Inference for Distributions over the Permutation Group

Jonathan Huang*   Carlos Guestrin*
Leonidas Guibas**

May 2008
CMU-ML-08-108

MACHINE LEARNING
DEPARTMENT

Carnegie Mellon

Inference for Distributions over the Permutation Group

Jonathan Huang* **Carlos Guestrin***
Leonidas Guibas**

May 2008
CMU-ML-08-108

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

Permutations are ubiquitous in many real-world problems, such as voting, ranking, and data association. Representing uncertainty over permutations is challenging, since there are $n!$ possibilities, and typical compact and factorized probability distribution representations, such as graphical models, cannot capture the mutual exclusivity Constraints associated with permutations. In this paper, we use the "low-frequency" terms of a Fourier decomposition to represent distributions over permutations compactly. We present *Kronecker conditioning*, a new general and efficient approach for maintaining and updating these distributions directly in the Fourier domain. Low order Fourier-based approximations, however, may lead to functions that do not correspond to valid distributions. To address this problem, we present an efficient quadratic program defined directly in the Fourier domain for projection the approximation onto a relaxation of the polytope of legal marginal distributions. We demonstrate the effectiveness of our approach on a real camera-based multi-person tracking scenario.

*School of Computer Science, Carnegie Mellon University, Pittsburgh, PA , USA
**Computer Science Department, Stanford University, Stanford, CA, USA

Keywords: Identity Management; Permutations; Approximate Inference; Group Theoretical Methods; Sensor Networks

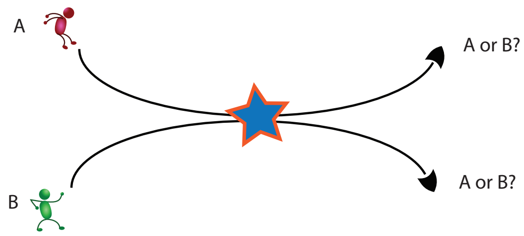


Figure 1: When two persons pass near each other, their identities can get confused.

1 Introduction

Probability distributions over permutations arise in a diverse variety of real world problems. While they were perhaps first studied in the context of gambling and card games, they have now been found to be applicable to many important problems in multi-object tracking, information retrieval, webpage ranking, preference elicitation, and voting.

As an example, consider the problem of tracking n persons based on a set of noisy measurements of identity and position. A typical tracking system might attempt to manage a set of n tracks along with an identity corresponding to each track, in spite of ambiguities from imperfect identity measurements. When the persons are well separated, the problem is easily decomposed and measurements about each individual can be clearly associated with a particular track. When persons pass near each other, however, confusion can arise as their signal signatures may mix; see Figure 1. After the individuals separate again, their positions may be clearly distinguishable, but their identities can still be confused, resulting in identity uncertainty which must be propagated forward in time with each person, until additional observations allow for disambiguation. This task of maintaining a belief state for the correct association between object tracks and object identities while accounting for local mixing events and sensor observations, was introduced in (Shin et al., 2003) and is called the *identity management problem*.

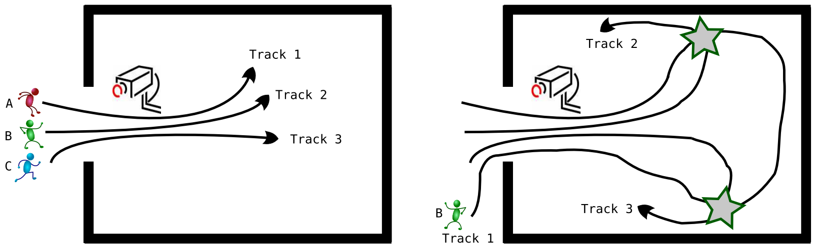
The identity management problem poses a challenge for probabilistic inference because it needs to address the fundamental combinatorial challenge that there is a factorial number of associations to maintain between tracks and identities. Distributions over the space of all permutations require storing at least

$n! - 1$ numbers, an infeasible task for all but very small n. Moreover, typical compact representations, such as graphical models, cannot efficiently capture the mutual exclusivity constraints associated with permutations.

While there have been many approaches for coping with the factorial complexity of maintaining a distribution over permutations, most attack the problem using one of two ideas — storing and updating a small subset of likely permutations, or, as in our case, restricting consideration to a tractable subspace of possible distributions. (Willsky, 1978) was the first to formulate the probabilistic filtering/smoothing problem for group-valued random variables. He proposed an efficient FFT based approach of transforming between primal and Fourier domains so as to avoid costly convolutions, and provided efficient algorithms for dihedral and metacyclic groups. (Kueh et al., 1999) show that probability distributions on the group of permutations are well approximated by a small subset of Fourier coefficients of the actual distribution, allowing for a principled tradeoff between accuracy and complexity. The approach taken in (Shin et al., 2005; Schumitsch et al., 2005; Schumitsch et al., 2006) can be seen as an algorithm for maintaining a particular fixed subset of Fourier coefficients of the log density. Most recently, (Kondor et al., 2007) allow for a general set of Fourier coefficients, but assume a restrictive form of the observation model in order to exploit an efficient FFT factorization.

In this work[1], we present several contributions which generalize and improve upon the past related work. We present a new and simple algorithm, called *Kronecker Conditioning*, which performs all probabilistic inference operations completely in the Fourier domain, allowing for a principled tradeoff between computational complexity and approximation accuracy. Our approach is fully general, in the sense that it can address any transition model or likelihood function that can be represented in the Fourier domain, such as those used in previous work, and can represent the probability distribution using any desired number of Fourier coefficients. We analyze the errors which can be introduced by bandlimiting a probability distribution and show how they propagate with respect to inference operations. Approximate conditioning based on bandlimited distributions can sometimes yield Fourier coefficients which do not correspond to any valid distribution, even returning negative "probabilities" on occasion — we address this issue by presenting a method for projecting the result back into the polytope of coefficients which correspond to nonnegative and consistent marginal probabilities using an efficient quadratic program. Finally, we empirically evaluate the accuracy of approximate inference on simulated data drawn from our model and further demonstrate the effectiveness of our approach on a real camera-based multi-person tracking scenario.

[1] A shorter version this work appeared in (Huang et al., 2007). We provide a more complete discussion of our Fourier based methods in this extended paper.



(a) *Before* (b) *After*

Figure 2: Identity Management example. Three people, Alice, Bob and Charlie enter a room and we receive a position measurement for each person at each time step. With no way to observe identities inside the room, however, we are confused whenever two tracks get too close. In this example, track 1 crosses with track 2, then with track 3, then leaves the room, at which point it is observed that the identity at Track 1 is in fact Bob.

2 Filtering over permutations

As a prelude to the general problem statement, we begin with a simple identity management problem on three tracks (illustrated in Figure 2) which we will use as a running example. In this problem, we observe a stream of localization data from three people walking inside a room. Except for a camera positioned at the entrance, however, there is no way to distinguish between identities once they are inside. In this example, an internal tracker declares that two tracks have 'mixed' whenever they get too close to each other and announces the identity of any track that enters or exits the room.

In our particular example, three people, Alice, Bob and Cathy, enter a room separately, walk around, and we observe Bob as he exits. The events for our particular example in the figure are recorded in Table 1. Since Tracks 2 and 3 never mix, we know that Cathy cannot be in Track 2 in the end, and furthermore, since we observe Bob to be in Track 1 when he exits, we can deduce that Cathy must have been in Track 3, and therefore Alice must have been in Track 2. Our simple example illustrates the combinatorial nature of the problem — in particular, reasoning about the mixing events allows us to exactly decide where Alice and Cathy were even though we only made an observation about Bob at the end.

Event #	*Event Type*
1	*Tracks 1 and 2 mixed*
2	*Tracks 1 and 3 mixed*
3	*Observed Identity Bob at Track 1*

Table 1: Table of Mixing and Observation events logged by the tracker.

In identity management, a permutation σ represents a joint assignment of identities to internal tracks, with $\sigma(i)$ being the track belonging to the ith identity. When people walk too closely together, their identities can be confused, leading to uncertainty over σ. To model this uncertainty, we use a *Hidden Markov Model (HMM)* on permutations, which is a joint distribution over latent permutations $\sigma^{(1)}, \ldots, \sigma^{(T)}$, and observed variables $z^{(1)}, \ldots, z^{(T)}$ which factors as:

$$P(\sigma^{(1)}, \ldots, \sigma^{(T)}, z^{(1)}, \ldots, z^{(T)}) = P(\sigma^{(1)})P(z^{(1)}|\sigma^{(1)}) \prod_{t=2}^{T} P(z^{t}|\sigma^{(t)}) \cdot P(\sigma^{(t)}|\sigma^{(t-1)}).$$

The conditional probability distribution $P(\sigma^{(t)}|\sigma^{(t-1)})$ is called the *transition model*, and might reflect, for example, that the identities belonging to two tracks were swapped with some probability by a mixing event. The distribution $P(z^{(t)}|\sigma^{(t)})$ is called the *observation model*, which might, for example, capture a distribution over the color of clothing for each individual.

We focus on *filtering*, in which one queries the HMM for the posterior at some time step, conditioned on all past observations. Given the distribution $P(\sigma^{(t)}|z^{(1)}, \ldots, z^{(t)})$, we recursively compute $P(\sigma^{(t+1)}|z^{(1)}, \ldots, z^{(t+1)})$ in two steps: a *prediction/rollup* step and a *conditioning* step. Taken together, these two steps form the well known *Forward Algorithm* (Rabiner, 1989). The prediction/rollup step multiplies the distribution by the transition model and marginalizes out the previous time step:

$$P(\sigma^{(t+1)}|z^{(1)}, \ldots, z^{(t)}) = \sum_{\sigma^{(t)}} P(\sigma^{(t+1)}|\sigma^{(t)})P(\sigma^{(t)}|z^{(1)}, \ldots, z^{(t)}).$$

The conditioning step conditions the distribution on an observation $z^{(t+1)}$ using Bayes rule:

$$P(\sigma^{(t+1)}|z^{(1)}, \ldots, z^{(t+1)}) \propto P(z^{(t+1)}|\sigma^{(t+1)})P(\sigma^{(t+1)}|z^{(1)}, \ldots, z^{(t)}).$$

Since there are $n!$ permutations, a single iteration of the algorithm requires $O((n!)^2)$ flops and is consequently intractable for all but very small n. The approach that we advocate is to maintain a compact approximation to the true distribution based on the Fourier transform. As we discuss later, the Fourier based approximation is equivalent to maintaining a set of low-order marginals, rather than the full joint, which we regard as being analogous to an *Assumed Density Filter* (Boyen & Koller, 1998). Although we focus on HMMs and filtering for concreteness, the approach we describe is useful for other probabilistic inference tasks over permutations, such as ranking objects and modeling user preferences.

3 Probability Distributions over the Symmetric Group

A permutation on n elements is a one-to-one mapping of the set $\{1, \ldots, n\}$ into itself and can be written as a tuple,

$$\sigma = [\sigma(1) \ \ \sigma(2) \ \ \ldots \ \ \sigma(n)],$$

where $\sigma(i)$ denotes where the ith element is mapped under the permutation (called *one line notation*). For example, $\sigma = [2 \ 3 \ 1 \ 4 \ 5]$ means that $\sigma(1) = 2$, $\sigma(2) = 3$, $\sigma(3) = 1$, $\sigma(4) = 4$, and $\sigma(5) = 5$. The set of all permutations on n elements forms a group under the operation of function composition — that is, if σ_1 and σ_2 are permutations, then

$$\sigma_1\sigma_2 = [\sigma_1(\sigma_2(1)) \ \ \sigma_1(\sigma_2(2)) \ \ \ldots \ \ \sigma_1(\sigma_2(n))]$$

is itself a permutation. The set of all $n!$ permutations is called the *Symmetric Group*, or just S_n.

We will actually notate the elements of S_n using the more standard *cycle notation*, in which a *cycle* $(i, j, k, \ldots, \ell)$ refers to the permutation which maps i to j, j to k, $\ldots$, and finally ℓ to i. Though not every permutation can be written as a single cycle, any permutation can always be written as a product of disjoint cycles. For example, the permutation $\sigma = [2 \ 3 \ 1 \ 4 \ 5]$ written in cycle notation is $\sigma = (1, 2, 3)(4)(5)$. The number of elements in a cycle is called the *cycle length* and we typically drop the length 1 cycles in cycle notation when it creates no ambiguity — in our example, $\sigma = (1, 2, 3)(4)(5) = (1, 2, 3)$. We refer to the identity permutation (which maps every element to itself) as ϵ.

A probability distribution over permutations can be thought of as a joint distribution on the n random variables $(\sigma(1), \ldots, \sigma(n))$ subject to the *mutual exclusivity constraints* that $P(\sigma : \sigma(i) = \sigma(j)) = 0$ whenever $i \neq j$. For example, in the identity management problem, Alice and Bob cannot both be in Track 1 simultaneously. Due to the fact that all of the $\sigma(i)$ are coupled in the joint distribution, graphical models, which might have otherwise exploited an underlying conditional independence structure, are ineffective. Instead, our Fourier based approximation achieves compactness by exploiting the *algebraic structure* of the problem.

3.1 Compact summary statistics

While continuous distributions like Gaussians are typically summarized using moments (like mean and variance), or more generally, expected features, it is not immediately obvious how one might, for example, compute the 'mean' of a distribution over permutations. There is a simple method that might spring to mind, however, which is to think of the permutations as *permutation matrices* and to average the matrices instead.

Example 1. *For example, consider the two permutations* $\epsilon, (1,2) \in S_3$ *(*ϵ *is the identity and* $(1,2)$ *swaps 1 and 2). We can associate the identity permutation* ϵ *with the* 3×3 *identity matrix, and similarly, we can associate the permutation* $(1,2)$ *with the matrix:*

$$(1,2) \mapsto \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

The 'average' of ϵ *and* $(1,2)$ *is therefore:*

$$\frac{1}{2}\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} + \frac{1}{2}\begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 1/2 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

As we will later show, computing the 'mean' (as described above) of a distribution over permutations, P, compactly summarizes P by storing a marginal distribution over each of $\sigma(1), \sigma(2), \dots, \sigma(n)$, which requires storing only $O(n^2)$ numbers rather than the full $O(n!)$ for the exact distribution. As an example, one possible summary might look like:

$$\widehat{P} = \left[\begin{array}{c|ccc} & \text{Alice} & \text{Bob} & \text{Cathy} \\ \hline \text{Track 1} & 2/3 & 1/6 & 1/6 \\ \text{Track 2} & 1/3 & 1/3 & 1/3 \\ \text{Track 3} & 0 & 1/2 & 1/2 \end{array}\right].$$

Such doubly stochastic "first-order summaries" have been studied in various settings (Shin et al., 2003; Helmbold & Warmuth, 2007). In identity management (Shin et al., 2003)[2], first-order summaries maintain, for example,

$$P(\text{Alice is at Track 1}) = 2/3,$$
$$P(\text{Bob is at Track 3}) = 1/2.$$

What cannot be captured by first-order summaries however, are the higher order statements like:

$$P(\text{Alice is in Track 1 } \textit{and} \text{ Bob is in Track 2}) = 0.$$

Over the next two sections, we will show that the first-order summary of a distribution $P(\sigma)$ can equivalently be viewed as the lowest frequency coefficients of the Fourier transform of $P(\sigma)$, and that by considering higher frequencies,

[2] Strictly speaking, a map from identities to tracks is not a permutation since a permutation always maps a set into itself. In fact, the set of all such identity-to-track assignments does not actually form a group since there is no way to compose any two such assignments to obtain a legitimate group operation. We abuse the notation by referring to these assignments as a group, but really the elements of the group here should be thought of as the 'deviation' from the original identity-to-track assignment (where only the tracks are permuted, for example, when they are confused). In the group theoretic language, there is a faithful group action of S_n on the set of all identity-to-track assignments.

we can capture higher order marginal probabilities in a principled fashion. Furthermore, the Fourier theoretic perspective, as we will show, provides a natural framework for formulating inference operations with respect to our compact summaries. In a nutshell, we will view the prediction/rollup step as a convolution and the conditioning step as a pointwise product — then we will formulate the two inference operations in the Fourier domain as a pointwise product and convolution, respectively.

4 The Fourier transform on finite groups

Over the last fifty years, the Fourier Transform has been ubiquitously applied to everything digital, particularly with the invention of the Fast Fourier Transform. On the real line, the Fourier Transform is a well-studied method for decomposing a function into a sum of sine and cosine terms over a spectrum of frequencies. Perhaps less familiar though, is its group theoretic generalization, which we review in this section with an eye towards approximating functions on S_n. For further information, see (Diaconis, 1988) and (Terras, 1999).

4.1 Group representation theory

The generalized definition of the Fourier Transform relies on the theory of group representations, which formalize the concept of associating permutations with matrices and are used to construct a complete basis for the space of functions on a group G, thus also playing a role analogous to that of sinusoids on the real line.

Definition 2. A *representation* of a group G is a map ρ from G to a set of invertible $d_\rho \times d_\rho$ matrix operators which preserves algebraic structure in the sense that for all $\sigma_1, \sigma_2 \in G$, $\rho(\sigma_1\sigma_2) = \rho(\sigma_1) \cdot \rho(\sigma_2)$. The matrices which lie in the image of ρ are called the *representation matrices*, and we will refer to d_ρ as the *degree* of the representation.

The requirement that $\rho(\sigma_1\sigma_2) = \rho(\sigma_1) \cdot \rho(\sigma_2)$ is analogous to the property that $e^{i(\theta_1+\theta_2)} = e^{i\theta_1} \cdot e^{i\theta_2}$ for the conventional sinusoidal basis. Each matrix entry, $\rho_{ij}(\sigma)$ defines some function over S_n:

$$\rho(\sigma) = \begin{bmatrix} \rho_{11}(\sigma) & \rho_{12}(\sigma) & \cdots & \rho_{1d_\rho}(\sigma) \\ \rho_{21}(\sigma) & \rho_{22}(\sigma) & \cdots & \rho_{2d_\rho}(\sigma) \\ \vdots & \vdots & \ddots & \vdots \\ \rho_{d_\rho 1}(\sigma) & \rho_{d_\rho 2}(\sigma) & \cdots & \rho_{d_\rho d_\rho}(\sigma) \end{bmatrix}, \tag{4.1}$$

and consequently, each representation ρ simultaneously defines a set of d_ρ^2 functions over S_n. We will eventually think of group representations as the set of Fourier basis functions onto which we can project arbitrary functions.

Example 3. *We begin by showing three examples of representations on the symmetric group.*

1. *The simplest example of a representation is called the* trivial representation $\rho_{(n)} : S_n \to \mathbb{R}^{1\times 1}$*, which maps each element of the symmetric group to 1, the multiplicative identity on the real numbers. The trivial representation is actually defined for every group, and while it may seem unworthy of mention, it plays the role of the constant basis function in the Fourier theory.*

2. *The* first-order permutation representation *of S_n, which we alluded to in Example 1, is the degree n representation, $\tau_{(n-1,1)}$ (we explain the terminology in Section 5) , which maps a permutation σ to its corresponding permutation matrix given by $[\tau_{(n-1,1)}(\sigma)]_{ij} = \mathbb{1}\{\sigma(j) = i\}$. For example, the first-order permutation representation on S_3 is given by:*

$$\tau_{(2,1)}(\epsilon) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad \tau_{(2,1)}(1,2) = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\tau_{(2,1)}(2,3) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \qquad \tau_{(2,1)}(1,3) = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

$$\tau_{(2,1)}(1,2,3) = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \qquad \tau_{(2,1)}(1,3,2) = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

3. *The* alternating representation *of S_n, maps a permutation σ to the determinant of $\tau_{(n-1,1)}(\sigma)$, which is $+1$ if σ can be equivalently written as the composition of an even number of pairwise swaps, and -1 otherwise. We write the alternating representation as $\rho_{(1,\ldots,1)}$ with n 1's in the subscript. For example, on S_4, we have:*

$$\rho_{(1,1,1,1)}((1,2,3)) = \rho_{(1,1,1,1)}((13)(12)) = +1.$$

The alternating representation can be interpreted as the 'highest frequency' basis function on the symmetric group, intuitively due to its high sensitivity to swaps. For example, if $\tau_{(1,\ldots,1)}(\sigma) = 1$, then $\tau_{(1,\ldots,1)}((12)\sigma) = -1$. In identity management, it may be reasonable to believe that the joint probability over all n identity labels should only change by a little if just two objects are mislabeled due to swapping — in this case, ignoring the basis function corresponding to the alternating representation should still provide an accurate approximation to the joint distribution.

In general, a representation corresponds to an overcomplete set of functions and therefore does not constitute a valid basis for any subspace of functions. For example, the set of nine functions on S_3 corresponding to $\tau_{(2,1)}$ span only four dimensions, because there are six normalization constraints (three on the row sums and three on the column sums), of which five are independent — and so there are five redundant dimensions. To find a valid complete basis for the space of functions on S_n, we will need to find a family of representations whose basis functions are independent, and span the entire $n!$-dimensional space of functions.

In the following two definitions, we will provide two methods for constructing a new representation from old ones such that the set of functions on S_n corresponding to the new representation is linearly *dependent* on the old representations. Somewhat surprisingly, it can be shown that dependencies which arise amongst the representations can always be recognized in a certain sense, to come from the two possible following sources (Serre, 1977).

Definition 4.

1. **Equivalence.** Given a representation ρ_1 and an invertible matrix C, one can define a new representation ρ_2 by "changing the basis" for ρ_1:

$$\rho_2(\sigma) \triangleq C^{-1} \cdot \rho_1(\sigma) \cdot C. \tag{4.2}$$

   We say, in this case, that ρ_1 and ρ_2 are *equivalent* as representations (written $\rho_1 \equiv \rho_2$), and the matrix C is known as the *intertwining operator*. Note that $d_{\rho_1} = d_{\rho_2}$.

   It can be checked that the functions corresponding to ρ_2 can be reconstructed from those corresponding to ρ_1. For example, if C is a permutation matrix, the matrix entries of ρ_2 are exactly the same as the matrix entries of ρ_1, only permuted.

2. **Direct Sum.** Given two representations ρ_1 and ρ_2, we can always form a new representation, which we will write as $\rho_1 \oplus \rho_2$, by defining:

$$\rho_1 \oplus \rho_2(\sigma) \triangleq \left[\begin{array}{c|c} \rho_1(\sigma) & 0 \\ \hline 0 & \rho_2(\sigma) \end{array}\right].$$

   $\rho_1 \oplus \rho_2$ is called the *direct sum representation*. For example, the direct sum of two copies of the trivial representation is:

$$\rho_{(n)} \oplus \rho_{(n)}(\sigma) = \left[\begin{array}{cc} 1 & 0 \\ 0 & 1 \end{array}\right],$$

   with four corresponding functions on S_n, each of which is clearly dependent upon the trivial representation itself.

Most representations can be seen as being equivalent to a direct sum of strictly smaller representations. Whenever a representation ρ can be decomposed as $\rho \equiv \rho_1 \oplus \rho_2$, we say that ρ is *reducible*. As an example, we now show that the first-order permutation representation is a reducible representation.

Example 5. *Instead of using the standard basis vectors $\{e_1, e_2, e_3\}$, the first-order permutation representation $\tau_{(2,1)}$ can be equivalently written with respect* *to a new basis* $\{v_1, v_2, v_3\}$*, where:*

$$v_1 = \frac{e_1 + e_2 + e_3}{|e_1 + e_2 + e_3|},$$
$$v_2 = \frac{-e_1 + e_2}{|-e_1 + e_2|},$$
$$v_3 = \frac{-e_1 - e_2 + 2e_3}{|-e_1 - e_2 + 2e_3|}.$$

To 'change the basis', we write the new basis vectors as columns in a matrix C*:*

$$C = \begin{bmatrix} | & | & | \\ v_1 & v_2 & v_3 \\ | & | & | \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{3}} & -\frac{\sqrt{2}}{2} & -\frac{1}{\sqrt{6}} \\ \frac{1}{\sqrt{3}} & \frac{\sqrt{2}}{2} & -\frac{1}{\sqrt{6}} \\ \frac{1}{\sqrt{3}} & 0 & \frac{2}{\sqrt{6}} \end{bmatrix},$$

and conjugate the representation $\tau_{(2,1)}$ *by* C *(as in Equation 4.2) to obtain the equivalent representation* $C^{-1} \cdot \tau_{(2,1)}(\sigma) \cdot C$*:*

$$C^{-1}\tau_{(2,1)}(\epsilon)C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad C^{-1}\tau_{(2,1)}(1,2)C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$C^{-1}\tau_{(2,1)}(2,3)C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{2} & \frac{\sqrt{3}}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{1}{2} \end{bmatrix} \qquad C^{-1}\tau_{(2,1)}(1,3)C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{2} & -\frac{\sqrt{3}}{2} \\ 0 & -\frac{\sqrt{3}}{2} & -\frac{1}{2} \end{bmatrix}$$

$$C^{-1}\tau_{(2,1)}(1,2,3)C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -\frac{1}{2} & -\frac{\sqrt{3}}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{1}{2} \end{bmatrix} \qquad C^{-1}\tau_{(2,1)}(1,3,2)C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -\frac{1}{2} & \frac{\sqrt{3}}{2} \\ 0 & -\frac{\sqrt{3}}{2} & -\frac{1}{2} \end{bmatrix}$$

The interesting property of this particular basis is that the new representation matrices all appear to be the direct sum of two smaller representations, a trivial representation, $\rho_{(3)}$ *as the top left block, and a degree 2 representation in the bottom right which we will refer to as* $\rho_{(2,1)}$*.*

Geometrically, the representation $\rho_{(2,1)}$ *can also be thought of as the group of rigid symmetries of the equilateral triangle with vertices:*

$$P_1 = \begin{bmatrix} \sqrt{3}/2 \\ 1/2 \end{bmatrix}, P_2 = \begin{bmatrix} -\sqrt{3}/2 \\ 1/2 \end{bmatrix}, P_3 = \begin{bmatrix} 0 \\ -1 \end{bmatrix}.$$

The matrix $\rho_{(2,1)}(1,2)$ *acts on the triangle by reflecting about the* x*-axis, and* $\rho_{(2,1)}(1,2,3)$ *by a* $\pi/3$ *counter-clockwise rotation.*

In general, there are infinitely many reducible representations. For example, given any dimension d, there is a representation which maps every element of a group G to the $d \times d$ identity matrix (the direct sum of d copies of the trivial representation). However, for any finite group, there exists a finite collection of

σ	$\rho_{(3)}$	$\rho_{(2,1)}$	$\rho_{(1,1,1)}$
ϵ	1	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$	1
$(1,2)$	1	$\begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$	-1
$(2,3)$	1	$\begin{bmatrix} 1/2 & \sqrt{3}/2 \\ \sqrt{3}/2 & -1/2 \end{bmatrix}$	-1
$(1,3)$	1	$\begin{bmatrix} 1/2 & -\sqrt{3}/2 \\ -\sqrt{3}/2 & -1/2 \end{bmatrix}$	-1
$(1,2,3)$	1	$\begin{bmatrix} -1/2 & -\sqrt{3}/2 \\ \sqrt{3}/2 & -1/2 \end{bmatrix}$	1
$(1,3,2)$	1	$\begin{bmatrix} -1/2 & \sqrt{3}/2 \\ -\sqrt{3}/2 & -1/2 \end{bmatrix}$	1

Table 2: The irreducible representation matrices of S_3.

atomic representations which can be used to build up any other representation using the direct sum operation. These representations are referred to as the *irreducibles* of a group, and they are defined simply to be the collection of representations (up to equivalence) which are not reducible. It can be shown that any representation of a finite group G is equivalent to a direct sum of irreducibles (Diaconis, 1988), and hence, for any representation τ, there exists a matrix C for which

$$C^{-1} \cdot \tau \cdot C = \bigoplus_{\rho} \bigoplus_{j=1}^{z_\rho} \rho,$$

where ρ ranges over all distinct irreducible representations of the group G, and the inner $\oplus$ refers to some finite number (z_ρ) of copies of each irreducible ρ.

As it happens, there are only three irreducible representations of S_3 (Diaconis, 1988), the trivial representation $\rho_{(3)}$, the degree 2 representation $\rho_{(2,1)}$, and the alternating representation $\rho_{(1,1,1)}$. The complete set of irreducible representation matrices of S_3 are shown in the Table 2. Unfortunately, the analysis of the irreducible representations for $n > 3$ is far more complicated and we postpone this more general discussion for Section 5.

4.2 The Fourier transform

The link between group representation theory and Fourier analysis is given by the celebrated *Peter-Weyl theorem* ((Diaconis, 1988; Terras, 1999; Sagan, 2001)) which says that the matrix entries of the irreducibles of G form a *complete* set of *orthogonal* basis functions on G.[3] The space of functions on S_3, for example, is orthogonally spanned by the 3! functions $\rho_{(3)}(\sigma)$, $[\rho_{(2,1)}(\sigma)]_{1,1}$, $[\rho_{(2,1)}(\sigma)]_{1,2}$, $[\rho_{(2,1)}(\sigma)]_{2,1}$, $[\rho_{(2,1)}(\sigma)]_{2,2}$ and $\rho_{(1,1,1)}(\sigma)$, where $[\rho(\sigma)]_{ij}$ denotes the (i,j) entry of the matrix $\rho(\sigma)$.

As a replacement for projecting a function f onto a complete set of sinusoidal basis functions (as one would do on the real line), the Peter-Weyl theorem suggests instead to project onto the basis provided by the irreducibles of G. As on the real line, this projection can be done by computing the inner product of f with each element of the basis, and we define this operation to be the generalized form of the Fourier Transform.

Definition 6. Let $f : G \to \mathbb{R}$ be any function on a group G and let ρ be any representation on G. The *Fourier Transform* of f at the representation ρ is defined to be the matrix of coefficients:

$$\hat{f}_\rho = \sum_\sigma f(\sigma)\rho(\sigma). \tag{4.3}$$

The collection of Fourier Transforms at all irreducible representations of G form *the Fourier Transform of f.*

There are two important points which distinguish this Fourier Transform from its familiar formulation on the real line — first, the outputs of the transform are matrix-valued, and second, the inputs to $\hat{f}$ are *representations* of G rather than real numbers. As in the familiar formulation, the Fourier Transform is invertible and the inversion formula is explicitly given by the Fourier Inversion Theorem.

Theorem 7 (Fourier Inversion Theorem)**.**

$$f(\sigma) = \frac{1}{|G|} \sum_\lambda d_{\rho_\lambda} \, Tr\left[\hat{f}^T_{\rho_\lambda} \cdot \rho_\lambda(\sigma)\right], \tag{4.4}$$

where λ indexes over the collection of irreducibles of G.

Note that the trace term in the inverse Fourier Transform is just the 'matrix dot product' between $\hat{f}_{\rho_\lambda}$ and $\rho_\lambda(\sigma)$, since $\mathrm{Tr}\left[A^T \cdot B\right] = \langle \mathrm{vec}(A), \mathrm{vec}(B)\rangle$, where by vec we mean mapping a matrix to a vector on the same elements arranged in column-major order.

We now provide several examples for intuition. For functions on the real line, the Fourier Transform at zero frequency gives the DC component of a signal. The same holds true for functions on a group; If $f : G \to \mathbb{R}$ is any function,

[3]Technically the Peter-Weyl result, as stated here, is only true if all of the representation matrices are unitary. That is, $\rho(\sigma)^*\rho(\sigma) = I$ for all $\sigma \in S_n$, where the matrix A^* is the conjugate transpose of A. For the case of real-valued (as opposed to complex-valued) matrices, however, the definitions of unitary and orthogonal matrices coincide.

While most representations are not unitary, there is a standard result from representation theory which shows that for *any* representation of G, there exists an equivalent unitary representation.

then since $\rho_{(n)} = 1$, the Fourier Transform of f at the trivial representation is constant, with $\hat{f}_{\rho_{(n)}} = \sum_\sigma f(\sigma)$. Thus, for any probability distribution P, we have $\hat{P}_{\rho_{(n)}} = 1$. If P were the uniform distribution, then $\hat{P}_\rho = 0$ at every irreducible ρ except at the trivial representation.

The Fourier Transform at $\tau_{(n-1,1)}$ also has a simple interpretation:

$$[\hat{f}_{\tau_{(n-1,1)}}]_{ij} = \sum_{\sigma \in S_n} f(\sigma)[\tau_{(n-1,1)}(\sigma)]_{ij} = \sum_{\sigma \in S_n} f(\sigma) \mathbb{1}\left\{\sigma(j) = i\right\} = \sum_{\sigma:\sigma(j)=i} f(\sigma).$$

The set $\Delta_{ij} = \{\sigma : \sigma(j) = i\}$ is the set of the $(n-1)!$ possible permutations which map element j to i. In identity management, Δ_{ij} can be thought of as the set of assignments which, for example, have Alice at Track 1. If P is a distribution, then $\hat{P}_{\tau_{(n-1,1)}}$ is a matrix of *first-order* marginal probabilities, where the (i,j)-th element is the marginal probability that a random permutation drawn from P maps element j to i.

Example 8. *Consider the following probability distribution on S_3:*

σ	ϵ	$(1,2)$	$(2,3)$	$(1,3)$	$(1,2,3)$	$(1,3,2)$
$P(\sigma)$	$1/3$	$1/6$	$1/3$	0	$1/6$	0

The set of all first order marginal probabilities is given by the Fourier transform at $\tau_{(2,1)}$:

$$\widehat{P}_{\tau_{(2,1)}} = \left[\begin{array}{c|ccc} & \mathbf{A} & \mathbf{B} & \mathbf{C} \\ \hline \mathbf{1} & 2/3 & 1/6 & 1/6 \\ \mathbf{2} & 1/3 & 1/3 & 1/3 \\ \mathbf{3} & 0 & 1/2 & 1/2 \end{array}\right].$$

In the above matrix, each column j represents a marginal distribution over the possible tracks that identity j can map to under a random draw from P. We see, for example, that Alice is at Track 1 with probability 2/3, or at Track 2 with probability 1/3. Simultaneously, each row i represents a marginal distribution over the possible identities that could have been mapped to track i under a random draw from P. In our example, Bob and Cathy are equally likely to be in Track 3, but Alice is definitely not in Track 3. Since each row and each column is itself a distribution, the matrix $\widehat{P}_{\tau_{(2,1)}}$ must be doubly stochastic. We will elaborate on the consequences of this observation later.

The Fourier transform of the same distribution at all irreducibles is:

$$\widehat{P}_{\rho_{(3)}} = 1, \quad \widehat{P}_{\rho_{(2,1)}} = \begin{bmatrix} 1/4 & \sqrt{3}/4 \\ \sqrt{3}/4 & 1/4 \end{bmatrix}, \quad \widehat{P}_{\rho_{(1,1,1)}} = 0.$$

The first-order permutation representation, $\tau_{(n-1,1)}$, captures the statistics of how a random permutation acts on a single object irrespective of where all of the other $n-1$ objects are mapped, and in doing so, compactly summarizes the distribution with only $O(n^2)$ numbers. Unfortunately, as mentioned in

Section 3, the Fourier transform at the first-order permutation representation cannot capture more complicated statements like:

$$P(\text{Alice and Bob occupy Tracks 1 and 2}) = 0.$$

To avoid collapsing away so much information, we might define richer summary statistics that might capture 'higher-order' effects. We define the *second-order unordered permutation representation* by:

$$[\tau_{(n-2,2)}(\sigma)]_{\{i,j\},\{k,\ell\}} = \mathbb{1}\left\{\sigma(\{k,\ell\}) = \{i,j\}\right\},$$

where we index the matrix rows and columns by unordered pairs $\{i,j\}$. The condition inside the indicator function states that the representation captures whether the pair of objects $\{k,\ell\}$ maps to the pair $\{i,j\}$, but is indifferent with respect to the ordering; i.e., either $k \mapsto i$ and $\ell \mapsto j$, *or*, $k \mapsto j$ and $\ell \mapsto i$.

Example 9. *For $n = 4$, there are six possible unordered pairs: $\{1,2\}$,$\{1,3\}$,$\{1,4\}$,$\{2,3\}$,$\{2,4\}$, and $\{3,4\}$. The matrix representation of the permutation $(1,2,3)$ is:*

$$\tau_{(2,2)}(1,2,3) = \left[\begin{array}{c|cccccc} & \{1,2\} & \{1,3\} & \{1,4\} & \{2,3\} & \{2,4\} & \{3,4\} \\ \hline \{1,2\} & 0 & 0 & 0 & 1 & 0 & 0 \\ \{1,3\} & 1 & 0 & 0 & 0 & 0 & 0 \\ \{1,4\} & 0 & 0 & 0 & 0 & 1 & 0 \\ \{2,3\} & 0 & 1 & 0 & 0 & 0 & 0 \\ \{2,4\} & 0 & 0 & 0 & 0 & 0 & 1 \\ \{3,4\} & 0 & 0 & 1 & 0 & 0 & 0 \end{array}\right].$$

The *second order ordered permutation representation*, $\tau_{(n-2,1,1)}$, is defined similarly:

$$[\tau_{(n-2,1,1)}(\sigma)]_{(i,j),(k,\ell)} = \mathbb{1}\left\{\sigma((k,\ell)) = (i,j)\right\},$$

where (k,ℓ) denotes an *ordered* pair. Therefore, $[\tau_{(n-2,1,1)}(\sigma)]_{(i,j),(k,\ell)}$ is 1 if and only if σ maps k to i *and* ℓ to j.

As in the first-order case, the Fourier transform of a probability distribution at $\tau_{(n-2,2)}$, returns a matrix of marginal probabilities of the form: $P(\sigma : \sigma(\{k,\ell\}) = \{i,j\})$, which captures statements like, "Alice and Bob occupy Tracks 1 and 2 with probability 1/2". Similarly, the Fourier transform at $\tau_{(n-2,1,1)}$ returns a matrix of marginal probabilities of the form $P(\sigma : \sigma((k,\ell)) = (i,j))$, which captures statements like, "Alice is in Track 1 *and* Bob is in Track 2 with probability 9/10".

We can go further and define third-order representations, fourth-order representations, and so on. In general however, the permutation representations as they have been defined above are reducible, intuitively due to the fact that it is possible to recover lower order marginal probabilities from higher order marginal probabilities. For example, one can recover the normalization constant (corresponding to the trivial representation) from the first order matrix of marginals by summing across either the rows or columns, and the first order marginal probabilities from the second order marginal probabilities by summing across appropriate matrix entries. To truly leverage the machinery of Fourier analysis, it is important to understand the Fourier transform at the irreducibles of the symmetric group, and in the next section, we show how to derive the irreducible representations of the Symmetric group by first defining permutation representations, then "subtracting off the lower-order effects".

5 Representation theory on the Symmetric group

In this section, we provide a brief introduction to the representation theory of the Symmetric group. Rather than giving a fully rigorous treatment of the subject, our goal is to give some intuition about the kind of information which can be captured by the irreducible representations of S_n. Roughly speaking, we will show that Fourier transforms on the Symmetric group, instead of being indexed by frequencies, are indexed by *partitions* of n (tuples of numbers which sum to n), and certain partitions correspond to more complex basis functions than others. For proofs, we point the reader to consult: (Diaconis, 1989; James & Kerber, 1981; Sagan, 2001; Vershik & Okounkov, 2006).

Instead of the singleton or pairwise marginals which were described in the previous section, we will now focus on using the Fourier coefficients of a distribution to query a much wider class of marginal probabilities. As an example, we will be able to compute the following (more complicated) marginal probability on S_6 using Fourier coefficients:

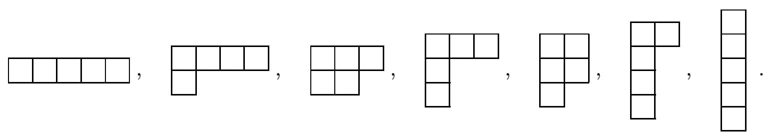
$$P\left(\sigma : \sigma\left(\left\{\begin{array}{|c|c|c|}\hline 1&2&3\\\hline 4&5\\\cline{1-2} 6\\\cline{1-1}\end{array}\right\}\right) = \left\{\begin{array}{|c|c|c|}\hline 1&2&6\\\hline 4&5\\\cline{1-2} 3\\\cline{1-1}\end{array}\right\}\right), \tag{5.1}$$

which we interpret as the joint marginal probability that the rows of the diagram on the left map to corresponding rows on the right as unordered sets. In other words, Equation 5.1 is the joint probability that *unordered* set $\{1, 2, 3\}$ maps to $\{1, 2, 6\}$, the unordered pair $\{4, 5\}$ maps to $\{4, 5\}$, and the singleton $\{6\}$ maps to $\{3\}$.

The diagrams in Equation 5.1 are known as *Ferrer's diagrams* and are commonly used to visualize *partitions* of n, which are defined to be unordered tuples of positive integers, $\lambda = (\lambda_1, \ldots, \lambda_\ell)$, which sum to n. For example, $\lambda = (3, 2)$ is a partition of $n = 5$ since $3 + 2 = 5$. Usually we write partitions as weakly decreasing sequences by convention, so the partitions of $n = 5$ are:

$$(5), \ (4,1), \ (3,2), \ (3,1,1), \ (2,2,1), \ (2,1,1,1), \ (1,1,1,1,1),$$

and their respective Ferrers diagrams are:



A *Young tabloid* is an assignment of the numbers $\{1, \ldots, n\}$ to the boxes of a Ferrers diagram for a partition λ, where each row represents an unordered set. There are 6 Young tabloids corresponding to the partition $\lambda = (2, 2)$, for example:

$$\left\{\begin{array}{|c|c|}\hline 1&2\\\hline 3&4\\\hline\end{array}\right\}, \left\{\begin{array}{|c|c|}\hline 1&3\\\hline 2&4\\\hline\end{array}\right\}, \left\{\begin{array}{|c|c|}\hline 1&4\\\hline 2&3\\\hline\end{array}\right\}, \left\{\begin{array}{|c|c|}\hline 2&3\\\hline 1&4\\\hline\end{array}\right\}, \left\{\begin{array}{|c|c|}\hline 2&4\\\hline 1&3\\\hline\end{array}\right\}, \left\{\begin{array}{|c|c|}\hline 3&4\\\hline 1&2\\\hline\end{array}\right\}.$$

The Young tabloid, $\begin{array}{|c|c|}\hline 1&2\\\hline 3&4\\\hline\end{array}$, for example, represents the two underordered sets $\{1, 2\}$ and $\{3, 4\}$, and if we were interested in computing the joint probability that $\sigma(\{1,2\}) = \{3,4\}$ and $\sigma(\{3,4\}) = \{1,2\}$, then we could write the problem in terms of Young tabloids as:

$$P\left(\sigma\ :\ \sigma\left(\left\{\begin{array}{|c|c|}\hline 1&2\\\hline 3&4\\\hline\end{array}\right\}\right) = \left\{\begin{array}{|c|c|}\hline 3&4\\\hline 1&2\\\hline\end{array}\right\}\right).$$

In general, we will be able to use the Fourier coefficients at irreducible representations to compute the marginal probabilities of Young tabloids. As we shall see, with the help of the *James Submodule theorem* (James & Kerber, 1981), the marginals corresponding to "simple" partitions will require very few Fourier coefficients to compute, which is one of the main strengths of working in the Fourier domain.

Example 10. *Imagine three separate rooms containing two tracks each, in which Alice and Bob are in room 1 occupying Tracks 1 and 2; Cathy and David are in room 2 occupying Tracks 3 and 4; and Eric and Frank are in room 3 occupying Tracks 5 and 6, but we are not able to distinguish which person is at which track in any of the rooms. Then*

$$P\left(\sigma\ :\ \left(\left\{\begin{array}{|c|c|}\hline A&B\\\hline C&D\\\hline E&F\\\hline\end{array}\right\}\right) \rightarrow \left\{\begin{array}{|c|c|}\hline 1&2\\\hline 3&4\\\hline 5&6\\\hline\end{array}\right\}\right) = 1.$$

It is in fact, possible to recast the first-order marginals which were described in the previous section in the language of Young tabloids by noticing that, for example, if 1 maps to 1, then the unordered set $\{2,\ldots,n\}$ must map to $\{2,\ldots,n\}$ since permutations are one-to-one mappings. The marginal probability that $\sigma(1) = 1$, then, is equal to the marginal probability that $\sigma(1) = 1$ *and* $\sigma(\{2,\ldots,n\}) = \{2,\ldots,n\}$. If $n = 6$, then the marginal probability written using Young tabloids is:

$$P\left(\sigma\ :\ \sigma\left(\left\{\begin{array}{|c|c|c|c|c|}\hline 2&3&4&5&6\\\hline 1\\\cline{1-1}\end{array}\right\}\right) = \left\{\begin{array}{|c|c|c|c|c|}\hline 2&3&4&5&6\\\hline 1\\\cline{1-1}\end{array}\right\}\right).$$

The first-order marginal probabilities correspond, therefore, to the marginal probabilities of Young tabloids of shape $\lambda = (n-1,1)$.

Likewise, the second-order unordered marginals correspond to Young tabloids of shape $\lambda = (n-2,2)$. If $n = 6$ again, then the marginal probability that $\{1,2\}$ maps to $\{2,4\}$ corresponds to the following marginal probability for tabloids:

$$P\left(\sigma\ :\ \sigma\left(\left\{\begin{array}{|c|c|c|c|}\hline 3&4&5&6\\\hline 1&2\\\cline{1-2}\end{array}\right\}\right) = \left\{\begin{array}{|c|c|c|c|}\hline 1&3&5&6\\\hline 2&4\\\cline{1-2}\end{array}\right\}\right).$$

The second-order *ordered* marginals are captured at the partition $\lambda = (n-2,1,1)$. For example, the marginal probability that $\{1\}$ maps to $\{2\}$ *and* $\{2\}$ maps to $\{4\}$ is given by:

$$P\left(\sigma\ :\ \sigma\left(\left\{\begin{array}{|c|c|c|c|}\hline 3&4&5&6\\\hline 1\\\cline{1-1} 2\\\cline{1-1}\end{array}\right\}\right) = \left\{\begin{array}{|c|c|c|c|}\hline 1&3&5&6\\\hline 2\\\cline{1-1} 4\\\cline{1-1}\end{array}\right\}\right).$$

And finally, we remark that the $(1, \ldots, 1)$ partition of n recovers all original probabilities since it asks for a joint distribution over $\sigma(1), \ldots, \sigma(n)$. The corresponding matrix of marginals has $n! \times n!$ entries.

To see how the marginal probabilities of Young tabloids of shape λ can be thought of as Fourier coefficients, we will define a representation (which we call the *permutation representation*) associated with λ and show that the Fourier transform of a distribution at a permutation representation gives marginal probabilities. We begin by fixing an ordering on the set of possible Young tabloids, $\{t_1\}, \{t_2\}, \ldots$, and define the permutation representation $\tau_\lambda(\sigma)$ to be the matrix:

$$[\tau_\lambda(\sigma)]_{ij} = \begin{cases} 1 & \text{if } \sigma(\{t_j\}) = \{t_i\} \\ 0 & \text{otherwise} \end{cases}. \tag{5.2}$$

It can be checked that the function τ_λ is indeed a valid representation of the Symmetric group, and therefore we can compute Fourier coefficients at τ_λ. If $P(\sigma)$ is a probability distribution, then

$$\begin{aligned} \left[\widehat{P}_{\tau_\lambda}\right]_{ij} &= \sum_{\sigma \in S_n} P(\sigma) \left[\tau_\lambda(\sigma)\right]_{ij}, \\ &= \sum_{\{\sigma \,:\, \sigma(\{t_j\}) = \{t_i\}\}} P(\sigma), \\ &= P(\sigma \,:\, \sigma(\{t_j\}) = \{t_i\}), \end{aligned}$$

and therefore, *the matrix of marginals corresponding to Young tabloids of shape λ is given* exactly *by the Fourier transform at the representation τ_λ*.

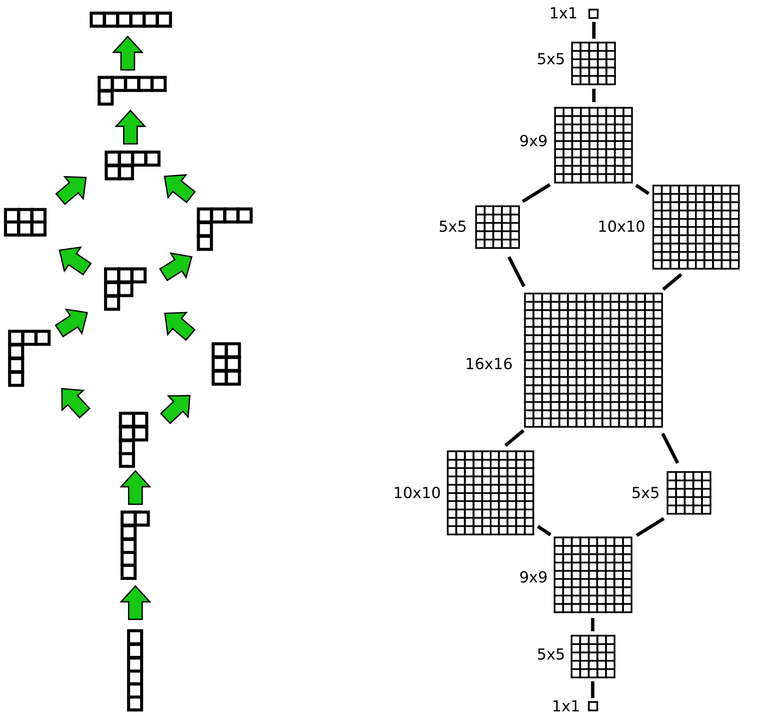
As we showed earlier, the simplest marginals (the zeroth order normalization constant), correspond to the Fourier transform at $\tau_{(n)}$, while the first-order marginals correspond to $\tau_{(n-1,1)}$, and the second-order unordered marginals correspond to $\tau_{(n-2,2)}$. The list goes on and on, with the marginals getting more complicated; At the other end of the spectrum, we have the Fourier coefficients at the representation $\tau_{(1,1,\ldots,1)}$ which exactly recover the original probabilities $P(\sigma)$.

We use the word 'spectrum' suggestively here, because the different levels of complexity for the marginals are highly reminiscent of the different frequencies for real-valued signals, and a natural question to ask is how the partitions might be ordered with respect to the 'complexity' of the corresponding basis functions. In particular how might one characterize this vague notion of complexity for a given partition?

The 'correct' characterization, as it turns out, is to use the *dominance ordering* of partitions, which, unlike the ordering on frequencies, is not a linear order, but rather, a partial order.

Definition 11 (Dominance Ordering)**.** Let λ, μ be partitions of n. Then $\lambda \trianglerighteq \mu$ (we say λ *dominates* μ), if for each i, $\sum_{k=1}^{i} \lambda_k \geq \sum_{k=1}^{i} \mu_k$.

For example, $(4,2) \trianglerighteq (3,2,1)$ since $4 \geq 3$, $4+2 \geq 3+2$, and $4+2+0 \geq 3+2+1$. However, $(3,3)$ and $(4,1,1)$ cannot be compared with respect to the dominance



(a) Dominance ordering for $n = 6$.

(b) Fourier coefficient matrices for S_6.

Figure 3: The dominance order for partitions of $n = 6$ are shown in the left diagram (a). Fat Ferrer's diagrams tend to be higher in the order and long, skinny diagrams tend to be lower. The corresponding Fourier coefficient matrices for each partition (at irreducible representations) are shown in the right diagram (b). Note that since the Fourier basis functions form a complete basis for the space of functions on the Symmetric group, there must be exactly $n!$ coefficients in total.

ordering since $3 \leq 4$, but $3 + 3 \geq 4 + 1$. The ordering over the partitions of $n = 6$ is depicted in Figure 3(a).

Partitions with fat Ferrers diagrams tend to be greater (with respect to dominance ordering) than those with skinny Ferrers diagrams. Intuitively, representations corresponding to partitions which are *high* in the dominance ordering are 'low frequency', while represensions corresponding to partitions which are *low* in the dominance ordering are 'high frequency'[4].

Having defined a family of intuitive permutation representations over the Symmetric group, we can now ask whether the permutation representations are irreducible or not: the answer in general, is to the negative, due to the fact that

[4]The direction of the ordering is slightly counterintuitive given the frequency interpretation, but is standard in the literature.

it is often possible to reconstruct lower order marginals by summing over the appropriate higher order marginal probabilities. However, it is possible to show that, for each permutation representation τ_λ, there exists a corresponding irreducible representation ρ_λ, which, loosely, captures all of the information at the 'frequency' λ which was not already captured at lower frequency irreducibles. Moreover, it can be shown that there exists no irreducible representation besides those indexed by the partitions of n. These remarkable results are formalized in the *James Submodule Theorem*, which we state here without proof (see (Diaconis, 1988; James & Kerber, 1981; Sagan, 2001)).

Theorem 12 (James' Submodule Theorem)**.**

1. *(Uniqueness) For each partition, λ, of n, there exists an irreducible representation, ρ_λ, which is unique up to equivalence.*
2. *(Completeness) Every irreducible representation of S_n corresponds to some partition of n.*
3. *There exists a matrix C_λ associated with each partition λ, for which*

$$C_\lambda^T \cdot \tau_\lambda(\sigma) \cdot C_\lambda = \bigoplus_{\mu \trianglerighteq \lambda} \bigoplus_{\ell=1}^{K_{\lambda\mu}} \rho_\mu(\sigma), \quad \textit{for all } \sigma \in S_n. \tag{5.3}$$

4. *$K_{\lambda\lambda} = 1$ for all partitions λ.*

In plain English, part (3) of the James Submodule theorem says that we can always reconstruct marginal probabilities of λ-tabloids using the Fourier coefficients at irreducibles which lie *at λ and above* in the dominance ordering, if we have knowledge of the matrix C_λ (which can be precomputed using methods detailed in Appendix B), and the multiplicities $K_{\lambda\mu}$. In particular, combining Equation 5.3 with the definition of the Fourier transform, we have that

$$\hat{f}_{\tau_\lambda} = C_\lambda \cdot \left[\bigoplus_{\mu \trianglerighteq \lambda} \bigoplus_{\ell=1}^{K_{\lambda\mu}} \hat{f}_{\rho_\mu} \right] \cdot C_\lambda^T, \tag{5.4}$$

and so to obtain marginal probabilities of λ-tabloids, we simply construct a block diagonal matrix using the appropriate irreducible Fourier coefficients, and conjugate by C_λ. The multiplicities $K_{\lambda\mu}$ are known as the *Kostka numbers* and can be computed using Young's rule (Sagan, 2001). To illustrate using a few examples, we have the following decompositions:

$$\begin{aligned}
\tau_{(n)} &\equiv \rho_{(n)}, \\
\tau_{(n-1,1)} &\equiv \rho_{(n)} \oplus \rho_{(n-1,1)}, \\
\tau_{(n-2,2)} &\equiv \rho_{(n)} \oplus \rho_{(n-1,1)} \oplus \rho_{(n-2,2)}, \\
\tau_{(n-2,1,1)} &\equiv \rho_{(n)} \oplus \rho_{(n-1,1)} \oplus \rho_{(n-1,1)} \oplus \rho_{(n-2,2)} \oplus \rho_{(n-2,1,1)}, \\
\tau_{(n-3,3)} &\equiv \rho_{(n)} \oplus \rho_{(n-1,1)} \oplus \rho_{(n-2,2)} \oplus \rho_{(n-3,3)}, \\
\tau_{(n-3,2,1)} &\equiv \rho_{(n)} \oplus \rho_{(n-1,1)} \oplus \rho_{(n-1,1)} \oplus \rho_{(n-2,2)} \oplus \rho_{(n-2,2)} \\
&\qquad \oplus \rho_{(n-2,1,1)} \oplus \rho_{(n-3,3)} \oplus \rho_{(n-3,2,1)}.
\end{aligned}$$

Intuitively, the irreducibles at a partition λ reflect the "pure" λ^{th}-order effects of the underlying distribution. In other words, the irreducibles at λ form a basis for functions that have "interesting" λ^{th}-order marginal probabilities, but uniform marginals at all partitions μ such that $\mu \rhd \lambda$.

Example 13. *As an example, we demonstrate a "preference" function which is "purely" second-order (unordered) in the sense that its Fourier coefficients are equal to zero at all irreducible representations except $\rho_{(n-2,2)}$ (and the trivial representation). Consider the function $f : S_n \to \mathbb{R}$ defined by:*

$$f(\sigma) = \begin{cases} 1 & \text{if } |\sigma(1) - \sigma(2)| \equiv 1 \ (mod\, n) \\ 0 & \text{otherwise} \end{cases} .$$

Intuitively, imagine seating n people at a round table with n chairs, but with the constraint that the first two people, Alice and Bob, are only happy if they are allowed to sit next to each other. In this case, f can be thought of as the indicator function for the subset of seating arrangements (permutations) which make Alice and Bob happy.

Since f depends only on the destination of the unordered pair $\{1, 2\}$, its Fourier transform is zero at all partitions μ such that $\mu \lhd (n-2, 2)$ ($\hat{f}_\mu = 0$). On the other hand, Alice and Bob have no individual preferences for seating, so the first-order "marginals" of f are uniform, and hence, $\hat{f}_{(n-1,1)} = 0$. The Fourier coefficients at irreducibles can be obtained from the second-order (unordered) "marginals" using Equation 5.3.

$$C^T_{(n-2,2)} \cdot \hat{P}_{\tau_{(n-2,2)}} \cdot C_{(n-2,2)} = \begin{bmatrix} \boxed{Z} & & \\ & \boxed{\quad 0 \quad} & \\ & & \boxed{\quad \hat{f}_{\rho_{(n-2,2)}} \quad} \end{bmatrix}.$$

The sizes of the irreducible representation matrices are typically much smaller than their corresponding permutation representation matrices. In the case of $\lambda = (1, \ldots, 1)$ for example, dim $\tau_\lambda = n!$ while dim $\rho_\lambda = 1$. There is a simple combinatorial algorithm, known as the *Hook Formula* (Sagan, 2001), for computing the dimension of ρ_λ. While we do not discuss it, we provide a few dimensionality computations here (Table 3) to facilitate a dicussion of complexity later. See Figure 3(b) for an example of what the matrices of a complete Fourier transform on S_6 would look like.

In practice, since the irreducible representation matrices are determined only up to equivalence, it is necessary to choose a basis for the irreducible representations in order to explicitly construct the representation matrices. As in (Kondor et al., 2007), we use the *Gel'fand-Tsetlin basis* which has several attractive properties, two advantages being that the matrices are real-valued and orthogonal. See Appendix A for details on constructing irreducible matrix representations with respect to the Gel'fand-Tsetlin basis.

λ	(n)	$(n-1,1)$	$(n-2,2)$	$(n-2,1,1)$	$(n-3,3)$	$(n-3,2,1)$
dim ρ_λ	1	$n-1$	$\frac{n(n-3)}{2}$	$\frac{(n-1)(n-2)}{2}$	$\frac{n(n-1)(n-5)}{6}$	$\frac{n(n-2)(n-4)}{3}$

Table 3: Dimensions of low-order irreducible representation matrices.

6 Inference in the Fourier domain

What we have shown thus far, is that there is a principled method for compactly summarizing distributions over permutations based on the idea of bandlimiting — saving only the low-frequency terms of the Fourier transform of a function, which, as we discussed, is equivalent to maintaining a set of low-order marginal probabilities. We now turn to the problem of performing probabilistic inference using our compact summaries. One of the main advantages of viewing marginals as Fourier coefficients is that it provides a natural principle for formulating inference, which is to rewrite all inference related operations with respect to the Fourier domain.

The idea of bandlimiting a distribution is ultimately moot, however, if it becomes necessary to transform back to the primal domain each time an inference operation is called. Naively, the Fourier Transform on S_n scales as $O((n!)^2)$, and even the fastest Fast Fourier Transforms for functions on S_n are no faster than $O(n^2 \cdot n!)$ (see (Maslen, 1998) for example). To resolve this issue, we present a formulation of inference which operates solely in the Fourier domain, allowing us to avoid a costly transform. We begin by discussing exact inference in the Fourier domain, which is no more tractable than the original problem because there are $n!$ Fourier coefficients, but it will allow us to introduce the bandlimiting approximation in the next section. There are two operations to consider: prediction/rollup, and conditioning. The assumption for the rest of this section is that the Fourier transforms of the transition and observation models are known. We discuss methods for obtaining the models in Section 8.

6.1 Fourier prediction/rollup

We will consider one particular class of transition models — that of random walks over a group, which assumes that $\sigma^{(t+1)}$ is generated from $\sigma^{(t)}$ by drawing a random permutation $\pi^{(t)}$ from some distribution $Q^{(t)}$ and setting $\sigma^{(t+1)} = \pi^{(t)}\sigma^{(t)}$ [5]. In our identity management example, $\pi^{(t)}$ represents a random identity permutation that might occur among tracks when they get close to each

[5] We place π on the left side of the multiplication because we want it to permute tracks and not identities. Had we defined π to map from tracks to identities (instead of identities to tracks), then π would be multiplied from the right. Besides left versus right multiplication, there are no differences between the two conventions.

other (what we call a *mixing event*). For example, $Q(1,2) = 1/2$ means that Tracks 1 and 2 swapped identities with probability $1/2$. The random walk model also appears in many other applications such as modeling card shuffles (Diaconis, 1988).

The motivation behind the random walk transition model is that it allows us to write the prediction/rollup operation as a *convolution* of distributions on the Symmetric group. The extension of the familiar notion of convolution to groups simple replaces additions and subtractions by analogous group operations (function composition and inverse, respectively):

Definition 14. Let Q and P be probability distributions on S_n. Define the *convolution*[6] of Q and P to be the function $[Q * P](\sigma_1) = \sum_{\sigma_2} Q(\sigma_1\sigma_2^{-1})P(\sigma_2)$.

Using Definition 14, we see that the prediction/rollup step can be written as:

$$\begin{aligned} P(\sigma^{(t+1)}) &= \sum_{\sigma^{(t)}} P(\sigma^{(t+1)}|\sigma^{(t)}) \cdot P(\sigma^{(t)}), \\ &= \sum_{\{(\sigma^{(t)},\pi^{(t)})\,:\,\sigma^{(t+1)}=\pi^{(t)}\cdot\sigma^{(t)}\}} Q^{(t)}(\pi^{(t)}) \cdot P(\sigma^{(t)}), \\ &\qquad \text{(Right-multiplying both sides of } \sigma^{(t+1)} = \pi^{(t)}\sigma^{(t)} \\ &\qquad\quad \text{by } (\sigma^{(t)})^{-1}\text{, we see that } \pi^{(t)} \text{ can be replaced by } \sigma^{(t+1)}(\sigma^{(t)})^{-1}), \\ &= \sum_{\sigma^{(t)}} Q^{(t)}(\sigma^{(t+1)} \cdot (\sigma^{(t)})^{-1}) \cdot P(\sigma^{(t)}), \\ &= \left[Q^{(t)} * P\right](\sigma^{(t+1)}). \end{aligned}$$

As with Fourier transforms on the real line, the Fourier coefficients of the convolution of distributions P and Q on groups can be obtained from the Fourier coefficients of P and Q individually, using the *convolution theorem* (see also (Diaconis, 1988)):

Proposition 15 (Convolution Theorem)**.** *Let Q and P be probability distributions on S_n. For any representation ρ,*

$$\left[\widehat{Q * P}\right]_\rho = \widehat{Q}_\rho \cdot \widehat{P}_\rho,$$

where the operation on the right side is matrix multiplication.

Therefore, assuming that the Fourier transforms $\widehat{P}_\rho^{(t)}$ and $\widehat{Q}_\rho^{(t)}$ are given, the prediction/rollup update rule is simply:

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

[6]Note that this definition of convolution on groups is *strictly* a generalization of convolution of functions on the real line, and is a non-commutative operation for non-abelian groups. Thus the distribution $P * Q$ is not necessarily the same as $Q * P$.

σ	$P^{(0)}$	$Q^{(1)}$	$P^{(1)}$	$Q^{(2)}$	$P^{(2)}$
ϵ	1	3/4	3/4	3/4	9/16
$(1,2)$	0	1/4	1/4	0	3/16
$(2,3)$	0	0	0	0	0
$(1,3)$	0	0	0	1/4	3/16
$(1,2,3)$	0	0	0	0	1/16
$(1,3,2)$	0	0	0	0	0

Table 4: Primal domain prediction/rollup example.

	$\widehat{P}^{(0)}$	$\widehat{Q}^{(1)}$	$\widehat{P}^{(1)}$	$\widehat{Q}^{(2)}$	$\widehat{P}^{(2)}$
$\rho_{(3)}$	1	1	1	1	1
$\rho_{(2,1)}$	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$	$\begin{bmatrix} \frac{1}{2} & 0 \\ 0 & 1 \end{bmatrix}$	$\begin{bmatrix} \frac{1}{2} & 0 \\ 0 & 1 \end{bmatrix}$	$\begin{bmatrix} \frac{1}{2} & -\frac{\sqrt{3}}{8} \\ -\frac{\sqrt{3}}{8} & \frac{5}{8} \end{bmatrix}$	$\begin{bmatrix} \frac{7}{16} & -\frac{\sqrt{3}}{8} \\ -\frac{\sqrt{3}}{16} & \frac{5}{8} \end{bmatrix}$
$\rho_{(1,1,1)}$	1	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$

Table 5: Fourier domain prediction/rollup example.

Note that the update only requires knowledge of $\hat{P}$ and does not require P. Furthermore, the update is *pointwise* in the Fourier domain in the sense that the coefficients at the representation ρ affect $\widehat{P}_\rho^{(t+1)}$ *only* at ρ. Consequently, prediction/rollup updates in the Fourier domain never increase the representational complexity. For example, if we maintain third-order marginals, then a single step of prediction/rollup called at time t returns the *exact* third-order marginals at time $t+1$, and nothing more.

Example 16. *We run the prediction/rollup routines on the first two time steps of the example in Figure 2, first in the primal domain, then in the Fourier domain. At each mixing event, two tracks, i and j, swap identities with some probability. Using a mixing model given by:*

$$Q(\pi) = \begin{cases} 3/4 & \text{if } \pi = \epsilon \\ 1/4 & \text{if } \pi = (i,j) \\ 0 & \text{otherwise} \end{cases},$$

we obtain results shown in Tables 4 and 5.

6.1.1 Limitations of random walk models

While the random walk assumption captures a rather general family of transition models, there do exist certain models which cannot be written as a random walk on a group. In particular, one limitation is that the prediction/rollup update for a random walk model can only increase the entropy of the distribution. As with

Kalman filters, localization is thus impossible without making observations. [7] (Shin et al., 2005) show that the entropy must increase for a certain kind of random walk on S_n (where π could be either the identity or the transposition (i,j)), but in fact, the result is easily generalized for any random walk mixing model and for any group.

Proposition 17.

$$H\left[P^{(t+1)}(\sigma^{(t+1)})\right] \geq \max\left\{H\left[Q^{(t)}(\tau^{(t)})\right], H\left[P^{(t)}(\sigma^{(t)})\right]\right\},$$

where $H\left[P(\sigma)\right]$ *denotes the statistical entropy functional,* $H[P(\sigma)] = -\sum_{\sigma\in G} P(\sigma)\log P(\sigma)$.

Proof. We have:

$$\begin{aligned}
P^{(t+1)}(\sigma^{(t+1)}) &= \left[Q^{(t)} * P^{(t)}\right](\sigma^{(t+1)}) \\
&= \sum_{\sigma^{(t)}} Q(\sigma^{(t+1)}\cdot(\sigma^{(t)})^{-1})P^{(t)}(\sigma^{(t)})
\end{aligned}$$

Applying the Jensen Inequality to the entropy function (which is concave) yields:

$$\begin{aligned}
H\left[P^{(t+1)}(\sigma^{(t+1)})\right] &\geq \sum_{\sigma^{(t)}} P^{(t)}(\sigma^{(t)})H\left[Q^{(t)}(\sigma\cdot(\sigma^{(t)})^{-1})\right], && \text{(Jensen's inequality)} \\
&= \sum_{\sigma^{(t)}} P^{(t)}(\sigma^{(t)})H\left[Q^{(t)}(\sigma)\right], && \text{(translation invariance of entropy)} \\
&= H\left[Q^{(t)}(\sigma)\right], && \text{(since } \textstyle\sum_{\sigma^{(t)}} P^{(t)}(\sigma^{(t)}) = 1\text{)}.
\end{aligned}$$

The proof that $H\left[P^{(t+1)}(\sigma^{(t+1)})\right] \geq H\left[P^{(t)}(\sigma^{(t)})\right]$ is similar with the exception that we must rewrite the convolution so that the sum ranges over $\tau^{(t)}$.

$$\begin{aligned}
P^{(t+1)}(\sigma^{(t+1)}) &= \left[Q^{(t)} * P^{(t)}\right](\sigma^{(t+1)}), \\
&= \sum_{\tau^{(t)}} Q^{(t)}(\tau^{(t)})P^{(t)}((\tau^{(t)})^{-1}\cdot\sigma^{(t+1)}).
\end{aligned}$$

□

Example 18. *This example is based on one from (Diaconis, 1988). Consider a deck of cards numbered* $\{1,\ldots,n\}$. *Choose a random permutation of cards by*

[7] In general, if we are not constrained to using linear Gaussian models, it *is* possible to localize with no observations. Consider a robot walking along the unit interval on the real line (which is not a group). If the position of the robot is unknown, one easy localization strategy might be to simply drive the robot to the right, with the knowledge that given ample time, the robot will slam into the 'wall', at which point it will have been localized. With random walk based models on groups however, these strategies are impossible — imagine the same robot walking around the unit circle — since, in some sense, the group structure prevents the existence of 'walls'.

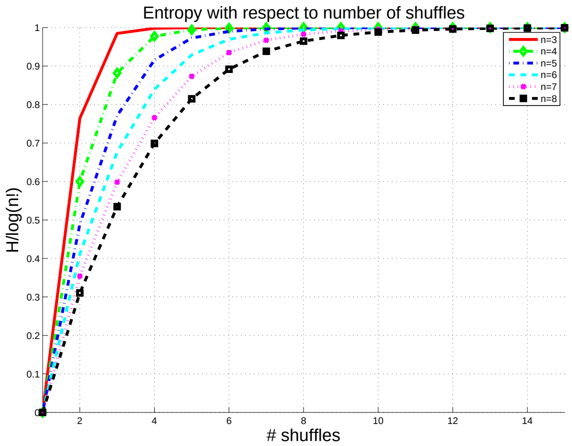


Figure 4: We start with a deck of cards in sorted order, and perform fifteen consecutive shuffles according to the rule given in Equation 6.1. The plot shows the entropy of the distribution over permutations with respect to the number of shuffles for $n = 3, 4, \ldots, 8$. When $H(P)/\log(n!) = 1$, the distribution has become uniform.

first picking two cards independently, and swapping (a card might be swapped with itself), yielding the following probability distribution over S_n:

$$Q(\pi) = \begin{cases} \frac{1}{n} & \textit{if } \pi = id \\ \frac{2}{n^2} & \textit{if } \pi \textit{ is a transposition} \\ 0 & \textit{otherwise} \end{cases} . \tag{6.1}$$

Repeating the above process for generating random permutations π gives a transition model for a hidden Markov model over the symmetric group. We can also see (Figure 4) that the entropy of the deck increases monotonically with each shuffle, and that repeated shuffles with $Q(\pi)$ eventually bring the deck to the uniform distribution.

6.2 Fourier conditioning

In contrast with the prediction/rollup operation, conditioning can potentially increase the representational complexity. As an example, suppose that we know the following first-order marginal probabilities:

$$P(\text{Alice is at Track 1 or Track 2}) = .9, \text{ and}$$

$$P(\text{Bob is at Track 1 or Track 2}) = .9.$$

If we then make the following first-order observation:

$$P(\text{Cathy is at Track 1 or Track 2}) = 1,$$

then it can be inferred that Alice and Bob cannot *both* occupy Tracks 1 and 2 at the same time, i.e.,

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

demonstrating that after conditioning, we are left with knowledge of second-order (unordered) marginals despite the fact that the prior and likelihood functions were only known up to first-order. Intuitively, the example shows that conditioning "smears" information from low-order Fourier coefficients to high-order coefficients, and that one cannot hope for a pointwise operation as was afforded by prediction/rollup. We now show precisely how irreducibles of different complexities "interact" with each other in the Fourier domain during conditioning.

An application of Bayes rule to find a posterior distribution $P(\sigma|z)$ after observing some evidence z requires two steps: a *pointwise product* of likelihood $P(z|\sigma)$ and prior $P(\sigma)$, followed by a normalization step:

$$P(\sigma|z) = \eta \cdot P(z|\sigma) \cdot P(\sigma).$$

For notational convenience, we will refer to the likelihood function as $L(z|\sigma)$ henceforth. We showed earlier that the normalization constant $\eta^{-1} = \sum_\sigma L(z|\sigma) \cdot P(\sigma)$ is given by the Fourier transform of $\widehat{L^{(t)}P^{(t)}}$ at the trivial representation — and therefore the normalization step of conditioning can be implemented by simply dividing each Fourier coefficient by the scalar $\left[\widehat{L^{(t)}P^{(t)}}\right]_{\rho_{(n)}}$.

The pointwise product of two functions f and g, however, is trickier to formulate in the Fourier domain. For functions on the real line, the pointwise product of functions can be implemented by convolving the Fourier coefficients of $\hat{f}$ and $\hat{g}$, and so a natural question is: can we apply a similar operation for functions over general groups? Our answer to this is that there is an analogous (but more complicated) notion of convolution in the Fourier domain of a general finite group. We present a convolution-based conditioning algorithm which we call *Kronecker Conditioning*, which, in contrast to the pointwise nature of the Fourier Domain prediction/rollup step, and much like convolution, smears the information at an irreducible ρ_ν to other irreducibles.

6.2.1 Fourier transform of the pointwise product

Our approach to computing the Fourier transform of the pointwise product in terms of $\hat{f}$ and $\hat{g}$ is to manipulate the function $f(\sigma)g(\sigma)$ so that it can be seen as the result of an inverse Fourier transform (Equation 4.4). Hence, the goal will be to find matrices R_ν (as a function of $\hat{f}, \hat{g}$) such that for any $\sigma \in G$,

$$f(\sigma) \cdot g(\sigma) = \frac{1}{|G|} \sum_\nu d_{\rho_\nu} \operatorname{Tr}\left(R_\nu^T \cdot \rho_\nu(\sigma)\right), \tag{6.2}$$

after which we will be able to read off the Fourier transform of the pointwise product as $\left[\widehat{fg}\right]_{\rho_\nu} = R_\nu$.

1. If A and B are square, $\mathrm{Tr}\,(A \otimes B) = (\mathrm{Tr}A) \cdot (\mathrm{Tr}B)$.
2. $(A \otimes B) \cdot (C \otimes D) = AC \otimes BD$.
3. Let A be an $n \times n$ matrix, and C an invertible $n \times n$ matrix. Then $\mathrm{Tr}A = \mathrm{Tr}\left(C^{-1}AC\right)$.
4. Let A be an $n \times n$ matrix and B_i be matrices of size $m_i \times m_i$ where $\sum_i m_i = n$. Then $\mathrm{Tr}\,(A \cdot (\bigoplus_i B_i)) = \sum_i \mathrm{Tr}\,(A_i \cdot B_i)$, where A_i is the block of A corresponding to block B_i in the matrix $(\bigoplus_i B_i)$.

Table 6: Matrix Identities used in Proposition 19.

For any $\sigma \in G$ we can write the pointwise product in terms $\hat{f}$ and $\hat{g}$ using the inverse Fourier transform:

$$\begin{aligned} f(\sigma) \cdot g(\sigma) &= \left[\frac{1}{|G|}\sum_{\lambda} d_{\rho_\lambda} \mathrm{Tr}\left(\hat{f}^T_{\rho_\lambda} \cdot \rho_\lambda(\sigma)\right)\right] \cdot \left[\frac{1}{|G|}\sum_{\mu} d_{\rho_\mu} \mathrm{Tr}\left(\hat{g}^T_{\rho_\mu} \cdot \rho_\mu(\sigma)\right)\right] \\ &= \left(\frac{1}{|G|}\right)^2 \sum_{\lambda,\mu} d_{\rho_\lambda} d_{\rho_\mu} \left[\mathrm{Tr}\left(\hat{f}^T_{\rho_\lambda} \cdot \rho_\lambda(\sigma)\right) \cdot \mathrm{Tr}\left(\hat{g}^T_{\rho_\mu} \cdot \rho_\mu(\sigma)\right)\right]. \quad (6.3) \end{aligned}$$

Now we want to manipulate this product of traces in the last line to be just one trace (as in Equation 6.2), by appealing to some properties of the *Kronecker Product*. The Kronecker product of an $n \times n$ matrix $U = (u_{i,j})$ by an $m \times m$ matrix V, is defined to be the $nm \times nm$ matrix

$$U \otimes V = \begin{pmatrix} u_{1,1}V & u_{1,2}V & \dots & u_{1,n}V \\ u_{2,1}V & u_{2,2}V & \dots & u_{2,n}V \\ \vdots & \vdots & \ddots & \vdots \\ u_{n,1}V & u_{n,2}V & \dots & u_{n,n}V \end{pmatrix}.$$

We summarize some important matrix properties in Table 6. The connection to our problem is given by matrix property 1. Applying this to Equation 6.3, we have:

$$\begin{aligned} \mathrm{Tr}\left(\hat{f}^T_{\rho_\lambda} \cdot \rho_\lambda(\sigma)\right) \cdot \mathrm{Tr}\left(\hat{g}^T_{\rho_\mu} \cdot \rho_\mu(\sigma)\right) &= \mathrm{Tr}\left(\left(\hat{f}^T_{\rho_\lambda} \cdot \rho_\lambda(\sigma)\right) \otimes \left(\hat{g}^T_{\rho_\mu} \cdot \rho_\mu(\sigma)\right)\right) \\ &= \mathrm{Tr}\left(\left(\hat{f}_{\rho_\lambda} \otimes \hat{g}_{\rho_\mu}\right)^T \cdot \left(\rho_\lambda(\sigma) \otimes \rho_\mu(\sigma)\right)\right), \end{aligned}$$

where the last line follows by Property 2. The term on the left, $\hat{f}_{\rho_\lambda} \otimes \hat{g}_{\rho_\mu}$, is a matrix of coefficients. The term on the right, $\rho_\lambda(\sigma) \otimes \rho_\mu(\sigma)$, itself happens to be a representation, called the *Kronecker (or Tensor) Product Representation*. In general, the Kronecker product representation is reducible, and so it can decomposed into a direct sum of irreducibles. In particular, if ρ_λ and ρ_μ are any two irreducibles of G, there exists a similarity transform $C_{\lambda\mu}$ such that, for any $\sigma \in G$,

$$C_{\lambda\mu}^{-1} \cdot [\rho_\lambda \otimes \rho_\mu](\sigma) \cdot C_{\lambda\mu} = \bigoplus_{\nu} \bigoplus_{\ell=1}^{z_{\lambda\mu\nu}} \rho_\nu(\sigma). \tag{6.4}$$

The $\oplus$ symbols here refer to a matrix direct sum as in Equation 2, ν indexes over all irreducible representations of S_n, while ℓ indexes over a number of *copies* of ρ_ν which appear in the decomposition. We index blocks on the right side of this equation by pairs of indices (ν, ℓ). The number of copies of each ρ_ν (for the tensor product pair $\rho_\lambda \otimes \rho_\mu$) is denoted by the integer $z_{\lambda\mu\nu}$, the collection of which, taken over all triples (λ, μ, ν), are commonly referred to as the *Clebsch-Gordan series*. Note that we allow the $z_{\lambda\mu\nu}$ to be zero, in which case ρ_ν does not contribute to the direct sum. The matrices $C_{\lambda\mu}$ are known as the *Clebsch-Gordan coefficients*. The *Kronecker Product Decomposition* problem is that of finding the irreducible components of the Kronecker product representation, and thus to find the Clebsch-Gordan series/coefficients for each pair of irreducible representations (ρ_λ, ρ_μ).

Decomposing the Kronecker product inside Equation 6.4 using the Clebsch-Gordan series and coefficients yields the desired Fourier transform, which we summarize in the form of a proposition. In the case that f and g are defined over an abelian group, then the following formulas reduce to the familiar form of convolution.

Proposition 19. *Let $\hat{f}, \hat{g}$ be the Fourier transforms of functions f and g respectively, and for each ordered pair of irreducibles (ρ_λ, ρ_μ), define: $A_{\lambda\mu} \triangleq C_{\lambda\mu}^{-1} \cdot \left(\hat{f}_{\rho_\lambda} \otimes \hat{g}_{\rho_\mu}\right) \cdot C_{\lambda\mu}$. Then the Fourier tranform of the pointwise product fg is:*

$$\left[\widehat{fg}\right]_{\rho_\nu} = \frac{1}{d_{\rho_\nu}|G|} \sum_{\lambda\mu} d_{\rho_\lambda} d_{\rho_\mu} \sum_{\ell=1}^{z_{\lambda\mu\nu}} A_{\lambda\mu}^{(\nu,\ell)}, \tag{6.5}$$

where $A_{\lambda\mu}^{(\nu,\ell)}$ is the block of $A_{\lambda\mu}$ corresponding to the (ν, ℓ) block in $\bigoplus_\nu \bigoplus_{\ell=1}^{z_{\lambda\mu\nu}} \rho_\nu$ from Equation 6.4.

Proof. We use the fact that $C_{\lambda\mu}$ is an orthogonal matrix for all pairs (ρ_λ, ρ_μ), i.e., $C_{\lambda\mu}^T \cdot C_{\lambda\mu} = I$.

$$\begin{aligned}
f(\sigma) \cdot g(\sigma) &= \left[\frac{1}{|G|} \sum_\lambda d_{\rho_\lambda} \mathrm{Tr}\left(\hat{f}_{\rho_\lambda}^T \cdot \rho_\lambda(\sigma)\right)\right] \cdot \left[\frac{1}{|G|} \sum_\mu d_{\rho_\mu} \mathrm{Tr}\left(\hat{g}_{\rho_\mu}^T \cdot \rho_\mu(\sigma)\right)\right] \\
&= \left(\frac{1}{|G|}\right)^2 \sum_{\lambda,\mu} d_{\rho_\lambda} d_{\rho_\mu} \left[\mathrm{Tr}\left(\hat{f}_{\rho_\lambda}^T \cdot \rho_\mu(\sigma)\right) \cdot \mathrm{Tr}\left(\hat{g}_{\rho_\mu}^T \cdot \rho_\mu(\sigma)\right)\right] \\
\text{(by Property 1)} &= \left(\frac{1}{|G|}\right)^2 \sum_{\lambda,\mu} d_{\rho_\lambda} d_{\rho_\mu} \left[\mathrm{Tr}\left(\left(\hat{f}_{\rho_\lambda}^T \cdot \rho_\lambda(\sigma)\right) \otimes \left(\hat{g}_{\rho_\mu}^T \cdot \rho_\mu(\sigma)\right)\right)\right] \\
\text{(by Property 2)} &= \left(\frac{1}{|G|}\right)^2 \sum_{\lambda,\mu} d_{\rho_\lambda} d_{\rho_\mu} \mathrm{Tr}\left(\left(\hat{f}_{\rho_\lambda} \otimes \hat{g}_{\rho_\mu}\right)^T \cdot (\rho_\lambda(\sigma) \otimes \rho_\mu(\sigma))\right)
\end{aligned}$$

$$
\begin{aligned}
\text{(by Property 3)} &= \left(\frac{1}{|G|}\right)^2 \sum_{\lambda,\mu} d_{\rho_\lambda} d_{\rho_\mu} \mathrm{Tr}\left(C_{\lambda\mu}^T \cdot \left(\hat{f}_{\rho_\lambda} \otimes \hat{g}_{\rho_\mu}\right)^T \cdot C_{\lambda\mu}\right. \\
&\qquad \left. \cdot C_{\lambda\mu}^T \cdot (\rho_\lambda(\sigma) \otimes \rho_\mu(\sigma)) \cdot C_{\lambda\mu}\right) \\
\text{(by definition of } C_{\lambda\mu} \text{ and } A_{\lambda\mu}) &= \left(\frac{1}{|G|}\right)^2 \sum_{\lambda,\mu} d_{\rho_\lambda} d_{\rho_\mu} \mathrm{Tr}\left(A_{\lambda\mu}^T \cdot \left(\bigoplus_\nu \bigoplus_{\ell=1}^{z_{\lambda\mu\nu}} \rho_\nu(\sigma)\right)\right) \\
\text{(by Property 4)} &= \frac{1}{|G|^2} \sum_{\lambda\mu} d_{\rho_\lambda} d_{\rho_\mu} \sum_\nu d_{\rho_\nu} \sum_{\ell=1}^{z_{\lambda\mu\nu}} \mathrm{Tr}\left(\left(d_{\rho_\nu}^{-1} A_{\lambda\mu}^{(\nu,\ell)}\right)^T \rho_\nu(\sigma)\right) \\
\text{(rearranging terms)} &= \frac{1}{|G|} \sum_\nu d_{\rho_\nu} \mathrm{Tr}\left[\left(\sum_{\lambda\mu} \sum_{\ell=1}^{z_{\lambda\mu\nu}} \frac{d_{\rho_\lambda} d_{\rho_\mu}}{d_{\rho_\nu}|G|} A_{\lambda\mu}^{(\nu,\ell)}\right)^T \rho_\nu(\sigma)\right].
\end{aligned}
$$

Recognizing the last expression as an inverse Fourier transform completes the proof. □

The Clebsch-Gordan series, $z_{\lambda\mu\nu}$, plays an important role in Equation 6.5, which says that the (ρ_λ, ρ_μ) crossterm contributes to the pointwise product at ρ_ν *only* when $z_{\lambda\mu\nu} > 0$. In the simplest case, we have that

$$
z_{(n),\mu,\nu} = \begin{cases} 1 & \text{if } \mu = \nu \\ 0 & \text{otherwise} \end{cases},
$$

which is true since $\rho_{(n)}(\sigma) = 1$ for all $\sigma \in S_n$. As another example, it is known that:

$$
\rho_{(n-1,1)} \otimes \rho_{(n-1,1)} \equiv \rho_{(n)} \oplus \rho_{(n-1,1)} \oplus \rho_{(n-2,2)} \oplus \rho_{(n-2,1,1)}, \tag{6.6}
$$

or equivalently,

$$
z_{(n-1,1),(n-1,1),\nu} = \begin{cases} 1 & \text{if } \nu \text{ is one of } (n),(n-1,1),(n-2,2), \text{ or } (n-2,1,1) \\ 0 & \text{otherwise} \end{cases}.
$$

So if the Fourier transforms of the likelihood and prior are zero past the first two irreducibles ((n) and $(n-1,1)$), then a single conditioning step results in a Fourier transform which, in general, carries second-order information at $(n-2,2)$ and $(n-2,1,1)$, but is guaranteed to be zero past the first four irreducibles (n), $(n-1,1)$, $(n-2,2)$ and $(n-2,1,1)$.

As far as we know, there are no analytical formulas for finding the entire Clebsch-Gordan series or coefficients, and in practice, these computations do in fact take a long time. We emphasize however, that as fundamental constants related to the irreducibles of the Symmetric group, they need only be computed *once* (like the digits of π, for example) and can be stored in a table for all future reference. For a detailed discussion of techniques for computing the Clebsch-Gordan series/coefficients, see Appendix B. We plan to make a set of precomputed coefficients available on the web, but for now we will assume throughout the rest of the paper that both the series and coefficients have been made available as a lookup table. We conclude our section on inference with a fully worked example of Kronecker conditioning.

Example 20. *For this example, refer to Table 2 for the representations of S_3. Given functions $f, g : S_3 \to \mathbb{R}$, we will compute the Fourier transform of the pointwise product $f \cdot g$.*

Since there are three irreducibles, there are nine tensor products $\rho_\lambda \otimes \rho_\mu$ to decompose, six of which are trivial either because they are one-dimensional, or involve tensoring against the trivial representation. The nontrivial tensor products to consider are $\rho_{(2,1)} \otimes \rho_{(1,1,1)}$, $\rho_{(1,1,1)} \otimes \rho_{(2,1)}$ and $\rho_{(2,1)} \otimes \rho_{(2,1)}$. The Clebsch-Gordan series for the nontrivial tensor products are:

	$z_{(2,1),(1,1,1),\nu}$	$z_{(1,1,1),(2,1),\nu}$	$z_{(2,1),(2,1),\nu}$
$\nu = (3)$	*0*	*0*	*1*
$\nu = (2,1)$	*1*	*1*	*1*
$\nu = (1,1,1)$	*0*	*0*	*1*

The Clebsch-Gordan coefficients for the nontrivial tensor products are given by the following orthogonal matrices:

$$C_{(2,1)\otimes(1,1,1)} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}, \quad C_{(1,1,1)\otimes(2,1)} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix},$$

$$C_{(2,1)\otimes(2,1)} = \frac{\sqrt{2}}{2}\begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & -1 & 0 & 1 \\ 0 & -1 & 0 & -1 \\ 1 & 0 & 1 & 0 \end{bmatrix}.$$

As in Proposition 19, define:

$$A_{(2,1)\otimes(1,1,1)} = C^T_{(2,1)\otimes(1,1,1)}\left(\hat{f}_{(2,1)} \otimes \hat{g}_{(1,1,1)}\right) C_{(2,1)\otimes(1,1,1)}, \tag{6.7}$$

$$A_{(1,1,1)\otimes(2,1)} = C^T_{(1,1,1)\otimes(2,1)}\left(\hat{f}_{(1,1,1)} \otimes \hat{g}_{(2,1)}\right) C_{(1,1,1)\otimes(2,1)}, \tag{6.8}$$

$$A_{(2,1)\otimes(2,1)} = C^T_{(2,1)\otimes(2,1)}\left(\hat{f}_{(2,1)} \otimes \hat{g}_{(2,1)}\right) C_{(2,1)\otimes(2,1)}, \tag{6.9}$$

Then Proposition 19 gives the following formulas:

$$\widehat{f \cdot g}_{\rho_{(3)}} = \frac{1}{3!} \cdot \left[\hat{f}_{\rho_{(3)}} \cdot \hat{g}_{\rho_{(3)}} + \hat{f}_{\rho_{(1,1,1)}} \cdot \hat{g}_{\rho_{(1,1,1)}} + 4 \cdot \left[A_{(2,1)\otimes(2,1)}\right]_{1,1}\right], \tag{6.10}$$

$$\begin{aligned}\widehat{f \cdot g}_{\rho_{(2,1)}} = \frac{1}{3!} \cdot \Big[&\hat{f}_{\rho_{(2,1)}} \cdot \hat{g}_{\rho_{(3)}} + \hat{f}_{\rho_{(3)}} \cdot \hat{g}_{\rho_{(2,1)}} + A_{(1,1,1)\otimes(2,1)} \\ &+ A_{(2,1)\otimes(1,1,1)} + 2 \cdot \left[A_{(2,1)\otimes(2,1)}\right]_{2:3,2:3}\Big],\end{aligned} \tag{6.11}$$

$$\widehat{f \cdot g}_{\rho_{(1,1,1)}} = \frac{1}{3!} \cdot \left[\hat{f}_{\rho_{(3)}} \cdot \hat{g}_{\rho_{(1,1,1)}} + \hat{f}_{\rho_{(1,1,1)}} \cdot \hat{g}_{\rho_{(3)}} + 4 \cdot \left[A_{(2,1)\otimes(2,1)}\right]_{4,4}\right], \tag{6.12}$$

where the notation $[A]_{a:b,c:d}$ denotes the block of entries in A between rows a and b, and between columns c and d (inclusive).

Using the above formulas, we can continue on Example 16 and compute the last update step in our identity management problem (Figure 2). At the final time step, we observe that Bob is at track 1 with 100% certainty. Our likelihood function is therefore nonzero only for the permutations which map Bob (the second identity) to the first track:

$$L(\sigma) \propto \begin{cases} 1 & \text{if } \sigma = (1,2) \text{ or } (1,3,2) \\ 0 & \text{otherwise} \end{cases}.$$

Algorithm 1: Pseudocode for the Fourier Prediction/Rollup Algorithm.

PREDICTIONROLLUP
input : $\hat{Q}^{(t)}_{\rho_\lambda}$ and $\hat{P}^{(t)}_{\rho_\lambda}$, $\rho_\lambda \in \Lambda$
output: $\hat{P}^{(t+1)}_{\rho_\lambda}$, $\rho_\lambda \in \Lambda$
foreach $\rho_\lambda \in \Lambda$ **do** $\hat{P}^{(t+1)}_{\rho_\lambda} \leftarrow \hat{Q}^{(t)}_{\rho_\lambda} \cdot \hat{P}^{(t)}_{\rho_\lambda}$;

The Fourier transform of the likelihood function is:

$$\widehat{L}_{\rho_{(3)}} = 2, \quad \widehat{L}_{\rho_{(2,1)}} = \begin{bmatrix} -3/2 & \sqrt{3}/2 \\ -\sqrt{3}/2 & 1/2 \end{bmatrix}, \quad \widehat{L}_{\rho_{(1,1,1)}} = 0. \tag{6.13}$$

Plugging the Fourier transforms of the prior distribution ($\widehat{P}^{(2)}$ from Table 5) and likelihood (Equation 6.13) into Equations 6.7, 6.8, 6.9, we have:

$$A_{(2,1)\otimes(1,1,1)} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad A_{(1,1,1)\otimes(2,1)} = \frac{1}{8}\begin{bmatrix} 1 & \sqrt{3} \\ -\sqrt{3} & -3 \end{bmatrix},$$

$$A_{(2,1)\otimes(2,1)} = \frac{1}{32}\begin{bmatrix} -7 & -\sqrt{3} & 11 & 5\sqrt{3} \\ -2\sqrt{3} & -10 & -6\sqrt{3} & -14 \\ 20 & 22\sqrt{3} & -4 & 4\sqrt{3} \\ -11\sqrt{3} & -23 & -\sqrt{3} & -13 \end{bmatrix}$$

To invoke Bayes rule in the Fourier domain, we perform a pointwise product using Equations 6.10, 6.11, 6.12, and normalize by dividing by the trivial coefficient, which yields the Fourier transform of the posterior distribution as:

$$\left[\widehat{P(\sigma|z)}\right]_{\rho_{(3)}} = 1, \quad \left[\widehat{P(\sigma|z)}\right]_{\rho_{(2,1)}} = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \left[\widehat{P(\sigma|z)}\right]_{\rho_{(1,1,1)}} = -1. \tag{6.14}$$

Finally, we can see that the result is correct by recognizing that the Fourier transform of the posterior (Equation 6.14) corresponds exactly to the distribution which is 1 at $\sigma = (1,2)$ and 0 everywhere else. Bob is therefore at Track 1, Alice at Track 2 and Cathy at Track 3.

σ	ϵ	$(1,2)$	$(2,3)$	$(1,3)$	$(1,2,3)$	$(1,3,2)$
$P(\sigma)$	0	1	0	0	0	0

7 Approximate inference by bandlimiting

We now consider the consequences of performing inference using the Fourier transform at a reduced set of coefficients. Important issues include understanding how error can be introduced into the system, and when our algorithms are expected to perform well as an approximation. Specifically, we fix a bandlimit λ^{MIN} and maintain the Fourier transform of P only at irreducibles which are at λ^{MIN} or above in the dominance ordering:

$$\Lambda = \{\rho_\lambda \;:\; \lambda \trianglerighteq \lambda^{MIN}\}.$$

Algorithm 2: Pseudocode for the Kronecker Conditioning Algorithm.

KroneckerConditioning

input : Fourier coefficients of the likelihood function, $\hat{L}_{\rho_\lambda},\ \rho_\lambda \in \Lambda_L$, and Fourier coefficients of the prior distribution, $\hat{P}_{\rho_\mu},\ \rho_\mu \in \Lambda_P$

output: Fourier coefficients of the posterior distribution, $\widehat{LP}_{\rho_\nu},\ \rho_\nu \in \Lambda_P$

foreach $\rho_\nu \in \Lambda_P$ **do** $\widehat{LP}_{\rho_\nu} \leftarrow \mathbf{0}$ *//Initialize Posterior*

//Pointwise Product

foreach $\rho_\lambda \in \Lambda_L$ **do**

  foreach $\rho_\mu \in \Lambda_P$ **do**

    $z \leftarrow CGseries(\rho_\lambda, \rho_\mu)$;

    $C_{\lambda\mu} \leftarrow CGcoefficients(\rho_\lambda, \rho_\mu)$; $A_{\lambda\mu} \leftarrow C_{\lambda\mu}^T \cdot \left(\hat{L}_{\rho_\lambda} \otimes \hat{P}_{\rho_\mu}\right) \cdot C_{\lambda\mu}$;

    for $\rho_\nu \in \Lambda_P$ *such that* $z_{\lambda\mu\nu} \neq 0$ **do**

      for $\ell = 1$ ***to*** $z_{\lambda\mu\nu}$ **do**

        $\left[\widehat{L^{(t)}P^{(t)}}\right]_{\rho_\nu} \leftarrow \left[\widehat{L^{(t)}P^{(t)}}\right]_{\rho_\nu} + \frac{d_{\rho_\lambda} d_{\rho_\mu}}{d_{\rho_\nu} n!} A_{\lambda\mu}^{(\nu,\ell)}$; *//$A_{\lambda\mu}^{(\nu,\ell)}$ is the (ν,ℓ) block of $A_{\lambda\mu}$*

$\eta \leftarrow \left[\widehat{L^{(t)}P^{(t)}}\right]_{\rho_{(n)}}^{-1}$;

foreach $\rho_\nu \in \Lambda$ **do** $\left[\widehat{L^{(t)}P^{(t)}}\right]_{\rho_\nu} \leftarrow \eta \left[\widehat{L^{(t)}P^{(t)}}\right]_{\rho_\nu}$ *//Normalization*

For example, when $\lambda^{MIN} = (n-2,1,1)$, Λ is the set $\{\rho_{(n)}, \rho_{(n-1,1)}, \rho_{(n-2,2)}$, and $\rho_{(n-2,1,1)}\}$, which corresponds to maintaining second-order (ordered) marginal probabilities of the form $P(\sigma((i,j)) = (k,\ell))$. During inference, we follow the procedure outlined in the previous section but discard the higher order terms which can be introduced during the conditioning step. Pseudocode for bandlimited prediction/rollup and Kronecker conditioning is given in Algorithms 1 and 2. We note that it is not necessary to maintain the same number of irreducibles for both prior and likelihood during the conditioning step. The first question to ask is: when should one expect a bandlimited approximation to be close to $P(\sigma)$ as a function? Qualitatively, if a distribution is relatively smooth, then most of its energy is stored in the low-order Fourier coefficients. However, in a phenomenon quite reminiscent of the Heisenberg uncertainty principle from quantum mechanics, it is exactly when the distribution is sharply concentrated at a small subset of permutations, that the Fourier projection is unable to faithfully approximate the distribution. We illustrate this uncertainty effect in Figure 5 by plotting the accuracy of a bandlimited distribution against the entropy of a distribution.

Even though the bandlimited distribution is sometimes a poor approximation to the true distribution, the marginals mainatined by our algorithm are often sufficiently accurate. And so instead of considering the approximation accuracy of the bandlimited Fourier transform to the true joint distribution, we consider the accuracy only at the marginals which are maintained by our method.

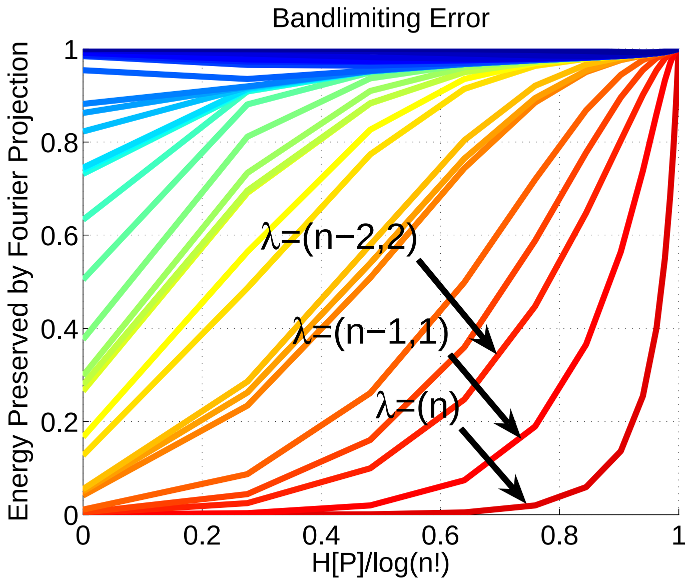
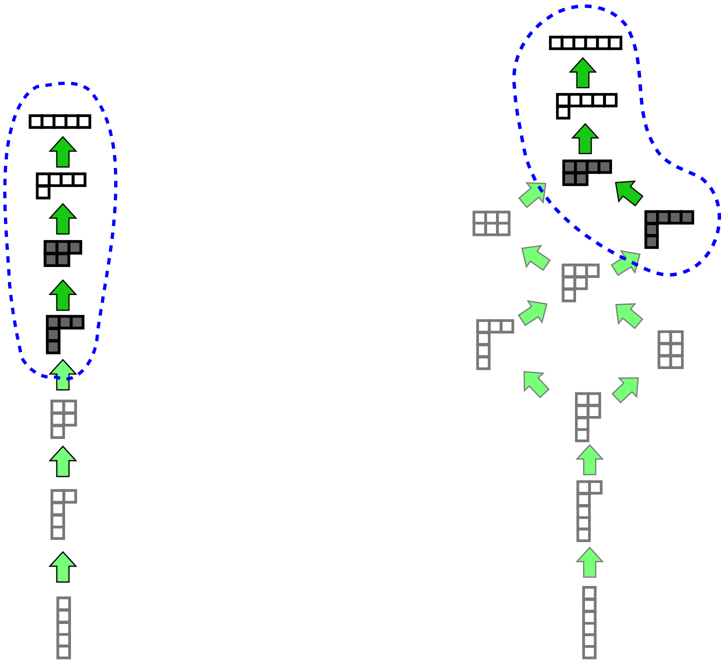


Figure 5: In general, smoother distributions are well approximated by low-order Fourier projections. In this graph, we show the approximation quality of the Fourier projections on distributions with different entropies, starting from sharply peaked delta distributions on the left side of the graph, which get iteratively smoothed until they becomes the maximum entropy uniform distribution on the right side. On the y-axis, we measure how much *energy* is preserved in the bandlimited approximation, which we define to be $\frac{|P'|^2}{|P|^2}$, where P' is the bandlimited approximation to P. Each line represents the approximation quality using a fixed number of Fourier coefficients. At one extreme, we achieve perfect signal reconstruction by using all Fourier coefficients, and at the other, we perform poorly on "spiky" distributions, but well on high-entropy distributions, by storing a single Fourier coefficient.

7.1 Sources of error during inference

We now analyze the errors incurred during our inference procedures with respect to the accuracy at maintained marginals. It is immediate that the Fourier domain prediction/rollup operation is *exact* due to its pointwise nature in the Fourier domain. For example, if we have the second order marginals at time $t = 0$, then we can find the exact second order marginals at all $t > 0$ if we only perform prediction/rollup operations. Instead, the errors in inference are only committed by Kronecker conditioning, where they are implicitly introduced at coefficients outside of Λ (by effectively setting the coefficients of the prior and likelihood at irreducibles outside of Λ to be zero), then propagated inside to the irreducibles of Λ.

In practice, we observe that the errors introduced at the low-order irre-



(a) $n = 5$ (b) $n = 6$

Figure 6: We show the dominance ordering for partitions of $n = 5$ and $n = 6$ again. By setting $\lambda^{MIN} = (3, 1, 1)$ and $(4, 1, 1)$ respectively, we keep the irreducibles corresponding to the partitions in the dotted regions. If we call Kronecker Conditioning with a first-order observation model, then according to Theorem 21, we can expect to incur some error at the Fourier coefficients corresponding to $(3, 1, 1)$ and $(3, 2)$ for $n = 5$, and $(4, 1, 1)$ and $(4, 2)$ for $n = 6$ (shown as shaded tableaux), but to be exact at first-order coefficients.

ducibles during inference are small if the prior and likelihood are sufficiently diffuse, which makes sense since the high-frequency Fourier coefficients are small in such cases. We can sometimes show that the update is *exact* at low order irreducibles if we maintain *enough* coefficients.

Theorem 21. *If $\lambda^{MIN} = (n-p, \lambda_2, \dots)$, and the Kronecker conditioning algorithm is called with a likelihood function whose Fourier coefficients are nonzero only at ρ_μ when $\mu \trianglerighteq (n-q, \mu_2, \dots)$, then the approximate Fourier coefficients of the posterior distribution are exact at the set of irreducibles:*

$$\Lambda_{EXACT} = \{\rho_\lambda \,:\, \lambda \trianglerighteq (n - |p-q|, \dots)\}.$$

Proof. See Appendix B. □

For example, if we call Kronecker conditioning by passing in third-order terms of the prior and first-order terms of the likelihood, then all first and second-order (unordered and ordered) marginal probabilities of the posterior distribution can be reconstructed without error.

7.2 Projecting to the marginal polytope

Despite the encouraging result of Theorem 21, the fact remains that consecutive conditioning steps can propagate errors to all levels of the bandlimited Fourier transform, and in many circumstances, results in a Fourier transform whose "marginal probabilities" correspond to no consistent joint distribution over permutations, and are sometimes negative. To combat this problem, we present a method for projecting to the space of coefficients corresponding to consistent joint distributions (which we will refer to as the *marginal polytope*) during inference.

We begin by discussing the first-order version of the marginal polytope projection problem. Given an $n \times n$ matrix, M, of real numbers, how can we decide whether there exists some probability distribution which has M as its matrix of first-order marginal probabilities? A necessary and sufficient condition, as it turns out, is for M to be *doubly stochastic*. That is, all entries of M must be nonnegative and all rows and columns of M must sum to one (the probability that Alice is at *some track* is 1, and the probability that *some identity* is at Track 3 is 1). The double stochasticity condition comes from the *Birkhoff-von Neumann* theorem (van Lint & Wilson, 2001) which states that a matrix is doubly stochastic *if and only if* it can be written as a convex combination of permutation matrices.

To "renormalize" first-order marginals to be doubly stochastic, some authors (Shin et al., 2003; Shin et al., 2005; Balakrishnan et al., 2004; Helmbold & Warmuth, 2007) have used the *Sinkhorn iteration*, which alternates between normalizing rows and columns independently until convergence is obtained. Convergence is guaranteed under mild conditions and it can be shown that the limit is a nonnegative doubly stochastic matrix which is closest to the original matrix in the sense that the Kullback-Leibler divergence is minimized (Balakrishnan et al., 2004).

There are several problems which cause the Sinkhorn iteration to be an unnatural solution in our setting. First, since the Sinkhorn iteration only works for nonnegative matrices, we would have to first cap entries to lie in the appropriate range, $[0, 1]$. More seriously, even though the Sinkhorn iteration would guarantee a doubly stochastic higher order matrix of marginals, there are several natural constraints which are violated when running the Sinkhorn iteration on higher-order marginals. For example, with second-order (ordered) marginals, it seems that we should at least enforce the following symmetry constraint:

$$P(\sigma : \sigma(k, \ell) = (i, j)) = P(\sigma : \sigma(\ell, k) = (j, i)),$$

which says, for example, that the marginal probability that Alice is in Track 1 and Bob is in Track 2 is the same as the marginal probability that Bob is in Track 2 and Alice is in Track 1. Another natural constraint that can be broken is what we refer to as *low-order marginal consistency.* For example, it should always be the case that:

$$P(j) = \sum_i P(i,j) = \sum_k P(j,k).$$

It should be noted that the doubly stochastic requirement is a special case of lower-order marginal consistency — we require that higher-order marginals be consistent on the 0^{th} order marginal.

While compactly describing the constraints of the marginal polytope exactly remains an open problem, we propose a method for projecting onto a *relaxed* form of the marginal polytope which addresses both symmetry and low-order consistency problems by operating directly on irreducible Fourier coefficients instead of on the matrix of marginal probabilities. After each conditioning step, we apply a 'correction' to the approximate posterior $P^{(t)}$ by finding the bandlimited function in the relaxed marginal polytope which is closest to $P^{(t)}$ in an L_2 sense. To perform the projection, we employ the Plancherel Theorem (Diaconis, 1988) which relates the L_2 distance between functions on S_n to a distance metric in the Fourier domain.

Proposition 22 (Plancherel Theorem)**.**

$$\sum_\sigma (f(\sigma) - g(\sigma))^2 \quad = \quad \frac{1}{|G|} \sum_\nu d_{\rho_\nu}\, Tr\left(\left(\hat{f}_{\rho_\nu} - \hat{g}_{\rho_\nu}\right)^T \cdot \left(\hat{f}_{\rho_\nu} - \hat{g}_{\rho_\nu}\right)\right). \tag{7.1}$$

To find the closest bandlimited function in the relaxed marginal polytope, we formulate a quadratic program whose objective is to minimize the right side of Equation 7.1, and whose sum is taken only over the set of maintained irreducibles, Λ, subject to the set of constraints which require all marginal probabilities to be nonnegative. We thus refer to our correction step as *Plancherel Projection.* Our quadratic program can be written as:

$$\begin{aligned}
&\text{minimize}_{\hat{f}^{proj}} && \sum_{\lambda \in \Lambda} d_\lambda \mathrm{Tr}\left[\left(\hat{f} - \hat{f}^{proj}\right)^T_{\rho_\lambda} \left(\hat{f} - \hat{f}^{proj}\right)_{\rho_\lambda}\right] \\
&\text{subject to:} && \left[\hat{f}^{proj}\right]_{(n)} = 1, \\
& && \left[C_{\lambda^{MIN}} \cdot \left(\bigoplus_{\mu \trianglerighteq \lambda^{MIN}} \bigoplus_{\ell=1}^{K_{\lambda^{MIN},\mu}} \hat{f}^{proj}_{\rho_\mu}\right) \cdot C^T_{\lambda^{MIN}}\right]_{ij} \geq 0, \qquad \text{for all } (i,j),
\end{aligned}$$

where $K_{\lambda^{MIN}}$ and $C_{\lambda^{MIN}}$ are the precomputed constants from Equation 5.4. We remark that even though the projection will produce a Fourier transform corresponding to nonnegative marginals which are consistent with each other, there might not necessarily exist a joint probability distribution on S_n consistent with those marginals except in the special case of first-order marginals.

Example 23. *In Example 20, we ran the Kronecker conditioning algorithm using all of the Fourier coefficients. If only the first-order coefficients are available, however, then the expressions for zeroth and first order terms of the posterior (Equations 6.10,6.11) become:*

$$\widehat{f \cdot g}_{\rho_{(3)}} = \frac{1}{3!} \cdot \left[\hat{f}_{\rho_{(3)}} \cdot \hat{g}_{\rho_{(3)}} + 4 \cdot \left[A_{(2,1)\otimes(2,1)} \right]_{1,1} \right], \tag{7.2}$$

$$\widehat{f \cdot g}_{\rho_{(2,1)}} = \frac{1}{3!} \cdot \left[\hat{f}_{\rho_{(2,1)}} \cdot \hat{g}_{\rho_{(3)}} + \hat{f}_{\rho_{(3)}} \cdot \hat{g}_{\rho_{(2,1)}} + 2 \cdot \left[A_{(2,1)\otimes(2,1)} \right]_{2:3,2:3} \right], \tag{7.3}$$

Plugging in the same numerical values from Example 20 and normalizing appropriately yields the approximate Fourier coefficients of the posterior:

$$\left[\widehat{P(\sigma|z)}\right]_{\rho_{(3)}} = 1 \qquad \left[\widehat{P(\sigma|z)}\right]_{\rho_{(2,1)}} = \begin{bmatrix} -10/9 & -77/400 \\ 77/400 & 4/3 \end{bmatrix},$$

which correspond to the following first-order marginal probabilities:

$$\hat{P}_{\tau_{(2,1)}} \left[\begin{array}{c|ccc} & A & B & C \\ \hline \textit{Track 1} & 0 & 11/9 & -2/9 \\ \textit{Track 2} & 1 & 0 & 0 \\ \textit{Track 3} & 0 & -2/9 & 11/9 \end{array} \right].$$

In particular, we see that the approximate matrix of 'marginals' contains negative numbers. Applying the Plancherel projection step, we obtain the following marginals:

$$\hat{P}_{\tau_{(2,1)}} \left[\begin{array}{c|ccc} & A & B & C \\ \hline \textit{Track 1} & 0 & 1 & 0 \\ \textit{Track 2} & 1 & 0 & 0 \\ \textit{Track 3} & 0 & 0 & 1 \end{array} \right],$$

which happen to be exactly the true posterior marginals. It should be noted however, that rounding the 'marginals' to be in the appropriate range would have worked in this particular example as well.

8 Probabilistic models of mixing and observations

While the algorithms presented in the previous sections are general in the sense that they work on all mixing and observation models, it is not always obvious how to compute the Fourier transform of a given model. In this section, we present ways to obtain such transforms for a few useful models.

8.1 Mixing models

The simplest mixing model for identity management assumes that with probability p, nothing happens, and that with probability $(1-p)$, the identities for tracks i and j are swapped. The probability distribution is therefore:

$$Q_{ij}(\pi) = \begin{cases} p & \text{if } \pi = \epsilon \\ 1-p & \text{if } \pi = (i,j) \\ 0 & \text{otherwise.} \end{cases}$$

Since Q_{ij} is such a sparse distribution (in the sense that $Q(\pi) = 0$ for most π), it is possible to directly compute $\widehat{Q}$ using Definition 6:

$$\widehat{Q}_{\rho_\lambda} = pI + (1-p)\rho_\lambda((i,j)),$$

where I refers to the $d_\lambda \times d_\lambda$ identity matrix, and $\rho_\lambda((i,j))$ is the irreducible representation matrix ρ_λ evaluated at the transposition (i,j) (which can be computed using the algorithms from Appendix A).

8.2 Observation models

The simplest model assumes that we can get observations of the form: 'track ℓ is color k' (which is essentially the model considered by (Kondor et al., 2007)). The probability of seeing color k at track ℓ given data association σ is

$$L(\sigma) = P(z_\ell = k|\sigma) = \alpha_{\sigma(\ell),k}, \tag{8.1}$$

where $\sum_k \alpha_{\sigma(\ell)k} = 1$. For each identity, the likelihood $L(\sigma) = P(z_\ell = k|\sigma)$ depends on a histogram over all possible colors. If the number of possible colors is K, then the likelihood model can be specified by an $n \times K$ matrix of probabilities. For example,

$$\alpha_{\sigma(\ell),k} = \left[\begin{array}{c|cccc} & k=\text{Red} & k=\text{Orange} & k=\text{Yellow} & k=\text{Green} \\ \hline \sigma(\ell)=\text{Alice} & 1/2 & 1/4 & 1/4 & 0 \\ \sigma(\ell)=\text{Bob} & 1/4 & 0 & 0 & 3/4 \\ \sigma(\ell)=\text{Cathy} & 0 & 1/2 & 1/2 & 0 \end{array} \right]. \tag{8.2}$$

Since the observation model only depends on a single identity, the first-order terms of the Fourier transform suffice to fully describe the likelihood. To compute the first-order Fourier coefficients, at irreducibles, we proceed by computing the first-order Fourier coefficients at the first-order permutation representation, then transforming to irreducible coefficients. The Fourier transform of the likelihood at the first-order permutation representation is given by:

$$\left[\widehat{L}_{\tau_{(n-1,1)}}\right]_{ij} = \sum_{\{\sigma:\sigma(j)=i\}} P\left(z_\ell = k|\sigma\right) = \sum_{\{\sigma:\sigma(j)=i\}} \alpha_{\sigma(\ell)k}.$$

To compute the ij-term, there are two cases to consider.

1. If $j = \ell$ (that is, if Track j is the same as the track that was observed), then the coefficient $\widehat{L}_{ij}$ is proportional to the probability that Identity i is color k.

$$\widehat{L}_{ij} = \sum_{\{\sigma:\sigma(\ell)=i\}} \alpha_{i,k} = (n-1)! \cdot \alpha_{i,k}. \tag{8.3}$$

2. If, on the other hand, $j \neq \ell$ (Track j is not the observed track)), then the coefficient $\hat{L}_{ij}$ is proportional to the sum over

$$\widehat{L}_{ij} = \sum_{\{\sigma:\sigma(j)=i\}} \alpha_{\sigma(\ell),k} \tag{8.4}$$

$$= \sum_{m\neq i} \sum_{\{\sigma:\sigma(j)=i \text{ and } \sigma(l)=m\}} \alpha_{\sigma(\ell),k} \tag{8.5}$$

$$= \sum_{m\neq i} (n-2)! \cdot \alpha_{m,k}. \tag{8.6}$$

Example 24. *We will compute the first-order marginals of the likelihood function on S_3 which arises from observing a "Red blob at Track 1". Plugging the values from the "Red" column of the α matrix (Equation 8.2) into Equation 8.3 and 8.6 yields the following matrix of first-order coefficients (at the $\tau_{(n-1,1)}$ permutation representation):*

$$\left[\hat{L}_{(n-1,1)}\right]_{ij} = \left[\begin{array}{c|ccc} & \textit{Track 1} & \textit{Track 2} & \textit{Track 3} \\ \hline \textit{Alice} & 1/4 & 1/4 & 1 \\ \textit{Bob} & 1/2 & 1/2 & 1/2 \\ \textit{Cathy} & 3/4 & 3/4 & 0 \end{array}\right]$$

The corresponding coefficients at the irreducible representations are:

$$\hat{L}_{(3)} = 1.5, \qquad \hat{L}_{(2,1)} = \begin{bmatrix} 0 & -\sqrt{3}/4 \\ 0 & -3/4 \end{bmatrix}, \qquad \hat{L}_{(1,1,1)} = 0.$$

9 Related work

Rankings and permutations have recently become an active area of research in machine learning due to their importance in information retrieval and preference elicitation. Rather than considering full distributions over permutations, many approaches, like RankSVM (Joachims, 2002) and RankBoost (Freund et al., 2003), have instead focused on learning a single 'optimal' ranking with respect to some objective function.

There are also several authors who have studied distributions over permutations/rankings (Mallows, 1957; Critchlow, 1985; Fligner & Verducci, 1986; Taylor et al., 2008; Lebanon & Mao, 2008). (Taylor et al., 2008) consider distributions over S_n which are induced by the rankings of n independent draws from n individually centered Gaussian distributions with equal variance. They compactly summarize their distributions using an $O(n^2)$ matrix which is conceptually similar to our first-order summaries and apply their techniques to ranking web documents. Most other previous approaches at directly modeling distributions on S_n, however, have relied on distance based models. For example, the Mallows model (Mallows, 1957) defines a Gaussian-like distribution over permutations as:

$$P(\sigma; c, \sigma_0) \propto \exp\left(-cd(\sigma, \sigma_0)\right), \tag{9.1}$$

where the function $d(\sigma, \sigma_0)$ is the *Kendall's tau distance* which counts the number of adjacent swaps that are required to bring σ^{-1} to σ_0^{-1}. Like Gaussians, distance based models tend to lack flexibility, and so (Lebanon & Mao, 2008) propose a nonparametric model of ranked (and partially ranked) data based on placing weighted Mallows kernels on top of training examples, which, as they show, can realize a far richer class of distributions, and can be learned efficiently. However, they do not address the inference problem, and it is not clear if one can efficiently perform inference operations like marginalization and conditioning in such models.

As we have shown in this paper, Fourier based methods (Diaconis, 1988; Kondor et al., 2007; Huang et al., 2007) offer a principled alternative method for compactly representing distributions over permutations and performing efficient probabilistic inference operations. Our work draws from two strands of research — one from the data association/identity management literature, and one from a more theoretical area on Fourier analysis in statistics. In the following, we review several of the works which have led up to our current Fourier based approach.

9.1 Previous work in identity management

The identity management problem has been addressed in a number of previous works, and is closely related to, but not identical with, the classical data association problem of maintaining correspondences between tracks and observations. Both problems need to address the fundamental combinatorial challenge that there is a factorial or exponential number of associations to maintain between tracks and identities, or between tracks and observations respectively. A vast literature already exists on the the data association problem, beginning with the *multiple hypothesis testing* approach (MHT) of (Reid, 1979). The MHT is a 'deferred logic' method in which past observations are exploited in forming new hypotheses when a new set of observations arises. Since the number of hypotheses can grow exponentially over time, various heuristics have been proposed to help cope with the complexity blowup. For example, one can choose to maintain only the k *best* hypotheses for some parameter k (Cox & Hingorani, 1994), using Murty's algorithm (Murty, 1968). But for such an approximation to be effective, k may still need to scale exponentially in the number of objects. A slightly more recent filtering approach is the *joint probabilistic data association filter* (JPDA) (Bar-Shalom & Fortmann, 1988), which is a suboptimal single-stage approximation of the optimal Bayesian filter. JPDA makes associations sequentially and is unable to correct erroneous associations made in the past (Poore, 1995). Even though the JPDA is more efficient than the MHT, the calculation of the JPDA association probabilities is still a #P-complete problem (Collins & Uhlmann, 1992), since it effectively must compute matrix permanents. Polynomial approximation algorithms to the JPDA association probabilities have recently been studied using Markov chain Monte Carlo (MCMC) methods (Oh et al., 2004; Oh & Sastry, 2005).

The identity management problem was first explicitly introduced in (Shin et al., 2003). Identity management differs from the classical data association problem in that its observation model is not concerned with the low-level tracking details but instead with high level information about object identities. (Shin et al., 2003) introduced the notion of the *belief matrix* approximation of the association probabilities, which collapses a distribution over all possible associations to just its first-order marginals. In the case of n tracks and n identities, the belief matrix B is an $n \times n$ doubly-stochastic matrix of non-negative entries b_{ij}, where b_{ij} is the probability that identity i is associated with track j. As we already saw in Section 4, the belief matrix approximation is equivalent to maintaining the zeroth- and first-order Fourier coefficients. Thus our current work is a strict generalization and extension of those previous results.

An alternative representation that has also been considered is an information theoretic approach (Shin et al., 2005; Schumitsch et al., 2005; Schumitsch et al., 2006) in which the density is parameterized as:

$$P(\sigma; \Omega) \propto \exp \operatorname{Tr}\left(\Omega^T \cdot \tau_{(n-1,1)}(\sigma)\right).$$

In our framework, the information form approach can be viewed as a method for maintaining the Fourier transform of the *log* probability distribution at only the first two irreducibles. The information matrix approach is especially attractive in a distributed sensor network setting, since, if the columns of the information matrix are distributed to leader nodes tracking the respective targets, then the observation events become entirely local operations, avoiding the more expensive Kronecker conditioning algorithm in our setting. On the other hand, the information matrix coefficients do not have the same intuitive marginals interpretation afforded in our setting, and moreover, prediction/rollup steps cannot be performed analytically in the information matrix form. As in many classical data structures problems there are representation trade-off issues: some operations are less expensive in one representation and some operations in the the other. The best choice in any particular scenario will depend on the ratio between observation and mixing events.

9.2 Previous work on Fourier-based approximations

The concept of using Fourier transforms to study probability distributions on groups is not new, with the earliest papers in this area having been published in the 1960s (Grenander, 1963). (Willsky, 1978) was the first to formulate the exact filtering problem in the Fourier domain for finite and locally compact Lie groups and contributed the first noncommutative Fast Fourier Transform algorithm (for Metacyclic groups). However, he does not address approximate inference, suggesting instead to always transform to the appropriate domain for which either the prediction/rollup or conditioning operations can be accomplished using a pointwise product. While providing significant improvements in complexity for smaller groups, his approach is still infeasible for our problem given the factorial order of the Symmetric group.

(Diaconis, 1988) utilized the Fourier transform to analyze probability distributions on the Symmetric group in order to study card shuffling and ranking problems. His work laid the ground for much of the progress made over the last two decades on probabilistic group theory and noncommutative FFT algorithms (Clausen & Baum, 1993; Rockmore, 2000).

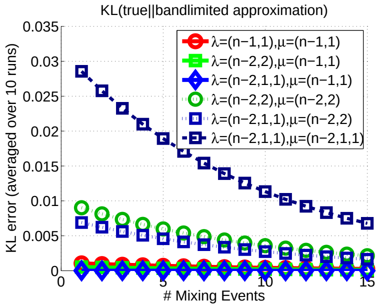
(Kondor et al., 2007) was the first to show that the data association problem could be efficiently approximated using FFT factorizations. In contrast to our framework where every model is assumed to be have been specified in the Fourier domain, they work with an observation model which can be written in the primal domain.

Conceptually, their conditioning algorithm applies the Inverse Fast Fourier Transform (IFFT) to the prior distribution, conditions in the primal domain using pointwise multiplication, then transforms back up to the Fourier domain using the FFT to obtain posterior Fourier coefficients. While their procedure would ordinarily be intractable because of the factorial number of permutations, they show that for simple observation models, such as that given in Equation 8.1, it is not necessary to perform the full FFT recursion to do a pointwise product. They exploit this observation to formulate a conditioning algorithm whose running time depends on the complexity of the observation model (which can roughly be measured by the number of irreducibles required to fully specify it). In the worst case, when the likelihood function is specified for each $\sigma \in S_n$, then the cost of conditioning is dominated by the cost of calling an FFT, which is $O(n! \log n!)$.

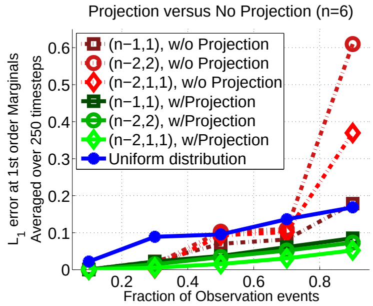
In the case that the observation model is specified at sufficiently many irreducibles, our conditioning algorithm (prior to the projection step) returns the same approximate probabilities as the FFT-based algorithm. For example, we can show that the observation model given in Equation 8.1 is fully specified by two Fourier components, and that both algorithms have identical output. In this setting, our asymptotic time complexity is $O(D^3n^2)$, where D is the degree of the largest maintained irreducible representation. The FFT-based algorithm saves a factor of D due to the fact that certain representation matrices can be shown to be sparse. Though we do not prove it, we observe that the Clebsch-Gordan coefficients C_{ij} are typically similarly sparse (see Figure 7(d)), which yields an equivalent running time in practice. In addition, Kondor et al. do not address the issue of projecting onto legal distributions, which, as we show in our experimental results is fundamental in practice.

10 Experimental results

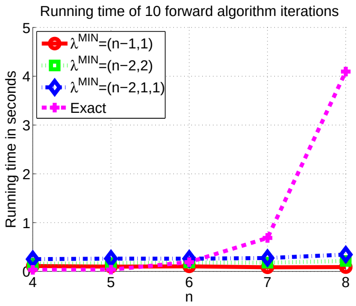
In this section we present the results of several experiments to validate our algorithm. We evaluate performance first by measuring the quality of our approximation for problems where the true distribution is known. Instead of measuring a distance between the true distribution and the inverse Fourier transform of our approximation, it makes more sense in our setting to measure error only at the marginals which are maintained by our approximation. In the results reported below, we measure the L_1 error between the true matrix of marginals and the approximation. If nonnegative marginal probabilities are guaranteed, it also makes sense to measure KL-divergence.



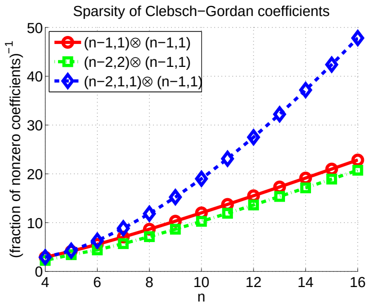
(a) Kronecker Conditioning Accuracy — we measure the accuracy of a single Kronecker conditioning operation after some number of mixing events.



(b) HMM Accuracy — we measure the average accuracy of posterior marginals over 250 timesteps, varying the proportion of mixing and observation events



(c) Running times: We compared running times of our polynomial time bandlimited inference algorithms against an exact algorithm with $O(n^3 n!)$ time complexity



(d) Clebsch-Gordan Sparsity: We measured the sparsity of the Clebsch-Gordan coefficients matrices by plotting the reciprocal of the fraction of nonzero entries against n. A straight line in the plot means that the number of nonzero entries scales linearly in n, and a convex curve scales better than linearly.

Figure 7:

10.1 Simulated data

We first tested the accuracy of a single Kronecker conditioning step by calling some number of pairwise mixing events (which can be thought roughly as a measure of entropy), followed by a single first-order observation. In the y-axis of Figure 7(a), we plot the Kullback-Leibler divergence between the true first-order marginals and approximate first-order marginals returned by Kronecker conditioning. We compared the results of maintaining first-order, and second-order (unordered and ordered) marginals. As shown in Figure 7(a), Kronecker conditioning is more accurate when the prior is smooth and unsurprisingly, when we allow for higher order Fourier terms. As guaranteed by Theorem 21, we also see that the first-order terms of the posterior are exact when we maintain second-order (ordered) marginals.

To understand how our algorithms perform over many timesteps (where errors can propagate to all Fourier terms), we compared to exact inference on synthetic datasets in which tracks are drawn at random to be observed or swapped. As a baseline, we show the accuracy of a uniform distribution. We observe that the Fourier approximation is better when there are either more mixing events (the fraction of conditioning events is smaller), or when more Fourier coefficients are maintained, as shown in Figure 7(b). We also see that the Plancherel Projection step is fundamental, especially when mixing events are rare.

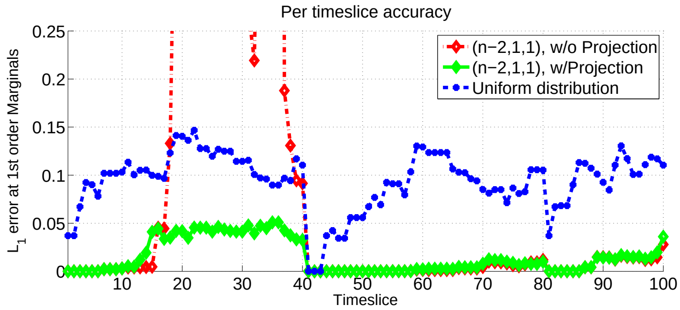
Figures 8(a) and 8(b) show the per-timeslice accuracy of two typical runs of the algorithm. The fraction of conditioning events is 50% in Figure 8(a), and 70% in Figure 8(b). What we typically observe is that while the projected and nonprojected accuracies are often quite similar, the nonprojected marginals can perform significantly worse during certain segments.

Finally, we compared running times against an exact inference algorithm which performs prediction/rollup in the Fourier domain and conditioning in the primal domain. Instead of the naive $O((n!)^2)$ complexity, its running time is a more efficient $O(n^3 n!)$ due to the Fast Fourier Transform (Clausen & Baum, 1993). It is clear that our algorithm scales gracefully compared to the exact solution (Figure 7(c)), and in fact, we could not run exact inference for $n > 8$ due to memory constraints. In Figure 7(d), we show empirically that the Clebsch-Gordan coefficients are indeed sparse, supporting our conjectured runtime of $O(D^2 n^2)$ instead of $O(D^3 n^2)$.

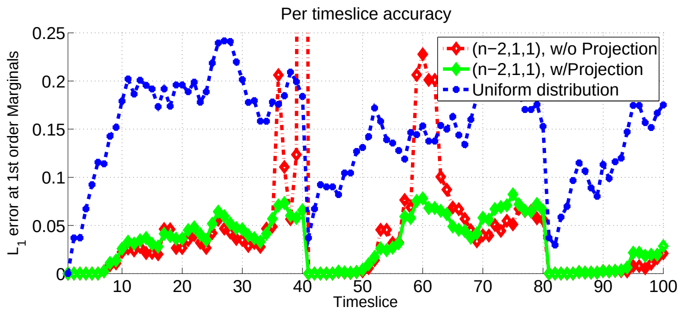
10.2 Real camera network

We also evaluated our algorithm on data taken from a real network of eight cameras (Fig. 9(a)). In the data, there are $n = 11$ people walking around a room in fairly close proximity. To handle the fact that people can freely leave and enter the room, we maintain a list of the tracks which are external to the room. Each time a new track leaves the room, it is added to the list and a mixing event is called to allow for m^2 pairwise swaps amongst the m external tracks.

The number of mixing events is approximately the same as the number of observations. For each observation, the network returns a color histogram of the blob associated with one track track. The task after conditioning on each observation is to predict identities for all tracks which are inside the room, and the evaluation metric is the fraction of accurate predictions. We compared against a baseline approach of predicting the identity of a track based on the most recently observed histogram at that track. This approach is expected to be accurate when there are many observations and discriminative appearance models, neither of which our problem afforded. As Figure 9(b) shows, both the baseline and first order model(without projection) fared poorly, while the projection step dramatically boosted the prediction accuracy for this problem. To illustrate the difficulty of predicting based on appearance alone, the rightmost bar reflects the performance of an *omniscient* tracker who knows the result of



(a) $n = 6$ with 50% mixing events and 50% observations



(b) $n = 6$ with 30% mixing events and 70% observations

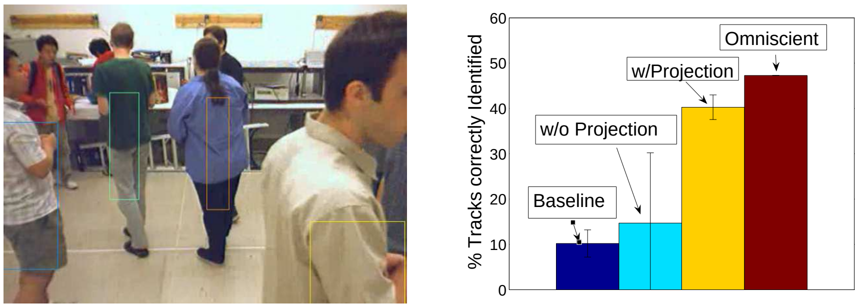
Figure 8: Accuracy as a function of time on two typical runs.

each mixing event and is therefore left only with the task of distinguishing between appearances. We conjecture that the performance of our algorithm (with projection) is near optimal.

11 Future research

There remain several possible extensions to the current work stemming from both practical and theoretical considerations. We list a few open questions and extensions in the following.

Adaptive filtering. While our current algorithms easily beat exact inference in terms of running time, they are still limited by a relatively high (though polynomial) time complexity. In practice however, it seems reasonable to believe that the "difficult" identity management problems typically involve only a small subset of people at a time. A useful extension of our work would be to devise an *adaptive* version of the algorithm which allocates more Fourier coefficients towards the identities which require higher order reasoning. We believe that this kind of extension would be the appropriate way to scale our algorithm to handling massive numbers of objects at a time.



(a) Sample Image (b) Accuracy for Camera Data

Figure 9: Evaluation on dataset from a real camera network.

Characterizing the marginal polytope. In our paper, we presented a projection of the bandlimited distribution to a certain polytope, which is exactly the marginal polytope for first-order bandlimited distributions, but strictly an outer bound for higher orders. An interesting project would be to generalize the Birkhoff-von Neumann theorem by exactly characterizing the marginal polytope at higher order marginals. We conjecture that the marginal polytope for low order marginals can be described with polynomially many constraints.

Learning in the Fourier domain. Another interesting problem is whether we can learn bandlimited mixing and observation models *directly in the Fourier domain*. Given fully observed permutations $\sigma_1, \ldots, \sigma_m$, drawn from a distribution $P(\sigma)$, a naive method for estimating $\hat{P}_\rho$ at low-order ρ is to simply observe that:

$$\hat{P}_\rho = \mathbb{E}_{\sigma \sim P}[\rho(\sigma)],$$

and so one can estimate the Fourier transform by simply averaging $\rho(\sigma_i)$ over all σ_i. However, since we typically do not observe full permutations in real applications like ranking or identity management, it would be interesting to estimate Fourier transforms using partially observed data. In the case of Bayesian learning, it may be possible to apply some of the techniques discussed in this paper.

Probabilistic inference on other groups. The Fourier theoretic framework presented in this paper is not specific to the Symmetric group - in fact, the prediction/rollup and conditioning formulations, as well as most of the results from Appendix B hold over any finite or compact Lie group. As an example, the noncommutative group of rotation operators in three dimensions, $SO(3)$, appears in settings which model the pose of a three dimensional object. Elements in $SO(3)$ might be used to represent the pose of a robot arm in robotics, or the orientation of a mesh in computer graphics; In many settings, it would be useful to have a compact representation of uncertainty over poses. We believe that there are many other application domains with algebraic structure where similar probabilistic inference algorithms might apply, and in particular, that noncommutative settings offer a particularly challenging but exciting opportunity for machine learning research.

12 Conclusions

We have presented an intuitive method for compactly summarizing distributions on permutations with Fourier analytic interpretations and tuneable approximation quality. We showed that the Fourier theoretic point of view makes it possible to formulate general inference operations completely in the Fourier domain. In particular, we developed the Kronecker Conditioning algorithm which performs a convolution-like operation on Fourier coefficients to find the Fourier transform of the posterior distribution. We analyzed the sources of error in our approximations and argued that bandlimited conditioning can result in Fourier coefficients which correspond to no valid distribution, but that the problem can be remedied by projecting to a relaxation of the marginal polytope.

Our evaluation on data from a camera network shows that our methods perform well when compared to the optimal solution in small problems, or to an omniscient tracker in large problems. Furthermore, we demonstrated that our projection step is fundamental to obtaining these high-quality results.

Finally we conclude by remarking again that the mathematical framework developed in our paper is quite general. In fact, both the prediction/rollup and conditioning formulations hold over any finite group, providing a principled method for approximate inference for problems with underlying group structure.

Acknowledgments

This work is supported in part by the Office of Naval Research under MURI N000140710747, the Army Research Office under grant W911NF-06-1-0275, the National Science Foundation under grants DGE-0333420, EEEC-540865, NeTS-NOSS 0626151 and TF 0634803, and by the Pennsylvania Infrastructure Technology Alliance (PITA). Carlos Guestrin was also supported in part by an Alfred P. Sloan Fellowship. We are grateful to Kyle Heath for collecting the camera data and to Robert Hough and Emre Oto for helpful and insightful discussions.

References

Balakrishnan, H., Hwang, I., & Tomlin, C. J. (2004). Polynomial approximation algorithms for belief matrix maintenance in identity management. *Proceedings of the 43rd IEEE Conference on Decision and Control, Bahamas.*

Bar-Shalom, Y., & Fortmann, T. (1988). *Tracking and data association.* Academic Press.

Boyen, X., & Koller, D. (1998). Tractable inference for complex stochastic processes. *UAI*.

Chen, J.-Q. (1989). *Group representation theory for physicists*. World Scientific.

Clausen, M., & Baum, U. (1993). Fast fourier transforms for symmetric groups: Theory and implementation. *Mathematics of Computations*, *61*, 833–847.

Collins, J., & Uhlmann, J. (1992). Efficient gating in data association with multivariate distributed states. *IEEE Trans. Aerospace and Electronic Systems*, *28*.

Cox, I., & Hingorani, S. (1994). An efficient implementation of Reid's multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking. *International Conf. on Pattern Recognition* (pp. 437–443).

Critchlow, D. E. (1985). *Metric methods for analyzing partially ranked data*. Springer-Verlag.

Diaconis, P. (1988). *Group Representations in Probability and Statistics*. Institute of Mathematical Statistics.

Diaconis, P. (1989). A generalization of spectral analysis with application to ranked data. *The Annals of Statistics*, *17*, 949–979.

Fligner, M., & Verducci, J. (1986). Distance based ranking models. *Journal of the Royal Statistical Society*, *48*.

Freund, Y., Iyer, R., Schapire, R. E., & Singer, Y. (2003). An efficient boosting algorithm for combining preferences. *JMLR.*, *4*, 933–969.

Grenander, U. (1963). *Probabilities on algebraic structures*. Wiley.

Helmbold, D. P., & Warmuth, M. K. (2007). Learning permutations with exponential weights. *COLT*.

Huang, J., Guestrin, C., & Guibas, L. (2007). Efficient inference for distributions on permutations. *Advances in Neural Information Processing Systems (NIPS)*. Vancouver, Canada.

James, G., & Kerber, A. (1981). *The representation theory of the symmetric group*. Addison-Wesley.

Joachims, T. (2002). Optimizing search engines using clickthrough data. *KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 133–142). New York, NY, USA: ACM.

Kondor, R. (2006). $\mathbb{S}_n$`ob`: a C++ library for fast Fourier transforms on the symmetric group. Available at `http://www.cs.columbia.edu/~risi/Snob/`.

Kondor, R., Howard, A., & Jebara, T. (2007). Multi-object tracking with representations of the symmetric group. *AISTATS*.

Kueh, K., Olson, T., Rockmore, D., & Tan, K. (1999). *Nonlinear approximation theory on finite groups* (Technical Report PMA-TR99-191). Department of Mathematics, Dartmouth College.

Lebanon, G., & Mao, Y. (2008). Non-parametric modeling of partially ranked data. *Advances in Neural Information Processing Systems 20* (pp. 857–864). Cambridge, MA: MIT Press.

Mallows, C. (1957). Non-null ranking models. *Biometrika*, *44*.

Maslen, D. (1998). The efficient computation of fourier transforms on the symmetric group. *Mathematics of Computation*, *67*, 1121–1147.

Murnaghan, F. (1938). The analysis of the kronecker product of irreducible representations of the symmetric group. *American Journal of Mathematics*, *60*, 761–784.

Murty, K. (1968). An algorithm for ranking all the assignments in order of increasing cost. *Operations Research*, *16*, 682–687.

Oh, S., Russell, S., & Sastry, S. (2004). Markov chain Monte Carlo data association for general multiple-target tracking problems. *Proc. of the IEEE International Conference on Decision and Control, Paradise Island, Bahamas.*

Oh, S., & Sastry, S. (2005). A polynomial-time approximation algorithm for joint probabilistic data association. *Proc. of the American Control Conference, Portland, OR.*

Poore, A. (1995). Multidimensional assignment and multitarget tracking. In *Partitioning data sets*, vol. 19, 169–196. DIMACS Series in Discrete Mathematics and Theoretical Computer Science.

Rabiner, L. (1989). A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, *77*, 257–286.

Reid, D. (1979). An algorithm for tracking multiple targets. *IEEE Trans. on Automatic Control*, *6*, 843–854.

Rockmore, D. N. (2000). The fft: An algorithm the whole family can use. *Computing in Science and Engineering*, *02*, 60–64.

Sagan, B. E. (2001). *The symmetric group.* Springer.

Schumitsch, B., Thrun, S., Bradski, G., & Olukotun, K. (2005). The information-form data association filter. *Proceedings of Conference on Neural Information Processing Systems (NIPS).* Cambridge, MA: MIT Press.

Schumitsch, B., Thrun, S., Guibas, L., & Olukotun, K. (2006). The identity management Kalman filter (imkf). *Proceedings of Robotics: Science and Systems*. Philadelphia, PA, USA.

Serre, J. P. (1977). *Linear representations of finite groups*. Springer-Verlag.

Shin, J., Guibas, L., & Zhao, F. (2003). A distributed algorithm for managing multi-target identities in wireless ad-hoc sensor networks. *IPSN*.

Shin, J., Lee, N., Thrun, S., & Guibas, L. (2005). Lazy inference on object identities in wireless sensor networks. *IPSN '05*.

Taylor, M., Guiver, J., Robertson, S., & Minka, T. (2008). Softrank: optimizing non-smooth rank metrics. *WSDM '08: Proceedings of the international conference on Web search and web data mining* (pp. 77–86). New York, NY, USA: ACM.

Terras, A. (1999). *Fourier analysis on finite groups and applications*. London Mathematical Society.

van Lint, J., & Wilson, R. (2001). *A course in combinatorics*. Cambridge University Press.

van Loan, C. F. (2000). The ubiquitous kronecker product. *J. Comput. Appl. Math.*, *123*, 85–100.

Vershik, A., & Okounkov, A. (2006). A new approach to the representation theory of symmetric groups. ii. *Journal of Mathematical Sciences*, *131*, 5471–5494.

Willsky, A. (1978). On the algebraic structure of certain partially observable finite-state markov processes. *Information and Control*, *38*, 179–212.

A Constructing irreducible representation matrices

In this section, we present (without proof) some standard algorithms for constructing the irreducible representation matrices with respect to the *Gel'fand-Tsetlin (GZ) basis* (for a more elaborate discussion, see, for example, (Kondor, 2006; Chen, 1989; Vershik & Okounkov, 2006)). There are several properties which make the irreducible representation matrices, written with respect to the GZ basis, fairly useful in practice. They are guaranteed to be, for example, real-valued and orthogonal. And as we will show, the matrices have certain useful sparsity properties that can be exploited in implementation.

We begin by introducing a few concepts relating to *Young tableaux* which are like Young tabloids with the distinction that the rows are considered as *ordered tuples* rather than *unordered sets*. For example, the following two diagrams are distinct as Young *tableaux*, but not as Young *tabloids*:

$$\begin{array}{|c|c|c|}\hline 1&2&3\\ \hline 4&5\\ \cline{1-2}\end{array} \neq \begin{array}{|c|c|c|}\hline 1&3&2\\ \hline 5&4\\ \cline{1-2}\end{array} \qquad \text{(as Young tableaux).}$$

A Young Tableau t is said to be *standard* if its entries are increasing to the right along rows and down columns. For example, the set of all standard Young Tableaux of shape $\lambda = (3,2)$ is:

$$\left\{\begin{array}{|c|c|c|}\hline 1&3&5\\ \hline 2&4\\ \cline{1-2}\end{array},\ \begin{array}{|c|c|c|}\hline 1&2&5\\ \hline 3&4\\ \cline{1-2}\end{array},\ \begin{array}{|c|c|c|}\hline 1&3&4\\ \hline 2&5\\ \cline{1-2}\end{array},\ \begin{array}{|c|c|c|}\hline 1&2&4\\ \hline 3&5\\ \cline{1-2}\end{array},\ \begin{array}{|c|c|c|}\hline 1&2&3\\ \hline 4&5\\ \cline{1-2}\end{array}\right\}. \tag{A.1}$$

Given a permutation $\sigma \in S_n$, one can always apply σ to a Young tableau t to get a new Young tableau, which we denote by $\sigma \circ t$, by permuting the labels within the tableau. For example,

$$(1,2)\circ \begin{array}{|c|c|c|}\hline \mathbf{1}&\mathbf{2}&3\\ \hline 4&5\\ \cline{1-2}\end{array} = \begin{array}{|c|c|c|}\hline \mathbf{2}&\mathbf{1}&3\\ \hline 4&5\\ \cline{1-2}\end{array}.$$

Note, however, that even if t is a standard tableau, $\sigma \circ t$ is not guaranteed to be standard.

The significance of the standard tableaux is that the set of all standard tableaux of shape λ can be used to index the set of GZ basis vectors for the irreducible representation ρ_λ. Since there are five total standard tableaux of shape $(3,2)$, we see, for example, that the irreducible corresponding to the partition $(3,2)$ is 5-dimensional. There is a simple recursive procedure for enumerating the set of all standard tableaux of shape λ, which we illustrate for $\lambda = (3,2)$.

Example 25. *If $\lambda = (3,2)$, there are only two possible boxes that the label 5 can occupy so that both rows and columns are increasing. They are:*

$$\begin{array}{|c|c|c|}\hline \ &\ &5\\ \hline \ &\ \\ \cline{1-2}\end{array},\ \textit{and}\ \begin{array}{|c|c|c|}\hline \ &\ &\ \\ \hline \ &5\\ \cline{1-2}\end{array}.$$

To enumerate the set of all standard tableaux of shape $(3,2)$, we need to fill the empty boxes in the above partially filled tableaux with the labels $\{1,2,3,4\}$ so that both rows and columns are increasing. Enumerating the standard tableaux of shape $(3,2)$ thus reduces to enumerating the set of standard tableaux of shapes $(2,2)$ and $(3,1)$, respectively. For $(2,2)$, the set of standard tableaux (which, in implementation would be computed recursively) is:

$$\left\{\begin{array}{|c|c|}\hline 1&3\\ \hline 2&4\\ \hline\end{array},\ \begin{array}{|c|c|}\hline 1&2\\ \hline 3&4\\ \hline\end{array}\right\},$$

and for $(3,1)$, the set of standard tableaux is:

$$\left\{\begin{array}{|c|c|c|}\hline 1&3&4\\ \hline 2\\ \cline{1-1}\end{array},\ \begin{array}{|c|c|c|}\hline 1&2&4\\ \hline 3\\ \cline{1-1}\end{array},\ \begin{array}{|c|c|c|}\hline 1&2&3\\ \hline 4\\ \cline{1-1}\end{array}\right\}.$$

The entire set of standard tableaux of shape $(3,2)$ is therefore:

$$\left\{\begin{array}{|c|c|c|}\hline 1&3&\mathbf{5}\\ \hline 2&4\\ \cline{1-2}\end{array},\ \begin{array}{|c|c|c|}\hline 1&2&\mathbf{5}\\ \hline 3&4\\ \cline{1-2}\end{array}\right\}\bigcup\left\{\begin{array}{|c|c|c|}\hline 1&3&4\\ \hline 2&\mathbf{5}\\ \cline{1-2}\end{array},\ \begin{array}{|c|c|c|}\hline 1&2&4\\ \hline 3&\mathbf{5}\\ \cline{1-2}\end{array},\ \begin{array}{|c|c|c|}\hline 1&2&3\\ \hline 4&\mathbf{5}\\ \cline{1-2}\end{array}\right\}.$$

Before explicitly constructing the representation matrices, we must define a signed distance on Young Tableaux called the *axial distance.*

Definition 26. The *axial distance*, $d_t(i,j)$, between entries i and j in tableau t, is defined to be:

$$d_t(i,j) \equiv (col(t,j) - col(t,i)) - (row(t,j) - row(t,i)),$$

where $row(t,i)$ denotes the row of label i in tableau t, and $col(t,i)$ denotes the column of label i in tableau t.

Intuitively, the axial distance between $i-1$ and i in a standard tableau t is equal to the (signed) number of steps that are required to travel from $i-1$ to i, if at each step, one is allowed to traverse a single box in the tableau in one of the four cardinal directions. For example, the axial distance from 3 to 4 with respect to tableau: $t = \begin{array}{|c|c|c|}\hline 1&2&3\\\hline 4&5\\\cline{1-2}\end{array}$ is:

$$\begin{aligned} d_t(3,4) &= \left(col\left(\begin{array}{|c|c|c|}\hline 1&2&3\\\hline 4&5\\\cline{1-2}\end{array}, 4\right) - col\left(\begin{array}{|c|c|c|}\hline 1&2&3\\\hline 4&5\\\cline{1-2}\end{array}, 3\right)\right) - \left(row\left(\begin{array}{|c|c|c|}\hline 1&2&3\\\hline 4&5\\\cline{1-2}\end{array}, 4\right) - row\left(\begin{array}{|c|c|c|}\hline 1&2&3\\\hline 4&5\\\cline{1-2}\end{array}, 3\right)\right) \\ &= (1-3) - (2-1) = -3 \end{aligned}$$

A.1 Constructing representation matrices for adjacent transpositions

In the following discussion, we will consider a fixed ordering, $t_1, \ldots, t_{d_\lambda}$, on the set of standard tableaux of shape λ and refer to both standard tableaux and columns of $\rho_\lambda(\sigma)$ interchangeably. Thus t_1 refers to first column, t_2 refers to the second column and so on. And we will index elements in $\rho_\lambda(\sigma)$ using pairs of standard tableau, (t_j, t_k).

To explicitly define the representation matrices with respect to the GZ basis, we will first construct the matrices for adjacent transpositions (i.e., permutations of the form $(i-1,i)$), and then we will construct arbitrary representation matrices by combining the matrices for the adjacent transpositions. The rule for constructing the matrix coefficient $[\rho_\lambda(i-1,i)]_{t_j,t_k}$ is as follows.

1. Define the (t_j,t_k) coefficient of $\rho_\lambda(i-1,i)$ to be zero if it is (1), off-diagonal ($j \neq k$) and (2), not of the form $(t_j, (i-1,i)\circ t_k)$.
2. If (t_j,t_k) is a diagonal element, (i.e., of the form (t_j,t_j)), define:
$$[\rho_\lambda(i-1,i)]_{t_j,t_j} = 1/d_{t_j}(i-1,i),$$
where $d_{t_j}(i-1,i)$ is the axial distance which we defined earlier in the section.
3. If (t_j,t_k) can be written as $(t_j,(i-1,i)\circ t_j)$ define:
$$[\rho_\lambda(i-1,i)]_{t_j,\sigma\circ t_j} = \sqrt{1 - 1/d^2_{t_j}(i-1,i)}.$$

Note that the only time that off-diagonal elements can be nonzero under the above rules is when $(i-i,i)\circ t_j$ happens to also be a standard tableau. If we apply an adjacent transposition, $\sigma = (i-1,i)$ to a standard tableau t, then $\sigma\circ t$ is guaranteed to be standard if and only if $i-1$ and i were neither in the same row nor column of t. This can be seen by examining each case separately.

Algorithm 3: Pseudocode for computing irreducible representations matrices with respect to the Gel'fand-Tsetlin basis at adjacent transpositions.

```
ADJACENTRHO
input  : i ∈ {2, ..., n}, λ
output: ρ_λ(i − 1, i)
ρ ← 0_{d_λ × d_λ};
foreach standard tableaux t of shape λ do
    d ← (col(t, i) − col(t, i − 1)) − (row(t, i) − row(t, i − 1));
    ρ(t, t) ← 1/d;
    if i − 1 and i are in different rows and columns of t then
        ρ((i − 1, i)(t), t) ← sqrt(1 − 1/d^2);
return ρ ;
```

1. $i-1$ **and** i **are in the same row or same column of** t**.** If i and $i-1$ are in the same row of t, then $i-1$ lies to the left of i. Applying $\sigma \circ t$ swaps their positions so that i lies to the left of $i-1$, and so we see that $\sigma \circ t$ cannot be standard. For example,

$$(3,4)\circ \begin{array}{|c|c|c|}\hline 1&2&5\\\hline \mathbf{3}&\mathbf{4}\\\cline{1-2}\end{array} = \begin{array}{|c|c|c|}\hline 1&2&5\\\hline \mathbf{4}&\mathbf{3}\\\cline{1-2}\end{array}.$$

   Similarly, we see that if i and $i-1$ are in the same column of t, $\sigma \circ t$ cannot be standard. For example,

$$(3,4)\circ \begin{array}{|c|c|c|}\hline 1&\mathbf{3}&5\\\hline 2&\mathbf{4}\\\cline{1-2}\end{array} = \begin{array}{|c|c|c|}\hline 1&\mathbf{4}&5\\\hline 2&\mathbf{3}\\\cline{1-2}\end{array}.$$

2. $i-1$ **and** i **are neither in the same row nor column of** t**.** In the second case, $\sigma \circ t$ can be seen to be a standard tableau due to the fact that $i-1$ and i are adjacent indices. For example,

$$(3,4)\circ \begin{array}{|c|c|c|}\hline 1&2&\mathbf{3}\\\hline \mathbf{4}&5\\\cline{1-2}\end{array} = \begin{array}{|c|c|c|}\hline 1&2&\mathbf{4}\\\hline \mathbf{3}&5\\\cline{1-2}\end{array}.$$

Therefore, to see if $(i-1,i)\circ t$ is standard, we need only check to see that $i-1$ and i are in different rows and columns of the tableau t. The pseudocode for constructing the irreducible representation matrices for adjacent swaps is summarized in Algorithm 3. Note that the matrices constructed in the algorithm are sparse, with no more than two nonzero elements in any given column.

Example 27. *We compute the representation matrix of* $\rho_{(3,2)}$ *evaluated at the adjacent transposition* $\sigma = (i-1,i) = (3,4)$*. For this example, we will use the enumeration of the standard tableaux of shape* $(3,2)$ *given in Equation A.1.*

For each $(3,2)$*-tableau* t_j*, we identify whether* $\sigma \circ t_j$ *is standard and compute the axial distance from* 3 *to* 4 *on the tableau* t_j*.*

j	*1*	*2*	*3*	*4*	*5*
t_j	$\begin{matrix}1&\mathbf{3}&5\\2&\mathbf{4}&\end{matrix}$	$\begin{matrix}1&2&5\\\mathbf{3}&\mathbf{4}&\end{matrix}$	$\begin{matrix}1&\mathbf{3}&\mathbf{4}\\2&5&\end{matrix}$	$\begin{matrix}1&2&\mathbf{4}\\\mathbf{3}&5&\end{matrix}$	$\begin{matrix}1&2&\mathbf{3}\\\mathbf{4}&5&\end{matrix}$
$(3,4)\circ t_j$	$\begin{matrix}1&\mathbf{4}&5\\2&\mathbf{3}&\end{matrix}$	$\begin{matrix}1&2&5\\\mathbf{4}&\mathbf{3}&\end{matrix}$	$\begin{matrix}1&\mathbf{4}&\mathbf{3}\\2&5&\end{matrix}$	$\begin{matrix}1&2&\mathbf{3}\\\mathbf{4}&5&\end{matrix}$	$\begin{matrix}1&2&\mathbf{4}\\\mathbf{3}&5&\end{matrix}$
$(3,4)\circ t_j$ *Standard?*	*No*	*No*	*No*	*Yes*	*Yes*
axial distance ($d_{t_j}(3,4)$)	*-1*	*1*	*1*	*3*	*-3*

Putting the results together in a matrix yields:,

$$\rho_{(3,2)}(3,4) = \left[\begin{array}{c|ccccc} & t_1 & t_2 & t_3 & t_4 & t_5 \\ \hline t_1 & -1 & & & & \\ t_2 & & 1 & & & \\ t_3 & & & 1 & & \\ t_4 & & & & \frac{1}{3} & \sqrt{\frac{8}{9}} \\ t_5 & & & & \sqrt{\frac{8}{9}} & -\frac{1}{3} \end{array}\right],$$

where all of the empty entries are zero.

A.2 Constructing representation matrices for general permutations

To construct representation matrices for general permutations, it is enough to observe that all permutations can be factored into a sequence of adjacent swaps. For example, the permutation $(1,2,5)$ can be factored into:

$$(1,2,5) = (4,5)(3,4)(1,2)(2,3)(3,4)(4,5),$$

and hence, for any partition λ,

$$\rho_\lambda(1,2,5) = \rho_\lambda(4,5)\cdot\rho_\lambda(3,4)\cdot\rho_\lambda(1,2)\cdot\rho_\lambda(2,3)\cdot\rho_\lambda(3,4)\cdot\rho_\lambda(4,5),$$

since ρ_λ is a group representation. Algorithmically, factoring a permutation into adjacent swaps looks very similar to the Bubblesort algorithm, and we show the pseudocode in Algorithm 4.

B Decomposing the tensor product representation

We now turn to the *Tensor Product Decomposition* problem, which is that of finding the irreducible components of the typically reducible tensor product representation. If ρ_λ and ρ_μ are irreducible representations of S_n, then there exists an intertwining operator $C_{\lambda\mu}$ such that:

$$C_{\lambda\mu}{}^{-1}\cdot(\rho_\lambda\otimes\rho_\mu(\sigma))\cdot C_{\lambda\mu} = \bigoplus_\nu\bigoplus_{\ell=1}^{z_{\lambda\mu\nu}}\rho_\nu(\sigma). \qquad \text{(B.1)}$$

Algorithm 4: Pseudocode for computing irreducible representation matrices for arbitrary permutations.

GETRHO
input : $\sigma \in S_n, \lambda$
output: $\rho_\lambda(\sigma)$ (a $d_\lambda \times d_\lambda$ matrix)

```
//Use Bubblesort to factor σ into a product of transpositions
k ← 0 ;
factors ← ∅;
for i = 1, 2, ..., n do
    for j = n, n − 1, ..., i + 1 do
        if σ(j) < σ(j − 1) then
            Swap(σ(j − 1), σ(j)) ;
            k ← k + 1 ;
            factors(k) ← j ;
//Construct representation matrix using adjacent transpositions
ρ_λ(σ) ← I_{d_λ × d_λ} ;
m ← length(factors);
for j = 1, ..., m do
    ρ_λ(σ) ← GETADJACENTRHO (factors(j), λ) · ρ_λ(σ) ;
```

In this section, we will present a set of numerical methods for computing the Clebsch-Gordan series ($z_{\lambda\mu\nu}$) and Clebsch-Gordan coefficients ($C_{\lambda\mu}$) for a pair of irreducible representations $\rho_\lambda \otimes \rho_\mu$. We begin by discussing two methods for computing the Clebsch-Gordan series. In the second section, we provide a general algorithm for computing the intertwining operators which relate two equivalent representations and discuss how it can be applied to computing the Clebsch-Gordan coefficients (Equation B.1) and the matrices which relate marginal probabilities to irreducible Fourier coefficients (Equation 5.4).

B.1 Computing the Clebsch-Gordan series

We begin with a simple, well-known algorithm based on *group characters* for computing the Clebsch-Gordan series that turns out to be computationally intractable, but yields several illuminating theoretical results. See (Serre, 1977) for proofs of the theoretical results cited in this section.

One of the main results of representation theory was the discovery that there exists a relatively compact way of encoding any representation up to equivalence with a vector which we call the *character* of the representation. If ρ is a representation of a group G, then the character of the representation ρ, is defined simply to be the trace of the representation at each element $\sigma \in G$:

$$\chi_\rho(\sigma) = \mathrm{Tr}\left(\rho(\sigma)\right).$$

The reason characters have been so extensively studied is that they uniquely characterize a representation up to equivalence in the sense that two characters

χ_{ρ_1} and χ_{ρ_2} are equal if and only if ρ_1 and ρ_2 are equivalent as representations. Even more surprising is that the space of possible group characters is orthogonally spanned by the characters of the irreducible representations. To make this precise, we first define an inner product on functions from G.

Definition 28. Let ϕ, ψ be two real-valued functions on G. The *inner product* of ϕ and ψ is defined to be:

$$\langle \phi, \psi \rangle \equiv \frac{1}{|G|} \sum_{\sigma \in G} \phi(\sigma)\psi(\sigma)$$

With respect to the above inner product, we have the following important result which allows us to test a given representation for irreducibility, and to test two irreducibles for equivalence.

Proposition 29. *Let χ_{ρ_1} and χ_{ρ_2} be characters corresponding to irreducible representations. Then*

$$\langle \chi_{\rho_1}, \chi_{\rho_2} \rangle = \begin{cases} 1 & \text{if } \rho_1 \equiv \rho_2 \\ 0 & \text{otherwise} \end{cases} .$$

Proposition 29 shows that the irreducible characters form an orthonormal set of functions. The next proposition says that the irreducible characters *span* the space of all possible characters.

Proposition 30. *Suppose ρ is any representation of G and which decomposes into irreducibles as:*

$$\rho \equiv \bigoplus_{\lambda} \bigoplus_{\ell=1}^{z_\lambda} \rho_\lambda,$$

where λ indexes over all irreducibles of G. Then:

1. *The character of ρ is a linear combination of irreducible characters ($\chi_\rho = \sum_\lambda z_\lambda \chi_{\rho_\lambda}$),*
2. *and the multiplicity of each irreducible, z_λ, can be recovered using $\langle \chi_\rho, \chi_{\rho_\lambda} \rangle = z_\lambda$.*

A simple way to decompose any group representation ρ, is given by Proposition 30, which says that we can take inner products of χ_ρ against the basis of irreducible characters to obtain the irreducible multiplicities z_λ. To treat the special case of finding the Clebsch-Gordan series, one observes that the character of the tensor product is simply the pointwise product of the characters of each tensor product factor.

Theorem 31. *Let ρ_λ and ρ_μ be irreducible representations with characters χ_λ, χ_μ respectively. Let $z_{\lambda\mu\nu}$ be the number of copies of ρ_ν in $\rho_\lambda \otimes \rho_\mu$ (hence, one term of the Clebsch-Gordan series). Then:*

1. *The character of the tensor product representation is given by:*

$$\chi_{\rho_\lambda \otimes \rho_\mu} = \chi_\lambda \cdot \chi_\mu = \sum_\nu z_{\lambda\mu\nu}\chi_\nu. \tag{B.2}$$

2. *The terms of the Clebsch-Gordan series can be computed using:*

$$z_{\lambda\mu\nu} = \frac{1}{|G|}\sum_{g\in G}\chi_\lambda(g)\cdot\chi_\mu(g)\cdot\chi_\nu(g), \tag{B.3}$$

and satisfy the following symmetry:

$$z_{\lambda\mu\nu} = z_{\lambda\nu\mu} = z_{\mu\lambda\nu} = z_{\mu\nu\lambda} = z_{\nu\lambda\mu} = z_{\nu\mu\lambda}. \tag{B.4}$$

Dot products for characters on the symmetric group can be done in $O(\#(n))$ time where $\#(n)$ is the number of partitions of the number n, instead of the naive $O(n!)$ time. In practice however, $\#(n)$ also grows too quickly for the character method to be tractable.

B.1.1 Murnaghan's formulas

A theorem by Murnaghan (Murnaghan, 1938) gives us a 'bound' on which representations can appear in the tensor product decomposition on S_n.

Theorem 32. *Let ρ_1, ρ_2 be the irreducibles corresponding to the partition $(n-p, \lambda_2, \dots)$ and $(n-q, \mu_2, \dots)$ respectively. Then the product $\rho_1 \otimes \rho_2$ does not contain any irreducibles corresponding to a partition whose first term is less than $n-p-q$.*

In view of the connection between the Clebsch-Gordan series and convolution of Fourier coefficients, Theorem 32 is analogous to the fact that for functions over the reals, the convolution of two compactly supported functions is also compactly supported.

We can use Theorem 32 to show that Kronecker conditioning is exact at certain irreducibles.

of Theorem 21. Let Λ denote the set of irreducibles at which our algorithm maintains Fourier coefficients. Since the errors in the prior come from setting coefficients outside of Λ to be zero, we see that Kronecker conditioning returns an approximate posterior which is exact at the irreducibles in

$$\Lambda_{EXACT} = \{\rho_\nu \,:\, z_{\lambda\mu\nu} = 0, \text{ where } \lambda \notin \Lambda \text{ and } \mu \trianglerighteq (n-q, \mu_2, \dots)\}.$$

Combining Theorem 32 with Equation B.4: if $z_{\lambda\mu\nu} > 0$, with $\lambda = (n-p, \lambda_2, \lambda_3, \dots), \mu = (n-q, \mu_2, \mu_3, \dots)$ and $\nu = (n-r, \nu_2, \nu_3, \dots)$, then we have that: $r \le p+q, p \le q+r$, and $q \le p+r$. In particular, it implies that $r \ge p-q$ and $r \ge q-p$, or more succinctly, $r \ge |p-q|$. Hence, if $\nu = (n-r, \nu_2, \dots)$, then $\rho_\nu \in \Lambda_{EXACT}$ whenever $r \le |p-q|$, which proves the desired result. □

The same paper (Murnaghan, 1938) derives several general Clebsch-Gordan series formulas for pairs of low-order irreducibles in terms of n, and in particular, derives the Clebsch-Gordan series for many of the Kronecker product pairs that one would likely encounter in practice. For example,

- $\rho_{(n-1,1)} \otimes \rho_{(n-1,1)} \equiv \rho_{(n)} \oplus \rho_{(n-1,1)} \oplus \rho_{(n-2,2)} \oplus \rho_{(n-2,1,1)}$
- $\rho_{(n-1,1)} \otimes \rho_{(n-2,2)} \equiv \rho_{(n-1,1)} \oplus \rho_{(n-2,2)} \oplus \rho_{(n-2,1,1)} \oplus \rho_{(n-3,3)} \oplus \rho_{(n-3,2,1)}$
- $\rho_{(n-1,1)} \otimes \rho_{(n-2,1,1)} \equiv \rho_{(n-1,1)} \oplus \rho_{(n-2,2)} \oplus \rho_{(n-2,1,1)} \oplus \rho_{(n-3,2,1)} \oplus \rho_{(n-3,1,1,1)}$
- $\rho_{(n-1,1)} \otimes \rho_{(n-3,3)} \equiv \rho_{(n-2,2)} \oplus \rho_{(n-3,3)} \oplus \rho_{(n-3,2,1)} \oplus \rho_{(n-4,4)} \oplus \rho_{(n-4,3,1)}$

B.2 Computing the Clebsch-Gordan coefficients

In this section, we consider the general problem of finding an orthogonal operator which decomposes an arbitrary representation, $X(\sigma)$, of a finite group G. Unlike the Clebsch-Gordan series which are basis-independent, intertwining operators must be recomputed if we change the underlying basis by which the irreducible representation matrices are constructed. However, for a fixed basis, we remind the reader that these intertwining operators need only be computed once and for all and can be stored in a table for future reference. Let X be any degree d group representation of G, and let Y be an equivalent direct sum of irreducibles, e.g.,

$$Y(\sigma) = \bigoplus_{\nu} \bigoplus_{\ell=1}^{z_\nu} \rho_\nu(\sigma), \tag{B.5}$$

where each irreducible ρ_ν has degree d_ν. We would like to compute an invertible (and orthogonal) operator C, such that $C \cdot X(\sigma) = Y(\sigma) \cdot C$, for all $\sigma \in G$. Throughout this section, we will assume that the multiplicities z_ν are known. To compute Clebsch-Gordan coefficients, for example, we would set $X = \rho_\lambda \otimes \rho_\mu$, and the multiplicities would be given by the Clebsch-Gordan series (Equation B.1). To find the matrix which relates marginal probabilities to irreducible coefficients, we would set $X = \tau_\lambda$, and the multiplicities would be given by the Kostka numbers (Equation 5.4).

We will begin by describing an algorithm for computing a basis for the space of all possible intertwining operators which we denote by:

$$\mathrm{Int}_{[X;Y]} = \{C \in \mathbb{R}^{d\times d} \ : \ C \cdot X(\sigma) = Y(\sigma) \cdot C, \ \ \forall \sigma \in G\}.$$

We will then discuss some of the theoretical properties of $\mathrm{Int}_{[X;Y]}$ and show how to efficiently select an *orthogonal* element of $\mathrm{Int}_{[X;Y]}$.

Our approach is to naively[8] view the task of finding elements of $\mathrm{Int}_{[X;Y]}$ as a similarity matrix recovery problem, with the twist that the similarity matrix must be consistent over all group elements. We first cast the problem of recovering a similarity matrix as a nullspace computation.

[8] In implementation, we use a more efficient algorithm for computing intertwining operators known as the *Eigenfunction Method* (EFM) (Chen, 1989). Unfortunately, the EFM is too complicated for us to describe in this paper. The method which we describe in this appendix is conceptually simpler than the EFM and generalizes easily to groups besides S_n.

Proposition 33. *Let A, B, C be matrices and let $K_{AB} = I \otimes A - B^T \otimes I$. Then $AC = CB$ if and only if vec$(C) \in$ Nullspace(K_{AB}).*

Proof. A well known matrix identity (van Loan, 2000) states that if A, B, C are matrices, then $\text{vec}(ABC) = \left(C^T \otimes A\right)\text{vec}(B)$. Applying the identity to $AC = CB$, we have:

$$\text{vec}(ACI) = \text{vec}(ICB),$$

and after some manipulation:

$$\left(I \otimes A - B^T \otimes I\right)\text{vec}(C) = 0,$$

showing that $\text{vec}(C) \in \text{Nullspace}(K_{AB})$. □

For each $\sigma \in G$, the nullspace of the matrix $K(\sigma)$ constructed using the above proposition as:

$$K(\sigma) = I \otimes Y(\sigma) - X(\sigma) \otimes I, \tag{B.6}$$

where I is a $d \times d$ identity matrix, corresponds to the space of matrices C_σ such that

$$C_\sigma \cdot X(\sigma) = Y(\sigma) \cdot C, \qquad \text{for all } \sigma \in G.$$

To find the space of intertwining operators which are consistent across all group elements, we need to find the intersection:

$$\bigcap_{\sigma \in G} Nullspace(K(\sigma)). \tag{B.7}$$

At first glance, it may seem that computing the intersection might require examining $n!$ nullspaces if $G = S_n$, but as luck would have it, most of the nullspaces in the intersection are extraneous, as we now show.

Definition 34. We say that a finite group G is *generated* by a set of *generators* $S = \{g_1, \ldots, g_m\}$ if every element of G can be written as a finite product of elements in S.

For example, the following three sets are all generators for S_n:

- $\{(1,2),(1,3),\ldots,(1,n)\}$,
- $\{(1,2),(2,3),(3,4),\ldots,(n-1,n)\}$, and
- $\{(1,2),(1,2,3,\ldots,n)\}$.

To ensure a consistent similarity matrix for all group elements, we use the following proposition which says that it suffices to be consistent on any set of generators of the group.

Proposition 35. *Let X and Y be representations of finite group G and suppose that G is generated by the elements $\sigma_1, \ldots, \sigma_m$. If there exists an invertible linear operator C such that $C \cdot X(\sigma_i) = Y(\sigma_i) \cdot C$ for each $i \in \{1, \ldots, m\}$, then X and Y are equivalent as representations with C as the intertwining operator.*

Proof. We just need to show that C is a similarity transform for any other element of G as well. Let π be any element of G and suppose π can be written as the following product of generators: $\pi = \prod_{i=1}^{n} \sigma_i$. It follows that:

$$\begin{aligned} C^{-1} \cdot Y(\pi) \cdot C &= C^{-1} \cdot Y\left(\prod_i \sigma_i\right) \cdot C = C^{-1} \cdot \left(\prod_i Y(\sigma_i)\right) \cdot C \\ &= (C^{-1} \cdot Y(\sigma_1) \cdot C)(C^{-1} \cdot Y(\sigma_2) \cdot C) \cdots (C^{-1} \cdot Y(\sigma_m) \cdot C) \\ &= \prod_i \left(C^{-1} \cdot Y(\sigma_i) \cdot C\right) = \prod_i X(\sigma_i) = X\left(\prod_i \sigma_i\right) = X(\pi) \end{aligned}$$

Since this holds for every $\pi \in G$, we have shown C to be an intertwining operator between the representations X and Y. □

The good news is that despite having $n!$ elements, S_n can be generated by just two elements, namely, $(1, 2)$ and $(1, 2, \ldots, n)$, and so the problem reduces to solving for the intersection of two nullspaces, $(K(1, 2) \cap K(1, 2, \ldots, n))$, which can be done using standard numerical methods. Typically, the nullspace is multidimensional, showing that, for example, the Clebsch-Gordan coefficients for $\rho_\lambda \otimes \rho_\mu$ are not unique even up to scale.

Because $\mathrm{Int}_{[X;Y]}$ contains singular operators (the zero matrix is a member of $\mathrm{Int}_{[X;Y]}$, for example), not every element of $\mathrm{Int}_{[X;Y]}$ is actually a legitimate intertwining operator as we require invertibility. In practice, however, since the singular elements correspond to a measure zero subset of $\mathrm{Int}_{[X;Y]}$, one method for reliably selecting an operator from $\mathrm{Int}_{[X;Y]}$ that "works" is to simply select a random element from the nullspace to be C. It may, however, be desirable to have an *orthogonal* matrix C which works as an intertwining operator. In the following, we discuss an object called the *Commutant Algebra* which will lead to several insights about the space $\mathrm{Int}_{[X;Y]}$, and in particular, will lead to an algorithm for 'modifying' any invertible intertwining operator C to be an *orthogonal* matrix.

Definition 36. The *Commutant Algebra* of a representation Y is defined to be the space of operators which commute with Y[9]:

$$\mathrm{Com}_Y = \{S \in \mathbb{R}^{d \times d} \ : \ S \cdot Y(\sigma) = Y(\sigma) \cdot S, \ \ \forall \sigma \in G\}.$$

The elements of the Commutant Algebra of Y can be shown to always take on a particular constrained form (shown using Schur's Lemma in (Sagan, 2001)). In particular, every element of Com_Y takes the form

$$S = \bigoplus_\nu \left(M_{z_\nu} \otimes I_{d_\nu}\right), \tag{B.8}$$

where M_{z_ν} is some $z_\nu \times z_\nu$ matrix of coefficients and I_{d_ν} is the $d_\nu \times d_\nu$ identity (recall that the z_ν are the multiplicities from Equation B.5). Moreover, it can

[9]Notice that the definition of the Commutant Algebra does not involve the representation X.

be shown that every matrix of this form must necessarily be an element of the Commutant Algebra.

The link between Com_Y and our problem is that the space of intertwining operators can be thought of as a 'translate' of the Commutant Algebra.

Lemma 37. *There exists a vector space isomorphism between $Int_{[X;Y]}$ and Com_Y.*

Proof. Let R be any invertible element of $\mathrm{Int}_{[X;Y]}$ and define the linear map $f : \mathrm{Com}_Y \to \mathbb{R}^{d\times d}$ by: $f : S \mapsto (S \cdot R)$. We will show that the image of f is exactly the space of intertwining operators. Consider any element $\sigma \in G$:

$$\begin{aligned}(S \cdot R) \cdot X(\sigma) \cdot (S \cdot R)^{-1} &= S \cdot R \cdot X(\sigma) \cdot R^{-1} \cdot S^{-1},\\ &= S \cdot Y(\sigma) \cdot S^{-1} \quad (\text{since } R \in \mathrm{Int}_{[X;Y]}),\\ &= Y(\sigma) \quad (\text{since } S \in \mathrm{Com}_Y).\end{aligned}$$

We have shown that $S \cdot R \in \mathrm{Int}_{[X;Y]}$, and since f is linear and invertible, we have that $\mathrm{Int}_{[X;Y]}$ and Com_Y are isomorphic as vector spaces. □

Using the lemma, we can see that the dimension of $\mathrm{Int}_{[X;Y]}$ must be the same as the dimension of Com_Y, and therefore we have the following expression for the dimension of $\mathrm{Int}_{[X;Y]}$.

Proposition 38.

$$\dim Int_{[X;Y]} = \sum_\nu z_\nu^2.$$

Proof. To compute the dimension of $\mathrm{Int}_{[X;Y]}$, we need to compute the dimension of Com_Y, which can be accomplished simply by computing the number of free parameters in Equation B.8. Each matrix M_{z_ν} is free and yields z_ν^2 parameters, and summing across all irreducibles ν yields the desired dimension. □

To select an orthogonal intertwining operator, we will assume that we are given some invertible $R \in \mathrm{Int}_{[X;Y]}$ which is not necessarily orthogonal (such as a random element of the nullspace of K (Equation B.6)). To find an orthogonal element, we will 'modify' R to be an orthogonal matrix by applying an appropriate rotation, such that $R \cdot R^T = I$. We begin with a simple observation about $R \cdot R^T$.

Lemma 39. *If both X and Y are orthogonal representations and R is an invertible member of $Int_{[X;Y]}$, then the matrix $R \cdot R^T$ is an element of Com_Y.*

Proof. Consider a fixed $\sigma \in G$. Since $R \in \mathrm{Int}_{[X;Y]}$, we have that:

$$X(\sigma) = R^{-1} \cdot Y(\sigma) \cdot R.$$

It is also true that:

$$X(\sigma^{-1}) = R^{-1} \cdot Y(\sigma^{-1}) \cdot R. \tag{B.9}$$

Algorithm 5: Pseudocode for computing an orthogonal intertwining operators

IntXY

input : A degree d orthogonal matrix representation X evaluated at permutations $(1,2)$ and $(1,\ldots,n)$, and the multiplicity z_ν, of the irreducible ρ_ν in X

output: A matrix C_ν with orthogonal rows such that $C_\nu^T \cdot \oplus^{z_\nu}\rho_\nu \cdot C_\nu = X$

$K_1 \leftarrow I_{d\times d} \otimes (\oplus^{z_\nu}\rho_\nu(1,2)) - X(1,2) \otimes I_{d\times d}$;

$K_2 \leftarrow I_{d\times d} \otimes (\oplus^{z_\nu}\rho_\nu(1,\ldots,n)) - X(1,\ldots,n) \otimes I_{d\times d}$;

$K \leftarrow [K_1; K_2]$; //*Stack* K_1 *and* K_2

$v \leftarrow \text{SparseNullspace}\big(K, z_\nu^2\big)$; //*Find the* d_ν^2*-dimensional nullspace*

$R \leftarrow \text{Reshape}(v; z_\nu d_\nu, d)$; //*Reshape* v *into a* $(z_\nu d_\nu) \times d$ *matrix*

$M \leftarrow \text{KroneckerFactors}(R \cdot R^T)$; //*Find* M *such that* $R \cdot R^T = M \otimes I_{d_\nu}$

$S_\nu \leftarrow \text{Eigenvectors}(M)$;

$C_\nu \leftarrow S_\nu^T \cdot R$;

$\text{NormalizeRows}(C_\nu)$;

Since $X(\sigma)$ and $Y(\sigma)$ are orthogonal matrices by assumption, Equation B.9 becomes:

$$X^T(\sigma) = R^{-1} \cdot Y^T(\sigma) \cdot R.$$

Taking transposes,

$$X(\sigma) = R^T \cdot Y(\sigma) \cdot (R^{-1})^T.$$

We now multiply both sides on the left by R, and on the right by R^T,

$$\begin{aligned} R \cdot X(\sigma) \cdot R^T &= R \cdot R^T \cdot Y(\sigma) \cdot (R^{-1})^T \cdot R^T \\ &= R \cdot R^T \cdot Y(\sigma). \end{aligned}$$

Since $R \in \text{Int}_{[X;Y]}$,

$$Y(\sigma) \cdot R \cdot R^T = R \cdot R^T \cdot Y(\sigma),$$

which shows that $R \cdot R^T \in \text{Com}_Y$. □

We can now state and prove our orthogonalization procedure, which works by diagonalizing the matrix $R \cdot R^T$. Due to its highly constrained form, the procedure is quite efficient.

Theorem 40. *Let X be any orthogonal group representation of G and Y an equivalent orthogonal irreducible decomposition (As in Equation B.5). Then for any invertible element $R \in Int_{[X;Y]}$, there exists an (efficiently computable) orthogonal matrix T such that the matrix $T \cdot R$ is an element of $Int_{[X;Y]}$ and is* orthogonal.

Proof. Lemma 39 and Equation B.8 together imply that the matrix $R \cdot R^T$ can always be written in the form

$$R \cdot R^T = \oplus_\nu \left(M_{z_\nu} \otimes I_{d_\nu} \right)$$

Since $R \cdot R^T$ is symmetric, each of the matrices M_{z_ν} is also symmetric and must therefore possess an orthogonal basis of eigenvectors. Define the matrix S_{z_ν} to be the matrix whose columns are the eigenvectors of M_{z_ν}.

The matrix $S = \oplus_\nu (S_{z_\nu} \otimes I_{d_\nu})$ has the following two properties:

1. $(S^T \cdot R)(S^T \cdot R)^T$ is a diagonal matrix:

   Each column of S is an eigenvector of $R \cdot R^T$ by standard properties of the direct sum and Kronecker product. Since each of the matrices, S_{z_ν}, is orthogonal, the matrix S is also orthogonal. We have:

   $$\begin{aligned} (S^T \cdot R)(S^T \cdot R)^T &= S^T \cdot R \cdot R^T \cdot S, \\ &= S^{-1} \cdot R \cdot R^T \cdot S, \\ &= D, \end{aligned}$$

   where D is a diagonal matrix of eigenvalues of $R \cdot R^T$.

2. $S^T \cdot R \in \mathrm{Int}_{[X;Y]}$:

   By Equation B.8, a matrix is an element of Com_Y if and only if it takes the form $\oplus_\nu (S_{z_\nu} \otimes I_{d_\nu})$. Since S can be written in the required form, so can S^T. We see that $S^T \in \mathrm{Com}_Y$, and by the proof of Lemma 37, we see that $S^T \cdot R \in \mathrm{Int}_{[X;Y]}$.

Finally, setting $T = D^{1/2} \cdot S^T$ makes the matrix $T \cdot R$ orthogonal (and does not change the fact that $T \cdot R \in \mathrm{Int}_{[X;Y]}$). ☐

We see that the complexity of computing T of is dominated by the eigenspace decomposition of M_{z_ν}, which is $O\left(z_\nu^3\right)$. Pseudocode for computing orthogonal intertwining operators is given Algorithm 5.

MACHINE LEARNING
DEPARTMENT
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213

Carnegie Mellon.

Carnegie Mellon University does not discriminate and Carnegie Mellon University is required not to discriminate in admission, employment, or administration of its programs or activities on the basis of race, color, national origin, sex or handicap in violation of Title VI of the Civil Rights Act of 1964, Title IX of the Educational Amendments of 1972 and Section 504 of the Rehabilitation Act of 1973 or other federal, state, or local laws or executive orders.

In addition, Carnegie Mellon University does not discriminate in admission, employment or administration of its programs on the basis of religion, creed, ancestry, belief, age, veteran status, sexual orientation or in violation of federal, state, or local laws or executive orders. However, in the judgment of the Carnegie Mellon Human Relations Commission, the Department of Defense policy of, "Don't ask, don't tell, don't pursue," excludes openly gay, lesbian and bisexual students from receiving ROTC scholarships or serving in the military. Nevertheless, all ROTC classes at Carnegie Mellon University are available to all students.

Inquiries concerning application of these statements should be directed to the Provost, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh PA 15213, telephone (412) 268-6684 or the Vice President for Enrollment, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh PA 15213, telephone (412) 268-2056

Obtain general information about Carnegie Mellon University by calling (412) 268-2000