An Automata Theory Perspective on Spectral Learning

Hankel Matrix Factorizations

Ariadna Quattoni
with R. Bailly, B. Balle, X. Carreras, F. Luque and A. Recasens

Universitat Politècnica de Catalunya

NIPS Workshop on Spectral Learning, 2012

Outline

- Weighted Automata and Functions Over Strings
- A Spectral Method from Hankel Factorizations
- Survey on Recent Applications to Learning Problems
- Spectral Learning of Finite State Transducers

Precedents

- Subspace methods for identification of linear dynamical systems
  [Overschee–Moor '94]
- Results on identifiability and learning of HMM and phylogenetic trees
  [Chang '96, Mossel–Roch '06]
- Query learning algorithms for DFA and Multiplicity Automata
  [Angluin '87, Bergadano–Varrichio '94]

Notation

- Finite alphabet $\Sigma = \{\sigma_1, \sigma_2, \dots, \sigma_r\}$
- Free monoid $\Sigma^\star = \{\epsilon, a, b, aa, ab, ba, bb, aaa, \dots\}$
- Functions over strings $f : \Sigma^\star \to \mathbb{R}$
- Examples:

$$f(x) = \mathbb{P}[x] \qquad \text{(probability of a string)}$$

$$f(x) = \mathbb{P}[x\Sigma^\star] \qquad \text{(probability of a prefix)}$$

$$f(x) = \mathbb{I}[x \in L] \qquad \text{(characteristic function of language L)}$$

$$f(x) = |x|_a \qquad \text{(number of } a\text{'s in } x\text{)}$$

$$f(x) = \mathbb{E}[|w|_x] \qquad \text{(expected number of substrings equal to } x\text{)}$$

Weighted Automata

- Class of WA parametrized by alphabet Σ and number of states n

$$\mathbf{A} = \langle \alpha_1, \alpha_\infty, \{A_\sigma\}_{\sigma \in \Sigma} \rangle$$

$$\begin{aligned} \alpha_1 &\in \mathbb{R}^n && \text{(initial weights)} \\ \alpha_\infty &\in \mathbb{R}^n && \text{(terminal weights)} \\ A_\sigma &\in \mathbb{R}^{n \times n} && \text{(transition weights)} \end{aligned}$$

- Computes a function $f_{\mathbf{A}} : \Sigma^\star \to \mathbb{R}$

$$f_{\mathbf{A}}(x) = f_{\mathbf{A}}(x_1 \cdots x_t) = \alpha_1^\top A_{x_1} \cdots A_{x_t} \alpha_\infty = \alpha_1^\top A_x \alpha_\infty$$

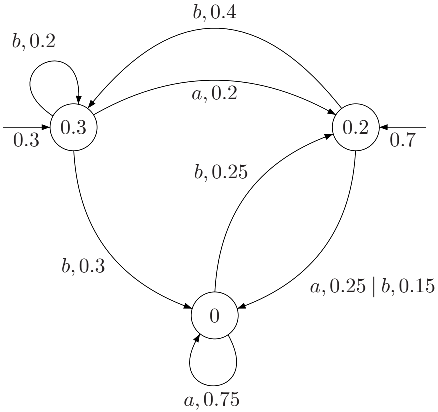
Examples – Probabilistic Finite Automata

- Compute / generate distributions over strings $\mathbb{P}[x]$

$$\alpha_1^\top = [0.3 \;\; 0 \;\; 0.7]$$

$$\alpha_\infty^\top = [0.2 \;\; 0 \;\; 0.2]$$

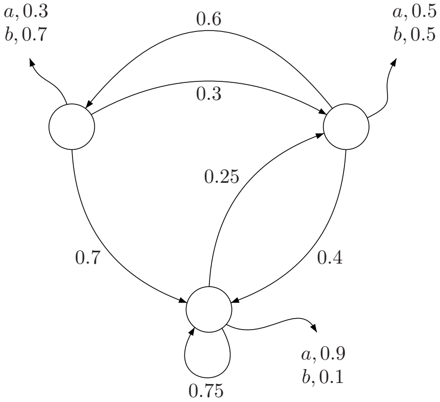
$$A_a = \begin{bmatrix} 0 & 0 & 0.2 \\ 0 & 0.75 & 0 \\ 0 & 0.25 & 0 \end{bmatrix}$$



Examples – Hidden Markov Models

- Generates infinite strings, computes probabilities of prefixes $\mathbb{P}[x\Sigma^\star]$
- Emission and transition are conditionally independent given state

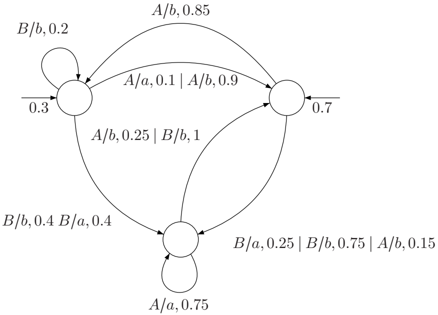
$$\alpha_1^\top = [0.3 \;\; 0.3 \;\; 0.4]$$
$$\alpha_\infty^\top = [1 \;\; 1 \;\; 1]$$
$$A_a = O_a \cdot T$$
$$T = \begin{bmatrix} 0 & 0.7 & 0.3 \\ 0 & 0.75 & 0.25 \\ 0 & 0.4 & 0.6 \end{bmatrix}$$
$$O_a = \begin{bmatrix} 0.3 & 0 & 0 \\ 0 & 0.9 & 0 \\ 0 & 0 & 0.5 \end{bmatrix}$$



Examples – Probabilistic Finite State Transducers

- Compute conditional probabilities $\mathbb{P}[y|x] = \alpha_1^\top A_x^y \alpha_\infty$ for pairs $(x, y) \in (\Sigma \times \Delta)^\star$, must have $|x| = |y|$
- Can also assume models factorized like in HMM

$$\alpha_1^\top = [0.3 \;\; 0 \;\; 0.7]$$
$$\alpha_\infty^\top = [1 \;\; 1 \;\; 1]$$
$$A_B^b = \begin{bmatrix} 0.2 & 0.4 & 0 \\ 0 & 0 & 1 \\ 0 & 0.75 & 0 \end{bmatrix}$$



WAs and Forward-Backward Recursions

Forward–Backward Factorization

- $\mathbf{A}$ defines *forward* and *backward* maps $f^F_{\mathbf{A}}, f^B_{\mathbf{A}} : \Sigma^\star \to \mathbb{R}^n$
- Such that for any splitting $x = y \cdot z$ one has $f_{\mathbf{A}}(x) = f^F_{\mathbf{A}}(y) \cdot f^B_{\mathbf{A}}(z)$

$$f^F_{\mathbf{A}}(y) = \alpha_1^\top A_y \qquad \text{and} \qquad f^B_{\mathbf{A}}(z) = A_z \alpha_\infty$$

Example

- For a PFA $\mathbf{A}$ and $i \in [n]$, one has
- $[f^F_{\mathbf{A}}(y)]_i = [\alpha_1^\top A_y]_i = \mathbb{P}[y \,,\, h_{|y|+1} = i]$
- $[f^B_{\mathbf{A}}(z)]_i = [A_z \alpha_\infty]_i = \mathbb{P}[z \,|\, h = i]$

Consequences

- String structure has direct relation to computation structure
- In particular, strings sharing prefixes or suffixes share computations
- Information on A_a can be recovered from $f_{\mathbf{A}}(yaz)$, $f^F_{\mathbf{A}}(y)$, and $f^B_{\mathbf{A}}(z)$:

$$f_{\mathbf{A}}(yaz) = f^F_{\mathbf{A}}(y) A_a f^B_{\mathbf{A}}(z)$$

The Hankel Matrix

- The *Hankel matrix* of $f : \Sigma^\star \to \mathbb{R}$ is $H_f \in \mathbb{R}^{\Sigma^\star \times \Sigma^\star}$
- For $y, z \in \Sigma^\star$, entries are defined by $H_f(y, z) = f(y \cdot z)$
- Given $\mathcal{P}, \mathcal{S} \subseteq \Sigma^\star$ will consider sub-blocks $H_f(\mathcal{P}, \mathcal{S}) \in \mathbb{R}^{\mathcal{P} \times \mathcal{S}}$
- Very *redundant* representation for f – $f(x)$ appears $|x| + 1$ times

$$
\begin{array}{cc}
 & \begin{array}{cccccc} \epsilon & a & b & aa & ab & \cdots \end{array} \\
\begin{array}{c} \epsilon \\ a \\ b \\ aa \\ ab \\ \vdots \end{array} &
\left[\begin{array}{cccccc}
 & & \vdots & & \vdots & \\
\cdots & \cdots & \vdots & \cdots & f(aab) & \\
 & & \vdots & & & \\
\cdots & \cdots & f(aab) & & & \\
 & & & & & \\
 & & & & &
\end{array}\right]
\end{array}
$$

Schützenberger's Theorem

Theorem: $\mathrm{rank}(H_f) \leqslant n$ if and only if $f = f_{\mathbf{A}}$ with $|\mathbf{A}| = n$

In particular, $\mathrm{rank}(H_f)$ is size of smallest WA for f

Proof ($\Leftarrow$)

- Write $F = f_{\mathbf{A}}^F(\Sigma^\star) \in \mathbb{R}^{\Sigma^\star \times n}$ and $B = f_{\mathbf{A}}^B(\Sigma^\star) \in \mathbb{R}^{n \times \Sigma^\star}$
- Note $H_f = F \cdot B$
- Then, $\mathrm{rank}(H_f) \leqslant n$

Proof ($\Rightarrow$)

- Assume $\mathrm{rank}(H_f) = n$
- Take *rank factorization* $H_f = F \cdot B$ with $F \in \mathbb{R}^{\Sigma^\star \times n}$ and $B \in \mathbb{R}^{n \times \Sigma^\star}$
- Let $\alpha_1^\top = F(\epsilon, [n])$ and $\alpha_\infty = B([n], \epsilon)$ (note $\alpha_1^\top \alpha_\infty = f(\epsilon)$)
- Let $A_\sigma = B([n], \sigma \cdot \Sigma^\star) \cdot B^+ \in \mathbb{R}^{n \times n}$ (note $A_\sigma \cdot B([n], x) = B([n], \sigma \cdot x)$)
- By induction on $|x|$ we get $\alpha_1^\top A_x \alpha_\infty = f(x)$

Schützenberger's Theorem

Theorem: $\operatorname{rank}(H_f) \leqslant n$ if and only if $f = f_{\mathbf{A}}$ with $|\mathbf{A}| = n$

In particular, $\operatorname{rank}(H_f)$ is size of smallest WA for f

Proof ($\Leftarrow$)

- Write $F = f^F_{\mathbf{A}}(\Sigma^\star) \in \mathbb{R}^{\Sigma^\star \times n}$ and $B = f^B_{\mathbf{A}}(\Sigma^\star) \in \mathbb{R}^{n \times \Sigma^\star}$
- Note $H_f = F \cdot B$
- Then, $\operatorname{rank}(H_f) \leqslant n$

Proof ($\Rightarrow$)

- Assume $\operatorname{rank}(H_f) = n$
- Take *rank factorization* $H_f = F \cdot B$ with $F \in \mathbb{R}^{\Sigma^\star \times n}$ and $B \in \mathbb{R}^{n \times \Sigma^\star}$
- Let $\alpha_1^\top = F(\epsilon, [n])$ and $\alpha_\infty = B([n], \epsilon)$ (note $\alpha_1^\top \alpha_\infty = f(\epsilon)$)
- Let $A_\sigma = B([n], \sigma \cdot \Sigma^\star) \cdot B^+ \in \mathbb{R}^{n \times n}$ (note $A_\sigma \cdot B([n], x) = B([n], \sigma \cdot x)$)
- By induction on $|x|$ we get $\alpha_1^\top A_x \alpha_\infty = f(x)$

Schützenberger's Theorem

Theorem: $\mathrm{rank}(H_f) \leqslant n$ if and only if $f = f_{\mathbf{A}}$ with $|\mathbf{A}| = n$

In particular, $\mathrm{rank}(H_f)$ is size of smallest WA for f

Proof ($\Leftarrow$)

- Write $F = f_{\mathbf{A}}^{F}(\Sigma^\star) \in \mathbb{R}^{\Sigma^\star \times n}$ and $B = f_{\mathbf{A}}^{B}(\Sigma^\star) \in \mathbb{R}^{n \times \Sigma^\star}$
- Note $H_f = F \cdot B$
- Then, $\mathrm{rank}(H_f) \leqslant n$

Proof ($\Rightarrow$)

- Assume $\mathrm{rank}(H_f) = n$
- Take *rank factorization* $H_f = F \cdot B$ with $F \in \mathbb{R}^{\Sigma^\star \times n}$ and $B \in \mathbb{R}^{n \times \Sigma^\star}$
- Let $\alpha_1^\top = F(\epsilon, [n])$ and $\alpha_\infty = B([n], \epsilon)$ (note $\alpha_1^\top \alpha_\infty = f(\epsilon)$)
- Let $A_\sigma = B([n], \sigma \cdot \Sigma^\star) \cdot B^+ \in \mathbb{R}^{n \times n}$ (note $A_\sigma \cdot B([n], x) = B([n], \sigma \cdot x)$)
- By induction on $|x|$ we get $\alpha_1^\top A_x \alpha_\infty = f(x)$

The Spectral Method

Idea: Use SVD decomposition to obtain a factorization of H

- Given H and H_σ over basis $(\mathcal{P}, \mathcal{S})$
- Compute *compact* SVD as $\mathrm{H} = USV^\top$ with

$$U \in \mathbb{R}^{\mathcal{P} \times n} \qquad S \in \mathbb{R}^{n \times n} \qquad V \in \mathbb{R}^{\mathcal{S} \times n}$$

- Let $A_\sigma = (\mathrm{H}V)^+(\mathrm{H}_\sigma V)$ – corresponds to rank factorization $\mathrm{H} = (\mathrm{H}V)V^\top$

Properties

- Easy to implement: just linear algebra
- Fast to compute: $O(\max\{|\mathcal{P}|, |\mathcal{S}|\}^3)$
- Noise tolerant: $\hat{\mathrm{H}} \approx \mathrm{H}$ and $\hat{\mathrm{H}}_\sigma \approx \mathrm{H}_\sigma$ implies $\hat{A}_\sigma \approx A_\sigma$

Spetral Methods from the Automata-Theory Perspective

- Algorithmic and Miscellaneous Problems
  - Interpretation as an optimization problem - from linear algebra to convex optimization
  - Finding a basis via random sampling - knowing $(\mathcal{P}, \mathcal{S})$
- Direct Applications
  - Learning stochastic rational languages - any probability distribution computed by WA
  - Learning probabilistic finite state transducers
  - Learning tree distributions
- Composition with Other Methods
  - Combination with matrix completion for learning non-stochastic functions - when $f : \Sigma^* \to \mathbb{R}$ is not related to a probability distribution

An Optimization Point of View

- Idea: Replace linear algebra with optimization primitives - make it possible to use the ML optimization toolkit.
- Algorithms
  - Spectral optimization: $\min_{\{A_\sigma\}, V_n^\top V_n = I} \sum_{\sigma \in \Sigma} ||H V_n A_\sigma - H_\sigma V_n||_F^2$
  - Convex relaxation: $\min_{A_\Sigma} ||H A_\Sigma - H_\Sigma||_F^2 + \tau ||A_\Sigma||_*$
- Properties
  - Equivalent in some situations and choice of parameters
  - Experiments show convex relaxation can be better in cases known to be difficult for the spectral method
- Open problems/ Future Work
  - Design problem-specific optimization algorithms
  - Constrain learned models imposing further regularizations, e.g. sparsity

Learning Probabilistic Finite State Transducers [BQC'11]

Idea: Learn a function $f : (\Sigma \times \Delta)^\star \to \mathbb{R}$ computing $\mathbb{P}[y|x]$ Learning Model

- Input is sample of aligned sequences (x^i, y^i), $|x^i| = |y^i|$
- Drawn i.i.d. from distribution $\mathbb{P}[x, y] = \mathbb{P}[y|x]\, D(x)$
- Want to assume as little as possible on D
- Performance measured against x generated from D

Properties

- Assuming independece $A_\sigma^\delta = O_\delta \cdot T_\sigma$, sample bound scales mildly with input alphabet $|\Sigma|$
- For applications, need to align sequences prior to learning – or use iterative procedures

Open problems / Current Work

- Deal with alignments inside the model

Learning PFST over very large input alphabets

- Goal: Learn a function $f : (\Sigma \times \Delta)^\star \to \mathbb{R}$ computing $\mathbb{P}[x, y]$ where Δ can be arbitrarily large.
- Idea: Transition function as a linear combination of basic transitions.
- Model
  - Assume a set of feature functions $\Phi(x) = [\phi_1(x), \dots, \phi_k(x)]$

$$
\begin{aligned}
f(x, y) &= \alpha_1^T A(\phi(x_1), y_1) \cdots, A(\phi(x_T), y_T)\alpha_\infty \\
&= \alpha_1^T \left(\sum_{l=1}^{k} \phi_l(x_1) O_l^{y_1} \right) \cdots \left(\sum_{l=1}^{k} \phi_l(x_T) O_l^{y_T} \right) \alpha_\infty
\end{aligned}
$$

Learning FST: Handling Missing Alignments

- Goal: Learn a function $g : (\Sigma^\star \times \Delta^\star) \to \mathbb{R}$ computing $\mathbb{P}[x, y]$
- Model:

$$\mathbb{P}(x, y) = \sum_{z \in \mathcal{Z}(x,y)} f(z)$$

- Aligned Sequences:

$$z = \begin{array}{ccc} y_1 & y_2 & y_3 \\ x_1 & x_2 & \end{array} \begin{bmatrix} y_4 \\ x_3 \end{bmatrix} \begin{bmatrix} y_5 \\ x_4 \end{bmatrix} \; y_6 \; \begin{bmatrix} y_7 \\ x_5 \end{bmatrix}$$

corresponds to a sequence of symbol pairs:

$$(x_1, \lambda)(x_2, \lambda)(\lambda, y_1)(\lambda, y_2)(\lambda, y_3)(x_3, y_4)(x_4, y_5)(\lambda, y_6)(x_5, y_7)$$

- WA over aligned sequences:

$$f(z) = \alpha_1^\top A_{x_1} A_{x_2} A^{y_1} A^{y_2} A^{y_3} A^{y_4}_{x_3} A^{y_5}_{x_4} A^{y_6} A^{y_7}_{x_5} \alpha_\infty$$

- Intuitively:
  - A_σ operators over $\Sigma^\star$
  - A^δ operators over $\Delta^\star$
  - A^σ_δ operators over $(\Sigma \times \Delta)^\star$

Forward-Backward Maps

$$z = \begin{array}{ccccc} y_1 & & y_2 & & y_3 \\ & x_1 & & x_2 & \end{array} \begin{bmatrix} y_4 \\ x_3 \end{bmatrix} \; y_5 \; \begin{bmatrix} y_6 \\ x_4 \end{bmatrix}$$

$$\begin{aligned}
f(z) &= \alpha_1^\top A_{x_{1:2}} A^{y_{1:3}} A_{x_3}^{y_4} A^{y_5} A_{x_4}^{y_6} \alpha_\infty \\
&= f^F\left(\begin{array}{ccccc} y_1 & & y_2 & & y_3 \\ & x_1 & & x_2 & \end{array} \begin{bmatrix} y_4 \\ x_3 \end{bmatrix} \right) \cdot f^B \left(y_5 \; \begin{bmatrix} y_6 \\ x_4 \end{bmatrix} \right) \\
&= f^F\left(\begin{array}{ccccc} y_1 & & y_2 & & y_3 \\ & x_1 & & x_2 & \end{array} \right) \cdot A_{x_3}^{y_4} \cdot f^B \left(y_5 \; \begin{bmatrix} y_6 \\ x_4 \end{bmatrix} \right)
\end{aligned}$$

Fully Observed Hankels

- Hankel over Aligned Sequences:

$$\begin{aligned}
H\left(\begin{smallmatrix} y_1 y_2 y_3 \\ x_1 x_2 \end{smallmatrix}, y_4 \begin{bmatrix} y_5 \\ x_3 \end{bmatrix}\right) &= f\left(\begin{smallmatrix} y_1 y_2 y_3 y_4 \\ x_1 x_2 \end{smallmatrix} \begin{bmatrix} y_5 \\ x_3 \end{bmatrix}\right) \\
H_{\sigma}\left(\begin{smallmatrix} y_1 y_2 y_3 \\ x_1 x_2 \end{smallmatrix}, y_4 \begin{bmatrix} y_5 \\ x_3 \end{bmatrix}\right) &= f\left(\begin{smallmatrix} y_1 y_2 y_3 y_4 \\ x_1 x_2 \sigma \end{smallmatrix} \begin{bmatrix} y_5 \\ x_3 \end{bmatrix}\right) \\
H^{\delta}\left(\begin{smallmatrix} y_1 y_2 y_3 \\ x_1 x_2 \end{smallmatrix}, y_4 \begin{bmatrix} y_5 \\ x_3 \end{bmatrix}\right) &= f\left(\begin{smallmatrix} y_1 y_2 y_3 \delta y_4 \\ x_1 x_2 \end{smallmatrix} \begin{bmatrix} y_5 \\ x_3 \end{bmatrix}\right) \\
H_{\sigma}^{\delta}\left(\begin{smallmatrix} y_1 y_2 y_3 \\ x_1 x_2 \end{smallmatrix}, y_4 \begin{bmatrix} y_5 \\ x_3 \end{bmatrix}\right) &= f\left(\begin{smallmatrix} y_1 y_2 y_3 \\ x_1 x_2 \end{smallmatrix} \begin{bmatrix} \delta \\ \sigma \end{bmatrix} y_4 \begin{bmatrix} y_5 \\ x_3 \end{bmatrix}\right)
\end{aligned}$$

- Hankel Factorizations:

$$H = FB$$

$$H_{\sigma}^{\delta} = F A_{\sigma}^{\delta} B$$

Hankel over Aligned Substrings

- A Hankel over aligned substrings:

$$H^\star\left(\begin{smallmatrix}\delta_1\\ \end{smallmatrix}\begin{bmatrix}\delta_2\\ \sigma_1\end{bmatrix}, \sigma_2\right) = H\left(\begin{smallmatrix}\Delta^\star\delta_1\\ \Sigma^\star\end{smallmatrix}\begin{bmatrix}\delta_2\\ \sigma_1\end{bmatrix}\ ,\ \begin{smallmatrix}\Delta^\star\\ \sigma_2\ \Sigma^\star\end{smallmatrix}\right)$$

- Problem: we do not observe aligned sequences!

Hankel over Aligned Substrings

- A Hankel over aligned substrings:

$$H^\star\left({}^{\delta_1}\begin{bmatrix}\delta_2\\ \sigma_1\end{bmatrix}, {}_{\sigma_2}\right) = H\left(\begin{smallmatrix}\Delta^\star\\ \Sigma^\star\end{smallmatrix}{}^{\delta_1}\begin{bmatrix}\delta_2\\ \sigma_1\end{bmatrix}\ ,\ {}_{\sigma_2}\begin{smallmatrix}\Delta^\star\\ \Sigma^\star\end{smallmatrix}\right)$$

- Problem: we do not observe aligned sequences!

Observable Statistics

$$\begin{aligned}
\mathbb{E}\left[|x,y|^{\delta_1\delta_2}_{\sigma_1\sigma_2}\right] &= \sum_{\substack{x_s,x_p\in\Sigma^\star \\ y_s,y_p\in\Delta^\star}} \mathbb{P}[x_p\,\sigma_1\sigma_2 x_s,\, y_p\,\delta_1\delta_2 y_s] \\
&= \begin{smallmatrix}\cdots\delta_1\delta_2\cdots\cdots\cdots\cdots\cdots \\ \cdots\cdots\cdots\cdots\cdots\sigma_1\sigma_2\cdots\end{smallmatrix} \quad + \quad \ldots \\
&+ \begin{smallmatrix}\cdots\cdots\delta_1 \\ \cdots\cdots\cdots\end{smallmatrix}\begin{bmatrix}\delta_2\\\sigma_1\end{bmatrix}\begin{smallmatrix}\cdots\cdots\cdots \\ \sigma_2\cdots\cdots\end{smallmatrix} \quad + \quad \begin{smallmatrix}\cdots\cdots \\ \cdots\cdots\end{smallmatrix}\begin{bmatrix}\delta_1\\\sigma_1\end{bmatrix}\begin{bmatrix}\delta_2\\\sigma_2\end{bmatrix}\begin{smallmatrix}\cdots\cdots \\ \cdots\cdots\end{smallmatrix} \\
&+ \quad \ldots \quad + \quad \begin{smallmatrix}\cdots\cdots\cdots\cdots\cdots\delta_1\delta_2\cdots \\ \cdots\sigma_1\sigma_2\cdots\cdots\cdots\cdots\cdots\end{smallmatrix}
\end{aligned}$$

where:

$$\begin{smallmatrix}\cdots\cdots\delta_1 \\ \cdots\cdots\cdots\end{smallmatrix}\begin{bmatrix}\delta_2\\\sigma_1\end{bmatrix}\begin{smallmatrix}\cdots\cdots\cdots \\ \sigma_2\cdots\cdots\end{smallmatrix} = H^\star\left(\delta_1\begin{bmatrix}\delta_2\\\sigma_1\end{bmatrix}, \sigma_2\right)$$

Guessing a good Hankel

- Idea:
  - Guess the entries in the Hankel
  - Use rank constraints and observables to guide your guess
- Observable Constraints:
  - $\mathbb{E}[|x, y|^{\delta_1 \delta_2}_{\sigma_1 \sigma_2}] =$ sums of entries in $H^\star$
- Optimization:

$$\min_{H^\star} ||\, H^\star||_* \quad \text{subject to: linear constraints on observables}$$

- Number of Variables in $H^\star$: $C \cdot |\Sigma \times \Delta|^4$

Conclusion

- Summary:
  - Spectral algorithms follow directly from classical algebraic methods for learning automata
  - Automata Theory Perspective in a Nutshell: the Hankel trick
    - Recipe to derive learning algorithms for many models.
    - Convex optimization + Hankel trick
  - Extensions: guess missing Hankel entries
- Future Directions:
  - Can we learn subclasses of PCFG?
- A reference:
  - R. Gavaldà and J. Castro, Learning Probability Distributions Generated by Finite-State Machines, tutorial at ICGI-2012