Representing Polynomial Ideals as Heterogeneous Graphs for Inductive Machine Learning

Rüdiger Nather
r.nather@uni-kassel.de

UNIKASSEL
VERSITÄT

7th July 2026

SCML
International Conference on Symbolic Computation and Machine Learning

Motivation

- Applying machine learning to mathematical structures is a rapidly growing topic
- Many machine learning approaches are build on finite-dimensional representations of their input data

Motivation – Polynomial Algebra

- Let's apply ML to polynomial algebra
  $\rightsquigarrow$ Specifically: generating sets of ideals
- Problem 1: Representation sets of polynomials/monomials/terms/...
  by single vector $\rightsquigarrow$ we lose information
- Problem 2: Representation of the underlying ring,
  typically implicit $\rightsquigarrow$ limited generalizability

Motivation – Polynomial Algebra

- Let's apply ML to polynomial algebra
  $\rightsquigarrow$ Specifically: generating sets of ideals

- Problem 1: Representation sets of polynomials/monomials/terms/...
  by single vector $\rightsquigarrow$ we lose information

- Problem 2: Representation of the underlying ring,
  typically implicit $\rightsquigarrow$ limited generalizability

- What if we just... don't?

Graph Neural Networks (GNNs)

- Operates on graphs, each node gets its own embedding
- Nodes aggregate information from their neighbours iteratively
  $\rightsquigarrow$ each node's embedding reflects its local graph structure
- GNNs operate on graphs of *arbitrary* size
  $\rightsquigarrow$ no fixed-dimensional bottleneck
  $\rightsquigarrow$ the architecture scales with the input
- Parameters are shared across all nodes and edges
  $\rightsquigarrow$ a model can generalise to graphs (and nodes/edges) that were *not* present during training

From Polynomials to Graphs

- Represent input as faithfully as possible
- What information do we need to include?
  - What are the nodes?
  - What are the edges?
  - Weighted graphs? Directed? Something else?

From Polynomials to Graphs

- Learning a heuristic for identifying the leading term
- What does leading term depend on?
  - (number of) variables
  - term/variable order
  - monomials in a polynomial

Observation

- Depends on the ring $\rightsquigarrow$ ideally included *in* the graph
- Need to tell the different components apart
- Different types of nodes and edges $\rightsquigarrow$ Heterogeneous graphs

Construction – I

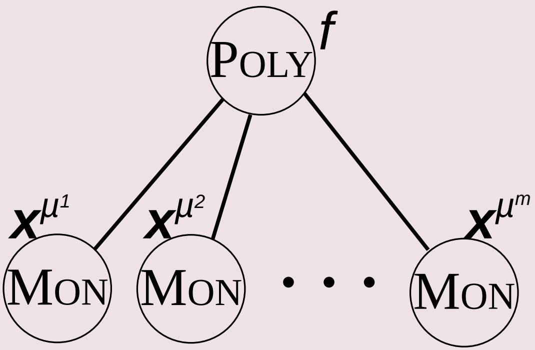
- Polynomials are sums of monomials
- Monomials are products of variables (plus coefficient)
- Monomials are ordered, order is derived from relation between variables

- Poly-node for each polynomial
- Mon-node for each monomial in a polynomial
- Var-node for each variable
- Edges between Var-nodes encode term order

Construction – II

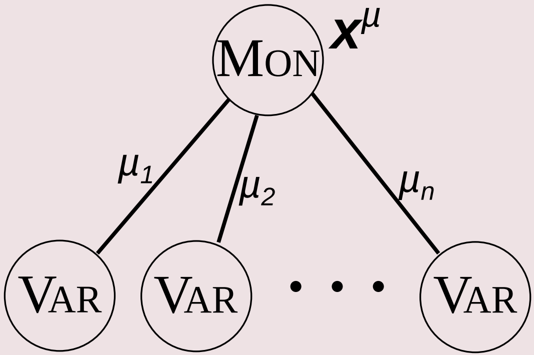
Polynomial

$$f = c_1 \mathbf{x}^{\mu^1} + \cdots + c_m \mathbf{x}^{\mu^m}$$



Monomial

$$\mathbf{x}^{\mu} = x_1^{\mu_1} \cdots x_n^{\mu_n}$$



Construction – III

“Every term order can be represented by matrix order” (Robbiano 1985)

lexicographic:

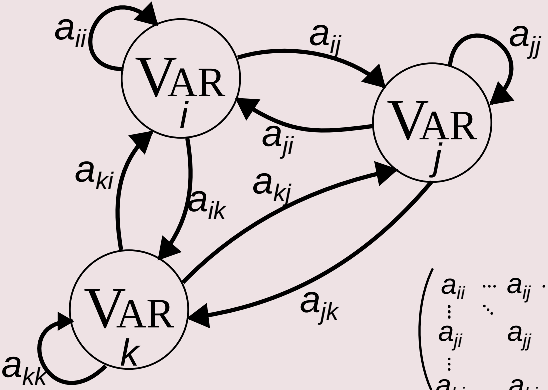
$$\begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & 1 \end{pmatrix}$$

degree reverse lexicographic:

$$\begin{pmatrix} 1 & 1 & \cdots & 1 \\ 0 & \cdots & 0 & -1 \\ \vdots & & .\cdot^{\cdot} & 0 \\ 0 & -1 & 0 & 0 \end{pmatrix}$$

Construction – III

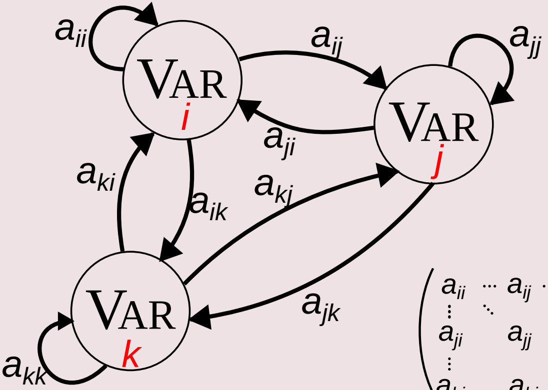
Variables



$$\begin{pmatrix} a_{ii} & \cdots & a_{ij} & \cdots & a_{ik} \\ \vdots & \ddots & & & \vdots \\ a_{ji} & & a_{jj} & & a_{jk} \\ \vdots & & & \ddots & \vdots \\ a_{ki} & \cdots & a_{kj} & \cdots & a_{kk} \end{pmatrix}$$

Construction – III

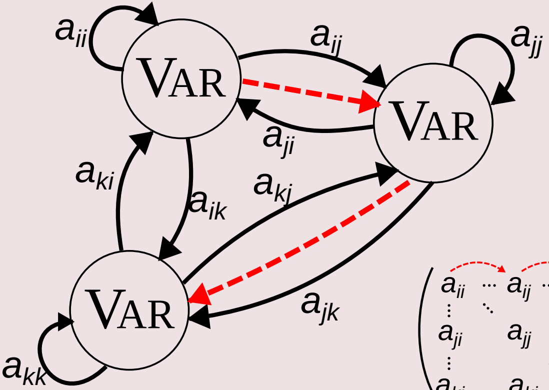
Variables



$$\begin{pmatrix} a_{ii} & \cdots & a_{ij} & \cdots & a_{ik} \\ \vdots & \ddots & & & \vdots \\ a_{ji} & & a_{jj} & & a_{jk} \\ \vdots & & & \ddots & \vdots \\ a_{ki} & \cdots & a_{kj} & \cdots & a_{kk} \end{pmatrix}$$

Construction – III

Variables

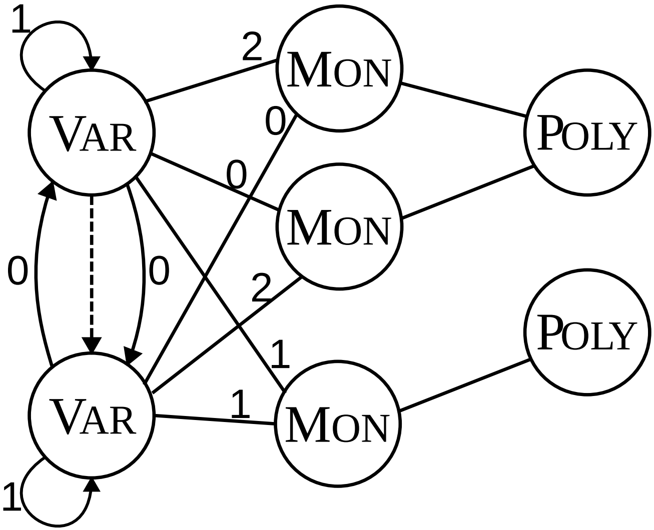


$$\begin{pmatrix} a_{ii} & \cdots & a_{ij} & \cdots & a_{ik} \\ \vdots & \ddots & & & \vdots \\ a_{ji} & & a_{jj} & & a_{jk} \\ \vdots & & & \ddots & \vdots \\ a_{ki} & \cdots & a_{kj} & \cdots & a_{kk} \end{pmatrix}$$

Example

$R = \mathbb{K}[x, y]$, *lex*-order

$$I = \langle x^2 + y^2, x\,y \rangle$$



Conclusion

- Presented a construction that encodes generating sets as heterogenous graphs
- Demonstrated how this construction is information-preserving
- Outlined how this enables inductive learning

Conclusion

- Presented a construction that encodes generating sets as heterogenous graphs
- Demonstrated how this construction is information-preserving
- Outlined how this enables inductive learning
- Alas, no experimental evalutation yet... (Sorry!)

Thank you for your attention!

Please feel free to ask questions!