Riffled Independence for Ranked Data

(extended version with proofs)

Jonathan Huang, Carlos Guestrin
School of Computer Science,
Carnegie Mellon University
Pittsburgh, PA 15213
{jch1,guestrin}@cs.cmu.edu

Abstract

Representing distributions over permutations can be a daunting task due to the fact that the number of permutations of n objects scales factorially in n. One recent way that has been used to reduce storage complexity has been to exploit probabilistic independence, but as we argue, full independence assumptions impose strong sparsity constraints on distributions and are unsuitable for modeling rankings. We identify a novel class of independence structures, called *riffled independence*, encompassing a more expressive family of distributions while retaining many of the properties necessary for performing efficient inference and reducing sample complexity. In riffled independence, one draws two permutations independently, then performs the *riffle shuffle*, common in card games, to combine the two permutations to form a single permutation. In ranking, riffled independence corresponds to ranking disjoint sets of objects independently, then interleaving those rankings. We provide a formal introduction and present algorithms for using riffled independence within Fourier-theoretic frameworks which have been explored by a number of recent papers.

1 Introduction

Distributions over permutations play a central role in a variety of applications such as multi-object tracking, visual feature matching, and ranking. In people tracking, for example, permutations represent joint assignments of individual identities to track positions, whereas in ranking, permutations represent the preference orderings of a list of objects. Representing distributions over permutations is a notoriously difficult problem since there are $n!$ permutations, and standard representations, such as graphical models, are not effective due to the mutual exclusivity constraints typically associated with permutations. The quest for exploitable problem structure has led researchers to consider a number of possibilities including distribution sparsity [17, 9], exponential family parameterizations [15, 5, 14, 16], algebraic/Fourier structure [13, 12, 6, 7], and probabilistic independence [8].

While sparse distributions have been successfully applied in certain tracking domains, we argue that they are less suitable in ranking problems where it might be necessary to model indifference over a large subset of objects. In contrast, Fourier-based methods handle smooth distributions well but are not easily scalable without making aggressive independence assumptions [8]. In this paper, we argue that while probabilistic independence might be useful for tracking applications, it is a poor approximation in ranking applications. We propose a novel generalization of independence, called *riffled independence*, which we believe to be far more suitable for modeling distributions over preference rankings, and develop algorithms for working with riffled independence in the Fourier domain. The following is a summary of the major contributions of our paper.

- We introduce an intuitive generalization of independence on permutations, which we call *riffled independence*, and show it to be a more appropriate notion of independence for ranked data.
- We show how to exploit riffled independence in a distribution to reduce sample complexity and to perform efficient inference.
- We introduce a novel family of distributions, called *biased riffle shuffles*, that are useful for riffled independence and propose an algorithm for computing its Fourier transform.

- We provide algorithms that can be used in the Fourier-theoretic framework of [13, 8, 7] for joining riffle independent factors (*RiffleJoin*), and for teasing apart the riffle independent factors from a joint (*RiffleSplit*), and provide theoretical and empirical evidence that our algorithms perform well.

2 Distributions on permutations and independence relations

In the context of ranking, a permutation $\sigma = [\sigma_1, \ldots, \sigma_n]$ represents a one-to-one mapping from n objects to n ranks, where, by $\sigma_j = i$ (or $\sigma(j) = i$), we mean that the j^{th} object is assigned rank i under σ. If we are ranking a list of fruits/vegetables enumerated as (1) Artichoke, (2) Broccoli, (3) Cherry, and (4) Dates, then the permutation $\sigma = [\sigma_A\, \sigma_B\, \sigma_C\, \sigma_D] = [2\,3\,1\,4]$ ranks Cherry first, Artichoke second, Broccoli third, and Dates last. The set of all permutations of $\{1, \ldots, n\}$ forms a group with respect to function composition called the *symmetric group* (written S_n). We will write $\tau\sigma$ to denote the permutation resulting from τ composed with σ (thus $[\tau\sigma](j) = \tau(\sigma(j))$). A distribution $h(\sigma)$, defined over S_n, can be viewed as a joint distribution over the n variables $\sigma = (\sigma_1, \ldots, \sigma_n)$ (where $\sigma_j \in \{1, \ldots, n\}$), subject to *mutual exclusivity constraints* which ensure that objects i and j do not map to the same rank ($h(\sigma_i = \sigma_j) = 0$ whenever $i \neq j$). Since there are $n!$ permutations, it is typically intractable to represent entire distributions and one can hope only to maintain compact summary statistics.

There have been a variety of methods proposed for summarizing distributions over permutations ranging from older ad-hoc methods such as maintaining k-best hypotheses [17] to the more recent Fourier-based methods which maintain a set of low-order summary statistics [19, 2, 11, 7]. The *first-order summary*, for example, stores a marginal probability of the form $h(\sigma : \sigma(j) = i)$ for every pair (i, j) and thus requires storing a matrix of only $O(n^2)$ numbers. For example, we might store the probability that apples are ranked first. More generally, one might store the *s^{th}-order marginals*, which are marginal probabilities of s-tuples. The second-order marginals, for example, take the form $h(\sigma : \sigma(k, \ell) = (i, j))$, and require $O(n^4)$ storage. Low-order marginals correspond, in a certain sense, to the low-frequency Fourier coefficients of a distribution over permutations. For example, the first-order matrix of $h(\sigma)$ can be reconstructed exactly from $O(n^2)$ of the lowest frequency Fourier coefficients of $h(\sigma)$, and the second-order matrix from $O(n^4)$ of the lowest frequency Fourier coefficients. In general, one requires $O(n^{2s})$ coefficients to exactly reconstruct s^{th}-order marginals, which quickly becomes intractable for moderately large n. To scale to larger problems, Huang et al. [8] demonstrated that, by exploiting *probabilistic independence*, one could dramatically improve the scalability of Fourier-based methods, e.g., for tracking problems, since confusion in data association only occurs over small independent subgroups of objects in many problems.

Probabilistic independence on permutations. Probabilistic independence assumptions on the symmetric group can simply be stated as follows. Consider a distribution h defined over S_n. Let X be a p-subset of $\{1, \ldots, n\}$, say, $\{1, \ldots, p\}$ and let $\bar{X}$ be its complement ($\{p+1, \ldots, n\}$) with size $q = n - p$. We say that $\sigma_X = (\sigma_1, \sigma_2, \ldots, \sigma_p)$ and $\sigma_{\bar{X}} = (\sigma_{p+1}, \ldots, \sigma_n)$ are *independent* if

$$h(\sigma) = f(\sigma_1, \sigma_2, \ldots, \sigma_p) \cdot g(\sigma_{p+1}, \ldots, \sigma_n).$$

Storing the parameters for the above distribution requires keeping $O(p! + q!)$ probabilities instead of the much larger $O(n!)$ size required for general distributions. Of course, $O(p! + q!)$ can still be quite large. Typically, one decomposes the distribution recursively and stores factors exactly for small enough factors, or compresses factors using Fourier coefficients (but using higher frequency terms than what would be possible without the independence assumption). In order to exploit probabilistic independence in the Fourier domain, Huang et al. [8] proposed algorithms for joining factors and splitting distributions into independent components in the Fourier domain.

Restrictive first-order conditions. Despite its utility for many tracking problems, however, we argue that the independence assumption on permutations implies a rather restrictive constraint on distributions, rendering independence highly unrealistic in ranking applications. In particular, using the mutual exclusivity property, it can be shown [8] that, if σ_X and $\sigma_{\bar{X}}$ are independent, then for some fixed p-subset $Y \subset \{1, \ldots, n\}$, σ_X is a permutation of elements in Y and $\sigma_{\bar{X}}$ is a permutation of its complement, $\bar{Y}$, with probability 1. Continuing with our vegetable/fruit example with $n = 4$, if the vegetable and fruit rankings, $\sigma_{veg} = [\sigma_A\, \sigma_B]$ and $\sigma_{fruit} = [\sigma_C\, \sigma_D]$, are known to be independent, then for $Y = \{1, 2\}$, the vegetables are ranked first and second with probability one, and the fruits are ranked third and last with probability one. Huang et al. [8] refer to this as the *first-order condition* because of the block structure imposed upon first-order marginals (see Fig. 1). In sports tracking, the first-order condition might say, quite reasonably, that there is potential identity confusion within tracks for the red team and within tracks for the blue team but no confusion between the

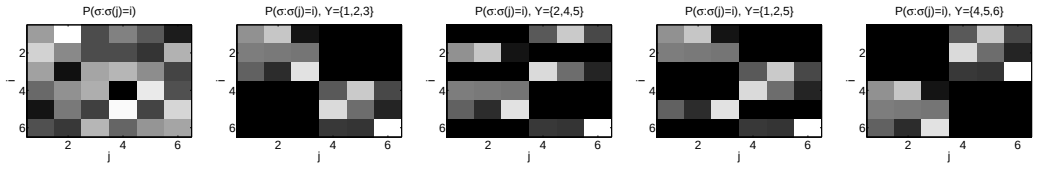


Figure 1: Example first-order matrices with $X = \{1, 2, 3\}$, $\bar{X} = \{4, 5, 6\}$ fully independent, where black means $h(\sigma : \sigma(j) = i) = 0$. In each case, there is some 3-subset Y which X is constrained to map to with probability one. Notice that, with respect to some rearranging of the rows, independence imposes a block-diagonal structure on first-order matrices.

two teams. In our ranking example however, the first-order condition forces the probability of any vegetable being in third place to be zero, even though both vegetables will, in general, have nonzero marginal probability of being in second place, which seems quite unrealistic. In the next section, we overcome the restrictive first-order condition with the more flexible notion of *riffled independence*.

3 Going beyond full independence: Riffled independence

The *riffle (or dovetail) shuffle* [1] is perhaps the most commonly used method of card shuffling, in which one cuts a deck of n cards into two piles, $X = \{1, \ldots, p\}$ and $\bar{X} = \{p + 1, \ldots, n\}$, with size p and $q = n - p$, respectively, and successively drops the cards, one by one, so that the two piles become interleaved (see Fig. 2) into a single deck again. Inspired by the riffle shuffle, we now present a novel relaxation of the full independence assumption, which we call *riffled independence*. Rankings that are riffle independent are formed by independently selecting rankings for two disjoint subsets of objects, then interleaving the two rankings using a riffle shuffle to form a final ranking over all objects. For example, we might first 'cut the deck' into two piles, vegetables (X) and fruits ($\bar{X}$), independently decide that Broccoli is preferred over Artichoke ($\sigma_B < \sigma_A$) and that Dates is preferred over Cherry ($\sigma_D < \sigma_C$), then finally interleave the fruit and vegetable rankings to form $\sigma_B < \sigma_D < \sigma_A < \sigma_C$ (i.e. $\sigma = [3\,1\,4\,2]$). Intuitively, riffled independence models complex relationships within each set X and $\bar{X}$ while allowing correlations between the sets to be modeled only through a constrained form of shuffling.

Figure 2: Riffle shuffling a standard deck of cards.

Riffle shuffling distributions. Mathematically, shuffles are modeled as random walks on S_n. The ranking σ' *after* a shuffle is generated from the ranking *prior* to that shuffle, σ, by drawing a permutation, τ from a *shuffling distribution* $m(\tau)$, and setting $\sigma' = \tau\sigma$. Given the distribution P over σ, we can find the distribution $h'(\sigma')$ after the shuffle via *convolution*: $h'(\sigma') = [m * h](\sigma') = \sum_{\{\sigma,\tau\,:\,\sigma'=\tau\sigma\}} m(\tau)h(\sigma)$. Note that we use the $*$ symbol to denote the convolution operation.

Besides the riffle shuffle, there are a number of different kinds of shuffles — the pairwise shuffle, for example, simply selects two cards at random and swaps them. The question then, is *what are the shuffling distributions m that correspond to riffle shuffles?* To answer this question, we use the distinguishing property of the riffle shuffle, that, after cutting the deck into two piles of size p and $q = n - p$, it must preserve the relative ranking relations within each pile. Thus, if the i^{th} card appears above the j^{th} card in one of the piles, then after shuffling, the i^{th} card *remains* above the j^{th} card. In our example, relative rank preservation says that if Artichoke is preferred over Broccoli prior to shuffling, it continues to be preferred over Broccoli after shuffling. Any allowable riffle shuffling distribution must therefore assign zero probability to permutations which do not preserve relative ranking relations. As it turns out, the set of permutations which *do* preserve these relations have a simple description.

Definition 1 (Riffle shuffling distribution)**.** Define the set of (p, q)*-interleavings* as:

$$\Omega_{p,q} \equiv \{\tau_Y = [Y_{(1)}\, Y_{(2)} \ldots Y_{(p)}\, \bar{Y}_{(1)}\, \bar{Y}_{(2)} \ldots \bar{Y}_{(q)}] \,:\, Y \subset \{1, \ldots, n\},\, |Y| = p\} \subset S_n,\; n = p + q,$$

where $Y_{(1)}$ represents the smallest element of Y, $Y_{(2)}$ the second smallest, etc. A distribution $m_{p,q}$ on S_n is called a *riffle shuffling distribution* if it assigns nonzero probability *only* to elements in $\Omega_{p,q}$.

The (p,q)-interleavings can be shown to preserve the relative ranking relations within each of the subsets $X = \{1,\dots,p\}$ and $\bar{X} = \{p+1,\dots,n\}$ upon multiplication.

Lemma 2. *Let $i,j \in X = \{1,\dots,p\}$ and let τ_Y be any (p,q)-interleaving in $\Omega_{p,q}$. Then $i<j$* if and only if $\tau_Y(i) < \tau_Y(j)$ *(i.e., permutations in $\Omega_{p,q}$ preserve relative ranking relations).*

Proof. By definition of τ_Y, we have that $\tau_Y(i) = Y_{(i)}$, and $\tau_Y(j) = Y_{(j)}$, where $Y_{(i)}$ is the i^{th} smallest element in the set Y, and $Y_{(j)}$ is the j^{th} smallest element in Y. We have that:

$$i < j \iff Y_{(i)} < Y_{(j)} \iff \tau_Y(i) < \tau_Y(j).$$

□

In our vegetable/fruits example, we have $n=4$, $p=2$, and so the collection of subsets of size p are: $\{\ \{1,2\},\{1,3\},\{1,4\},\{2,3\},\{2,4\},\{3,4\}\ \}$, and the set of $(2,2)$-interleavings is given by: $\Omega_{p,q} = \{[1\,2\,3\,4],[1\,3\,2\,4],[1\,4\,2\,3],[2\,3\,1\,4],[2\,4\,1\,3],[3\,4\,1\,2]\}$. Note that the number of possible interleavings is $|\Omega_{p,q}| = \binom{n}{p} = \binom{n}{q} = 4!/(2!2!) = 6$. One possible riffle shuffling distribution on S_4 might, for example, assign uniform probability ($m^{unif}_{2,2}(\sigma) = 1/6$) to each permutation in $\Omega_{2,2}$ and zero probability to everything else, reflecting indifference between vegetables and fruits. We now formally define our generalization of independence where a distribution which fully factors independently is allowed to undergo a single riffle shuffle.

Definition 3 (Riffled independence)**.** The subsets $X = \{1,\dots,p\}$ and $\bar{X} = \{p+1,\dots,n\}$ are said to be *riffle independent* if $h = m_{p,q} * (f(\sigma_p)\cdot g(\sigma_q))$, with respect to some riffle shuffling distribution $m_{p,q}$ and distributions f,g, respectively. We denote riffled independence by: $h = f \perp_{m_{p,q}} g$, and refer to f,g as *riffled factors*.

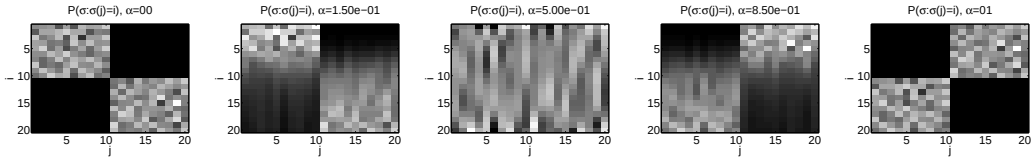
To draw from h, one independently draws a permutation σ_p, of cards $\{1,\dots,p\}$, a permutation σ_q, of cards $\{p+1,\dots,n\}$, and a (p,q)-interleaving τ_Y, then shuffles to obtain $\sigma = \tau_Y[\sigma_p\,\sigma_q]$. In our example, the rankings $\sigma_p = [2\,1]$ (Broccoli preferred to Artichoke) and $\sigma_q = [4\,3]$ (Cherry preferred to Dates) are selected, then shuffled (multiplied by $\tau_{\{1,3\}} = [1\,3\,2\,4]$) to obtain $\sigma = [3\,1\,4\,2]$.

We remark that setting $m_{p,q}$ to be the delta distribution on *any* of the (p,q)-interleavings in $\Omega_{p,q}$ recovers the definition of ordinary probabilistic independence, and thus riffled independence is a strict generalization thereof. Just as in the full independence regime, where the distributions f and g are marginal distributions of rankings of X and $\bar{X}$, in the riffled independence regime, they can be thought of as marginal distributions of the *relative rankings* of X and $\bar{X}$.

Biased riffle shuffles. There is, in the general case, a significant increase in storage required for riffled independence over full independence. In addition to the $O(p!+q!)$ storage required for distributions f and g, we now require $O(\binom{n}{p})$ storage for the nonzero terms of the riffle shuffling distribution $m_{p,q}$. Instead of representing all possible riffle shuffling distributions, however, we now introduce a family of useful riffle shuffling distributions which can be described using only a handful of parameters. The simplest riffle shuffling distribution is the *uniform riffle shuffle*, $m^{unif}_{p,q}$, which assigns uniform probability to all (p,q)-interleavings and zero probability to all other elements in S_n. Used in the context of riffled independence, $m^{unif}_{p,q}$ models potentially complex relations within X and $\bar{X}$, but only captures the simplest possible correlations across subsets. We might, for example, have complex preference relations amongst vegetables and amongst fruits, but be completely indifferent with respect to the subsets, vegetables and fruits, as a whole.

1 DRAWRIFFLEUNIF(p,q,n) // $(p+q=n)$
2 **with prob** q/n // drop from right pile
3 $\sigma^- \leftarrow$ DRAWRIFFLEUNIF$(p,q-1,n-1)$
4 **foreach** *i* **do** $\sigma(i) \leftarrow \begin{cases} \sigma^-(i) & \text{if } i<n \\ n & \text{if } i=n \end{cases}$
5 **otherwise** // drop from left pile
6 $\sigma^- \leftarrow$ DRAWRIFFLEUNIF$(p-1,q,n-1)$
7 **foreach** *i* **do** $\sigma(i) \leftarrow \begin{cases} \sigma^-(i) & \text{if } i<p \\ n & \text{if } i=p \\ \sigma^-(i-1) & \text{if } i>p \end{cases}$
8 **return** σ

Algorithm 1: Recurrence for drawing $\sigma \sim m^{unif}_{p,q}$ (Base case: return $\sigma=[1]$ if $n=1$).



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

Figure 3: First-order matrices with a deck of 20 cards, $X = \{1, \ldots, 10\}$, $\bar{X} = \{11, \ldots, 20\}$, *riffle independent* and various settings of α. Note that nonzero blocks 'bleed' into zero regions (compare to Fig. 1). Setting $\alpha = 0$ or 1 recovers full independence, where a subset of objects is preferred over the other with probability one.

There is a simple recursive method for uniformly drawing (p, q)-interleavings. Starting with a deck of n cards cut into a left pile ($\{1, \ldots, p\}$) and a right pile ($\{p + 1, \ldots, n\}$), pick one of the piles with probability proportional to its size (p/n for the left pile, q/n for the right) and drop the bottommost card, thus mapping either card p or card n to rank n. Then recurse on the $n - 1$ remaining undropped cards, drawing a $(p - 1, q)$-interleaving if the right pile was picked, or a $(p, q - 1)$-interleaving if the left pile was picked. See Alg. 1.

It is natural to consider generalizations where one is preferentially biased towards dropping cards from the left hand over the right hand (or vice-versa). We model this bias using a simple one-parameter family of distributions in which cards from the left and right piles drop with probability proportional to αp and $(1 - \alpha)q$, respectively, instead of p and q. We will refer to α as the *bias parameter*, and the family of distributions parameterized by α as the *biased riffle shuffles*.[1] In the context of rankings, biased riffle shuffles provide a simple model for expressing groupwise preferences (or indifference) for an entire subset X over $\bar{X}$ or vice-versa. The bias parameter α can be thought of as a knob controlling the preference for one subset over the other, and might reflect, for example, a preference for fruits over vegetables, or perhaps indifference between the two subsets. Setting $\alpha = 0$ or 1 recovers the full independence assumption, preferring objects in X (vegetables) over objects in $\bar{X}$ (fruits) with probability one (or vice-versa), and setting $\alpha = .5$, recovers the uniform riffle shuffle (see Fig. 3). Finally, there are a number of straightforward generalizations of the biased riffle shuffle that one can use to realize richer distributions. For example, α might depend on the number of cards that have been dropped from each pile (allowing perhaps, for distributions to prefer crunchy *fruits* over crunchy *vegetables*, but soft *vegetables* over soft *fruits*).

4 Between independence and conditional independence

We have presented riffle independent distributions as fully independent distributions which have been convolved by a certain class of shuffling distributions. In this section, we provide an alternative view of riffled independence based on conditional independence, showing that the notion of riffled independence lies somewhere between full and conditional independence, and in a sense, can be thought of as full independence without the first-order condition.

In Section 3, we formed a ranking by first independently drawing permutations π_p and π_q, of object sets $\{1, \ldots, p\}$ (vegetables) and $\{p + 1, \ldots, n\}$ (fruits), respectively, drawing a (p, q)-interleaving (i.e., a relative ranking permutation, $\tau_Y \in \Omega_{p,q}$), and shuffling to form $\sigma = \tau_Y[\pi_p\ \pi_q]$. Thus, an object $i \in \{1, \ldots, p\}$ is ranked in position $\tau_Y(\pi_p(i))$ after shuffling (and an object $j \in \{p+1, \ldots, n\}$ is ranked in position $\tau_Y(\pi_q(j))$). An equivalent way to form the same σ, however, is to *first* draw an interleaving $\tau_Y \in \Omega_{p,q}$, then, conditioned on the choice of Y, draw independent permutations of the sets Y and $\bar{Y}$. In our example, we might first draw the (2,2)-interleaving $[1\,3\,2\,4]$ (so that after shuffling, we would obtain $\sigma_{Veg} < \sigma_{Fruit} < \sigma_{Veg} < \sigma_{Fruit}$). Then we would draw a permutation of the vegetable ranks ($Y = \{1, 3\}$), say, $[3\,1]$, and a permutation of the fruit ranks ($\bar{Y} = \{2, 4\}$), $[4\,2]$, to obtain a final ranking over all items: $\sigma = [3\,1\,4\,2]$, or $\sigma_B < \sigma_D < \sigma_A < \sigma_C$.

It is tempting to think that riffled independence is *exactly* the conditional independence assumption, in which case the distribution would factor as $h(\sigma) = h(Y) \cdot h(\sigma_X|Y) \cdot h(\sigma_{\bar{X}}|Y)$. The general case of conditional independence, however, has $O(\binom{n}{p}(p! + q! + 1))$ parameters, while riffled independence requires only $O(\binom{n}{p} + p! + q!)$ parameters.

[1] The recurrence in Alg. 1 has appeared in various forms in literature [1]. We are the first to (1) use the recurrence to Fourier transform $m_{p,q}$, and to (2) consider biased versions. The biased riffle shuffles in [4] are not similar to our biased riffle shuffles. See Appendix A for details.

We now provide a simple correspondence between the conditional independence view of riffled independence presented in this section to the shuffle theoretic definition from Section 3 (Def. 3). Define the map ϕ, which, given a permutation of Y (or $\bar{Y}$), returns the permutation in $\sigma_p \in S_p$ (or S_q) such that $[\sigma_p]_i$ is the rank of $[\sigma_X]_i$ *relative* to the set Y. For example, if the permutation of the vegetable ranks is $\sigma_X = [3\,1]$ (with Artichoke ranked third, Broccoli first), then $\phi(\sigma_X) = [2\,1]$ since, relative to the set of vegetables, Artichoke is ranked second, and Broccoli first.

Proposition 4. *Consider a riffle independent $h = f \perp_{m_{p,q}} g$. For each $\sigma \in S_n$, h factors as $h(\sigma) = h(Y) \cdot h(\sigma_X|Y) \cdot h(\sigma_{\bar{X}}|Y)$, with $h(Y) = m(\tau_Y)$, $h(\sigma_X|Y) = f(\phi(\sigma_X))$, and $h(\sigma_{\bar{X}}) = g(\phi(\sigma_{\bar{X}}))$.*

Proposition 4 is useful because it shows that the probability of a single ranking can be computed without summing over the entire symmetric group (a convolution)— a fact that might not be obvious from Definition 3. The factorization $h(\sigma) = m(\tau_Y) f(\phi(\sigma_X)) g(\phi(\sigma_{\bar{X}}))$ also suggests that riffled independence behaves essentially like full independence (without the first-order condition), where, in addition to the independent variables σ_X and $\sigma_{\bar{X}}$, we also independently randomize over the subset Y. To prove Proposition 4, we first establish a simple fact.

Lemma 5. *Let $\sigma = [\sigma_X\ \sigma_{\bar{X}}]$ be any permutation in S_n and Y the set consisting of elements in σ_X. Define $\pi_p = \phi(\sigma_X)$, $\pi_q = \phi(\sigma_{\bar{X}})$, and $\tau_Y = [sort(Y)\ sort(\bar{Y})]$. Then $\sigma = \tau_Y[\pi_p\ (\pi_q + p)]$.*

Proof. Let $1 \leq i \leq p$. We want to show that $\sigma(i) = \tau_Y(\pi_p(i))$. Since $\pi_p \in S_p$, $\pi_p(i)$ is some number between 1 and p, and thus $\tau_Y(\pi_p(i)) = [\text{sort}(Y)](\pi_p(i))$. By definition of the function ϕ, $\pi_p(i)$ is the rank of i relative to the set Y, and so $[\text{sort}(Y)](\pi_p(i))$ is the *absolute rank* (e.g., the rank relative to the entire set, $\{1, \ldots, n\}$) of i assigned by σ. Therefore, we conclude that $\sigma(i) = \tau_Y(\pi_p(i))$. The proof for $p + 1 \leq i \leq n$ is similar. □

Example 6. *Let $\sigma = [4\,6\,2\,8\,7\,5\,3\,1] \in S_n$ (hence, $\sigma_X = [4\,6\,2\,8]$, $\sigma_{\bar{X}} = [7\,5\,3\,1]$). The set Y is $\{4, 6, 2, 8\}$, $\pi_p = \phi([4\,6\,2\,8]) = [2\,3\,1\,4]$, and $\pi_q = \phi([7\,5\,3\,1]) = [4\,3\,2\,1]$. Finally, we set $\tau_Y = [2\,4\,6\,8\,1\,3\,5\,7]$. Lemma 5 then tells us that:*

$$\begin{aligned}
\sigma &= \tau_Y[\pi_p\ (\pi_q + p)] \\
&= [2\,4\,6\,8\,1\,3\,5\,7][2\,3\,1\,4\ (4+p)\ (3+p)\ (2+p)\ (1+p)] \\
&= [2\,4\,6\,8\,1\,3\,5\,7][2\,3\,1\,4\,8\,7\,6\,5] \\
&= [4\,6\,2\,8\,7\,5\,3\,1]
\end{aligned}$$

Proof (of Proposition 4). Let $X = \{1, \ldots, p\}$ and let $S_p \times S_q$ denote the subgroup: $\{\sigma : \sigma(X) \subset X\}$. The function $f \cdot g$ (fully-independent) is supported on $S_p \times S_q$. Assume that $h = m_{p,q} * (f \cdot g)$.

Let $\sigma = [\sigma_X\ \sigma_{\bar{X}}]$ be any permutation in S_n and let Y be the p-subset corresponding to the values in σ_X. Notice that $\tau_Y^{-1}\sigma$ is always in $S_p \times S_q$, and $\tau_{Y'}^{-1}\sigma$ lies outside of $S_p \times S_q$ for any $Y' \neq Y$.

$$\begin{aligned}
[m_{p,q} * (f \cdot g)](\sigma) &= \sum_{\sigma' \in S_n} m_{p,q}(\sigma') \cdot [f \cdot g](\sigma'^{-1}\sigma), \\
&= \sum_{\tau \in \Omega_{p,q}} m_{p,q}(\tau) \cdot [f \cdot g](\tau^{-1}\sigma), \\
&\quad \text{(since } m_{p,q} \text{ is supported on } \Omega_{p,q}\text{)}, \\
&= m_{p,q}(\tau_Y) \cdot [f \cdot g](\tau_Y^{-1}\sigma), \\
&\quad \text{(since } f \cdot g \text{ is supported on } S_p \times S_q\text{)}, \\
&= m_{p,q}(\tau_Y) \cdot [f \cdot g]([\pi_p\ (\pi_q + p)]), \\
&\quad \text{(by Lemma 5)}, \\
&= m_{p,q}(\tau_Y) \cdot f(\pi_p) \cdot g(\pi_q), \\
&= m_{p,q}(\tau_Y) \cdot f(\phi(\sigma_X)) \cdot g(\phi(\sigma_{\bar{X}})).
\end{aligned}$$

□

One immediate consequence is that we can show, just as in the full independence regime, that conditioning operations on certain observations and MAP (maximum a posteriori) assignment problems decompose according to riffled independence structure.

Proposition 7 (Probabilistic inference decompositions)**.** *Consider riffle independent prior and likelihood functions, h_{prior} and h_{like}, on S_n which factor as: $h_{prior} = f_{prior} \perp_{m_{prior}} g_{prior}$ and $h_{like} = f_{like} \perp_{m_{like}} g_{like}$, respectively. The posterior distribution under Bayes rule can be written as the riffle independent distribution: $h_{post} \propto (f_{prior} \odot f_{like}) \perp_{m_{prior} \odot m_{like}} (g_{prior} \odot g_{like})$,where the $\odot$ symbol denotes the pointwise product operation.*

Proof (of Proposition 7). For any element $\sigma \in S_n$, we have:

$$\begin{aligned} h_{post}(\sigma) &\propto (m_{like}(\tau_Y) \cdot f_{like}(\phi(\sigma_X)) \cdot g_{like}(\phi(\sigma_{\bar{X}}))) \\ &\quad \cdot (m_{prior}(\tau_Y) \cdot f_{prior}(\phi(\sigma_X)) \cdot g_{prior}(\phi(\sigma_{\bar{X}}))), \\ &\propto (m_{like}(\tau_Y) \cdot m_{prior}(\tau_Y)) \\ &\quad \cdot (f_{like}(\phi(\sigma_X)) \cdot f_{prior}(\phi(\sigma_X))) \\ &\quad \cdot (g_{like}(\phi(\sigma_{\bar{X}})) \cdot g_{prior}(\phi(\sigma_{\bar{X}}))), \\ &\propto (f_{prior} \odot f_{like}) \perp_{m_{like} \odot m_{prior}} (g_{prior} \odot g_{like}). \end{aligned}$$

□

A similar result allows us to also perform MAP (*maximum a posteriori*) assignments by maximizing each of the distributions $m_{p,q}$, f and g, independently and combining the results.

Proposition 8. *Consider a distribution h which factors as $h = f \perp_m g$. Let $\pi_p = \arg\max_\pi f(\pi)$, $\pi_q = \arg\max_\pi g(\pi)$, and $\tau_Y = \arg\max_{\tau_Y} m(\tau_Y)$. We have:* $\arg\max_\sigma h(\sigma) = \tau_Y[\pi_p\,(\pi_q + p)]$.

Pairwise rankings. Some ranked datasets come in the form of pairwise comparisons, with records of the form "object i is preferred to object j". The pairwise ranking model [7] (for objects i and j), is defined over S_n as:

$$g_{like}(\sigma) = \delta^n_{\sigma(i)<\sigma(j)}(\sigma) = \begin{cases} \beta & \text{if } \sigma(i) < \sigma(j) \\ 1-\beta & \text{otherwise} \end{cases}, \quad 0 \le \beta \le 1,$$

and reflects the fact that object i is preferred to object j (with probability β). If objects i and j both belong to one of the sets, say $\bar{X}$, then we can show, as a corollary of Proposition 7, that only one factor requires an update using Bayes rule. If vegetables and fruits are riffle independent, for example, then less computation would be required to compare a fruit against a fruit than to compare a fruit against a vegetable.

Corollary 9. *Conditioning on the pairwise ranking model can be performed on S_q instead of all of S_n using the update: $h_{post} \leftarrow f_{prior} \perp_{m_{p,q}} \left(g_{prior} \odot \delta^q_{\sigma(i)<\sigma(j)}\right)$.*

Before proving Corollary 9, we first prove another corollary of Lemma 2.

Corollary 10 (of Lemma 2)**.** *Let $\sigma = [\sigma_X\ \sigma_{\bar{X}}]$, $i, j \in X$, and $\tau_Y = [sort(Y)\ sort(\bar{Y})]$. Then we have: $\sigma(i) < \sigma(j)$* if and only if $\tau_Y^{-1}(\sigma(i)) < \tau_Y^{-1}(\sigma(j))$.

Proof. First notice that $\sigma(i) < \sigma(j)$ if and only if $\tau_Y\tau_Y^{-1}\sigma(i) < \tau_Y\tau_Y^{-1}\sigma(j)$. Since $i, j \in X$, $\sigma(i), \sigma(j) \in Y$, and since all elements in Y are in the first p ranks in the permutation τ, we have that both elements $\tau_Y^{-1}(\sigma(i))$ and $\tau_Y^{-1}(\sigma(j))$ are in the set $\{1, \ldots, p\}$. Applying Lemma 2, we see that $\tau_Y^{-1}\sigma(i) < \tau_Y^{-1}\sigma(j)$. The other direction is similar. □

Proof (of Corollary 9). This proof will rely on Proposition 4. For simplicity, we will actually assume that $i, j \in \{1, \ldots, p\}$ for the proof. It is enough, by Proposition 7,to show that the indicator function $\delta^n_{\sigma(i)<\sigma(j)}$, factors into riffle independent components over S_n. Let $m^{unif}_{p,q}$ be the uniform riffle shuffing distribution (which assigns uniform probability to the relative ranking permutations, and zero probability to all other permutation), let $f : S_p \to \mathbb{R}$ be $\delta^p_{\sigma(i)<\sigma(j)}$, the pairwise ranking likelihood on S_p, and let $g : S_q \to \mathbb{R}$ be the uniform distribution on S_q.

For any $\sigma \in S_n$, write (using Lemma 5) $\sigma = \tau_Y[\pi_p\ (\pi_q + p)]$. We have:

$$\begin{aligned}
\delta^n_{\sigma(i)<\sigma(j)}(\sigma) &= \delta^n_{\sigma(i)<\sigma(j)}(\tau_Y^{-1}\sigma), \\
&\qquad \text{(by Corollary 10)}, \\
&= \delta^n_{\sigma(i)<\sigma(j)}([\pi_p\ (\pi_q + p)]), \\
&= \delta^p_{\sigma(i)<\sigma(j)}(\pi_p), \\
&\qquad \text{(since } \delta^n_{\sigma(i)<\sigma(j)}([\pi_p\ (\pi_q + p)] \text{ only depends on } \pi_p), \\
&\propto m^{unif}_{p,q}(\tau_Y) \cdot f(\pi_p) \cdot g(\pi_q), \quad \text{(by Proposition 4)}, \\
&\propto f \perp_{m^{unif}_{p,q}} g.
\end{aligned}$$

Then use Proposition 7 to conclude the result. □

5 Fourier domain algorithms: *RiffleJoin* and *RiffleSplit*

In this section, we present two algorithms for working with riffled independence in the Fourier theoretic framework of [13, 8, 7] — one algorithm for merging riffled factors to form a joint distribution (*RiffleJoin*), and one for extracting riffled factors from a joint (*RiffleSplit*). We begin with a brief introduction to Fourier theoretic inference on permutations (see [11, 7] for a detailed exposition). Unlike its analog on the real line, the Fourier transform of a function on S_n takes the form of a collection of Fourier coefficient *matrices* ordered with respect to frequency. Discussing the analog of frequency for functions on S_n, is beyond the scope of our paper, and, given a distribution h, we simply index the Fourier coefficient matrices of h as $\widehat{h}_0$, $\widehat{h}_1$, ..., $\widehat{h}_K$ ordered with respect to some measure of increasing complexity. We use $\widehat{h}$ to denote the complete collection of Fourier coefficient matrices. One rough way to understand this complexity, as mentioned in Section 2, is by the fact that the low-frequency Fourier coefficient matrices of a distribution can be used to reconstruct low-order marginals. For example, the first-order matrix of marginals of h can always be reconstructed from the matrices $\hat{h}_0$ and $\hat{h}_1$. As on the real line, many of the familiar properties of the Fourier transform continue to hold. The following are several basic properties used in this paper:

Proposition 11 (Properties of the Fourier transform, see [2])**.** *Consider any $f, g : S_n \to \mathbb{R}$.*

- *(Linearity) For any $\alpha, \beta \in \mathbb{R}$, $[\widehat{\alpha f + \beta g}]_i = \alpha \widehat{f}_i + \beta \widehat{g}_i$ holds at all frequency levels i.*
- *(Convolution) The Fourier transform of a convolution is a product of Fourier transforms: $[\widehat{f * g}]_i = \widehat{f}_i \cdot \widehat{g}_i$, for each frequency level i, where the operation $\cdot$ is matrix multiplication.*
- *(Normalization) The first coefficient matrix, $\hat{f}_0$, is a scalar and equals $\sum_{\sigma \in S_n} f(\sigma)$.*

A number of papers in recent years ([13, 6, 8, 7]) have considered approximating distributions over permutations using a truncated (bandlimited) set of Fourier coefficients and have proposed inference algorithms that operate on these Fourier coefficient matrices. For example, one can perform generic marginalization, Markov chain prediction, and conditioning operations using only Fourier coefficients without ever having to perform an inverse Fourier transform. Additionally, Huang et al. [8] introduced two Fourier domain algorithms, *Join* and *Split*, for combining independent factors to form joint distributions and for extracting the factors from a joint distribution, respectively.

In this section, we provide generalizations of the algorithms in [8] that we call *RiffleJoin* and *RiffleSplit*. We will assume that $X = \{1, \ldots, p\}$, $\bar{X} = \{p+1, \ldots, n\}$ and that we are given a riffle independent distribution $h : S_n \to \mathbb{R}$ ($h = f \perp_{m_{p,q}} g$). We also, for the purposes of this section, assume that the parameters for the distribution $m_{p,q}$ are known, though it will not matter for the *RiffleSplit* algorithm. Although we begin each of the following discussions as if all of the Fourier coefficients are provided, we will be especially interested in algorithms that work well in cases where only a truncated set of Fourier coefficients are present, and where h is only *approximately* riffle independent.

***RiffleJoin*.** Given the Fourier coefficients of f, g, and m, we can compute the Fourier coefficients of h using Definition 3 by applying the Join algorithm from [8] and the *Convolution Theorem* (Prop. 11), which tells us that the Fourier transform of a convolution can be written as a pointwise product of Fourier transforms. To compute the $\hat{h}_\lambda$, our *RiffleJoin* algorithm simply calls the Join algorithm on $\widehat{f}$ and $\widehat{g}$, and convolves the result by $\widehat{m}$ (see Alg. 2). In general, it may be intractable to Fourier transform the riffle shuffling distribution $m_{p,q}$. However, for the class of biased riffle

RIFFLEJOIN($\widehat{f}, \widehat{g}$)
$\widehat{h'} = \text{JOIN}(\widehat{f}, \widehat{g})$;
foreach *frequency level* i **do**
    $\widehat{h}_i \leftarrow \left[\widehat{m^{\alpha}_{p,q}}\right]_i \cdot \widehat{h'}_i$;
return $\widehat{h}$;

Algorithm 2: Pseudocode for *RiffleJoin*

RIFFLESPLIT($\widehat{h}$)
foreach *frequency level* i **do**
    $\widehat{h'}_i \leftarrow \left[\widehat{m}^{unif}_{p,q}\right]^T_i \cdot \widehat{h}_i$;
$[\widehat{f}, \widehat{g}] \leftarrow \text{SPLIT}(\widehat{h'})$;
Normalize $\hat{f}$ and $\hat{g}$;
return $\hat{f}, \hat{g}$;

Algorithm 3: Pseudocode for *RiffleSplit*

shuffles from Section 3, one can efficiently compute the low-frequency terms of $\widehat{m^{\alpha}_{p,q}}$ by employing the recurrence relation in Alg. 1. In particular, Alg. 1 expresses a biased riffle shuffle on S_n as a linear combination of biased riffle shuffles on S_{n-1}. By invoking linearity of the Fourier transform (Prop. 11), one can efficiently compute $\widehat{m^{\alpha}_{p,q}}$ via a dynamic programming approach. See Appendix A for a description of our algorithm for Fourier transforming biased riffle shuffles. To the best of our knowledge, we are the first to compute the Fourier transform of riffle shuffling distributions.

***RiffleSplit*.** Given the Fourier coefficients of the riffle independent distribution h, we would like to tease apart the riffle factors f and g. From the RiffleJoin algorithm, we saw that for each frequency level i, $\hat{h}_i = [\widehat{m_{p,q}}]_i \cdot [\widehat{f \cdot g}]_i$. The first solution to the splitting problem that might occur is to perform a deconvolution by multiplying each $\widehat{h}_i$ term by the inverse of the matrix $[\widehat{m_{p,q}}]_i$ (to form $[\widehat{m_{p,q}}]_i^{-1} \cdot \widehat{h}_i$) and call the Split algorithm from [8] on the result. Unfortunately, the matrix $[\widehat{m_{p,q}}]_i$ is, in general, non-invertible. Instead, our RiffleSplit algorithm left-multiplies each $\widehat{h}_i$ term by $\left[\widehat{m}^{unif}_{p,q}\right]^T_i$, which can be shown to be equivalent to convolving the distribution h by the '*dual shuffle*', m^*, defined as $m^*(\sigma) = m^{unif}_{p,q}(\sigma^{-1})$. While convolving by m^* does not produce a distribution that factors independently, the Split algorithm from [8] can still be shown to recover the Fourier transforms $\hat{f}$ and $\hat{g}$:

Theorem 12. *If $h = f \perp_{m_{p,q}} g$, then RiffleSplit (Alg. 3) (with $\hat{h}$ as input), returns $\hat{f}$ and $\hat{g}$* exactly.

Proof (of Theorem 12). This follows from the more general result in Theorem 14 whose proof can be found in Appendix B. □

As with RiffleJoin, it is necessary to compute the Fourier coefficients of $m^{unif}_{p,q}$, which we can again accomplish via the recurrence in Alg. 1. It is also necessary to normalize the output of Split to sum to one, but fortunately, normalizing a function f can be performed in the Fourier domain simply by dividing each Fourier coefficient matrix by $\widehat{f}_0$ (Prop. 11).

Theoretical guarantees. We now briefly summarize several results which show how, (1) our algorithms perform when called with a truncated set of Fourier coefficients, and (2) when RiffleSplit is called on a distribution which is only *approximately* riffle independent.

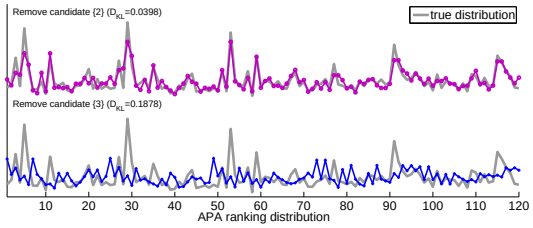
Theorem 13. *Given enough Fourier terms to reconstruct the k^{th}-order marginals of f and g, RiffleJoin returns enough Fourier terms to* exactly *reconstruct the k^{th}-order marginals of h. Likewise, given enough Fourier terms to reconstruct the k^{th}-order marginals of h, RiffleSplit returns enough Fourier terms to* exactly *reconstruct the k^{th}-order marginals of both f and g.*

Proof (of Theorem 13). This result is a simple consequence of the well-known convolution theorem 11 and Theorems 9 and 12 from Huang et al. [8]. Theorem 9 from [8] states that, given s^{th}-order marginals of factors f and g, the Join algorithm can reconstruct the s^{th}-order marginals of the joint distribution $f \cdot g$, *exactly*. Since the riffled independence joint is $m * (f \cdot g)$ and convolution operations are pointwise in the Fourier domain (Proposition 11), then given enough Fourier terms to reconstruct the s^{th}-order marginals of the function $m^{p,q}$, we can also reconstruct the s^{th}-order marginals of the riffle independent joint distribution from the output of RiffleSplit. □

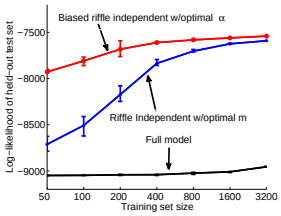
Theorem 14. *Let h be* any *distribution on S_n and $m_{p,q}$ any riffle shuffling distribution on S_n. If $[\widehat{f'}, \widehat{g'}] = \text{RIFFLESPLIT}(\widehat{h})$, then (f', g') is the minimizer of the problem:*

$$\textit{minimize}_{f,g} \;\; D_{KL}(h || f \perp_{m_{p,q}} g), \;\; (\textit{subject to:} \textstyle\sum_{\sigma_p} f(\sigma_p) = 1, \sum_{\sigma_q} g(\sigma_q) = 1),$$

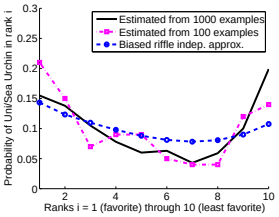
where D_{KL} is the Kullback-Leibler divergence.
The proof of Theorem 14 is deferred to Appendix B.



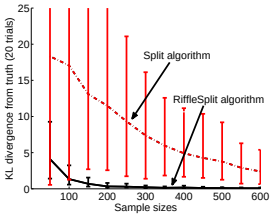
(a) Purple line: approximation to vote distribution when candidate 2 is riffle independent; Blue line: approximation when candidate 3 is riffle independent.



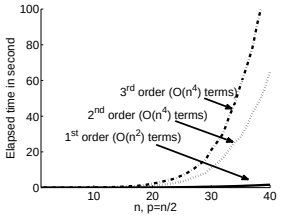
(b) Average log-likelihood of held out test examples from the Sushi dataset



(c) First-order probabilities of Uni (sea urchin) (Sushi dataset) rankings.



(d) Estimating a riffle independent distribution using various sample sizes



(e) Running time plot of RiffleJoin

Figure 4: Experiments

6 Experiments

In this section, we discuss several experiments demonstrating riffled independence in real data and validating our Fourier-domain algorithms.

APA dataset. The APA dataset [3] is a collection of 5738 ballots from a 1980 presidential election of the American Psychological Association where members ordered five candidates from favorite to least favorite. We first perform an exhaustive search for subsets X and $\bar{X}$ that are closest to riffle independent (with respect to D_{KL}), and find that candidate 2 is nearly riffle independent of the remaining candidates. In Fig. 4(a) we plot the true vote distribution and the best approximation by a distribution in which candidate 2 is riffle independent of the rest. For comparison, we plot the result of splitting off candidate 3 instead of candidate 2, which one can see to be an inferior approximation.

The APA, as described by Diaconis [3], is divided into "*academicians and clinicians who are on uneasy terms*". In 1980, candidates $\{1, 3\}$ and $\{4, 5\}$ fell on opposite ends of this political spectrum with candidate 2 being somewhat independent. Diaconis conjectured that voters choose one group over the other, and then choose within. We are now able to verify his conjecture in a riffled independence sense. After removing candidate 2 from the distribution, we perform a search within candidates $\{1, 3, 4, 5\}$ to again find *nearly* riffle independent subsets. We find that $X = \{1, 3\}$ and $\bar{X} = \{4, 5\}$ are very nearly riffle independent and thus are able to verify that candidate sets $\{2\}$, $\{1, 3\}$, $\{4, 5\}$ are indeed grouped in a riffle independent sense in the APA data. Finally since there are two opposing groups within the APA, the riffle shuffling distribution for sets $\{1, 3\}$ and $\{4, 5\}$ is not well approximated by a biased riffle shuffle. Instead, we fit a mixture of two biased riffle shuffles to the data and found the bias parameters of the mixture components to be $\alpha_1 \approx .67$ and $\alpha_2 \approx .17$, indicating that the two components oppose each other (since α_1 and α_2 lie on either side of .5).

Sushi dataset. The sushi dataset [10] consists of 5000 full rankings of ten types of sushi. Compared to the APA data, it has more objects, but fewer examples. We divided the data into training and test sets and estimated the true distribution in three ways: (1) directly from samples,(2) using a riffle independent distribution (split evenly into two groups of five) with the optimal shuffling distribution m, and (3) with a biased riffle shuffle (and optimal bias α). Fig. 4(b) plots testset log-likelihood as a function of training set size — we see that riffle independence assumptions can help significantly to lower the sample complexity of learning. Biased riffle shuffles, as can be seen, are a useful learning bias with very small samples. As an illustration, see Fig. 4(c) which shows the first-order marginals of Uni (Sea Urchin) rankings, and the biased riffle approximation.

Approximation accuracy. To understand the behavior of RiffleSplit in approximately riffle independent situations, we draw sample sets of varying sizes from a riffle independent distribution

on S_8 (with bias parameter $\alpha = .25$) and use RiffleSplit to estimate the riffle factors from the empirical distribution. In Fig. 4(d), we plot the KL-divergence between the true distribution and that obtained by applying RiffleJoin to the estimated riffle factors. With small sample sizes (far less than $8!$), we are able to recover accurate approximations despite the fact that the empirical distributions are not exactly riffle independent. For comparison, we ran the experiment using the Split algorithm [8] to recover the riffle factors. Somewhat surprisingly, one can show that Split also recovers the riffle factors, albeit without the optimality guarantee that we have shown for Rifflesplit (Theorem 14) and therefore requires far more samples to reliably approximate h.

Running times. In general, the complexity of Split is cubic ($O(d^3)$) in the dimension of each Fourier coefficient matrix [8]. The complexity of RiffleJoin/RiffleSplit is $O(n^2d^3)$, in the worst case when $p \sim O(n)$. If we precompute the Fourier coefficients of $m_{p,q}$, (which requires $O(n^2d^3)$) for each coefficient matrix, then the complexity of RiffleSplit is also $O(d^3)$. In Fig. 4(e), we plot running times of RiffleJoin (no precomputation) as a function of n (setting $p = \lceil n/2 \rceil$) scaling up to $n = 40$.

7 Future Directions and Conclusions

There are many open questions. For example, several papers note that graphical models cannot compactly represent distributions over permutations due to mutual exclusivity. An interesting question which our paper opens, is whether it is possible to use something similar to graphical models by substituting conditional generalizations of riffled independence for ordinary conditional independence. Other possibilities include going beyond the algebraic approach and studying riffled independence in non-Fourier frameworks and developing statistical (riffled) independence tests.

In summary, we have introduced riffled independence and discussed how to exploit such structure in a Fourier-theoretic framework. Riffled independence is a new tool for analyzing ranked data and has the potential to offer novel insights into datasets both new and old. We believe that it will lead to the development of fast inference and low sample complexity learning algorithms.

Acknowledgements
This work is supported in part by the ONR under MURI N000140710747 as well as the Young Investigator Program grant N00014-08-1-0752. We also thank Khalid El-Arini for providing valuable feedback on an initial draft of the paper.

References

[1] D. Bayer and P. Diaconis. Trailing the dovetail shuffle to its lair. *The Annals of Probability*, 1992.

[2] P. Diaconis. *Group Representations in Probability and Statistics*. IMS Lecture Notes, 1988.

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

[4] J. Fulman. The combinatorics of biased riffle shuffles. *Combinatorica*, 18(2):173–184, 1998.

[5] D. P. Helmbold and M. K. Warmuth. Learning permutations with exponential weights. In *COLT*, 2007.

[6] J. Huang, C. Guestrin, and L. Guibas. Efficient inference for distributions on permutations. In *NIPS*, 2007.

[7] J. Huang, C. Guestrin, and L. Guibas. Fourier theoretic probabilistic inference over permutations. *JMLR*, 10, 2009.

[8] J. Huang, C. Guestrin, X. Jiang, and L. Guibas. Exploiting probabilistic independence for permutations. In *AISTATS*, 2009.

[9] S. Jagabathula and D. Shah. Inferring rankings under constrained sensing. In *NIPS*, 2008.

[10] T. Kamishima. Nantonac collaborative filtering: recommendation based on order responses. In *KDD*, pages 583–588, 2003.

[11] R. Kondor. *Group Theoretical Methods in Machine Learning*. PhD thesis, Columbia University, 2008.

[12] R. Kondor and K. M. Borgwardt. The skew spectrum of graphs. In *ICML*, pages 496–503, 2008.

[13] R. Kondor, A. Howard, and T. Jebara. Multi-object tracking with representations of the symmetric group. In *AISTATS*, 2007.

[14] G. Lebanon and Y. Mao. Non-parametric modeling of partially ranked data. In *NIPS*, 2008.

```
RIFFLEHAT(p, q)
n ← p + q ;
Initialize m̂^prev, m̂^curr as arrays of p + 1 Fourier transform data structures ;
for i = 1, 2, ..., n do
    for j = max(0, p − n + i), ..., min(i, p) do
        if j == 0 or j == i then
            m̂^curr[j] ← δ̂_{ϵ∈S_i} ;
        else
            m̂^curr[j] ← ((i−j)/i) EMBED(m̂^prev[j], i − 1, i)
                        + (i/j) CONVOLVE(EMBED(m̂^prev[j − 1], i − 1, i), δ̂_{(i,i−1,...,j)});
        m̂^prev ← m̂^curr ;
return m̂^curr[p];
```

Algorithm 4: Pseudocode for computing the Fourier transform of the uniform riffle shuffling distribution using dynamic programming.

[15] M. Meila, K. Phadnis, A. Patterson, and J. Bilmes. Consensus ranking under the exponential model. Technical Report 515, University of Washington, Statistics Department, April 2007.

[16] J. Petterson, T. Caetano, J. McAuley, and J. Yu. Exponential family graph matching and ranking. *CoRR*, abs/0904.2623, 2009.

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

[18] B. Sagan. *The Symmetric Group*. Springer, 2001.

[19] J. Shin, N. Lee, S. Thrun, and L. Guibas. Lazy inference on object identities in wireless sensor networks. In *IPSN*, 2005.

Appendix A: Fourier transforming the biased riffle shuffle

We describe the recurrence satisfied by $m_{p,q}^{unif}$, allowing one to write $m_{p,q}^{unif}$, a distribution on S_n, in terms of $m_{p,q-1}^{unif}$ and $m_{p-1,q}^{unif}$, distributions over S_{n-1} (see Alg. 1 in main paper). Given a function $f : S_{n-1} \to \mathbb{R}$, we will define the *embedded* function $f \uparrow_{n-1}^{n}$: $S_n \to \mathbb{R}$ by $f \uparrow_{n-1}^{n} (\sigma) = f(\sigma_1, \ldots, \sigma_{n-1})$ if $\sigma(n) = n$, and 0 otherwise. Algorithm 1 can be then rephrased as a legitimate recurrence relation as follows.

Proposition 15. *The uniform riffle shuffling distribution $m_{p,q}^{unif}$ obeys the recurrence relation:*

$$m_{p,q}^{unif} = \left[\left(\frac{p}{p+q} \right) \cdot m_{p-1,q}^{unif} \uparrow_{n-1}^{n} * \delta_{(p+1,\ldots,n)} \right] + \left[\left(\frac{q}{p+q} \right) \cdot m_{p,q-1}^{unif} \uparrow_{n-1}^{n} \right], \tag{7.1}$$

with base cases: $m_{0,n}^{unif} = m_{n,0}^{unif} = \delta_\epsilon$, where δ_ϵ is the delta function at the identity permutation.

Note that by taking the support sizes of each of the functions in the above recurrence, we recover the following well known recurrence for binomial coefficients:

$$\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}, \text{ with base case } \binom{n}{0} = \binom{n}{n} = 1. \tag{7.2}$$

The *biased riffle shuffle* is defined by:

$$m_{p,q}^{\alpha} \propto \left[\left(\frac{\alpha p}{p+q} \right) \cdot m_{p-1,q}^{unif} \uparrow_{n-1}^{n} * \delta_{(p+1,\ldots,n)} \right] + \left[\left(\frac{(1-\alpha)q}{p+q} \right) \cdot m_{p,q-1}^{unif} \uparrow_{n-1}^{n} \right] \tag{7.3}$$

Writing the recursion in the form of Equation 7.1 provides a construction of the uniform riffle shuffle as a sequence of operations on smaller distribution which can be performed completely with respect to Fourier coefficients. In particular, given the Fourier coefficients of a function $f : S_{n-1} \to \mathbb{R}$, one can construct the Fourier coefficients of the embedding $f \uparrow_{n-1}^{n}$ by applying the branching rule (see [18, 12] for details). Using the linearity property, the Convolution Theorem 11 and the fact that

embeddings can be performed in the Fourier domain, we arrive at the equivalent Fourier-theoretic recurrence for each frequency level i.

$$\left[\widehat{m_{p,q}^{unif}}\right]_i = \left(\frac{p}{p+q}\right) \cdot \left[\widehat{m_{p-1,q}^{unif}} \uparrow_{n-1}^{n}\right]_i \cdot \rho_i(p+1,\ldots,n) + \left(\frac{q}{p+q}\right) \cdot \left[\widehat{m_{p,q-1}^{unif}}\right]_i \tag{7.4}$$

where ρ_i is the i^{th} irreducible representation matrix evaluated at the cycle $(p+1,\ldots,n)$ (see [7] for details on irreducible representations). Implementing the recurrence (Equation 7.4) in code can naively result in an exponential time algorithm if one is not careful. It is necessary to use dynamic programming to be sure not to recompute things that were already computed. In Algorithm 4, we present pseudocode of such a dynamic programming approach, which builds a 'Pascal's triangle' similar to that which might be constructed to compute a table of binomial coefficients. The pseudocode assumes the existence of Fourier domain algorithms for convolving distributions and for embedding a distribution over S_{n-1} into S_n.

Appendix B: Proof of Theorem 14

Proof (of Theorem 14). Fix any $m_{p,q}$. Given an arbitrary distribution $h : S_n \to \mathbb{R}$, we want to find distributions f, and g, such that $\tilde{h} = f \perp_{m_{p,q}} g$ minimizes $D_{KL}\left(h \mid\mid \tilde{h}\right)$. Thus, we want to minimize the functional: $\sum_{\sigma \in S_n} h(\sigma) \log\left(\frac{h(\sigma)}{\tilde{h}(\sigma)}\right)$, or equivalently, maximize:

$$\begin{aligned}\sum_{\sigma \in S_n} h(\sigma) \log \tilde{h}(\sigma) &= \sum_{\sigma \in S_n} h(\sigma) \log(m_{p,q}(\tau_{Y(\sigma)}) \cdot f(\phi(\sigma_X)) \cdot g(\phi(\sigma_{\bar{X}}))), \\ &= \sum_{\sigma \in S_n} h(\sigma) \log m_{p,q}(\tau_{Y(\sigma)}) + \sum_{\sigma \in S_n} h(\sigma) \log f(\phi(\sigma_X)) + \sum_{\sigma \in S_n} h(\sigma) \log g(\phi(\sigma_{\bar{X}})),\end{aligned}$$

subject to the constraints that f and g, sum to one. Forming the Lagrangian, we have:

$$\begin{aligned}\mathcal{L}(f,g,\mu,\nu) = \sum_{\sigma} h(\sigma) \log f(\phi(\sigma_X)) + \mu(1 - \sum_{\pi_p \in S_p} f(\pi_p)) \\ + \sum_{\sigma} h(\sigma) \log g(\phi(\sigma_{\bar{X}})) + \nu(1 - \sum_{\pi_q \in S_q} g(\pi_q)) + \text{constants}.\end{aligned}$$

Differentiating the Lagrangian (and setting the partials to zero), we have:

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial f(\pi_p)} &= \sum_{\sigma \,:\, \phi(\sigma_X) = \pi_p} \frac{h(\sigma)}{f(\pi_p)} + \mu = 0, \\ \frac{\partial \mathcal{L}}{\partial g(\pi_q)} &= \sum_{\sigma \,:\, \phi(\sigma_{\bar{X}}) = \pi_q} \frac{h(\sigma)}{g(\pi_q)} + \nu = 0.\end{aligned}$$

Therefore,

$$f(\pi_p) \propto \sum_{\{\sigma \,:\, \phi(\sigma_X) = \pi_p\}} h(\sigma), \;\text{ and }\; g(\pi_q) \propto \sum_{\{\sigma \,:\, \phi(\sigma_{\bar{X}}) = \pi_q\}} h(\sigma).$$

By Lemma 5, we can write the factors f and g as:

$$\begin{aligned}f(\pi_p) &\propto \sum_{\pi_q \in S_q} \sum_{\tau_Y \in \Omega_{p,q}} h(\tau_Y[\pi_p\,(\pi_q + p)]), \text{ and} \\ g(\pi_q) &\propto \sum_{\pi_p \in S_p} \sum_{\tau_Y \in \Omega_{p,q}} h(\tau_Y[\pi_p\,(\pi_q + p)]).\end{aligned}$$

To compute the functions f and g, we can therefore first form the function: $h'(\sigma) = \sum_{\tau_Y \in \Omega_{p,q}} h(\tau_Y \sigma)$, and call the Split algorithm from [8] on h' (which depends only on elements in $S_p \times S_q$). Let $m_{p,q}^{unif}$ be the uniform riffle shuffling distribution on S_n which assigns uniform

probability to all relative ranking permutations in $\Omega_{p,q}$, and zero probability to all other elements in S_n. Define the '*dual shuffle*' $m^*_{p,q}$ by: $m^*_{p,q}(\sigma) = m^{unif}_{p,q}(\sigma^{-1})$ for all $\sigma \in S_n$. We can rewrite the function $h' : S_n \to \mathbb{R}$ as a convolution of the dual-shuffle with h:

$$\begin{aligned} h'(\sigma) &\propto \sum_{\pi \in S_n} m^{unif}_{p,q}(\pi) h(\pi\sigma), \\ &\propto \sum_{\pi \in S_n} m^*_{p,q}(\pi) h(\pi^{-1}\sigma), \\ &\propto [m^*_{p,q} * h](\sigma). \end{aligned}$$

Applying the convolution theorem, we have that $\widehat{h'}_{\lambda_i} = [\widehat{m^*_{p,q}}]_{\lambda_i} \cdot \widehat{h}_{\lambda_i}$. The only thing that is left is to show that Fourier coefficient matrices of the uniform riffle shuffle are exactly the same as the Fourier coefficient matrices of its dual shuffle, except transposed. Let i denote any frequency level and let ρ_i be its corresponding irreducible representation for S_n. We will assume, for the purposes of this proof, that all irreducible representation matrices are orthogonal.

$$\begin{aligned} [\widehat{m^{unif}_{p,q}}]_i &= \sum_{\sigma \in S_n} m^{unif}_{p,q}(\sigma) \rho_i(\sigma), \\ &= \sum_{\sigma \in S_n} m^{unif}_{p,q}(\sigma^{-1}) \rho_i(\sigma^{-1}), \\ &= \sum_{\sigma \in S_n} m^*_{p,q}(\sigma) \rho_i(\sigma)^T, \\ &= \left([\widehat{m^*_{p,q}}]_i \right)^T. \end{aligned}$$

and the Split algorithm from [8] yields the finally RiffleSplit algorithm. □