

Minimization of Graph Weighted Models over Circular Strings

Guillaume Rabusseau

McGill University - IVADO - Reasoning and Learning Lab



ETAPS / FoSSaCS 2018 - Thessaloniki

Learning with Structured Data

Supervised Learning:

Learn $f : \mathcal{X} \rightarrow \mathcal{Y}$ from a sample $\{(x_1, y_1), \dots, (x_N, y_N)\} \subset \mathcal{X} \times \mathcal{Y}$.

Learning with Structured Data

Supervised Learning:

Learn $f : \mathcal{X} \rightarrow \mathcal{Y}$ from a sample $\{(x_1, y_1), \dots, (x_N, y_N)\} \subset \mathcal{X} \times \mathcal{Y}$.

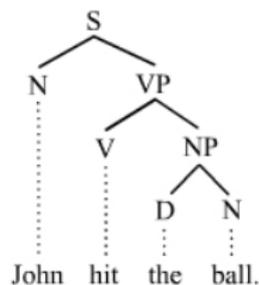
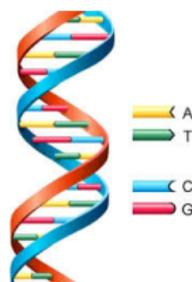
- Learning algorithms often assume $\mathcal{X} = \mathbb{R}^d$.

Learning with Structured Data

Supervised Learning:

Learn $f : \mathcal{X} \rightarrow \mathcal{Y}$ from a sample $\{(x_1, y_1), \dots, (x_N, y_N)\} \subset \mathcal{X} \times \mathcal{Y}$.

- Learning algorithms often assume $\mathcal{X} = \mathbb{R}^d$.
- How can one **learn with structured inputs** such as strings and trees?



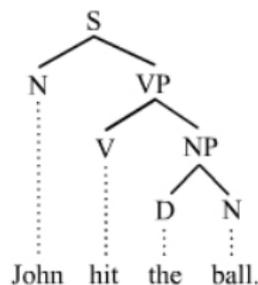
- Intersection of **Theoretical Computer Science** and **Machine Learning**...

Learning with Structured Data

Supervised Learning:

Learn $f : \mathcal{X} \rightarrow \mathcal{Y}$ from a sample $\{(x_1, y_1), \dots, (x_N, y_N)\} \subset \mathcal{X} \times \mathcal{Y}$.

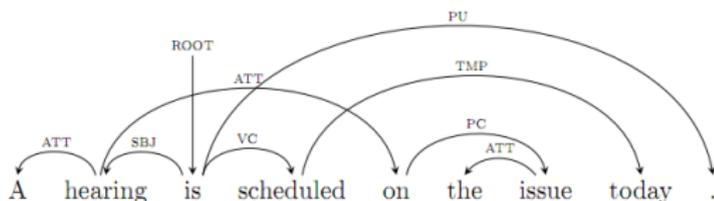
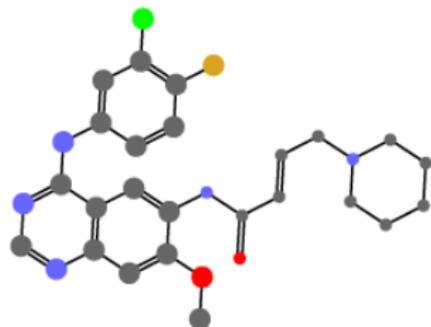
- Learning algorithms often assume $\mathcal{X} = \mathbb{R}^d$.
- How can one **learn with structured inputs** such as strings and trees?



- Intersection of **Theoretical Computer Science** and **Machine Learning**...
- **Weighted Automata**: robust model to represent functions defined over structured objects (and in particular **probability distributions**).

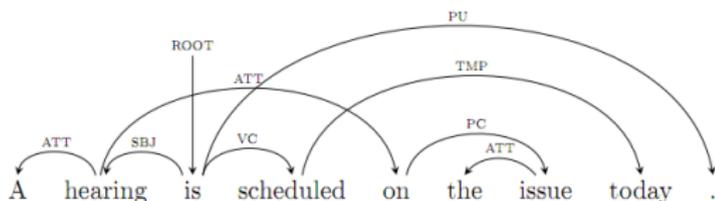
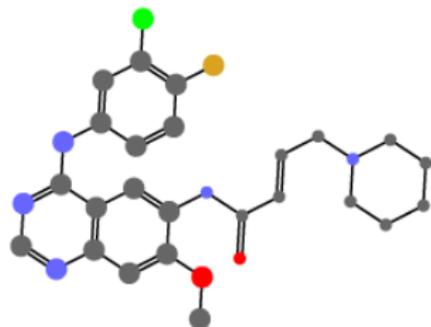
Long Term Objective

- Long term objective: **learning algorithm** for functions defined over **graphs**.
- **Spectral learning** of weighted automata has been proposed for strings [Bailly et al., Hsu et. al, 2009] and trees [Bailly et al., 2010]



Long Term Objective

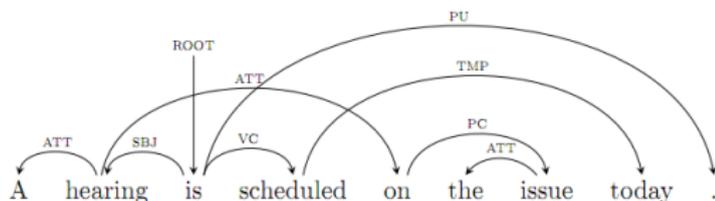
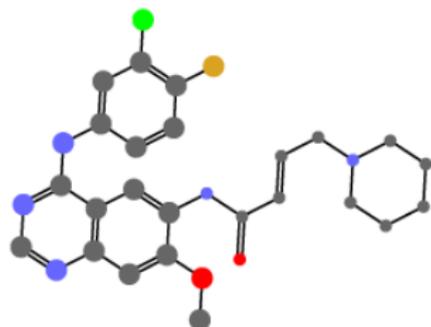
- Long term objective: **learning algorithm** for functions defined over **graphs**.
- **Spectral learning** of weighed automata has been proposed for strings [Bailly et al., Hsu et. al, 2009] and trees [Bailly et al., 2010]



- Extending automata to graphs is challenging.
 - No suitable model of weighed graph automata...

Long Term Objective

- Long term objective: **learning algorithm** for functions defined over **graphs**.
- **Spectral learning** of weighed automata has been proposed for strings [Bailly et al., Hsu et. al, 2009] and trees [Bailly et al., 2010]



- Extending automata to graphs is challenging.
 - No suitable model of weighted graph automata...
- Tensor networks view of weighted automata can naturally be extended to labeled graphs :
 - ↪ **Graph Weighted Models** (GWMs) [Bailly, GR, Denis, LATA'15, JCSS'18]

Minimization of GWMs over Circular Strings

- Long term objective: learning algorithm for GWMs (over \mathbb{R})
→ Tedious task...
- In this work,
 - ▶ we focus on **GWMs restricted to the family of circular strings** (GWM^c),
 - ▶ we study the **equivalence and minimization problems**.
- These problems are more challenging with GWM^c than with weighted automata (WA):
 - ▶ $\text{WA (over } \mathbb{R}) \simeq \text{Linear algebra}$
 - ▶ $\text{GWM}^c \simeq \text{Non-commutative algebra}$

Outline

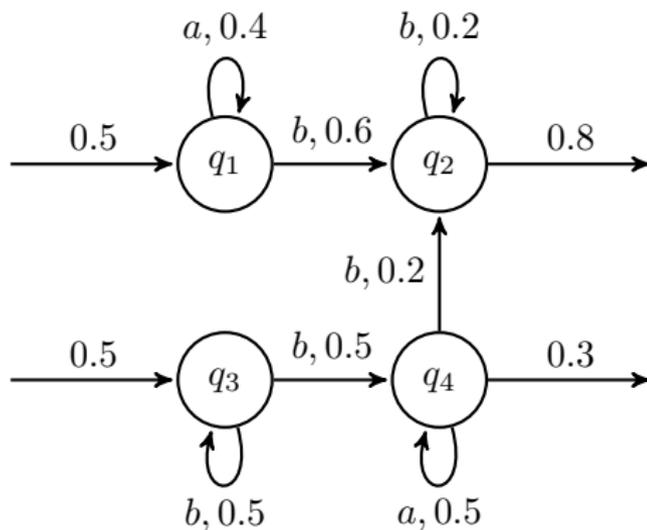
- 1 Preliminaries
 - Weighted Automata
 - Tensor Networks and Weighted Automata
 - Graph Weighted Models
- 2 Graph Weighted Model over Circular Strings (GWM^c)
- 3 Equivalence of GWM^c s
- 4 Minimization of GWM^c s

String Weighted Automata (WA)

- Σ a finite alphabet (e.g. $\{a, b\}$), Σ^* strings on Σ (e.g. *abba*)
- A WA computes a function $f : \Sigma^* \rightarrow \mathbb{R}$

String Weighted Automata (WA)

- Σ a finite alphabet (e.g. $\{a, b\}$), Σ^* strings on Σ (e.g. $abba$)
- A WA computes a function $f : \Sigma^* \rightarrow \mathbb{R}$
- States and weighted transitions:



String Weighted Automata (WA)

- Σ a finite alphabet (e.g. $\{a, b\}$), Σ^* strings on Σ (e.g. $abba$)
- A WA computes a function $f : \Sigma^* \rightarrow \mathbb{R}$

- Algebraic representation: WA M with n states:

$M = (\alpha, \{\mathbf{M}^\sigma\}_{\sigma \in \Sigma}, \omega)$ where

$\alpha \in \mathbb{R}^n$ initial weights vector

$\omega \in \mathbb{R}^n$ final weights vector

$\mathbf{M}^\sigma \in \mathbb{R}^{n \times n}$ transition weights matrix for each $\sigma \in \Sigma$

String Weighted Automata (WA)

- Σ a finite alphabet (e.g. $\{a, b\}$), Σ^* strings on Σ (e.g. $abba$)
- A WA computes a function $f : \Sigma^* \rightarrow \mathbb{R}$

- Algebraic representation: WA M with n states:

$M = (\alpha, \{\mathbf{M}^\sigma\}_{\sigma \in \Sigma}, \omega)$ where

$\alpha \in \mathbb{R}^n$ initial weights vector

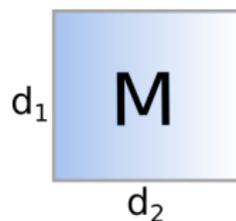
$\omega \in \mathbb{R}^n$ final weights vector

$\mathbf{M}^\sigma \in \mathbb{R}^{n \times n}$ transition weights matrix for each $\sigma \in \Sigma$

- M computes a function $f_M : \Sigma^* \rightarrow \mathbb{R}$ defined by

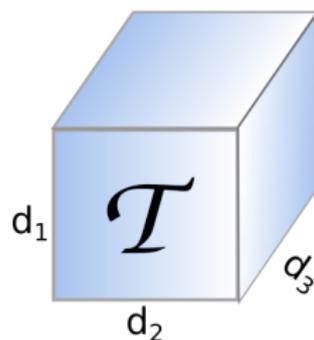
$$f_M(\sigma_1\sigma_2 \cdots \sigma_k) = \alpha^\top \mathbf{M}^{\sigma_1} \mathbf{M}^{\sigma_2} \cdots \mathbf{M}^{\sigma_k} \omega$$

Detour: Tensors



$$\mathbf{M} \in \mathbb{R}^{d_1 \times d_2}$$

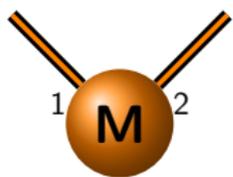
$$\mathbf{M}_{ij} \in \mathbb{R} \text{ for } i \in [d_1], j \in [d_2]$$



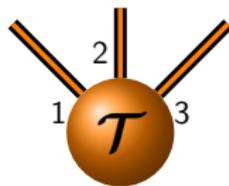
$$\mathcal{T} \in \mathbb{R}^{d_1 \times d_2 \times d_3}$$

$$(\mathcal{T}_{ijk}) \in \mathbb{R} \text{ for } i \in [d_1], j \in [d_2], k \in [d_3]$$

Detour: Tensor Networks

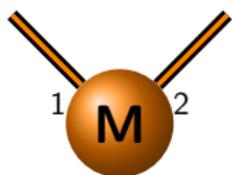


Matrix: $\mathbf{M}_{i_1 i_2}$

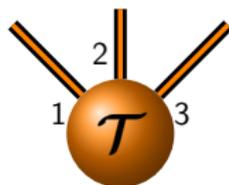


3rd order tensor: $\mathcal{T}_{i_1 i_2 i_3}$

Detour: Tensor Networks

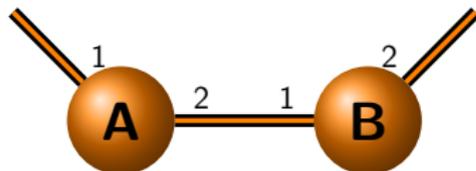


Matrix: $\mathbf{M}_{i_1 i_2}$



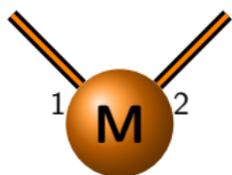
3rd order tensor: $\mathcal{T}_{i_1 i_2 i_3}$

Matrix product:

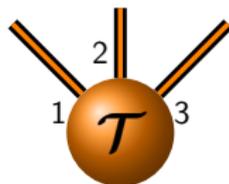


$$(\mathbf{AB})_{i_1, i_2} = \sum_k \mathbf{A}_{i_1 k} \mathbf{B}_{k i_2}$$

Detour: Tensor Networks

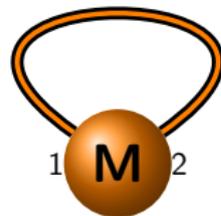


Matrix: $\mathbf{M}_{i_1 i_2}$



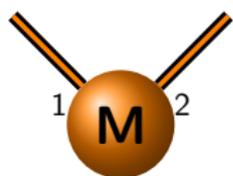
3rd order tensor: $\mathcal{T}_{i_1 i_2 i_3}$

Trace:

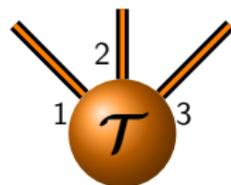


$$\text{Tr}(\mathbf{M}) = \sum_i \mathbf{M}_{ii}$$

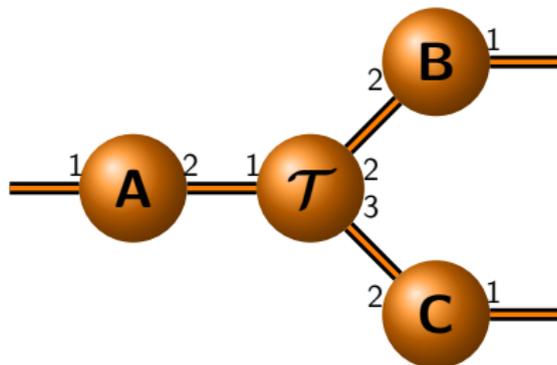
Detour: Tensor Networks



Matrix: $\mathbf{M}_{i_1 i_2}$



3rd order tensor: $\mathcal{T}_{i_1 i_2 i_3}$



Tensor times matrices:

$$(\mathcal{T} \times_1 \mathbf{A} \times_2 \mathbf{B} \times_3 \mathbf{C})_{i_1, i_2, i_3} = \sum_{k_1 k_2 k_3} \mathcal{T}_{k_1 k_2 k_3} \mathbf{A}_{i_1 k_1} \mathbf{B}_{i_2 k_2} \mathbf{C}_{i_3 k_3}$$

String Weighted Automata (WA)

- WA M with n states: $M = (\alpha, \{\mathbf{M}^\sigma\}_{\sigma \in \Sigma}, \omega)$ where
 - $\alpha \in \mathbb{R}^n$ initial weights vector
 - $\omega \in \mathbb{R}^n$ final weights vector
 - $\mathbf{M}^\sigma \in \mathbb{R}^{n \times n}$ transition weights matrix for each $\sigma \in \Sigma$

- M computes a function $f_M : \Sigma^* \rightarrow \mathbb{R}$ defined by

$$f_M(\sigma_1\sigma_2 \cdots \sigma_k) = \alpha^\top \mathbf{M}^{\sigma_1} \mathbf{M}^{\sigma_2} \cdots \mathbf{M}^{\sigma_k} \omega$$

String Weighted Automata (WA)

- WA M with n states: $M = (\alpha, \{\mathbf{M}^\sigma\}_{\sigma \in \Sigma}, \omega)$ where
 - $\alpha \in \mathbb{R}^n$ initial weights vector
 - $\omega \in \mathbb{R}^n$ final weights vector
 - $\mathbf{M}^\sigma \in \mathbb{R}^{n \times n}$ transition weights matrix for each $\sigma \in \Sigma$
- M computes a function $f_M : \Sigma^* \rightarrow \mathbb{R}$ defined by

$$f_M(\sigma_1\sigma_2 \cdots \sigma_k) = \alpha^\top \mathbf{M}^{\sigma_1} \mathbf{M}^{\sigma_2} \cdots \mathbf{M}^{\sigma_k} \omega$$

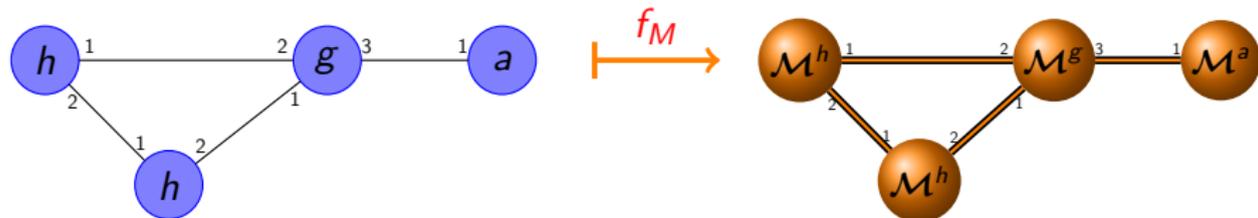


Graph Weighted Models [R. Bailly*, GR*, F. Denis, LATA'15/JCSS'18]

- $\mathcal{F} = \{a(\cdot), h(\cdot, \cdot), g(\cdot, \cdot, \cdot)\}$
- GWM: vector $\mathcal{M}^a \in \mathbb{R}^n$, matrix $\mathcal{M}^h \in \mathbb{R}^{n \times n}$, tensor $\mathcal{M}^g \in \mathbb{R}^{n \times n \times n}$

Graph Weighted Models [R. Bailly*, GR*, F. Denis, LATA'15/JCSS'18]

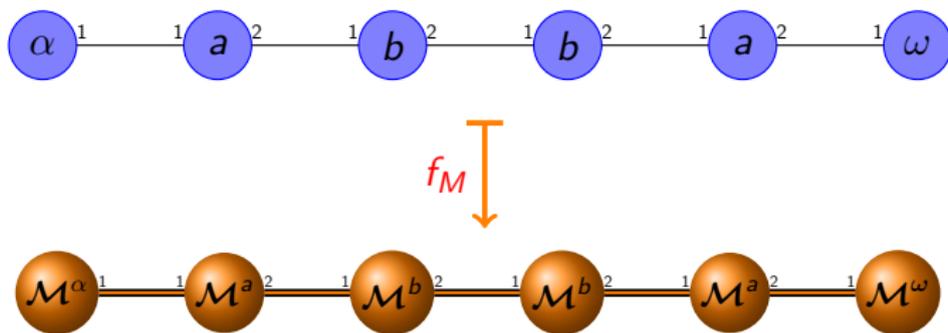
- $\mathcal{F} = \{a(\cdot), h(\cdot, \cdot), g(\cdot, \cdot, \cdot)\}$
- GWM: vector $\mathcal{M}^a \in \mathbb{R}^n$, matrix $\mathcal{M}^h \in \mathbb{R}^{n \times n}$, tensor $\mathcal{M}^g \in \mathbb{R}^{n \times n \times n}$



$$f_M(G) = \sum_{i_1, i_2, i_3, i_4 \in [n]} \mathcal{M}_{i_1, i_2}^h \mathcal{M}_{i_2, i_3}^h \mathcal{M}_{i_3, i_1, i_4}^g \mathcal{M}_{i_4}^a$$

Graph Weighted Models [R. Bailly*, **GR***, F. Denis, LATA'15/JCSS'18]

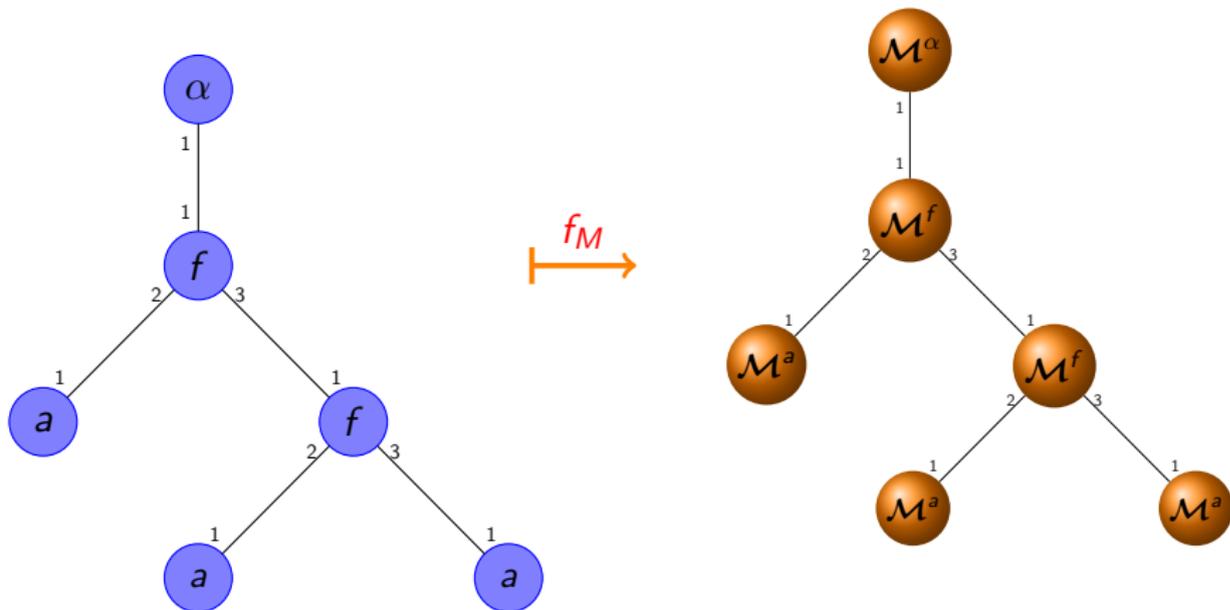
- $\mathcal{F} = \{\alpha(\cdot), a(\cdot, \cdot), b(\cdot, \cdot), \omega(\cdot)\}$
- GWM: $\mathcal{M}^\alpha, \mathcal{M}^\omega \in \mathbb{R}^n$, $\mathcal{M}^a, \mathcal{M}^b \in \mathbb{R}^{n \times n}$



$$\begin{aligned}
 f_M(G) &= \sum_{i_1, i_2, i_3, i_4, i_5 \in [n]} \mathcal{M}_{i_1}^\alpha \mathcal{M}_{i_1, i_2}^a \mathcal{M}_{i_2, i_3}^b \mathcal{M}_{i_3, i_4}^b \mathcal{M}_{i_4, i_5}^a \mathcal{M}_{i_5}^\omega \\
 &= \alpha^\top \mathbf{M}^a \mathbf{M}^b \mathbf{M}^b \mathbf{M}^a \omega
 \end{aligned}$$

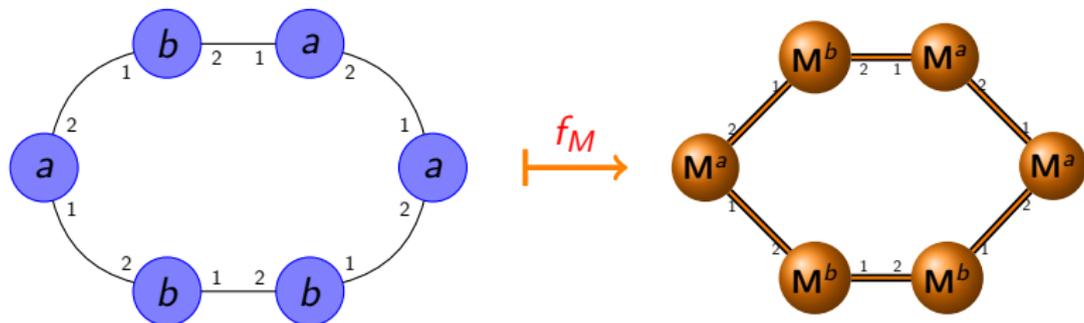
Graph Weighted Models [R. Bailly*, GR*, F. Denis, LATA'15/JCSS'18]

- $\mathcal{F} = \{f(\cdot, \cdot, \cdot), a(\cdot), \alpha(\cdot)\}$
- GWM: $\mathcal{M}^f \in \mathbb{R}^{n \times n \times n}$, $\mathcal{M}^\alpha, \mathcal{M}^a \in \mathbb{R}^n$



Graph Weighted Models [R. Bailly*, GR*, F. Denis, LATA'15/JCSS'18]

- $\mathcal{F} = \{a(\cdot, \cdot), b(\cdot, \cdot)\}$
- GWM: $\mathbf{M}^a, \mathbf{M}^b \in \mathbb{R}^{n \times n}$



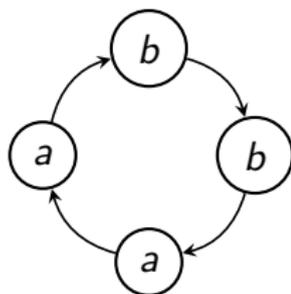
$$\begin{aligned}
 f_M(G) &= \sum_{i_1, i_2, i_3, i_4, i_5, i_6 \in [n]} \mathbf{M}_{i_1, i_2}^a \mathbf{M}_{i_2, i_3}^b \mathbf{M}_{i_3, i_4}^a \mathbf{M}_{i_4, i_5}^a \mathbf{M}_{i_5, i_6}^b \mathbf{M}_{i_6, i_1}^b \\
 &= \text{Tr}(\mathbf{M}^a \mathbf{M}^b \mathbf{M}^a \mathbf{M}^a \mathbf{M}^b \mathbf{M}^b)
 \end{aligned}$$

Outline

- 1 Preliminaries
- 2 Graph Weighted Model over Circular Strings (GWM^c)
- 3 Equivalence of GWM^c s
- 4 Minimization of GWM^c s

Graph Weighted Model over Circular Strings (GWM^c)

- A **circular string** is a string closed onto itself.



→ $aaba \simeq abaa \simeq baaa \simeq aaab$

- A d -dimensional **GWM^c** A is given by $\{\mathbf{A}^\sigma\}_{\sigma \in \Sigma} \subset \mathbb{R}^{d \times d}$
- A computes a function $f_A : \Sigma^+ \rightarrow \mathbb{R}$ defined by

$$f_A(\sigma_1 \sigma_2 \cdots \sigma_k) = \text{Tr}(\mathbf{A}^{\sigma_1} \mathbf{A}^{\sigma_2} \cdots \mathbf{A}^{\sigma_k})$$

Relations between GWM^c and WA

- WAs and GWM^cs are closely related:

$$f_M(x) = \alpha^\top \mathbf{M}^{x_1} \mathbf{M}^{x_2} \dots \mathbf{M}^{x_k} \omega \quad \text{vs.} \quad f_A(x) = \text{Tr}(\mathbf{A}^{x_1} \mathbf{A}^{x_2} \dots \mathbf{A}^{x_k})$$

Relations between GWM^c and WA

- WAs and GWM^c s are closely related:

$$f_M(x) = \alpha^\top \mathbf{M}^{x_1} \mathbf{M}^{x_2} \dots \mathbf{M}^{x_k} \omega \quad \text{vs.} \quad f_A(x) = \text{Tr}(\mathbf{A}^{x_1} \mathbf{A}^{x_2} \dots \mathbf{A}^{x_k})$$

In fact, $\text{GWM}^c \subsetneq \text{WA}$:

- f is WA-recognizable $\not\Rightarrow$ f is GWM^c -recognizable.
 - \hookrightarrow any f that is not invariant under cyclic permutations.
 - \hookrightarrow the constant function $f : x \mapsto 0.5$

Relations between GWM^c and WA

- WAs and GWM^c s are closely related:

$$f_M(x) = \alpha^\top \mathbf{M}^{x_1} \mathbf{M}^{x_2} \dots \mathbf{M}^{x_k} \omega \quad \text{vs.} \quad f_A(x) = \text{Tr}(\mathbf{A}^{x_1} \mathbf{A}^{x_2} \dots \mathbf{A}^{x_k})$$

In fact, $\text{GWM}^c \subsetneq \text{WA}$:

- f is WA-recognizable $\not\Rightarrow$ f is GWM^c -recognizable.
 - \hookrightarrow any f that is not invariant under cyclic permutations.
 - \hookrightarrow the constant function $f : x \mapsto 0.5$
- f is GWM^c -recognizable \Rightarrow f is WA-recognizable.
 - \hookrightarrow If $\mathbf{M}^\sigma = \mathbf{I} \otimes \mathbf{A}^\sigma$ and $\alpha = \omega = \text{vec}(\mathbf{I})$ then $f_M(x) = f_A(x)$ for all $x \in \Sigma^*$.

Outline

- 1 Preliminaries
- 2 Graph Weighted Model over Circular Strings (GWM^c)
- 3 Equivalence of GWM^c s
- 4 Minimization of GWM^c s

Equivalence Problem

- Given two GWM^c can we decide if they compute the same function?

Equivalence Problem

- Given two GWM^c can we decide if they compute the same function?
- **YES**: Equivalence of real-valued WAs is decidable in polynomial time...
 1. Convert the two GWM^c into WAs
 2. Use WAs algorithm to decide equivalence

Equivalence Problem

- Given two GWM^c can we decide if they compute the same function?
- **YES**: Equivalence of real-valued WAs is decidable in polynomial time...
 1. Convert the two GWM^c into WAs
 2. Use WAs algorithm to decide equivalence
- What if we do not want to rely on WAs algorithms?
 - ↪ This will give insights on fundamental GWM properties...

Equivalence of Weighted Automata

Proposition

Given two WAs $M_1 = (\alpha_1, \{\mathbf{M}_1^\sigma\}_{\sigma \in \Sigma}, \omega_1)$, $M_2 = (\alpha_2, \{\mathbf{M}_2^\sigma\}_{\sigma \in \Sigma}, \omega_2)$ with n states,

$f_{M_1} = f_{M_2} \Leftrightarrow M_1$ and M_2 are related by a change of basis

i.e. $\alpha_1^\top \mathbf{P} = \alpha_2^\top$, $\mathbf{P}^{-1} \omega_1 = \omega_2$, $\mathbf{P}^{-1} \mathbf{M}_1^\sigma \mathbf{P} = \mathbf{M}_2^\sigma \forall \sigma \in \Sigma$ for some \mathbf{P} .

Equivalence of Weighted Automata

Proposition

Given two WAs $M_1 = (\alpha_1, \{\mathbf{M}_1^\sigma\}_{\sigma \in \Sigma}, \omega_1)$, $M_2 = (\alpha_2, \{\mathbf{M}_2^\sigma\}_{\sigma \in \Sigma}, \omega_2)$ with n states,

$f_{M_1} = f_{M_2} \Leftrightarrow M_1$ and M_2 are related by a change of basis

i.e. $\alpha_1^\top \mathbf{P} = \alpha_2^\top$, $\mathbf{P}^{-1} \omega_1 = \omega_2$, $\mathbf{P}^{-1} \mathbf{M}_1^\sigma \mathbf{P} = \mathbf{M}_2^\sigma \forall \sigma \in \Sigma$ for some \mathbf{P} .

\hookrightarrow Not true for GWM^c ! With $\Sigma = \{\sigma\}$:

$$\mathbf{A}^\sigma = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \quad \text{and} \quad \tilde{\mathbf{A}}^\sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

we have $f(x) = \tilde{f}(x) = 2$ for all $x \in \Sigma^*$ but \mathbf{A}^σ is not diagonalizable...

Detour: Finite-dimensional Algebras

A few definitions:

- An algebra \mathcal{A} is **nilpotent** if $\exists n: \mathcal{A}^n = \{0\}$.
- The **radical** $\text{Rad}(\mathcal{A})$ of \mathcal{A} is its maximal nilpotent ideal.
- An algebra is **semi-simple** if its radical is $\{0\}$.

Detour: Finite-dimensional Algebras

A few definitions:

- An algebra \mathcal{A} is **nilpotent** if $\exists n: \mathcal{A}^n = \{0\}$.
- The **radical** $\text{Rad}(\mathcal{A})$ of \mathcal{A} is its maximal nilpotent ideal.
- An algebra is **semi-simple** if its radical is $\{0\}$.

Example. Let \mathcal{G} be the algebra generated by $\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$. We have

$$\bullet \mathcal{G} = \left\{ \begin{bmatrix} \alpha & \beta \\ 0 & \alpha \end{bmatrix} : \alpha, \beta \in \mathbb{R} \right\}$$

Detour: Finite-dimensional Algebras

A few definitions:

- An algebra \mathcal{A} is **nilpotent** if $\exists n: \mathcal{A}^n = \{0\}$.
- The **radical** $\text{Rad}(\mathcal{A})$ of \mathcal{A} is its maximal nilpotent ideal.
- An algebra is **semi-simple** if its radical is $\{0\}$.

Example. Let \mathcal{G} be the algebra generated by $\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$. We have

- $\mathcal{G} = \left\{ \begin{bmatrix} \alpha & \beta \\ 0 & \alpha \end{bmatrix} : \alpha, \beta \in \mathbb{R} \right\}$
- $\mathcal{G}_1 = \left\{ \begin{bmatrix} \alpha & 0 \\ 0 & \alpha \end{bmatrix} : \alpha \in \mathbb{R} \right\}$, $\mathcal{G}_2 = \left\{ \begin{bmatrix} 0 & \beta \\ 0 & 0 \end{bmatrix} : \beta \in \mathbb{R} \right\}$ are ideals of \mathcal{G} .

Detour: Finite-dimensional Algebras

A few definitions:

- An algebra \mathcal{A} is **nilpotent** if $\exists n: \mathcal{A}^n = \{0\}$.
- The **radical** $\text{Rad}(\mathcal{A})$ of \mathcal{A} is its maximal nilpotent ideal.
- An algebra is **semi-simple** if its radical is $\{0\}$.

Example. Let \mathcal{G} be the algebra generated by $\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$. We have

- $\mathcal{G} = \left\{ \begin{bmatrix} \alpha & \beta \\ 0 & \alpha \end{bmatrix} : \alpha, \beta \in \mathbb{R} \right\}$
- $\mathcal{G}_1 = \left\{ \begin{bmatrix} \alpha & 0 \\ 0 & \alpha \end{bmatrix} : \alpha \in \mathbb{R} \right\}$, $\mathcal{G}_2 = \left\{ \begin{bmatrix} 0 & \beta \\ 0 & 0 \end{bmatrix} : \beta \in \mathbb{R} \right\}$ are ideals of \mathcal{G} .

$\Rightarrow \text{Rad}(\mathcal{G}) = \mathcal{G}_2$ and \mathcal{G} is not semi-simple.

Equivalence of GWM^cs

We say that a GWM $A = \{\mathbf{A}^\sigma\}_{\sigma \in \Sigma}$ is a **semi-simple GWM^c** if the algebra generated by the matrices $\{\mathbf{A}^\sigma\}_{\sigma \in \Sigma}$ is semi-simple.

Equivalence of GWM^c s

We say that a GWM $A = \{\mathbf{A}^\sigma\}_{\sigma \in \Sigma}$ is a **semi-simple GWM^c** if the algebra generated by the matrices $\{\mathbf{A}^\sigma\}_{\sigma \in \Sigma}$ is semi-simple.

Theorem

*Two d -dimensional **semi-simple** GWM^c A and B compute the same function if and only if they are related by a change of basis, (i.e. $\exists \mathbf{P}$ such that $\mathbf{A}^\sigma = \mathbf{P}^{-1} \mathbf{B}^\sigma \mathbf{P}$ for all $\sigma \in \Sigma$).*

Equivalence of GWM^c s

We say that a GWM $A = \{\mathbf{A}^\sigma\}_{\sigma \in \Sigma}$ is a **semi-simple GWM^c** if the algebra generated by the matrices $\{\mathbf{A}^\sigma\}_{\sigma \in \Sigma}$ is semi-simple.

Theorem

*Two d -dimensional **semi-simple** GWM^c A and B compute the same function if and only if they are related by a change of basis, (i.e. $\exists \mathbf{P}$ such that $\mathbf{A}^\sigma = \mathbf{P}^{-1} \mathbf{B}^\sigma \mathbf{P}$ for all $\sigma \in \Sigma$).*

Theorem

Any function that can be computed by a GWM^c can be computed by a semi-simple GWM^c of the same dimension.

\hookrightarrow ... and such a semi-simple GWM^c can be computed in polynomial time.

Equivalence of GWM^cs (cont'd)

The proofs of the two previous theorems fundamentally rely on the [Wedderburn-Malcev Theorem](#):

Theorem

*Let \mathcal{A} be a finite-dimensional algebra **over a field of characteristic 0**. There exists a semi-simple subalgebra $\tilde{\mathcal{A}}$ of \mathcal{A} which is isomorphic to $\mathcal{A}/\text{Rad}(\mathcal{A})$ and such that $\mathcal{A} = \tilde{\mathcal{A}} \oplus \text{Rad}(\mathcal{A})$ (direct sum of vector spaces).*

Equivalence of GWM^cs (cont'd)

The proofs of the two previous theorems fundamentally rely on the [Wedderburn-Malcev Theorem](#):

Theorem

*Let \mathcal{A} be a finite-dimensional algebra **over a field of characteristic 0**. There exists a semi-simple subalgebra $\tilde{\mathcal{A}}$ of \mathcal{A} which is isomorphic to $\mathcal{A}/\text{Rad}(\mathcal{A})$ and such that $\mathcal{A} = \tilde{\mathcal{A}} \oplus \text{Rad}(\mathcal{A})$ (direct sum of vector spaces).*

and on the fact that the corresponding surjective homomorphism $\pi: \mathcal{A} \rightarrow \tilde{\mathcal{A}}$ is trace preserving, i.e $\text{Tr}(\mathbf{A}) = \text{Tr}(\pi(\mathbf{A}))$ for all $\mathbf{A} \in \mathcal{A}$.

Equivalence of GWM^cs (cont'd)

Example. Let \mathcal{G} be the algebra generated by $\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$. We have

- $\mathcal{G} = \left\{ \begin{bmatrix} \alpha & \beta \\ 0 & \alpha \end{bmatrix} : \alpha, \beta \in \mathbb{R} \right\}$
- $\mathcal{G}_1 = \left\{ \begin{bmatrix} \alpha & 0 \\ 0 & \alpha \end{bmatrix} : \alpha \in \mathbb{R} \right\}$, $\mathcal{G}_2 = \left\{ \begin{bmatrix} 0 & \beta \\ 0 & 0 \end{bmatrix} : \beta \in \mathbb{R} \right\}$ are ideals of \mathcal{G} .

Here we have $\text{Rad}(\mathcal{G}) = \mathcal{G}_2$ and $\mathcal{G} = \mathcal{G}_1 \oplus \mathcal{G}_2$.

Outline

- 1 Preliminaries
- 2 Graph Weighted Model over Circular Strings (GWM^c)
- 3 Equivalence of GWM^c s
- 4 Minimization of GWM^c s

GWM^c Minimization

Characterization of minimality:

Theorem

Let A be a GWM^c given by the set of matrices $\{\mathbf{A}^\sigma\}_{\sigma \in \Sigma} \subset \mathbb{R}^{d \times d}$. Then, A is minimal if and only if the linear space

$$E = \bigcap_{x \in \Sigma^d} \ker(\mathbf{A}^x) = \{\mathbf{v} \in \mathbb{F}^d : \mathbf{A}^x \mathbf{v} = \mathbf{0} \text{ for all } x \in \Sigma^d\}$$

is trivial, i.e. $E = \{\mathbf{0}\}$.

GWM^c Minimization

Characterization of minimality:

Theorem

Let A be a GWM^c given by the set of matrices $\{\mathbf{A}^\sigma\}_{\sigma \in \Sigma} \subset \mathbb{R}^{d \times d}$. Then, A is minimal if and only if the linear space

$$E = \bigcap_{x \in \Sigma^d} \ker(\mathbf{A}^x) = \{\mathbf{v} \in \mathbb{F}^d : \mathbf{A}^x \mathbf{v} = \mathbf{0} \text{ for all } x \in \Sigma^d\}$$

is trivial, i.e. $E = \{\mathbf{0}\}$.

Minimization algorithm:

1. Let $\mathbf{\Pi} \in \mathbb{R}^{d \times d}$ be the orthogonal projection onto E .
2. Let $\mathbf{U} \in \mathbb{F}^{d \times R}$ such that $\mathbf{I} - \mathbf{\Pi} = \mathbf{U}\mathbf{U}^\top$ ($R = \dim(E^\perp)$).
3. Define R -dimensional GWM^c \hat{A} with $\hat{\mathbf{A}}^\sigma = \mathbf{U}^\top \mathbf{A}^\sigma \mathbf{U}$ for each $\sigma \in \Sigma$.

$\Rightarrow \hat{A}$ is a minimal GWM^c computing f_A .

Conclusion

- We studied two fundamental problems for GWMs restricted to circular strings
- Theory of GWMs relies on non-commutative algebras rather than simple linear algebra

Conclusion

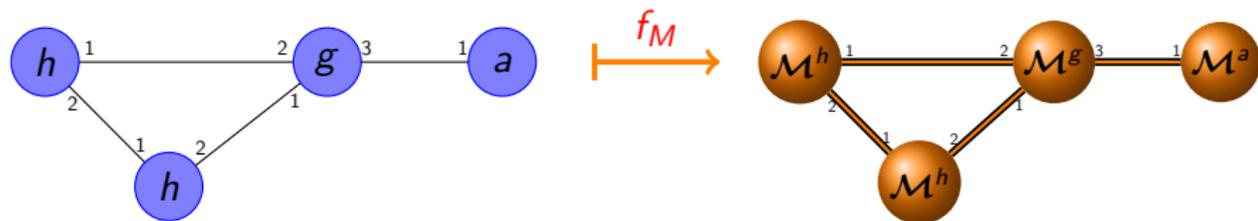
- We studied two fundamental problems for GWMs restricted to circular strings
- Theory of GWMs relies on non-commutative algebras rather than simple linear algebra

Future work:

- Extend the minimization/equivalence algorithms to GWMs defined over larger families of graphs.
 - ▶ First step: extend the notion of semi-simple GWMs.
- Learning GWMs:
 - ▶ Ongoing work on 2-dimensional words (preliminary results at LearnAut'18).

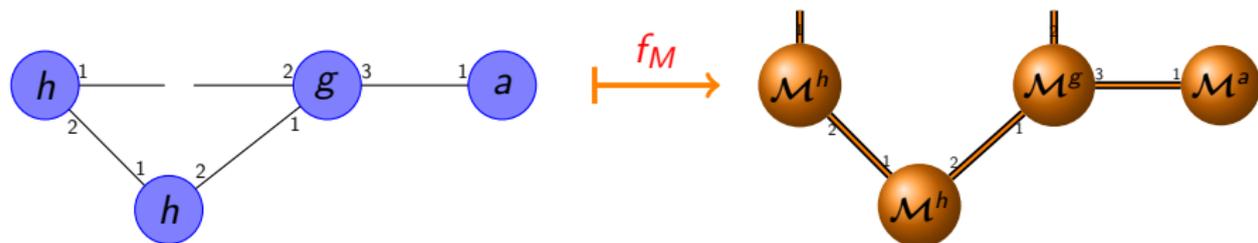
Semi-simple GWMs

GWMs maps **open graphs** to matrices:



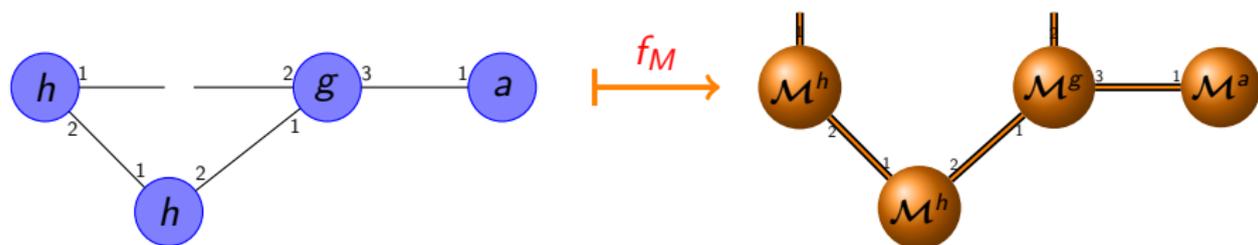
Semi-simple GWMs

GWMs maps **open graphs** to matrices:



Semi-simple GWMs

GWMs maps **open graphs** to matrices:



For any graph $G \in \mathcal{S}$ in some family of graph, for any edge e of G , $f_M(G \setminus e)$ is a matrix.

A GWM is semi-simple if the algebra generated by all such matrices is semi-simple...