Journal of Software for Algebra and Geometry

```
                        gap> g:= SymmetricGroup( 4 );
                        Sym( [ 1 .. 4 ] )
i5 : betti(t,Weights=>{1,0})
                        gap> tbl:= CharacterTable( g );;  HasIrr( tbl );
                        false
        0 1  2  3 4     gap> tblmod2:= CharacterTable( tbl, 2 );
o5 = total: 1 4 13 14 4 BrauerTable( Sym( [ 1 .. 4 ] ), 2 )
        0: 1  .  .  . .
        1: . 2  2  4 2  gap> tblmod2 = CharacterTable( tbl, 2 );
        2: . 2  5  6 .  true
        3: . . 4  . 2
        4: . . . 4 .    gap> tblmod2 = BrauerTable( tbl, 2 );
        5: . . 2 . .    true
o5 : Be                 gap> tblmod2 = BrauerTable( 2 )
i6 : betti(t,Weights=>{0, true

        0 1  2  3 4     gap> libtbl:= CharacterTable( "M" );
o6 = total: 1 4 13 14 4 CharacterTable( "M" )
        0: 1  .  .  . .  ap> CharacterTableRegular( libtbl, 2 );
        2: . 2  5  6 .   raue ab  "M" )
        3: . . 4  . 2   gap> BrauerTable( libtbl, 2 );
        4: . . . 4 .    fail
        5: . . 2 . .
                        gap> CharacterTable( "Symmetric", 4 );
o6 : BettiTally         CharacterTable( "Sym(4)" )
i7 : t1 = betti(t,Weights=>{1,1})
                        gap> ComputedBrauerTables( tbl );
        0 1  2  3 4     [ , BrauerTable( Sym( [ 1 .. 4 ] ), 2 ) ]
o7 = total: 1 4 13 14 4
        0: 1  .  .  . .
        1: . .  .  . .
        2: . .  .  . .
        3: . 2  .  . .
        4: . .  .  . .
        5: . 2  .  . .
        6: . .  1  . .
        7: . .  8  6 .
        8: . .  4  8 4

o7 : BettiTally
i8 : peek t1

o8 = BettiTally{(0, {0, 0}, 0) => 1 }
               (1, {2, 2}, 4) => 2
               (1, {3, 3}, 6) => 2
               (2, {3, 7}, 10) => 2
               (2, {4, 4}, 8) => 1
               (2, {4, 5}, 9) => 4
               (2, {5, 4}, 9) => 4
               (2, {7, 3}, 10) => 2
               (3, {4, 7}, 11) => 4
               (3, {5, 5}, 10) =>
               (3, {7, 4}, 11) => 4
               (4, {5, 7}, 12) => 2
               (4, {7,
```

```
ring r1 = 32003,(x,y,z),ds;
int a,b,c,t=11,5,3,0;
poly f = x^a+y^b+z^(3*c)+x^(c+2)*y^(c-1)+x^
          x^(c-2)*y^c*(y^2+t*x)^2;
option(noprot);
timer=1;
ring r2 = 32003,(x,y,z),dp;
poly f=imap(r1,f);
ideal j=jacob(f);
vdim(std(j));
==> 536
vdim(std(j+f));
==> 195
timer=0;  // reset timer
```

Computing maximum likelihood estimates for Gaussian graphical models with Macaulay2

Carlos Améndola, Luis David García Puente,
Roser Homs, Olga Kuznetsova and Harshit J. Motwani

Computing maximum likelihood estimates
for Gaussian graphical models with Macaulay2

CARLOS AMÉNDOLA, LUIS DAVID GARCÍA PUENTE,
ROSER HOMS, OLGA KUZNETSOVA AND HARSHIT J. MOTWANI

ABSTRACT: We introduce the package *GraphicalModelsMLE* for computing the maximum likelihood
estimates (MLEs) of a Gaussian graphical model in the computer algebra system *Macaulay*2. This
package allows the computation of MLEs for the class of loopless mixed graphs. Additional functionality
allows the user to explore the underlying algebraic structure of the model, such as its maximum likelihood
degree and the ideal of score equations.

1. INTRODUCTION. The purpose of the package *GraphicalModelsMLE* is to extend the functionality of
[Macaulay2] related to algebraic statistics, specifically allowing computations of maximum likelihood
estimates of Gaussian graphical models. While *GraphicalModels* is an existing package that already
provides useful information such as conditional independence ideals and vanishing ideals for such models,
the fundamental statistical inference task of computing maximum likelihood estimates is missing. This
package aims to fill this void and also extend the functionality of *GraphicalModels* to handle more
general types of graphs, in particular, *loopless mixed graphs* (LMG). This class of graphs was introduced
in [Sadeghi and Lauritzen 2014] in order to unify the Markov theory of several classical types of graphs
such as undirected graphs, directed acyclic graphs, summary graphs and ancestral graphs [Lauritzen
1996].

The algebraic framework of *Macaulay*2 permits us to use both commutative algebra and numerical
algebraic geometry to obtain a guaranteed global optimal solution by computing all critical points of
the log-likelihood function. This is different from the classical statistical approach of the *R* package
ggm [Marchetti 2006], and more in line with the recent numerical algebraic geometry approach from the
package *LinearCovarianceModels.jl* in *Julia* [Sturmfels et al. 2020]. The package *GraphicalModelsMLE*
is a complement to these two, handling some Gaussian graphical models not covered by them (*LinearCo-
varianceModels.jl* version 0.2 and *ggm* version 2.5). The capabilities of our package are limited by the
feasibility of the Gröbner basis computations involved.

Given a data sample of n independent and identically distributed random vectors $X^{(1)}, \ldots, X^{(n)}$
that follow an m-dimensional multivariate Gaussian distribution $\mathcal{N}(\mu, \Sigma)$, the *maximum likelihood*
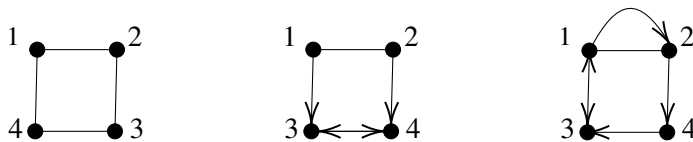
Figure 1. An undirected graph (left), a mixed graph with no directed cycles (centre), and a mixed graph with directed cycles (right).

estimate (MLE) for the covariance matrix Σ is the matrix that best explains the observed data, in the sense that it maximizes the likelihood function of the Gaussian model (see Section 3).

2. GRAPHICAL MODELS OF LOOPLESS MIXED GRAPHS. A *mixed graph* $G = (V, E)$ is a graph with undirected edges $i - j$, directed edges $i \to j$ and bidirected edges $i \leftrightarrow j$. A *directed cycle* is a cycle formed by directed edges after identifying the vertices that are connected by undirected or bidirected edges. A *loopless mixed graph* (LMG) is a mixed graph without loops or directed cycles. We allow double edges of the types directed-undirected and directed-bidirected. See Figure 1 (left and centre) for examples and Figure 1 (right) for a nonexample.

Following [Sullivant et al. 2010], we assume the nodes of G are partitioned as $V = U \cup W$, such that:

- If $i - j$ in G then $i, j \in U$.

- If $i \leftrightarrow j$ in G then $i, j \in W$.

- There is no directed edge $i \to j$ in G such that $i \in W$ and $j \in U$.

Our definition differs from the one in [Sadeghi and Lauritzen 2014] in that we do not allow multiple edges of the same type, which is due to the setup of the *Graphs* package. Also note that the partition of vertices excludes multiple edges of the undirected-bidirected type. In addition, we prohibit directed cycles, which ensures there is an ordering on the vertices such that all vertices in U come before vertices in W, and whenever $i \to j$ we have $i < j$. For simplicity of the algorithm, in our *Macaulay2* implementation we will require the graph to be provided with such an ordering of the vertices (see more details in the description of `partitionLMG` in Section 6).

A Gaussian graphical model imposes constraints on the covariance matrix of a Gaussian distribution. More precisely, a loopless mixed graph $G = (V, E)$ gives rise to the space of covariance matrices $\Sigma \in \mathbb{R}^{|V| \times |V|}$ of the form [Sullivant et al. 2010, Section 2.3]

$$\Sigma \quad = \quad (I - \Lambda)^{-T} \begin{bmatrix} K^{-1} & \mathbf{0} \\ \mathbf{0} & \Psi \end{bmatrix} (I - \Lambda)^{-1}, \tag{2.1}$$

where

(i) $\Lambda = [\lambda_{ij}] \in \mathbb{R}^{|V| \times |V|}$ is such that $\lambda_{ij} = 0$ whenever $i \to j \notin E$;

(ii) $K = [k_{ij}] \in \mathbb{R}^{|U| \times |U|}$ is symmetric positive definite such that $k_{ij} = 0$ whenever $i - j \notin E$;

(iii) $\Psi = [\psi_{ij}] \in \mathbb{R}^{|W| \times |W|}$ is symmetric positive definite such that $\psi_{ij} = 0$ whenever $i \leftrightarrow j \notin E$.
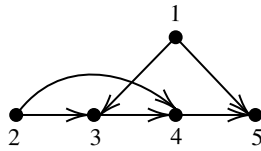
Figure 2. Verma graph.

3. MAXIMUM LIKELIHOOD ESTIMATES. Let the data sample consist of n independent identically distributed random vectors $X^{(1)}, \ldots, X^{(n)}$ sampled from an m-dimensional Gaussian distribution $\mathcal{N}_m(\mu, \Sigma)$. The parameter space of the corresponding statistical model is $\Theta = \mathbb{R}^m \times \Theta_2 \subseteq \mathbb{R}^m \times PD_m$, where Θ_2 is the space of covariance matrices Σ and PD_m is the cone of $m \times m$ positive-definite matrices. The maximum likelihood estimates for the covariance matrix are determined by maximizing the log-likelihood function

$$\ell(\Sigma) = -\frac{n}{2} \log \det \Sigma - \frac{n}{2} \operatorname{tr}(S\Sigma^{-1}) \tag{3.1}$$

over $\Sigma \in \Theta_2$ [Sullivant 2018, Proposition 7.1.9], where S is the sample covariance matrix. The function `solverMLE` allows us to compute this optimum when Θ_2 is induced by (2.1). It does so by calculating the critical points of the log-likelihood function and selecting the points corresponding to the maximum value in the cone of positive definite matrices. The default output is the maximum value of $\ell(\Sigma)$, the list of maximum likelihood estimates for the covariance matrix and the maximum likelihood degree of the model.

For undirected graphs, the MLE for the covariance matrix is known to be the unique positive definite critical point of the likelihood function. In particular, it is a positive definite matrix completion to the partial sample covariance matrix. See [Uhler 2012, Theorem 2.1] or [Drton et al. 2009, Theorem 2.1.14] for more details.

Example 3.2. We consider the directed acyclic graph G known as the Verma graph; see Figure 2 and [Drton et al. 2009, Example 3.3.14]. We take as sample data the columns of a real matrix U generated with the command `random` in *Macaulay*2 and compute the MLE for the covariance matrix that best explains the data within the graphical model given by G.

```
i1: loadPackage "GraphicalModelsMLE";

i2: G=digraph{{1,3},{1,5},{2,3},{2,4},{3,4},{4,5}};

i3: U=matrix {{.0137595, .983763, .963969, .152094, .0453326},
              {.527344, .597575, .777622, .97937, .112339},
              {.097922, .300712, .333058, .824002, .420228},
              {.849322, .594136, .114729,.69734, .98773},
              {.764547, .42209, .480193, .246573, .846734}};

i4: solverMLE(G,U)

o4 = (8.77485, | .115729     -6.32685e-18 -.0387187 .00115181 .102733   |, 1)
               | 6.10884e-18 .053294      .0392544  -.0356783 .00701454 |
               | -.0387187   .0392544     .0807822  -.0278223 -.0289767 |
               | .00115181   -.0356783    -.0278223 .105095   -.0196375 |
               | .102733     .00701454    -.0289767 -.0196375 .148723   |
```

Example 3.3. We compute the MLE for the covariance matrix of the graphical model associated to the undirected 4-cycle; see Figure 1 (left). We encode the sample data by a matrix U generated as in Example 3.2 and compute the sample covariance matrix $S = \frac{1}{n}UU^T$.

```
i1 : loadPackage "GraphicalModelsMLE";

i2 : G=graph{{1,2},{2,3},{3,4},{4,1}};

i3 : S=matrix {{.105409, -.0745495, -.0186132, .0621907},
               {-.0745495, .0783734,-.00844503,-.0872842},
               {-.0186132, -.00844503, .128307, .0230245},
               {.0621907, -.0872842, .0230245,.109849}};

i4 : solverMLE(G,S,SampleData=>false)

o4 = (6.62005, | .105409   -.0745495  .0124099    .0621907  |, 5)
               | -.0745495 .0783734   -.00844503 -.0439427 |
               | .0124099  -.00844503 .128307     .0230245  |
               | .0621907  -.0439427  .0230245    .109849   |
```

Note that all entries in the MLE for the covariance matrix coincide with the entries in the sample covariance matrix except for those corresponding to nonedges of the graph. See [Michałek and Sturmfels 2021, Example 12.16] for more on a positive definite matrix completion problem associated to the 4-cycle.

For more general types of graphs, uniqueness of the positive definite critical points is no longer guaranteed. In the mixed graph in Example 3.4, the optimization problem has a global maximum, but there are also local maxima; see Example 4.3.

Example 3.4. We compute the MLE for the covariance matrix of the graphical model associated to the loopless mixed graph with undirected edge $1 - 2$, directed edges $1 \to 3$, $2 \to 4$ and bidirected edge $3 \leftrightarrow 4$; see Figure 1 (centre). S is a sample covariance matrix computed from sample data encoded in a rational matrix obtained again with the command `random`.

```
i2 : G = mixedGraph(graph{{1,2}},digraph{{1,3},{2,4}},bigraph{{3,4}});

i3 : S=matrix {{34183/50000, 716539/10000000, 204869/250000, 12213/25000},
               {716539/10000000, 112191/500000, 309413/1000000, 1803/4000},
               {204869/250000, 309413/1000000, 3849/3125, 15172/15625},
               {12213/25000, 1803/4000, 15172/15625, 4487/4000}};

i4 : solverMLE(G,S,SampleData=>false)

o4 = (9.36624, {| .68366    .0716539 1.00282    .234375   |}, 5)
               | .0716539 .224382  .105105    .733937   |
               | 1.00282  .105105  1.76955    -.0700599 |
               | .234375  .733937  -.0700599 2.97432    |
```

4. IDEAL OF SCORE EQUATIONS. The critical points of the log-likelihood function $\ell(\Sigma)$ are the solutions to the system of equations obtained by taking partial derivatives of ℓ with respect to all variables in the entries of Σ from our construction in (2.1) and setting them to zero:

$$-\frac{\partial}{\partial(\cdot)}\det \Sigma - \det \Sigma \frac{\partial}{\partial(\cdot)}\text{tr}(S\Sigma^{-1}) = 0, \tag{4.1}$$

where $\frac{\partial}{\partial(\cdot)}$ stands for the partial derivatives with respect to the variables λ_{ij}, k_{ij}, ψ_{ij} in the entries of the covariance matrix Σ from (2.1). These polynomial equations are called *score equations*. The command `scoreEquations` returns the ideal generated by such polynomials, which lives in the polynomial

ring $\mathbb{Q}[\lambda_{ij}, k_{ij}, \psi_{ij}]$. From an algebraic perspective, this ideal is already of interest on its own; see [Sullivant 2018, Chapter 7].

Note that the log-likelihood function depends both on the sample covariance matrix and the graphical model. Therefore our implementation of `scoreEquations` requires as input the sample data along with information about the model. The latter is obtained via the command `gaussianRing` in the package *GraphicalModels* (see [García-Puente et al. 2013] and examples in Section 6), which produces a ring associated to the graph G that stores all relevant features of the graphical model.

Example 4.2. We compute the ideal of score equations associated to the 4-cycle after creating the graph G as in Example 3.3. We now consider as input data the sample data encoded in the columns of the integer matrix U below, obtained via the command `random`.

```
i5 : U=matrix{{3,5,9,5},{1,6,1,5},{2,9,6,6},{2,5,0,4}};

i6 : J=scoreEquations(gaussianRing G,U);

o6 : Ideal of QQ[k    , k    , k    , k    , k    , k    , k    , k    ]
                  1,1   2,2   3,3   4,4   1,2   1,4   2,3   3,4

i7 : dim J

o7 = 0
```

The ideal of the score equations J is generated by fourteen nonhomogeneous polynomials within $\mathbb{Q}[k_{1,1}, k_{1,2}, k_{1,4}, k_{2,2}, k_{2,3}, k_{3,3}, k_{3,4}, k_{4,4}]$: 4 linear polynomials and 10 quadratic polynomials such as $1312002k_{3,4}^2 - 387081k_{1,2} + 109860k_{1,4} + 1972025k_{2,3} - 898518k_{3,4} - 291556$. Since this ideal is zero-dimensional, the log-likelihood function $\ell(\Sigma)$ defined in (3.1) has finitely many complex critical points, as will be discussed in Section 5.

Example 4.3. We want to obtain all local maxima of the log-likelihood function associated to the graphical model in Example 3.4. We write λ as l and ψ as p in the code. The score equations generate an ideal in $\mathbb{Q}[k_{1,1}, k_{2,2}, k_{1,2}, l_{1,3}, l_{2,4}, p_{3,3}, p_{4,4}, p_{3,4}]$ and we display their solutions in the *Macaulay2* session below. We retrieve the covariance matrix Σ with rational entries in variables $k_{1,1}, k_{2,2}, k_{1,2}, l_{1,3}, l_{2,4}, p_{3,3}, p_{4,4}$, and $p_{3,4}$ by requiring the optional output `CovarianceMatrix` in `scoreEquations`.

```
i5 : R = gaussianRing G;

i6 : (J,Sigma)=scoreEquations(R,S,SampleData=>false,CovarianceMatrix=>true);

i7 : dim J, degree J

o7 = (0, 5)

i8 : sols=zeroDimSolve(J);netList sols

     +-----------------------------------------------------------------------+
o8 = |{1.51337, 4.61101, -.483277, 1.46684, 3.27093, .298576, .573665, -.41385} |
     +-----------------------------------------------------------------------+
     |{1.51337, 4.61101, -.483277, 1.39884+.440525*ii, 2.45466-.923165*ii,   |
     |.144129+.120574*ii,.0696297-.184692*ii,-.19668+.0553853*ii}            |
     +-----------------------------------------------------------------------+
     |{1.51337, 4.61101, -.483277, 1.39884-.440525*ii, 2.45466+.923165*ii,   |
     |.144129-.120574*ii, .0696297+.184692*ii, -.19668-.0553853*ii}          |
     +-----------------------------------------------------------------------+
     |{1.51337, 4.61101, -.483277, .684147, .979681, .430388, .453924, .381688} |
     +-----------------------------------------------------------------------+
     |{1.51337, 4.61101, -.483277, .988484, 1.64649, .279607, .245722, .0952865}|
     +-----------------------------------------------------------------------+
```
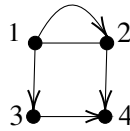
Figure 3. LMG with two types of edges between 1 and 2.

How many of the 3 real critical points correspond to positive definite matrices that are local maxima of the log-likelihood function? We first check that they correspond to positive definite matrices by substituting the three real solutions in the covariance matrix Σ.

```
i9 : checkPD(apply(sols,i->sub(Sigma,matrix{coordinates(i)})))

o9 = |.68366    .0716539 1.00282   .234375   |,| .68366    .0716539 .467724  .070198  |,
     |.0716539 .224382  .105105   .733937   | | .0716539 .224382  .0490218 .219823  |
     |1.00282  .105105  1.76955   -.0700599 | | .467724  .0490218 .75038   .429714  |
     |.234375  .733937  -.0700599 2.97432   | | .070198  .219823  .429714  .66928   |
                                  | .68366    .0716539 .675787 .117978  |
                                  | .0716539 .224382  .0708287 .369443  |
                                  | .675787  .0708287 .947611  .211905  |
                                  | .117978  .369443  .211905  .854009  |
```

The MLE for the covariance matrix obtained in Example 3.4 corresponds to the first positive definite matrix in the list above. The eigenvalues of the Hessian matrix computed below tell us which kind of critical point we have for each of the 3 real solutions — for a discussion about the properties of positive-semidefinite matrices; see [Parrilo 2013, Appendix A].

```
-- compute Jacobian matrix (i.e. score equations)
i10 : scoreEq=-1/det Sigma*jacobianMatrixOfRationalFunction(det Sigma)-
jacobianMatrixOfRationalFunction(trace(S*(inverse Sigma)));

-- compute Hessian matrix
i11 : Hessian=matrix for f in flatten entries scoreEq list
flatten entries jacobianMatrixOfRationalFunction(f);

-- compute eigenvalues of the Hessian matrix evaluated at real points in sols
i12 : apply({sols_0,sols_3,sols_4},i->eigenvalues sub(Hessian,matrix{coordinates(i)}))

o12 = {{-.516478   }, {-.516478 }, {-.516478 }}
       {-.271913   }  {-.271913 }  {-.271913 }
       {-.0464172  }  {-.0464172}  {-.0464172}
       {-9869730000}  {-414.15  }  {-59.7135 }
       {-128887     }  {-28.6352 }  {1.52598  }
       {-58261.1    }  {-6.1936  }  {-2.80504 }
       {-773.513    }  {-1.64689 }  {-11.5533 }
```

The first two points are local maxima and the last point is a saddle point. This shows that the log-likelihood function of this model is not a concave function, see [Drton 2006].

Example 4.4. Next we compute the ideal of score equations associated to a mixed graph that has two different types of edges connecting the same two vertices: directed edges $1 \to 3$, $1 \to 2$, $2 \to 4$, $3 \to 4$ and the undirected edge $1 - 2$; see Figure 3.

```
i2 : G = mixedGraph(digraph{{1,3},{1,2},{2,4},{3,4}},graph{{1,2}});

i3 : R = gaussianRing G;

i4 : U = random(RR^4,RR^4);

i5 : J=scoreEquations(R,U);

i6 : dim J

o6 = 1
```

Note that in this case, as opposed to Example 4.2, the log-likelihood function (3.1) has infinitely many complex critical points. Since our package evaluates the objective function in all critical points, in such a scenario the MLE cannot be computed.

5. MAXIMUM LIKELIHOOD DEGREE.   The *maximum likelihood degree* (ML degree) of a model is defined as the number of complex critical points of the log-likelihood function $\ell(\Sigma)$ from (3.1) for generic sample data; see [Sullivant 2018, Definition 7.1.4]. For a more algebraic flavour of the notion of ML degree, see [Michałek et al. 2016, Definition 5.4].

Note that the ML degree is only well defined when the ideal of score equations is zero-dimensional. A typical way where this fails is where the model becomes nonidentifiable. See, for example, [Améndola et al. 2020], for some sufficient conditions to avoid nonidentifiability and preservation of dimension of the model in terms of the number of parameters.

It is important to observe that for generic data the solutions to score equations are all distinct; see [Améndola et al. 2021, Remark 2.1, Lemma 2.2]. Computing the algebraic degree of the zero-dimensional score equations ideal via the `degree` function in *Macaulay*2 is equivalent to computing the number of complex solutions - without multiplicity - to the score equations (4.1).

In our implementation of the `MLdegree` function in *Macaulay*2, a random sample matrix is used as sample data. Therefore, the ML degree of the graphical model we provide is correct with probability 1.

Example 5.1. The ML degree of the 4-cycle can be directly computed as follows:

```
i2 : G=graph{{1,2},{2,3},{3,4},{4,1}};
i3 : MLdegree(gaussianRing G)
o3 = 5
```

In the case of ideals of score equations with positive dimension, `MLdegree` will still compute the degree of the ideal but this no longer matches the number of solutions to the score equations.

Example 5.2. Continuing with Example 4.4, where the ideal of score equations is 1-dimensional, `MLdegree` does not provide a meaningful answer.

```
i2: G=mixedGraph(digraph{{1,3},{1,2},{2,4},{3,4}},graph{{1,2}});
i3: MLdegree(gaussianRing G)
error: the ideal of score equations has dimension 1 > 0,
so ML degree is not well defined. The degree of this ideal is 2.
```

6. UPDATES IN RELATED PACKAGES.   *GraphicalModelsMLE* relies on the new package *StatGraphs* 0.1 and the updated packages *Graphs* 0.3.3 and *GraphicalModels* 2.0 (see [García-Puente et al. 2013] for version 1.0).

We created a dedicated package *StatGraphs* for graph theoretic functions relevant to algebraic statistics. It contains the functions `isCyclic`, `isSimple`, `isLoopless` and `partitionLMG` to deal with loopless mixed graphs.

The function `partitionLMG` computes the partition $V = U \cup W$ of vertices of a loopless mixed graph described in Section 2. Vertices in the input graph need to be ordered such that all vertices in U come before vertices in W, and if there is a directed edge $i \to j$, then $i < j$.

Example 6.1. The vertices of the loopless mixed graph in Example 3.4 are partitioned into $U = \{1, 2\}$ and $W = \{3, 4\}$.

```
i1 : loadPackage "StatGraphs";

i2 : G = mixedGraph(digraph {{1,3},{2,4}},bigraph{{3,4}},graph{{1,2}});

i3 : partitionLMG G

o3 = ({1, 2}, {3, 4})

o3 : Sequence
```

The central object in the implementation of our MLE algorithm is `gaussianRing` from the package *GraphicalModels*.

Example 6.2. We compute the ring associated to the graph in Example 6.1 and display the variables of the ring as entries of matrices. We write λ as l and ψ as p in the code.

```
i4 : loadPackage "GraphicalModels";

i5 : R=gaussianRing G;

i6 : undirectedEdgesMatrix R

o6 = | k_(1,1) k_(1,2) |
     | k_(1,2) k_(2,2) |

i7 : directedEdgesMatrix R

o7 = | 0 0 l_(1,3) 0       |
     | 0 0 0       l_(2,4) |
     | 0 0 0       0       |
     | 0 0 0       0       |

i8 : bidirectedEdgesMatrix R

o8 = | p_(3,3) p_(3,4) |
     | p_(3,4) p_(4,4) |

i9 : covarianceMatrix R

o9 = | s_(1,1) s_(1,2) s_(1,3) s_(1,4) |
     | s_(1,2) s_(2,2) s_(2,3) s_(2,4) |
     | s_(1,3) s_(2,3) s_(3,3) s_(3,4) |
     | s_(1,4) s_(2,4) s_(3,4) s_(4,4) |
```

In version 2.0 of *GraphicalModels*, we updated the functionality of the method `gaussianRing` and its related methods in order to accept loopless mixed graphs with undirected, directed and bidirected edges.

Note that mixed graphs that include undirected edges are required to have an ordering compatible with `partitionLMG`. For mixed graphs with only directed and bidirected edges this is no longer necessary, as in version 1.0 of *GraphicalModels*.

SUPPLEMENT. The online supplement contains version 1.0 of *GraphicalModelsMLE*.

REFERENCES.

[Améndola et al. 2020] C. Améndola, P. Dettling, M. Drton, F. Onori, and J. Wu, "Structure learning for cyclic linear causal models", pp. 999–1008 in *Proceedings of the 36th Conference on Uncertainty in Artificial Intelligence*, edited by J. Peters and D. Sontag, Proceedings of Machine Learning Research **124**, PMLR, 2020.

[Améndola et al. 2021] C. Améndola, L. Gustafsson, K. Kohn, O. Marigliano, and A. Seigal, "The maximum likelihood degree of linear spaces of symmetric matrices", *Matematiche* (*Catania*) **76**:2 (2021), 535–557. MR

[Drton 2006] M. Drton, "Computing all roots of the likelihood equations of seemingly unrelated regressions", *J. Symbolic Comput.* **41**:2 (2006), 245–254. MR Zbl

[Drton et al. 2009] M. Drton, B. Sturmfels, and S. Sullivant, *Lectures on algebraic statistics*, Oberwolfach Seminars **39**, Birkhäuser Verlag, Basel, 2009. MR Zbl

[García-Puente et al. 2013] L. D. García-Puente, S. Petrović, and S. Sullivant, "Graphical models", *J. Softw. Algebra Geom.* **5** (2013), 1–7. MR Zbl

[Lauritzen 1996] S. L. Lauritzen, *Graphical models*, Oxford Statistical Science Series **17**, Oxford University Press, 1996. MR Zbl

[Macaulay2] D. R. Grayson and M. E. Stillman, "Macaulay2: a software system for research in algebraic geometry", available at http://www.math.uiuc.edu/Macaulay2.

[Marchetti 2006] G. M. Marchetti, "Independencies Induced from a Graphical Markov Model After Marginalization and Conditioning: The R Package ggm", *Journal of Statistical Software* **15**:6 (2006), 1–15.

[Michałek and Sturmfels 2021] M. Michałek and B. Sturmfels, *Invitation to nonlinear algebra*, Graduate Studies in Mathematics **211**, American Mathematical Society, Providence, RI, 2021. MR Zbl

[Michałek et al. 2016] M. Michałek, B. Sturmfels, C. Uhler, and P. Zwiernik, "Exponential varieties", *Proc. Lond. Math. Soc.* (3) **112**:1 (2016), 27–56. MR Zbl

[Parrilo 2013] P. A. Parrilo, "Semidefinite optimization", pp. 3–46 in *Semidefinite optimization and convex algebraic geometry*, MOS-SIAM Ser. Optim. **13**, SIAM, Philadelphia, PA, 2013. MR Zbl

[Sadeghi and Lauritzen 2014] K. Sadeghi and S. Lauritzen, "Markov properties for mixed graphs", *Bernoulli* **20**:2 (2014), 676–696. MR Zbl

[Sturmfels et al. 2020] B. Sturmfels, S. Timme, and P. Zwiernik, "Estimating linear covariance models with numerical nonlinear algebra", *Algebr. Stat.* **11**:1 (2020), 31–52. MR Zbl

[Sullivant 2018] S. Sullivant, *Algebraic statistics*, Graduate Studies in Mathematics **194**, American Mathematical Society, Providence, RI, 2018. MR Zbl

[Sullivant et al. 2010] S. Sullivant, K. Talaska, and J. Draisma, "Trek separation for Gaussian graphical models", *Ann. Statist.* **38**:3 (2010), 1665–1685. MR Zbl

[Uhler 2012] C. Uhler, "Geometry of maximum likelihood estimation in Gaussian graphical models", *The Annals of Statistics* **40**:1 (2012), 238 – 261. Zbl

CARLOS AMÉNDOLA:

carlos.amendola@mis.mpg.de
Max Planck Institute for Mathematics in the Sciences, Leipzig, Germany

LUIS DAVID GARCÍA PUENTE:

lgarciapuente@coloradocollege.edu
Department of Mathematics and Computer Science, Colorado College, Colorado Springs, CO, United States

Roser Homs:

roser.homs@tum.de
Department of Mathematics, Technical University of Munich, Munich, Germany

Olga Kuznetsova:

kuznetsova.olga@gmail.com
Department of Mathematics and Systems Analysis, Aalto University, Aalto, Finland

Harshit J. Motwani:

harshitjitendra.motwani@ugent.be
Department of Mathematics: Algebra and Geometry, Ghent University, Ghent, Belgium