Automata Learning

Borja Balle

Amazon Research Cambridge[1]

Foundations of Programming Summer School (Oxford) — July 2018

[1] Based on work completed before joining Amazon

Brief History of Automata Learning

1967 Gold: Regular languages are learnable in the limit

1987 Angluin: Regular languages are learnable from queries

1993 Pitt & Warmuth: PAC-learning DFA is NP-hard

1994 Kearns & Valiant: Cryptographic hardness

⋮ Clark, Denis, de la Higuera, Oncina, others: Combinatorial methods meet statistics and linear algebra

2009 Hsu-Kakade-Zhang & Bailly-Denis-Ralaivola: Spectral learning

Goals of This Tutorial

Goals

- Motivate spectral learning techniques for weighted automata and related models on sequential and tree-structured data
- Provide the key intuitions and fundamental results to effectively navigate the literature
- Survey some formal learning results and give overview of some applications
- Discuss role of linear algebra, concentration bounds, and learning theory in this area

Non-Goals

- Dive deep into applications: instead pointers will be provided
- Provide an exhaustive treatment of automata learning: beyond the scope of an introductory lecture
- Give complete proofs of the presented results: illuminating proofs will be discussed, technical proofs omitted

Outline

1. Sequential Data and Weighted Automata
2. WFA Reconstruction and Approximation
3. PAC Learning for Stochastic WFA
4. Statistical Learning for WFA
5. Beyond Sequences: Transductions and Trees
6. Conclusion

Outline

Learning Sequential Data

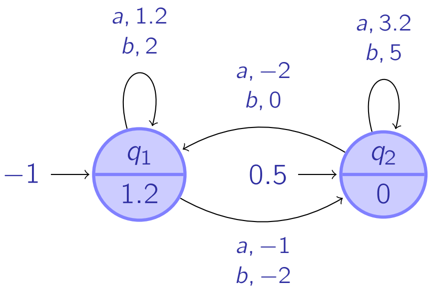
- Sequential data arises in numerous applications of Machine Learning:
  - Natural language processing
  - Computational biology
  - Time series analysis
  - Sequential decision-making
  - Robotics
- Learning from sequential data requires specialized algorithms
  - The most common ML algorithms assume the data can be represented as vectors of a fixed dimension
  - Sequences can have arbitrary length, and are compositional in nature
  - Similar things occur with trees, graphs, and other forms of structured data
- Sequential data can be diverse in nature
  - Continuous vs. discrete time vs. only order information
  - Continuous vs. discrete observations

Functions on Strings

- In this lecture we focus on sequences represented by strings on a finite alphabet: $\Sigma^\star$
- The goal will be to learn a function $f : \Sigma^\star \to \mathbb{R}$ from data
- The function being learned can represent many things, for example:
  - A *language* model: $f(\text{sentence})$ = likelihood of observing a sentence in a specific natural language
  - A *protein scoring* model: $f(\text{aminoacid sequence})$ = predicted activity of a protein in a biological reaction
  - A *reward* model: $f(\text{action sequence})$ = expected reward an agent will obtain after executing a sequence of actions
  - A *network* model: $f(\text{packet sequence})$ = probability that a sequence of packets will successfully transmit a message through a network
- These functions can be identified with a weighted language $f \in \mathbb{R}^{\Sigma^\star}$, an infinite-dimensional object
- In order to learn such functions we need a finite representation: **weighted automata**

Weighted Finite Automata

Graphical Representation



Algebraic Representation

$$\boldsymbol{\alpha} = \begin{bmatrix} -1 \\ 0.5 \end{bmatrix} \quad \boldsymbol{\beta} = \begin{bmatrix} 1.2 \\ 0 \end{bmatrix}$$

$$\mathbf{A}_a = \begin{bmatrix} 1.2 & -1 \\ -2 & 3.2 \end{bmatrix}$$

$$\mathbf{A}_b = \begin{bmatrix} 2 & -2 \\ 0 & 5 \end{bmatrix}$$

Weighted Finite Automaton

A WFA A with $n = |A|$ states is a tuple $A = \langle \boldsymbol{\alpha}, \boldsymbol{\beta}, \{\mathbf{A}_\sigma\}_{\sigma \in \Sigma} \rangle$ where $\boldsymbol{\alpha}, \boldsymbol{\beta} \in \mathbb{R}^n$ and $\mathbf{A}_\sigma \in \mathbb{R}^{n \times n}$

Language of a WFA

With every WFA $A = \langle \boldsymbol{\alpha}, \boldsymbol{\beta}, \{\mathbf{A}_\sigma\} \rangle$ with n states we associate a weighted language $f_A : \Sigma^\star \to \mathbb{R}$ given by

$$f_A(x_1 \cdots x_T) = \sum_{q_0, q_1, \dots, q_T \in [n]} \boldsymbol{\alpha}(q_0) \left(\prod_{t=1}^{T} \mathbf{A}_{x_t}(q_{t-1}, q_t) \right) \boldsymbol{\beta}(q_T)$$

$$= \boldsymbol{\alpha}^\top \mathbf{A}_{x_1} \cdots \mathbf{A}_{x_T} \boldsymbol{\beta} = \boldsymbol{\alpha}^\top \mathbf{A}_x \boldsymbol{\beta}$$

Recognizable/Rational Languages

A weighted language $f : \Sigma^\star \to \mathbb{R}$ is recognizable/rational if there exists a WFA A such that $f = f_A$. The smallest number of states of such a WFA is $\mathrm{rank}(f)$. A WFA A is minimal if $|A| = \mathrm{rank}(f_A)$.

Observation: The minimal A is not unique. Take any invertible matrix $\mathbf{Q} \in \mathbb{R}^{n \times n}$, then

$$\boldsymbol{\alpha}^\top \mathbf{A}_{x_1} \cdots \mathbf{A}_{x_T} \boldsymbol{\beta} = (\boldsymbol{\alpha}^\top \mathbf{Q})(\mathbf{Q}^{-1} \mathbf{A}_{x_1} \mathbf{Q}) \cdots (\mathbf{Q}^{-1} \mathbf{A}_{x_T} \mathbf{Q})(\mathbf{Q}^{-1} \boldsymbol{\beta})$$

Examples: DFA, HMM

Deterministic Finite Automata

- Weights in $\{0, 1\}$
- Initial: α indicator for initial state
- Final: β indicates accept/reject state
- Transition: $\mathbf{A}_\sigma(i, j) = \mathbb{I}[i \xrightarrow{\sigma} j]$
- $f_A : \Sigma^\star \to \{0, 1\}$ defines regular language

Hidden Markov Model

- Weights in $[0, 1]$
- Initial: α distribution over initial state
- Final: β vector of ones
- Transition: $\mathbf{A}_\sigma(i, j) = \mathbb{P}[i \xrightarrow{\sigma} j] = \mathbb{P}[i \to j]\mathbb{P}[i \xrightarrow{\sigma}]$
- $f_A : \Sigma^\star \to [0, 1]$ defines dynamical system

Hankel Matrices

research

Given a weighted language $f : \Sigma^\star \to \mathbb{R}$ define its Hankel matrix $\mathbf{H}_f \in \mathbb{R}^{\Sigma^\star \times \Sigma^\star}$ as

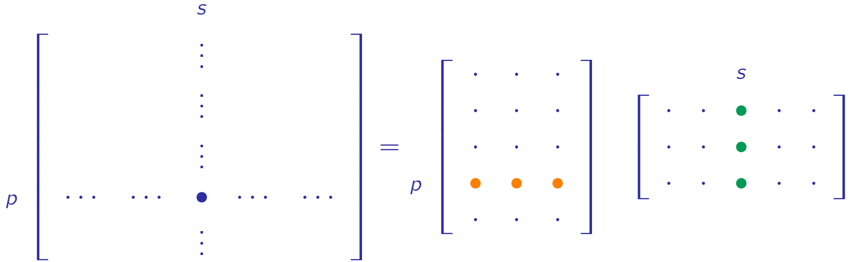
$$\mathbf{H}_f = \begin{array}{c} \\ \epsilon \\ a \\ b \\ \vdots \\ p \\ \vdots \end{array} \begin{array}{c} \begin{array}{cccccc} \epsilon & a & b & \cdots & s & \cdots \end{array} \\ \left[\begin{array}{cccccc} f(\epsilon) & f(a) & f(b) & & \vdots & \\ f(a) & f(aa) & f(ab) & & \vdots & \\ f(b) & f(ba) & f(bb) & & \vdots & \\ & & & & & \\ \cdots & \cdots & \cdots & & f(p \cdot s) & \\ & & & & & \end{array}\right] \end{array}$$

Fliess–Kronecker Theorem [Fli74]

The rank of $\mathbf{H}_f$ is finite if and only if f is rational, in which case $\text{rank}(\mathbf{H}_f) = \text{rank}(f)$

Intuition for the Fliess–Kronecker Theorem

$\mathbf{H}_{f_A} \in \mathbb{R}^{\Sigma^\star \times \Sigma^\star}$ $\qquad \mathbf{P}_A \in \mathbb{R}^{\Sigma^\star \times n}$ $\qquad \mathbf{S}_A \in \mathbb{R}^{n \times \Sigma^\star}$



$$f_A(p_1 \cdots p_T \cdot s_1 \cdots s_{T'}) = \underbrace{\boldsymbol{\alpha}^\top \mathbf{A}_{p_1} \cdots \mathbf{A}_{p_T}}_{\boldsymbol{\alpha}_A(p)} \; \underbrace{\mathbf{A}_{s_1} \cdots \mathbf{A}_{s_{T'}} \boldsymbol{\beta}}_{\boldsymbol{\beta}_A(s)}$$

Note: We call $\mathbf{H}_f = \mathbf{P}_A \mathbf{S}_A$ the *forward-backward factorization* induced by A

Outline

From Hankel to WFA

$$f(p_1 \cdots p_T s_1 \cdots s_{T'}) = \boldsymbol{\alpha}^\top \mathbf{A}_{p_1} \cdots \mathbf{A}_{p_T} \mathbf{A}_{s_1} \cdots \mathbf{A}_{s_{T'}} \boldsymbol{\beta}$$

$$\mathrm{H} = \begin{array}{c} \\ \\ \\ p \\ \\ \end{array} \overset{\displaystyle s}{\begin{bmatrix} & & \cdot & & \\ & & \cdot & & \\ & & \cdot & & \\ \cdot & \cdot & f(ps) & \cdot & \cdot \\ & & \cdot & & \end{bmatrix}} = \begin{bmatrix} \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \bullet & \bullet & \bullet \\ \cdot & \cdot & \cdot \end{bmatrix} \begin{bmatrix} \cdot & \cdot & \bullet & \cdot & \cdot \\ \cdot & \cdot & \bullet & \cdot & \cdot \\ \cdot & \cdot & \bullet & \cdot & \cdot \end{bmatrix}$$

$$f(p_1 \cdots p_T \sigma s_1 \cdots s_{T'}) = \boldsymbol{\alpha}^\top \mathbf{A}_{p_1} \cdots \mathbf{A}_{p_T} \mathbf{A}_a \mathbf{A}_{s_1} \cdots \mathbf{A}_{s_{T'}} \boldsymbol{\beta}$$

$$\mathrm{H}_\sigma = \begin{array}{c} \\ \\ \\ p \\ \\ \end{array} \overset{\displaystyle s}{\begin{bmatrix} & & \cdot & & \\ & & \cdot & & \\ & & \cdot & & \\ \cdot & \cdot & f(pas) & \cdot & \cdot \\ & & \cdot & & \end{bmatrix}} = \begin{bmatrix} \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \bullet & \bullet & \bullet \\ \cdot & \cdot & \cdot \end{bmatrix} \begin{bmatrix} \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet \end{bmatrix} \begin{bmatrix} \cdot & \cdot & \bullet & \cdot & \cdot \\ \cdot & \cdot & \bullet & \cdot & \cdot \\ \cdot & \cdot & \bullet & \cdot & \cdot \end{bmatrix}$$

Algebraically: Factorizing $\mathbf{H}$ lets us solve for $\mathbf{A}_a$

$$\mathbf{H} = \mathbf{P}\ \mathbf{S} \quad \Longrightarrow \quad \mathbf{H}_\sigma = \mathbf{P}\ \mathbf{A}_\sigma\ \mathbf{S} \quad \Longrightarrow \quad \mathbf{A}_\sigma = \mathbf{P}^+\ \mathbf{H}_\sigma\ \mathbf{S}^+$$

Aside: Moore–Penrose Pseudo-inverse

For any $\mathbf{M} \in \mathbb{R}^{n \times m}$ there exists a unique *pseudo-inverse* $\mathbf{M}^+ \in \mathbb{R}^{m \times n}$ satisfying:

- $\mathbf{M}\mathbf{M}^+\mathbf{M} = \mathbf{M}$, $\mathbf{M}^+\mathbf{M}\mathbf{M}^+ = \mathbf{M}^+$, and $\mathbf{M}^+\mathbf{M}$ and $\mathbf{M}\mathbf{M}^+$ are symmetric
- If $\operatorname{rank}(\mathbf{M}) = n$ then $\mathbf{M}\mathbf{M}^+ = \mathbf{I}$, and if $\operatorname{rank}(\mathbf{M}) = m$ then $\mathbf{M}^+\mathbf{M} = \mathbf{I}$
- If $\mathbf{M}$ is square and invertible then $\mathbf{M}^+ = \mathbf{M}^{-1}$

Given a system of linear equations $\mathbf{M}\mathbf{u} = \mathbf{v}$, the following is satisfied:

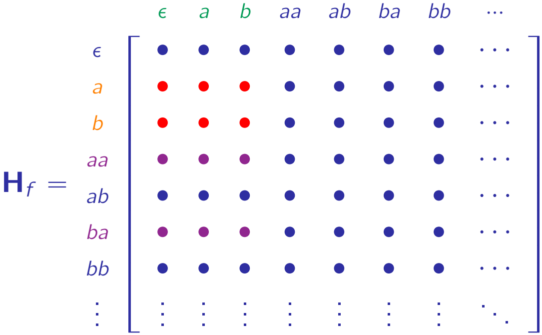
$$\mathbf{M}^+\mathbf{v} = \underset{\mathbf{u} \in \operatorname{argmin} \|\mathbf{M}\mathbf{u} - \mathbf{v}\|_2}{\operatorname{argmin}} \|\mathbf{u}\|_2 .$$

In particular:

- If the system is completely determined, $\mathbf{M}^+\mathbf{v}$ solves the system
- If the system is underdetermined, $\mathbf{M}^+\mathbf{v}$ is the solution with smallest norm
- If the system is overdetermined, $\mathbf{M}^+\mathbf{v}$ is the minimum norm solution to the least-squares problem $\min \|\mathbf{M}\mathbf{u} - \mathbf{v}\|_2$

Finite Hankel Sub-Blocks

Given finite sets of prefixes and suffixes $\mathcal{P}, \mathcal{S} \subset \Sigma^\star$ and infinite Hankel matrix $\mathbf{H}_f \in \mathbb{R}^{\Sigma^\star \times \Sigma^\star}$ we define the sub-block $\mathbf{H} \in \mathbb{R}^{\mathcal{P} \times \mathcal{S}}$ and for $\sigma \in \Sigma$ the sub-block $\mathbf{H}_\sigma \in \mathbb{R}^{\mathcal{P}\sigma \times \mathcal{S}}$



WFA Reconstruction from Finite Hankel Sub-Blocks

research

Suppose $f : \Sigma^\star \to \mathbb{R}$ has rank n and $\epsilon \in \mathcal{P}, \mathcal{S} \subset \Sigma^\star$ are such that the sub-block $\mathbf{H} \in \mathbb{R}^{\mathcal{P} \times \mathcal{S}}$ of $\mathbf{H}_f$ satisfies $\operatorname{rank}(\mathbf{H}) = n$.

Let $A = \langle \boldsymbol{\alpha}, \boldsymbol{\beta}, \{\mathbf{A}_\sigma\} \rangle$ be obtained as follows:

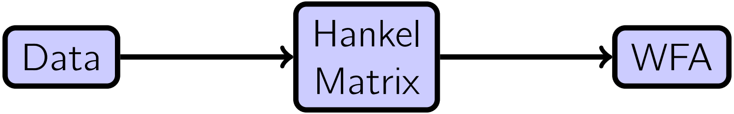
1. Compute a rank factorization $\mathbf{H} = \mathbf{P}\mathbf{S}$; i.e. $\operatorname{rank}(\mathbf{P}) = \operatorname{rank}(\mathbf{S}) = \operatorname{rank}(\mathbf{H})$
2. Let $\boldsymbol{\alpha}^\top$ (resp. $\boldsymbol{\beta}$) be the ϵ-row of $\mathbf{P}$ (resp. ϵ-column of $\mathbf{S}$)
3. Let $\mathbf{A}_\sigma = \mathbf{P}^+ \mathbf{H}_\sigma \mathbf{S}^+$, where $\mathbf{H}_\sigma \in \mathbb{R}^{\mathcal{P} \cdot \sigma \times \mathcal{S}}$ is a sub-block of $\mathbf{H}_f$

Claim The resulting WFA computes f and is minimal

Proof

- Suppose $\tilde{A} = \langle \tilde{\boldsymbol{\alpha}}, \tilde{\boldsymbol{\beta}}, \{\tilde{\mathbf{A}}_\sigma\} \rangle$ is a minimal WFA for f.
- It suffices to show there exists an invertible $\mathbf{Q} \in \mathbb{R}^{n \times n}$ such that $\boldsymbol{\alpha}^\top = \tilde{\boldsymbol{\alpha}}^\top \mathbf{Q}$, $\mathbf{A}_\sigma = \mathbf{Q}^{-1} \tilde{\mathbf{A}}_\sigma \mathbf{Q}$ and $\boldsymbol{\beta} = \mathbf{Q}^{-1} \tilde{\boldsymbol{\beta}}$.
- By minimality $\tilde{A}$ induces a rank factorization $\mathbf{H} = \tilde{\mathbf{P}}\tilde{\mathbf{S}}$ and also $\mathbf{H}_\sigma = \tilde{\mathbf{P}}\tilde{\mathbf{A}}_\sigma\tilde{\mathbf{S}}$.
- Since $\mathbf{A}_\sigma = \mathbf{P}^+ \mathbf{H}_\sigma \mathbf{S}^+ = \mathbf{P}^+ \tilde{\mathbf{P}} \tilde{\mathbf{A}}_\sigma \tilde{\mathbf{S}} \mathbf{S}^+$, take $\mathbf{Q} = \tilde{\mathbf{S}}\mathbf{S}^+$.
- Check $\mathbf{Q}^{-1} = \mathbf{P}^+ \tilde{\mathbf{P}}$ since $\mathbf{P}^+ \tilde{\mathbf{P}} \tilde{\mathbf{S}} \mathbf{S}^+ = \mathbf{P}^+ \mathbf{H} \mathbf{S}^+ = \mathbf{P}^+ \mathbf{P} \mathbf{S} \mathbf{S}^+ = \mathbf{I}$.

WFA Learning Algorithms via the Hankel Trick



1. Estimate a Hankel matrix from data
   - For stochastic automata: counting empirical frequencies
   - In general: empirical risk minimization
   - Inductive bias: enforcing low-rank Hankel will yield less states in WFA
   - Parameters: rows and columns of Hankel sub-block
2. Recover a WFA from the Hankel matrix
   - Direct application of WFA reconstruction algorithm

Question: How robust to noise are these steps? Can we the learned WFA is a good representation of the data?

Norms on WFA

research

Weighted Finite Automaton

A WFA with n states is a tuple $A = \langle \boldsymbol{\alpha}, \boldsymbol{\beta}, \{\mathbf{A}_\sigma\}_{\sigma \in \Sigma} \rangle$ where $\boldsymbol{\alpha}, \boldsymbol{\beta} \in \mathbb{R}^n$ and $\mathbf{A}_\sigma \in \mathbb{R}^{n \times n}$

Let $p, q \in [1, \infty]$ be Hölder conjugate $\frac{1}{p} + \frac{1}{q} = 1$.

The (p, q)-norm of a WFA A is given by

$$\|A\|_{p,q} = \max \left\{ \|\boldsymbol{\alpha}\|_p, \|\boldsymbol{\beta}\|_q, \max_{\sigma \in \Sigma} \|\mathbf{A}_\sigma\|_q \right\} \quad ,$$

where $\|\mathbf{A}_\sigma\|_q = \sup_{\|\mathrm{v}\|_q \leqslant 1} \|\mathbf{A}_\sigma \mathbf{v}\|_q$ is the q-induced norm.

Example For probabilistic automata $A = \langle \boldsymbol{\alpha}, \boldsymbol{\beta}, \{\mathbf{A}_\sigma\} \rangle$ with $\boldsymbol{\alpha}$ probability distribution, $\boldsymbol{\beta}$ acceptance probabilities, $\mathbf{A}_\sigma$ row (sub-)stochastic matrices we have $\|A\|_{1,\infty} = 1$

Perturbation Bounds: Automaton→Language [Bal13]

research

Suppose $A = \langle \boldsymbol{\alpha}, \boldsymbol{\beta}, \{\mathbf{A}_\sigma\} \rangle$ and $A' = \langle \boldsymbol{\alpha}', \boldsymbol{\beta}', \{\mathbf{A}'_\sigma\} \rangle$ are WFA with n states satisfying $\|A\|_{p,q} \leqslant \rho$, $\|A'\|_{p,q} \leqslant \rho$, $\max\{\|\boldsymbol{\alpha} - \boldsymbol{\alpha}'\|_p, \|\boldsymbol{\beta} - \boldsymbol{\beta}'\|_q, \max_{\sigma \in \Sigma} \|\mathbf{A}_\sigma - \mathbf{A}'_\sigma\|_q\} \leqslant \Delta$.

Claim The following holds for any $x \in \Sigma^\star$:

$$|f_A(x) - f_{A'}(x)| \leqslant (|x| + 2)\rho^{|x|+1}\Delta \quad .$$

Proof By induction on $|x|$ we first prove $\|\mathbf{A}_x - \mathbf{A}'_x\|_q \leqslant |x|\rho^{|x|-1}\Delta$:

$$\|\mathbf{A}_{x\sigma} - \mathbf{A}'_{x\sigma}\|_q \leqslant \|\mathbf{A}_x - \mathbf{A}'_x\|_q\|\mathbf{A}_\sigma\|_q + \|\mathbf{A}'_x\|_q\|\mathbf{A}_\sigma - \mathbf{A}'_\sigma\|_q \leqslant |x|\rho^{|x|}\Delta + \rho^{|x|}\Delta = (|x| + 1)\rho^{|x|}\Delta \quad .$$

$$\begin{aligned}
|f_A(x) - f_{A'}(x)| &= |\boldsymbol{\alpha}^\top \mathbf{A}_x \boldsymbol{\beta} - \boldsymbol{\alpha}'^\top \mathbf{A}'_x \boldsymbol{\beta}'| \leqslant |\boldsymbol{\alpha}^\top(\mathbf{A}_x\boldsymbol{\beta} - \mathbf{A}'_x\boldsymbol{\beta}')| + |(\boldsymbol{\alpha} - \boldsymbol{\alpha}')^\top \mathbf{A}'_x\boldsymbol{\beta}'| \\
&\leqslant \|\boldsymbol{\alpha}\|_p\|\mathbf{A}_x\boldsymbol{\beta} - \mathbf{A}'_x\boldsymbol{\beta}'\|_q + \|\boldsymbol{\alpha} - \boldsymbol{\alpha}'\|_p\|\mathbf{A}'_x\boldsymbol{\beta}'\|_q \\
&\leqslant \|\boldsymbol{\alpha}\|_p\|\mathbf{A}_x\|_q\|\boldsymbol{\beta} - \boldsymbol{\beta}'\|_q + \|\boldsymbol{\alpha}\|_p\|\mathbf{A}_x - \mathbf{A}'_x\|_q\|\boldsymbol{\beta}'\|_q + \|\boldsymbol{\alpha} - \boldsymbol{\alpha}'\|_p\|\mathbf{A}'_x\|_q\|\boldsymbol{\beta}'\|_q \\
&\leqslant \rho^{|x|+1}\|\boldsymbol{\beta} - \boldsymbol{\beta}'\|_q + \rho^2\|\mathbf{A}_x - \mathbf{A}'_x\|_q + \rho^{|x|+1}\|\boldsymbol{\alpha} - \boldsymbol{\alpha}'\|_p \\
&\leqslant \rho^{|x|+1}\Delta + \rho^2\rho^{|x|-1}|x|\Delta + \rho^{|x|+1}\Delta \quad .
\end{aligned}$$

Aside: Singular Value Decomposition (SVD)

research

For any $\mathbf{M} \in \mathbb{R}^{n \times m}$ with $\operatorname{rank}(\mathbf{M}) = k$ there exists a *singular value decomposition*

$$\mathbf{M} = \mathbf{U}\mathbf{D}\mathbf{V}^\top = \sum_{i=1}^{k} \mathfrak{s}_i \mathbf{u}_i \mathbf{v}_i^\top$$

- $\mathbf{D} \in \mathbb{R}^{k \times k}$ diagonal contains k sorted *singular values* $\mathfrak{s}_1 \geqslant \mathfrak{s}_2 \geqslant \cdots \geqslant \mathfrak{s}_k > 0$
- $\mathbf{U} \in \mathbb{R}^{n \times k}$ contains k *left singular vectors*, i.e. orthonormal columns $\mathbf{U}^\top \mathbf{U} = \mathbf{I}$
- $\mathbf{V} \in \mathbb{R}^{m \times k}$ contains k *right singular vectors*, i.e. orthonormal columns $\mathbf{V}^\top \mathbf{V} = \mathbf{I}$

Properties of SVD

- $\mathbf{M} = (\mathbf{U}\mathbf{D}^{1/2})(\mathbf{D}^{1/2}\mathbf{V}^\top)$ is a rank factorization
- Can be used to compute the pseudo-inverse as $\mathbf{M}^+ = \mathbf{V}\mathbf{D}^{-1}\mathbf{U}^\top$
- Provides optimal low-rank approximations. For $k' < k$, $\mathbf{M}_{k'} = \mathbf{U}_{k'}\mathbf{D}_{k'}\mathbf{V}_{k'}^\top = \sum_{i=1}^{k'} \mathfrak{s}_i \mathbf{u}_i \mathbf{v}_i^\top$ satisfies

$$\mathbf{M}_{k'} \in \operatorname*{argmin}_{\operatorname{rank}(\hat{M}) \leqslant k'} \|\mathbf{M} - \hat{\mathbf{M}}\|_2$$

Perturbation Bounds: Hankel→Automaton [Bal13]

research

- Suppose $f : \Sigma^\star \to \mathbb{R}$ has rank n and $\varepsilon \in \mathcal{P}, \mathcal{S} \subset \Sigma^\star$ are such that the sub-block $\mathbf{H} \in \mathbb{R}^{\mathcal{P} \times \mathcal{S}}$ of $\mathbf{H}_f$ satisfies $\operatorname{rank}(\mathbf{H}) = n$
- Let $A = \langle \boldsymbol{\alpha}, \boldsymbol{\beta}, \{\mathbf{A}_\sigma\} \rangle$ be obtained as follows:
  1. Compute the SVD factorization $\mathbf{H} = \mathbf{P}\mathbf{S}$; i.e. $\mathbf{P} = \mathbf{U}\mathbf{D}^{1/2}$ and $\mathbf{S} = \mathbf{D}^{1/2}\mathbf{V}^\top$
  2. Let $\boldsymbol{\alpha}^\top$ (resp. $\boldsymbol{\beta}$) be the ϵ-row of $\mathbf{P}$ (resp. ϵ-column of $\mathbf{S}$)
  3. Let $\mathbf{A}_\sigma = \mathbf{P}^+ \mathbf{H}_\sigma \mathbf{S}^+$, where $\mathbf{H}_\sigma \in \mathbb{R}^{\mathcal{P} \cdot \sigma \times \mathcal{S}}$ is a sub-block of $\mathbf{H}_f$
- Suppose $\hat{\mathbf{H}} \in \mathbb{R}^{\mathcal{P} \times \mathcal{S}}$ and $\hat{\mathbf{H}}_\sigma \in \mathbb{R}^{\mathcal{P} \cdot \sigma \times \mathcal{S}}$ satisfy $\max\{\|\mathbf{H} - \hat{\mathbf{H}}\|_2, \max_\sigma \|\mathbf{H}_\sigma - \hat{\mathbf{H}}_\sigma\|_2\} \leqslant \Delta$
- Let $\hat{A} = \langle \hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\beta}}, \{\hat{\mathbf{A}}_\sigma\} \rangle$ be obtained as follows:
  1. Compute the SVD rank-n approximation $\hat{\mathbf{H}} \approx \hat{\mathbf{P}}\hat{\mathbf{S}}$; i.e. $\hat{\mathbf{P}} = \hat{\mathbf{U}}_n \hat{\mathbf{D}}_n^{1/2}$ and $\hat{\mathbf{S}} = \hat{\mathbf{D}}_n^{1/2} \hat{\mathbf{V}}_n^\top$
  2. Let $\hat{\boldsymbol{\alpha}}^\top$ (resp. $\hat{\boldsymbol{\beta}}$) be the ϵ-row of $\hat{\mathbf{P}}$ (resp. ϵ-column of $\hat{\mathbf{S}}$)
  3. Let $\hat{\mathbf{A}}_\sigma = \hat{\mathbf{P}}^+ \hat{\mathbf{H}}_\sigma \hat{\mathbf{S}}^+$

Claim For any pair of Hölder conjugate (p, q) we have

$$\max\{\|\boldsymbol{\alpha} - \hat{\boldsymbol{\alpha}}\|_p, \|\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}\|_q, \max_\sigma \|\mathbf{A}_\sigma - \hat{\mathbf{A}}_\sigma\|_q\} \leqslant \mathcal{O}(\Delta)$$

Outline

Probabilities on Strings

Suppose the function $f : \Sigma^\star \to \mathbb{R}$ to be learned computes "probabilities": $f(x) \in [0, 1]$

Stochastic Languages

- Probability distribution over all strings: $\sum_{x \in \Sigma^\star} f(x) = 1$
- Can sample finite strings and try to learn the distribution

Dynamical Systems

- Probability distribution over strings of fixed length: for all $t \geqslant 0$, $\sum_{x \in \Sigma^t} f(x) = 1$
- Can sample (potentially infinite) prefixes and try to learn the dynamics

Hankel Estimation from Strings [HKZ09, BDR09]

Data: $S = \{x^1, \ldots, x^m\}$ containing m i.i.d. string from some distribution f over $\Sigma^\star$

Empirical Hankel matrix:

$$\hat{f}_S(x) = \frac{1}{m} \sum_{i=1}^{m} \mathbb{I}[x^i = x] \qquad \hat{\mathbf{H}}(p,s) = \hat{f}_S(p \cdot s)$$

Properties:

- Unbiased and consistent: $\lim_{m \to \infty} \hat{\mathbf{H}} = \mathbb{E}[\hat{\mathbf{H}}] = \mathbf{H}$
- Data inefficient:

$$S = \left\{ \begin{array}{c} aa,\ b,\ bab,\ a, \\ bbab,\ abb,\ babba,\ abbb, \\ ab,\ a,\ aabba,\ baa, \\ abbab,\ baba,\ bb,\ a \end{array} \right\} \longrightarrow \hat{\mathbf{H}} = \begin{array}{c} \\ \epsilon \\ a \\ b \\ ba \end{array} \begin{array}{c} \begin{array}{cc} a & b \end{array} \\ \begin{bmatrix} .19 & .06 \\ .06 & .06 \\ .00 & .06 \\ .06 & .06 \end{bmatrix} \end{array}$$

Hankel Estimation from Prefixes [BCLQ14]

research

Data: $S = \{x^1, \ldots, x^m\}$ containing m i.i.d. string from some distribution f over $\Sigma^\star$

Empirical Prefix Hankel matrix:

$$\bar{f}_S(x) = \frac{1}{m} \sum_{i=1}^{m} \mathbb{I}[x^i \in x\Sigma^\star]$$

Properties:

- $\mathbb{E}[\bar{f}_S(x)] = \sum_{y \in \Sigma^\star} f(xy) = \mathbb{P}_f[x\Sigma^\star]$
- If f is computed by WFA A, then

$$\mathbb{P}_f[x\Sigma^\star] = \sum_{y \in \Sigma^\star} f(xy) = \sum_{y \in \Sigma^\star} \boldsymbol{\alpha}^\top \mathbf{A}_x \mathbf{A}_y \boldsymbol{\beta} = \boldsymbol{\alpha}^\top \mathbf{A}_x \left(\sum_{y \in \Sigma^\star} \mathbf{A}_y \boldsymbol{\beta} \right)$$

$$= \boldsymbol{\alpha}^\top \mathbf{A}_x \left(\sum_{t \geqslant 0} (\mathbf{A}_{\sigma_1} + \cdots + \mathbf{A}_{\sigma_k})^t \boldsymbol{\beta} \right) = \boldsymbol{\alpha}^\top \mathbf{A}_x \left(\sum_{t \geqslant 0} \mathbf{A}^t \boldsymbol{\beta} \right)$$

$$= \boldsymbol{\alpha}^\top \mathbf{A}_x (\mathbf{I} - \mathbf{A})^{-1} \boldsymbol{\beta} = \boldsymbol{\alpha}^\top \mathbf{A}_x \bar{\boldsymbol{\beta}}$$

Hankel Estimation from Substrings [BCLQ14]

Data: $S = \{x^1, \dots, x^m\}$ containing m i.i.d. string from some distribution f over $\Sigma^\star$

Empirical Substring Hankel matrix:

$$\tilde{f}_S(x) = \frac{1}{m} \sum_{i=1}^{m} |x^i|_x \qquad |x^i|_x = \sum_{u,v \in \Sigma^\star} \mathbb{I}[x^i = uxv]$$

Properties:

- $\mathbb{E}[\tilde{f}_S(x)] = \sum_{u,v\in\Sigma^\star} f(uxv) = \sum_{y\in\Sigma^\star} |y|_x f(y) = \mathbb{E}_{y\sim f}[|y|_x]$
- If f is computed by WFA A, then

$$\mathbb{E}_{y\sim f}[|y|_x] = \sum_{y\in\Sigma^\star} |y|_x f(y) = \sum_{u,v\in\Sigma^\star} \boldsymbol{\alpha}^\top \mathbf{A}_u \mathbf{A}_x \mathbf{A}_v \boldsymbol{\beta}$$

$$= \boldsymbol{\alpha}^\top (\mathbf{I} - \mathbf{A})^{-1} \mathbf{A}_x (\mathbf{I} - \mathbf{A})^{-1} \boldsymbol{\beta} = \bar{\boldsymbol{\alpha}}^\top \mathbf{A}_x \bar{\boldsymbol{\beta}}$$

Hankel Estimation from a Single String [BM17]

research

Data: $x = x_1 \cdots x_m \cdots$ sampled from some dynamical system f over Σ

Empirical One-string Hankel matrix:

$$\mathring{f}_m(x) = \frac{1}{m} \sum_{i=1}^{m} \mathbb{I}[x_i x_{i+1} \cdots \in x\Sigma^\star]$$

Properties:

- $\mathbb{E}[\mathring{f}_m(x)] = \frac{1}{m} \sum_{u \in \Sigma^{<m}} f(ux) = \frac{1}{m} \sum_{i=0}^{m-1} \mathbb{P}_f[\Sigma^i x]$
- If f is computed by WFA A, then

$$\frac{1}{m} \sum_{i=0}^{m-1} \mathbb{P}_f[\Sigma^i x] = \frac{1}{m} \sum_{u \in \Sigma^{<m}} f(ux) = \frac{1}{m} \sum_{u \in \Sigma^{<m}} \boldsymbol{\alpha}^\top \mathbf{A}_u \mathbf{A}_x \boldsymbol{\beta}$$

$$= \left(\frac{1}{m} \sum_{i=0}^{m-1} \boldsymbol{\alpha}^\top \mathbf{A}^i \right) \mathbf{A}_x \boldsymbol{\beta} = \bar{\boldsymbol{\alpha}}_m^\top \mathbf{A}_x \boldsymbol{\beta}$$

Concentration Bounds for Hankel Estimation

- Consider a sub-block $\mathbf{H}$ over $(\mathcal{P}, \mathcal{S})$ fixed and the sample size $m \to \infty$
- In general one can show: with high probability over a sample S of size m

$$\|\hat{\mathbf{H}}_S - \mathbf{H}\| = O\left(\frac{1}{\sqrt{m}}\right)$$

  where
  - The hidden constants depend on the dimension of the sub-block $\mathcal{P} \times \mathcal{S}$ and properties of the strings in $\mathcal{P} \cdot \mathcal{S}$
  - The norm $\| \bullet \|$ can be either the operator or the Frobenius norm
  - Under the assumptions in the previous slides we can replace $\hat{\mathbf{H}}_S$ by $\bar{\mathbf{H}}_S$ (on prefixes), $\tilde{\mathbf{H}}_S$ (on substrings) or $\mathring{\mathbf{H}}_m$ (single trajectory)
- Proofs rely on a diversity of concentration inequalities; they can be found in **[DGH16, BM17]**

Aside: McDiarmid's Inequality

research

Let $\Phi : \Omega^m \to \mathbb{R}$ be such that

$$\forall i \in [m] \qquad \sup_{x_1,\ldots,x_m,x_i' \in \Omega} |\Phi(x_1,\ldots,x_i,\ldots,x_m) - \Phi(x_1,\ldots,x_i',\ldots,x_m)| \leqslant c$$

If $X = (X_1,\ldots,X_m)$ are i.i.d. from some distribution over Ω:

$$\mathbb{P}\left[\Phi(X) \geqslant \mathbb{E}\Phi(X) + t\right] \leqslant \exp\left(-\frac{2t^2}{mc^2}\right)$$

Equivalently, the following holds with probability at least $1-\delta$ over X:

$$\Phi(X) < \mathbb{E}\Phi(X) + c\sqrt{\frac{m}{2}\log(1/\delta)}$$

A Simple Proof via McDiarmid's Inequality [Bal13]

- Let $\Phi(x^1, \ldots, x^m) = \Phi(S) = \|\mathbf{H} - \hat{\mathbf{H}}_S\|_F$ with x^i i.i.d. from a distribution on $\Sigma^\star$
- Note $\hat{\mathbf{H}}_S = \frac{1}{m}\sum_{i=1}^m \hat{\mathbf{H}}_{x^i}$, where $\hat{\mathbf{H}}_x(p,s) = \mathbb{I}[p \cdot s = x]$
- Defining $c_{\mathcal{P},\mathcal{S}} = \max_x |\{(p,s) \in \mathcal{P} \times \mathcal{S} : p \cdot s = x\}| = \max_x \|\hat{\mathbf{H}}_x\|_F^2$ we get

$$|\Phi(S) - \Phi(S')| \leqslant \|\hat{\mathbf{H}}_S - \hat{\mathbf{H}}_{S'}\|_F = \frac{1}{m}\|\hat{\mathbf{H}}_{x^i} - \hat{\mathbf{H}}_{x^{i'}}\|_F \leqslant \frac{2}{m}\max\{\|\hat{\mathbf{H}}_{x^i}\|_F, \|\hat{\mathbf{H}}_{x^{i'}}\|_F\} \leqslant \frac{2\sqrt{c_{\mathcal{P},\mathcal{S}}}}{m}$$

- Using Jensen's inequality we can bound the expectation $\mathbb{E}\Phi(S) = \mathbb{E}\|\mathbf{H} - \hat{\mathbf{H}}_S\|_F$ as

$$\left(\mathbb{E}\|\mathbf{H} - \hat{\mathbf{H}}_S\|_F\right)^2 \leqslant \mathbb{E}\|\mathbf{H} - \hat{\mathbf{H}}_S\|_F^2 = \sum_{p,s}\mathbb{E}(\mathbf{H}(p,s) - \hat{\mathbf{H}}_S(p,s))^2 = \sum_{p,s}\mathbb{V}\hat{\mathbf{H}}_S(p,s)$$

$$= \frac{1}{m}\sum_{p,s}\mathbf{H}(p,s)(1 - \mathbf{H}(p,s)) \leqslant \frac{1}{m}(c_{\mathcal{P},\mathcal{S}} - \|\mathbf{H}\|_F^2) \leqslant \frac{c_{\mathcal{P},\mathcal{S}}}{m}$$

- By McDiarmid, w.p. $\geqslant 1 - \delta$: $\|\mathbf{H} - \hat{\mathbf{H}}_S\|_F \leqslant \sqrt{\frac{c_{\mathcal{P},\mathcal{S}}}{m}} + \sqrt{\frac{2c_{\mathcal{P},\mathcal{S}}}{m}\log(1/\delta)} = O(1/\sqrt{m})$

PAC Learning Stochastic WFA [BCLQ14]

Setup:

- Unknown $f : \Sigma^\star \to \mathbb{R}$ with $\mathrm{rank}(f) = n$ defining probability distribution on $\Sigma^\star$
- Data: $x^{(1)}, \ldots, x^{(m)}$ i.i.d. strings sampled from f
- Parameters: n and $\mathcal{P}, \mathcal{S}$ such that $\epsilon \in \mathcal{P} \cap \mathcal{S}$ and the sub-block $\mathbf{H} \in \mathbb{R}^{\mathcal{P} \times \mathcal{S}}$ satisfies $\mathrm{rank}(\mathbf{H}) = n$

Algorithm:

1. Estimate Hankel matrices $\hat{\mathbf{H}}$ and $\hat{\mathbf{H}}_\sigma$ for all $\sigma \in \Sigma$ using empirical probabilities

$$\hat{f}(x) = \frac{1}{m} \sum_{i=1}^{m} \mathbb{I}[x^{(i)} = x]$$

2. Return $\hat{A} = \mathrm{Spectral}(\hat{\mathbf{H}}, \{\hat{\mathbf{H}}_\sigma\}, n)$

Analysis:

- Running time is $O(|\mathcal{P} \cdot \mathcal{S}| m + |\Sigma||\mathcal{P}||\mathcal{S}| n)$
- With high probability $\sum_{|x| \leqslant L} |f(x) - \hat{A}(x)| = O\left(\frac{L^2 |\Sigma| \sqrt{n}}{\sigma_n(\mathrm{H})^2 \sqrt{m}}\right)$

Outline

research

1. Sequential Data and Weighted Automata
2. WFA Reconstruction and Approximation
3. PAC Learning for Stochastic WFA
4. Statistical Learning for WFA
5. Beyond Sequences: Transductions and Trees
6. Conclusion

Statistical Learning Framework

Motivation

- PAC learning focuses on the realizable case: the samples come from model in known class
- In practice this is unrealistic: real data is not generated from a "nice" model
- The non-realizable setting is the natural domain of statistical learning theory[2]

Setup (for strings with real labels)

- Let D be a distribution over $\Sigma^\star \times \mathbb{R}$, and $S = \{(x^i, y^i)\}$ a sample with m i.i.d. examples
- Let $\mathcal{H}$ be a hypothesis class of functions of type $\Sigma^\star \to \mathbb{R}$
- Let $\ell : \mathbb{R} \times \mathbb{R} \to \mathbb{R}_+$ be a (convex) loss function
- The goal of statistical learning theory is to use S to find $\hat{f} \in \mathcal{H}$ that approximates

$$f^* = \operatorname*{argmin}_{f \in \mathcal{H}} \mathbb{E}_{(x,y)\sim D}[\ell(f(x), y)]$$

[2] And *agnostic* PAC learning, but we will not discuss this setting here.

Empirical Risk Minimization for WFA

research

- For a large sample and a fixed $f \in \mathcal{H}$ we have

$$L_D(f;\ell) := \mathbb{E}_{(x,y)\sim D}[\ell(f(x),y)] \approx \frac{1}{m}\sum_{i=1}^{m} \ell(f(x^i),y^i) =: \hat{L}_S(f;\ell)$$

- A classical approach is consider the *empirical risk minimization* rule

$$\hat{f} = \operatorname*{argmin}_{f \in \mathcal{H}} \hat{L}_S(f;\ell)$$

- For "string to real" learning problems we want to choose a hypothesis class $\mathcal{H}$ in which
  - The ERM problem can be solved efficiently
  - We can guarantee that $\hat{f}$ will not overfit the data

Generalization Bounds and Rademacher Complexity

- The risk of overfitting can be controlled with generalization bounds of the form: for any D, with prob. $1-\delta$ over $S \sim D^m$

$$L_D(f;\ell) \leqslant \hat{L}_S(f;\ell) + C(S,\mathcal{H},\ell) \qquad \forall f \in \mathcal{H}$$

- Rademacher complexity provides bounds for any $\mathcal{H} = \{f : \Sigma^\star \to \mathbb{R}\}$

$$\mathfrak{R}_m(\mathcal{H}) = \mathbb{E}_{S\sim D^m}\mathbb{E}_\sigma\left[\sup_{f\in\mathcal{H}} \frac{1}{m}\sum_{i=1}^{m}\sigma_i f(x^i)\right] \quad \text{where} \;\; \sigma_i \sim \mathrm{unif}(\{+1,-1\})$$

- For a bounded Lipschitz loss ℓ with probability $1-\delta$ over $S \sim D^m$ (e.g. see **[MRT12]**)

$$L_D(f;\ell) \leqslant \hat{L}_S(f;\ell) + O\left(\mathfrak{R}_m(\mathcal{H}) + \sqrt{\frac{\log(1/\delta)}{m}}\right) \qquad \forall f \in \mathcal{H}$$

Bounding the Weights

- Given a pair of Hölder conjugate integers p, q $(1/p + 1/q = 1)$, define a norm on WFA given by

$$\|A\|_{p,q} = \max\left\{\|\boldsymbol{\alpha}\|_p, \|\boldsymbol{\beta}\|_q, \max_{a\in\Sigma}\|\mathbf{A}_a\|_q\right\}$$

- Let $\mathcal{A}_n \subset \mathcal{WFA}_n$ be the class of WFA with n states given by

$$\mathcal{A}_n = \{A \in \mathcal{WFA}_n \mid \|A\|_{p,q} \leqslant R\}$$

Theorem [BM15b, BM18]

The Rademacher complexity of $\mathcal{A}_n$ for $R \leqslant 1$ is bounded by

$$\mathfrak{R}_m(\mathcal{A}_n) = O\left(\frac{L_m}{m} + \sqrt{\frac{n^2|\Sigma|\log(m)}{m}}\right) \quad ,$$

where $L_m = \mathbb{E}_S[\max_i |x^i|]$.

Bounding the Language

research

- Given $p \in [1, \infty]$ and a language $f : \Sigma^\star \to \mathbb{R}$ define its p-norm as

$$\|f\|_p = \left(\sum_{x \in \Sigma^\star} |f(x)|^p \right)^{1/p}$$

- Let $\mathcal{R}_p$ be the class of languages given by

$$\mathcal{R}_p = \{ f : \Sigma^\star \to \mathbb{R} : \|f\|_p \leqslant R \}$$

Theorem [BM15b, BM18]

The Rademacher complexity of $\mathcal{R}_p$ satisfies

$$\mathfrak{R}_m(\mathcal{R}_2) = \Theta\left(\frac{R}{\sqrt{m}} \right) \quad , \qquad \mathfrak{R}_m(\mathcal{R}_1) = O\left(\frac{R C_m \sqrt{\log(m)}}{m} \right)$$

where $C_m = \mathbb{E}_S[\sqrt{\max_x |\{i : x^i = x\}|}]$.

Aside: Schatten Norms

research

- For a matrix $\mathbf{M} \in \mathbb{R}^{n \times m}$ with $\mathrm{rank}(\mathbf{M}) = k$ let $\mathfrak{s}_1 \geqslant \mathfrak{s}_2 \geqslant \cdots \geqslant \mathfrak{s}_k > 0$ be its singular values
- Arrange them in a vector $\mathfrak{s} = (\mathfrak{s}_1, \ldots, \mathfrak{s}_k)$
- For any $p \in [1, \infty]$ we define the p-Schatten norm of $\mathbf{M}$ as

$$\|\mathbf{M}\|_{\mathrm{S},p} = \|\mathfrak{s}\|_p$$

- Some of these norms have given names:
  - $p = \infty$: spectral or operator norm
  - $p = 2$: Frobenius or Hilbert–Schmidt norm
  - $p = 1$: nuclear or trace norm
- In some sense, the nuclear norm is the best convex approximation to the rank function (i.e. its convex envelope)

Bounding the Matrix

research

Given $R > 0$ and $p \geqslant 1$ define the class of infinite Hankel matrices

$$\mathcal{H}_p = \left\{\mathbf{H} \in \mathbb{R}^{\Sigma^\star \times \Sigma^\star} \;\middle|\; \mathbf{H} \in \mathrm{Hankel}, \|\mathbf{H}\|_{\mathrm{S},p} \leqslant R\right\}$$

Theorem [BM15b, BM18]

The Rademacher complexity of $\mathcal{H}_p$ satisfies

$$\mathfrak{R}_m(\mathcal{H}_2) = O\left(\frac{R}{\sqrt{m}}\right) \quad , \qquad \mathfrak{R}_m(\mathcal{H}_1) = O\left(\frac{R\log(m)\sqrt{W_m}}{m}\right) \quad ,$$

where $W_m = \mathbb{E}_S\left[\min_{\mathrm{split}(S)} \max\left\{\max_p \sum_i 1[p^i = p], \max_s \sum_i 1[s^i = s]\right\}\right]$.

Note: split(S) contains all possible prefix-suffix splits $x^i = p^i s^i$ of all strings in S

Direct Gradient-Based Methods

- The ERM problem on the class $\mathcal{A}_n$ can be solved with (stochastic) projected gradient descent:

$$\min_{A \in \mathcal{WFA}_n} \frac{1}{m} \sum_{i=1}^{m} \ell(A(x^i), y^i) \quad \text{s.t. } \|A\|_{p,q} \leqslant R$$

- Example gradient computation with $x = abca$ and weights in $\mathbf{A}_a$:

$$\begin{aligned} \nabla_{\mathbf{A}_a} \ell(A(x), y) &= \frac{\partial \ell}{\partial \hat{y}}(A(x), y) \cdot \left(\nabla_{\mathbf{A}_a} \boldsymbol{\alpha}^\top \mathbf{A}_a \mathbf{A}_b \mathbf{A}_c \mathbf{A}_a \boldsymbol{\beta}\right) \\ &= \frac{\partial \ell}{\partial \hat{y}}(A(x), y) \cdot \left(\boldsymbol{\alpha}\boldsymbol{\beta}^\top \mathbf{A}_a^\top \mathbf{A}_c^\top \mathbf{A}_b^\top + \mathbf{A}_c^\top \mathbf{A}_b^\top \mathbf{A}_a^\top \boldsymbol{\alpha}\boldsymbol{\beta}^\top\right) \end{aligned}$$

- Can solve classification ($y^i \in \{+1, -1\}$) and regression ($y^i \in \mathbb{R}$) with differentiable ℓ
- Optimization is highly non-convex – might get stuck in local optimum – but its commonly done in RNN
- Automatic differentiation can automate gradient computations

Hankel Matrix Completion [BM12]

- Learn a finite Hankel matrix over $\mathcal{P} \times \mathcal{S}$ directly from data by solving the *convex* ERM

$$\hat{\mathbf{H}} = \operatorname*{argmin}_{\mathrm{H} \in \mathbb{R}^{\mathcal{P} \times \mathcal{S}}} \frac{1}{m} \sum_{i=1}^{m} \ell(\mathbf{H}(x^i), y^i) \quad \text{s.t.} \quad \|\mathbf{H}\|_{\mathrm{S},p} \leqslant R$$

$$\left\{ \begin{array}{c} (\mathrm{bab},1),\ (\mathrm{bbb},0) \\ (\mathrm{aaa},3),\ (\mathrm{a},1) \\ (\mathrm{ab},1),\ (\mathrm{aa},2) \\ (\mathrm{aba},2),\ (\mathrm{bb},0) \end{array} \right\} \longrightarrow \begin{array}{c} \\ a \\ b \\ aa \\ ab \\ ba \\ bb \end{array} \begin{array}{c} \begin{array}{ccc} \epsilon & a & b \end{array} \\ \left[\begin{array}{ccc} 1 & 2 & 1 \\ ? & ? & 0 \\ 2 & 3 & ? \\ 1 & 2 & ? \\ ? & ? & 1 \\ 0 & ? & 0 \end{array} \right] \end{array}$$

- Recover a WFA from $\hat{\mathbf{H}}$ using the spectral reconstruction algorithm
- Rademacher complexity of $\mathcal{H}_p$ and algorithmic stability [BM12] can be used to guarantee generalization

Outline

Sequence-to-Sequence Modelling in NLP and RL

research

- Many NLP applications involve pairs of input-output sequences:
  - Sequence tagging (one output tag per input token) e.g.: part of speech tagging

input:	Ms.	Haag	plays	Elianti
output:	NNP	NNP	VBZ	NNP

  - Transductions (sequence lenghts might differ) e.g.: spelling correction

    input: a p l e

    output: a p p l e
- Sequence-to-sequence models also arise naturally in RL:
  - An agent operating in an MPD or POMDP enviroment collects traces of the form

input (actions):	a_1	a_2	a_3	$\cdots$
output (observation, rewards):	(o_1, r_1)	(o_2, r_2)	(o_3, r_3)	$\cdots$
- For these applications we want to learn functions of the form $f : (\Sigma \times \Delta)^\star \to \mathbb{R}$ or more generally $f : \Sigma^\star \times \Delta^\star \to \mathbb{R}$ (can model using ϵ-transitions)

Learning Transducers with Hankel Matrices

- Given input and output alphabets Σ and Δ we can define IO-WFA[3] as

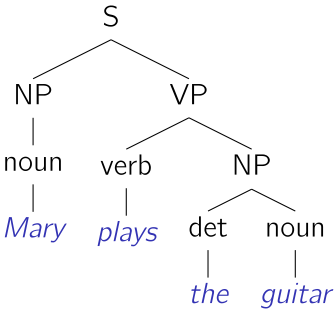
$$A = \langle \boldsymbol{\alpha}, \boldsymbol{\beta}, \{\mathbf{A}_{\sigma,\delta}\} \rangle$$

- The language computed by a IO-WFA can have diverse interpretations, for $(x, y) \in (\Sigma \times \Delta)^\star$:
  - Tagging: $f(x, y) =$ compatiblity score of output y on input x
  - Dynamics modelling: $f(x, y) = \mathbb{P}[y|x]$, probability of observations given outputs
  - Reward modelling: $f(x, y) = \mathbb{E}[r_1 + \cdots + r_t]$, expected reward from action-observation sequence
- The Hankel trick applies to this setting as well with $\mathbf{H}_f \in \mathbb{R}^{(\Sigma\times\Delta)^\star \times (\Sigma\times\Delta)^\star}$
- For applications and concrete algorithms see **[BSG09, BQC11, QBCG14, BM17]**

[3] Other nomenclatures: weighted finite state transition (WFST), predictive state representation (PSR), input-output observable operator model (IO-OOM)

Trees in NLP

- Parsing tasks in NLP require predicting a tree for a sequence: modelling dependencies inside a sentence, document, etc



- Models on trees are also useful to learn more complicated languages: weighted context-free languages (instead of regular)
- Applications involve different types of models and levels of supervision
  - Labelled trees, unlabelled trees, yields, etc.

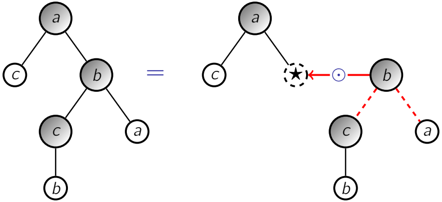
Weighted Tree Automata (WTA)

- Take a ranked alphabet $\Sigma = \Sigma_0 \cup \Sigma_1 \cup \cdots$
- A weighted tree automaton with n states is a tuple $A = \langle \boldsymbol{\alpha}, \{\mathbf{T}_\tau\}_{\tau \in \Sigma_{\geqslant 1}}, \{\boldsymbol{\beta}_\sigma\}_{\sigma \in \Sigma_0} \rangle$ where

$$\boldsymbol{\alpha}, \boldsymbol{\beta}_\sigma \in \mathbb{R}^n \qquad \mathbf{T}_\tau \in (\mathbb{R}^n)^{\otimes \mathrm{rk}(\tau)+1}$$

- A defines a function $f_A = \mathrm{Trees}_\Sigma \to \mathbb{R}$ through recursive vector-tensor contractions
- Similar expressive power as WCFG and L-WCFG

Inside-Outside Factorization in WTA



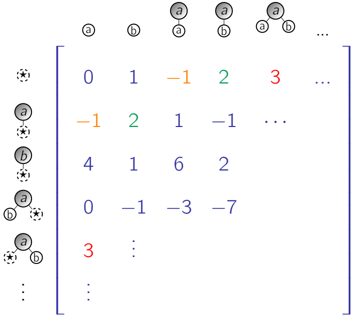
For any inside-outside decomposition of a tree:

$$
\begin{aligned}
f(t) &= \boldsymbol{\alpha}_{t_o}^{\top}\boldsymbol{\beta}_{t_i} && (\text{let } t = t_o[t_i]) \\
&= \boldsymbol{\alpha}_{t_o}^{\top}\mathbf{T}_{\sigma}(\boldsymbol{\beta}_{t_1}, \boldsymbol{\beta}_{t_2}) && (\text{let } t_i = \sigma(t_1, t_2)) \\
&= \boldsymbol{\alpha}_{t_o}^{\top}\mathbf{T}_{\sigma}^{(2)}(\boldsymbol{\beta}_{t_1} \otimes \boldsymbol{\beta}_{t_2}) && (\text{flatten tensor})
\end{aligned}
$$

Learning WTA with Hankel Matrices

There exist analogues of:

- The Hankel matrix for $f : \mathrm{Trees}_\Sigma \to \mathbb{R}$ corresponding to inside-outside decompositions



- The Fliess–Kronecker theorem **[BLB83]**
- The spectral learning algorithm **[BHD10]** and variants thereof **[CSC+12, CSC+13, CSC+14]**

Outline

1. Sequential Data and Weighted Automata
2. WFA Reconstruction and Approximation
3. PAC Learning for Stochastic WFA
4. Statistical Learning for WFA
5. Beyond Sequences: Transductions and Trees
6. Conclusion

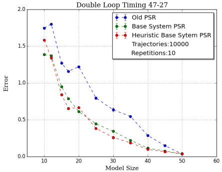
And It Works Too!

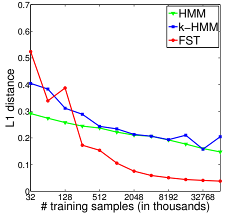
Spectral methods are competitive against traditional methods:

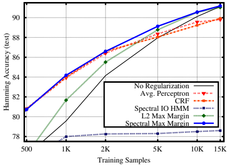
- Expectation maximization
- Conditional random fields
- Tensor decompositions

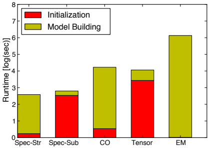
In a variety of problems:

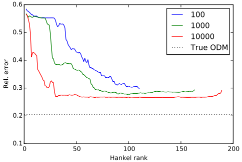
- Sequence tagging
- Constituency and dependency parsing
- Timing and geometry learning
- POS-level language modelling

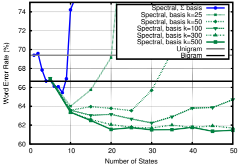


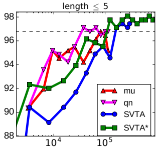


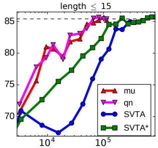


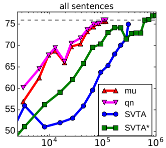












Open Problems and Current Trends

- Optimal selection of $\mathcal{P}$ and $\mathcal{S}$ from data
- Scalable convex optimization over sets of Hankel matrices
- Constraining the output WFA (eg. probabilistic automata)
- Relations between learning and approximate minimisation
- How much of this can be extended to WFA over semi-rings?
- Spectral methods for initializing non-convex gradient-based learning algorithms

Conclusion

Take home points

- A single building block based on SVD of Hankel matrices
- Implementation only requires linear algebra
- Analysis involves linear algebra, probability, convex optimization
- Can be made practical for a variety of models and applications

Want to know more?

- EMNLP'14 tutorial (with slides, video, and code)
  `https://borjaballe.github.io/emnlp14-tutorial/`
- Survey papers **[BM15a, TJ15]**
- Python toolkit Sp2Learn **[ABDE16]**
- Neighbouring literature: Predictive state representations (PSR) **[LSS02]** and Observable operator models (OOM) **[Jae00]**

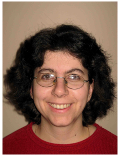
Thanks To All My Collaborators!

Xavier Carreras

Mehryar Mohri

Prakash Panangaden

Joelle Pineau



Doina Precup

Ariadna Quattoni

- Guillaume Rabusseau
- Franco M. Luque
- Pierre-Luc Bacon
- Pascale Gourdeau
- Odalric-Ambrym Maillard
- Will Hamilton
- Lucas Langer
- Shay Cohen
- Amir Globerson

References I

D. Arrivault, D. Benielli, F. Denis, and R. Eyraud.
Sp2learn: A toolbox for the spectral learning of weighted automata.
In *ICGI*, 2016.

B. Balle.
Learning Finite-State Machines: Algorithmic and Statistical Aspects.
PhD thesis, Universitat Politècnica de Catalunya, 2013.

B. Balle, X. Carreras, F.M. Luque, and A. Quattoni.
Spectral learning of weighted automata: A forward-backward perspective.
Machine Learning, 2014.

R. Bailly, F. Denis, and L. Ralaivola.
Grammatical inference as a principal component analysis problem.
In *ICML*, 2009.

References II

R. Bailly, A. Habrard, and F. Denis.
A spectral approach for probabilistic grammatical inference on trees.
In *ALT*, 2010.

Symeon Bozapalidis and Olympia Louscou-Bozapalidou.
The rank of a formal tree power series.
Theoretical Computer Science, 27(1-2):211–215, 1983.

B. Balle and M. Mohri.
Spectral learning of general weighted automata via constrained matrix completion.
In *NIPS*, 2012.

B. Balle and M. Mohri.
Learning weighted automata (invited paper).
In *CAI*, 2015.

B. Balle and M. Mohri.
On the rademacher complexity of weighted automata.
In *ALT*, 2015.

References III

B. Balle and O.-A. Maillard.
Spectral learning from a single trajectory under finite-state policies.
In *ICML*, 2017.

B. Balle and M. Mohri.
Generalization Bounds for Learning Weighted Automata.
Theoretical Computer Science, 716:89–106, 2018.

B. Balle, A. Quattoni, and X. Carreras.
A spectral learning algorithm for finite state transducers.
In *ECML-PKDD*, 2011.

B. Boots, S. Siddiqi, and G. Gordon.
Closing the learning-planning loop with predictive state representations.
In *Proceedings of Robotics: Science and Systems VI*, 2009.

S. B. Cohen, K. Stratos, M. Collins, D. P. Foster, and L. Ungar.
Spectral learning of latent-variable PCFGs.
In *ACL*, 2012.

References IV

S. B. Cohen, K. Stratos, M. Collins, D. P. Foster, and L. Ungar.
Experiments with spectral learning of latent-variable PCFGs.
In *NAACL-HLT*, 2013.

S. B. Cohen, K. Stratos, M. Collins, D. P. Foster, and L. Ungar.
Spectral learning of latent-variable PCFGs: Algorithms and sample complexity.
Journal of Machine Learning Research, 2014.

François Denis, Mattias Gybels, and Amaury Habrard.
Dimension-free concentration bounds on hankel matrices for spectral learning.
Journal of Machine Learning Research, 17:31:1–31:32, 2016.

M. Fliess.
Matrices de Hankel.
Journal de Mathématiques Pures et Appliquées, 1974.

References V

D. Hsu, S. M. Kakade, and T. Zhang.
A spectral algorithm for learning hidden Markov models.
In *COLT*, 2009.

H. Jaeger.
Observable operator models for discrete stochastic time series.
Neural Computation, 2000.

M. Littman, R. S. Sutton, and S. Singh.
Predictive representations of state.
In *NIPS*, 2002.

Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar.
Foundations of machine learning.
MIT press, 2012.

A. Quattoni, B. Balle, X. Carreras, and A. Globerson.
Spectral regularization for max-margin sequence tagging.
In *ICML*, 2014.

References VI

M. R. Thon and H. Jaeger.

Links between multiplicity automata, observable operator models and predictive state representations: a unified learning framework.

Journal of Machine Learning Research, 2015.

Automata Learning

Borja Balle

Amazon Research Cambridge[4]

Foundations of Programming Summer School (Oxford) — July 2018

[4] Based on work completed before joining Amazon