Thesis Proposal
Probabilistic Reasoning with Permutations:
A Fourier-Theoretic Approach

Jonathan Huang

November 15, 2008

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:
Carlos Guestrin, Chair
John Lafferty
Drew Bagnell
Leonidas Guibas, Stanford
Alex Smola, Yahoo! Research

*Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy.*

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 common factorized probability distribution representations, such as graphical models, are inefficient due to the mutual exclusivity constraints that are typically associated with permutations.

This thesis explores a new approach for probabilistic reasoning with permutations based on the idea of approximating distributions using their low-frequency Fourier components. We use a generalized Fourier transform defined for functions on permutations, but unlike the widely used Fourier analysis on the circle or the real line, Fourier transforms of functions on permutations take the form of ordered collections of matrices. As we show, maintaining the appropriate set of low-frequency Fourier terms corresponds to maintaining matrices of simple marginal probabilities which summarize the underlying distribution. We show how to derive the Fourier coefficients of a variety of probabilistic models which arise in practice and that many useful models are either well-approximated or exactly represented by low-frequency (and in many cases, sparse) Fourier coefficient matrices.

In addition to showing that Fourier representations are both compact and intuitive, we show how to cast common probabilistic inference operations in the Fourier domain, including marginalization, conditioning on evidence, and factoring based on probabilistic independence. The algorithms presented in this thesis are fully general and work gracefully in *bandlimited settings* where only a partial subset of Fourier coefficients is made available.

From the theoretical side, we tackle several problems in understanding the consequences of the bandlimiting approximation. We present results in this thesis which illuminate the nature of error propagation in the Fourier domain and propose methods for mitigating their effects.

Finally we demonstrate the effectiveness of our approach on several real datasets and show that our methods, in addition to being well-founded theoretically, are also scalable and provide superior results in practice.

Contents

Chapter 1

Introduction

Probability distributions over permutations arise in a diverse variety of problems involving matchings, rankings, and assignment. While they were perhaps first studied in the context of gambling and card games, the problem of reasoning with uncertainty over permutations makes an appearance in many real problems in computing. Probabilistic reasoning problems over permutations, however, are not amenable to the typical representations afforded by machine learning such as Bayesian networks and Markov random fields. This thesis explores alternative representations and inference algorithms for dealing with permutations. Before discussing our approach in more detail, we summarize some application domains where permutations play a crucial role.

Card shuffling. Historically, distributions over permutations were perhaps first studied in the context of card games and gambling. In the context of cards, a permutation can be thought of as a configuration of a deck of cards (“King of Hearts on top, 3 of Spades second, etc.”) and uncertainty emerges due to partial observability and some number of shuffles that typically precede a card trick or game. Some questions that one might want to answer are: “What suite does the top card in the deck most likely belong to?”, or “What is the probability that my opponent has a full house?”. Another typical question is, “How many shuffles are required to sufficiently randomize a deck?”. Bayer and Diaconis [1992] analyzed the commonly used *riffle shuffle* (see Figure 1.2(a)) and remarkably found that seven typical shuffles are sufficient to bring the distribution over configurations of a standard 52 card deck to be close to uniform (and that further shuffles do not help very much). The card shuffling problem has been tackled in a number of papers over the last two decades. See, for example, [Aldous and Diaconis, 1986, Bayer and Diaconis, 1992, Diaconis, 1988].

Probabilistic Matching. Matching problems have been studied by both mathematicians and computer scientists in a variety of settings. In computer vision, they are studied under the umbrella of ‘correspondence problems’ which arise, for example, when one matches feature points between two images of the same scene or between the vertices of meshes (See Figures 1.1(a), 1.1(b),1.1(c) for an example).

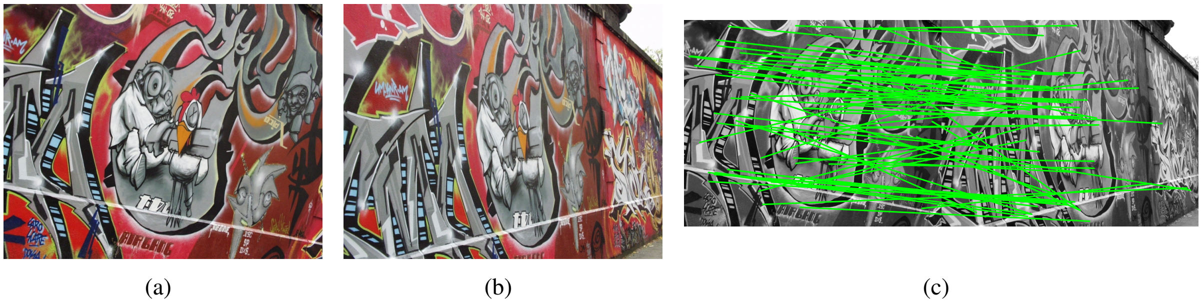
In the simplest case of bipartite matching, there is a weight associated with each edge between two disjoint sets of vertices A and B, and the objective is to find the highest weight matching between vertices from A and B. Bipartite matching can be solved in polynomial time due to the fact that its constraints can be written as a totally unimodular matrix. For many matching problems that arise in vision, however, simple bipartite matching is too simple and more complicated constraints are required such as those allowed for by graph and hypergraph generalizations of the bipartite matching problem. (Hyper)graph generalizations are typically NP-hard, but there are many approaches that have been shown to work successfully in practice [Cour et al., 2007, Leordeanu and Hebert, 2005].

Previous work in matching has largely focused on finding a single permutation which maximizes some objective function, but very few authors (see [Helmbold and Warmuth, 2007, Zass and Shashua, 2008]) have tackled the probabilistic matching problem, where one asks for a probability for each possible permutation. Probabilistic matchings can be useful in problems where the task is to find a sequence of matchings and one must be robust to matching errors made in the past.

Ranking. Distributions over rankings (and votes) arise in a multitude of information retrieval tasks. Based on user information, prior site visits, mouseclicks, and other possible information, one would like to produce a ranking of websites, preferred films or books for a given search query. Distributions over rankings arise due to variations in preferences among different people. For example, some people prefer to rent romantic comedies, while others enjoy action movies. Such population effects are not captured by a single ranking but rather, an entire probability distribution over rankings. Distributions can also arise due to uncertainty over the preference relations of a particular person. For example, ranking data is more often than not only available in partial form, where users provide *partially* specified rankings (“My five favorite movies in no particular order are. . . ”). Data are sometimes available in the form of *ratings* rather than direct rankings and it often makes sense to convert them to rankings before data analysis since ratings across people are often incompatible (“My perfect 10 might not be your perfect 10”).

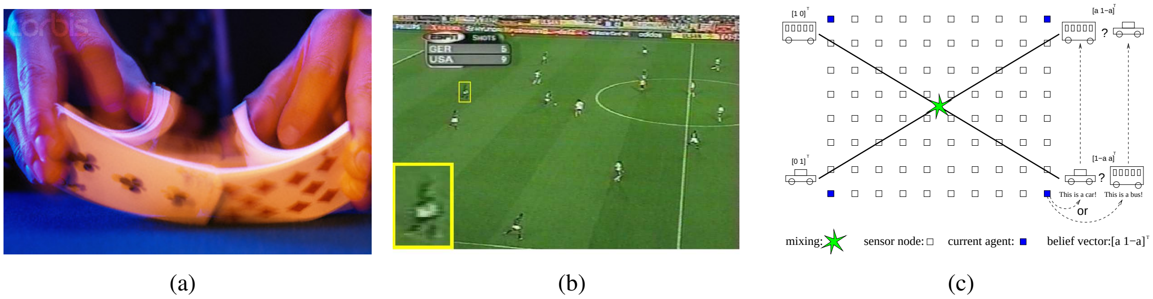
Ranking problems have been studied by many statisticians over the past decades including: [Critchlow, 1985, Daugherty et al., 2007, Diaconis, 1989, Fligner and Verducci, 1986, Lebanon and Lafferty, 2002, Lebanon and Mao, 2007, Mallows, 1957, Taylor et al., 2008]

Identity Management. Consider the problem of tracking n objects (vehicles or people, for example) 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 objects are well separated, the problem is easily decomposed and measurements about each individual object can be clearly associated with a particular track. When objects pass near each other, however, confusion can arise as their signal signatures may mix; see Figure 1.2(c). After the individual objects 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 object, 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*. Identity management and the closely related problem of date association have been addressed in a number of previous works, including: [Balakrishnan et al., 2004, Bar-Shalom and Fortmann, 1988, Collins and Uhlmann, 1992, Cox and Hingorani, 1994, Guibas, 2008, Murty, 1968, Oh and Sastry, 2005, Oh et al., 2004, Poore, 1995, Reid, 1979, Schumitsch et al., 2006a,b, Shin et al., 2005].



(a) (b) (c)

Figure 1.1: Feature point matching example: (a) and (b) are images of the same scene taken from two similar viewpoints. (c) shows seventy detected *SIFT* feature points [Lowe, 2004] in each image and a possible match (correspondence) between the two sets of points.



(a) (b) (c)

Figure 1.2: Card shuffling: (a) demonstrates the standard "riffle-shuffle" on a deck of cards; Identity management examples: (b) illustrates a multiperson tracking scenario (soccer game) where identity information is weak due to low-resolution imagery (image from Efros et al. [2003]), and (c) (image from Guibas [2008]) illustrates a "mixing" event in which two tracks swap identity with some probability.

1.1 Related work

Exact probabilistic inference on permutations is difficult because it requires us to address the fundamental combinatorial challenge that there are $n!$ permutations. 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. While maintaining a small subset of permutations can be effective for sharply concentrated distributions, it is inappropriate in situations with more uncertainty. The Fourier based approach presented in this thesis seeks instead to approximate probability distributions by linear combinations of low-frequency Fourier basis functions, allowing for smooth compact approximations. Unlike the familiar signal processing theory however, it is assumed that holding a signal over all permutations is *always* infeasible, and so many of our contributions deal with formulating inference operations that work with Fourier coefficients.

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 Schumitsch et al. [2006a,b], Shin et al. [2005] 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. For further related work, see: [Clausen and Baum, 1993, Diaconis, 1989, Kondor, 2007, 2008, Kondor and Borgwardt, 2008, Kondor and Lafferty, 2002, Malm, 2005, Maslen, 1998, Shin et al., 2003, Terras, 1999].

1.2 Proposal

In this thesis, we seek compact representations and efficient inference schemes for probability distributions over permutations. This work will represent an in-depth exploration of the idea of compactly summarizing distributions over permutations with truncated Fourier transforms. Along with introducing general and scalable algorithms for performing probabilistic inference with Fourier coefficients, we will present theory for understanding the errors that arise from bandlimiting, and demonstrate our methods on a variety of applications. For relevant publications, see Huang et al. [2007, 2008].

Chapter 2

Distributions on the symmetric group

We begin by introducing several important concepts about permutations. We first discuss basic properties of the symmetric group and then turn to probability distributions over the symmetric group. W introduce a family of marginal-based statistics that can be used to compactly summarize distributions over the symmetric group which would otherwise be too large to store in memory. No familiarity with group theory will be assumed in this proposal — for background, see Diaconis [1988], Lang [1965].

2.1 Permutations

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. We will use ϵ to denote the *identity* element which maps everything to itself (thus in one-line notation, $\epsilon = [1 \;\; 2 \;\; \ldots \;\; n]$). And to each permutation σ, there exists a unique *inverse* σ^{-1} for which $\sigma^{-1}\sigma = \sigma\sigma^{-1} = \epsilon$. For example, the inverse of the permutation $\sigma = [1 \;\; 4 \;\; 5 \;\; 3 \;\; 2]$ is $\sigma^{-1} = [1 \;\; 5 \;\; 4 \;\; 2 \;\; 3]$. The set of all $n!$ permutations is called the *symmetric group*, or just S_n.

We will sometimes notate the elements of S_n using the more standard (and compact) *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, $\sigma = (1, 2, 3)(4)(5) = (1, 2, 3)$. Finally, we will refer to two-cycles (which take the form (i, j)) as *transpositions* or *swaps*).

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. When it comes to probabilistic reasoning, the mutual exclusivity constraints are at first blush, a curse, rendering typical representations such as graphical models ineffective, due to the fact that all of the $\sigma(i)$ are coupled in the joint distribution. Instead of exploiting conditional-independence structure, our Fourier based approximations achieve compactness by exploiting an alternative *algebraic structure*.

2.2 Compact summaries

Since there are $n!$ permutations, it is infeasible for all but very small n to consider storing full distributions over S_n. In this section, we consider a few ideas for compactly summarizing distributions by their marginal probabilities. Perhaps surprisingly, these deceptively simple summary statistics form the basis for the powerful machinery of Fourier analysis which we will leverage in later chapters.

The first-order summary. 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 (thus resulting in a 'mean' which might not itself be a permutation matrix).

Example 1. *For example, consider the two permutations $\epsilon, (1,2) \in S_3$. 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 discuss in more detail, computing the 'mean' (as described above) of a distribution over permutations, P, compactly summarizes P by storing a marginal distribution over each of $\sigma(1), \sigma(2), \ldots,$ $\sigma(n)$, 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 [Helmbold and Warmuth, 2007, Shin et al., 2003]. In identity management [Shin et al., 2003][1], 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 statements like:

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

Higher order summaries. One might consider maintaining summaries at a ‘finer resolution’ by keeping a higher order matrix of marginals. For example, the second-order matrix of marginals has its rows and columns indexed by ordered pairs of identities and its $(ij, k\ell)$-entry is the *second-order marginal probability* that $\sigma(k) = i$ *and* $\sigma(\ell) = j$. Instead of storing only $O(n^2)$, the second-order summary requires that we store $O(n^4)$ probabilities.

We can go further and define third-order summaries, fourth-order summaries, and so on,[2] achieving finer and finer resolution at the cost of storing more and more numbers. As we will see, however, there is more to maintaining a matrix of marginals than meets the eye — and in particular, there are a number of constraints that must be enforced for a matrix to correspond to some true distribution. For example, any matrix of marginals is necessarily doubly-stochastic. However, there are in general, many more constraints that must be enforced.

2.3 Two themes: *Representation* and *Inference*

The two running themes throughout the remainder of this thesis deal with the related problems of *representation* and *inference*. In the next chapter (Chapter 3), we tackle the problem of *representing* a distribution on permutations. We will discuss how 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, we can capture the same information stored at higher order marginals in a principled and redundancy-free fashion. Much like the story with graphical models, compact representation does not necessarily imply efficient inference, and in chapters 4 and 5, we tackle the *inference* problem and show that the Fourier-theoretic framework provides a natural framework for formulating most common inference operations with respect to our compact summaries.

[1] 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.

[2] We will also use the term *zeroth-order* marginal to denote the normalization constant, $\sum_\sigma P(\sigma) = 1$.

Chapter 3

Representation: Fourier analysis on S_n

The Fourier series decomposition allows one to write any function $f : [0,1] \to \mathbb{C}$ as a linear combination of trigonometric basis functions [Marsden and Hoffman, 1993]:

$$f(x) = \sum_{m=-\infty}^{\infty} \alpha_m e^{i2\pi mx},$$

where $\alpha_m \in \mathbb{C}$ for each m. The squared magnitude of each α_m ($|\alpha_m|^2$) measures the "energy" of f contained at frequency m — for example, if $f(x)$ is purely sinusoidal with one frequency, then only one of the α_m is nonzero. The collection of trigonometric functions $e^{i2\pi m}$ forms a *complete orthogonal basis* for the space of functions on the interval $[0,1]$; thus any collection of α_m's corresponds to some function f and moreover, f is uniquely determined. Since the collection of frequency responses, $\{\alpha_m\}_{m=-\infty}^{\infty}$ can themselves be thought of as a function in "frequency space", we call the mapping between the functions $(f \mapsto \{\alpha_m\})$ the *Fourier transform.*

Fourier analytic methods are now widely employed in almost all science and engineering disciplines due in part to the development of efficient algorithms for computing Fourier transforms such as the FFT [Cooley and Tukey, 1965]. We will be using Fourier methods for "compressing" signals by dropping high frequency terms resulting in a smoother approximation to the original signal:

$$f(x) \approx \sum_{m=-B}^{B} \alpha_m e^{i2\pi mx},$$

in an operation known as *bandlimiting*.

In this chapter, we provide a gentle introduction to a generalization of the Fourier transform for functions defined on the symmetric group. It will turn out to be the case that the Fourier transform of a function $f : S_n \to \mathbb{R}$ is intimately connected with a class of marginal-type queries that generalize the previously introduced first and second-order queries, and we will leverage the Fourier theory later in Chapter 4 to perform efficient approximate inference operations. Instead of providing a rigorous introduction to Fourier analysis on S_n, however, we will simply highlight some of the key ideas and motivate the intuition that low-order marginals are somehow closely related to low-frequency Fourier coefficients. See our paper Huang et al. [2008], and Diaconis [1988], Kondor [2006], Sagan [2001], Terras [1999] for more details.

3.1 Partitions and marginals

We begin our introduction to Fourier analysis, not with a definition of the Fourier transform, but with a discussion of frequency. While frequencies are indexed by real numbers on the real line (e.g., $20Hz$, $30Hz$), we will now motivate the idea that the appropriate frequency analog for S_n is indexed by *partitions* of n. We define a partition of n to be an unordered tuple of positive integers $\lambda = (\lambda_1, \ldots, \lambda_\ell)$, summing to n. For example, $\lambda = (3, 2)$ is a partition of $n = 5$ since $3 + 2 = 5$. Partitions are usually written as weakly decreasing sequences by convention, so some partitions of $n = 5$ are: (5), $(4, 1)$, $(3, 2)$, $(3, 1, 1)$, etc...

We will identify each type of marginal with some unique *partition* of n by associating the s^{th}-order marginal probabilities with the partition $\lambda = (n - s, 1, \ldots, 1)$, where there are s trailing 1's. Thus $\lambda = (n - 1, 1)$ refers to the first-order marginals, while $(n - 2, 1, 1)$ refers to the second-order marginals, and so on. We will say that the s^{th}-order marginals are "of type λ". General partitions (not of the form $\lambda = (n - s, 1, \ldots, 1)$) will in fact also be associated with marginal type queries. For example, we will use the partition $\lambda = (n - 2, 2)$ to refer to marginals of unordered pairs: $P(\sigma : \sigma(\{k, \ell\}) \to \{i, j\})$ (e.g., the probability that Alice and Bob occupy tracks 1 and 2, in either order), and $\lambda = (n - 3, 3)$ refers to marginals of unordered triples. See Huang et al. [2008] for more details about the semantics of partitions. Given a distribution f, we will refer to the pertinent matrix of marginals of type λ as $\tilde{f}_\lambda$ and denote the dimension of $\tilde{f}_\lambda$ by D_λ. Thus the first-order marginals of f will be denoted by $\tilde{f}_{(n-1,1)}$ and $D_{(n-1,1)} = n$.

Example 2. *As an example, we define the following probability distribution on S_4:*

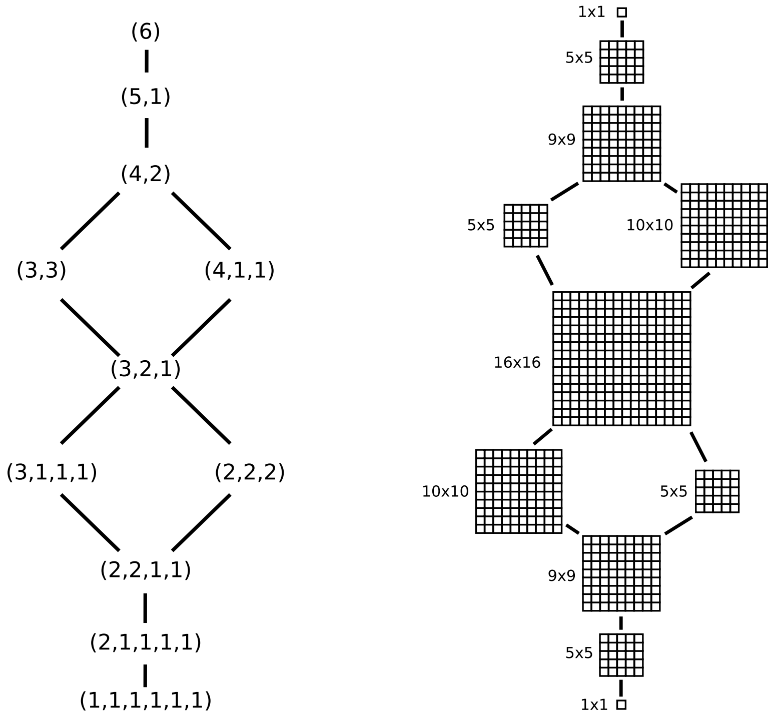
$$f(\sigma) = \begin{cases} 1/3 & \textit{if } \sigma = [1\,2\;3\,4],\ \sigma = [1\,2\,4\,3],\ \textit{or } \sigma = [1\,3\,2\,4], \\ 0 & \qquad\qquad\qquad \textit{otherwise} \end{cases} ,$$

and compute the matrix of marginals of type $\lambda = (n - 2, 2) = (2, 2)$. Note that $D_\lambda = 6$ in this case since there are six distinct unordered pairs from $\{1, 2, 3, 4\}$.

$$\tilde{f}_\lambda = \left[\begin{array}{c|cccccc} & \{1,2\} & \{1,3\} & \{1,4\} & \{2,3\} & \{2,4\} & \{3,4\} \\ \hline \{1,2\} & 2/3 & 1/3 & 0 & 0 & 0 & 0 \\ \{1,3\} & 1/3 & 1/3 & 1/3 & 0 & 0 & 0 \\ \{1,4\} & 0 & 1/3 & 2/3 & 0 & 0 & 0 \\ \{2,3\} & 0 & 0 & 0 & 2/3 & 1/3 & 0 \\ \{2,4\} & 0 & 0 & 0 & 1/3 & 1/3 & 1/3 \\ \{3,4\} & 0 & 0 & 0 & 0 & 1/3 & 2/3 \end{array} \right].$$

As we discussed earlier, the simplest marginals (the zeroth order normalization constant), correspond to the partition $\lambda = (n)$, while the first-order marginals correspond to $\lambda = (n - 1, 1)$, and the second-order unordered marginals correspond to $\lambda = (n - 2, 2)$. These 'simple' marginals intuitively carry low-frequency information, but as the list of partitions goes on, the marginals get far more complicated, and at the other end of the spectrum, we have marginals corresponding to $\lambda = (1, 1, \ldots, 1)$ which exactly recover the original probabilities $P(\sigma)$ and therefore must contain information at all frequencies.

Dominance order. If partitions correspond to frequency, it is natural to ask how the partitions of n might be ordered with respect to the 'complexity' of the corresponding basis functions. We now answer the question of how one might 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 *partial* order.



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

(b) Fourier coefficient matrices for S_6.

Figure 3.1: 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.

Definition 3 (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 ordering since $3 \leq 4$, but $3+3 \geq 4+1$. The ordering over the partitions of $n = 6$ is depicted in Figure 3.1(a).

Partitions with fewer parts tend to be greater (with respect to dominance ordering) than those with many parts. Intuitively, marginals corresponding to partitions which are *high* in the dominance ordering are 'low frequency', while marginals corresponding to partitions which are *low* in the dominance ordering are 'high frequency'[1]. Thus we think of marginals of type $(3,2,1)$ as being higher frequency than those of type $(5,1)$, for example.

3.2 Matrix-valued basis functions.

While we have motivated the marginal probabilities as being closely related to Fourier coefficients, they are not themselves Fourier coefficients due to the dependences (redundancies) that arise among the matri-

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

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 3.1: Dimensions of low-order Fourier coefficient matrices.

ces of marginals. Given a distribution f, we list a few of the possible dependencies that can occur in $\tilde{f}$ below.

- For every matrix of marginals, $\tilde{f}_\lambda$, we have a *double stochasticity* constraint which says that for all j, we have $\sum_i [\tilde{f}_\lambda]_{ij} = 1$ and for all i, we have $\sum_j [\tilde{f}_\lambda]_{ij} = 1$ (every j must map to some i and every j must be mapped to by some i with probability 1).
- There are certain symmetry constraints for various types of marginals. For example, at the second-order marginals, we know that the following two probabilities must be equal:

  $$P(\sigma : \sigma(Alice, Bob) = (Track\,1, Track\,2)),$$
  $$P(\sigma : \sigma(Bob, Alice) = (Track\,2, Track\,1)).$$

  As another example, if $n = 4$ and we are considering second-order (unordered) marginals, then the following two probabilities must also be equal:

  $$P(\sigma : \sigma(\{Alice, Bob\}) = \{Track\,1, Track\,2\}),$$
  $$P(\sigma : \sigma(\{Cathy, David\}) = \{Track\,3, Track\,4\}).$$

Instead of mapping a partition to an overcomplete matrix of marginals, the Fourier transform on S_n assigns to each partition λ, a smaller matrix (which we denote by $\hat{f}_\lambda$) with dimension $d_\lambda \leq D_\lambda$, which satisfies the following property, which we will state in the form of a theorem.

Theorem 4 (James Submodule Theorem)**.** *There exists a matrix C_λ and nonnegative integers $\kappa_{\lambda\mu}$ for each partition μ of n, such that:*

$$\tilde{f}_\lambda = C_\lambda^T \cdot \left(\bigoplus_\mu \bigoplus_{\ell=1}^{\kappa_{\lambda\mu}} \hat{f}_\mu \right) \cdot C_\lambda, \tag{3.2.1}$$

where the numbers $\kappa_{\lambda\mu}$ (called Kostka numbers*) are nonzero only if $\mu \trianglerighteq \lambda$.*

In plain English, the James Submodule theorem states that the marginals of type λ are always reconstructible from the Fourier coefficients at partitions above λ in the dominance ordering. There are two main points that differentiate the Fourier transform on S_n from the common Fourier transform on the reals.

1. Instead of being a function of frequency, the Fourier transform of a function defined over the symmetric group is a function of partitions, which as we discussed, can be partially ordered with respect to the dominance relation.
2. The Fourier transform of a function on the symmetric group is *matrix-valued* (rather than scalar-valued). While we will refrain from explicitly defining the underlying basis functions in this proposal, the fact that the Fourier coefficients are matrix-valued implies that the Fourier basis functions themselves are matrix-valued functions. We provide a list of expressions for the dimensions d_λ for low order partitions in Table 3.1 and a diagram of the Fourier coefficient matrices of S_6 in Figure 3.1(b).

As in the familiar incarnation of the theory, the Fourier transform of a function defined over a group satisfies several properties which can be exploited in computation. We list two useful properties below (see Diaconis [1988], for example).

Proposition 5. *The Fourier transform obeys the following properties:*

1. *(Linearity) Let $f, g : G \to \mathbb{R}$ be any two functions on a group G. If $h = \alpha f + \beta g$ for scalars α and β, then at any partition λ, we have $\hat{h}_\lambda = \alpha \hat{f}_\lambda + \beta \hat{g}_\lambda$.*
2. *(Shifting) If two functions f and g are related by a shift (i.e. $g(\sigma) = f(\pi\sigma\tau)$), for constant permutations $\pi, \tau \in S_n$), then: $\hat{g}_\lambda = \hat{\delta}_\pi \cdot \hat{f}_\lambda \cdot \hat{\delta}_\tau$ holds at all partitions λ, where δ_π is the indicator function of the permutation π.*

3.3 Fourier transforms of useful functions

Given a distribution f over the symmetric group, computing the Fourier coefficients $\hat{f}_\lambda$ at all partitions λ is intractable in the most general case. However, for indicator functions of certain subgroups (or associated cosets), it is possible to efficiently compute low-order coefficient matrices. The purpose of the following examples is to, first, exhibit functions that are "naturally bandlimited" in a sense and second, to define function "primitives" which can be combined (Proposition 5) to construct a variety of useful probabilistic models in the Fourier domain (see Section 4.4). For the sake of brevity, we will not provide explicit construction algorithms here.

Indicator functions of $S_k \subset S_n$. We identify S_k isomorphically with the subgroup of elements in S_n which fix the last $n - k$ elements:

$$S_k \equiv \{\sigma \in S_n \ : \ \sigma(i) = i, \text{ for } i = k+1, \ldots, n\}.$$

For example, as a subgroup of S_4, S_2 contains just the two elements $S_2 = \{[1\,2\,3\,4], [2\,1\,3\,4]\} \subset S_4$. The indicator function δ_{S_k} takes a particularly simple (and low-rank) form and can be efficiently computed.

Proposition 6. *The Fourier transform of the subgroup indicator function δ_{S_k} is a diagonal matrix at every partition λ and is nonzero only at partitions λ such that $\lambda \trianglerighteq (k, 1, \ldots, 1)$.*

Indicator functions of $S_k \times S_{n-k} \subset S_n$. We identify the direct product $S_k \times S_{n-k}$ isomorphically with the subgroup of elements in S_n which map elements of $\{1, \ldots, k\}$ into the set $\{1, \ldots, k\}$:

$$S_k \times S_{n-k} \equiv \{\sigma \in S_n \ : \ \sigma(1, \ldots, k) = \{1, \ldots, k\}\}.$$

As in the case of δ_{S_k}, the indicator $\delta_{S_k \times S_{n-k}}$ is sparse, low-rank, and efficiently computed. While actually quantifying the sparsity of $\hat{\delta}_{S_k \times S_{n-k}}$ is beyond the scope of this document, we can state a few basic properties.

Proposition 7. *The Fourier transform of the subgroup indicator function $\delta_{S_k \times S_{n-k}}$ is a symmetric matrix at every partition, and is nonzero at (and only at) partitions of the form $\lambda = (n - s, s)$ where $s \leq k$.*

3.4 Discussion

In this chapter, we showed that partitions are the appropriate analog of frequency in the Fourier domain for permutations and that the Fourier coefficients corresponding to partitions of n can be ordered from least complex to most complex using a relation known as the dominance ordering. Finally, we discussed a principled approximation method known as bandlimiting which works by saving only the low-frequency terms of the Fourier transform of a function, and is equivalent to maintaining a set of low-order marginal probabilities.

Chapter 4

Inference: Fourier domain formulations

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. In this chapter, we discuss two common inference operations and present efficient solutions which operate completely in the Fourier domain.

4.1 Filtering over permutations

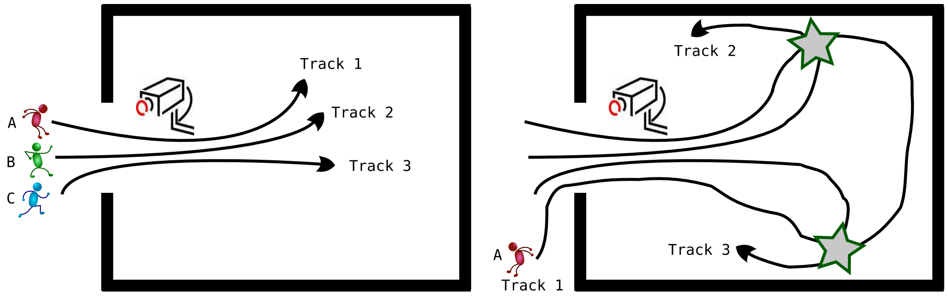
As a prelude to the general problem, we begin with a simple identity management problem on three tracks (illustrated in Figure 4.1) which we will use as a running example. In this problem, we observe a stream of localization data from three people walking inside a room. Except for a camera positioned 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 4.1. 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 4.1: 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



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

Figure 4.1: 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.

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

4.2 Inference operations in 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 Malm [2005], 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. Our approach is unlike previous approaches [Felzenszwalb et al., 2004, Willsky, 1978] which use FFTs for efficient convolution during inference, but do not work primarily in the Fourier domain. In the rest of the chapter, we will consider the prediction/rollup and conditioning operations assuming that the Fourier coefficients of the transition and observation models are known.

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 $\sigma^{(t+1)} = \pi^{(t)}\sigma^{(t)}$ [1]. In our identity management example, $\pi^{(t)}$ represents a random identity permutation that might occur among tracks when they get close to each 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 8. Let Q and P be probability distributions on S_n. Define the *convolution*[2] of Q and P to be the function $[Q * P](\sigma_1) = \sum_{\sigma_2} Q(\sigma_1\sigma_2^{-1})P(\sigma_2)$.

On the real line, it is a well-known fact that, colloquially put, "convolution in the time domain is multiplication in the frequency domain". 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 a similar *convolution theorem* (see also Diaconis [1988]).

Proposition 9 (Convolution Theorem)**.** *Let Q and P be probability distributions on S_n. For any partition λ,*

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

where the operation on the right side is matrix multiplication.

It can be shown (see Huang et al. [2008]) that under the random walk assumption, the prediction/rollup step can be written as a convolution of the transition model $Q^{(t)}(\tau)$ and the distribution at time t, $P(\sigma^{(t)})$. Therefore, assuming that the Fourier transforms $\widehat{P}_\rho^{(t)}$ and $\widehat{Q}_\rho^{(t)}$ are given, the prediction/rollup update rule is simply:

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

[1] 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.

[2] 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 $Q * P$.

An important property of the update is that it only requires knowledge of $\hat{P}$ and does not require P, and moreover, 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.

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.

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. Now since the normalization constant $\eta^{-1} = \sum_\sigma L(z|\sigma) \cdot P(\sigma)$ is given by the zeroth-order Fourier term, normalization can be implemented by simply dividing each Fourier coefficient by the zeroth-order term of the posterior $\left[\widehat{L^{(t)}P^{(t)}}\right]_{(n)}$.

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? The question is somewhat nontrivial since it is not immediately clear what it might mean to 'convolve' collections of matrices (of different dimensions). The 'correct' generalization of real multiplication for groups, as it turns out, is Kronecker multiplication of Fourier coefficient matrices [Willsky, 1978], followed by a projection of sorts onto the Fourier basis.

We will content ourselves to simply state the result below and say that it reduces to the regular definition of convolution for commutative groups.

Proposition 10. *Let $\hat{f}, \hat{g}$ be the Fourier transforms of functions f and g respectively. For each triple of partitions (λ, μ, ν) there exists a positive integer $z_{\lambda,\mu,\nu}$ and projection operators $P_{\lambda\mu}^{(\nu,\ell)}$ for each $\ell \in \{1, 2, \ldots, z_{\lambda\mu\nu}\}$ such that the Fourier tranform of the pointwise product fg is:*

$$\left[\widehat{fg}\right]_\nu = \sum_{\lambda\mu} \sum_{\ell=1}^{z_{\lambda\mu\nu}} (P_{\lambda\mu}^{(\nu,\ell)})^T \cdot \left(\hat{f}_\lambda \otimes \hat{g}_\mu\right) \cdot P_{\lambda\mu}^{(\nu,\ell)}. \tag{4.2.1}$$

The multiplicities, $z_{\lambda\mu\nu}$, and the projection operators, $P_{\lambda\mu}^{(\nu,\ell)}$, are known as the *Clebsch-Gordan series and coefficient matrices* respectively. See [Huang et al., 2008, Murnaghan, 1938] [3] for a proof of Proposition 10 and a discussion of methods for efficiently precomputing Clebsch-Gordan series/coefficients for low-frequency Fourier terms.

4.3 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:

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

For example, when $\lambda^{MIN} = (n-2, 1, 1)$, Λ is the set $\left\{\rho_{(n)}, \rho_{(n-1,1)}, \rho_{(n-2,2)}, \text{and } \rho_{(n-2,1,1)}\right\}$, 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. 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.

Even though the bandlimited distribution is sometimes a poor approximation to the true distribution, the marginals maintained 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.

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 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 irreducibles 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.

[3] For the sake of simplicity, the notation in this proposal is slightly different with that from Huang et al. [2008].

Theorem 11. *If $\lambda^{MIN} = (n-p, \lambda_2, \dots)$, and the Fourier domain 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:*

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

For example, if we condition in the Fourier domain 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.

Projecting to the marginal polytope. Despite the encouraging result of Theorem 11, 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 and 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 [Balakrishnan et al., 2004, Helmbold and Warmuth, 2007, Shin et al., 2003, 2005] 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:

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

Mixing Models		
Pairwise mixing	S_2	Identity confusion at tracks 1 and 2.
k-subset mixing	S_k	Identity confusion at tracks in $\{1, 2, 4, 6\}$.
Insertion mixing	n/a	Insert top card anywhere in the deck.
Observation Models		
Single track observation	S_{n-1}	Alice is at Track 1.
Multitrack observation	S_{n-k}	Alice is at Track 1, Bob is at Track 2, etc.
Bluetooth observation	$S_k \times S_{n-k}$	The girls occupy tracks $\{1, 2, 6, 8\}$
Pairwise ranking observation	S_{n-2}	Ubuntu is better than Windows

Table 4.2: Several useful types of mixing and observation models are summarized in the above table. In many of these cases, computing the appropriate Fourier transform reduces to computing the Fourier transform of the indicator function of some related subgroup of S_n, and so we also mention the relevant subgroup in the second column. In the third column we provide an example illustrating the semantics of each model.

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 12 (Plancherel Theorem)**.**

$$\sum_{\sigma}(f(\sigma) - g(\sigma))^2 = \frac{1}{|G|}\sum_{\nu} d_{\rho_\nu} Tr\left(\left(\hat{f}_{\rho_\nu} - \hat{g}_{\rho_\nu}\right)^T \cdot \left(\hat{f}_{\rho_\nu} - \hat{g}_{\rho_\nu}\right)\right). \qquad (4.3.1)$$

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 4.3.1, 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*.

4.4 Examples of common probabilistic models.

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 discuss a collection of useful models for which we *can* efficiently compute Fourier coefficients or even provide a closed-form expression. See Table 4.4 for a summary of the various possibilities. We discuss mixing models (models that we convolve against) and observation models (models that we condition on using Bayes rule). We will not provide explicit algorithms — the purpose of this section is to show a subset of the many possible models and to give an idea of the theoretical questions that we can answer about these models.

4.4.1 Mixing models

Pairwise mixing. 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 for the *pairwise mixing model* is therefore:

$$Q_{ij}(\pi) = \begin{cases} p & \text{if } \pi = \epsilon \\ 1-p & \text{if } \pi = (i,j) \\ 0 & \text{otherwise.} \end{cases} \tag{4.4.1}$$

Since Q_{ij} is supported on only two permutations, constructing the Fourier transform of a pairwise mixing model is straightforward from the definition of the Fourier transform.

k-subset mixing. It is not always appropriate to mix only two people at once and so we would like to formulate a mixing model which occurs over a subset of tracks, $X = \{t_1, \ldots, t_k\} \subset \{1, \ldots, n\}$. One way to 'mimic' the desired effect is to repeatedly draw pairs (i,j) from $\{t_1, \ldots, t_k\}$ and to convolve against the pairwise mixing models Q_{ij}. A better alternative is to construct the Fourier coefficient matrices for the *k-subset mixing model*:

$$Q_X(\pi) = \begin{cases} \frac{1}{k!} & \text{if } \pi \in S_X \subset S_n \\ 0 & \text{otherwise} \end{cases}, \tag{4.4.2}$$

where S_X is the subgroup of permutations which fix all elements which are not members of X. The k-subset mixing model says that with uniform probability, the set of tracks in X experienced some permutation of their respective identities. We see that in the case of subset mixing, the Q distribution is simply a multiple of the indicator function of the subgroup S_X. Since S_X can be viewed as a shifted version of S_k, we have the following fact.

Proposition 13. *The Fourier transform of the k-subset mixing model is a symmetric matrix at every partition λ and is nonzero only at partitions λ such that $\lambda \trianglerighteq (k, 1, \ldots, 1)$.*

Insertion mixing. As another example, we can consider the *insertion mixing model* in which we take the top card in some deck of n cards, and with uniform probability, insert it *anywhere* in the deck, preserving all other original relative orderings. The distribution for the insertion mixing model is given by:

$$Q^{insertion}(\pi) = \begin{cases} \frac{1}{n} & \text{if } \pi \text{ is a cycle of the form } (j, j-1, \ldots, 1) \\ 0 & \text{otherwise.} \end{cases} \tag{4.4.3}$$

Since the insertion mixing model is supported on n permutations, it is again simple to directly construct the Fourier transform from the definition.

4.4.2 Observation models

Single and multitrack observation. In the *single track observation model*, we acquire an identity measurement z_j at track j. In the simplest version of the model, we write the likelihood function as:

$$P(z_j = i|\sigma) = \begin{cases} \pi & \text{if } \sigma(j) = i \\ \frac{1-\pi}{n-1} & \text{otherwise} \end{cases}, \tag{4.4.4}$$

where i ranges over all n possible identities.

Equation 4.4.4 is useful when we receive measurements directly as single identities ("Alice is at Track 1 with such and such probability"). It is, however, far more common to receive lower level measurements that *depend* only upon a single identity, which we formalize with the following assumption:

$$P(z_j|\sigma) = P(z_j|\sigma(j)). \tag{4.4.5}$$

For example, we might have a color histogram over each individual ("Alice loves to wear green") and observe a single color per timestep. Or we might acquire observations in the form of color histograms and choose to model a distribution over all possible color histograms. Unsurprisingly, the single track observation model can be represented using only first-order Fourier terms.

We can also handle joint observations ("green blob at Track 1 *and* orange blob at Track 2") resulting in a model which requires higher order terms.

Proposition 14. *The Fourier transform of the subgroup indicator function δ_{S_k} is a diagonal matrix at every partition λ and is nonzero only at partitions λ such that $\lambda \trianglerighteq (k, 1, \ldots, 1)$.*

Bluetooth observations. We sometimes receive measurements in the form of unordered lists. For example, the *bluetooth model* is the likelihood function that arises if tracks $\{1, \ldots, p\}$ are within range of a bluetooth detector and we receive a measurement that identities $\{1, \ldots, p\}$ are in range. In sports, we might observe that the first p tracks belong to the red team and that the last q tracks belong to the blue team. And finally, in *approval voting*, one specifies a subset of approved candidates rather than, for example, picking a single favorite.

We consider two options for bluetooth-type situations. The first option is similar to the single track observation model and says that with some probability we receive the correct unordered list, and with some probability, we receive some other list drawn uniformly at random:

$$P^{bluetooth1}(z_{\{t_1,\ldots,t_k\}} = \{i_1, \ldots, i_k\} | \sigma) = \begin{cases} \pi & \text{if } \sigma(\{i_1, \ldots, i_k\}) = \{t_1, \ldots, t_k\} \\ \frac{1-\pi}{\binom{n}{k}-1} & \text{otherwise} \end{cases} . \quad (4.4.6)$$

Since the model can be written as the sum of a constant function and the indicator function of a coset of the subgroup $S_p \times S_q$, it has the same bandlimiting properties as the indicator function of $S_p \times S_q$. In particular, we know that $\hat{P}$ is nonzero at exactly $k + 1$ partitions:

Proposition 15. *$\hat{P}^{bluetooth1}_{\lambda}$ is nonzero only at partitions of the form $\lambda = (n - s, s)$, where $s \leq k$.*

In the second option, we allow for more error-tolerance by setting the likelihood to be proportional to the number of tracks that are correctly returned in the measurement:

$$P^{bluetooth2}(z_{\{t_1,\ldots,t_k\}} = \{i_1, \ldots, i_k\} | \sigma) \propto |\{t_1, \ldots, t_k\} \cap \sigma(\{i_1, \ldots, i_k\})|. \quad (4.4.7)$$

Our second bluetooth model can be expressed using only first order terms.

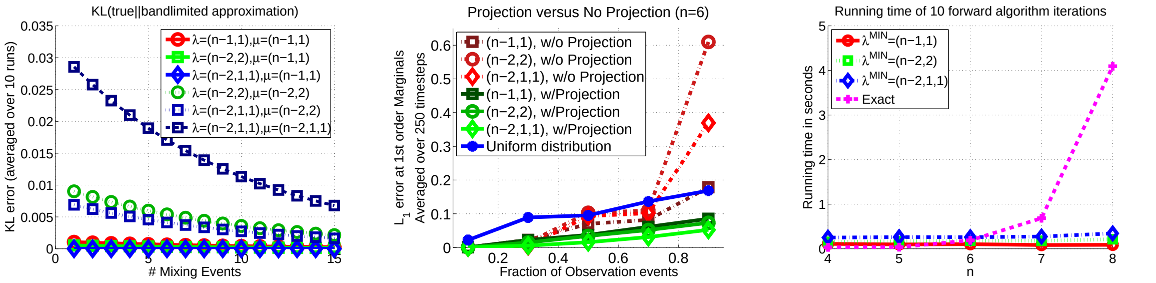
Proposition 16. *$\hat{P}^{bluetooth2}_{\lambda}$ is nonzero only at the first two partitions $\lambda = (n), (n - 1, 1)$.*

Pairwise ranking observation. Finally in the *pairwise ranking model*, we consider observations of the form "object j is ranked higher than object i" which can appear in various forms of voting and preference elicitation ("I like candidate x better than candidate y") or webpage/advertisement ranking. Our pairwise ranking model simply assigns higher probability to observations which agree with the ordering of i and j in σ.

$$P^{rank}(z_{ij} | \sigma) = \begin{cases} \pi & \text{if } \sigma(i) < \sigma(j) \\ 1 - \pi & \text{otherwise} \end{cases} . \quad (4.4.8)$$

Perhaps unsurprisingly, pairwise ranking models can be sufficiently captured by first-order and second-order (ordered) Fourier coefficients.

Proposition 17. *The Fourier coefficients of the pairwise ranking model, $\hat{P}^{rank}_{\lambda}$, are nonzero only at three partitions: $\lambda = (n)$, $(n - 1, 1)$, and $(n - 2, 1, 1)$.*



(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

Figure 4.2:

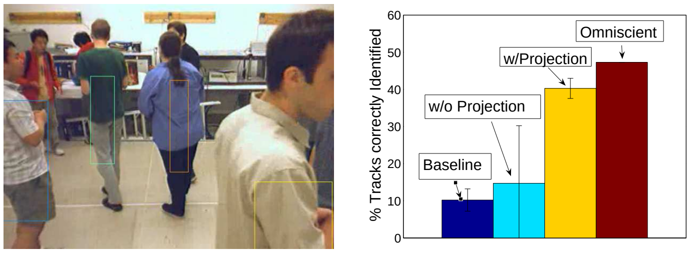
4.5 Experimental results

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.

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 4.2(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 4.2(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 11, 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 4.2(b). We also see that the Plancherel Projection step is fundamental, especially when mixing events are rare.

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 and Baum, 1993]. It is clear that our algorithm scales gracefully compared to the exact solution (Figure 4.2(c)), and in fact, we could not run exact inference for $n > 8$ due to memory constraints.



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

Figure 4.3: Evaluation on dataset from a real camera network.

Real camera network. We also evaluated our algorithm on data taken from a real network of eight cameras (Fig. 4.3(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 4.3(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 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.

4.6 Discussion

In this chapter, we have shown 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 remark that the Fourier domain inference framework discussed in this chapter is quite general. In fact, both the prediction/rollup and conditioning formulations hold over any finite group (and compact Lie group), providing a principled method for approximate inference for many problems with underlying group structure.

Chapter 5

Inference: Scaling to larger problems

We have shown in the previous chapters that low-frequency Fourier coefficients capture intuitive marginals and discussed several general and efficient approximate inference operations, like marginalization and conditioning, which can be performed completely in the Fourier domain. Despite having reduced *exponential time exact* inference to *polynomial time approximate* inference, however, the approach from the previous chapter suffers from two shortcomings in scalability and accuracy:

- While low-frequency Fourier terms provide a principled approximation to the underlying distribution, requiring only polynomially many numbers, the polynomials can grow too fast in practice.
- Bandlimited approximations which discard high frequencies are most effective with diffuse distributions since smooth functions tend to be well approximated by linear combinations of low frequency basis functions, but are less effective at approximating highly peaked distributions.

In a sense, the two shortcomings listed above are at odds with each other since we can always achieve better approximations to sharp functions by maintaining higher frequency Fourier coefficients. But an interesting observation is that when the distribution is sharp, it often makes more sense to break up the problem into smaller parts and to reason about disjoint subsets of objects independently of each other.

Consider again, the *identity management* problem, for example, that arises in multiobject tracking, where one must maintain a belief over the joint one-to-one assignment of n tracks to n identities (Alice is at Track 1, Bob is at Track 2, etc.). If we are completely uncertain about the assignment of people to tracks, and have a uniform distribution over permutation, this smooth distribution can be represented with only one parameter in the Fourier domain. At the limit when we know the location of every identity, our distribution becomes very peaked, and we need to maintain $n!$ Fourier coefficients. In this peaked setting, however, there is no reason to track all n identities jointly, and we can break up the problem into n subproblems. In this chapter, we propose a principled method based on exploiting probabilistic independence which overcomes both issues and show that in practice, we can indeed "get the best of both worlds". The contributions of this chapter include:

- Two algorithms, *Join* and *Split*, which operate entirely in the Fourier domain for combining factors to form a joint distribution and factoring a distribution, respectively.
- Theoretical results showing how many Fourier terms are required in our Join/Split algorithms to achieve a desired number of Fourier terms in the result.
- A method for detecting probabilistic independence using the Fourier coefficients of a distribution.
- An approach for adaptively decomposing large identity management problems into much smaller ones, improving previous methods both in scalability and approximation quality.

5.1 Independence in the Fourier domain

We seek to understand probabilistic independence in the Fourier domain, and specifically to characterize the structure that independence imposes on the Fourier coefficient matrices of distributions on the symmetric group. In contrast with the previous chapter (Ch. 4), the results of this section will be specific to distributions over permutations and will not carry over to other groups.

While bandlimiting our representation can decrease the storage cost from $O(n!)$ to some polynomial in n, maintaining the s^{th}-order marginals requires, in the worst-case, $O(n^{2s})$ space. Thus, for small n we can maintain higher order coefficients (larger s), but this representation quickly becomes intractable as n becomes large. Over the next sections, we will show how probabilistic independence is manifested in the Fourier coefficients of a distribution, and how, by exploiting this independence, we can break our distribution into smaller subgroups, which can allow us to maintain higher-order coefficients.

First-order conditions. We begin with a simple condition on the matrix of first-order marginal probabilities implied by independence.

Definition 18. Consider any subset $X \subset \{1, \ldots, n\}$ and its complement $\bar{X} \subset \{1, \ldots, n\}$. X and $\bar{X}$ are independent under a distribution $h(\sigma)$ if $h(\sigma)$ factors as the following product of distributions over X and $\bar{X}$: $h(\sigma) = f(\sigma_X) \cdot g(\sigma_{\bar{X}})$.

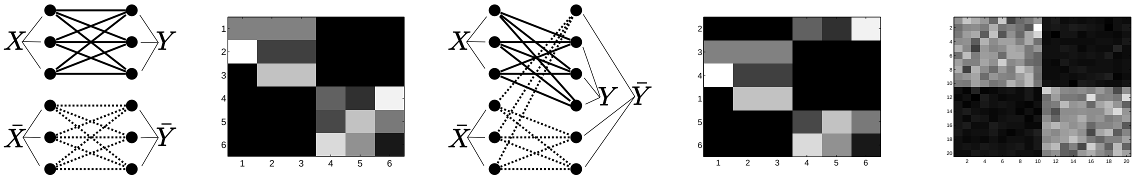
If $X = \{1, \ldots, p\}$, for example, $h(\sigma) = f(\sigma_1, \ldots, \sigma_p) g(\sigma_{p+1}, \ldots, \sigma_n)$. We will refer to X and $\bar{X}$ as *cliques* since the variables of X and $\bar{X}$ form disjoint cliques in the graphical model representation of the above independence relation. In this section, we discuss a simple first-order criterion for independence on the symmetric group since it provides some intuition for the higher-order case.

Due to the mutual exclusivity constraints associated with permutations, a necessary (but insufficient) condition for a distribution on permutations h to factor into a product of factors over X and $\bar{X}$, respectively, is that there must exist a subset $Y \subset \{1, \ldots, n\}$ of the same size as X such that, with probability 1, elements of X map to Y and elements of $\bar{X}$ map to $\bar{Y}$. We will refer to the above condition as the *first-order independence criterion*. Intuitively, a distribution can only factor into independent parts if the set $\{1, \ldots, n\}$ can be partitioned into disjoint subsets of objects which do not interact with one another. See Figures 5.1(a) and 5.1(c) for example.

Lemma 19 (first-order independence criterion)**.** *If σ_X and $\sigma_{\bar{X}}$ are independent under the distribution $h(\sigma)$ (i.e., $h(\sigma) = f(\sigma_X) \cdot g(\sigma_{\bar{X}})$), then there exists a subset $Y \subset \{1, \ldots, n\}$ with $|Y| = |X|$ such that $h(\sigma) = 0$ unless $\sigma_X \subset Y$.*[1]

To see why the first-order independence criterion is an insufficient indicator of independence, consider the simple example of a distribution on S_4 which always maps the set $X = \{1, 2\}$ to $Y = \{1, 2\}$ and the set $\bar{X} = \{3, 4\}$ to $\bar{Y} = \{3, 4\}$, but is constrained to map 1 to 1 whenever 3 maps to 3. In this case, the 1^{st}-order marginals exhibit independence, but the distribution is not independent when we examine the higher order components. Figure 5.1 illustrates a tracking problem where first-order independence may hold, but higher-order independence does not. Despite its insufficiency however, the first-order independence plays a crucial role for us in several ways — for example, we will later discuss how it can serve as a first pass at detecting independence as it reduces the detection problem into a clustering-like problem.

[1] With some abuse of notation, we use $\sigma_X \subset Y$ to denote the fact that the permutation σ maps elements X (people) to some permutation of elements Y (tracks).



(a) (b) (c) (d) (e)

Figure 5.1: Two examples illustrating the disjoint subsets criterion. In (a) and (c), we show a bipartite graph between identities (left) and tracks (right) with edges indicating that some identity is associated with some track with nonzero probability. By the disjoint subsets criterion, if X and $\bar{X}$ are indeed independent, we can partition the tracks into subsets Y and $\bar{Y}$ with $|X| = |Y|$ and $|\bar{X}| = |\bar{Y}|$. In (b) and (d), we show an example of what the corresponding first-order marginals would look like. In practice, we expect first-order independence to only hold approximately, as in (e).

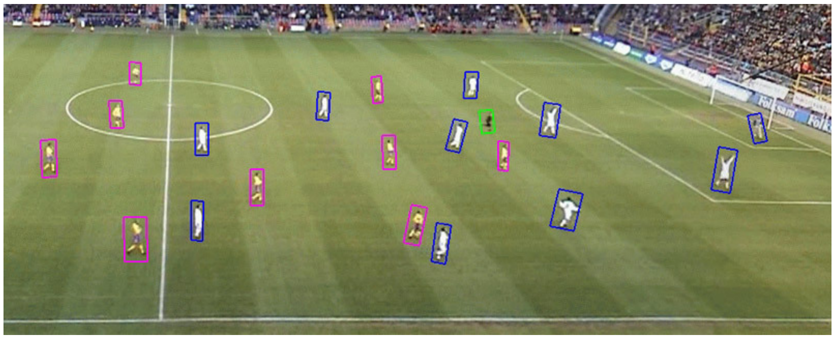


Figure 5.2: (Figure from Sullivan and Carlsson [2006]) An example where the first-order independence condition does not imply higher-order condition. One can imagine tracking the white and yellow teams independently, but in doing so, losing the ability to reason from the fact that certain players on the white team always guard certain players on the yellow team.

Higher order characterizations and algorithms. We can now generalize the first-order independence condition to higher-orders. As with the first-order matrix of marginals, we will see higher-order Fourier coefficient matrices are block-diagonal with respect to a particular basis; but in addition, we will see that the higher-order nonzero blocks also have *Kronecker product* structure. We state the main result in the following proposition.

Proposition 20. *Given Fourier coefficients of two independent factors f and g, the Fourier coefficient matrices of the joint distribution h, are:*

$$\widehat{h}_\lambda = \left[\widehat{f\cdot g}\right]_\lambda = {L^\lambda_{\mu\nu}}^T \cdot \bigoplus_{\mu,\nu}\bigoplus_{\ell=1}^{c^\lambda_{\mu,\nu}} \left(\widehat{f}_\mu \otimes \widehat{g}_\nu\right)\cdot L^\lambda_{\mu\nu}. \tag{5.1.1}$$

Eqn. 5.1.1 characterizes the form of the Fourier matrices of the joint distribution at all frequencies. We see that each nonzero block has Kronecker structure at higher orders and that the coefficients of the joint are redundant in the sense that information at lower frequencies of the factors f and g are duplicated to multiple higher frequencies of h. As it turns out, the multiplicities, $c^\lambda_{\mu,\nu}$, are equivalent to what mathematicians have studied in different contexts as *Littlewood-Richardson (LR)* coefficients. The LR coefficients tell us which crossterms contribute to the joint. For example, it can be shown that that first-order terms corresponding to the partition $(n-1,1)$ can be reconstructed using only three terms, $(p)\otimes(q)$, $(p-1,1)\otimes(q)$, and $(p)\otimes(q-1,1)$. Computing the LR coefficients has been shown, in general, to be a $\#P$-complete problem Narayanan [2006]. For low-order Fourier terms (corresponding to partitions with only a few rows), however, the *Littlewood-Richardson rule* (Sagan [2001]) computes the LR coefficients in reasonable time (see appendix).

5.1.1 Algorithms and Approximation

We now discuss algorithms for merging independent factors to form a joint (*Join*), and for extracting independent factors from a joint (*Split*) based on our Fourier domain factorization (Proposition 20). There are two problems that one encounters in practice; first, it is impossible to maintain a complete set of Fourier coefficients, and second, it is rare for distributions to factor completely. We present novel theoretical results in this section addressing both issues and show that our algorithms behave reasonably in bandlimited and near-independent (rather than fully-independent) settings.

Join. The simplest operation of the two is the *Join* algorithm — which is a straightforward implementation of Equation 5.1.1. Given the Fourier transforms $\hat{f}$ and $\hat{g}$, the Fourier transform of the joint, $\hat{h}$, can be constructed by forming the direct sum of appropriate tensor product terms $\hat{f}_\mu \otimes \hat{g}_\nu$, and conjugating by the precomputed coupling matrix $L^\lambda_{\mu\nu}$. The complexity of the Join operation is dominated by the cost of matrix multiplication ($O(d_\lambda^3)$ for each partition λ), and is therefore no more expensive than the convolution operations from the previous chapter.

One might worry that we would require maintaining high-frequency terms of the independent factors in order to construct low frequency terms of the joint. We show, using the Littlewood-Richardson rule, that this is not the case when we maintain s^{th}-order marginals. For any integer s such that $0 \leq s < n$, define the following partitions:

$$\lambda^{MIN} = (n-s, \underbrace{1,\ldots,1}_{s \text{ times}}), \qquad \mu^{MIN} = (p-k, \underbrace{1,\ldots,1}_{k \text{ times}}), \qquad \nu^{MIN} = (q-\ell, \underbrace{1,\ldots,1}_{\ell \text{ times}}),$$

where $k = \min(s, p-1)$ and $\ell = \min(s, q-1)$. We have the following guarantee.

Theorem 21. *Given marginals of type μ^{MIN} for f and of type ν^{MIN} for g,* Join *returns Fourier coefficients of the joint distribution h which can reconstruct marginals of type λ^{MIN}* exactly.

Theorem 21 formalizes the intuitive idea that it is possible, using the Join algorithm, to exactly construct s^{th}-order marginals of the joint distribution using only the s^{th}-order marginals of each independent factor. The proof of Theorem 21 is given in the appendix. A more general principle holds for other partitions which do not take the form $\lambda^{MIN} = (n-s, 1, \ldots, 1)$ (see Appendix), but we will focus on the simpler and more intuitive case of s^{th}-order marginals.

Split. Given the Fourier transform of the joint, $\hat{h}$, we wish to formulate an algorithm which computes the Fourier coefficients of the factors, $\hat{f}$ and $\hat{g}$, assuming that the sets $X = \{1, \ldots, p\}$ and $\bar{X} = \{p+1, \ldots, n\}$ are independent under h. One can imagine "inverting" the Join algorithm by computing $L^\lambda_{\mu\nu} \cdot \hat{h}_\lambda \cdot {L^\lambda_{\mu\nu}}^T$ and reading off the $\hat{f}_\mu$ and $\hat{g}_\nu$ from the resulting matrix, $\bigoplus_{\mu,\nu} \bigoplus_{\ell=1}^{c^\lambda_{\mu\nu}} \hat{f}_\mu \otimes \hat{g}_\nu$. The difficulty is that the matrices $\hat{f}_\mu \otimes \hat{g}_\nu$, in general, only determine $\hat{f}_\mu$ and $\hat{g}_\nu$ up to a scaling factor, and in the approximate case when X and $\bar{X}$ are only "nearly" independent, the appropriate blocks of the matrix $L^\lambda_{\mu\nu} \cdot \hat{h}_\lambda \cdot {L^\lambda_{\mu\nu}}^T$ do not take the form $A \otimes B$.

Happily though, we are in fact able to always construct coefficients of $\hat{f}$ and $\hat{g}$ using *only* blocks of the form $\hat{f}_\mu \otimes 1$, or $1 \otimes \hat{g}_\nu$, allowing us to literally read off the matrices for $\hat{f}_\mu$ and $\hat{g}_\nu$.

Theorem 22. *For any $\mu \trianglerighteq \mu^{MIN}$, there exists a block of $L^\lambda_{\mu\nu} \cdot \hat{h}_\lambda \cdot {L^\lambda_{\mu\nu}}^T$ for some $\lambda \trianglerighteq \lambda^{MIN}$ which is identically equal to $\hat{f}_\mu$.*

Likewise, for any $\nu \trianglerighteq \nu^{MIN}$, there exists a block of $L^\lambda_{\mu\nu} \cdot \hat{h}_\lambda \cdot {L^\lambda_{\mu\nu}}^T$ for some $\lambda \trianglerighteq \lambda^{MIN}$ which is identically equal to $\hat{g}_\nu$. See Algorithm 5.1 for pseudocode for the Split algorithm and the appendix for more details.

Algorithm 5.1: Pseudocode for the *Split* algorithm.

```
SPLIT
foreach partition $\mu$ of p such that $\mu \trianglerighteq \mu^{MIN}$ do
    $\lambda \leftarrow (\mu_1 + n - p, \mu_2, \dots)$ ;
    $\hat{f}_\mu \leftarrow (\mu, (q))$-block of the matrix $L^{\lambda}_{\mu\nu} \cdot \hat{h}_\lambda \cdot {L^{\lambda}_{\mu\nu}}^T$ ;
end
foreach partition $\nu$ of q such that $\nu \trianglerighteq \nu^{MIN}$ do
    $\lambda \leftarrow (\nu_1 + n - q, \nu_2, \dots)$ ;
    $\hat{g}_\nu \leftarrow ((p), \nu)$-block of the matrix $L^{\lambda}_{\mu\nu} \cdot \hat{h}_\lambda \cdot {L^{\lambda}_{\mu\nu}}^T$ ;
end
Normalize $\hat{f}$ and $\hat{g}$;
```

As a corollary, we obtain a converse to Theorem 21 which says that given the s^{th}-order marginals of the joint, we will be able to recover the s^{th}-order marginals of the factors.

Corollary 23. *Given marginals of type λ^{MIN} of the joint h,* Split *returns Fourier coefficients of the factors f and g which can be used to exactly reconstruct marginals of type μ^{MIN} and ν^{MIN}, respectively.*

Although exploiting independence can significantly reduce computation, it is rare for full independence to hold in practice (Figure 5.1(e), for example). Consider calling the Split algorithm on a distribution which does *not* factor into distributions on S_p and S_q. Ideally, one would, in this case, hope to obtain the Fourier transform of the appropriate marginal distributions of $(1, \dots, p)$ and $(p+1, \dots, n)$. As it turns out, the Split algorithm can return exact marginals whenever first-order independence conditions are satisfied.

Theorem 24. *Whenever first-order independence conditions hold for a distribution h, the output of Split can be used to exactly reconstruct* all marginals *of h.*

When first-order independence does not hold, the resulting coefficients do not correspond to a properly normalized distribution, and in particular, $\hat{f}_{(p)}$ is the amount of mass assigned to elements of the subgroup $S_p \times S_q$ (instead of 1). However, since $\hat{f}$ still corresponds to a positive function, one can easily normalize $\hat{f}$ by dividing all coefficients by the zeroth-order Fourier coefficient, $\hat{f}_{(p)}$, without requiring a projection to the marginal polytope as in Huang et al. [2007]. Thus when a distribution is near first-order independent, we recover approximate marginals.

Detecting independence. We now discard the assumption that $X = Y = \{1, \dots, p\}$ and deal with the problem of explicitly finding sets X and Y such that $h(\sigma_X \subset Y) = 1$ and $h(\sigma_{\bar{X}} \subset \bar{Y}) = 1$ as in the disjoint subsets criterion. We begin with the simple observation that, if we *knew* the sets X and Y, then the first-order matrix of marginals would be rendered block diagonal under an appropriate reordering of the rows and columns (Figure 5.1(b)). Since X and Y are unknown, our task is to find permutations of the rows and columns of the first-order matrix of marginals (Figure 5.1(d)) to obtain a block diagonal matrix. Viewing the matrix of first-order marginals as a set of edge weights on a bipartite graph between tracks and identities, we approach the detection step as a *biclustering* problem (in which one simultaneously clusters the tracks and identities) with an extra *balance* constraint forcing $|X| = |Y|$. In our experiments, we use the SVD-based technique presented in Zha et al. [2001] which finds bipartite graph partitions optimizing the normalized cut measure, modified to satisify the balance constraint.

Assuming now that we have obtained the sets X and Y via the above clustering step, we can call the Split algorithm by first renaming the tracks and identities so that $X = Y = \{1, \dots, p\}$. Suppose that, to achieve this reordering, we must permute the X (people) using a permutation π_1 and the Y (tracks) using π_2. The *Shift Theorem* (Prop. 5) can be applied to reorder the Fourier coefficients according to these new labels, and we can then apply our splitting procedure unchanged.

We have focused on detecting independence in the first-order sense. As discussed in Section 5.1, first-order independence is necessary, but insufficient for higher order independence. However, as we showed in Section 5.1.1, we can approximately recover marginal probabilities when a distribution is near first-order independent. Furthermore, our biclustering approach can also be viewed as a first pass for proposing candidate splits. Once this factoring is performed, we can measure its effect on higher orders, e.g., using the Plancherel Theorem Diaconis [1988] to measure the distance between the original coefficients and the factored result, and decide whether or not to retain the partition.

5.2 Example: Adaptive identity management.

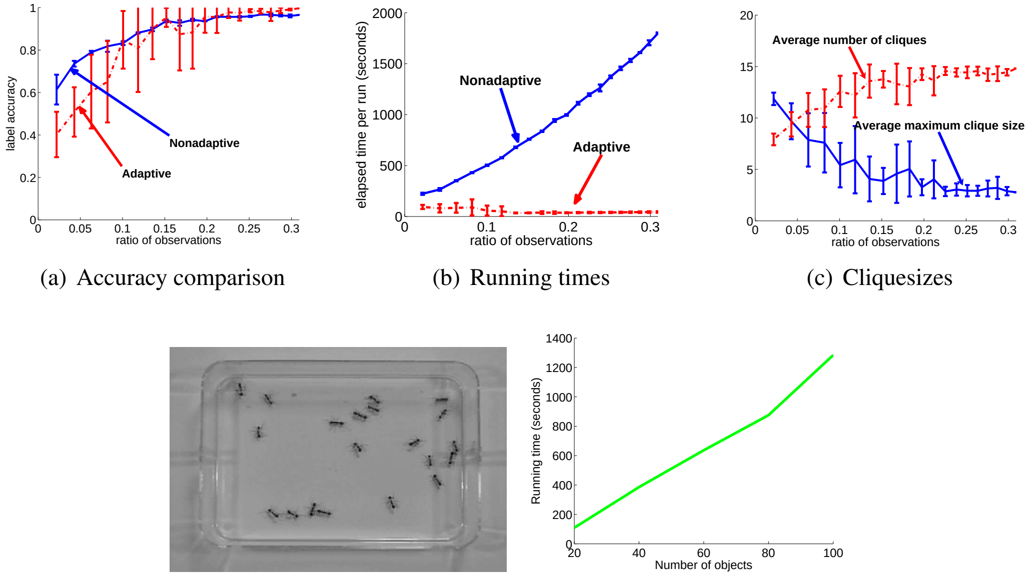
As an application, we use the algorithms described here in the identity management setting. In both Huang et al. [2007] and Kondor et al. [2007], one reasons *jointly* over assignments of all n tracks to all n identities. In realistic settings however, we believe that it is often sufficient to only reason over small cliques of tracks at a time. Thus instead of maintaining Fourier coefficients over all of S_n, we search for independent cliques and *adaptively* split the distribution into factors over smaller cliques whenever possible.

In our adaptive approach, we maintain a collection of disjoint cliques over the tracks and identities. After conditioning on any observation, we attempt to split. We also force splits whenever cliques grow to be too large to handle. Upon splitting, we allow the representational size to grow to higher orders — thus for very large n, we might only maintain first-order coefficients, but for smaller sized cliques, we might choose to represent higher-order coefficients. In practice, since the joint distribution is only nearly-independent, we also perform the Plancherel projection step described in Huang et al. [2007] for enforcing positive marginals after we perform a split step. Finally, whenever mixing events occur between tracks belonging to distinct cliques, we merge the cliques using our Join algorithm and perform a mixing on the newly formed joint distribution.

5.3 Experimental results

We evaluted our adaptive identity management algorithm on a biotracking dataset from Khan et al. [2006]. In their data, there are 20 ants (Fig. 5.3(d)) moving in an enclosed area. The data is interesting for our purposes since it is a relatively large n compared to many multiobject tracking datasets with interesting movement patterns and plenty of mixing events (which we log whenever ants walk within some distance of each other). At each timestep, we allow each ant to 'reveal' its identity with some probability (in our experiments, ranging from $p_{obs} = .005$ to $p_{obs} = .05$ per timeframe), and our task is to jointly label all tracks with identities for all timeframes. We measure accuracy using the fraction of correctly labeled tracks over the entire sequence (note that the accuracy of random guessing is $1/n = 5\%$ in expectation). As a splitting criterion, we decide to to split if, after clustering, the sum over all off-block elements fall below a certain threshold ϵ (in all experiments, we fixed $\epsilon = 1/(2n)$).

In Figures 5.3(a) and 5.3(b), we compare the performance of an adaptive approach against the nonadaptive algorithm from Huang et al. [2007] as we vary the ratio of observations to mixing events. Figure 5.3(a) shows that the two algorithms perform similarly in accuracy, with the nonadaptive approach faring slightly better with fewer observations (due to more diffuse distributions) and slightly worse with more observations (due to the fact that the adaptive approach can represent higher-order Fourier terms). The real advantage of our adaptive approach is shown in Figure 5.3(b) which plots a running time comparison. Since the conditioning step is the complexity bottleneck of performing inference in the Fourier domain, the running time scales according to the proportion observations. However, since the adaptive algorithm



(a) Accuracy comparison

(b) Running times

(c) Cliquesizes

(d) Running times vs. n

(e) Running times vs. n

Figure 5.3: Experimental results on biotracking data

typically conditions smaller cliques on average (especially with more observations), we see that it is a far more scalable algorithm. In Figure 5.3(c), we plot the average number of cliques and sizes of cliques which were formed in the same experiment. As expected, we see that the cliques get smaller and more numerous as the number of observations grows.

Finally, we simulated larger tracking problems by taking m different segments of the ant data and tracking $m \cdot n$ ants at the same time allowing for ants to 'teleport' to other segments with some probability. Figure 5.3(e) shows a comparison of average running time for these larger problems. Note that at such sizes, we can no longer feasibly run the original nonadaptive algorithms from Huang et al. [2007], Kondor et al. [2007].

5.4 Discussion

A pervasive technique in machine learning for making large problems tractable is to exploit independence structures for decomposing large problems into much smaller ones. It is the structure of (conditional) independence, for example, which has made Bayes net and Markov random field representations so powerful. In this paper, we have contributed to the existing collection of efficient Fourier-theoretic inference operations by presenting a formulation of probabilistic independence for permutations based on the Littlewood-Richardson decomposition. While such decompositions have been used in mathematics, we are the first to use them in the context of probabilistic independence and to consider their bandlimiting properties. Additionally, we showed our algorithms to be well-behaved in near-independent scenarios and that we can even obtain exact marginals whenever first-order independence is satisfied. Combined with the bandlimited inference algorithms from Huang et al. [2007], Kondor et al. [2007], we believe that our algorithms will contribute to making these Fourier methods highly scalable and practical.

Finally we view our contributions as a first step towards understanding and exploiting more intermediate notions which lie somewhere between full independence and fully connected, such as conditional or context-specific independence which have proven themselves to be indispensible in the fields of machine learning and AI.

Chapter 6

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 (from both the statistics and machine learning communities) who have studied distributions over permutations/rankings [Critchlow, 1985, Fligner and Verducci, 1986, Lebanon and Mao, 2007, Mallows, 1957, Meila et al., 2007, Taylor et al., 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 exponential family models. For example, the Mallows model [Mallows, 1957] defines a Gaussian-like distribution over permutations as:

$$P(\sigma; c, \sigma_0) \propto \exp\left(-cd(\sigma, \sigma_0)\right), \tag{6.0.1}$$

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}.

Distance based exponential family models have the advantage that they can compactly represent distributions for very large n, and admit conjugate prior distributions Meila et al. [2007]. Estimating parameters has been a popular problem for statisticians — recovering the optimal σ_0 from data is known as the *consensus ranking* or *rank aggregation* problem and is known to be NP-hard Bartholdi et al. [1989]. Many authors have focused on approximation algorithms instead.

Like Gaussian distribution, distance based models also tend to lack flexibility, and so Lebanon and Mao [2007] 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 this proposal, Fourier based methods [Diaconis, 1988, Huang et al., 2007, Kondor et al., 2007] offer a principled alternative method for compactly representing distributions over permutations and performing efficient probabilistic inference operations. Our work draws primarily 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 to our approach and place our contributions in the context of prior research.

6.1 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 and 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 and 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 and 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 and Sastry, 2005, Oh et al., 2004].

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 Chapter 3, 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 [Schumitsch et al., 2006a,b, Shin et al., 2005] in which the density is parameterized as:

$$P(\sigma;\Omega) \propto \exp \mathrm{Tr}\left(\Omega^T \cdot M(\sigma)\right),$$

where $M(\sigma)$ denotes the permutation matrix associated with σ. 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.

6.2 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 and 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 certain classes of observation models, 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 terms, 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 used in Kondor et al. [2007] is fully specified by two Fourier components, and that both algorithms have identical output. In this setting, our asymptotic time complexity is $O(D^3 n^2)$, where D is the degree of the largest maintained coefficient matrix. 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 are typically similarly sparse 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. We remark that our independence factorizations (Chapter 5) can be used in exactly the same way in their setting and would conceptually offer the same scaling benefits.

Chapter 7

Conclusions and Future Directions

We have presented the beginnings of a framework for efficiently reasoning with permutations in the Fourier domain. To summarize a few of the main contributions to-date, we have covered the following topics.

- We have formulated general and efficient probabilistic inference algorithms which operate entirely in the Fourier domain allowing for a principled tradeoff between computational complexity and approximation accuracy. Among the possible operations are:
  1. Computing marginals of a distribution,
  2. Conditioning on evidence,
  3. Detecting independent subsets of variables,
  4. Joining independent factors to form a joint distribution, and
  5. Splitting a distribution into independent factors.
- We have presented derivations of the Fourier transforms of a variety of common mixing and likelihood models.
- We have analyzed the errors which can be introduced by bandlimiting a probability distribution and presented theoretical results showing how the errors 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.
- We developed a method for adaptively decomposing large inference problems into much smaller ones based on exploiting probabilistic independence, and demonstrated improvements over non-adaptive methods in both scalability and accuracy.
- We empirically evaluated our algorithms on several challenging real datasets and showed that in addition to its theoretical properties, our methods perform well in practice.

7.1 Future directions, open questions

There are many possible extensions and a plethora of unsolved questions that we would like to answer.

Alternative scaling approaches. While full independence might work as an approximation for certain identity management scenarios, it is clearly far too limiting an assumption for most realistic applications, particularly those relating to ranking and matching. In contrast with graphical models which offer a graceful 'interpolation' between naive Bayes models and fully connected models, it is unlikely that conditional independence will work well in the permutation setting. However there are a few natural alternatives to explore which will allow us to scale up to massive problems without making such strong assumptions on problem structure. For example, another popular variety of independence structure is context-specific independence. More promising is a generalization of full independence that we call *shuffled independence* that may be more useful in settings involved ranked data. Instead of assuming that the joint factors as a product of marginals over σ_p and σ_q: $h(\sigma) = f(\sigma_p) \cdot g(\sigma_q)$, shuffled independence assumes that the joint can be written as a product of marginal probabilities of *relative rankings* followed by a convolution, allowing for probability mass to be smeared into the 'zero regions'.

Approximation guarantees. We have some basic results about the errors that can be introduced by bandlimiting and how those errors can be propagated by typical inference operations. Currently, what is missing is a Boyen/Koller-like result (Boyen and Koller [1998]) which would presumably bound the deviation from the true distribution at all future timesteps assuming certain conditions on the mixing distribution. While it is true that the KL-divergence between the true distribution and the approximate distribution can be shown to decrease monotonically at each timestep, we do not have similar results yet for the conditioning step and it is unclear how one should work with the KL-divergence functional in the Fourier domain. Any theoretical bound is more likely to be stated in terms of an L_2 related distance.

Optimization. A crucial problem which we have ignored up until now has been that of optimizing a function over permutations. A natural question that one might ask is whether it is easy to maximize a bandlimited function? In particular, for a fixed bandlimiting level, we would like to know if it is possible to maximize functions within the bandlimited class in polynomial time. Things seem rosy given that the maximum value of a first-order function on permutation can be found in polynomial time ($O(n^3)$) using the popular Hungarian (Kuhn-Munkres) algorithm Munkres [1957]. Unfortunately, beyond first-order, functions become far more difficult to optimize, and in fact we can show that the problem of optimizing a second-order function is NP-hard:

Theorem 25. *Any instance of the traveling salesman problem* (TSP) *can be reduced to the problem of optimizing a second-order function on permutations in polynomial time.*

To make matters worse, the reduction given in Theorem 25 is an approximation preserving reduction and there are no constant factor approximation algorithms for the general TSP case. However, it would be useful in many situations to have optimization routines which account for higher order effects. To this end, we plan to explore various strategies for optimizing these bandlimited functions that work well in practice. Another avenue to explore is to search for problem structure which can be exploited to guarantee optimality or near-optimality in certain cases (like convexity and submodularity, respectively).

Measuring uncertainty. It would be useful to measure the uncertainty of a distribution over permutations. For example, the entropy functional is one common such measure. The challenging part of writing entropy in terms of Fourier coefficients, however, is that logarithms are not easily expressed using Fourier coefficients. Furthermore, with a truncated set of coefficients, it is impossible to hope for an exact measure, and the best one could do would be to derive upper/lower bounds on the uncertainty measure. We have considered Taylor approximations to the entropy functional, but so far there has not been much success with the current approach due to poor performance in bandlimited settings. It seems more useful to look for other measures of uncertainty that are more naturally expressible with Fourier coefficients.

Learning. 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.

Another unaddressed issue is sampling. In signal processing, the well-known Shannon-Nyquist theorem assures us that we can exactly reconstruct a signal if we sample at twice the highest frequency contained in the signal. Is there an analogous result that would hold over the symmetric group? If a signal is sparse with respect to some basis, can we adapt the results of compressed sensing [Candès and Tao, 2006, Candès et al., 2005, Donoho, 2004, Jagabathula and Shah, 2008] to the symmetric group?

Open questions in Fourier analysis. The following list is a compilation of several open theoretical problems relating to Fourier analysis that have come up in our past research.

- Some of the coupling matrices (Clebsch-Gordan, Littlewood-Richardson) which are used in this thesis seem so fundamental and yet there are essentially no known closed-form expressions for them on the Symmetric group (there *do* exist closed-form expressions for several Lie groups) While a variety of generic methods exist to compute coupling matrices for finite groups (Chen [1989], Huang et al. [2008]), it would be not only be more theoretically satisfying to have closed-form expressions, but they would presumably be more compact. To this end, we have worked out several formulas for low-order coupling matrices which convert between marginals and irreducible Fourier coefficients. We would like to derive similar CG and LR formulas.

  An even more ambitious question is: does there exist a way factor a large coupling matrices into a product of sparse matrices? If we can, then we can compute the change of basis more efficiently. We are optimistic that there do exist such factorizations, but they will not be easy to find.

  Surprisingly, many of the FFT [Clausen and Baum, 1993] methods for the symmetric group do not require knowledge of coupling matrices and Kondor et al. [2007] was able to exploit this fact to formulate an efficient algorithm for conditioning observations of the form "Identity i is at track j". An interesting project would be search for Fourier bases which would allow one to condition using other types of observations without knowledge of coupling matrices.
- A somewhat similar goal that we have been pursuing is in searching for closed-form or efficient algorithms for computing the Fourier transforms of various probabilistic models such as those outlined in Section 4.4. We now know a lot of facts about these models, but there is still work to be done with regards to exploiting known sparsity structure, writing down closed-form expressions, and exploring the Fourier transforms of more interesting models.
- In Section 4.3, 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 at higher frequencies. 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 by polynomially many constraints.

- The Fourier theoretic framework in Chapter 4, as we mentioned, is not specific to the symmetric group, with most of its results carrying over to finite and compact Lie groups. 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.

  In a similar vein, we are also considering sets which do not have group structure but are acted upon by some group. One such structure is known as the *rook monoid* which is the set of partial matchings between two collections A and B. We believe that studying distributions over the rook monoid will prove to be useful for matching problems in computer vision which need to deal with spurious measurements that need not be matched with anything.

Applications. Finally, despite the fact that the identity management problem has provided an ideal setting for our algorithms up until now, we believe that our contributions can make an impact in many other application domains such as ranking and matching, where one must reason explicitly with some representation of uncertainty over permutations.

Rankings are a particularly important application today given their relevance in advertising and search engines, and the Fourier approach may allow us to think about these problems in a new light since higher-order information seems to be fairly important in ranking problems (for example, one would want to infer facts like “people who like movie X also like movie Y”). Ranking problems, on the other hand, are also especially difficult due to n often in the thousands and massive amounts of available data. Our current approach for scaling to large problems by exploiting probabilistic independence is inappropriate for the ranking setting due to the hard first-order independence condition. Other ranking challenges include understanding how one would condition on partially specified data (in the form of relative rankings, or approval sets, etc.) and making use of side information to generalize to unseen objects.

We are also investigating the possibility of applying our Fourier-based methods in matching/correspondence type problems from computer vision and graphics. While matching problems from vision are similar to identity management problems, they sometimes rely more on higher-order information (like distances between detected feature points) and less on first-order information (like local appearance descriptors). Since matching has been addressed in numerous works, we will need to understand the types of scenarios in which reasoning with uncertainty is a crucial aspect of the problem and our algorithms would be expected to be more practical than naive methods.

7.2 Expected Contributions and Timeline

The proposed work will only touch on a selection of the problems listed above. In addition to the contributions already enumerated in this chapter, the main expected additional contributions include:

- An understanding of the connection between our methods and the distance based exponential family models that have been popular in the statistics literature.
- An optimization algorithm that is fast in practice and (unlike the Hungarian algorithm) can make use of higher frequency Fourier terms

- A method to learn models of ranked data with respect to which we can perform efficient inference for unseen test data (this is like the structure learning problem from graphical models).

The following is a proposed timeline leading up to a thesis defense in the Summer of 2010.

- (Spring, 2009)
    - Investigate more general forms of independence for permutations
    - Examine ranking datasets and understand when these generalized notions of independence are manifested in real data.
    - Develop a method for exploiting such structure for scalable inference.
    - Develop a way to learn model parameters of ranked data.
    - Validate on real datasets by testing on an inference problem.
    - Compare results to exponential family based results.
- (Fall, 2009)
    - Explore the optimization problem.
    - Draw theoretical connections to related literature on exponential family models and consensus rankings.
    - Begin writing thesis.
- (Spring/Summer, 2010)
    - Wrap up implementations and experiments.
    - Finish writing thesis.
    - Thesis defense.

Bibliography

David Aldous and Persi Diaconis. Shuffling cards and stopping times. *American Mathematical Monthly*, 93(5):333–348, 1986. 1

H. Balakrishnan, I. Hwang, and C. J. Tomlin. Polynomial approximation algorithms for belief matrix maintenance in identity management. In *Proceedings of the 43rd IEEE Conference on Decision and Control, Bahamas*, 2004. 1, 4.3

Y. Bar-Shalom and T.E. Fortmann. *Tracking and Data Association*. Academic Press, 1988. 1, 6.1

J. Bartholdi, C. A. Tovey, and M. Trick. Voting schemes for which it can be difficult to tell who won. *Social Choice and Welfare*, 6(2), 1989. 6

Dave Bayer and Persi Diaconis. Trailing the dovetail shuffle to its lair. *The Annals of Probability*, 1992. 1

X. Boyen and D. Koller. Tractable inference for complex stochastic processes. In *UAI*, 1998. 4.1, 7.1

Emmanuel J. Candès and Terence Tao. Near-optimal signal recovery from random projections: Universal encoding strategies? *IEEE Transactions on Information Theory*, 52(12):5406–5425, 2006. 7.1

Emmanuel J. Candès, Justin Romberg, and Terence Tao. Stable signal recovery from incomplete and inaccurate measurements. 2005. 7.1

Jin-Quan Chen. *Group Representation Theory for Physicists*. World Scientific, 1989. 7.1

Michael Clausen and Ulrich Baum. Fast fourier transforms for symmetric groups: Theory and implementation. *Mathematics of Computations*, 61(204):833–847, 1993. 1.1, 4.5, 6.2, 7.1

J.B. Collins and J.K. Uhlmann. Efficient gating in data association with multivariate distributed states. *IEEE Trans. Aerospace and Electronic Systems*, 28, 1992. 1, 6.1

James W. Cooley and John W. Tukey. An algorithm for the machine calculation of complex fourier series. *Mathematics of Computation*, 19(90):297–301, 1965. 3

Timothee Cour, Praveen Srinivasan, and Jianbo Shi. Balanced graph matching. In *Advances in Neural Information Processing Systems 19*. MIT Press, Cambridge, MA, 2007. 1

I.J. Cox and S.L. Hingorani. An efficient implementation of Reid's multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking. In *International Conf. on Pattern Recognition*, pages 437–443, 1994. 1, 6.1

D. E. Critchlow. *Metric Methods for Analyzing Partially Ranked Data*. Springer-Verlag, 1985. 1, 6

Zajj Daugherty, Alexander K. Eustis, Gregory Minton, and Michael E. Orrison. Voting, the symmetric group, and representation theory, 2007. 1

P. Diaconis. *Group Representations in Probability and Statistics*. IMS Lecture Notes, 1988. 1, 2, 3, 3.2, 4.2, 4.2, 4.3, 5.1.1, 6, 6.2

Persi Diaconis. A generalization of spectral analysis with application to ranked data. *The Annals of Statistics*, 17(3):949–979, 1989. 1, 1.1

David Donoho. Compressed sensing. 2004. 7.1

Alexei A. Efros, Alexander C. Berg, Greg Mori, and Jitendra Malik. Recognizing action at a distance. In *IEEE International Conference on Computer Vision*, pages 726–733, Nice, France, 2003. 1.2

Pedro F. Felzenszwalb, Daniel P. Huttenlocher, and Jon M. Klein berg. Fast algorithms for large-state-space hmms with applications to web usage analysis. In Sebastian Thrun, Lawrence Saul, and Bernhard Schölkopf, editors, *Advances in Neural Information Processing Systems 16*. 2004. 4.2

M.A. Fligner and J.S. Verducci. Distance based ranking models. *Journal of the Royal Statistical Society*, 48, 1986. 1, 6

Yoav Freund, Raj Iyer, Robert E. Schapire, and Yoram Singer. An efficient boosting algorithm for combining preferences. *JMLR.*, 4:933–969, 2003. ISSN 1533-7928. 6

Ulf Grenander. *Probabilities on Algebraic Structures*. Wiley, 1963. 6.2

Leonidas J. Guibas. The identity management problem — a short survey. In *Proc. International Conf. on Information Fusion*, 2008. 1, 1.2

David P. Helmbold and Manfred K. Warmuth. Learning permutations with exponential weights. In *COLT*, 2007. 1, 2.2, 4.3

J. Huang, C. Guestrin, and L. Guibas. Efficient inference for distributions on permutations. In *NIPS 2007*, 2007. 1.2, 5.1.1, 5.2, 5.3, 5.4, 6

J. Huang, C. Guestrin, and L. Guibas. Inference for distributions over the permutation group. Technical Report CMU-ML-08-108, Machine Learning Department, Carnegie Mellon University, May 2008. 1.2, 3, 3.1, 4.2, 4.2, 3, 7.1

Srikanth Jagabathula and Devavrat Shah. Inferring rankings under constrained sensing. In *NIPS 2008*, 2008. 7.1

Thorsten Joachims. Optimizing search engines using clickthrough data. In *KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 133–142, New York, NY, USA, 2002. ACM. 6

Zia Khan, Tucker Balch, and Frank Dellaert. Mcmc data association and sparse factorization updating for real time multitarget tracking with merged and multiple measurements. *IEEE PAMI*, 28(12), 2006. 5.3

R. Kondor. $\mathbb{S}_n$`ob`: a C++ library for fast Fourier transforms on the symmetric group, 2006. Available at `http://www.cs.columbia.edu/~risi/Snob/`. 3

R. Kondor, A. Howard, and T. Jebara. Multi-object tracking with representations of the symmetric group. In *AISTATS*, 2007. 1.1, 5.2, 5.3, 5.4, 6, 6.2, 7.1

Risi Kondor. The skew spectrum of functions on finite groups and their homogeneous spaces, 2007. 1.1

Risi Kondor. *group theoretical methods in machine learning*. PhD thesis, Columbia University, 2008. 1.1

Risi Kondor and Karsten M. Borgwardt. The skew spectrum of graphs. In *ICML*, pages 496–503, 2008. 1.1

Risi Kondor and John Lafferty. Diffusion kernels on graphs and other discrete structures. In *ICML*, 2002. 1.1

K.L. Kueh, T. Olson, D. Rockmore, and K.S. Tan. Nonlinear approximation theory on finite groups. Technical Report PMA-TR99-191, Department of Mathematics, Dartmouth College, 1999. 1.1

Serge Lang. *Algebra*. Addison-Wesley, 1965. 2

G. Lebanon and J. Lafferty. Cranking: Combining rankings using conditional probability models on permutations. In *Proceedings International Conference on Machine Learning*, 2002. 1

G. Lebanon and Y. Mao. Non-parametric modeling of partially ranked data. In *NIPS 2007*, 2007. 1, 6, 6

Marius Leordeanu and Martial Hebert. A spectral technique for correspondence problems using pairwise constraints. In *International Conference of Computer Vision (ICCV)*, volume 2, pages 1482 – 1489, October 2005. 1

David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60:91–110, 2004. 1.1

C.L. Mallows. Non-null ranking models. *Biometrika*, 44, 1957. 1, 6

Eric Malm. Decimation-in-frequency fast fourier transforms for the symmetric group. Harvey Mudd undergraduate thesis, 2005. 1.1, 4.2

Jerrold Marsden and Michael Hoffman. *Elementary Classical Analysis*. W.H. Freeman, 1993. 3

David Maslen. The efficient computation of fourier transforms on the symmetric group. *Mathematics of Computation*, 67:1121–1147, 1998. 1.1, 4.2

Marina Meila, Kapil Phadnis, Arthur Patterson, and Jeff Bilmes. Consensus ranking under the exponential model. Technical Report 515, University of Washington, Statistics Department, April 2007. 6, 6

J. Munkres. Algorithms for the assignment and transportation problems. *Journal of the Society of Industrial and Applied Mathematics*, 5(1):32–38, 1957. 7.1

F.D. Murnaghan. The analysis of the kronecker product of irreducible representations of the symmetric group. *American Journal of Mathematics*, 60(3):761–784, 1938. 4.2

K.G. Murty. An algorithm for ranking all the assignments in order of increasing cost. *Operations Research*, 16:682–687, 1968. 1, 6.1

H. Narayanan. On the complexity of computing kostka numbers and littlewood-richardson coefficients. *J. Algebraic Comb.*, 24(3):347–354, 2006. 5.1

Songhwai Oh and Shankar Sastry. A polynomial-time approximation algorithm for joint probabilistic data association. In *Proc. of the American Control Conference, Portland, OR*, 2005. 1, 6.1

Songhwai Oh, Stuart Russell, and Shankar Sastry. Markov chain Monte Carlo data association for general multiple-target tracking problems. In *Proc. of the IEEE International Conference on Decision and Control, Paradise Island, Bahamas*, 2004. 1, 6.1

A.B. Poore. Multidimensional assignment and multitarget tracking. In *Partitioning Data Sets*, volume 19, pages 169–196. DIMACS Series in Discrete Mathematics and Theoretical Computer Science, 1995. 1, 6.1

Lawrence Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989. 4.1

D.B. Reid. An algorithm for tracking multiple targets. *IEEE Trans. on Automatic Control*, 6:843–854, 1979. 1, 6.1

Daniel N. Rockmore. The fft: An algorithm the whole family can use. *Computing in Science and Engineering*, 02(1):60–64, 2000. 6.2

B. Sagan. *The Symmetric Group*. Springer, April 2001. ISBN 0387950672. 3, 5.1

B. Schumitsch, S. Thrun, G. Bradski, and K. Olukotun. The information-form data association filter. In *NIPS*. 2006a. 1, 1.1, 6.1

B. Schumitsch, S. Thrun, L. Guibas, and K. Olukotun. The identity management Kalman filter (imkf). In *Proceedings of Robotics: Science and Systems*, Philadelphia, PA, USA, August 2006b. 1, 1.1, 6.1

J. Shin, L. Guibas, and F. Zhao. A distributed algorithm for managing multi-target identities in wireless ad-hoc sensor networks. In *IPSN*, 2003. 1, 1.1, 2.2, 4.3, 6.1

Jaewon Shin, Nelson Lee, Sebastian Thrun, and Leonidas Guibas. Lazy inference on object identities in wireless sensor networks. In *IPSN '05*, 2005. 1, 1.1, 4.3, 6.1

Josephine Sullivan and Stefan Carlsson. Tracking and labelling of interacting multiple targets. In *ECCV (3)*, pages 619–632, 2006. 5.2

Michael Taylor, John Guiver, Stephen Robertson, and Tom Minka. Softrank: optimizing non-smooth rank metrics. In *WSDM '08: Proceedings of the international conference on Web search and web data mining*, pages 77–86, New York, NY, USA, 2008. ACM. 1, 6

Audrey Terras. *Fourier Analysis on Finite Groups and Applications*. London Mathematical Society, 1999. 1.1, 3

J. van Lint and R.M. Wilson. *A Course in Combinatorics*. Cambridge University Press, 2001. 4.3

A. Willsky. On the algebraic structure of certain partially observable finite-state markov processes. *Information and Control*, 38:179–212, 1978. 1.1, 4.2, 4.2, 6.2

Ron Zass and Amnon Shashua. Probabilistic graph and hypergraph matching. In *Computer Vision and Pattern Recognition (CVPR)*, June 2008. 1

Hongyuan Zha, Xiaofeng He, Chris Ding, Horst Simon, and Ming Gu. Bipartite graph partitioning and data clustering. In *CIKM '01*, pages 25–32, New York, NY, USA, 2001. ACM. 5.1.1