Select Lab

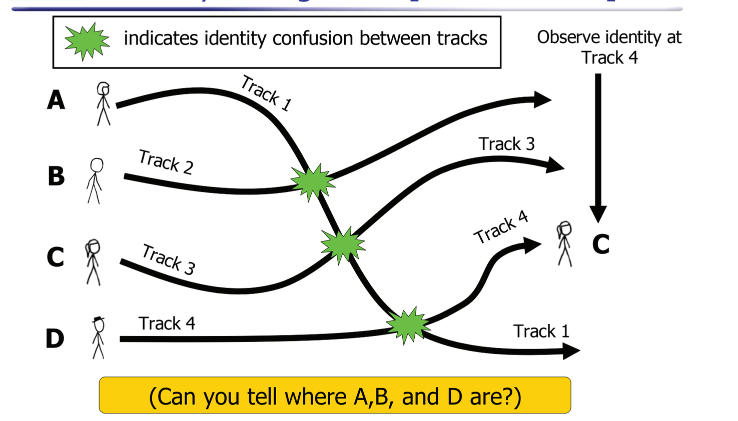
Efficient Inference for Distributions on Permutations

Jonathan Huang Carlos Guestrin Leonidas Guibas

Carnegie Mellon University Stanford University

Carnegie Mellon

Identity Management [Shin et al. '03]



Reasoning with Permutations

- We model uncertainty in identity management with **distributions over permutations**

Identities

Track permutations (A B C D)	P(σ)
1 2 3 4	0
2 1 3 4	0
1 3 2 4	1/10
3 1 2 4	0
2 3 1 4	1/20
3 2 1 4	1/5
1 2 4 3	0
2 1 4 3	0

"**A**lice is at Track **1**, and **B**ob is at Track **3**, and **C**athy is at Track **2**, and **D**avid is at Track **4** with **probability 1/10**"

Probability of each track permutation

How many permutations?

- There are **n!** permutations!

n	n!	Memory required to store n! doubles
9	362,880	3 megabytes
12	4.8×10^{8}	9.5 terabytes
15	1.31×10^{12}	1729 petabytes
24	6.2×10^{23}	4.5×10^{12} petabytes

~ Avogadro's Number

- Graphical models not appropriate due to mutual exclusivity constraints which lead to a fully connected graph
  - If **A** is in Track **1**, then **B** cannot be in Track **1**

Objectives

- Permutations appear in many real world problems!
  - Identity Management / Data Association
  - Card Shuffling Analysis
  - Rankings and Voting Analysis
- We will:
  - Find a **principled, compact representation** for **distributions over permutations** with **tuneable approximation quality**
  - Reformulate **Hidden Markov Model inference** operations with respect to our new representation:
    - Marginalization
    - Conditioning

1st order summaries

- An idea: store **marginal probabilities** that identity **j** maps to track **i**

Identities

Track permutations (A B C D)	P(σ)
1 2 3 4	0
2 1 3 4	0
1 3 2 4	1/10
3 1 2 4	0
2 3 1 4	1/20
3 2 1 4	1/5
1 2 4 3	0
2 1 4 3	0

"**D**avid is at Track **4** with **probability:**
=1/10+1/20+1/5
=7/20"

1st order summaries

- We can **summarize a distribution** using a **matrix of 1st order marginals**
- Requires storing only $\mathbf{n^2}$ numbers!
- Example:

Tracks	A	B	C	D
1	3/10	0	1/2	1/5
2	1/5	1/2	3/10	0
3	3/10	1/5	1/20	9/20
4	1/5	3/10	3/20	7/20

Identities

"**B**ob is at Track **2** with **zero probability**"

"**C**athy is at Track **3** with **probability 1/20**"

The problem with 1st order

- What 1st order summaries **can** capture:
  - P(**A**lice is at Track **1**) = **3/5**
  - P(**B**ob is at Track **2**) = **1/2**
- Now suppose:
  - Tracks **1** and **2** are close, and **A**lice and **B**ob are not next to each other…
  - Then P(**{A**lice,**B**ob**}** occupy Tracks **{1,2}**) = **0**

Unordered Pairs

Moral: 1st order summaries cannot capture higher order dependencies!

2nd order summaries

- Idea #2: store **marginal probabilities** that identities **{k,l}** map to tracks **{i,j}**

Identities

Track permutations (A B C D)	P(σ)
1 2 3 4	0
2 1 3 4	0
1 3 2 4	1/10
3 1 2 4	0
2 3 1 4	1/20
3 2 1 4	1/5
1 2 4 3	0
2 1 4 3	0

"**A**lice and **B**ob occupy Tracks **1** and **2** with **zero probability**"

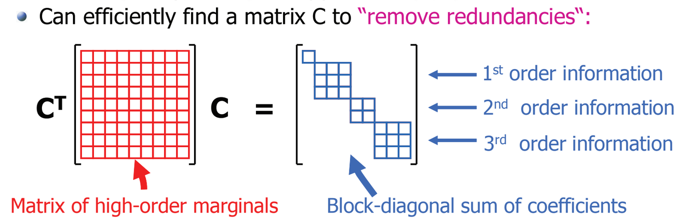
- Requires storing $\mathbf{O(n^4)}$ numbers – one for each pair of unordered pairs, (**{i,j}**,**{k,l}**)

Et cetera…

- **And so forth…** We can define:
  - **3rd-order** marginals
  - **4th-order** marginals
  - **…**
  - **nth-order** marginals (which recovers the original distribution but requires n! numbers)
- **Fundamental Trade-off**: we can capture higher-order dependencies at the cost of storing more numbers

Discarding redundancies

- Matrices of marginal probabilities carry redundant information
  - Example on 4 identities: the probability that {**A**lice,**B**ob} occupy Tracks {**1**,**2**} must be the same as the probability that {**C**athy,**D**avid} occupy Tracks {**3**,**4**}
- Can efficiently find a matrix C to "remove redundancies":



- Instead of storing marginals, only store this minimal set of coefficients (from which marginals can be reconstructed)

The Fourier view

- Our minimal set of coefficients can be interpreted as a **generalized Fourier basis**!! [Diaconis, '88]
- General picture provided by group theory:
  - The space of functions on a group can be decomposed into Fourier components with the familiar properties: Orthogonality, Plancherel's theorem, Convolution theorem, …
- For permutations, simple marginals are "low-frequency":
  - 1st order marginals are "lowest-frequency" basis functions
  - 2nd order (unordered) marginals are 2nd lowest-frequency basis functions
  - …

Group representations

- The **analog of sinusoidal basis functions** for groups are called group representations
- A group representation, ρ of a group G is a **map from G to the set of $d_\rho \times d_\rho$ matrices** such that for all $\sigma_1,\sigma_2 \in$ G:

$$\rho(\sigma_1\sigma_2) = \rho(\sigma_1)\cdot\rho(\sigma_2)$$

This is like: $e^{i\theta_1}\cdot e^{i\theta_2} = e^{i(\theta_1+\theta_2)}$

- Example: The **trivial representation** τ_0 is defined by:

$$\tau_0(\sigma) = 1, \qquad \forall\sigma\in G$$

- The trivial representation is the **constant basis function** and captures the **normalization constant** of a distribution in the generalized Fourier theory

The Fourier Transform

- Each matrix entry of a representation is its own basis function!
- Define the Fourier Transform **of a function f, at the representation** ρ to be the **projection** of f onto the basis given by ρ:

$$\hat{f}_\rho = \sum_{\sigma\in G} f(\sigma)\rho(\sigma)$$

- Note that:
  - Generalized Fourier transforms are **matrix-valued**!
  - And are **functions of representation** (instead of frequency)!
- For most ρ, we end up with an overcomplete basis
  - But… there are only two ways in which linear dependencies can appear in group representations

Example: 1st order representation

- Define the **1st order permutation representation** by:

$$[\tau_1(\sigma)]_{ij} = \begin{cases} 1 & \text{if } \sigma(j) = i \\ 0 & \text{otherwise} \end{cases}.$$

- Example:

$$\tau_1([123]) = \begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix} \quad \tau_1([213]) = \begin{bmatrix}0&1&0\\1&0&0\\0&0&1\end{bmatrix} \quad \tau_1([132]) = \begin{bmatrix}1&0&0\\0&0&1\\0&1&0\end{bmatrix}$$

$$\tau_1([321]) = \begin{bmatrix}0&0&1\\0&1&0\\1&0&0\end{bmatrix} \quad \tau_1([231]) = \begin{bmatrix}0&0&1\\1&0&0\\0&1&0\end{bmatrix} \quad \tau_1([312]) = \begin{bmatrix}0&1&0\\0&0&1\\1&0&0\end{bmatrix}$$

- The Fourier transform of a distribution P, at the 1st order permutation representation is exactly the 1st order matrix of marginal probabilities of P!

Two sources of overcompleteness

1. Can combine two representations $\rho_1,\ \rho_2$ to get a new representation $\rho_1\oplus\rho_2$ (called the direct sum representation):

$$\rho_1\oplus\rho_2 = \left[\begin{array}{c|c}\rho_1 & 0\\ \hline 0 & \rho_2\end{array}\right]$$

Example: (direct sum of two trivial representations)

$$\tau_0\oplus\tau_0(\sigma) = \begin{bmatrix}1&0\\0&1\end{bmatrix} \qquad \forall\sigma\in G$$

2. Given a representation ρ_1, can "change the basis" to get a new representation by conjugating with an invertible matrix, C:

$$\rho_2(\sigma) = C^{-1}\cdot\rho_1(\sigma)\cdot C$$

ρ_1 and ρ_2 are called equivalent representations

Dealing with overcompleteness

- There is a set of **"atomic"** representations which are not equivalent to any direct sum of smaller representations – called **irreducible representations**.
- Maschke's Theorem: Every representation is equivalent to a direct sum of irreducibles!
  - e.g. for any representation ρ

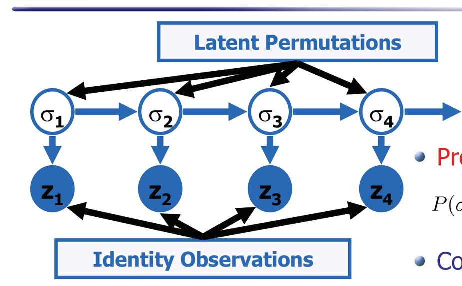
$$\rho(\sigma) = C^{-1}\cdot\bigoplus_k\bigoplus_{\ell=1}^{z_k}\rho_k(\sigma)\cdot C$$

multiplicity

irreducibles

- Connection to Fourier analysis:
  - Peter-Weyl Theorem: Irreducibles form a **complete orthogonal basis** for the space of functions on a group
  - We only store Fourier coefficients at irreducibles

Hidden Markov model inference



- Prediction/Rollup:

$$P(\sigma_t) = \sum_{\sigma_{t-1}} P(\sigma_t|\sigma_{t-1})P(\sigma_{t-1})$$

- Conditioning:

$$P(\sigma|z) \propto L(z|\sigma)P(\sigma)$$

- Problem Statement: For each timestep, return posterior marginal probabilities conditioned on all past observations
- To do inference using Fourier coefficients, we need to cast all inference operations in the Fourier domain

Prediction/Rollup

- Assume σ_{t+1} is generated by the (random walk) rule:
  - Draw $\tau \sim$ **Q(τ)** ← **Mixing Model**
  - Set $\sigma_{t+1} = \tau\cdot\sigma_t$
- For example, **Q([2 1 3 4])=½** means that Tracks **1** and **2** swapped identities with probability ½.
- Prediction/Rollup can be written as a **convolution**:

$$P_{t+1}(\sigma_{t+1}) = \underbrace{\sum_{\sigma_t} P(\sigma_{t+1}|\sigma_t)P_t(\sigma_t)}_{\text{Convolution } (Q*P_t)!}$$

Fourier Domain Prediction/Rollup

- Convolutions are pointwise products in the Fourier domain!
  - Can update individual frequency components independently:
  - Update rule: for each irreducible ½

$$\hat{P}_\rho^{(t+1)} \leftarrow \hat{Q}_\rho^{(t)}\cdot\hat{P}_\rho^{(t)}$$

matrix multiplication

- Fourier domain Prediction/Rollup is exact on the maintained Fourier components!

Conditioning

- **Bayes rule** is a **pointwise product** of the **likelihood function** and **prior distribution**:

$$\underbrace{P(\sigma|z)}_{\text{Posterior}} \propto \underbrace{P(z|\sigma)}_{\text{Likelihood}}\underbrace{P(\sigma)}_{\text{Prior}}$$

- Example likelihood function:
  - P(**z=green** | σ(**A**lice)=Track **1**) = 9/10
  - ("If **A**lice is at Track **1**, then we see **green** at Track **1** with probability **9/10**")

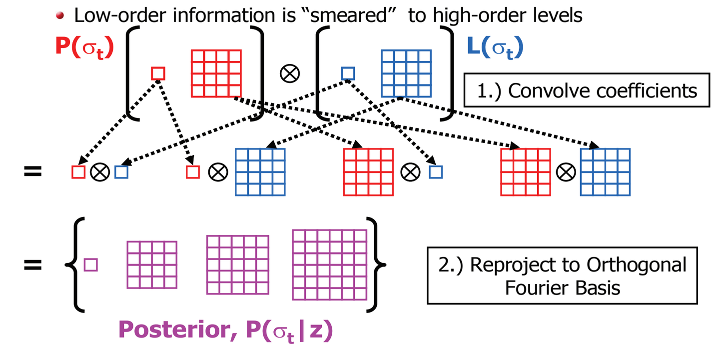
Conditioning

- Suppose we know **1st order marginals of the prior** distribution:
  - P(**A**lice is at Track **1** or Track **2**)=.9
  - P(**B**ob is at Track **1** or Track **2**)=.9
- Then we make a **1st order observation**:
  - "**C**athy is at Track **1** or Track **2** with probability 1"
- (This means that **A**lice and **B**ob cannot both be at Tracks **1** and **2**!)
  - P(**{A**lice,**B**ob**}** occupy Tracks **{1,2}**)=0

Moral: Conditioning increases the representation complexity!

Kronecker Conditioning

- Conditioning is **convolution in the Fourier domain** [Willsky, '78]
  - (except with Kronecker products)
  - Low-order information is "smeared" to high-order levels



Kronecker Conditioning

- More formally, Let $\hat{L}$ and $\hat{P}$ be the Fourier transforms of the likelihood function and prior distribution:
  - For each ordered pair of irreducibles, (ρ_i,ρ_j) define the matrix:

$$A_{ij} = C_{ij}^{-1}\cdot\left(\hat{f}_{\rho_i}\otimes\hat{g}_{\rho_j}\right)\cdot C_{ij}.$$

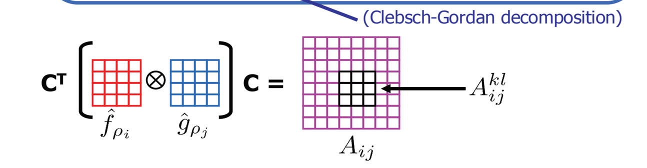
Then the Fourier transform of the posterior is:

$$\left[\widehat{LP}\right]_{\rho_k} = \frac{1}{d_{\rho_k}|G|}\sum_{ij} d_{\rho_i}d_{\rho_j}\sum_{\ell=1}^{z_{ijk}} A_{ij}^{k\ell},$$

where A_{ij}^{kl} is the block of A_{ij} corresponding to the (k,ℓ) block in:

$$\rho_i\otimes\rho_j(\sigma) = C_{ij}^T\cdot\left(\bigoplus_k\bigoplus_{\ell=1}^{z_{ijk}}\rho_k\right)\cdot C_{ij}.$$

(Clebsch-Gordan decomposition)



Bandlimiting and error analysis

- For tractability, discard "high-frequency" coefficients
  - Equivalently, maintain low-order marginals
- Fourier domain Prediction/Rollup **is exact** ☺
- Kronecker Conditioning **introduces error** ☹
- But if we have **enough** coefficients, then Kronecker conditioning **is exact** at a subset of low-frequency terms! ☺

Theorem. If the Kronecker Conditioning Algorithm is called using p^{th} order terms of the prior and q^{th} order terms of the likelihood, then the $(|p-q|)^{th}$ order marginals of the posterior can be reconstructed **without error**.

Dealing with negative numbers

- Consecutive Conditioning steps propagates errors to all frequency levels
- Errors can cause our marginal probabilities to be negative! ☹
- **Solution:** Project to the Marginal Polytope (Fourier coefficients corresponding to nonnegative marginal probabilities) ☺
  - Minimize the distance to the Marginal Polytope in the Fourier domain by using the Plancherel theorem:

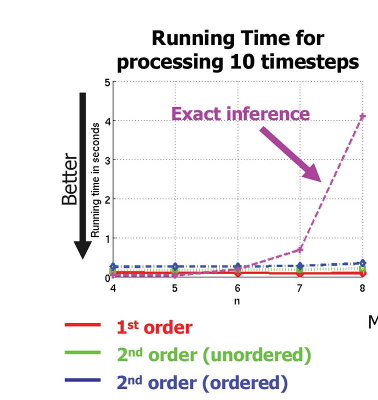
$$\sum_\sigma (f(\sigma)-g(\sigma))^2 = \frac{1}{|G|}\sum_k d_{\rho_k}\mathrm{Tr}\left(\left(\hat{f}_{\rho_k}-\hat{g}_{\rho_k}\right)^T\cdot\left(\hat{f}_{\rho_k}-\hat{g}_{\rho_k}\right)\right)$$

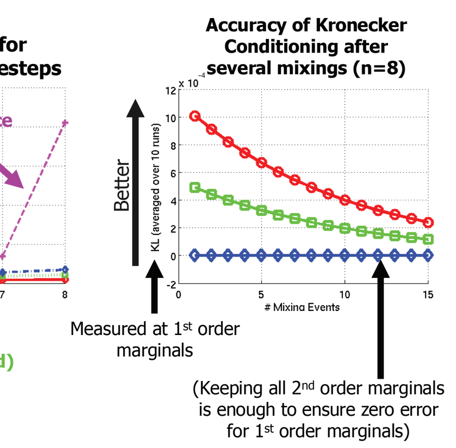
  - Minimization can be written as an efficient Quadratic program!

Algorithm summary

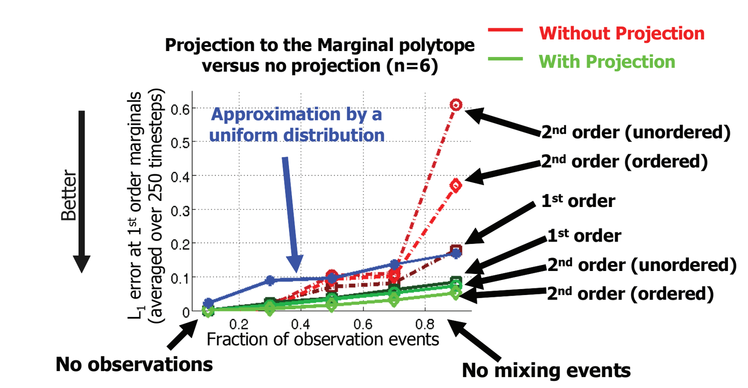
- **Initialize prior** Fourier coefficient matrices $\hat{P}^{(0)}$
- For each timestep t = 1,2,…,T
  - **Prediction/Rollup:**
    - For all coefficient matrices $\hat{P}_i^{(t)}$
      - $\hat{P}_i^{(t)} \leftarrow \hat{Q}_i^{(t)}\cdot\hat{P}_i^{(t-1)}$
  - **Conditioning**
    - For all pairs of coefficient matrices $(\hat{P}_i^{(t)},\hat{P}_j^{(t)})$
      - Compute $\hat{P}_i^{(t)}\otimes\hat{P}_j^{(t)}$ and reproject to the orthogonal Fourier basis using the Clebsch-Gordan decomposition
  - **Drop high frequency coefficients** of $\hat{P}^{(t)}$
  - **Project** $\hat{P}^{(t)}$ to the Marginal polytope using a Quadratic program
- Return marginal probabilities for all timesteps

Simulated data drawn from HMM

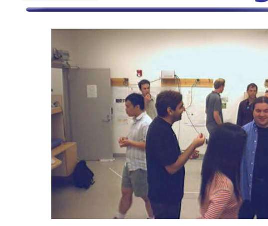




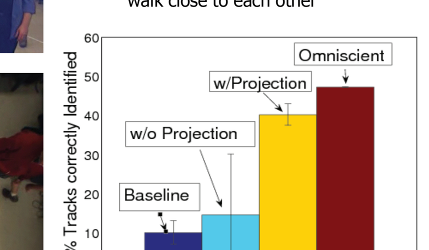
Simulated data drawn from HMM



Tracking with a camera network



- **Camera Network** data:
  - **8** cameras, multiple viewpoints, occlusion effects
  - **11** individuals in lab environment
  - Identity observations obtained from color histograms
  - Mixing events declared when people walk close to each other



Conclusions

- Presented an intuitive, principled representation for distributions on permutations with
  - Fourier-analytic interpretations, and
  - Tuneable approximation quality
- Formulated general and efficient inference operations directly in the Fourier Domain
- Analyzed sources of error which can be introduced by bandlimiting and showed how to combat them by projecting to the marginal polytope
- Evaluated approach on real camera network application and simulated data