Identity Management Problem
— Reasoning and Inference over Permutations

Xiaoye Jiang
Stanford University

Joint work with J. Huang, C. Guestrin, L. Guibas
Identity management [Shin et al., ‘03]

Identity Mixing @Tracks 1,2

Where is Donald Duck?
Mixing @Tracks 1,2

Mixing @Tracks 1,3

Mixing @Tracks 1,4

Where is Track 1

Where is Track 2

Where is Track 3

Where is Track 4

Identity management
Reasoning with Permutations

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

<table>
<thead>
<tr>
<th>Identities</th>
<th>Track permutations</th>
<th>Probability of each track permutation</th>
</tr>
</thead>
<tbody>
<tr>
<td>A B C D</td>
<td>1 2 3 4</td>
<td>[0, 0, 1/10, 0, 0, 0, 0, 0, 0]</td>
</tr>
<tr>
<td></td>
<td>2 1 3 4</td>
<td>[0, 0, 0, 1/20, 0, 0, 0, 0, 0]</td>
</tr>
<tr>
<td></td>
<td>1 3 2 4</td>
<td>[0, 1/10, 0, 0, 1/5, 0, 0, 0, 0]</td>
</tr>
<tr>
<td></td>
<td>3 1 2 4</td>
<td>[0, 0, 0, 0, 0, 0, 0, 0, 0]</td>
</tr>
<tr>
<td></td>
<td>2 3 1 4</td>
<td>[0, 0, 1/20, 0, 0, 0, 0, 0, 0]</td>
</tr>
<tr>
<td></td>
<td>3 2 1 4</td>
<td>[0, 0, 0, 0, 1/5, 0, 0, 0, 0]</td>
</tr>
<tr>
<td></td>
<td>1 2 4 3</td>
<td>[0, 0, 0, 0, 0, 0, 0, 0, 0]</td>
</tr>
<tr>
<td></td>
<td>2 1 4 3</td>
<td>[0, 0, 0, 0, 0, 0, 0, 0, 0]</td>
</tr>
</tbody>
</table>
How many permutations?

- There are $n!$ permutations!

<table>
<thead>
<tr>
<th>$n$</th>
<th>$n!$</th>
<th>Memory required to store $n!$ doubles</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>362,880</td>
<td>3 megabytes</td>
</tr>
<tr>
<td>12</td>
<td>$4.8 \times 10^8$</td>
<td>9.5 terabytes</td>
</tr>
<tr>
<td>15</td>
<td>$1.31 \times 10^{12}$</td>
<td>1729 petabytes</td>
</tr>
</tbody>
</table>

My advisor won’t buy me this much memory!

- Graphical models are not effective due to mutual exclusivity constraints ("Alice and Bob cannot both be at Track 1 simultaneously")
Objectives

• We would like to:
  − Find a *principled, compact representation* for *distributions over permutations* with *tuneable approximation quality*
  − Reformulate *Markov Model inference* operations with respect to our new representation:
    • Marginalization
    • Conditioning
1\textsuperscript{st} order summaries

- An idea: For each (identity \( j \), track \( i \)) pair, store \textit{marginal probability} that \( j \) maps to \( i \)

<table>
<thead>
<tr>
<th>Identities</th>
<th>P(( \sigma ))</th>
</tr>
</thead>
<tbody>
<tr>
<td>A B C D</td>
<td></td>
</tr>
<tr>
<td>1 2 3 4</td>
<td>0</td>
</tr>
<tr>
<td>2 1 3 4</td>
<td>0</td>
</tr>
<tr>
<td>1 3 2 4</td>
<td>1/10</td>
</tr>
<tr>
<td>3 1 2 4</td>
<td>0</td>
</tr>
<tr>
<td>2 3 1 4</td>
<td>1/20</td>
</tr>
<tr>
<td>3 2 1 4</td>
<td>1/5</td>
</tr>
<tr>
<td>1 2 4 3</td>
<td>0</td>
</tr>
<tr>
<td>2 1 4 3</td>
<td>0</td>
</tr>
</tbody>
</table>

“David is at Track 4 with probability: \(=1/10+1/20+1/5\) = 7/20”
1\textsuperscript{st} order summaries

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

<table>
<thead>
<tr>
<th>Tracks</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3/10</td>
<td>0</td>
<td>1/2</td>
<td>1/5</td>
</tr>
<tr>
<td>2</td>
<td>1/5</td>
<td>1/2</td>
<td>3/10</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>3/10</td>
<td>1/5</td>
<td>1/20</td>
<td>3/20</td>
</tr>
<tr>
<td>4</td>
<td>1/5</td>
<td>3/10</td>
<td>3/20</td>
<td>7/20</td>
</tr>
</tbody>
</table>

"Bob is at Track 2 with zero probability"

"Cathy is at Track 3 with probability 1/20"
The problem with 1\textsuperscript{st} order

- What 1\textsuperscript{st} order summaries can capture:
  - \(P(\text{Alice is at Track 1}) = 3/5\)
  - \(P(\text{Bob is at Track 2}) = 1/2\)

- Now suppose:
  - Tracks 1 and 2 are close,
  - Alice and Bob are not next to each other
  - \(P(\{\text{Alice,Bob}\} \text{ occupy Tracks } \{1,2\}) = 0\)

1\textsuperscript{st} order summaries cannot capture higher order dependencies!
2\textsuperscript{nd} order summaries

- Idea #2: store **marginal probabilities** that unordered pairs of identities \( \{k,l\} \) map to pairs of tracks \( \{i,j\} \)

<table>
<thead>
<tr>
<th>Identities</th>
<th>Track permutations</th>
<th>( P(\sigma) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>A B C D</td>
<td>1 2 3 4</td>
<td>0</td>
</tr>
<tr>
<td>2 1 3 4</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>1 3 2 4</td>
<td>1/10</td>
<td></td>
</tr>
<tr>
<td>3 1 2 4</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>2 3 1 4</td>
<td>1/20</td>
<td></td>
</tr>
<tr>
<td>3 2 1 4</td>
<td>1/5</td>
<td></td>
</tr>
<tr>
<td>1 2 4 3</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>2 1 4 3</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

"Alice and Bob occupy Tracks 1 and 2 with zero probability"
**2\textsuperscript{nd} order summaries**

<table>
<thead>
<tr>
<th></th>
<th>{A,B}</th>
<th>{A,C}</th>
<th>{A,D}</th>
</tr>
</thead>
<tbody>
<tr>
<td>{1,2}</td>
<td>0</td>
<td>2/5</td>
<td>1/10</td>
</tr>
<tr>
<td>{1,3}</td>
<td>1/10</td>
<td>2/5</td>
<td>3/10</td>
</tr>
<tr>
<td>{1,4}</td>
<td>1/5</td>
<td>1/5</td>
<td>1/10</td>
</tr>
</tbody>
</table>

“**Alice and Bob occupy Tracks 1 and 4 with probability 1/5**”
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 \{Alice, Bob\} occupy Tracks \{1,2\} must be the same as the probability that \{Cathy, David\} occupy Tracks \{3,4\}

- Can efficiently find a matrix \(C\) to “remove redundancies“:

\[
CT \begin{bmatrix}
\begin{array}{cccc}
\vdots & 
\vdots & 
\vdots & 
\vdots \\
\vdots & 
\vdots & 
\vdots & 
\vdots \\
\vdots & 
\vdots & 
\vdots & 
\vdots \\
\vdots & 
\vdots & 
\vdots & 
\vdots \\
\end{array}
\end{bmatrix}
C = \begin{bmatrix}
\begin{array}{cccc}
\vdots & 
\vdots & 
\vdots & 
\vdots \\
\vdots & 
\vdots & 
\vdots & 
\vdots \\
\vdots & 
\vdots & 
\vdots & 
\vdots \\
\vdots & 
\vdots & 
\vdots & 
\vdots \\
\end{array}
\end{bmatrix}
\]

- Instead of storing marginals, only store these blocks of coefficients (from which marginals can be reconstructed)
Completeness

• If we have enough coefficients (by removing the redundancies from $n^{th}$ order marginals), we can reconstruct the original distribution:

(Complete Basis for functions over permutations)
The compact representations can be viewed as a **generalized Fourier basis** [Diaconis, ’88]:

- The familiar properties hold: *Linearity*, *Orthogonality*, *Completeness*, *Plancherel’s (Parseval’s) theorem*, *Convolution theorem*, ...

- **To do inference** using low dimensional **Fourier projections**, we need to cast all inference operations **in the Fourier domain** 😊
• Problem statement: For each timestep, find posterior marginals conditioned on all past observations

• Need to formulate inference routines with respect to Fourier coefficients!

Mixing Model – “e.g. Tracks 2 and 3 swapped identities with probability ½”

Observation Model – “e.g. see green blob at track 3”
Hidden Markov model inference

• Two basic inference operations for Hidden Markov Models:
  - **Prediction/rollup:**
    \[ P_{t+1}(\sigma_{t+1}) = \sum_{\sigma_t} P(\sigma_{t+1}|\sigma_t) P_t(\sigma_t) \]
  - **Conditioning:**
    \[ P(\sigma|z) \propto P(z|\sigma) P(\sigma) \]

• How can we do these operations without enumerating all n! permutations?
Prediction/Rollup

- We assume that $\sigma_{t+1}$ is generated by the rule:
  - Draw $\tau \sim Q(\tau)$
  - Set $\sigma_{t+1} = \tau \cdot \sigma_t$

- For example, $Q([2 \ 1 \ 3 \ 4]) = \frac{1}{2}$ means that Tracks 1 and 2 swapped identities with probability $\frac{1}{2}$.

- Prediction/Rollup can be written as a convolution:

$$P_{t+1}(\sigma_{t+1}) = \sum_{\sigma_t} P(\sigma_{t+1} | \sigma_t) P_t(\sigma_t)$$

Convolution $(Q \ast P_t)$!
Fourier Domain Prediction/Rollup

- Convolutions are **pointwise products** in the Fourier domain:

\[ P(\sigma_t) \square \begin{array}{c}
\end{array} Q(\tau) = P(\sigma_{t+1}) \]

**Prediction/Rollup does not increase** the representation complexity!
Conditioning

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

\[ P(\sigma | z) \propto P(z | \sigma) P(\sigma) \]

- Example likelihood function:
  - \( P(z=\text{green} | \sigma(\text{Alice})=\text{Track 1}) = 9/10 \)
  - (“Prob. we see **green** at Track 1 given Alice is at Track 1 is 9/10”)
Kronecker Conditioning

Pointwise products correspond to convolution in the Fourier domain [Willsky, ‘78] (except with Kronecker Products in our case)

Our algorithm handles any prior and any likelihood, generalizing the previous FFT-based conditioning method [Kondor et al., ‘07]
Conditioning

- Conditioning increases the representation complexity!
- Example: Suppose we start with 1st order marginals of the prior distribution:
  - $P(\text{Alice is at Track 1 or Track 2}) = 0.9$
  - $P(\text{Bob is at Track 1 or Track 2}) = 0.9$
  - …
- Then we make a 1st order observation:
  - "Cathy is at Track 1 or Track 2 with probability 1"
- (This means that Alice and Bob cannot both be at Tracks 1 and 2!)
  - $P(\{\text{Alice, Bob}\} \text{ occupy Tracks } \{1, 2\}) = 0$

Need to store 2nd order probabilities after conditioning!
Bandlimiting

- After conditioning, we **discard** “high-frequency” coefficients
  - Equivalently, we **maintain** low-order marginals
- Example:
  - [Diagram showing a grid with red squares kept and black squares discarded]
Error analysis

- Fourier domain Prediction/Rollup is exact 😊
- Kronecker Conditioning introduces error 😞
- But…
  - If enough coefficients are maintained, 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.
Kronecker Conditioning experiments

Error of Kronecker Conditioning, n=8
(as a function of diffuseness)

Measured at 1\textsuperscript{st} order marginals

(Keeping 3\textsuperscript{rd} order marginals is enough to ensure zero error for 1\textsuperscript{st} order marginals)
Dealing with negative numbers

• Consecutive Conditioning steps can propagate errors to all frequency levels
• Errors can sometimes cause our marginal probabilities to be negative! 😞

• Our Solution: Project to relaxed Marginal Polytope (space of Fourier coefficients corresponding to nonnegative marginal probabilities)
  - Projection can be formulated as an efficient Quadratic Program in the Fourier domain
Simulated data drawn from HMM

Projection to the Marginal polytope versus no projection (n=6)

Approximation by a uniform distribution

Better

Averaged over 250 timesteps

$L^1$ error at 1st order Marginals

Without Projection

With Projection

1$^{st}$ order

2$^{nd}$ order

3$^{rd}$ order

1$^{st}$ order

2$^{nd}$ order

3$^{rd}$ order
Running Time comparison

Running time of 10 forward algorithm iterations

Running time in seconds

Better

Exact inference

3rd order

2nd order

1st order
Tracking with a camera network

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

![Graph showing improvement with projection](image)

- Omniscient tracker
- Projections are **crucial** in practice!!
Summary of Fourier Approach

• 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
Fourier theoretic approaches

- Approximate distributions over permutations with **low frequency basis functions** [Kondor2007, Huang2007]

\[ f(x) = 0.6 \cdot x + 0.2 \cdot x + 0.5 \cdot x + 0.3 \cdot x \]

- Fourier analysis on the real line
- Fourier analysis on \( S_n \) (Permutations of \( n \) objects)
- Sinusoidal basis

Fourier coefficients ↔ Fourier basis functions

???
Uncertainty Principle: a signal $f$ cannot be sparsely represented in both the time and Fourier domains.
Uncertainty principle on permutations

- Confusion between tracks 1, 2
- Confusion between tracks 3, 4

Decomposed distributions can be captured more accurately, with fewer coefficients.

fraction of energy stored in all frequencies

frequency

harder to represent

Keep full joint distributions over $S_8$.

No uncertainty within group, no mixing between groups.

Uniform on 1

2 subgroups of 4

8 subgroups of 1 (peaked dist.)

Keep full joint distribution over $S_1$.
Adaptive decompositions

• **Our approach:** adaptively factor problem into subgroups allowing for higher order representations for smaller subgroups

Claim: Adaptive Identity Management can be highly scalable, more accurate for sharp distributions

“This is Bob”
(and Bob was originally in the Blue group)
Contributions

• Characterization of constraints on Fourier coefficients on permutations implied by probabilistic independence

• Two algorithms: for factoring a distribution (Split) and combining independent factors in the Fourier domain (Join)

• Adaptive algorithm for scalable identity management (handles up to $n \sim 100$ tracks)
First-order independence condition

- **Independence**
  \[
  P(\sigma(i) = k_1 \text{ and } \sigma(j) = k_2) = 0
  \]
  \[
  P(\sigma(i) = k_1) \cdot P(\sigma(j) = k_2)
  \]
  Product of first-order marginals

- **Mutual Exclusivity**
  - “Alice and Bob not both at Track
  \[
  P(\sigma(i) = k \text{ and } \sigma(j) = k) = 0
  \]
First-order independence condition

Not independent

Independent

Can verify condition using first-order marginals
First-order independence

- First-order condition is insufficient:

  “Alice guards Bob”

First-order independence ignores the fact that Alice and Bob are always next to each other!

“Alice is in red team”

“Bob is in blue team”

image from [sullivan06]
The problem with first-order

- First-order marginals look like:

\[ P(A_{\text{lice is at Track 1}}) = \frac{3}{5} \]
\[ P(B_{\text{ob is at Track 2}}) = \frac{1}{2} \]

- Now suppose Alice guards Bob, and...

Can write as second-order marginal:

\[ P(\{\text{Alice,Bob}\} \text{ occupy Tracks } \{1,2\}) = 0 \]

- Tracks 1,2 very far apart
Second-order summaries

- Store summaries for ordered pairs:

<table>
<thead>
<tr>
<th>Tracks</th>
<th>Identities</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1,2)</td>
<td>(A,B) 1/6</td>
</tr>
<tr>
<td>(2,1)</td>
<td>1/12</td>
</tr>
<tr>
<td>(1,3)</td>
<td>1/12</td>
</tr>
<tr>
<td>(3,1)</td>
<td>1/12</td>
</tr>
</tbody>
</table>

“Bob is at Track 1 and Alice is at Track 3 with probability 1/12”

store marginal probability that identities (k,l) map to tracks (i,j)

- 2\textsuperscript{rd} order summary requires $O(n^4)$ storage
Trade-off: capture higher frequencies by storing more numbers.

Remark: high-order marginals contain low-order information.

<table>
<thead>
<tr>
<th>Margin</th>
<th>Fourier Interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0\text{th} order</td>
<td><em>Lowest</em> frequency Fourier coefficient</td>
</tr>
<tr>
<td>1\text{st} order</td>
<td>Reconstructible from $O(n^2)$ lowest frequency coefficients</td>
</tr>
<tr>
<td>2\text{nd} order</td>
<td>Reconstructible from $O(n^4)$ lowest frequency coefficients</td>
</tr>
<tr>
<td>3\text{rd} order</td>
<td>Reconstructible from $O(n^6)$ lowest frequency coefficients</td>
</tr>
<tr>
<td>\ldots</td>
<td>\ldots</td>
</tr>
<tr>
<td>n\text{th} order</td>
<td>Requires all $n!$ Fourier coefficients</td>
</tr>
</tbody>
</table>

Sum over entire distribution (always equal to 1)

Recovers original distribution, requires storing $n!$ numbers
Fourier coefficient matrices

- Fourier coefficients on permutations are a collection of square matrices ordered by “frequency”:

- Bandlimiting - keep a truncated set of coefficients
- Fourier domain inference – prediction/conditioning in the Fourier domain
  - [Kondor et al, AISTATS07]
  - [Huang et al, NIPS07]
Back to independence

- Need to consider two operations
  - Groups **join** when tracks from two groups mix
  - Groups **split** when an observation allows us to reason over smaller groups independently

“This is Bob”
(and Bob was originally in the Blue group)
Problems

- If the joint distribution $h$ factors as a product of distributions $f$ and $g$:

$$h(\sigma) = f(\sigma) \cdot g(\sigma)$$

Distribution over tracks $\{1,\ldots,p\}$  Distribution over tracks $\{p+1,\ldots,n\}$

(Join problem) Find Fourier coefficients of the joint $h$ given Fourier coefficients of factors $f$ and $g$?

(Split problem) Find Fourier coefficients of factors $f$ and $g$ given Fourier coefficients of the joint $h$?
First-order join

- Given first-order marginals of $f$ and $g$, what does the matrix of first-order marginals of $h$ look like?
Higher-order joining

• Given Fourier coefficients of the factors $f$ and $g$ at each frequency level:

\[
\hat{f}_{\lambda_1}, \hat{f}_{\lambda_2}, \hat{f}_{\lambda_3}, \ldots \quad \hat{g}_{\lambda_1}, \hat{g}_{\lambda_2}, \hat{g}_{\lambda_3}, \ldots
\]

• Compute Fourier coefficients of the joint distribution $h$ at each frequency level:

\[
\hat{h}_{\lambda_1}, \hat{h}_{\lambda_2}, \hat{h}_{\lambda_3}, \ldots
\]
Higher-order joining

- Joining for higher-order coefficients gives similar block-diagonal structure
  - Also get *Kronecker product structure* for each block

Blocks appear multiple times (multiplicities related to *Littlewood-Richardson coefficients*)

\[
\hat{h}_\lambda = L^T_\lambda \cdot L_\lambda
\]
Problems

• If the joint distribution $h$ factors as a product of distributions $f$ and $g$:

$$h(\sigma) = f(\sigma) \cdot g(\sigma)$$

Distribution over tracks $\{1, \ldots, p\}$

Distribution over tracks $\{p+1, \ldots, n\}$

(Join problem) Find Fourier coefficients of the joint $h$ given Fourier coefficients of factors $f$ and $g$?

(Split problem) Find Fourier coefficients of factors $f$ and $g$ given Fourier coefficients of the joint $h$?
Splitting

• Want to “invert” the Join process:

Consider recovering 2\textsuperscript{nd} Fourier block

Our approach: search for blocks of the form:

\[ \hat{f}_\lambda \otimes 1 \quad 1 \otimes \hat{g}_\lambda \]

\[ \hat{h}_\lambda = L^T_\lambda \cdot \]

Need to recover \( A, B \) from \( A \otimes B \) – only possible to do up to scaling factor

\textbf{Theorem}: these blocks always exist! (and are efficient to find)
Marginal preservation

- **Problem**: In practice, never have entire set of Fourier coefficients!

- **Marginal preservation guarantee**:
  - Conversely, get a similar guarantee for splitting
  - (Usually get some higher order information too)

**Theorem**: Given $m^{th}$-order marginals for independent factors, we exactly recover $m^{th}$-order marginals for the joint.
Detecting independence

To adaptively split large distributions, need to detect independent subsets.

Can use **(bi)clustering** on matrix of marginals to discover an appropriate ordering!

Balance constraint: force nonzero blocks to be square.

In practice, get unordered identities, tracks…

Considerals with appropriate ordering on identities and tracks.
First-order independence

- First-order condition is insufficient: "Alice guards Bob"

Even when higher-order independence does not hold:

**Theorem**: Whenever first-order independence holds, Split returns exact marginals of each subset of tracks.

"Alice is in red team"  "Bob is in blue team"

Can check for higher order independence after detecting at first-order

- What if we call Split when **only the first-order condition** is satisfied?
Experiments - accuracy

dataset from [Khan et al. 2006]
Experiments – running time

- **Graph 1:**
  - Y-axis: elapsed time per run (seconds)
  - X-axis: ratio of observations
  - Two lines: nonadaptive and adaptive

- **Graph 2:**
  - Y-axis: Running time (seconds)
  - X-axis: Number of ants
  - Two lines: nonadaptive and adaptive
Final Conclusions

Scalable and adaptive identity management algorithm to track up to n=100 objects

Two new algorithms marginalization, conditioning, join, split

 Completely Fourier-theoretic characterization of probabilistic independence

Thank you!