is the IPSP1 procedure for adjusting the covariance matrix

```
function [omega, iter, timeSA, newmap, newnparts, newmul, newpartMul oldmul,
oldpartMul] = ipsp2_rec(GraphName, samples, initnparts, sa_input,
threshold)
%IPSP2 Compute the mle of Gaussian graphical model proposed by Xu, Guo and
%      Tang (2013) A localized implementation of the iterative proportional
%      scaling procedure for Gaussian graphical models
% [omega, iter, timeSA, newmap, newnparts, newmul, newpartMul oldmul,
% oldpartMul] = ipsp2_rec(GraphName, samples, initnparts, sa_input,
% threshold)
%
%Input:
%   GraphName: a string which is the name of the file containing the graph
%   samples: numeric matrix. Columns are for variables and rows are for
%           samples.
%   initnparts: The number of blocks to partition the graph.
%   sa_input: the parameter for simulated annealing algorithm
%   threshold: threshold for convergence. Default value is 1e-5.
%             Iterations stop when max(max(abs(omega^{t+1}-omega^{t})))
%             is less than threshold
%Output:
%   omega: the mle of inverse covariance matrix of the model
%   iter:  the total number of interactions
```

(h) “iipsmp.m” is the IIPSMP procedure for adjusting the covariance matrix

```
function [Sigma, timeSA, iter] = iipsmp(GraphName, samples, threshold,
similarity, partOrNot)
%IIPSMP Compute the mle of Gaussian graphical model proposed by Xu, Guo and
%      Tang (2013) A localized implementation of the iterative proportional
%      scaling procedure for Gaussian graphical models
% [Sigma, timeSA, iter] = iipsmp(GraphName, samples, threshold, similarity,
% partOrNot)
%
%Input:
```

```
%   GraphName: a string which is the name of the file containing the graph
%   samples: numeric matrix. Columns are for variables and rows are for
%            samples.
%   threshold:  threshold for convergence. Default value is 1e-5.
%               Iterations stop when max(max(abs(Sigma^{t+1}-Sigma^{t})))
%               is less than threshold
%   similarity: if similarity > 0 && similarity < 1, we merge nodes,
%               otherwise we do not merge
%   partOrNot:  if partOrNot == 1, we partition the cliques in each node,
%               otherwis we do not partition
%Output:
%   Sigma: the mle of covariance matrix of the model
%   timeSA: the cputime of Simulated annealing for choosing optimal nparts
%   when partOrNot = 1
%   iter:  the total number of interactions
```

(i) "ihtmp.m" is the IHTMP procedure for adjusting the inverse covariance matrix

```
function [omega, timeSA, iter] = ihtmp(GraphName, samples, threshold,
similarity, partOrNot)
%IHTMP Compute the mle of Gaussian graphical model proposed by Xu, Guo and
%      Tang (2013) A localized implementation of the iterative proportional
%      scaling procedure for Gaussian graphical models
% [Omega, timeSA, iter] = ihtmp(GraphName, samples, threshold, similarity,
% partOrNot)
%
%Input:
%   GraphName: a string which is the name of the file containing the graph
%   samples: numeric matrix. Columns are for variables and rows are for
%            samples.
%   threshold:  threshold for convergence. Default value is 1e-5.
%               Iterations stop when max(max(abs(Sigma^{t+1}-Sigma^{t})))
%               is less than threshold
%   similarity: if similarity > 0 && similarity < 1, we merge nodes,
%               otherwise we do not merge
%   partOrNot:  if partOrNot == 1, we partition the cliques in each node,
%               otherwis we do not partition
%Output:
%   Omega: the mle of inverse covariance matrix of the model
%   timeSA: the cputime of Simulated annealing for choosing optimal nparts
%   when partOrNot = 1
%   iter:  the total number of interactions
```

(j) "sigmaHatDecomp.m" is the procedure for adjusting the covariance matrix by decompose the graph G to its maximal prime subgraphs

```
function sigma = sigmaHatDecomp(GraphName, samples, fun)
%sigmaHatDecomp Compute the mle of Gaussian graphical model proposed by Xu,
```

```
Guo and
%       Tang (2013) A localized implementation of the iterative proportional
%       scaling procedure for Gaussian graphical models
% Sigma = sigmaHatDecomp(GraphName, samples, fun)
%
% Input:
%   GraphName: a string which is the name of the file containing the graph
%   samples: numeric matrix. Columns are for variables and rows are for
%            samples.
%   fun: fun is a string which is one of 'iipsmp', 'iipsm', 'ipsp1, and 'iips'
% Output:
%   Sigma: the mle of covariance matrix of the model
```

(k) "omegaHatDecomp.m" is the procedure for adjusting the inverse covariance matrix by decompose the graph G to its maximal prime subgraphs

```
function omega = omegaHatDecomp(GraphName, samples, fun)
%omegaHatDecomp Compute the mle of Gaussian graphical model proposed by Xu,
Guo and
%       Tang (2013) A localized implementation of the iterative proportional
%       scaling procedure for Gaussian graphical models
% Omega = omegaHatDecomp(GraphName, samples, fun)
%
% Input:
%   GraphName: a string which is the name of the file containing the graph
%   samples: numeric matrix. Columns are for variables and rows are for
%            samples.
%   fun: fun is a string which is one of 'ihtmp', 'ihtm', 'ipsp2, and 'iht'
% Output:
%   Omega: the mle of inverse covariance matrix of the model
```

(l) The file "example.m" shows how to use above functions.