Spectral Learning of Latent Variable Models and its Interpretation as an Optimization Problem

– a CSML pizza talk –

Gabi Teodoru
joint work with Maneesh Sahani

Special thanks (in no particular order):

Jeff Beck, Lars Büsing, Bharath Sriperumbudur, Byron Boots, YeeWhye Teh and Arthur Gretton

Outline

1. Spectral learning
   - Previous work
   - Optimization interpretation
   - Re-deriving some previous algorithms
2. Spectral learning for Latent Markov Lattices
   - The model
   - Learning: Three tricks
3. Future work

What is Spectral Learning?

- Spectral Learning is a learning algorithm for latent variable graphical models
  - More details coming up in a few slides
- Pros:
  - Consistent estimator of parameters; computable
  - Quick, one-shot learning
- Cons:
  - Statistically inefficient
  - Estimates a function of the parameters; additional steps are needed to recover the parameters
  - Numerical instability

Previous work

- Spectral algorithms:
  - Spectral learning for HMMs (Hsu et al, 2009)
  - Reduced-rank HMMs (Siddiqi et al, 2010)
  - Hilbert space embeddings of HMMs (Song et al, 2010)
  - Spectral learning of latent tree graphical models (Parikh et al, 2011)
- Related algorithms and ideas:
  - Subspace identification (SSID) for Linear Dynamical Systems (LDS) [aka Kalman Filter]
  - Observable Operator Models (OOM) / Predictive state representations (PSR)

Optimization interpretation: Motivation

- What previous work showed:
  - In the limit of infinite data, the algorithm identifies the true parameters
  - Finite sample bounds relate the finite sample size to prediction errors
- What previous work authors' claimed in talks:
  - Algorithm is convex, and therefore does not get stuck in local optima (unlike EM)
- What I wanted to see:
  - The cost function
    - (… they didn't have one)

Optimization interpretation

There is a common optimization interpretation for all previously mentioned spectral algorithms and SSID.

Pros of optimization interpretation

- Allows improving the cost functions
  - Adding regularizers
  - Using the Generalized Method of Moments
- Provides a way to use the data more efficiently
- Generalizes the method to other models

Constructing the optimization interpretation

- Step 1: Given a latent variable model parametrized by θ, pick a transformation of these parameters $\phi = \phi(\theta)$
- Step 2: Choose two sets of model statistics

$$\mathbb{E}_{\phi}\left[f_1(\mathbf{X}) \right] \qquad \mathbb{E}_{\phi}\left[f_2(\mathbf{X}) \right]$$

- Step 3: Choose two functions

$$g_1(\bullet, \phi) \qquad g_2(\bullet, \phi)$$

such that

$$g_1\left(\mathbb{E}_{\phi}\left[f_1(\mathbf{X}) \right], \phi\right) = g_2\left(\mathbb{E}_{\phi}\left[f_1(\mathbf{X}) \right], \phi\right) \forall \phi \in \Phi$$

Examples

$$g_1\left(\mathbb{E}_\phi\left[f_1(\mathbf{X})\right],\phi\right) = g_2\left(\mathbb{E}_\phi\left[f_1(\mathbf{X})\right],\phi\right) \forall \phi \in \Phi$$

- LDS: $f_1 = E\left[X_3 X_1^{\mathbf{T}}\right]$ $f_2 = E\left[X_2 X_1^{\mathbf{T}}\right]$
  $g_1(f_1,\phi) = f_1$ $g_2(f_2,\phi) = \mathbf{\Phi} f_2$
- HMM: $f_1 = P(X_{3..1})$ $f_2 = P(X_{2..1})$
  $g_1(f_1,\phi) = f_1$ $g_2(f_2,\phi) = \mathbf{\Phi} f_2$
- SCFG: $f_1 = P(X_{1..3})$ $f_2 = \left[P(X_{1..2}), P(X_{2..3})\right]$
  $g_1(f_1,\phi) = f_1$ $g_2(f_2,\phi) = \sum_k \left[f_2\right]_{i,k} \phi_k \delta_k$

The cost function

- Since

$$g_1\left(\mathbb{E}_\phi\left[f_1(\mathbf{X})\right],\phi\right) = g_2\left(\mathbb{E}_\phi\left[f_1(\mathbf{X})\right],\phi\right)\ \forall \phi \in \Phi$$

if we have finite samples from the model,

$$g_1\left(f_1(\mathbf{x}),\phi\right) \approx g_2\left(f_2(\mathbf{x}),\phi\right)$$

the dumbest thing we could possibly do is:

$$\min_\phi d\left(g_1\left(f_1(\mathbf{x}),\phi\right), g_2\left(f_2(\mathbf{x}),\phi\right)\right)$$

If the functions g_1, g_2 are linear in the 2nd parameter, and the distance is quadratic, then we have a convex optimization!

Adding bells and whistles

- Original cost function:

$$C(\mathbf{x},\phi) = d\left(g_1\left(f_1(\mathbf{x}),\phi\right), g_2\left(f_2(\mathbf{x}),\phi\right)\right)$$

- With regularization:

$$C(\mathbf{x},\phi) = d\left(g_1, g_2\right) + \Omega(\phi)$$

- Generalized Method of Moments:

$$C(\mathbf{x},\phi) = \left(g_1 - g_2\right)^{\mathbf{T}} \mathbf{W} \left(g_1 - g_2\right)$$

GMM iteratively re-estimates

$$\mathbf{W} = \left(g_1 - g_2\right)\left(g_1 - g_2\right)^{\mathbf{T}}$$

Example 1: SSID for LDS

- Hidden RVs $H_t \in \mathbb{R}^m$, observables $X_t \in \mathbb{R}^n$, $\mathrm{m} \leq \mathrm{n}$
- Distribution: $H_t, X_t \sim \mathrm{Normal}$
  - $H_t = \mathbf{T}H_{t-1} + \mathrm{noise} \leftrightarrow E[H_t] = \mathbf{T}E[H_{t-1}]$
  - $X_t = \mathbf{O}H_t + \mathrm{noise} \leftrightarrow E[X_t] = \mathbf{O}E[H_t]$
  - Assume $\mathbf{T}_{m\times m}$ and $\mathbf{O}_{n\times m}$ are full rank
- Special relationship and spectral coefficients:
  - $E[X_{t+1}] = \mathbf{O}\mathbf{T}E[H_t]$
  - $E[H_t] = \mathbf{O}^{\dagger}E[X_t]$

  $\Big\} \rightarrow E[X_{t+1}] = \mathbf{O}\mathbf{T}\mathbf{O}^{\dagger}E[X_t]$
  - $E[X_{t+1}] = \mathbf{\Phi}E[X_t] \quad \leftrightarrow \quad \mathbf{\Phi} = \mathbf{O}\mathbf{T}\mathbf{O}^{\dagger}$
- Cost function: $\left\| E[X_{t+1}] - \mathbf{\Phi}E[X_t] \right\|_2^2$

Example 1: SSID for LDS

- Minimization: $\boldsymbol{\Phi}^* = \arg\min_{\boldsymbol{\Phi}} \left\| E\left[X_{t+1}\right] - \boldsymbol{\Phi} E\left[X_t\right] \right\|_F$
- Cost function is underdetermined!
  - Solution 1: $\boldsymbol{\Phi}_t^* = \arg\min_{\boldsymbol{\Phi}_t} \left\| E\left[X_{t+1} X_{t-1}^{\mathbf{T}}\right] - \boldsymbol{\Phi}_t E\left[X_t X_{t-1}^{\mathbf{T}}\right] \right\|_F$
  - Solution 2: $\boldsymbol{\Phi}^* = \arg\min_{\boldsymbol{\Phi}} \sum_t \left\| E\left[X_{t+1}\right] - \boldsymbol{\Phi} E\left[X_t\right] \right\|_F$
    - May be expressed also by horizontal concatenation:

$$\boldsymbol{\Phi}^* = \arg\min_{\boldsymbol{\Phi}} d\left(\left[E[X_t]\ E[X_{t+1}]\ E[X_{t+2}]\ldots\right], \boldsymbol{\Phi}\left[E[X_{t-1}]\ E[X_t]\ E[X_{t+1}]\ldots\right]\right)$$

Example 1: SSID for LDS

- How are the LDS spectral parameters useful?
  - Parameters are identifiable up to a transformation:

$$E[H_t] = \mathbf{T}E[H_{t-1}]$$
$$E[X_t] = \mathbf{O}E[H_t]$$

Example 1: SSID for LDS

- How are the LDS spectral parameters useful?
  - Parameters are identifiable up to a transformation:

$$E\left[\mathbf{S}H_t\right] = \mathbf{S}\mathbf{T}\mathbf{S}^{-1}E\left[\mathbf{S}H_{t-1}\right]$$
$$E\left[X_t\right] = \mathbf{O}\mathbf{S}^{-1}E\left[\mathbf{S}H_t\right]$$

Example 1: SSID for LDS

- How are the LDS spectral parameters useful?
  - Parameters are identifiable up to a transformation:

$$E\left[\mathbf{S}H_t\right] = \mathbf{S}\mathbf{T}\mathbf{S}^{\dagger}E\left[\mathbf{S}H_{t-1}\right]$$
$$E\left[X_t\right] = \mathbf{O}\mathbf{S}^{\dagger}E\left[\mathbf{S}H_t\right]$$

Example 1: SSID for LDS

- How are the LDS spectral parameters useful?
  - Parameters are identifiable up to a transformation:

$$\left.\begin{aligned} E\left[\mathbf{S}H_t\right] &= \mathbf{S}\mathbf{T}\mathbf{S}^{\dagger}E\left[\mathbf{S}H_{t-1}\right] \\ E\left[X_t\right] &= \mathbf{O}\mathbf{S}^{\dagger}E\left[\mathbf{S}H_t\right] \end{aligned}\right\} \rightarrow \begin{cases} E\left[H'_t\right] = \mathbf{S}\mathbf{T}\mathbf{S}^{\dagger}E\left[H'_{t-1}\right] \\ E\left[X_t\right] = \mathbf{O}\mathbf{S}^{\dagger}E\left[H'_t\right] \end{cases}$$

- So what were the spectral coefficients we learned?

$$\begin{aligned} \boldsymbol{\Phi} &= \mathbf{O}\mathbf{T}\mathbf{O}^{\dagger} & \boldsymbol{\Phi}^{\dagger} &= \mathbf{O}\mathbf{T}^{-1}\mathbf{O}^{\dagger} \\ \boldsymbol{\Phi} &= \mathbf{O}\mathbf{T}\left(\mathbf{T}\mathbf{T}^{-1}\right)\mathbf{O}^{\dagger} & \boldsymbol{\Phi}^{\dagger} &= \mathbf{O}\left(\mathbf{O}\mathbf{T}\right)^{\dagger} \\ \boldsymbol{\Phi} &= \left(\mathbf{O}\mathbf{T}\right)\mathbf{T}\left(\mathbf{O}\mathbf{T}\right)^{\dagger} \end{aligned}$$

- The spectral coefficients matrix and its pseudoinverse provide the corresponding transformations of the model parameters

Example 1: SSID for LDS

- Hidden RVs $H_t \in \mathbb{R}^m$, observables $X_t \in \mathbb{R}^n$, $\mathrm{m} \leq \mathrm{n}$
- Distribution: $H_t, X_t \sim \mathrm{Normal}$
  - $H_t = \mathbf{T}H_{t-1} + \mathrm{noise} \leftrightarrow E[H_t] = \mathbf{T}E[H_{t-1}]$
  - $X_t = \mathbf{O}H_t + \mathrm{noise} \leftrightarrow E[X_t] = \mathbf{O}E[H_t]$
  - Assume $\mathbf{T}_{m\times m}$ and $\mathbf{O}_{n\times m}$ are full rank
- Special relationship and spectral coefficients:
  - $E[X_{t+1}] = \mathbf{O}\mathbf{T}E[H_t]$
  - $E[H_t] = \mathbf{O}^{\dagger}E[X_t]$

  $\Big\} \rightarrow E[X_{t+1}] = \mathbf{O}\mathbf{T}\mathbf{O}^{\dagger}E[X_t]$
  - $E[X_{t+1}] = \mathbf{\Phi}E[X_t] \quad \leftrightarrow \quad \mathbf{\Phi} = \mathbf{O}\mathbf{T}\mathbf{O}^{\dagger}$
- Cost function: $\left\| E[X_{t+1}] - \mathbf{\Phi}E[X_t] \right\|_F$

Adding the SVD

- Hidden RVs $H_t \in \mathbb{R}^m$, observables $X_t \in \mathbb{R}^n$, $\mathrm{m} \leq \mathrm{n}$
- Distribution: $H_t, X_t \sim \mathrm{Normal}$
  - $H_t = \mathbf{T}H_{t-1} + \mathrm{noise} \leftrightarrow E[H_t] = \mathbf{T}E[H_{t-1}]$
  - $X_t = \mathbf{O}H_t + \mathrm{noise} \leftrightarrow E[X_t] = \mathbf{O}E[H_t]$
  - Assume $\mathbf{T}_{m\times m}$ and $\mathbf{O}_{n\times m}$ are full rank
- $\mathbf{\Phi} = \mathbf{OTO}^{\dagger}$ implies that $\mathbf{\Phi}$ has rank m
- Make this a constraint of the optimization:

$$\begin{cases} \min_{\mathbf{\Phi}} d\left(E[X_{t+1}], \mathbf{\Phi}E[X_t]\right) \\ \text{s.t. } \mathrm{rank}\left(\mathbf{\Phi}\right) \leq m \end{cases}$$

Adding the SVD

$$\begin{cases} \min\limits_{\mathbf{\Phi}} \left\| E\left[X_{t+1}\right] - \mathbf{\Phi} E\left[X_t\right] \right\|_F \\ \text{s.t. rank}\left(\mathbf{\Phi}\right) \leq m \end{cases}$$

is equivalent to:

$$\min_{\mathbf{\Phi}} \left\| \mathbf{U}^{\mathbf{T}} E\left[X_{t+1}\right] - \mathbf{\Phi}\mathbf{U}^{\mathbf{T}} E\left[X_t\right] \right\|_F$$

where $\mathbf{U}$ is the matrix of the first m-left singular vectors of

$$E\left[X_t X_{t-1}^{\mathbf{T}}\right] \quad \text{or} \quad \left[E\left[X_t\right]\ E\left[X_{t+1}\right]\ E\left[X_{t+2}\right]\ldots\right]$$

Example 2: Spectral Learning of HMMs

- Hidden $H_t \in \{1..m\}$,observables $X_t \in \{1..n\}$, $\mathrm{m} \leq \mathrm{n}$
- Parameters: $\mathbf{T}$, $\mathbf{O}$
  - $\mathbf{p}_{H_{t+1}} = \mathbf{T}\mathbf{p}_{H_t}$
  - $\mathbf{p}_{X_t} = \mathbf{O}\mathbf{p}_{H_t}$
  - Assume $\mathbf{T}_{m\times m}$ and $\mathbf{O}_{n\times m}$ are full rank
- Special relationship and spectral coefficients:
  - $\left.\begin{array}{l} \mathbf{p}_{X_{t+1}} = \mathbf{O}\mathbf{T}\mathbf{p}_{H_t} \\ \mathbf{p}_{H_t} = \mathbf{O}^{\dagger}\mathbf{p}_{X_t} \end{array}\right\} \rightarrow \mathbf{p}_{X_{t+1}} = \mathbf{O}\mathbf{T}\mathbf{O}^{\dagger}\mathbf{p}_{X_t}$
  - $\mathbf{p}_{X_{t+1}} = \mathbf{\Phi}\mathbf{p}_{X_t} \quad \leftrightarrow \quad \mathbf{\Phi} = \mathbf{O}\mathbf{T}\mathbf{O}^{\dagger}$
- Spectral coefficients are a transform of a conditional probability:
  - Useless for computing the joint distribution $p(x_{1..t})$

Example 2: Spectral Learning of HMMs

- Use probability matrices: $\left[\mathbf{P}_{X,Y}\right]_{x,y} = P(X = x, Y = y)$
- We had from the previous slide:
  - $\mathbf{p}_{H_{t+1}} = \mathbf{T}\mathbf{p}_{H_t}$
  - $\mathbf{p}_{X_t} = \mathbf{O}\mathbf{p}_{H_t}$
  - Assume $\mathbf{T}_{m\times m}$ and $\mathbf{O}_{n\times m}$ are full rank, with $m \le n$
- Special relationship and spectral coefficients:
  - $\left.\begin{array}{l} \mathbf{p}_{X_{t+1}} = \mathbf{O}\mathbf{T}\mathbf{p}_{H_t} \\ \mathbf{p}_{H_t} = \mathbf{O}^{\dagger}\mathbf{p}_{X_t} \end{array}\right\} \rightarrow \mathbf{p}_{X_{t+1}} = \mathbf{O}\mathbf{T}\mathbf{O}^{\dagger}\mathbf{p}_{X_t}$
  - $\mathbf{p}_{X_{t+1}} = \mathbf{\Phi}\mathbf{p}_{X_t} \quad \leftrightarrow \quad \mathbf{\Phi} = \mathbf{O}\mathbf{T}\mathbf{O}^{\dagger}$

Example 2: Spectral Learning of HMMs

- Use probability matrices: $\left[\mathbf{P}_{X,Y}\right]_{x,y} = P(X = x, Y = y)$
- We had from the previous slide:
  - $\mathbf{p}_{H_{t+1}} = \mathbf{T}\mathbf{p}_{H_t}$
  - $\mathbf{p}_{X_t} = \mathbf{O}\mathbf{p}_{H_t}$
  - Assume $\mathbf{T}_{m\times m}$ and $\mathbf{O}_{n\times m}$ are full rank, with $m \le n$
- Special relationship and spectral coefficients:
  - $\left[\mathbf{P}_{X_{t+1},X_t}\right]_{\bullet,x_t} = \mathbf{O}\mathbf{T}\,\mathrm{diag}\left([\mathbf{O}]_{x_t,\cdot}\right)\mathbf{p}_{H_t}$
  - $\mathbf{p}_{H_t} = \mathbf{O}^{\dagger}\mathbf{p}_{X_t}$

    $\Big\} \rightarrow \left[\mathbf{P}_{X_{t+1},X_t}\right]_{\bullet,x_t} = \mathbf{O}\mathbf{T}\,\mathrm{diag}\left([\mathbf{O}]_{x_t,\cdot}\right)\mathbf{O}^{\dagger}\mathbf{p}_{X_t}$
  - $\left[\mathbf{P}_{X_{t+1},X_t}\right]_{\bullet,x_t} = \mathbf{\Phi}_{x_t}\mathbf{p}_{X_t} \quad \leftrightarrow \quad \mathbf{\Phi}_{x_t} = \mathbf{O}\mathbf{T}\,\mathrm{diag}\left[\mathbf{O}_{x_t,\cdot}\right]\mathbf{O}^{\dagger}$
- This can be used to compute the joint, since:

$$\left[\mathbf{P}_{X_{t+1},X_t,X_{\text{past}}}\right]_{\bullet,x_t,\mathbf{x}_{\text{past}}} = \mathbf{\Phi}_{x_t}\left[\mathbf{P}_{X_t,X_{\text{past}}}\right]_{\bullet,\mathbf{x}_{\text{past}}}$$

Example 2: Spectral Learning of HMMs

$$\begin{cases} \min\limits_{\mathbf{\Phi}_{x_t}} \left\| \left[\mathbf{P}_{X_{t+1},X_t,X_{\text{past}}} \right]_{\bullet,x_t,\mathbf{x}_{\text{past}}} - \mathbf{\Phi}_{x_t} \left[\mathbf{P}_{X_t,X_{\text{past}}} \right]_{\bullet,\mathbf{x}_{\text{past}}} \right\|_F \\ \text{s.t. } \operatorname{rank}\left(\mathbf{\Phi}_{x_t} \right) \le m \end{cases}$$

Spectral learning for other models

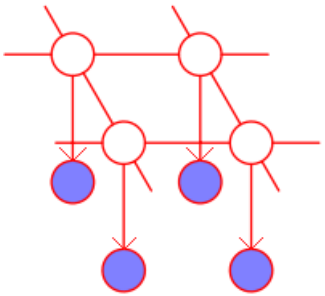
$$C(\mathbf{x},\phi) = d\left(g_1\left(f_1(\mathbf{x}),\phi\right), g_2\left(f_2(\mathbf{x}),\phi\right)\right)$$

- $f_{1,2}(\mathbf{x})$ are Hilbert space embeddings
  - Hilbert space embeddings of HMMs (Song et al, 2010)
- $f_{1,2}(\mathbf{x})$ are certain sets of joint probabilities in a tree chosen such that linear relationships hold
  - Spectral learning of latent tree graphical models (Parikh et al, '11)

- Other models on which I've tried spectral learning:
  - PCFGs (can only learn from strings of length $1..3$)
  - Finite State Transducers (adds an extra layer of hidden variables; optimization becomes bilinear least squares)
  - Markov Random Fields (that's the segue…)

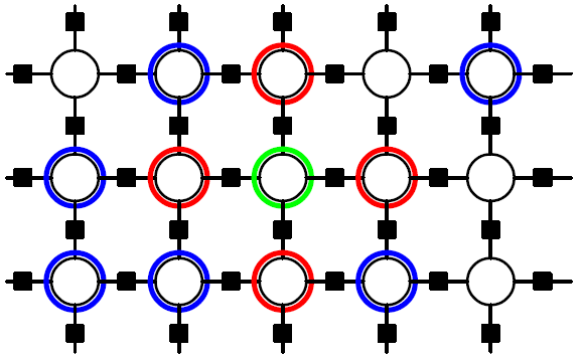
MRFs and Latent Markov Lattices

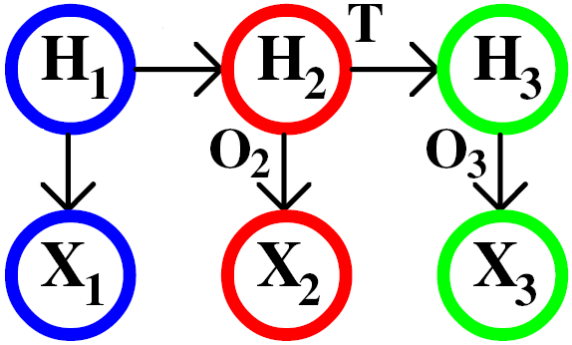
- Hidden $H_i \in \{1..m\}$, observables $X_i \in \{1..n\}$, $\mathrm{m} \le \mathrm{n}$
- Distribution: $P(\mathbf{x}) \propto \sum_{\mathbf{h}} \left(\prod_i P(x_i \mid h_i) \right) \left(\prod_{(i,j) \in \text{edges}} \theta_{h_i, h_j} \right)$
- Parameters:
  - $\theta_{ab} = \theta_{ba} \ge 0$
  - $P(X_i = a \mid H_i = b) = [\mathbf{O}]_{ab}$. $\mathbf{O}$ is n-by-m and full rank
- For concreteness, we'll look at 2D square lattices:



Learning LMLs: Trick #1: HMM equivalence

- Construct a new set of variables:
  - $\mathbb{H}_3 = H_i, \mathbb{X}_3 = X_i$
  - $\mathbb{H}_2 = \text{Markov Boundary}(H_i), \mathbb{X}_2 = \text{child}(\mathbb{H}_2)$
  - $\mathbb{H}_1 = H_\sigma, \mathbb{H}_1 \cap (\mathbb{H}_2 \cup \mathbb{H}_3) = \varnothing, \mathbb{X}_1 = X_\sigma$
- This is a Hidden Markov Model





- Parameters:
  - $\mathbf{O}_3 = \mathbf{O}$
  - $[\mathbf{O}_2]_{\sigma,\tau} = P(\mathbb{X}_2 = \sigma \mid \mathbb{H}_2 = \tau), \mathbf{O}_{2_{n^4 \times m^4}}$
    $= P(X_{2,1} = \sigma_1, X_{2,2} = \sigma_2, \ldots \mid H_{2,1} = \tau_1, H_{2,2} = \tau_2, \ldots)$
  - $[\mathbf{T}]_{\sigma,i} = P(\mathbb{H}_3 = i \mid \mathbb{H}_2 = \tau)$. Assume $\mathbf{T}_{m \times m^4}$ is full rank

Learning LMLs: Trick #2: Tensor re-slicing

- Solve:
  - $\min_{\mathbf{\Phi}_{\mathbf{x}_2}} d\left(\left[\mathbf{P}_{\mathbb{X}_3,\mathbb{X}_2,\mathbb{X}_1}\right]_{\bullet,\mathbf{x}_2,\bullet}, \mathbf{\Phi}_{\mathbf{x}_2}\left[\mathbf{P}_{\mathbb{X}_2,\mathbb{X}_1}\right]_{\bullet,\bullet}\right)$ s.t. $\operatorname{rank}\left(\mathbf{\Phi}_{\mathbf{x}_2}\right) \leq m$
  - $\mathbf{\Phi}_{\mathbf{x}_2} = \mathbf{O}_3 \mathbf{T} \operatorname{diag}\left(\left[\mathbf{O}_2\right]_{\mathbf{x}_2,\bullet}\right) \mathbf{O}_2^{\dagger}$
  - This spectral parameterization is useless!
- Solve:
  - $\min_{\mathbf{\Phi}_{\mathbf{x}_3}} d\left(\left[\mathbf{P}_{\mathbb{X}_3,\mathbb{X}_2,\mathbb{X}_1}\right]_{\mathbf{x}_3,\bullet,\bullet}, \mathbf{\Phi}_{\mathbf{x}_3}\left[\mathbf{P}_{\mathbb{X}_2,\mathbb{X}_1}\right]_{\bullet,\bullet}\right)$ s.t. $\operatorname{rank}\left(\mathbf{\Phi}_{\mathbf{x}_3}\right) \leq m$
  - $\mathbf{\Phi}_{\mathbf{x}_3} = \mathbf{O}_2 \operatorname{diag}\left(\left[\mathbf{O}_3\right]_{\mathbf{x}_3,\bullet} \mathbf{T}\right) \mathbf{O}_2^{\dagger}$
  - This is somewhat more useful…
    - Simply a re-slicing of the tensor above

Learning LMLs: Trick #3: Eigendecomposition

- Warning: Hand-wavy math
- The eigenvectors of the spectral parameters:

$$\mathbf{\Phi}_{\mathbf{x}_3} = \mathbf{O}_2 \operatorname{diag}\left(\left[\mathbf{O}_3\right]_{\mathbf{x}_3,\bullet} \mathbf{T}\right) \mathbf{O}_2^{\dagger}$$

  are the columns of $\mathbf{O}_2$, up to scaling

  – due to model symmetries, the matrix $\mathbf{\Phi}_{\mathbf{x}_3}$ is guaranteed to have repeated eigenvalues

  – also due to model symmetries: the eigenvectors we can recuperate are sufficient to identify the model

Computing the MRF model parameters

- Compute $\mathbf{O} = \mathbf{O}_3$ using its relationships to $\mathbf{O}_2$:
  - $[\mathbf{O}_2]_{\sigma,\tau} = \prod_i [\mathbf{O}]_{\sigma_i,\tau_i}$ or $[\mathbf{O}_2]_{[i,i,i,i],[j,j,j,j]} = \left([\mathbf{O}]_{i,j}\right)^4$
  - $\sum_{\sigma:\sigma_i=a} [\mathbf{O}_2]_{\sigma,\tau} = [\mathbf{O}]_{a,\tau_i}$
- May be computed by multiple methods:
  - minimize least squares in log space (linear)
  - exponentiation
  - marginalization
  - minimize least squares
- Compute then the remaining unknown eigenvectors of $\mathbf{O}_2$ from the now known $\mathbf{O}$

Computing the MRF model parameters

- Using the original slicing of the spectral parameters, compute $\mathbf{T}$:

$$\mathbf{\Phi}_{\mathbf{x}_2} = \mathbf{O}_3 \mathbf{T} \operatorname{diag}\left([\mathbf{O}_2]_{\mathbf{x}_2,\bullet} \right) \mathbf{O}_2^{\dagger}$$

- Compute θ_{ab} (up to proportionality):

$$[\mathbf{T}]_{i,\sigma} = Z_\sigma^{-1} \prod_j \theta_{i,\sigma_j} \text{ or } [\mathbf{T}]_{i,[j,j,j,j]} = Z_\sigma^{-1} \theta_{i,\sigma_j}^4$$

- May be computed by multiple methods:
  - minimize least squares in log space (linear)
  - exponentiation (and some divisions)
  - minimize least squares

Future work

- Spectral learning
  - Other models
    - More unknowns (bilinear least squares, etc.)
    - Non-quadratic cost functions
- Spectral Learning of MRFs:
  - Make it work in practice
  - Also make it work for non-symmetric MRFs
  - Run experiments
  - Cry