Spectral learning of graphical distributions

Raphael Bailly

Spectral learning of graph distributions

Raphaël BAILLY*
LIF
Marseille
raphael.bailly@lif.univ-mrs.fr

June 8, 2012

Abstract

This work draws on previous works regarding spectral learning algorithm for structured data (see [5], [9], [2], [1], [8]). We present an extension of the *Hidden Markov Models*, called *Graphical Weighted Models (GWM)*, whose purpose is to model distributions over labeled graphs. We describe the spectral algorithm for GWM, which generalizes the previous spectral algorithms for sequences and trees. We show that this algorithm is *consistant*, and we provide statistical convergence bounds for the parameters estimate and for the learned distribution.

1 Introduction

Graphical Weighted Models (GWMs), and *Directed Graphical Weighted Models (DGWMs)*, are probabilistic models which generalizes HMMs and *Observable Operator Models* (see [6]). They are related to graphical models with discrete latent variables. The vertices labels $\boldsymbol{x}_i$ are distributed conditionnaly to their corresponding latent variable $\boldsymbol{y}_i$. One supposes that the hidden variables satisfy the *Markov property* w.r.t the graphical structure.

For any fixed graphical structure, these GWMs can model a distribution over vertices labels. Moreover, they can describe at the same time a distribution over all possible graphical structures.

We present a spectral algorithm which is a generalization of the spectral algorithm in [5]. This algorithm provides an estimate of the parameters which is *consistant*, and not prone to local extrema.

The section 2 describes some preliminary notions, section 3 introduces the *Directed Graphical Weighted Model(DGWM)*, section 4 presents the spectral algorithm, and section 5 adresses the convergence results. The section 6 introduces the *Graphical Weighted Model(GWM)* and the spectral algorithm for GWM. We conclude with the section 7.

*Lampada

2 Preliminaries

The objects we consider in this paper are labeled DAGs (*Directed Acyclic Graphs*), and labeled graphs – i.e. each vertice is labeled with a symbol x belonging to an alphabet $\mathcal{F}$. Each symbol has an incoming and an outgoing arity for the directed model, an overall arity for the undirected model. Each *port* (incoming or outgoing) has a *port number*. For instance, let us consider the alphabet $\mathcal{F} = \{f_1^{2,3}, j_{1,2}, h^1, i_1^2\}$, meaning that the symbol f has an incoming arity 2 (with port numbers 2 and 3) and an outgoing arity 1 (port number 1). Fig. 1 represents a DAG example built with $\mathcal{F}$.

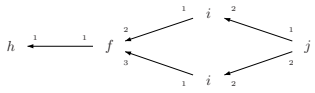

Figure 1: A *Directed Acyclic Graph (DAG) g*.

2.1 Sequences as DAGS

Sequences are particular cases of DAGs: the set of sequences built upon an alphabet $\Sigma = \{a, b, c \dots\}$ will correspond to DAGs with $\mathcal{F} = \{i^1, t_1, a_1^2, b_1^2, c_1^2 \dots\}$. The sequence *aabca* will correspond to the DAG represented Fig. 2.

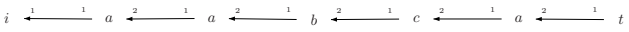

Figure 2: Sequence as a *Directed Acyclic Graph (DAG)*.

It is clear that any sequence built with Σ can be represented by a DAG built with $\mathcal{F}$, and conversely, that any DAG built with $\mathcal{F}$ corresponds to a sequence built with Σ.

2.2 Trees as DAGS

Trees are also particular cases of DAGs: the set of trees built with an alphabet $\Sigma = \{f_2, b_1, c_0 \dots\}$ corresponds to a DAG with set of symbols $\mathcal{F} = \{i^1, f_1^{2,3}, b_1^2, c_1 \dots\}$ where the parent port will have the port number 1. The tree $f(b(c), f(c, c))$ corresponds to the DAG Fig.3:

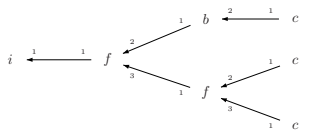

Figure 3: Tree as a *Directed Acyclic Graph (DAG)*.

Again, it is clear that there is a bijection between trees built with $\Sigma = \{f_2, b_1, c_0 \dots\}$ and DAGs built with $\mathcal{F} = \{i^1, f_1^{2,3}, b_1^2, c_1 \dots\}$.

In the rest of the paper, one will use functional notation for trees and sequences, including the symbols i and t. For instance, the sequence $aabca$ will be denoted $i(a(a(b(c(a(t))))))$, and the tree $f(b(c), f(c, c))$ will be denoted $i(f(b(c), f(c, c)))$.

3 Directed Graphical Weighted Model

In this section, we extend HMMs to DAGs. To do this, one needs a fuctionnal notation for DAGs.

3.1 Notations

One defines an *incomplete DAG (I-DAG)* as a DAG having unconnected vertices. Examples of I-DAG are represented Fig. 4.

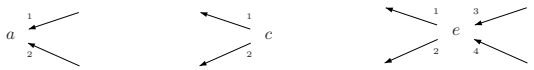

Figure 4: I-DAGs examples: $a^{1,2}$, $c_{1,2}$ and $e_{1,2}^{3,4}$.

Definition 1. *Let $g_1, \dots, g_k$ be I-DAGs. The I-DAG $g = (g_1, \dots, g_k)$ is obtained by juxtaposing the I-DAGs g_i, from the top to the bottom. The I-DAG g^k is $(g, \dots, g)$ repeated k times.*

Definition 2. *Let $g_1, \dots, g_k$ be I-DAGs. The I-DAG $g = g_1 g_2 \dots g_k$ is obtained by juxtaposing the I-DAGs g_i, from left to right. The incoming edges of g_i corresponds to outgoing edges of g_{i+1}.*

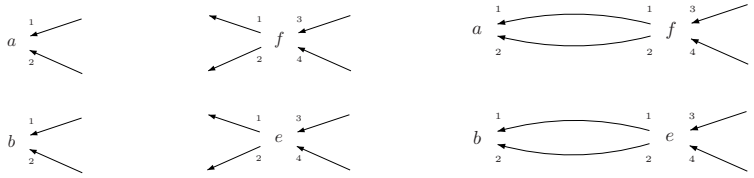

Figure 5: I-DAGs examples: $(a^{1,2}, b^{1,2})$, $(f_{1,2}^{3,4}, e_{1,2}^{3,4})$ and $(a^{1,2}, b^{1,2})(f_{1,2}^{3,4}, e_{1,2}^{3,4})$

Definition 3. *The I-DAG $\mathbf{1}$ denotes the identity element for the operation $(\cdot, \cdot)$ – i.e. $(\mathbf{1}^k, g_k^{k'}) = (g_k^{k'}, \mathbf{1}^{k'}) = g_k^{k'}$.*

Definition 4. *Let g be an I-DAG with n incoming ports and n^* outgoing ports. Let σ be a permutation of $\{1,\dots n\}$ and σ^* be a permutation of $\{1,\dots n^*\}$. One will denote $[g]^{\sigma}_{\sigma^*}$ the I-DAGs obtained by permutation of incoming ports (resp. outgoing) with σ (resp. σ^*). (see Fig. 6)*

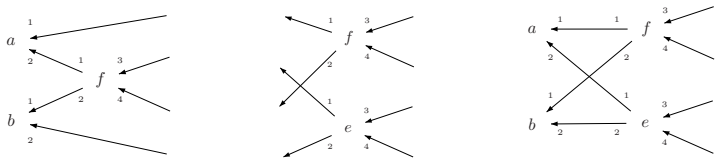

Figure 6: I-DAGs $(a^{1,2},b^{1,2})(\mathbf{1}^1,f^{3,4}_{1,2},\mathbf{1}^1)$, $[(f^{3,4}_{1,2},e^{3,4}_{1,2})]_{1324}$ and $(a^{1,2},b^{1,2})[(f^{3,4}_{1,2},e^{3,4}_{1,2})]_{1324}$

Proposition 1. *Any DAG g can be written as*

$$[(v_{1,1},\dots,v_{1,k_1})]^{\sigma_1}_{\sigma_1^*}\cdot\dots\cdot[(v_{n,1},\dots,v_{n,k_n})]^{\sigma_n}_{\sigma_n^*}$$

where v_i belongs to $\mathcal{F}\cup\{\mathbf{1}\}$.

Proof. Let V be the set of vertices of g. One defines a partition of $V=V_0\cup\dots\cup V_k$ as:

- V_0= set of vertices having only incoming edges
- V_{i+1}= set of vertices having only incoming edges, or outgoing edges towards vertices of $V_0\cup\dots\cup V_i$.

Each V_i is completed with k_i symbols $\mathbf{1}$, in order to have the same number of incoming edges for V_i and outgoing edges for V_{i+1}. One defines the I-DAG $g_i=(v_i^1,\dots,v_i^{n_i})_{v_i^j\in V_i}$, and the permutations σ_i such as the incoming ports of g_i corresponds to outgoing ports of g_{i+1}. One define

$$g=[g_0]^{\sigma_0}\dots[g_k]^{\sigma_k}$$

□

There exists obviously several way to denote a particular DAG: the DAG g Fig.1 can be written $(h^1)(f_1^{2,3})(i_1^2,i_1^2)(j_{1,2})$ or $(h^1)(f_1^{2,3})(i_1^2,\mathbf{1}^1)(\mathbf{1}^1,i_1^2)(j_{1,2})$. When g is a tree or a sequence, this notation corresponds to the usual functional notation.

Definition 5. *A* Directed Graphical Weighted Model (DGWM) *is given by:*

- *a rank d, an d-dimensionnal vector space E over $\mathbb{R}$ called* state space*, or* residual space*.*

- *for each symbol f belonging to the alphabet, with incoming arity n and outgoing arity n^*, a real matrix $\boldsymbol{f}$ of dimensions (d^{n^*}, d^n).*

Definition 6. *Let d be an integer, and let M be a (d^{n^*}, d^n) matrix, let σ be a permutation of $\{1, \dots n\}$, and let σ^* be a permutation of $\{1, \dots n^*\}$. One will denote by $[M]^{\sigma}_{\sigma^*}$ the matrix satisfying, for any column vectors $v_1, \dots, v_n$ and row vectors $v_1^*, \dots, v_{n^*}^*$:*

$$(v_1^* \otimes \cdots \otimes v_{n^*}^*) \cdot M \cdot (v_1 \otimes \cdots \otimes v_n) = (v_{\sigma^*(1)}^* \otimes \cdots \otimes v_{\sigma^*(n^*)}^*) \cdot [M]^{\sigma}_{\sigma^*} \cdot (v_{\sigma(1)} \otimes \cdots \otimes v_{\sigma(n)})$$

This matrix is unique, and it can be deduced from M by a permutation of rows and columns. More precisely:

Lemma 1. *Let σ be a permutation of $\{1, \dots n\}$, and σ^* be a permutation of $\{1, \dots n^*\}$. There exists a column permutation $\bar{\sigma}$ and a row permutation $\bar{\sigma}^*$ such that, for any (d^{n^*}, d^n) matrix M, one has $[M]^{\sigma}_{\sigma^*} = \bar{\sigma}(\bar{\sigma^*}(M))$.*

Definition 7. *Let g be a DAG. Let A be a DGWM of rank d over the symbols of g. Let $\boldsymbol{I}$ be the identity matrix of rank d. One defines the mapping r_A applied to $g = [(v_{1,1}, \dots, v_{1,k_1})]^{\sigma_1}_{\sigma_1^*} \cdot \cdots \cdot [(v_{n,1}, \dots, v_{n,k_n})]^{\sigma_n}_{\sigma_n^*}$ by:*

$$r_A(g) = [(\boldsymbol{v_{1,1}} \otimes \cdots \otimes \boldsymbol{v_{1,k_1}})]^{\sigma_1}_{\sigma_1^*} \cdot \cdots \cdot [(\boldsymbol{v_{n,1}} \otimes \cdots \otimes \boldsymbol{v_{n,k_n}})]^{\sigma_n}_{\sigma_n^*}$$

Proposition 2. *The value $r_A(g)$ does not depend on the functional notation for g.*

Proof. (Sketch) By induction. One uses the property $(A \cdot B) \otimes (C \cdot D) = (A \otimes C) \cdot (B \otimes D)$, thus $(\boldsymbol{I} \otimes \boldsymbol{x}_2^1) \cdot (\boldsymbol{y}_2^1 \otimes \boldsymbol{I}) = (\boldsymbol{y}_2^1 \otimes \boldsymbol{x}_2^1)$. □

4 Spectral Algorithm

Definition 8. *One will call n-order* prefix *(resp.* suffix*) an I-DAG, having n incomplete incoming ports (resp. n incomplete outgoing ports). A prefix (resp. suffix) is by default a 1-order prefix (resp. suffix). A missing vertice is denoted $\star$.*

Example 1. *The following objects are prefixes: $i(f(b(c), f(c, \star)))$, $i(f(b(c), f(\star, c)))$, $i(f(\star, f(c, c)))$. Tree prefixes are also called* contexts*.*

Example 2. *The following objects are suffixes: $\star(f(b(c), f(c, c)))$, $\star(f(c, c))$.*

Example 3. *Prefixes and suffixes for sequences are 1-order. Tree suffixes are 1-order suffixes.*

Definition 9. *Let p be an n-order prefix, and $s_1, \dots, s_n$ be 1-order suffixes. One will denote by $p[s_1, \dots, s_n]$ The DAG built by pluging $\star$ symbols of p to $\star$ symbols of each s_i.*

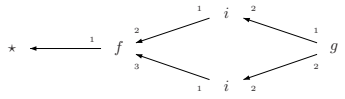

Figure 7: A DAG prefix

Example 4. *Let* $p = i(f(b(c), f(\star, c)))$ *and* $s = \star(f(c, c))$*, one has* $p[s] = i(f(b(c), f(f(c, c), c)))$

Definition 10. *The* rank *of a mapping* r *computed by a DGWM is the minimal rank of a DGWM computing* r*.*

Definition 11. *An* n*-order* generalized prefix *(resp.* n*-order* generalized suffix*) is a set of* n*-order prefixes (resp. set of* n*-order suffixes) having undefined vertices. A generalized prefix (resp. generalized suffix) is a* 1*-order generalized prefix (resp.* 1*-order generalized suffix). An undefined vertice is denoted* $\cdot$*. The set of* n*-order* generalized prefix *(resp.* n*-order* generalized suffix*) is the set of any* n*-order prefix (resp.* n*-order suffix) obtained by replacing the* $\cdot$ *corresponding to an incoming port by any* 1*-order suffix, and the* $\cdot$ *corresponding to an outgoing port by any* 1*-order prefix.*

Example 5. *The generalized sequence prefix* $\cdot(a(a(\star)))$ *is the set* $\{i(a(a(\star))); i(b(a(a(\star)))); i(a(a(b(a(a(\star)))))); \ldots$
It is the set of prefixes ending with $a(a(\star))$*.*

Definition 12. *Let* U *be a set of generalized prefixes, let* V *be a set of generalized suffixes. The* Hankel matrix *for the set* U *and* V *of a real-value mapping over DAGs* r *is the matrix* $X^r_{U,V}$ *defined by* $(X^r_{U,V})_{u,v} = \sum_{p \in U, s \in V} r(p[s])$*.*

Example 6. *Let* $t = i(b(a(a(c)), b(c, d))))$ *be a tree, let* $U = \{i(\star), \cdot(a(\star)), \cdot(b(\star, \cdot)), \cdot(b(\cdot, \star))\}$ *and let* $V = \{\star(a(\cdot)), \star(b(\cdot, \cdot)), \star(c) \star (d)\}$*, and let* r *be the counting mapping of the set* $S = \{t\}$ *(i.e* $r(t) = Card(t' \in S | t' = t)$*). The Hankel matrix* $X^r_{U,V}$ *is:*

$$\begin{array}{c} \\ i(\star) \\ \cdot(a(\star)) \\ \cdot(b(\star, \cdot)) \\ \cdot(b(\cdot, \star)) \end{array} \begin{array}{c} \begin{array}{cccc} \star(a(\cdot)) & \star(b(\cdot, \cdot)) & \star(c) & \star(d) \end{array} \\ \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix} \end{array}$$

Let r be a fixed real-value mapping over DAGs built upon a set of symbols $\mathcal{F} = \{\ldots x^k_{k^*} \ldots\}$. Let p be a prefix, let s be a suffix, let P be the set of all generalized prefixes, and let S be the set of all generalized suffixes. One can define the following objects:

- $\bar{p} : S \mapsto \mathbb{R}$ such that $\bar{p}(s) = r(p[s])$
- $\bar{s} : P \mapsto \mathbb{R}$ such that $\bar{s}(p) = r(p[s])$

One can define the two following vector spaces:

- E^*_r = vector space spanned by $\{\bar{p}\}_{p \in P}$

- E_r = vector space spanned by $\{\bar{s}\}_{s\in S}$

For each symbol $x^k_{k^*} \in \mathcal{F}$ one can define a mapping $\dot{x}^k_{k^*} : P^{k^*} \times S^k \mapsto \mathbb{R}$ defined by $\dot{x}^k_{k^*}((p_1,\dots,p_{k^*}),(s_1,\dots,s_k)) = r((p_1,\dots,p_{k^*})(x^k_{k^*})(s_1,\dots,s_k))$, which extends naturally to a $(k+k^*)$-linear mapping $\dot{x}^k_{k^*} : (E^*_r)^{\otimes k^*} \times (E_r)^{\otimes k} \mapsto \mathbb{R}$ defined by $\dot{x}^k_{k^*}(\sum_i(\bar{p}^i_1,\dots,\bar{p}^i_{k^*}),\sum_j(\bar{s}^j_1,\dots,\bar{s}^j_k)) = \sum_{i,j} r((p^i_1,\dots,p^i_{k^*})(x^k_{k^*})(s^j_1,\dots,s^j_k))$, where $E^{\otimes k}_r$ is the tensor product of E_r repeated k times.

4.1 Model properties

Let A be a DGWM over $\mathcal{F} = \{\dots x^k_{k^*} \dots\}$, with a state space $E = \mathbb{R}^d$. Each $\boldsymbol{x}^k_{k^*}$ is a matrix which can be seen as a k-linear mapping $E^{\otimes k} \mapsto E^{\otimes k^*}$. The dimension of E^*_r is the dimension of E_r, and it is also the rank of the Hankel matrix X^r.

The residual spaces E_r and E^*_r can be seen as subspace of E, thus $\text{rank}(X^r) \leq d$.

Definition 13. *A DGWM A computing a mapping r_A is called* simple *if the state space dimension is equal to the rank of the Hankel matrix X^{r_A}.*

For the rest of the paper, we will make the assumption that the mapping we want to estimate is computed by a *simple* DGWM. This implies that E, E_r and E^*_r have rank d, and that $\boldsymbol{x}^k_{k^*}$ is entirely defined by its restriction to $E^{\otimes k}_r \times (E^*_r)^{\otimes k^*}$.

Let X be the Hankel matrix of r, and let $X_{x^k_{k^*}}$ be the matrix defined by:

$$(X_{x^k_{k^*}})_{(p_1,\dots,p_{k^*}),(s_1,\dots,s_k)} = r((p_1,\dots,p_{k^*})(x^k_{k^*})(s_1,\dots,s_k))$$

Let v_S be the coordinates of $v \in E_r$ in the basis $S = \{\bar{s_1}\dots\}$, and v_{1_P} be the coordinates of $v \in E^*_r$ in the basis $\mathbf{1}_P = \{\mathbf{1}_{p_1}\dots\}$. X is a transformation matrix from S to $\mathbf{1}_P$. Let X^+ be a pseudo-inverse of X. The matrix $X_{x^k_{k^*}}$ applied to $(\bar{s}_1)_S,\dots,(\bar{s}_k)_S$ represents the vector $\overline{x^k_{k^*}(s_1,\dots,s_k)}_{1_P}$ in the basis $\mathbf{1}_P$. From this, one can say that the matrix $X^+X_{x^k_{k^*}}$ is the matrix of $\dot{x}^k_{k^*}$ in the basis S.

Proposition 3. *Let r be a real-value mapping over DAGs built from $\mathcal{F} = \{\dots x^k_{k^*} \dots\}$. One defines the following operators:*

$$\dot{\boldsymbol{x}}^k_{k^*} = X^+ X_{x^k_{k^*}}$$

One then has:

$$r(g) = r([(v_1,\dots,v_{k_1})]^{\sigma_1}_{\sigma^*_1} \cdot \dots \cdot [(v_1,\dots,v_{k_n})]^{\sigma_n}_{\sigma^*_n})$$

$$= [(\dot{\boldsymbol{v}}_{\mathbf{1}} \otimes \dots \otimes \dot{\boldsymbol{v}}_{\boldsymbol{k_1}}]^{\sigma_1}_{\sigma^*_1} \cdot \dots \cdot [(\dot{\boldsymbol{v}}_{\mathbf{1}} \otimes \dots \otimes \dot{\boldsymbol{v}}_{\boldsymbol{k_n}})]^{\sigma_n}_{\sigma^*_n}$$

The principle of the spectral algorithm is the following: one supposes that r is computed by a rank-d simple DGWM, and one performs the *singular value decomposition (SVD)* of $X = W^* \Sigma W^T$, with Σ a diagonal matrix of size d. One then has:

Proposition 4. *Let r a mapping computed by a simple DGWM over $\mathcal{F} = \{\dots x^k_{k^*} \dots\}$. Let $X^r = W^* \Sigma W^T$. Then the DGWM defined by:*

$$\boldsymbol{x}^k_{k^*} = (\Sigma^{-1} W^*)^{\otimes k^*} X_{x^k_{k^*}} (W^T)^{\otimes k}$$

computes r.

Proof. The proof is very similar to the proofs of corresponding results in other works, e.g. [5]. □

Data: A sample $S = \{s_i, 1 \leq i \leq |S|\}$ i.i.d. according to a distribution p, a rank d, an alphabet $\mathcal{F}$, a set of prefixes $U = \{u_1, \dots\}$, a set of suffixes $V = \{v_1, \dots\}$.
Result: A Graphical Weighted Model A computing an estimate of p
begin
  $X_{i,j} \leftarrow p_S(u_i v_j)$
  for *each* $x^k_{k^*} \in F$ **do**
    $X_{x, i_1 \dots i_{k^*}, j_1 \dots j_k} \leftarrow p_S((u_1, \dots, u_{k^*}) x^k_{k^*}(v_1, \dots, v_k))$
  $X = W^* \Sigma W^T$
  for *each* $x^k_{k^*} \in F$ **do**
    $\boldsymbol{x}^k_{k^*} \leftarrow (\Sigma^{-1} W^{*T})^{\otimes k^*} X_{x^k_{k^*}} (W)^{\otimes k}$
  return $A = \{\boldsymbol{x}^k_{k^*}\}_{\boldsymbol{x}^k_{k^*} \in F}$
end

Algorithm 1: Spectral Algorithm for DGWMs

5 Concistency

Let p be a probability distribution over DAGs, computed by a simple DGWM of rank d. Let X be the Hankel matrix of p for $U = \{u_1, \dots\}$ and $V\{v_1, \dots\}$. Let x be a prefix (resp . y be a suffix). Let $m_V(x) = Card(\{v \in V | x \subset v\}$ (resp. $m_U(x) = Card(\{u \in U | y \subset u\}$). Let $m = max_{c \in V}(|c|).max_{g \in G}(m_V(g) m_U(g)$. Let S be a sample of size N i.i.d. with respect to p. Let X_S be the Hankel matrix of the empirical distribution p_S. Let $\lambda_1 \geq \dots \geq \lambda_d \dots$ be the singular values of X, the lower beeing λ_d.

5.1 Parameter estimate

Proposition 5. *Let $\{\boldsymbol{x}^k_{k^*}\}$ be a DGWM computing p, provided by the spectral algorithm on target values. Let $\{\boldsymbol{x}_{\boldsymbol{S}}{}^k_{k^*}\}$ be a DGWM estimating p, provided by*

the spectral algorithm on empirical values. Let $\|\|\|_F$ *be the Frobenius norm on matrices. Let* δ *be a confidence parameter. One then has, with probability* $1-\delta$*:*

$$\|\boldsymbol{x}^k_{k^*} - \boldsymbol{x}_{\boldsymbol{S}}{}^k_{k^*}\|_F = O\left(\sqrt{\frac{m^3 d^{2(k+k^*)+1} k \log(\frac{1}{\delta})}{N\lambda_d^{2k+2}}}\right)$$

Proof. (Sketch.) One first obtains a concentration inequality for the Hankel matrix, as in [5]. One then uses matrix perturbation results for singular values and singular vectors, as in [10]. □

5.2 Simple convergence

Proposition 6. *Let* $\{\boldsymbol{x}^k_{k^*}\}$ *be a DGWM computing* p*, provided by the spectral algorithm on target values. Let* r_S *be the mapping computed by the DGWM provided by the spectral algorithm on empirical values. Let* g *be a DAG, and* k *the maximal arity of a symbol occuring in* g*. Let* $\|\|\|_F$ *be the Frobenius norm on matrices. Let* $M = \max_{x\in F} \|\boldsymbol{x}\|_F$*. Let* δ *be a confidence parameter. One then has, with probability* $1-\delta$*:*

$$|p(t) - r_S(t)| = O\left(M^{|t|}\sqrt{\frac{m^3 d^{2k+1} k \log(\frac{1}{\delta})}{N\lambda_d^{2k+2}}}\right)$$

Proof. (Sketch.) One uses the linearity of matrix product and tensor product, with the properties $\|M\cdot N\|_F \leq \|M\|_F\|N\|_F$ and $\|M\otimes N\|_F = \|M\|_F\|N\|_F$. □

Example 7. *Sample:* $\{(a,a)g;(a,b)g;(a,b)g;(b,a)g\}$*. Prefixes:*$\{a,b\}$*. Suffixes:*$\{(\star,a)g,(\star,b)g\}$*.* $X = W^*\Sigma W^T$*.*

$$X = \begin{pmatrix} 1/4 & 1/2 \\ 1/4 & 0 \end{pmatrix}, X_a = \begin{pmatrix} 1/4 & 1/2 \end{pmatrix}, X_b = \begin{pmatrix} 1/4 & 0 \end{pmatrix}, X_g^T = \begin{pmatrix} 1/4 & 1/2 & 1/4 & 0 \end{pmatrix}$$

$$W^* = \begin{pmatrix} -0.9732 & -0.2298 \\ -0.2298 & 0.9732 \end{pmatrix}, \Sigma = \begin{pmatrix} 0.5721 & 0 \\ 0 & 0.2185 \end{pmatrix}, W^T = \begin{pmatrix} -0.5257 & -0.8506 \\ 0.8506 & -0.5257 \end{pmatrix}$$

DGWA provided by the spectral algorithm:

$$\boldsymbol{a} = Xa\cdot W = \begin{pmatrix} -0.5568 & -0.0502 \end{pmatrix}, \boldsymbol{b} = Xb\cdot W = \begin{pmatrix} -0.1314 & 0.2127 \end{pmatrix}$$

$$\boldsymbol{g}^T = \left(((\Sigma^{-1}W^{*T})\otimes(\Sigma^{-1}W^{*T}))X_g\right)^T = \begin{pmatrix} 1.2361 & -3.2361 & -1.2361 & -3.2361 \end{pmatrix},$$

One can check that $(\boldsymbol{a}\otimes\boldsymbol{b})\cdot\boldsymbol{g} = 1/2$, $(\boldsymbol{a}\otimes\boldsymbol{a})\cdot\boldsymbol{g} = 1/4$, $(\boldsymbol{b}\otimes\boldsymbol{a})\cdot\boldsymbol{g} = 1/4$, $(\boldsymbol{b}\otimes\boldsymbol{b})\cdot\boldsymbol{g} = 0$.

6 Undirected Graphical Weighted Model

We define here a GWM for undirected graphs. The idea is to consider a model which computes the same result, for any acyclic direction of a given graph.

Definition 14. *A* Graphical Weighted Model (GWM) *is given by:*

- *a rank d, and a d-dimensional vector space E over $\mathbb{C}$, called* state space, *or* residual space.
- *for each n-arity symbol $f_{1,\dots,n}$, a complex vector $\boldsymbol{f} = \boldsymbol{f}_{1,\dots,n}$ of dimension d^n.*
- *for each symbol $f^{i_1\dots i_k}_{i_{k+1}\dots i_n}$, a (d^k, d^{n-k})-complex matrix $\boldsymbol{f}^{i_1\dots i_k}_{i_{k+1}\dots i_n}$ which can be deduced from $\boldsymbol{f}$, and satisfying, for any $v_1 \dots v_n \in E$:*

$$(v_{i_1}^T \otimes \cdots \otimes v_{i_k}^T)\boldsymbol{f}^{i_1\dots i_k}_{i_{k+1}\dots i_n}(v_{i_{k+1}} \otimes \cdots \otimes v_n) = (v_1^T \otimes \cdots \otimes v_n^T)\boldsymbol{f}$$

Example 8.

$$\boldsymbol{f}^T = \boldsymbol{f}^T_{1,2} = \begin{pmatrix} f_{11} & f_{12} & f_{21} & f_{22} \end{pmatrix}, \boldsymbol{f}^{2,1} = \begin{pmatrix} f_{11} & f_{21} & f_{12} & f_{22} \end{pmatrix}, \boldsymbol{f}^1_2 = \begin{pmatrix} f_{11} & f_{21} \\ f_{12} & f_{22} \end{pmatrix}$$

One can check that

$$(v_1^T \otimes v_2^T)\boldsymbol{f} = v_2^T \boldsymbol{f}^1_2 v_1 = \boldsymbol{f}^{2,1}(v_2 \otimes v_1)$$

Proposition 7. *Let g be a graph, and A be a GWM. Let g_1 and g_2 be two different DAGs corresponding to g. Then $r_A(g_1) = r_A(g_2)$.*

Proof. The proof is simple, though quite technical, and can be found in the longer version of this paper. It is mainly based on the property $(\boldsymbol{x}^T \otimes \boldsymbol{I}) \cdot (\boldsymbol{I} \otimes \boldsymbol{y}) = (\boldsymbol{y}^T \otimes \boldsymbol{I}) \cdot (\boldsymbol{I} \otimes \boldsymbol{x})$. □

6.1 Spectral algorithm for GWM

Data: A sample $S = \{s_i, 1 \leq i \leq |S|\}$ i.i.d. according to a distribution p, a rank d, an alphabet F, a set of prefixes $U = \{u_1, \dots\}$.
Result: A Weighted Model A computing an estimate of p
begin
  $X_{i,j} \leftarrow p_S(u_i u_j)$
  for *each* $x^k_{k^*} \in F$ **do**
    $X_{x,i_1\dots i_{k^*},j_1\dots j_k} \leftarrow p_S((u_1, \dots, u_{k^*})x^k_{k^*}(u_1, \dots, u_k))$
  $X = U\Sigma U^T$
  for *each* $x^k_{k^*} \in F$ **do**
    $\boldsymbol{x}^k_{k^*} \leftarrow (\Sigma^{-1/2}U^T)^{\otimes k^*} X_{x^k_{k^*}} (U\Sigma^{-1/2})^{\otimes k}$
  return $A = \{\boldsymbol{x}^k_{k^*}\}_{\boldsymbol{x}^k_{k^*} \in F}$
end

Algorithm 2: Spectral Algorithm for GWMs

Using the same prefix and suffix sets U implies that X is symetric, and thus X can be written $X = U\Sigma U^T$, with $U^{-1} = U^T$. If one of the eigenvalues of X is negative, the provided GWM has complex parameters – though the computed value is real. Thus, the model provided is always a GWM.

Example 9. *Prefixes:*$\{a; b; (\star, a)g_{1,2}; \star, b)g_{1,2}\}$. *Suffixes:*$\{a; b; (\star, a)g_{1,2}; \star, b)g_{1,2}\}$. *Sample:* $\{(a, a)g_{1,2}; (a, b)g_{1,2}; (a, b)g_{1,2}; (b, a)g_{1,2}\}$. $X = V\Sigma V^T$.

GWA provided by the spectral algorithm:

$$\boldsymbol{a} = Xa \cdot (V\Sigma^{-1/2}) = \begin{pmatrix} 0.5205i & -0.0759i & -0.0759 & -0.5205 \end{pmatrix}$$

$$\boldsymbol{b} = Xb \cdot (V\Sigma^{-1/2}) = \begin{pmatrix} 0.1229i & 0.3217i & 0.3217 & -0.1229 \end{pmatrix}$$

$$\boldsymbol{g}_{1,2}^T = \boldsymbol{g}^{1,2} = X_{g^{1,2}} \cdot \Big((V\Sigma^{-1/2}) \otimes (V\Sigma^{-1/2})\Big) =$$

$$\begin{pmatrix} -0.3536 & -0.5721 & -0.5721i & 0.3536i & -0.2186 & 0.3536 & 0.3536i & 0.2185i \\ -0.2185i & 0.3536i & -0.3536 & -0.2185 & 0.3536i & 0.5720i & -0.5720 & 0.3536 \end{pmatrix}$$

One can check that

$$(\boldsymbol{a}\otimes\boldsymbol{a})\cdot\boldsymbol{g}_{1,2} = 1/4 \;\;,\;\; (\boldsymbol{a}\otimes\boldsymbol{b})\cdot\boldsymbol{g}_{1,2} = 1/2 \;\;,\;\; (\boldsymbol{b}\otimes\boldsymbol{a})\cdot\boldsymbol{g}_{1,2} = 1/4 \;\;,\;\; (\boldsymbol{b}\otimes\boldsymbol{b})\cdot\boldsymbol{g}_{1,2} = 0$$

One can also check that

$$\boldsymbol{a}\cdot\boldsymbol{g}_2^1\cdot\boldsymbol{a}^T = 1/4 \;\;,\;\; \boldsymbol{a}\cdot\boldsymbol{g}_2^1\cdot\boldsymbol{b}^T = 1/4 \;\;,\;\; \boldsymbol{b}\cdot\boldsymbol{g}_2^1\cdot\boldsymbol{a}^T = 1/2 \;\;,\;\; \boldsymbol{b}\cdot\boldsymbol{g}_2^1\cdot\boldsymbol{b}^T = 0$$

7 Conclusion

In several previous works, it has been shown that the spectral methods can be efficient in practical applications, like in reinforcement learning ([4]), or in natural language processing ([7]) – both in accuracy and computational time.

The spectral algorithm presented in this paper, as it is fast – the cost of a single thin SVD – and not prone to local extrema issues, should perform well in problems where graphical models are generally used in combination with EM-like learning methods for parameter estimate. It can also be used in a density estimation task for distributions on graphs.

One should also be able to extend this spectral algorithm to graphs with continuous observations, as in [9], or consider online versions of the algorithm, like in [3].

References

[1] Animashree Anandkumar, Kamalika Chaudhuri, Daniel Hsu, Sham M. Kakade, Le Song, and Tong Zhang. Spectral methods for learning multivariate latent tree structure. In John Shawe-Taylor, Richard S. Zemel, Peter L. Bartlett, Fernando C. N. Pereira, and Kilian Q. Weinberger, editors, *NIPS*, pages 2025–2033, 2011.

[2] Borja Balle, Ariadna Quattoni, and Xavier Carreras. A spectral learning algorithm for finite state transducers. In Dimitrios Gunopulos, Thomas Hofmann, Donato Malerba, and Michalis Vazirgiannis, editors, *ECML/PKDD (1)*, volume 6911 of *Lecture Notes in Computer Science*, pages 156–171. Springer, 2011.

[3] Byron Boots and Geoffrey J. Gordon. An online spectral learning algorithm for partially observable nonlinear dynamical systems. In Wolfram Burgard and Dan Roth, editors, *AAAI*. AAAI Press, 2011.

[4] Byron Boots, Sajid M. Siddiqi, and Geoffrey J. Gordon. Closing the learning-planning loop with predictive state representations. *I. J. Robotic Res.*, 30(7):954–966, 2011.

[5] D. Hsu, S.M. Kakade, and T. Zhang. A spectral algorithm for learning hidden markov models - long version. Technical report, Arxiv archive, 2009. `http://arxiv.org/abs/0811.4413`.

[6] Herbert Jaeger, Mingjie Zhao, and Andreas Kolling. Efficient estimation of ooms. In *NIPS*, 2005.

[7] Franco M. Luque, Ariadna Quattoni, Borja Balle, and Xavier Carreras. Spectral learning for non-deterministic dependency parsing. In Walter Daelemans, Mirella Lapata, and Lluís Màrquez, editors, *EACL*, pages 409–419. The Association for Computer Linguistics, 2012.

[8] Ankur P. Parikh, Le Song, and Eric P. Xing. A spectral algorithm for latent tree graphical models. In Lise Getoor and Tobias Scheffer, editors, *ICML*, pages 1065–1072. Omnipress, 2011.

[9] Le Song, Byron Boots, Sajid M. Siddiqi, Geoffrey J. Gordon, and Alexander J. Smola. Hilbert space embeddings of hidden markov models. In Johannes Fürnkranz and Thorsten Joachims, editors, *ICML*, pages 991–998. Omnipress, 2010.

[10] L. Zwald and G. Blanchard. On the convergence of eigenspaces in kernel principal component analysis. In *Proceedings of NIPS'05*, 2006.

8 Appendix- Undirected Graphical Weighted Model

8.1 Proof of Proposition 6

Let g be a DAG. We use notations of Proposition 1. Let us remark that one can split any set of the partition $V = V_0 \cup \cdots \cup V_k$, for instance $V_k = V_k' \cup V_k''$. By sorting correctly the incoming and outgoing ports, one has:

$$(\bigotimes_{v_h \in V_k} \boldsymbol{v}_h) = ((\bigotimes_{v_i' \in V_k'} \boldsymbol{v'}_i) \otimes \boldsymbol{I}^{\otimes n'})(\boldsymbol{I}^{\otimes n''} \otimes (\bigotimes_{v_j'' \in V_k''} \boldsymbol{v''}_j))$$

This comes directly from the property $(A \otimes B)(C \otimes D) = (AC) \otimes (BD)$:

$$((\bigotimes_{v_i' \in V_k'} \boldsymbol{v'}_i) \otimes \boldsymbol{I}^{\otimes n'})(\boldsymbol{I}^{\otimes n''} \otimes (\bigotimes_{v_j'' \in V_k''} \boldsymbol{v''}_j)) = ((\bigotimes_{v_i' \in V_k'} \boldsymbol{v'}_i)\boldsymbol{I}^{\otimes n''}) \otimes (\boldsymbol{I}^{\otimes n'}(\bigotimes_{v_j'' \in V_k''} \boldsymbol{v''}_j)) = (\bigotimes_{v_h \in V_k} \boldsymbol{v}_h)$$

Consequently, by splitting all classes into singles, one can choose any total order of the vertices $v_0 < v_1 < \cdots < v_k$ consistant with the partial order induced by the orientation, and computing the value with that order (meaning by that $V_i = \{v_i\}$) does not depend on the chosen order.

We will first expose some simple results about graphs:

Lemma 2. *Let G be a graph, and D an acyclic direction of the edges. From D, one can deduce a partial order of the vertices V which can be completed into a total order. Conversly, any total order on the set of vertices leads to a direction D.*

Proof. Let us take the convention that an edge from v_1 to v_2 means $v_1 > v_2$. Acyclicity implies that it is an order relation. □

Lemma 3. *Let O and O' be two total orders on a finite set. One can go from O to O' with a combination of transposition of contiguous elements.*

Lemma 4. *A transposition of contiguous elements in the total order of the set of contiguous vertices corresponds to flipping the direction of the edge between those two vertices (thus preserving acyclicity).*

From the former lemmas, one can check that it is sufficient to prove the claim in the case of flipping the direction of an edge while preserving acyclicity.

Combinig this with the former remark that one can choose a total order to perform the computing, the assertion boils down to showing the following lemma:

Lemma 5. *Let x be a symbol, k and l are two sets of indices, i is supposed to be the last indice of x's ports. Let y be a symbol, m and n are two sets of indices, j is supposed to be the last indice of y's ports. The rank of the model is d. $|n|$ is the size of the set of indices n. One has the following equality:*

$$(\boldsymbol{x}_k^{li} \otimes \boldsymbol{I}^{\otimes |m|})(\boldsymbol{I}^{\otimes |l|} \otimes \boldsymbol{y}_{jm}^{n}) = (\boldsymbol{I}^{\otimes |k|} \otimes \boldsymbol{y}_{m}^{jn})(\boldsymbol{x}_{ki}^{l} \otimes \boldsymbol{I}^{\otimes |n|})$$

corresponding to the equality of the valuations of the two I-DAGs:

Figure 8: I-DAG 1 and I-DAG 2.

Proof. The matrices $\boldsymbol{x}^l_{ki}$ and $\boldsymbol{x}^{li}_k$ can be written as:

$$\boldsymbol{x}^l_{ki} = \begin{pmatrix} X_{kl} \end{pmatrix}, \boldsymbol{x}^{li}_k = \begin{pmatrix} X^T_{kl} \end{pmatrix}$$

whith

$$X_{kl} = \begin{pmatrix} x_{kl1} \\ \vdots \\ x_{kld} \end{pmatrix}$$

The same idea holds for $\boldsymbol{y}^n_{jm}$ and $\boldsymbol{y}^{jn}_m$, whith $Y_j = (y_{mnj})$:

$$\boldsymbol{y}^n_{jm} = \begin{pmatrix} Y_1 \\ \vdots \\ Y_d \end{pmatrix}, \boldsymbol{y}^{jn}_m = \begin{pmatrix} Y_1 & \dots & Y_d \end{pmatrix}$$

One then has the following equalities:

$$\boldsymbol{x}^l_{ki} \otimes \boldsymbol{I}^{\otimes|n|} = \begin{pmatrix} X_{kl} \otimes \boldsymbol{I}^{\otimes|n|} \end{pmatrix}, \boldsymbol{x}^{li}_k \otimes \boldsymbol{I}^{\otimes|m|} = \begin{pmatrix} X^T_{kl} \otimes \boldsymbol{I}^{\otimes|m|} \end{pmatrix}$$

and

$$\boldsymbol{I}^{\otimes|l|} \otimes \boldsymbol{y}^n_{jm} = \begin{pmatrix} Y_1 & 0 & 0 \\ \vdots & 0 & 0 \\ Y_d & 0 & 0 \\ 0 & Y_1 & 0 \\ 0 & \vdots & 0 \\ 0 & Y_d & 0 \\ 0 & 0 & Y_1 \\ 0 & 0 & \vdots \\ 0 & 0 & Y_d \end{pmatrix}, \boldsymbol{I}^{\otimes|m|} \otimes \boldsymbol{y}^{jn}_m = \begin{pmatrix} Y_1 & \dots & Y_d & 0 & \dots & 0 & 0 & \dots & 0 \\ 0 & \dots & 0 & Y_1 & \dots & Y_d & 0 & \dots & 0 \\ 0 & \dots & 0 & 0 & \dots & 0 & Y_1 & \dots & Y_d \end{pmatrix}$$

This boils down to show the following equality for any i, j:

$$(X^T_{kl} \otimes \boldsymbol{I}^{\otimes|m|}) \cdot \begin{pmatrix} Y_1 \\ \vdots \\ Y_d \end{pmatrix} = \begin{pmatrix} Y_1 & \dots & Y_d \end{pmatrix} \cdot X_{kl} \otimes \boldsymbol{I}^{\otimes|n|}$$

which are both equal to

$$\sum_i x_{kli} Y_i$$

□