arXiv:2601.15807v1 [stat.CO] 22 Jan 2026

OSCAR中的代数统计

´TOBIAS BOEGE、ANTONY DELLAVECCHIA、MARINAGARROTE-LOPEZ 和 BENJAMIN HOLLERING

Abstract. 我们介绍 OSCAR 计算机代数系统中的代数统计部分。我们概述其可扩展的设计，并重点介绍其功能，包括用于共享结果和创建数据库的数据类型序列化，以及先进的隐式化算法。

1. Introduction

代数统计运用代数几何、交换代数和组合学中的工具来解决统计学问题。这种方法基于这样一种认识：在许多合理且实用的情形下，参数统计模型 $\mathcal{M}$ 是某个 $\varphi(\Theta)$“简单”半代数集在多项式（或有理）映射 Θ 下的像 φ。许多有趣的统计性质 $\mathcal{M}$ 都由其扎里斯基闭包$V = \mathcal{M} = \varphi(\Theta)$所刻画，可以利用格罗布纳基等计算机代数方法进行研究。自 Diaconis and Sturmfels 的开创性论文¨建立了费舍尔精确检验与环面理想的格罗布纳基之间出人意料的联系以来 [22]„ 代数方法在统计学中发现了许多新的应用，包括条件独立蕴含¨结构和统计模型的参数可识别性[3, 13, 14, 24, 33, 38], 、因果发现 [5, 6, 15],以及最大似然估计 [23, 41], [17, 18, 30]。如需深入了解代数统计，以及代数工具解决统计问题的各种方式，请参阅 [39]。

本文介绍了计算机代数系统的 `AlgebraicStatistics` 部分OSCAR [20, 34] ，该部分首次出现在 1.2.0 版本中。它为一系列统计任务提供代数解法。目前的功能主要围绕图模型和系统发育学展开。我们利用 `Julia` 编程语言的类型系统创建了一个框架，让用户或未来的开发者能够轻松地在现有功能的基础上构建新的模型族。

`Julia` 生态系统正稳步发展，并将 OSCAR与功能强大的可视化、网络、数据库和数值代数几何工具包等连接起来。希望运行本文所讨论示例的读者，可通过 https://github.com/dmg-lab/OscarAlgebraicStatistics找到一个 Jupyter 笔记本。

2. The 图模型

图模型是从图中导出的统计模型。图的顶点对应随机变量，而边通过在参数化中规定稀疏模式，限制某些随机变量的影响

日期：2026年1月23日。2020 数学主题分类。62R01、62-04、13P25（主要），62H22、92-08、14-04（次要）。关键词和短语。代数统计、代数系统发育学、图模型、计算机代数。

其他随机变量的方式是规定参数化中的稀疏模式。通过这种方式，图会影响模型中分布的定性特征，通常是施加条件独立约束。由于具有良好的可解释性，图模型已广泛应用于各个科学领域：生物学家试图将物种谱系组织成系统发育网络 [7, 37]，计量经济学家 [42] 和社会科学家 [12] 进行因果建模，工程师将其用于计算机视觉 [28]，，环境科学家则构建了加利福尼亚州水库的具体图模型 [40]。这引发了人们对其理论基础的广泛兴趣；参见 [31, 32, 35]。

一般而言，图模型是一种参数统计模型，由有理映射 $\varphi_{\mathcal{G}}\colon N \to M$ 的像给出，其具体形式取决于图 $\mathcal{G}$ 以及分布类型的选择：

- 对于高斯随机变量，$\Sigma = \varphi_{\mathcal{G}}(\theta) \in \mathrm{PD}_n$ 是协方差矩阵。
- 对于具有 d_1、……、$d_n \geq 1$ 个状态的 n 离散随机变量，$p = \varphi_{\mathcal{G}}(\theta) \in \Delta(d_1$、……、$d_n)$ 是由原子概率组成的非负张量。

图 $\mathcal{G}$ 可以有许多不同类型，例如无向图或有向图，每种类型都会产生不同的参数化 $\varphi_{\mathcal{G}}$ 和不同的模型。

我们将若干常见的图模型族实现为抽象参数类型 `GraphicalModel{T, L}` 的实例。该框架的主要思路是让用户能够实现自己的图模型类型，同时尽可能复用现有代码。用户可以通过重载相应的属性函数，创建自己的图模型 $\mathcal{M} = \varphi_{\mathcal{G}}(N) \subseteq M$，该模型由有理映射 $\varphi_{\mathcal{G}}\colon N \to M$ 的像给出

- `parameter_ring` 返回参数空间 N 的坐标环，
- `model_ring` 是环境空间 M 的坐标环，
- `parametrization` 是从 `model ring` 到 `parameter ring`的环同态，它是参数化 $\varphi_{\mathcal{G}}$ 的拉回。

我们将在第 2.1.1 节中展示一个例子。

Julia的类型系统支持复杂的方法分派功能；感兴趣的读者可参阅 [11, §3、§4] 。这里补充说明一下相关术语：函数在 Julia中是一个名称，指向一组特定于类型的方法。调用 `parametrization(M)`会调用函数 `parametrization`中最适合处理参数 `M`类型的方法。因此，我们可以提供一个统一的接口，以涵盖整个“图模型谱系”。在第 3 节中，我们将展示 Julia的多重分派理念如何大显身手：对于某些类型的图，如果有定理支持计算捷径，我们就能透明地提供更快的算法。

遵循高斯型与离散型的二分法，我们提供以下具体类型

- `GaussianGraphicalModel{T, L}`和 `DiscreteGraphicalModel{T, L}`；
- `PhylogeneticModel{T, L}`和 `GroupBasedPhylogeneticModel{T, L}`。

例如，可以像这样创建一个高斯图模型：`julia>G = 图_来自_边（无向，`

```
julia>M = 高斯_图_模型(G)
具有 4 个节点和 4 条边的无向图上的高斯图模型

julia> typeof(M)
高斯图模型{图{无向}, Nothing}
```

正在从无向图 M 创建 G 可推断类型参数 T = 图{无向}。 T 的可能取

- Graph{Directed}、 Graph{Undirected} 或 MixedGraph;
- PhylogeneticTree 或 PhylogeneticNetwork。

标签类型 L 是来自 NamedTuple 的类型，属于 Julia的标准库。它可通过一个 GraphMap ，从 polymake为图附加额外数据。第 2.1.1 节展示了如何利用它实现带颜色的高斯图模型，相关介绍见 [27]。

2.1. 无向图上的高斯图模型。上述会话中的对象 M 表示一个定义在四个顶点上的无向高斯图模型。该模型是一个半代数集，构造如下。设 $\mathcal{G} = ([n], E)$ 是一个无向图，并令 $\mathcal{L}_G = \{K \in \mathbb{C}^{n\times n}_{\mathrm{sym}} : \ k_{ij} = 0$ ，对于 $(i, j), \ \in E(G)\}$ } 表示由 $\mathcal{G}$ 的边所支撑的对称矩阵构成的线性空间。那么与 $\mathcal{G}$ 对应的高斯图模型为 $\mathcal{M}_\mathcal{G} = \{\Sigma \in \mathbb{R}^{n\times n} : \ \Sigma^{-1} \in \mathcal{L}_G\} \cap \mathbf{PD}_n$ ，它是逆线性空间与 PD_n 的交集。模型 $\mathcal{M}_\mathcal{G}$ 也可以看作有理映射的像

$$\begin{aligned} \varphi_\mathcal{G} : \mathcal{L}_G \cap \mathrm{PD}_n &\to \mathrm{PD}_n \\ K &\mapsto K^{-1}. \end{aligned} \tag{1}$$

在代数统计中，人们通常忽略 $\Sigma \in \mathcal{M}_\mathcal{G}$ 正定这一约束，转而研究模型的代数闭包 $V_\mathcal{G} = \mathcal{M}_\mathcal{G}$。该闭包通过将有理映射 $\varphi_\mathcal{G}$ 扩展到整个 $\mathcal{L}_G$ 来参数化。逐个案例观察可知，我们感兴趣的统计性质只取决于 $V_\mathcal{G}$，因此这种做法是合理的。从算法角度来看，处理簇 $V_\mathcal{G}$ 比处理半代数集 $\mathcal{M}_\mathcal{G}$容易得多。

这个图模型也可以定义为满足一组条件独立（CI）语句的正定矩阵集合，这组语句称为 $\mathcal{G}$的全局马尔可夫性质。由于高斯随机向量满足条件独立$A \perp\!\!\!\perp B \mid C$ 当且仅当 rank($\Sigma_{A\cup C, B\cup C}$) $< \#C + 1$，满足一组 CI 语句 $\mathcal{C}$ 的多元高斯分布集合也可以看作簇 $V_\mathcal{C} = V(I_\mathcal{C})$ $= \{\Sigma \in \mathbb{C}^{n\times n}_{\mathrm{sym}}$ 的交集，其中：对所有 $A \perp\!\!\!\perp B \mid C \in \mathcal{C}\}$，在 PD_n 条件下，rank($\Sigma_{A\cup C, B\cup C}$) $< \#C + 1$ 。用户可以计算条件独立理想 $I_\mathcal{C}$ ，以及模型的消失理想 $\mathcal{I}(V_\mathcal{G})$。例如，在我们贯穿全文的例子中：

```
julia>C = 全局_markov(graph(M))2 个元



Julia>J = ci_ideal(gaussian_ring(M), C)
由.....生成的理想
  -s[1, 2]*s[2, 3]*s[4, 4] + s[1, 2]*s[2, 4]*s[3, 4] + … -s[1, 1]*s
```

```
Julia>I = 消失_理想(M)
由.....生成的理想
  -s[1, 2]*s[2, 3]*s[4, 4] + s[1, 2]*s[2, 4]*s[3, 4] + ... -s[1, 1]*s[2, 3]*s[3, 4] + s[1, 1]
```

这两个理想的生成方式不同，通常并不相同；但对于图 G，它们恰好相同，运行I == J即可验证。

2.1.1. 引入颜色。本部分通过示例展示我们的框架如何支持新类型图模型的实现。根据 [27]，彩色（无向高斯）图模型可通过进一步限制线性空间 $\mathcal{L}_{\mathcal{G}}$ 得到。图 $\mathcal{G}$ 扩展了一个映射 c，该映射为每个顶点和边分配一种颜色。我们通过以下方式定义对称矩阵 K 的线性空间 $\mathcal{L}_{\mathcal{G},\ c}$：

- $k_{ij} = 0$ 若 $ij \notin E$,
- $k_{ij} = k_{st}$ 若 $ij,\ st \in E$ 且 $c(ij) = c(st)$，并且
- $k_{ii} = k_{ss}$ 若 $i,\ s \in V$ 且 $c(i) = c(s)$。

来自 (1) 的参数化映射被限制在 $\mathcal{L}_{\mathcal{G},\ c}$ 但其他方面保持不变。在 Julia中，我们将边和顶点的颜色创建为哈希映射（ Julia中的 Dict），并将它们连同图类型和用于标记的名称一起传递给graph from labeled edges 函数。

```
朱莉娅>边_标签 = 字典((1,      4)  =>"绿色",   (2, 3)  => "绿色",
                          (3, 4)  =>"蓝色" , (1, 2)  => "蓝色");

朱莉娅>顶点_标签 = 字典(1 => "红色",  2 => "红色",  3 => "黄色",  4 => "黄色");

julia>G = 图_由_标记的_边(无向, 边_标签, 顶点_标签; 名称=:颜色);

julia> Mc  =高斯_图_模型(G);

julia> Mc 的类型
高斯图              Model{Graph{无向}, @命名元组{颜色::Oscar.GraphMap{无向}}}

julia> Ic  =消失_理想(Mc)
由 -s[1, 2]*s[2, 3]*s[4, 4] + s[1, 2]*s[2, 4]*s[3, 4] + ... -s[1, 1]*s[2, 3]*s[3, 4] + s
```

上面的代码片段表明，将标记为 color 的无向图传递给gaussian graphical model 函数会生成一个 GaussianGraphicalModel{T, L} 对象，其中包含L = @NamedTuple{颜色::Oscar.图映射{无向}}。不过，vanishing ideal有点奇怪：它与未着色模型的理想相同！

这是因为 OSCAR 1.7.0 没有内置可用于处理 color 标记的方法。因此，将调用 vanishing ideal 的通用方法，而该方法不会考虑 color。要让 OSCAR 了解着色图模型，我们只需重载 parameter ring：

```
Colo
@Na
多元
para
[[G.
in 顶
15 颜
in er
式环
Dict
G.co
```

我们逐行解析这段代码。第 1 行从 OSCAR 导入函数 parameter ring 和一些内部类型。显式导入该函数是为了能够通过定义新方法来扩展它（重载），而导入这些类型则能让代码更易读。第 3–5 行为我们的彩色高斯模型类型（ColoredGGM{Undirected}）定义了一个别名，以提高可读性。核心方法的定义从第 7 行开始。这里使用 @attr 宏，使参数环对每个模型只计算一次，然后缓存起来。第 7–10 行定义的函数签名如下：给定一个定义在无向图上的彩色高斯模型， parameter ring 返回一个元组，其中第一项是一个以 $\mathbb{Q}$ 为基环的多元多项式环，第二项是一个从边和顶点映射到该环生成元的映射（哈希映射）。第 11–14 行创建多项式环，以便将图中的每种不同颜色表示为一个变量。请注意，如果 color 是图 G 上的标记，就可以将其作为属性访问，也就是说， G.color[e] 会返回边 e 的颜色。这段代码的其余部分创建了一个从每种颜色到多项式环中相应变量的映射（第 15–16 行），然后根据颜色，利用该映射创建一个从图的边和顶点到多项式环中相应变量的映射（第 18–23 行）。

只要图模型现在涉及一个名称为 color 的标记， Julia 的多重分派就会优先选择上面专用化的 parameter ring。在着色模型中， parametrization函数 $K \mapsto K^{-1}$ 和 model ring 保持不变，因此我们不必重载它们;¯ 正如数学定义中那样，只需通过 parameter ring 限制参数化映射的定义域即可。默认的 vanishing ideal 方法只需计算 parametrization 的核，因此也会自动变得正确： julia> include("/path/to/overloaded/function/file.jl");

```
julia> Mc  =高斯_图形_模型(G);
```

```
julia> Ic  =消失_理想(Mc)
由 s[3, 3] -s[4, 4]s[1, 4] -s[2, 3]s[1, 3] -s[2, 4]
```

将此示例的输出与 [26,示例 2.2 和 2.3]进行比较。

2.2. **系统发育学**。 系统发育学提供了代数统计模型的另一个典型例子。其目标是根据观测到的遗传数据（例如 DNA序列）重建生物实体的进化史。这段历史由一棵系统发育树 $\mathcal{T} = (V,\ E)$（一个有向无环图），其根位于某个内部节点 r（或者，更一般地说，是一个系统发育网络，我们将在部分 2.2.2中讨论）。我们将遗传状态的演化建模为 $\mathcal{T}$ 上的马尔可夫过程。对于每个节点 $v \in V$，我们关联一个随机变量 X_v，其取值属于核苷酸状态空间 $\{\mathtt{A}, \mathtt{C}, \mathtt{G}, \mathtt{T}\}$。

在此框架下，该模型的参数化由马尔可夫过程的随机参数定义。参数空间 Θ 由根节点处的概率分布 π 以及一组 4×4 转移矩阵 $\mathcal{M} = \{M_e\}_{e \in E}$构成。对于每条有向边 $e = u \to v$，矩阵项 $M_e(x,\ y) = P(X_v = y \mid X_u = x)$ 表示父节点 X_u 处的状态 x 替换为子节点 X_v处状态 y 的概率。当不对 Θ施加约束时，这称为一般马尔可夫模型。经典生物学模型（例如*Jukes–Cantor*模型、*Kimura*模型、*GTR*模型）对应于子模型，其中 Θ 受到反映生化性质的线性约束限制。

该系统发育模型 $\mathcal{M}_\mathcal{T}$ 对应于在 $\mathcal{T}$ 的叶节点处可观测到的概率分布集合（由 $[n]$标记）。它是一种带隐变量的离散有向图模型。在局部马尔可夫性质下，树上的联合分布可分解为各条边上的分布。然而，由于内部节点表示潜在（不可观测的）祖先物种，因此在叶节点处观测到特定模式 $i_1 \dots i_n$ 的概率，是通过对内部状态进行边缘化得到的。这定义了多项式映射 $\varphi_\mathcal{T}\colon \Theta \to \mathcal{M}_\mathcal{T}$，其坐标函数如下

$$p_{i_1 \dots i_n} = \sum_{\substack{x_v \in \{\mathtt{A},\mathtt{C},\mathtt{G},\mathtt{T}\}, \\ v \text{ internal}}} \left(\pi(x_r) \prod_{u \to v \in E} M_e(x_u, x_v) \right).$$

因此，系统发育模型 $\mathcal{M}_\mathcal{T}$ 是参数空间 Θ 在映射 $\varphi_\mathcal{T}$下的像。代数统计通过分析系统发育不变量 来研究 $\mathcal{M}_\mathcal{T}$ ——这些是叶概率的多项式，在 $\mathcal{M}_\mathcal{T}$ 上取值为零，并生成定义该模型扎里斯基闭包的理想。这些不变量是区分不同树拓扑的重要工具，无需任何先验知识。

这些结构可通过 `PhylogeneticModel` 类型构造和操作。该对象封装了树拓扑、模型类型以及底层多项式环。

示例 2.1。 在本例中，我们构造一棵具有 3 个叶节点的简单树，并为转移矩阵指定 Jukes–Cantor 模型结构。请注意，虽然我们定义了一个带有变量 和 的通用矩阵

结构，但 phylogenetic model 函数会自动为每条边 i 分配不同的参数（a_i， b_i），从而允许使用不同的分支长度。

```
julia>T = 系统发育_树（图_来自_边(有向, [[4,1], [4,2], [4,3]]))
系数类型为 QQFieldElem 的系统发育树

# 定义 Jukes–Cantor 替换结构
# a：保持不变的概率，b：发生突变的概率
julia> M_JC = [:a :b :b :b;
               :b :a :b :b;
               :b :b :a :b;
               :b :b :b :a];

julia> PM  =系统发育_模型(T, M_JC)
具有 3 个叶节点和 3 条边的树上的系统发育模型
  根分布为 [1//4                  , 1//4, 1//4, 1//4] ，转移矩阵的形式为.....

julia> 参数_环(PM)[1]
有理数域上的 6 元变量多元多项式环 a[1], a[2], a[3], b[1], ..., b[3]


julia>条目_转移_矩阵(PM, 3, 3, 边(4, 1))a[1]


julia>入口_转移_矩阵(PM, 1, 2, 边(4, 2))b[2]
```

函数 model ring_生成叶概率的环境环，而parametrization 返回多项式映射 $\varphi_{\mathcal{T}}$。最后，使用 vanishing ideal计算定义该模型的系统发育不变量：

```
julia> Rp, p = 模型_环(PM); p
包含 5 个条目的 Dict{Tuple{Vararg{Int64}}, 有理数域多元多项式环元素}: (
   1, 1, 1) => p[1,1,1 ]
  (1, 2, 1) => p[1,2,1]
  ...

julia> phi  =参数化(PM);

# 计算所有叶节点都观察到状态 A（索引 1）的概率julia>phi(p[(1, 1, 1)])



julia> 消失_理想(PM)
由以下元素生成的理想
  2p[1,2,3]^3 - p[1,2,3]^2p[1,2,2] - p[1,2,3]^2p[1,2,1] - p[1,2,3]^2p[1,1,2] +p
```

OSCAR 包括若干常见的生物学模型，例如

- 群基模型 jukes cantor model、 kimura2 model和 kimura3 model，
- general markov model，以及 general_time reversible model。

2.2.1. 群基模型。 群基模型（例如，Jukes–Cantor模型、Kimura 2参数或3参数模型）可通过离散傅里叶变换进行特定的线性坐标变换。这一变换将转移矩阵对角化，并将概率坐标转换为傅里叶坐标（记作 $q_{i_1 \ldots i_n}$）。在这个新的坐标系中，参数化映射 ψ_T 简化为关于傅里叶参数（转移矩阵的特征值）的单项式映射。从代数角度看，这将该模型确定为一个环面簇，这种结构显著简化了计算任务，详见 [39] 以了解更多详情。

为了利用这一结构，我们提供了 `GroupBasedPhylogeneticModel`类型。实例通过特定函数（如 `jukes cantor model(T)`）构造，并包含一个底层的 `PhylogeneticModel` 对象，可通过 `phylogenetic model`函数访问。此外，从傅里叶坐标到概率坐标的显式线性变换可通过 `coordinate change`获得，反向映射则由`inverse coordinate change`提供。

示例 2.2. 本例展示了 定理2.1 中定义的模型，使用`GroupBasedPhylogeneticModel`类型直接通过傅里叶参数和傅里叶坐标进行计算。

```
julia> PM  =jukes_cantor_model(T)  # Define a Jukes--Cantor model on the tree T




julia> 参数_环(PM)[1]
有理数域上的 6 元多项式环，变量为 x[1]，x[2]，x[3]，y[1]，...，y[3]


julia> Rq，q = 模型_环(PM); q
包含 5 个条目的 Dict{Tuple{Vararg{Int64}}, QQMPolyRingElem}:
  (1，1，1) => q[1,1,1]
  (2，3，4) => q[2,3,4]
  ...

julia>条目_傅里叶_参数(PM，3，边(4，1))y[1]


julia>条目_傅里叶_参数(PM，1，边(4，2))x[2]


julia> 消去_理想(PM)
由以下元素生成的理想
  -q[2,3,4]^2*q[1,1,1] + q[2,2,1]*q[2,1,2]*q[1,2,2]

julia> 坐标_变换(PM)
从有理数域上 5 个变量的多元多项式环到有理数域上 5 个变量的多元多项式环的环同态，由 q
```

2.2.2. 系统发育网络。 系统发育网络将树推广为能够对杂交、水平基因转移或重组等网状进化事件进行建模的结构。与树不同，网络允许内部节点（称为网状化或杂交节点）的入度大于一。在代数框架中，模型参数不仅包括根分布和转移矩阵，还包括网状化节点处的杂交参数。如果节点 v 有父节点 u_1、.....、u_k，则 v 处的状态由 u_i 处的状态以概率 λ_i 决定，其中 $\sum \lambda_i = 1$。

OSCAR 支持构建1级系统发育网络（即每个双连通分量至多包含一个环）及其相关代数模型，使用 `phylogenetic network`构造器即可。

示例 2.3. 我们在三叶太阳网络 上构建一个Jukes–Cantor模型（定义为由一个中心三环和三个悬挂汇节点组成）。如输出所示，参数环明确包含杂交变量（记为 `l`），多项式映射 `f` 将傅里叶坐标表示为底层树参数化的凸组合。

```
julia> E = [[4,1], [5,2], [6,3], [5,4], [6,4], [5,6]];

julia>N = 系统发育_网络（图_来自_边（有向，E））含有杂交节点{4}



julia> PMN  =Jukes_Cantor_模型(N)  # Define a Jukes--Cantor model on the network




julia> 参数_环(PMN)[1] # The parameter ring now includes hybrid parameters
含 14 个变量的多元多项式环 l[1,                        1], l[1, 2], x[1], x[2], ..., y[6]
  在有理数域上
julia>条目_混合_参数(PMN, 边(5,4))
l[1, 1]

julia> R, q = 模型_环(PMN);

Julia> phi  =参数化(PMN);

julia> phi(q[1,1,1])
l[1, 1]*x[1]*x[2]*x[3]*x[4]*x[5] + l[1, 2]*x[1]*x[2]*x[3]*x[5]*x[6]
```

此代码是创建 小型系统发育树 库的 FAIR 数据库这一更广泛计划的一部分 [10]。 第 4 节 概述了我们如何将这些计算结果整合到数据库中。

3. Computing Vanishing Ideals of Parametric Statistical Models

代数统计中的许多核心问题都涉及计算多项式映射或有理映射（通常涉及许多变量）的像的消失理想，因此从根本上说属于隐式化问题 [39]。解决隐式化任务的标准方法是写出有理映射的图（清除分母后），然后对分母进行饱和，最后计算消去理想的 Gr¨obner 基，以消去参数

。这种方法适用于任意有理映射，且只需实现 parametrization方法即可。然而，这些默认方法在处理大型示例时往往无法终止，尤其是对于系统发育网络这类涉及许多变量的模型。

我们实现了专用化技术，可显著加速图模型和系统发育学中常见的问题类型。这些加速依赖于额外的结构，而这些结构已由代数统计中的定理证明。例如，基于群的系统发育模型在其傅里叶参数下是环面模型（参见第 2.2.1 部分）。了解这一点后，我们便借助 4ti2 软件包 [2] 中的快速工具来计算 Gr¨obner 基。我们还整合了近期开发的两种方法：多重分次隐式化 [19] 和双有理隐式化 [16]。以下示例展示了如何使用这些函数，以及它们带来的优势。除非另有明确说明，所有实验均在运行 Linux 6.12 的机器上进行，该机器配备英特尔 i7-9750H 和 32 GB 主内存。我们的比较使用 OSCAR 1.7.0、 Julia 1.12.4 和 Macaulay2 版本 1.25.11。

示例 3.1。 我们首先展示如何使用我们的软件在四叶日光网络上创建基于群的 Kimura 3 模型 [29] ，然后计算其消失理想中次数不超过三的分量。

```
julia> G1 =图_来自_edges(有向,[[5,1],[6,5],[7,6],[7,8],[8,5],[6,2],[7,3],[8,4]]);

julia> N1 =系统发育_network(G      1)
含有杂交节点 {5} 且边为 (5, 1)(6,2)(6, 5)(7, 3)(7, 6)(7, 8)(8, 4)(8, 5) 的 1级系统发育


julia>M = Kimura模型3_model(N1)
在 1级网络上的基于群的系统发育模型，含有 1 个杂交节点、4 个叶节点和 8 条边，根分布为 [1//


   [:a :b :c :d;
    :b :a :d :c;
    :c :d :a :b;
    :d :c :b :a]
  以及形式为 [:x, :y, :z, :t] 的傅里叶参数。
julia> phi  =参数化(M)
环同态
    从有理数域上含 64 个变量的多元多项式环到有理数域上含 34 个


  q[4,4,4,4]->l[1, 1]*x[6]*t[1]*t[2]*t[3]*t[4]*t[5]*t[7] + ... q



julia> comps = 分量_的_核(3, phi)
Oscar.FinAbGroupElemDict{Vector{QQMPolyRingElem}}，底层字典为




julia>组件[3, 0, 2, 1, 1, 0, 2, 3, 3, 0, 1, 0, 1, 0, 0, 0, 1, 1, 0, 1, 1, 1, 1, 1]1 元素向量
```

叶节点	总次数	最小生成元数	Macaulay2时间（秒）	OSCAR时间（秒）
4	2	12	0.472	0.0498
4	3	64	117	1.06
5	2	648	329	0.951
5	3	18560	-	372

TABLE 1. 比较 Macaulay2 和 OSCAR 中多重分次隐式化实现对基于群的 Kimura 3 模型在四叶和五叶日光网络上的运行时间。

表 1展示了我们实现多重分次隐式化（称为components of kernel）的运行时间 OSCAR ，并与 Macaulay2 中针对 Kimura 3 网络的实现进行了比较。

注 3.2。 多重分次隐式化的部分步骤可以并行化；参见 [19]。由于 OSCAR不是线程安全的，因此并行实现的唯一办法是使用多个进程。也就是说，各进程不共享内存，必须通过序列化（和反序列化）消息进行通信。这会带来一些开销，具体取决于消息的类型及其编码，因此在规模较小的情况下，计算速度往往会明显变慢。例如，表 1中的计算并行运行时耗时长了几个数量级。不过，对于四个状态的三叶一般马尔可夫模型，我们成功计算出了消失理想的 5 次分量。该计算在配备 Intel Xeon Silver 4216 的机器上使用 32 个进程和 150 GB 主内存运行，耗时 2.59 小时。这是一个很大的改进，因为估算非并行计算时间约为两天。为求完整起见，我们展示如何设置该计算。

```
using Oscar tree = 图_由_边(有向,[[4,1],[4,2],[4,3]]) model = 一般
```

计算结果已上传至 Zenodo，可在此处找到 https://zenodo.org/records/18312720；参见第 4 节，了解更多关于存储计算结果的信息。

例 3.3。 Sullivant [38] 推测，高斯 DAG 模型的消失理想可以通过将其条件独立理想在某些主子式处饱和来计算。如今，这一猜想已成为定理 [15]，其证明方法也被推广为一种“双有理隐式化”方案，适用于更多类别的模型，详见 [16]。由此得到的基于饱和的方法是 OSCAR 高斯图模型的默认方法（同样利用了类型和多重分派）。仍可通过“algorithm”关键字参数使用基于消元的方法，下面我们将比较这两种方法。

```
julia> usingOSCAR、TimerOutputs; julia> consttto =
```

```
julia> [@timeit 到 "专用化" 消失_理想(M) 对于 _ 在 1:20]; julia> [@timeit 到 "消元" 消失_理想
```

		Time			分配次数		
总计 / 测量占比:		388秒 / 73.3%			3.36GiB / 0.9%		
部分	调用次数	time	%tot	avg	分配	%tot	avg
消元	20	284s	100.0%	14.2秒	17.5 MiB	55.6%	894 KiB
专用化	20	62.5毫秒	0.0%	3.13毫秒	14.0MiB	44.4%	714KiB

对于这个示例，基于消元的方法平均耗时 14.2 秒，而专用例程仅需 3.1 毫秒，同时占用的内存也略少。在 Macaulay2 [25] 中通过 GraphicalModels包 [8,9] 使用 时间 gaussianVanishingIdeal gaussianRing digraph F 完成相同的消失理想计算则需要 362 秒。

4. Serialization

序列化是将内存中的数据转换为易于存储或传输的格式的过程。数据往往比创建它的软件更有价值。例如，研究科学家或数学家可能对计算结果感兴趣，却没有条件亲自运行该计算。序列化结果的好处是无需重新运行大型计算。我们已为自己的类型实现序列化功能，使用户能够将结果存储为 mrdi 文件格式 [21]，可通过 OSCAR save 和 load 函数使用。 mrdi这种格式是基于 JSON 的格式 [36], ，在许多方面都很实用。首先，由于它是基于文本而非二进制的，因此文件可供人类阅读。其次，JSON 文件可以存入文档型数据库，例如 MongoDB [1] ，这类数据库可提供实用的查询 API。一个例子是 OscarDB。尽管仍处于早期阶段，但目前已可通过 OSCAR 访问，并包含来自 https://algebraicphylogenetics.org 网站的所有系统发育模型 [4]。未来，我们计划让网站从 OscarDB获取数据。下面简要展示如何与 OscarDB交互。

```
julia> 数据库 =Oscar.OscarDB.获取_数据库();

julia> stm = db["AlgebraicStatistics.SmallTreeModels"];

julia>jc_查询 = Oscar.OscarDB.查找(stm, 字典("data.model_type" => "JC"));

julia>jc_模型 = 收集(jc_查询)
6-   元素 Vector{Any}: 小型树系统发育模
```

```
julia> 系统发育_模型(first(jc_models))
具有 3 个叶节点和 3 条边的树上的系统发育模型
  根分布为 [1//4                    , 1//4, 1//4, 1//4] 且转移矩阵的形式为
   [:a :b :b :b; :b :
```

```
julia> 基于群_的_系统发育_模型(first(jc_models))
基于群的系统发育模型，定义在具有 3 个叶节点和 3 条边的树上，根分布为 [1//
```

发现有趣的图模型集合的用户，可以使用我们的序列化方法将其导入自己的 MongoDB，也可以联系我们，将他们的集合展示在 `OscarDB` 中。我们还在之前讨论独立进程之间的通信时提到了序列化方法，见 定理 3.1 。借助我们的序列化方法，用户可以根据需要在多个进程中自由使用我们的图模型类型。

致谢。 我们谨向 Benjamin Lorenz 和 Max Horn 致谢，感谢他们严谨地审查代码。我们也感谢 Tabea Bacher、Chistiane G¨orgen、Marius Neubert 和 Matthias Zach 提供的宝贵意见并参与讨论。T.B. 获得欧盟“地平线 2020”研究与创新计划资助，资助依据为玛丽·斯克·沃多夫斯卡-居里资助协议，编号 101110545。A.D.V 获得德国研究联合会（DFG，德国研究基金会）的资助，具体来自“MaRDI（数学研究数据倡议）”（DFG NFDI 29/1，项目编号 460135501）。M.G-L 部分获得加泰罗尼亚自治区政府研究与大学部 Beatriu de Pin´ós 博士后计划的资助（编号 2024BP00235）。B.H. 部分获得亚历山大·冯·洪堡基金会的资助。

References

[1] MongoDB 文档。https://docs.mongodb.com/，2025年。[2] 4ti2 团队。4ti2——用于线性空间中的代数、几何和组合问题的软件包。网址：https://4ti2.github.io。版本 1.6.9。[3] Yulia Alexandr、Kristen Dawson、Hannah Friedman、Fatemeh Mohammadi、Pardis Semnani 和 Teresa Yu。含隐变量的条件独立理想分解，2025年。[4] 代数系统发育学合作组。代数系统发育学中的小型树和网络数据库，2024年。版本 0.1。[5] Elizabeth S. Allman、Hector Ba˜nos、Marina Garrote-Lopez 和 John A. Rhodes。根据基因树四元组识别1级物种网络。《数学生物学通报》，86(9):110，2024年。[6] Elizabeth S Allman、Catherine Matias 和 John A Rhodes。含有多个观测变量的潜在结构模型中的参数可识别性。《统计年鉴》，37(6A):3099–3132，2009年12月。[7] Elizabeth S. Allman 和 John A. Rhodes。《生物学中的数学模型：导论》。剑桥大学出版社，2004年。

[8] Carlos Am´endola、Luis David Garc´ıa Puente、Roser Homs Pons、Olga Kuznetsova、Harshit J Motwani、Sonja Petrovi´c、Mike Stillman 和 Seth Sullivant。GraphicalModels：离散图模型和高斯图模型。版本 2.0。Macaulay2 软件包，网址：https://github.com/Macaulay2/M2/tree/master/M2/Macaulay2/packages。[9] Carlos Am´endola、Luis David Garc´ıa Puente、Roser Homs Pons、Olga Kuznetsova、Harshit J Motwani、Sonja Petrovi´c、Mike Stillman 和 Seth Sullivant。图模型。《代数与几何软件期刊》，5，2013年。[10] Tabea Bacher、Marina Garrote-L´opez、Christiane G¨orgen 和 Marius J. Neubert。让数学在线资源更加公平：以小型系统发育树为例，2025年。[11] Jeff Bezanson、Alan Edelman、Stefan Karpinski 和 Viral B Shah。Julia：一种数值计算的新方法。《SIAM评论》，59(1):65–98，2017年。[12] Hubert M. Blalock, Jr.，编。《社会科学中的因果模型》。Routledge，第2版，1985年。[13] Tobias Boege。高斯条件独立推断问题。博士论文，马格德堡奥托·冯·格里克大学，2022年。[14] Tobias Boege、Mathias Drton、Benjamin Hollering、Sarah Lumpp、Pratik Misra 和 Daniela Schkoda。平稳扩散中的条件独立。《随机过程及其应用》，2025年。[15] Tobias Boege、Kaie Kubjas、Pratik Misra 和 Liam Solus。有色高斯有向无环图模型。《英国皇家统计学会会刊B辑：统计方法学》，第 qkaf068 页，2025年11月。[16] Tobias Boege 和 Liam Solus。统计模型的实双有理隐式化，2024年。[17] Fabrizio Catanese、Serkan Ho¸sten、Amit Khetan 和 Bernd Sturmfels。极大似然度。《美国数学杂志》，128(3):671–697，2006年。[18] Jane Ivy Coons、Shelby Cox、Aida Maraj 和 Ikenna Nometa。布朗运动树模型的极大似然度：星形树与根不变性。《符号计算杂志》，132:22，2026年。编号/文章号 102482。[19] Joseph Cummings 和 Benjamin Hollering。多重分次多项式映射的隐式化计算。《符号计算期刊》，132，2026年。[20] Wolfram Decker、Christian Eder、Claus Fieker、Max Horn 和 Michael Joswig，编。《计算机代数系统 OSCAR：算法与示例》，《数学中的算法与计算》第32卷。施普林格，第1版，2025年。[21] Antony Della Vecchia、Michael Joswig 和 Benjamin Lorenz。数学软件的公平文件格式。载于 Kevin Buzzard、Alicia Dickenstein、Bettina Eick、Anton Leykin 和 Yue Ren 编，《数学软件——ICMS 2024》，第234–244页，Cham，2024年。施普林格·自然瑞士。[22] Persi Diaconis 和 Bernd Sturmfels。用于从条件分布中抽样的代数算法。《统计年鉴》，26(1):363–397，1998年。[23] Mathias Drton、Marina Garrote-L´opez、Niko Nikov、Elina Robeva 和 Y. Samuel Wang。含不相交环的线性非高斯模型的因果发现。载于 S Chiappa 和 S Magliacane 编，《第41届UAI会议论文集》，《机器学习研究论文集》第286卷，第1064–1083页，2025年。[24] Alex Fink。条件概率交集公理的二项式理想。《代数组合杂志》，33(3):455–463，2011年。[25] Daniel R. Grayson 和 Michael E. Stillman。Macaulay2：用于代数几何研究的软件系统。网址：https://www.math.uiuc.edu/Macaulay2/。版本 1.16。[26] Hannah G¨obel 和 Pratik Misra。有色高斯环的线性关系，2025年。[27] Søren Højsgaard 和 Steffen L. Lauritzen。具有边与顶点对称性的图高斯模型。《英国皇家统计学会会刊B辑：统计方法学》，70(5):1005–1027，2008年。[28] Qiang Ji。《计算机视觉中的概率图模型》。学术出版社，2019年。[29] 木村资生。估计同源核苷酸序列之间的进化距离。《美国国家科学院院刊》，78(1):454–458，1981年1月。[30] Steffen Lauritzen、Caroline Uhler 和 Piotr Zwiernik。全正性条件下高斯模型中的最大似然估计。《统计年鉴》，47(4):1835–1863，2019年。[31] Steffen L. Lauritzen。《图模型》，《牛津统计科学丛书》第17卷。牛津大学出版社，1996年。[32] Marloes Maathuis、Mathias Drton、Steffen Lauritzen 和 Martin Wainwright，编。《图模型手册》。《Chapman Hall/CRC现代统计方法手册》。CRC出版社，2019年。

[33] Franti˘sek Mat´u˘s。《高斯向量中的条件独立与多项式环》。载于 Gabriele Kern-Isberner、Wilhelm R¨odder 和 Friedhelm Kulmann 编，《条件、信息与推理》，第152–161页。施普林格，2005年。[34] OSCAR——开源计算机代数研究系统，1.7.0版，2026年。[35] Judea Pearl。《智能系统中的概率推理：可信推理网络》。Morgan Kaufmann 表征与推理系列。Morgan Kaufmann，加利福尼亚州圣马特奥，1988年。[36] Felipe Pezoa、Juan L Reutter、Fernando Suarez、Mart´ín Ugarte 和 Domagoj Vrgo˘c。《JSON 模式基础》。载于《第*25*届万维网国际会议论文集》，第263–273页。国际万维网会议指导委员会，2016年。[37] Charles Semple 和 Mike Steel。《系统发育学》，《牛津数学与应用讲座系列》第24卷。牛津大学出版社，2003年。[38] Seth Sullivant。高斯贝叶斯网络的代数几何。《应用数学进展》，40(4)：482–513，2008年。[39] Seth Sullivant。《代数统计》，《数学研究生课程》第194卷。美国数学会，2018年。[40] A. Taeb、J. T. Reager、M. Turmon 和 V. Chandrasekaran。加利福尼亚州水库系统的统计图模型。《水资源研究》，53(11)：9721–9739，2017年。[41]Y. Samuel Wang 和 Mathias Drton。非高斯性条件下的高维因果发现。《*Biometrika*》，107(1)：41–59，2019年。[42] Herman Wold。因果关系与计量经济学。《*Econometrica*》，22(2)：162–177，1954年。

UiT – The Arctic University of Norway电子邮箱地址：post@taboege.de

Technical University of Berlin电子邮箱地址：vecchia@math.tu-berlin.de

Universitat Pompeu Fabra电子邮箱地址：marina.garrote@upf.edu

Max Planck Institute for Mathematics in the Sciences电子邮箱地址：benjamin.hollering@mis.mpg.de