Electronic Journal of Statistics
Vol. 6 (2012) 199–230
ISSN: 1935-7524
DOI: 10.1214/12-EJS670

Uncovering the riffled independence structure of ranked data

Jonathan Huang

Stanford University, Stanford California, 94305
e-mail: jhuang11@stanford.edu

and

Carlos Guestrin

Carnegie Mellon University, Pittsburgh Pennsylvania, 15213
e-mail: 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. Within the context of ranking, riffled independence corresponds to ranking disjoint sets of objects independently, then interleaving those rankings. In this paper, we provide a formal introduction to riffled independence and propose an automated method for discovering sets of items which are riffle independent from a training set of rankings. We show that our clustering-like algorithms can be used to discover meaningful latent coalitions from real preference ranking datasets and to learn the structure of hierarchically decomposable models based on riffled independence.

AMS 2000 subject classifications: Primary 68T37, 60C05; secondary 60B15.
Keywords and phrases: Riffled independence, permutations, rankings, structure learning, group theoretic methods, probabilistic modeling.

Received July 2010.

Contents

1. Introduction

Ranked data appears ubiquitously in various statistics and machine learning application domains. Rankings are useful, for example, in reasoning about preference lists in surveys [21], search results in information retrieval applications [34], and ballots in certain elections [7] and even the ordering of topics and paragraphs within a document [4]. As with many challenging learning problems, one must contend with an intractably large state space when dealing with rankings since there are $n!$ ways to rank n objects. In building a statistical model over rankings, simple (yet flexible) models are therefore preferable because they are typically more computationally tractable and less prone to overfitting.

A popular and highly successful approach for achieving such simplicity for distributions involving large collections of interdependent variables has been to exploit conditional independence structures (e.g., naive Bayes, graphical models). With ranking problems, independence-based relations are harder to exploit due to the *mutual exclusivity* constraints which constrain any two items to map to different ranks in a given ranking.

In this paper, we present a novel, relaxed notion of independence, called *riffled independence*, in which one ranks disjoint subsets of items independently, then interleaves the subset rankings to form a joint ranking of the item set. Riffled independence appears naturally in many ranked datasets — as we show, political coalitions in elections often lead to pronounced riffled independence constraints in the vote histograms.

The following is a roadmap of our main contributions:[1]

[1] This paper is an extended presentation of our previous papers [15], which was the first introduction of riffled independence, and [17], which studied hierarchical models based on riffle independent decompositions.

- Section 2 gives a broad overview of several approaches for modeling probability distributions over permutations. In particular, we summarize the results of [18], which studied probabilistic independence relations in distributions on permutations.
- In Section 3, we introduce our main contribution: an intuitive generalization of the notion of independence for permutations, *riffled independence*, based on interleaving independent rankings of subsets of items. We show riffled independence to be a more appropriate notion of independence for ranked data and exhibit evidence that riffle independence relations can approximately hold in real ranked datasets. We also discuss the problem of estimating parameters of a riffle independent model from ranking data.
- We use Section 4 to define a family of simple and interpretable distributions over rankings, called hierarchical riffle independent models, in which subsets of items are iteratively interleaved into larger and larger subsets in a recursive stagewise fashion.
- Section 5 tackles the problem of structure learning for our riffle independent models. We propose a method for partitioning the item set so that the subsets of the partition are as close to riffle independent as possible. and apply our method to perform model selection from training data.

For the sake of brevity, proofs of theoretical results are omitted in this paper. We refer interested readers to the supplementary material [19].

2. Distributions on rankings

A *ranking* $\sigma = (\sigma(1), \ldots, \sigma(n))$ is a one-to-one association between n items and ranks, where $\sigma(j) = i$ means that the j^{th} item is assigned rank i under σ. By convention, we will think of low ranked items as being *preferred* over higher ranked items (thus, ranking an item in first place means that it is the most preferred out of all items). We will also refer to a ranking σ by its inverse, $\llbracket \sigma^{-1}(1), \ldots, \sigma^{-1}(n) \rrbracket$ (called an *ordering* and denoted with double brackets instead of parentheses), where $\sigma^{-1}(i) = j$ also means that the j^{th} item is assigned rank i under σ. The reason for using both notations is due to the fact that certain concepts will be more intuitive to express using either the ranking or ordering notation.

Example 1. As a running example, we will consider ranking a list of 6 items consisting of fruits and vegetables enumerated below:

1. *Corn* (**C**)
2. *Peas* (**P**)
3. *Lemon* (**L**)
4. *Orange* (**O**)
5. *Fig* (**F**)
6. *Grapes* (**G**)

The ranking $\sigma = (3, 1, 5, 6, 2, 4)$ means, for example, that Corn is ranked third, Peas is ranked first, Lemon is ranked fifth, and so on. In ordering notation, the same ranking is expressed as: $\sigma = \llbracket \mathbf{P}, \mathbf{F}, \mathbf{C}, \mathbf{G}, \mathbf{L}, \mathbf{O} \rrbracket$. Finally we will use $\sigma(3) = \sigma(\mathbf{L}) = 5$ to denote the rank of the third item, Lemon.

Permutations and the symmetric group Rankings are similar to *permutations*, which are 1-1 mappings from the set $\{1, \ldots, n\}$ into itself, the subtle difference being that rankings map between two *different* sets of size n. In this paper, we will use the same notation for permutations and rankings, but use permutations to refer to (1-1) functions which rearrange the ordering of the item set or the ranks. If τ is a permutation of the set of ranks, then given a ranking σ, one can rearrange the ranks by left-composing with τ. Thus, the ranking $\tau\sigma$ maps item i to rank $\tau(\sigma(i))$. On the other hand, if τ is a permutation of the item set, one can rearrange the item set by right-composing with τ^{-1}. Thus, if item j was relabeled as item $i = \tau(j)$, then $\sigma(\tau^{-1}(i))$ returns the rank of item j with respect to the original item ordering. Finally, we note that the composition of any two permutations is itself a permutation, and the collection of all $n!$ permutations forms a group, commonly known as the *symmetric group*, or S_n.

We consider a random variable $\boldsymbol{\sigma}$ taking values in S_n with probability mass function $h(\boldsymbol{\sigma})$.[2] The distribution corresponding to $h(\boldsymbol{\sigma})$ can also be viewed as a joint distribution over the n variables $(\boldsymbol{\sigma}(1), \ldots, \boldsymbol{\sigma}(n))$ (where $\boldsymbol{\sigma}(j) \in \{1, \ldots, n\}$), subject to *mutual exclusivity constraints* which stipulate that two objects cannot simultaneously map to the same rank, or alternatively, that two ranks cannot simultaneously be occupied by the same object ($h(\boldsymbol{\sigma}(i) = \boldsymbol{\sigma}(j)) = 0$ whenever $i \neq j$).

Example 2 (APA election data)**.** As another running example, we analyze the well known APA election dataset that was first used by [6] and has since been studied in many works. The dataset is a collection of 5738 ballots from a 1980 presidential election of the American Psychological Association where members rank ordered candidates by preference. The names of the five candidates that year were (1) William Bevan, (2) Ira Iscoe, (3) Charles Kiesler, (4) Max Siegle, and (5) Logan Wright [25].

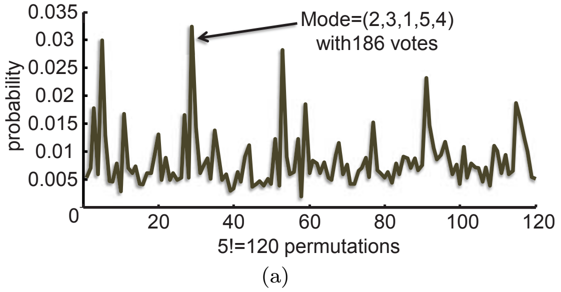
Since there are five candidates, there are $5! = 120$ rankings, and in Figure 1(a) we plot the proportion of votes that each ranking received. Instead of concentrating at just a small set of rankings, the vote distribution in the APA dataset is diffuse with every ranking receiving some number of votes.

For interpretability, we also visualize the matrix of first-order marginals in which the (i, j) entry represents the number of voters who assigned rank i to candidate j. Figure 1(b) represents the first-order matrix using grayscale levels to represent numbers of voters. What can be seen is that overall, candidate 3 (C. Kiesler) received the highest number of votes for rank 1 (and incidentally, won the election). The vote distribution gives us a story that goes far deeper than simply telling us who the winner was, however. [6], for example, noticed that candidate 3 also had a "hate" vote — a good number of voters placed him in the last rank. We will let this story further unfold via a series of examples.

2.1. Dealing with factorial possibilities

The fact that there are factorially many rankings poses significant challenges for learning and inference. First, there is no way to tractably represent arbitrary

[2] In this paper, we use boldface $\boldsymbol{\sigma}$ to refer to the random variable and σ to refer to realizations of the random variable.



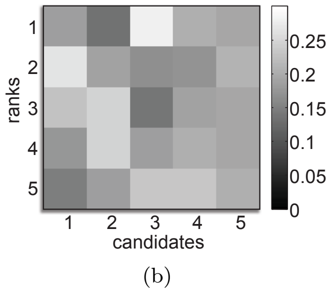


FIG 1. *APA (American Psyochological Association) election data. (a) vote distribution: percentage of votes for each of* $5! = 120$ *possible rankings — the mode of the distribution is* $\sigma = (2,3,1,5,4)$. *(b) Matrix of first order marginals: the* $(i,j)^{th}$ *entry reflects the number of voters who ranked candidate* j *in the* i^{th} *rank.*

distributions over rankings for large n. Second, the naive algorithmic complexity of common probabilistic operations is also intractable for such distributions. Computing the marginal probability, $h(\boldsymbol{\sigma}(i) < \boldsymbol{\sigma}(j))$, that item i is preferred to item j, for example, requires a summation over $O((n-2))!)$ elements. Finally, even if storage and computation issues were resolved, one would still have sample complexity issues to contend with — for nontrivial n, it is impractical to hope that each of the $n!$ possible rankings would appear even once in a training set of rankings.

The quest for exploitable problem structure has led researchers to consider a number of possibilities which we briefly summarize now.

Parametric models We will not be able to do justice to the sheer volume of previous work on parametric ranking models. Parametric probabilistic models over the space of rankings have a rich tradition in statistics, [9, 10, 13, 24, 25, 27, 29, 36], and to this day, researchers continue to expand upon this body of work. For example, the well known *Mallows model*, which is often thought of as an analog of the normal distribution for permutations, parameterizes a distribution with a "mean" permutation and a precision/spread parameter.

The models proposed in this paper generalize many classical models from the statistical ranking literature, allowing for more expressive distributions to be captured. At the same time, our methods form a conceptual bridge to popular models (i.e., graphical models) from machine learning which, rather than relying explicitly on a prespecified parametric form, simply work within a family of distributions that are consistent with some set of conditional independence assumptions [22].

Sparse methods Sparse methods for summarizing distributions range from older ad-hoc approaches such as maintaining k-best hypotheses [30] to the recent compressed sensing inspired approaches discussed in [8, 20]. Such approaches assume that there are at most k permutations which own all (or almost all) of the probability mass, where k scales either sublinearly or as a low degree polynomial in n. While sparse distributions have been successfully applied in certain tracking domains, we argue that they are often less suitable in ranking problems where it might be necessary to model indifference over a large subset of objects. If one is approximately indifferent among a subset of k objects, then there are at least $k!$ rankings with nonzero probability mass. As an example, one can see that the APA vote distribution (Figure 1(a)) is clearly not a sparse distribution, with each ranking having received some nonzero number of votes.

Fourier-based (low-order) methods Another recent thread of research has centered around *Fourier-based methods* which maintain a set of low-order summary statistics [6, 16, 23, 33]. The *first-order summary*, for example, stores a marginal probability of the form $h(\boldsymbol{\sigma}(j) = i)$ for every pair (i, j) and thus requires storing a matrix of only $O(n^2)$ numbers. In our fruits/vegetables example, we might store the probability that Figs are ranked first, or the probability that Peas is ranked last. See Figure 1(b) for a grayscale visualization of the first order marginals for the APA dataset. More generally, one might store s^{th}-*order marginals*, which are marginal probabilities of s-tuples. The second-order marginals, for example, take the form $h(\boldsymbol{\sigma}(k, \ell) = (i, j))$.

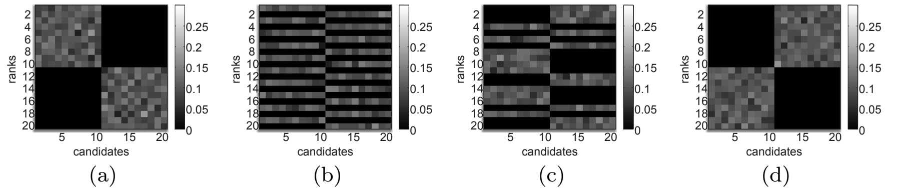
Low-order marginals turn out to be intimately tied to a generalized form of Fourier analysis. Fourier transforms for functions on permutations have been studied for some decades now [5, 6, 26, 31, 35]. In contrast with sparse methods, Fourier methods handle diffuse distributions well but are not easily scalable since, in general, one requires $O(n^{2s})$ coefficients to exactly reconstruct s^{th}-order marginals, which becomes intractable for moderately large n.

2.2. Fully independent subsets of items

To scale to larger problems, [18] 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 assumptions on the symmetric group can simply be stated as follows. Let A be a p-subset of $\{1, \ldots, n\}$, say, $\{1, \ldots, p\}$ and let B be its complement $(\{p + 1, \ldots, n\})$ with size $q = n - p$. We say that $\boldsymbol{\sigma}(\mathbf{A}) = (\boldsymbol{\sigma}(1), \boldsymbol{\sigma}(2), \ldots, \boldsymbol{\sigma}(p))$ and $\boldsymbol{\sigma}(\mathbf{B}) = (\boldsymbol{\sigma}(p + 1), \ldots, \boldsymbol{\sigma}(n))$ are *independent* if:

$$h(\boldsymbol{\sigma} = \sigma) = f_A(\boldsymbol{\sigma}(\mathbf{A}) = \sigma(A)) \cdot g_B(\boldsymbol{\sigma}(\mathbf{B}) = \sigma(B)), \text{ for all } \sigma \in S_n. \tag{2.1}$$

Storing parameters for the above distribution requires keeping $O(p! + q!)$ probabilities instead of the much larger $O(n!)$ required for general distributions. Of course, $O(p! + q!)$ can still be large, and typically, one decomposes the distribution recursively, storing factors exactly for small enough factors, or compressing factors using Fourier coefficients (but with higher frequency terms than what would be possible without independence assumptions). In order to exploit independence in the Fourier domain, [18] proposed algorithms for joining factors and splitting distributions into independent components in the Fourier domain.



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

Fig 2. *Example first-order matrices with $A = \{1,2,3\}$, $B = \{4,5,6\}$ fully independent, where black means $h(\sigma : \sigma(j) = i) = 0$. In each case, there is some 3-subset A' which A 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.*

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 [18] that, if $\boldsymbol{\sigma}(\mathbf{A})$ and $\boldsymbol{\sigma}(\mathbf{B})$ are independent, then A and B are not allowed to map to the same ranks. That is, for some fixed p-subset $A' \subset \{1,\ldots,n\}$, $\boldsymbol{\sigma}(\mathbf{A})$ is a permutation of elements in A' and $\boldsymbol{\sigma}(\mathbf{B})$ is a permutation of its complement, B', with probability 1.

Example 3. Continuing with our vegetable/fruit example with $n = 6$, if the vegetable and fruit rankings,

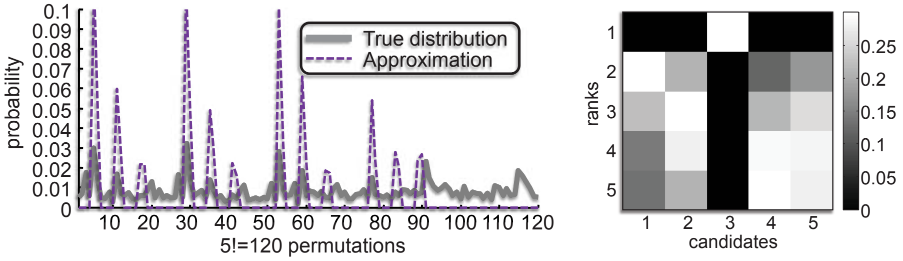
$$\boldsymbol{\sigma}(\mathbf{A}) = [\sigma(\text{Corn}), \sigma(\text{Peas})], \text{ and } \boldsymbol{\sigma}(\mathbf{B}) = [\sigma(\text{Lemons}), \sigma(\text{Oranges}), \sigma(\text{Figs}), \sigma(\text{Grapes})],$$

are known to be independent. Then for $A' = \{1,2\}$, the vegetables occupy the first and second ranks with probability one, and the fruits occupy ranks $B' = \{3,4,5,6\}$ with probability one, reflecting that vegetables are always preferred over fruits according to this distribution.

[18] refers to this restrictive constraint as the *first-order condition* because of the block structure imposed upon first-order marginals (see Figure 2). In our ranking example, 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 unrealistic.

Example 4 (APA election data (continued))**.** Consider approximating the APA vote distribution by a factorized distribution (as in Equation 2.1). In Figure 3, we plot (in solid purple) the factored distribution which is closest to the true distribution with respect to total variation distance. In our approximation, candidate 3 is constrained to be independent of the remaining four candidates and maps to rank 1 with probability 1.

While capturing the fact that the "winner" of the election should be candidate 3, the fully factored distribution can be seen to be a poor approximation, assigning zero probability to most permutations even if all permutations received a positive number of votes. Since the support of the true distribution is not



(a) (b)

Fig 3. *Approximating the APA vote distribution by a factored distribution in which candidate 3 is independent of candidates* $\{1, 2, 4, 5\}$. *(a) in thick gray, the true distribution, in dotted purple, the approximation. Notice that the factored distribution assigns zero probability to most permutations. (b) matrix of approximate first order marginals.*

contained within the support of the approximation, the Kullback-Leibler (KL) divergence, $D_{KL}(h_{true}; h_{approx})$ is infinite.

In the next section, we overcome the restrictive first-order condition with the more flexible notion of *riffled independence*.

3. *Riffled independence*: definitions and examples

The *riffle (or dovetail) shuffle* [2] is perhaps the most commonly used method of card shuffling, in which one cuts a deck of n cards into two piles, $A = \{1, \ldots, p\}$ and $B = \{p+1, \ldots, n\}$, with size p and $q = n - p$, respectively, and successively drops the cards, one by one, so that the two piles become interleaved into a single deck again. Inspired by the riffle shuffle, we 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. Intuitively, riffled independence models complex relationships within each set A and B while allowing correlations between the sets to be modeled only through a constrained form of shuffling.

Example 5. Consider generating a ranking of vegetables and fruits. We might first 'cut the deck' into two piles, a pile of vegetables (A) and a pile of fruits (B), and in a first stage, independently decide how to rank each pile. For example, within vegetables, we might decide that Peas are preferred to Corn: $[\![\mathbf{P}, \mathbf{C}]\!] = [\![Peas, Corn]\!]$. Similarly, within fruits, we might decide on the ranking: $[\![\mathbf{L}, \mathbf{F}, \mathbf{G}, \mathbf{O}]\!] = [\![Lemons, Figs, Grapes, Oranges]\!]$ (Lemons preferred to Figs, Figs preferred to Grapes, Grapes preferred to Oranges).

In the second stage of our model, the fruit and vegetable rankings are interleaved to form a full preference ranking over all six items. For example, if the interleaving is given by: $[\![Veg, Fruit, Fruit, Fruit, Veg, Fruit]\!]$, then the resulting full ranking is: $\sigma = [\![Peas, Lemons, Figs, Grapes, Corn, Oranges]\!]$.

3.1. Convolution based definition of riffled independence

There are two ways to define riffled independence, and, we will first provide a definition using convolutions, a view inspired by our card shuffling intuitions. Mathematically, shuffles are modeled as random walks on the symmetric group. The ranking $\boldsymbol{\sigma}$ *after* a shuffle is generated from the ranking *prior* to that shuffle, $\boldsymbol{\sigma}'$, by drawing a permutation, $\boldsymbol{\tau}$ from an *interleaving distribution* $m(\boldsymbol{\tau})$, and setting $\boldsymbol{\sigma} = \boldsymbol{\tau}\boldsymbol{\sigma}'$ (the composition of the mapping $\boldsymbol{\tau}$ with $\boldsymbol{\sigma}'$). Given the distribution h' over $\boldsymbol{\sigma}'$, we can find the distribution $h(\boldsymbol{\sigma})$ *after* the shuffle via the formula: $h(\boldsymbol{\sigma} = \sigma) = \sum_{\sigma',\tau\,:\,\sigma=\tau\sigma'} m(\boldsymbol{\tau} = \tau)h'(\boldsymbol{\sigma}' = \sigma')$. This operation which combines the distributions m and h is commonly known as *convolution*:

Definition 6. Let m and h' be probability distributions on S_n. The *convolution* of the distributions is the function:

$$[m * h'](\boldsymbol{\sigma} = \sigma) \equiv \sum_{\pi \in S_n} m(\boldsymbol{\tau} = \pi) \cdot h'(\boldsymbol{\sigma}' = \pi^{-1}\sigma).$$

We use the $*$ symbol to denote the convolution operation. Note that $*$ is not in general commutative (hence, $m * h' \neq h' * m$).

Besides the riffle shuffle, there are a number of different shuffling strategies — the pairwise shuffle, for example, simply selects two cards at random and swaps them. The question then, is *what are interleaving 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 Peas is preferred over Corn prior to shuffling, they continue to be preferred over Corn 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 7 (Interleaving distributions)**.** The (p, q)*-interleavings* are defined as the following set:

$$\Omega_{p,q} \equiv \{\tau \in S_n \,:\, \tau(1) < \tau(2) < \cdots < \tau(p), \text{ and } \tau(p+1) < \tau(p+2) < \cdots < \tau(n)\}.$$

A distribution $m_{p,q}$ on S_n is called an *interleaving distribution* if it assigns nonzero probability mass *only* to elements in $\Omega_{p,q}$.

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

Lemma 8. *Let $i, j \in A = \{1, \dots, p\}$ (or $i, j \in B = \{p + 1, \dots, n\}$) and let τ be any (p, q)-interleaving in $\Omega_{p,q}$. Then $i < j$ if and only if $\tau(i) < \tau(j)$ (i.e., permutations in $\Omega_{p,q}$ preserve relative ranking relations).*

Example 9. In our vegetable/fruits example, we have $n = 6$, $p = 2$ (two vegetables, four fruits). The set of $(2, 4)$-interleavings is:

$$\Omega_{2,4} = \left\{ \begin{array}{ccccc} (\mathbf{1,2,3,4,5,6}), & (\mathbf{1,3,2,4,5,6}), & (\mathbf{1,4,2,3,5,6}), & (\mathbf{1,5,2,3,4,6}), & (\mathbf{1,6,2,3,4,5}), \\ (\mathbf{2,3,1,4,5,6}), & (\mathbf{2,4,1,3,5,6}), & (\mathbf{2,5,1,3,4,6}), & (\mathbf{2,6,1,3,4,5}), & (\mathbf{3,4,1,2,5,6}), \\ (\mathbf{3,5,1,2,4,5}), & (\mathbf{3,6,1,2,4,5}), & (\mathbf{4,5,1,2,3,6}), & (\mathbf{4,6,1,2,3,5}), & (\mathbf{5,6,1,2,3,4}) \end{array} \right\},$$

or written in ordering notation,

$$\Omega_{2,4} = \left\{ \begin{array}{ccccc} \llbracket\mathbf{VVFFFF}\rrbracket, & \llbracket\mathbf{VFVFFF}\rrbracket, & \llbracket\mathbf{VFFVFF}\rrbracket, & \llbracket\mathbf{VFFFVF}\rrbracket, & \llbracket\mathbf{VFFFFV}\rrbracket, \\ \llbracket\mathbf{FVVFFF}\rrbracket, & \llbracket\mathbf{FVFVFF}\rrbracket, & \llbracket\mathbf{FVFFVF}\rrbracket, & \llbracket\mathbf{FVFFFV}\rrbracket, & \llbracket\mathbf{FFVVFF}\rrbracket, \\ \llbracket\mathbf{FFVFVF}\rrbracket, & \llbracket\mathbf{FFVFFV}\rrbracket, & \llbracket\mathbf{FFFVVF}\rrbracket, & \llbracket\mathbf{FFFVFV}\rrbracket, & \llbracket\mathbf{FFFFVV}\rrbracket \end{array} \right\}.$$

Note that the number of possible interleavings is $|\Omega_{p,q}| = \binom{n}{p} = \binom{n}{q} = 6!/(2!4!) = 15$. One possible riffle shuffling distribution on S_6 might, for example, assign uniform probability ($m^{unif}_{2,4}(\boldsymbol{\tau} = \tau) = 1/15$) to each permutation $\tau \in \Omega_{2,4}$ 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 undergoes a single riffle shuffle.

Definition 10 (Riffled independence)**.** The subsets $A = \{1, \ldots, p\}$ and $B = \{p + 1, \ldots, n\}$ are said to be *riffle independent* if for all $\sigma \in S_n$,

$$h(\boldsymbol{\sigma} = \sigma) = m_{p,q} * (f_A(\boldsymbol{\sigma}(\mathbf{A}) = \sigma(A)) \cdot g_B(\boldsymbol{\sigma}(\mathbf{B}) = \sigma(B))),$$

with respect to some interleaving distribution $m_{p,q}$ and distributions f_A, g_B, respectively. We will notate the riffled independence relation as $A \perp_m B$, and refer to f_A, g_B as *relative ranking factors*.

Notice that without the additional convolution, the definition of riffled independence reduces to the fully independent case given by Equation 2.1.

Example 11. Consider drawing a ranking from a riffle independent model. One starts with two piles of cards, A and B, stacked together in a deck. In our fruits/vegetables setting, if we always prefer vegetables to fruits, then the vegetables occupy positions $\{1, 2\}$ and the fruits occupy positions $\{3, 4, 5, 6\}$. In the first step, rankings of each pile are drawn independent. For example, we might have the rankings: $\sigma(\text{Veg}) = (2, 1)$ and $\sigma(\text{Fruit}) = (4, 6, 5, 3)$, constituting a draw from the fully independent model described in Section 2.2. In the second stage, the deck of cards is cut and interleaved by an independently selected element $\tau \in \Omega_{2,4}$. For example, if:

$$\tau = (2, 3, 1, 4, 5, 6) = \llbracket Fruit, Veg, Veg, Fruit, Fruit, Fruit \rrbracket,$$

then the joint ranking is:

$$\begin{aligned} \tau(\sigma(Veg), \sigma(Fruit)) &= (2, 3, 1, 4, 5, 6)(2, 1, 4, 6, 5, 3) = (3, 2, 4, 6, 5, 1), \\ &= \llbracket Grapes, Peas, Corn, Lemon, Fig, Orange \rrbracket. \end{aligned}$$

3.2. Alternative definition of riffled independence

It is possible to rewrite the definition of riffled independence so that it does not involve a convolution. We first define functions which map a given full ranking to relative rankings and interleavings for A and B.

Definition 12.

- (*Absolute ranks*): Given a ranking $\sigma \in S_n$, and a subset $A \subset \{1, \ldots, n\}$, $\sigma(A)$ denotes the *absolute ranks* of items in A.
- (*Relative ranking map*): Let $\phi_A(\sigma)$ denote the ranks of items in A *relative* to the set A. For example, in the ranking $\sigma = [\![\mathbf{P}, \mathbf{L}, \mathbf{F}, \mathbf{G}, \mathbf{C}, \mathbf{O}]\!]$, the relative ranks of the vegetables is $\phi_A(\sigma) = [\![\mathbf{P}, \mathbf{C}]\!] = [\![Peas, Corn]\!]$. Thus, while corn is ranked fifth in σ, it is ranked second in $\phi_A(\sigma)$. Similarly, the relative ranks of the fruits is $\phi_B(\sigma) = [\![\mathbf{L}, \mathbf{F}, \mathbf{G}, \mathbf{O}]\!] = [\![Lemons, Figs, Grapes, Oranges]\!]$.
- (*Interleaving map*): Likewise, let $\tau_{A,B}(\sigma)$ denote the way in which the sets A and B are interleaved by σ. For example, using the same σ as above, the interleaving of vegetables and fruits is $\tau_{A,B}(\sigma) = [\![Veg, Fruit, Fruit, Fruit, Veg, Fruit]\!]$. In ranking notation (as opposed to ordering notation), $\tau_{A,B}$ can be written as $(\mathrm{sort}(\sigma(A)), \mathrm{sort}(\sigma(B)))$. Note that for every possible interleaving, $\tau \in \Omega_{p,q}$ there are exactly $p! \times q!$ distinct permutations which are associated to τ by the interleaving map.

Using the above maps, we now provide an algebraic expression for how any permutation σ can be uniquely decomposed into an interleaving composed with relative rankings of A and B, which have been "stacked" into one deck.

Lemma 13. *Let $A = \{1, \ldots, p\}$, and $B = \{p+1, \ldots, n\}$. Any ranking $\sigma \in S_n$ can be decomposed* uniquely *as an interleaving $\tau \in \Omega_{p,q}$ composed with a ranking of the form $(\pi_p, \pi_q + p)$, where $\pi_p \in S_p$, $\pi_q \in S_q$, and $\pi_q + p$ means that the number p is added to every rank in π_q. Specifically, $\sigma = \tau(\pi_p, \pi_q + p)$ with $\tau = \tau_{A,B}(\sigma)$, $\pi_p = \phi_A(\sigma)$, and $\pi_q = \phi_B(\sigma)$.*

Lemma 13 shows that one can think of a triplet $(\tau \in \Omega_{p,q}, \sigma_p \in S_p, \sigma_q \in S_q)$ as the coordinates which uniquely specify any ranking of items in $A \cup B$. Using the decomposition, we now state an equivalent, perhaps more intuitive definition of riffled independence in terms of relative ranking and interleaving maps.

Definition 14. Sets A and B are said to be *riffle independent* if and only if, for every $\sigma \in S_n$, the joint distribution h factors as:

$$h(\boldsymbol{\sigma} = \sigma) = m(\boldsymbol{\tau} = \tau_{A,B}(\sigma)) \cdot f_A(\boldsymbol{\sigma}(\mathbf{A}) = \phi_A(\sigma)) \cdot g_B(\boldsymbol{\sigma}(\mathbf{B}) = \phi_B(\sigma)). \quad (3.1)$$

Proposition 15. *Definitions 10 and 14 are equivalent.*

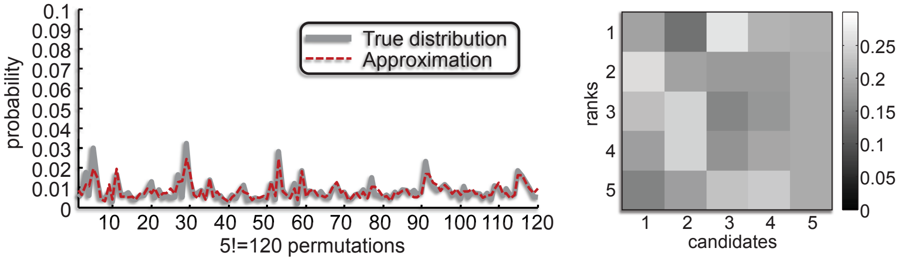
Discussion We have presented two ways of thinking about riffled independence. Our first formulation, in terms of convolution, is motivated by the connections between riffled independence and card shuffling theory. Our second formulation on the other hand, shows the concept of riffled independence to be remarkably simple — that the probability of a single ranking can be computed without summing over all rankings (required in convolution) — a fact which may not have been obvious from Definition 10.

Special cases There are a number of special case distributions captured by the riffled independence model that are useful for intuition. We discuss these extreme cases in the following list.

- (*Uniform and delta distributions*): Setting the interleaving distribution and both relative ranking factors to be uniform distributions yields the uniform distribution over all full rankings. Similarly, setting the same distributions to be delta distributions (which assign zero probability to all rankings but one) always yields a delta distribution.
  It is interesting to note that while A and B are always fully independent under a delta distribution, they are never independent under a uniform distribution. However, both uniform and delta distributions factor *riffle independently* with respect to any partitioning of the item set. Thus, not only is $A = \{1, \dots, p\}$ riffle independent $B = \{p+1, \dots, n\}$, but in fact, any set A is riffle independent of its complement.
- (*Uniform interleaving distributions*): Setting the interleaving distribution to be uniform reflects complete indifference between the sets A and B, even if f and g encode complex preferences within each set alone.
- (*Delta interleaving distributions*): Setting the interleaving distribution, $m_{p,q}$, to be a 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 (see Figure 2). Just as in the full independence regime, where the distributions f and g are marginal distributions of absolute rankings of A and B, in the riffled independence regime, f and g can be thought of as marginal distributions of the *relative rankings* of item sets A and B.

Example 16 (APA election data (continued))**.** Like the independence assumptions commonly used in naive Bayes models, we would rarely expect riffled independence to exactly hold in real data. Instead, it is more appropriate to view riffled independence assumptions as a form of model bias that ensures learnability for small sample sizes, which as we have indicated, is almost always the case for distributions over rankings.

Can we ever expect riffled independence to be approximately manifested in a real dataset? In Figure 4(a), we plot (dotted red) a riffle independent approximation to the true APA vote distribution (thick gray) which is optimal with respect to KL-divergence (we will explain how to obtain the approximation in the remainder of the paper). The approximation in Figure 4(a) is obtained by assuming that the candidate set $\{1, 3, 4, 5\}$ is riffle independent of $\{2\}$, and as can be seen, is quite accurate compared to the truth. Figure 4(b) exhibits the first order marginals of the approximating distribution (see Figure 1(b)). We will discuss the interpretation of the result further in Section 4.



(a) (b)

Fig 4. *Approximating the APA vote distribution by riffle independent distributions. (a) approximate distribution when candidate 2 is riffle independent of remaining candidates; (b) is the corresponding matrix of first order marginals.*

3.3. Computation

We briefly introduce two related computational issues related to riffled independence — the problems of (1) estimating model parameters, and (2), computing marginals of a joint distribution which we know factors riffle independently.

Parameter estimation Given a set of i.i.d. training examples, $\sigma^{(1)}, \ldots, \sigma^{(m)}$, we are interested in the problem of estimating the parameter distributions $m_{p,q}$, f_A, and g_B. In this section we assume a *known structure* (that the partitioning of the item set into subsets A and B is known and fixed). Since our variables are discrete, computing the maximum likelihood parameter estimates consists of forming counts of the number of training examples consistent with a given interleaving or relative ranking. Thus, the MLE parameters in our problem are simply given by the following formulas:

$$m_{p,q}^{MLE}(\tau) \propto \sum_{i=1}^{m} \mathbb{1}\left[\tau = \tau_{A,B}(\sigma^{(i)})\right],$$

$$f_A^{MLE}(\sigma_A) \propto \sum_{i=1}^{m} \mathbb{1}\left[\sigma_A = \phi_A(\sigma^{(i)})\right], \qquad g_B^{MLE}(\sigma_B) \propto \sum_{i=1}^{m} \mathbb{1}\left[\sigma_B = \phi_B(\sigma^{(i)})\right].$$

Efficient computation of marginals Another useful operation is that of computing marginal probabilities of a distribution (for the purposes of visualization, for example). We now state a simple result that shows that given the first-order marginal probabilities of each factor in a riffle independent distribution h, it is possible to compute the first-order marginal probabilities of h.

Theorem 3.1. *Let A and B be riffle independent with respect to distribution h. If M is the matrix of first-order marginals of the interleaving distribution $m_{p,q}$ and F and G the first-order marginals of relative ranking distributions f_A and g_B, respectively, then the first order matrix of marginals of h is given by $H = M \cdot (F \oplus G)$, where $\oplus$ represents the direct sum operation and $\cdot$ represents ordinary matrix multiplication.*

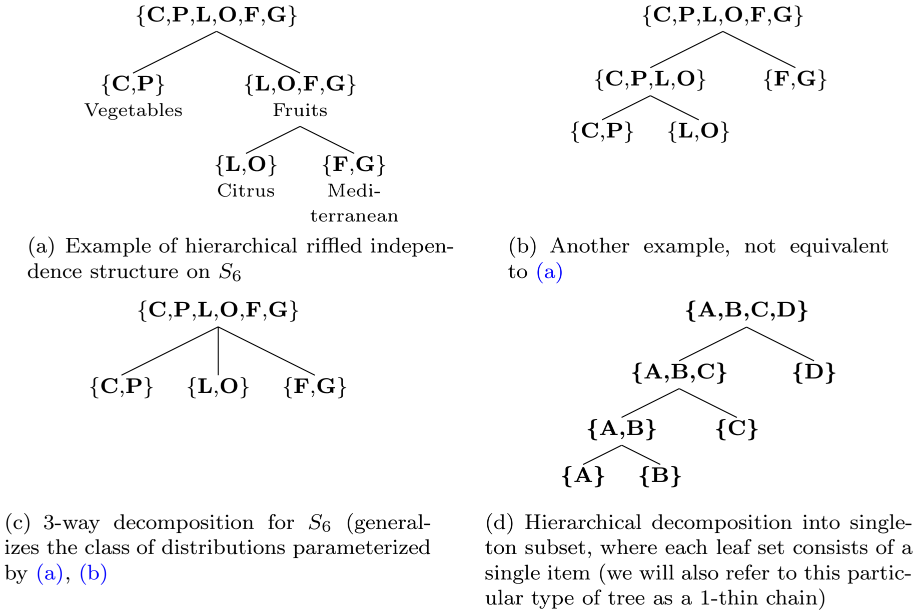
We note that one way to establish Theorem 3.1 is to use the convolution theorem from Fourier analysis (see [15]) to convert the convolutions of distributions to matrix products of Fourier terms. Using the convolution theorem, the same result can also be generalized to higher order terms by replacing the word "first-order" by "kth-order" everywhere. We use the result of Theorem 3.1 to produce first-order marginals for each of the datasets in Section 6.

4. Hierarchical riffle independent decompositions

Thus far, we have focused exclusively on understanding riffle independent models with a single binary partitioning of the full item set. In this section we explore a natural model simplification which comes from the simple observation that, since the relative ranking distributions f_A and g_B are again distributions over rankings, the sets A and B can further be decomposed into riffle independent subsets. We call such models *hierarchical riffle independent decompositions.* Continuing with our running example, one can imagine that the fruits are further partitioned into two sets, a set consisting of citrus fruits ((**L**) Lemons and (**O**) Oranges) and a set consisting of mediterranean fruits ((**F**) Figs and (**G**) Grapes). To generate a full ranking, one first draws rankings of the citrus and mediterranean fruits independently ($[\![\mathbf{L}, \mathbf{O}]\!]$ and $[\![\mathbf{G}, \mathbf{F}]\!]$, for example). Secondly, the two sets are interleaved to form a ranking of all fruits ($[\![\mathbf{G}, \mathbf{L}, \mathbf{O}, \mathbf{F}]\!]$). Finally, a ranking of the vegetables is drawn ($[\![\mathbf{P}, \mathbf{C}]\!]$) and interleaved with the fruit rankings to form a full joint ranking: $[\![\mathbf{P}, \mathbf{G}, \mathbf{L}, \mathbf{O}, \mathbf{F}, \mathbf{C}]\!]$. Notationally, we can express the hierarchical decomposition as $\{\mathbf{P}, \mathbf{C}\} \perp_{m_1} (\{\mathbf{L}, \mathbf{O}\} \perp_{m_2} \{\mathbf{F}, \mathbf{G}\})$. We can also visualize hierarchies using trees (see Figure 5(a) for our example). The subsets of items which appear as leaves in the tree will be referred to as *leaf sets*.

A natural question to ask is: if we used a different hierarchy with the same leaf sets, would we capture the same distributions? For example, does a distribution which decomposes according to the tree in Figure 5(b) also decompose according to the tree in Figure 5(a)? The answer, in general, is no, due to the fact that distinct hierarchies impose different sets of independence assumptions, and as a result, different structures can be well or badly suited to modeling a given dataset. Consequently, it is important to use the "correct" structure if possible.

Shared independence structure It is interesting to note, however, that while the two structures in Figures 5(a) and 5(b) encode distinct families of distributions, it is possible to identify a set of independence assumptions common to both structures. In particular since both structures have the same leaf sets, any distributions consistent with either of the two hierarchies must also be consistent with what we call a 3-*way decomposition*. We define a d-way decomposition to be a distribution with a single level of hierarchy, but instead of partitioning the entire item set into just two subsets, one partitions into d subsets, then interleaves the relative rankings of each of the d subsets together to form a joint ranking of items. Any distribution consistent with either Figure 5(b) or 5(a) must consequently also be consistent with the structure of Figure 5(c). More generally, we have:



(a) Example of hierarchical riffled independence structure on S_6

(b) Another example, not equivalent to (a)

(c) 3-way decomposition for S_6 (generalizes the class of distributions parameterized by (a), (b)

(d) Hierarchical decomposition into singleton subset, where each leaf set consists of a single item (we will also refer to this particular type of tree as a 1-thin chain)

Fig 5. *Examples of distinct hierarchical riffle independent structures.*

Proposition 17. *If h is a hierarchical riffle independent model with d leaf sets, then h can also be written as a d-way decomposition.*

In general, knowing the hierarchical decomposition of a model is more desirable than knowing its d-way decomposition which may require many more parameters $\big(O(\frac{n!}{\prod_i d_i!})$, where i indexes over leaf sets$\big)$. For example, the n-way decomposition requires $O(n!)$ parameters and captures every distribution over permutations.

Thin chain models There is a class of particularly simple hierarchical models which we will refer to as k-thin chain models. By a k-thin chain model, we refer to a hierarchical structure in which the size of the smaller set at each split in the hierarchy is fixed to be a constant and can therefore be expressed as: $(A_1 \perp_m (A_2 \perp_m (A_3 \perp_m \dots)))$, $|A_i| = k$, for all i.

See Figure 5(d) for an example of 1-thin chain. We view thin chains as being somewhat analogous to thin junction tree models [1], in which cliques are never allowed to have more than k variables. When $k \sim O(1)$, for example, the number of model parameters scales polynomially in n. To draw rankings from a thin chain model, one sequentially inserts items independently, one group of size k at a time, into the full ranking.

Example 18 (APA election data (continued))**.** The APA, as described by [7], is divided into "*academicians and clinicians who are on uneasy terms*". In 1980, candidates $\{1, 3\}$ (W. Bevan and C. Kiesler who were research psychologists) and $\{4, 5\}$ (M.Siegle and L. Wright, who were clinical psychologists) fell on op-

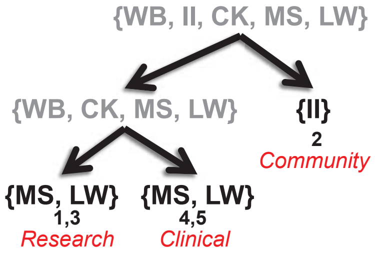


Fig 6. *Hierarchical structure learned from APA data.*

posite ends of this political spectrum with candidate 2 (I. Iscoe) being somewhat independent. Diaconis conjectured that voters choose one group over the other, and then choose within. We are now able to verify Diaconis' conjecture using our riffled independence framework. 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 $A = \{1,3\}$ and $B = \{4,5\}$ are very nearly riffle independent (with respect to KL divergence) 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. We remark that in a later work, [25] identified candidate 2 (I. Iscoe) as belonging to yet a third group of psychologists called *community psychologists*. The hierarchical structure that best describes the APA data is shown in Figure 6.

5. Structure discovery

Since different hierarchies impose different independence assumptions, we would like to find the structure that is optimally suited to modeling a given ranking dataset. On some datasets, a natural hierarchy might be available — for example, if one were familiar with the typical politics of APA elections, then it may have been possible to "guess" the optimal hierarchy. However, for general ranked data, it is not always obvious what kinds of groupings riffled independence will lead to, particularly for large n. Should fruits really be riffle independent of vegetables? Or are green foods riffle independent of red foods?

Over the next two sections, we address the problem of automatically discovering hierarchical structures from training data. Key among our observations is the fact that while item ranks cannot be independent due to mutual exclusivity, relative ranks between sets of items are not subject to the same constraints. More than simply being a 'clustering' algorithm, however, our procedure can be thought of as a structure learning algorithm, like those from the graphical models literature [22], which find the optimal (riffled) independence decomposition of a distribution.

This section addresses how one might find the optimal structure if there is only one level of partitioning and two leaf sets, A, B. Alternatively, we want to find the topmost partitioning of the tree. In Section 5.5, we use this base case as part of a top-down approach for learning a hierarchy.

5.1. Problem statement

Given a training set of rankings, $\sigma^{(1)}$, $\sigma^{(2)}$, $\ldots, \sigma^{(m)} \sim h$, drawn i.i.d. from a distribution in which a subset of items, $A \subset \{1, \ldots, n\}$, is riffle independent of its complement, B, the problem which we address in this section is that of automatically determining the sets A and B. If h does not *exactly* factor riffle independently, then we would like to find the riffle independent approximation which is *closest* to h in some sense. Formally, we would like to solve the problem:

$$\arg\min_{A} \min_{m,f,g} \; D_{KL}(\hat{h}(\cdot)\,||\,m(\tau_{A,B}(\cdot))f(\phi_A(\cdot))g(\phi_B(\cdot))), \tag{5.1}$$

where $\hat{h}$ is the empirical distribution of training examples and D_{KL} is the Kullback-Leibler divergence measure. Equation 5.1 is a seemingly reasonable objective since it can also be interpreted as maximizing the likelihood of the training data. In the limit of infinite data, Equation 5.1 can be shown via the Gibbs inequality to attain its minimum, zero, at the subsets A and B, if and only if the sets A and B are truly riffle independent of each other.

For small problems, one can actually solve Problem 5.1 using a single computer by evaluating the approximation quality of each subset A and taking the minimum, which was the approach taken in Example 18. However, for larger problems, one runs into time and sample complexity problems since optimizing the globally defined objective function (Equation 5.1) requires relearning all model parameters (m, f_A, and g_B) for each of the exponentially many subsets of $\{1, \ldots, n\}$. In fact, for large sets A and B, it is rare that one would have enough samples to estimate the relative ranking parameters f_A and g_B without already having discovered the hierarchical riffle independent decompositions of A and B.

We next propose a low-order proxy objective function, reminiscent of clustering, which we will use instead of Equation 5.1. As we show, our new objective will be more tractable to compute and have lower sample complexity for estimation. The idea of using a low order proxy objective is similar to an idea which was recently introduced in [32], which determines optimally thin separators with respect to the Bethe free energy approximation (of the entropy) rather than a typical log-likelihood objective. The resulting sample complexity analysis is based on the mutual information sample complexity bounds derived in [14], which was also used in [3] for developing a structure learning algorithm for thin junction trees with provably polynomial sample complexity.

5.2. Proposed objective function

The approach we take is to minimize a different measure that exploits the observation that *absolute ranks of items in A are fully independent of relative ranks of items in B, and vice versa* (which we prove in Proposition 19). With our vegetables and fruits, for example, knowing that Figs is ranked first among all six items (the absolute rank of a fruit) should give no information about whether

Corn is preferred to Peas (the relative rank of vegetables). More formally, given a subset $A = \{a_1, \ldots, a_\ell\}$, recall that $\sigma(A)$ denotes the vector of (absolute) ranks assigned to items in A by $\boldsymbol{\sigma}$ (thus, $\boldsymbol{\sigma}(\mathbf{A}) = (\boldsymbol{\sigma}(a_1), \boldsymbol{\sigma}(a_2), \ldots, \boldsymbol{\sigma}(a_\ell)))$. We propose to minimize an alternative objective function:

$$\mathcal{F}(A) \equiv I(\boldsymbol{\sigma}(\mathbf{A})\ ;\ \phi_B(\boldsymbol{\sigma})) + I(\boldsymbol{\sigma}(\mathbf{B})\ ;\ \phi_A(\boldsymbol{\sigma})), \tag{5.2}$$

where I denotes the mutual information (defined between two variables $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ by $I(\boldsymbol{X}_1; \boldsymbol{X}_2) \equiv D_{KL}(P(\boldsymbol{X}_1, \boldsymbol{X}_2)||P(\boldsymbol{X}_1)P(\boldsymbol{X}_2))$.

The function $\mathcal{F}$ does not have the same likelihood interpretation as the objective function of Equation 5.1. However, it can be thought of as a composite likelihood of two models, one in which the relative rankings of A are independent of absolute rankings of B, and one in which the relative rankings of B are independent of absolute rankings of A. With respect to distributions which satisfy (or approximately satisfy) both models (i.e., the riffle independent distributions), minimizing $\mathcal{F}$ *is* equivalent to (or approximately equivalent to) maximizing the log likelihood of the data. Furthermore, we can show that $\mathcal{F}$ is guaranteed to detect riffled independence:

Proposition 19. *$\mathcal{F}(A) = 0$ is a necessary and sufficient criterion for a subset $A \subset \{1, \ldots, n\}$ to be riffle independent of its complement, B.*

As with Equation 5.1, optimizing $\mathcal{F}$ is still intractable for large n. However, $\mathcal{F}$ motivates a natural proxy, in which we replace the mutual informations defined over all n variables by a sum of mutual informations defined over just three variables at a time.

Definition 20 (Tripletwise mutual informations)**.** Given any triplet of distinct items, (i, j, k), we define the tripletwise mutual information term, $I_{i;j,k} \equiv I(\boldsymbol{\sigma}(i)\,;\,\boldsymbol{\sigma}(j) < \boldsymbol{\sigma}(k))$, which can be computed as follows:

$$I(\boldsymbol{\sigma}(i)\,;\,\boldsymbol{\sigma}(j) < \boldsymbol{\sigma}(k)) = \sum_{\sigma(i)} \sum_{\sigma(j)<\sigma(k)} h(\sigma(i), \sigma(j) < \sigma(k)) \log \frac{h(\sigma(i), \sigma(j) < \sigma(k))}{h(\sigma(i))h(\sigma(j) < \sigma(k))},$$

where the inside summation runs over two values, true/false, for the binary variable $\boldsymbol{\sigma}(j) < \boldsymbol{\sigma}(k)$.

To evaluate how riffle independent two subsets A and B are, we want to examine the triplets that straddle the two sets.

Definition 21 (Internal and Cross triplets)**.** We define $\Omega^{cross}_{A,B}$ to be the set of triplets which "cross" from set A to set B: $\Omega^{cross}_{A,B} \equiv \{(i; j, k)\ :\ i \in A, j, k \in B\}$. $\Omega^{cross}_{B,A}$ is similarly defined. We also define Ω^{int}_A to be the set of triplets that are internal to A: $\Omega^{int}_A \equiv \{(i; j, k)\ :\ i, j, k \in A\}$, and again, Ω^{int}_B is similarly defined.

Our proxy objective function can be written as the sum of the mutual information evaluated over all of the crossing triplets:

$$\tilde{\mathcal{F}}(A) \equiv \sum_{(i,j,k)\in\Omega^{cross}_{A,B}} I_{i;j,k} + \sum_{(i,j,k)\in\Omega^{cross}_{B,A}} I_{i;j,k}. \tag{5.3}$$

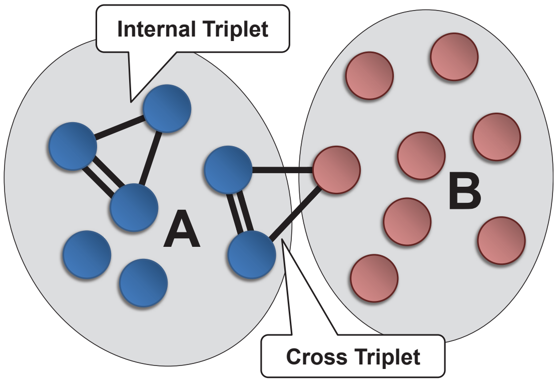


Fig 7. *Above is a graphical depiction of the problem of finding riffle independent subsets. A triangle with vertices (i, j, k) represents the term $I_{i;j,k}$. Since the $I_{i;j,k}$ are not invariant with respect to a permutation of the indices i, j, and k, the triangles are directed, and we therefore use double bars represent the nodes j, k for the term $I_{i;j,k}$. Note that if the tripletwise terms were instead replaced by edgewise terms, the problem would simply be a standard clustering problem.*

$\tilde{\mathcal{F}}$ can be viewed as a low order version of $\mathcal{F}$, involving mutual information computations over triplets of variables at a time instead of n-tuples. The mutual information $I_{i;j,k}$, for example, reflects how much the rank of a vegetable (i) tells us about how two fruits $(j,\ k)$ compare. If A and B are riffle independent, then we know that $I_{i;j,k} = 0$ for any (i, j, k) such that $i \in A$, $j, k \in B$ (and similarly for any (i, j, k)) such that $i \in B$, $j, k \in A$. Given that fruits and vegetables are riffle independent sets, knowing that Grapes is preferred to Figs should give no information about the absolute rank of Corn, and therefore $I_{Corn;Grapes,Figs}$ should be zero.

The objective $\tilde{\mathcal{F}}$ is somewhat reminiscent of typical graphcut and clustering objectives. Instead of partitioning a set of nodes based on sums of pairwise similarities, we partition based on sums of tripletwise affinities. We show a graphical depiction of the problem in Figure 7, where cross triplets in $(\Omega^{cross}_{A,B}, \Omega^{cross}_{B,A})$ have low weight and internal triplets in $(\Omega^{int}_{A}, \Omega^{int}_{B})$ have high weight. The objective is to find a partition such that the sum over cross triplets is low. In fact, the problem of optimizing $\tilde{\mathcal{F}}$ can be seen as an instance of the weighted, directed hypergraph cut problem [11]. Note that the word *directed* is significant for us, because, unlike typical clustering problems, our triplets are not symmetric (for example, $I_{i;jk} \neq I_{j;ik}$), resulting in a nonstandard and poorly understood optimization problem.

5.3. Low-order detectability assumptions

When does $\tilde{\mathcal{F}}$ detect riffled independence? It is not difficult to see, for example, that $\tilde{\mathcal{F}} = 0$ is a necessary condition for riffled independence, since $A \perp_m B$ implies $I_{a;b,b'} = 0$. We have:

Proposition 22. *If A and B are riffle independent sets, then $\tilde{\mathcal{F}}(A) = 0$.*

However, the converse of Proposition 22 is not true in full generality without accounting for dependencies that involve larger subsets of variables. Just as the pairwise independence assumptions that are commonly used for randomized algorithms [28][3] do not imply full independence between two sets of variables, there exist distributions which "look" riffle independent from tripletwise marginals but do not factor upon examining higher-order terms. Nonetheless, in most practical scenarios, we expect $\tilde{\mathcal{F}} = 0$ to imply riffled independence.

5.4. Estimating the objective from samples

We have so far argued that $\tilde{\mathcal{F}}$ is a reasonable function for finding riffle independent subsets. However, since we only have access to samples rather than the true distribution h itself, it will only be possible to compute an approximation to the objective $\tilde{\mathcal{F}}$. In particular, for every triplet of items, (i, j, k), we must compute an estimate of the mutual information $I_{i;j,k}$ from i.i.d. samples drawn from h, and so the question is: how many samples will we need in order for the approximate version of $\tilde{\mathcal{F}}$ to remain a reasonable objective function?

In the following, we denote the estimated value of $I_{i;j,k}$ by $\hat{I}_{i;j,k}$. For each triplet, we use a regularized procedure due to [14] to estimate mutual information and adapt his sample complexity bound to our problem:

Lemma 23. *For any fixed triplet (i, j, k), the mutual information $I_{i;j,k}$ can be estimated to within an accuracy of Δ with probability at least $1 - \gamma$ using $S(\Delta, \gamma) \equiv O\big(\frac{n^2}{\Delta^2} \log^2 \frac{n}{\Delta} \log \frac{n}{\gamma}\big)$ i.i.d. samples.*

The approximate objective function is therefore:

$$\hat{\mathcal{F}}(A) \equiv \sum_{(i,j,k)\in\Omega^{cross}_{A,B}} \hat{I}_{i;j,k} + \sum_{(i,j,k)\in\Omega^{cross}_{B,A}} \hat{I}_{i;j,k}.$$

What we want to show now is that, if there exists a unique way to partition $\{1, \ldots, n\}$ into riffle independent sets, then given enough training examples, our approximation $\hat{\mathcal{F}}$ uniquely singles out the correct partition as its minimum with high probability. A class of riffle independent distributions for which the uniqueness requirement is satisfied consists of the distributions for which A and B are *strongly connected* according to the following definition.

Definition 24. A subset $A \subset \{1, \ldots, n\}$ is called *ϵ-third-order strongly connected* (we will often just say *strongly connected*) if, for every triplet $i, j, k \in A$ with i, j, k distinct, we have $I_{i;j,k} > \epsilon$.

If A and B are riffle independent with both sets strongly connected, then we can ensure that riffled independence is detectable and that the partition is unique. We have the following probabilistic guarantee.

[3] A pairwise independent family of random variables is one in which any two members are marginally independent. Subsets with larger than two members may not necessarily factor independently, however.

Theorem 25. *Let A and B be ϵ-third order strongly connected riffle independent sets, and suppose $|A| = k$. Given $S(\Delta, \epsilon) \equiv O\big(\frac{n^4}{\epsilon^2} \log^2 \frac{n}{\epsilon} \log \frac{n}{\gamma}\big)$ i.i.d. samples, the minimum of $\hat{\mathcal{F}}$ is achieved at exactly the subsets A and B with probability at least $1 - \gamma$.*

We remark that the strong connectivity assumptions used in Theorem 25 are stronger than necessary — and with respect to certain interleaving distributions, it can even be the case that the estimated objective function singles out the correct partition when all of internal triplets belonging to A and B have zero mutual information. Moreover, in some cases, there are multiple valid partitionings of the item set. For example the uniform distribution is a distribution in which every subset $A \subset \{1, \ldots, n\}$ is riffle independent of its complement. In such cases, multiple solutions are equally good when evaluated under $\tilde{\mathcal{F}}$, but not its sample approximation, $\hat{\mathcal{F}}$.

5.5. Structure discovery algorithms

Having now designed a function that can be tractably estimated, we turn to the problem of learning the hierarchical structure of a distribution from training examples. Instead of directly optimizing an objective in the space of possible hierarchies, we take a top-down approach in which the item sets are recursively partitioned by optimizing $\hat{\mathcal{F}}$ until some stopping criterion is met (for example, when the leaf sets are smaller than some k).

Exhaustive optimization Optimizing the function $\hat{\mathcal{F}}$ requires searching through the collection of subsets of size $|A| = k$, which, when performed exhaustively, requires $O\left(\binom{n}{k}\right)$ time. An exhaustive approach thus runs in exponential time, for example, when $k \sim O(n)$.

However, when the size of k is known and small ($k \sim O(1)$), the optimal partitioning of an item set can be found in polynomial time by exhaustively evaluating $\hat{\mathcal{F}}$ over all k-subsets. Moreover, the sample complexity in the small-k regime is less than that of Theorem 25 which makes no assumptions on the size of k:

Corollary 26. *Under the conditions of Theorem 25, one needs at most $S(\Delta, \epsilon) \equiv O\big(\frac{n^2}{\epsilon^2} \log^2 \frac{n}{\epsilon} \log \frac{n}{\gamma}\big)$ samples to recover the exact riffle independent partitioning with probability $1 - \gamma$.*

When k is small, we can therefore use exhaustive optimization to learn the structure of k-thin chain models (Section 4) in polynomial time. The structure learning problem for thin chains is to discover how the items are partitioned into groups, which group is inserted first, which group is inserted second, and so on. To learn the structure of a thin chain, we can use exhaustive optimization to learn the topmost partitioning of the item set, then recursively learn a thin chain model for the items in the larger subset.

```
AnchorsPartition
input  : training set $\{\sigma^{(1)}, \dots, \sigma^{(m)}\}$, $k \equiv |A|$
output: Riffle independent partitioning of item set, $(A_{best}, B_{best})$
Fix $a_1$ to be any item ;
forall $a_2 \in \{1, \dots, n\}$, $a_1 \neq a_2$ do
    Estimate $\hat{I}_{x;a_1,a_2}$ for all $x \neq a_1, a_2$;
    $\hat{I}^k \leftarrow k^{th}$ smallest item in $\{\hat{I}_{x;a_1,a_2}; x \neq a_1, a_2\}$ ;
    $A_{a_1,a_2} \leftarrow \{x : \hat{I}_{x;a_1,a_2} \leq \hat{I}^k\}$ ;
end
$A_{best} \leftarrow \arg\min_{a_1,a_2} \hat{\mathcal{F}}(A_{a_1,a_2})$;
$B_{best} \leftarrow \{1, \dots, n\} \backslash A_{best}$ ;
return $[A_{best}, B_{best}]$;
```

Algorithm 1: Pseudocode for partitioning using the *Anchors* method

Handling arbitrary partitions using anchors When k is large, or even unknown, $\hat{\mathcal{F}}$ cannot be optimized using exhaustive methods. Instead, we propose a simple algorithm for finding A and B based on the following observation. If an oracle could identify any two elements of the set A, say, a_1, a_2, in advance, then the quantity $I_{x;a_1,a_2} = I(x; a_1 < a_2)$ indicates whether the item x belongs to A or B since $I_{x;a_1,a_2}$ is nonzero in the first case, and zero in the second case.

For finite training sets, when I is only known approximately, one can sort the set $\{I_{x;a_1,a_2} \,;\, x \neq a_1, a_2\}$ and if k is known, take the k items closest to zero to be the set B (when k is unknown, one can use a threshold to infer k). Since we compare all items against a_1, a_2, we refer to a_1 and a_2 as "anchors".

Of course a_1, a_2 are not known in advance, but by fixing a_1 to be an arbitrary item, one can repeat the above method for all $n - 1$ settings of a_2 to produce a collection of $O(n^2)$ candidate partitions. Each partition is then scored using the approximate objective $\hat{\mathcal{F}}$, and a final partition is selected as the minimum over the candidates. See Algorithm 1. In cases when k is not known, we evaluate partitions for all possible settings of k using $\hat{\mathcal{F}}$.

Since the Anchors method does not require searching over subsets, it can be significantly faster than an exhaustive optimization of $\hat{\mathcal{F}}$. Moreover, by assuming ϵ-third order strong connectivity as in the previous section, one can use similar arguments to derive sample complexity bounds.

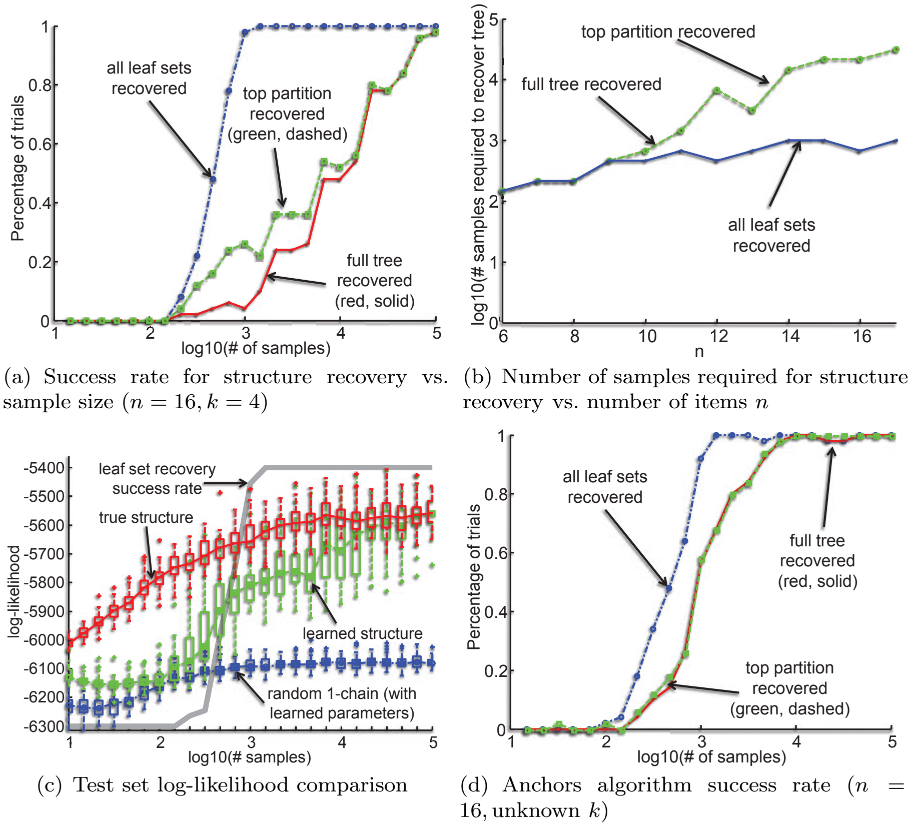
Corollary 27 (of Theorem 25)**.** *Let A and B be ϵ-third order strongly connected riffle independent sets, and suppose $|A| = k$. Given $S(\Delta, \epsilon)$ i.i.d. samples, the output of the Anchors algorithm is exactly $[A, B]$ with probability $1 - \gamma$. In particular, the Anchors estimator is consistent.*

6. Experiments

We have analyzed the APA data extensively throughout the paper. In this section, we demonstrate our algorithms on simulated data as well as other real datasets.

6.1. Simulated data

We first apply our methods to synthetic data to show that, given enough samples, our algorithms *do* effectively recover the optimal hierarchical structures which generated the original datasets. For various settings of n, we simulated data drawn jointly from a k-thin chain model (for $k = 4$) with a random parameter setting for each structure and applied our exact method for learning thin chains to each sample. First, we investigated the effect of varying sample size on the proportion of trials (out of fifty) for which our algorithms were able to (a) recover the underlying tree structure *exactly*, (b) recover the topmost partition correctly, or (c) recover all leaf sets correctly (but possibly out of order). Figure 8(a) shows the result for $n = 16$. Figure 8(b), shows, as a function of n, the number of samples that were required in the same experiments to (a) *exactly* recover the underlying structure or (b) recover the correct leaf sets, for at least 90% of the trials. What we can observe from the plots is that, given enough samples, reliable structure recovery *is* indeed possible. It is also interesting to note that recovery of the correct leaf sets can be done with much fewer samples than are required for recovering the full hierarchical structure of the model.



(a) Success rate for structure recovery vs. sample size ($n = 16, k = 4$)

(b) Number of samples required for structure recovery vs. number of items n

(c) Test set log-likelihood comparison

(d) Anchors algorithm success rate ($n = 16$, unknown k)

Fig 8. *Structure discovery experiments on synthetic data.*

After learning a structure for each sample, we learned model parameters and evaluated the log-likelihood of each model on 200 test examples drawn from the true distributions. In Figure 8(c), we compare log-likelihoods when (a) the true structure is given, (b) a k-thin chain is learned with known k, and (c) when we use a randomly generated 1-chain structure. As expected, knowing the structure results in the best performance, and the 1-chain is overconstrained. However, our structure learning algorithm is eventually able to match the performance of the true structure given enough samples. It is also interesting to note that the jump in performance at the halfway point in the plot coincides with the jump in the success rate of discovering all leaf sets correctly — we conjecture that performance is sometimes less sensitive to the hierarchy, as long as the leaf sets have been correctly discovered.

To test the Anchors algorithm, we ran the same simulation using Algorithm 1 on data drawn from hierarchical models with no fixed k. We generated roughly balanced structures, meaning that item sets were recursively partitioned into (almost) equally sized subsets at each level of the hierarchy. From Figure 8(d), we see that the Anchors algorithm can also discover the true structure given enough samples. Interestingly, the difference in sample complexity for discovering leaf sets versus discovering the full tree is not nearly as pronounced as in Figure 8(a). We believe that this is due to the fact that the balanced trees have less depth than the thin chains, leading to fewer opportunities for our greedy top-down approach to commit errors.

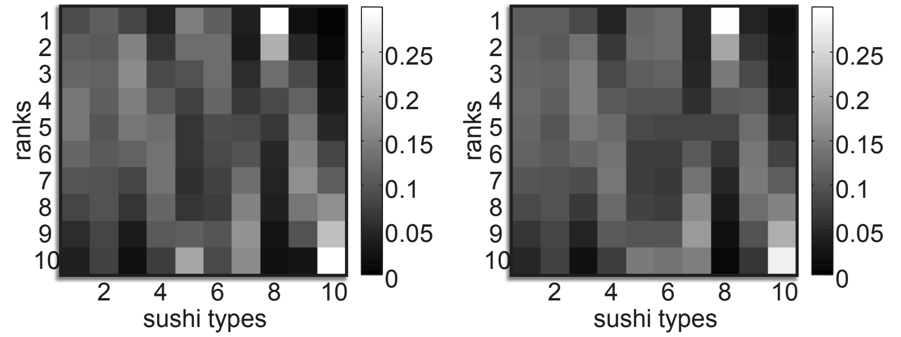
6.2. Data analysis: Sushi preference data

We now turn to analyzing real datasets. For our first analysis, we examine a sushi preference ranking dataset [21] consisting of 5000 full rankings of ten types of sushi. The items are enumerated in Figure 9. Note that, compared to the APA election data, the sushi dataset has twice as many items, but fewer examples.

Structure learning on the sushi dataset Figure 12 shows the hierarchical structure that we learn using the entire sushi dataset. Since the sushi are not prepartitioned into distinct coalitions, it is somewhat more difficult than with, say, the APA data, to interpret whether the estimated structure makes sense. However, parts of the tree certainly seem like reasonable groupings. For example, all of the tuna related sushi types have been clustered together. Tamago and kappa-maki (egg and cucumber rolls) are "safer", typically more boring choices, while uni and sake (sea urchin and salmon roe) are more daring. Anago (sea

1. ebi (shrimp)	2. anago (sea eel)	3. maguro (tuna)
4. ika (squid)	5. uni (sea urchin)	6. sake (salmon roe)
7. tamago (egg)	8. toro (fatty tuna)	9. tekka-maki (tuna roll)
	10. kappa-maki (cucumber roll)	

Fig 9. *List of sushi types in the [21] dataset.*



(a) First-order marginals of Sushi preference data

(b) Riffle independent approximation of first-order marginals with learned hierarchy

Fig 10. *Sushi first order marginals.*

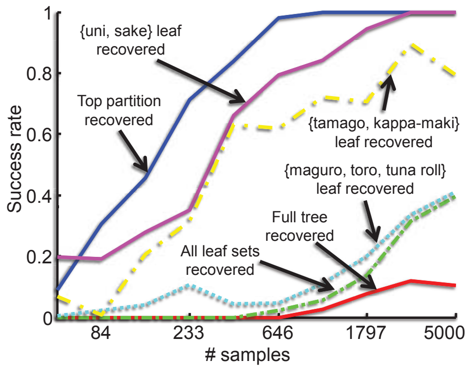


Fig 11. *Stability of bootstrapped tree 'features' of the sushi dataset.*

eel), is the odd man out in the estimated hierarchy, being partitioned away from the remaining items at the top of the tree.

To understand the behavior of our algorithm with smaller sample sizes, we looked for features of the tree from Figure 12 which remained stable even when learning with smaller sample sizes. Figure 11 summarizes the results of our bootstrap analysis for the sushi dataset, in which we resample from the original training set 200 times at each of different sample sizes and plot the proportion of learned hierarchies which, (a) recover 'sea eel' as the topmost partition, (b) recover all leaf sets correctly, (c), recover the entire tree correctly, (d) recover the tuna-related sushi leaf set, (e) recover the {tamago, kappa-maki} leaf set, and (f) recover the {uni, sake} leaf set.

6.3. Data analysis: Irish election data

We next applied our algorithms to a larger Irish House of Parliament (Dáil Éireann) election dataset from the Meath constituency in Ireland. The Dáil

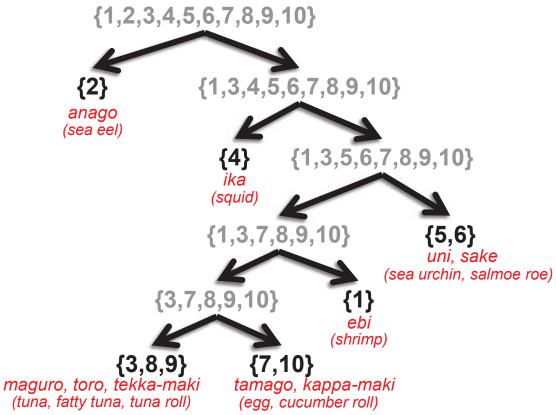


Fig 12. *Learned hierarchy for sushi dataset using all 5000 rankings.*

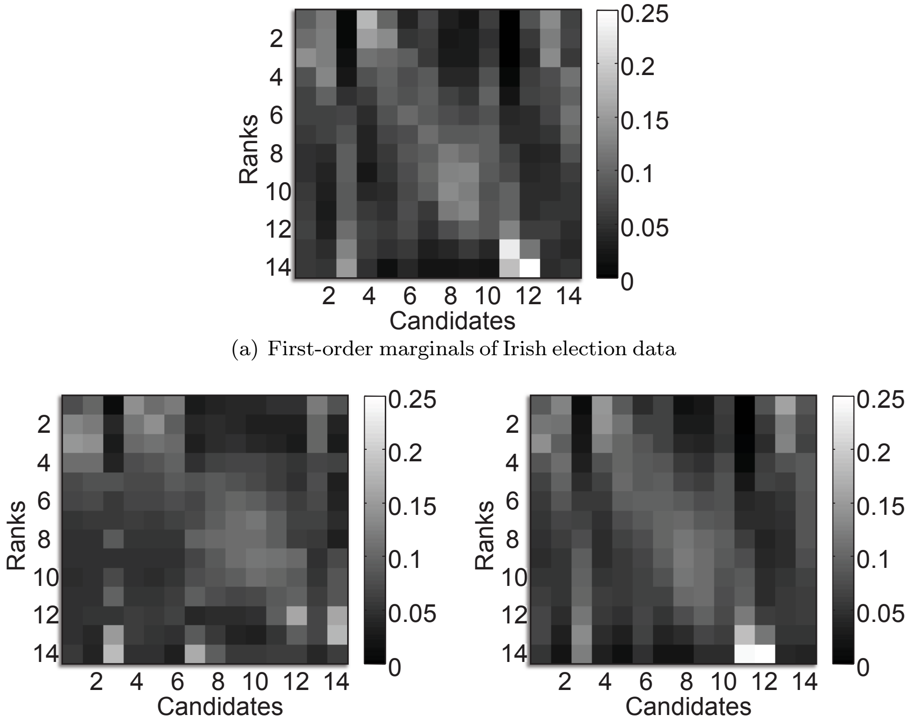
	Candidate	Party
1	Brady, J.	Fianna Fáil
2	Bruton, J.	Fine Gael
3	Colwell, J.	Independent
4	Dempsey, N.	Fianna Fáil
5	English, D.	Fine Gael
6	Farrelly, J.	Fine Gael
7	Fitzgerald, B.	Independent
8	Kelly, T.	Independent
9	O'Brien, P.	Independent
10	O'Byrne, F.	Green Party
11	Redmond, M.	Christian Solidarity
12	Reilly, J.	Sinn Féin
13	Wallace, M.	Fianna Fáil
14	Ward, P.	Labour

Fig 13. *List of candidates from the Meath constituency election in 2002 for five seats in the Dáil Éireann (reproduced from [12]).*

Éireann uses the *single transferable vote* (STV) election system, in which voters rank candidates. In the Meath constituency, there were 14 candidates in the 2002 election, running for five allotted seats. The candidates identified with the two major rival political parties, Fianna Fáil and Fine Gael, as well as a number of smaller parties (Figure 13). See [12] for more election details as well as an alternative analysis. In our experiments, we used a subset of roughly 2500 fully ranked ballots from the election.

To summarize the dataset, Figure 14(a) shows the estimated first-order marginals. Candidates $\{1, 2, 4, 5, 6, 13\}$ form the set of "major" party candidates belonging to either Fianna Fáil or Fine Gael, and as shown, fared much better in the election than the minor party candidates. Notably, candidates 11 and 12 (belonging to the Christian Solidary Party and Sinn Féin, respectively) received on average, the lowest ranks in the election. One of the differences between the two candidates, however, is that a significant portion of the electorate also ranked the Sinn Féin candidate high.

Though it may not be clear how one might partition the candidates, a natural idea might be to assume that the major party candidates (A) are riffle independent of the minor party candidates (B). In Figure 14(b), we show the first-order marginals corresponding to an approximation in which A and B are riffle independent. Visually, the approximate marginals can be seen to be roughly similar



(a) First-order marginals of Irish election data

(b) Riffle independent approximation of first-order marginals with A = {Fianna Fáil, Fine Gael}, and B = {everything else}

(c) Riffle independent approximation of first-order marginals with learned hierarchy in Figure 15

Fig 14. *Irish first order marginals.*

to the exact marginals, however there are significant features of the matrix which are not captured by the approximation — for example, the columns belonging to candidates 11 and 12 are not well approximated. In Figure 14(c), we plot the approximation corresponding to a learned hierarchy, which we discuss next. As can be seen, the marginals obtained via structure learning are visually much closer to the exact marginals.

Structure discovery on the Irish election data As with the APA data, both the exhaustive optimization of $\hat{\mathcal{F}}$ and the Anchors algorithm returned the same tree, with running times of 69.7 seconds and 2.1 seconds respectively (not including the 3.1 seconds required for precomputations). The resulting tree is shown (only up to depth 4), in Figure 15. As expected, the candidates belonging to the two major parties, Fianna Fáil and Fine Gael, are neatly partitioned into their own leaf sets. The topmost leaf is the Sinn Fein candidate, indicating that voters tended to insert him into the ranking independently of all of the other 13 candidates.

To understand our algorithm with smaller sample sizes, we looked for features of the tree from Figure 15 which remained stable even when learning with smaller

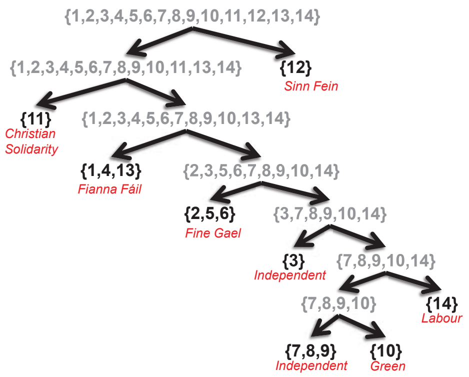
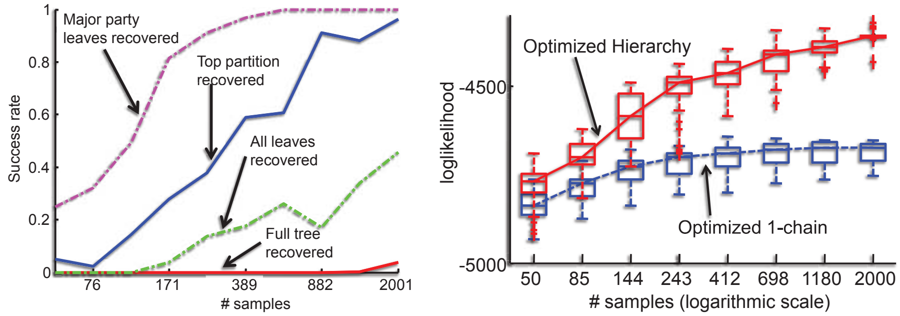


Fig 15. *Learned hierarchy for Irish Election dataset using all 2500 ballots.*



(a) Stability of bootstrapped tree 'features' of the Irish dataset

(b) Log-likelihood of held out test examples from the Irish dataset using optimized structures

Fig 16. *Structure Discovery Experiments: Irish Election dataset.*

sample sizes. In Figure 16(a), we resample from the original training set 200 times at different sample sizes and plot the proportion of learned hierarchies which, (a) recover the Sinn Fein candidate as the topmost leaf, (b) partition the two major parties into leaf sets, and (c) agree with the original tree on all leaf sets, and (d) recover the entire tree. Note that while the dataset is insufficient to support the full tree structure, even with about 100 training examples, candidates belonging to the major parties are consistently grouped, indicating strong party influence in the election.

We compared the results between learning a general hierarchy (without fixed k) and learning a 1-thin chain model on the Irish data. Figure 16(b) shows the log-likelihoods achieved by both models on a held-out test set as the training set size increases. For each training set size, we subsampled the Irish dataset 100 times to produce confidence intervals. Again, even with small sample sizes, the hierarchy outperforms the 1-chain and continually improves with more training data. One might think that the hierarchical models, which use more parameters are prone to overfitting, but in practice, the models learned by our algorithm devote extra parameters towards modeling correlations among the two major parties. As our results suggest, such intraparty correlations are crucial for achieving good modeling performance.

7. Conclusions

Exploiting independence structure for efficient inference and low sample complexity is a simple yet powerful idea, pervasive throughout the machine learning literature, showing up in the form of Bayesian networks, Markov random fields, and more. For rankings, independence can be problematic due to mutual exlusivity constraints, and we began our paper by indicating a need for a useful generalization of independence.

The main contribution of our paper is the definition of such a generalized notion, namely, riffled independence. There are a number of natural questions that immediately follow any such definition, such as:

- Does the generalization retain any of the computational advantages of probabilistic independence?
- Can we find evidence that such generalized independence relations hold (or approximately hold) in real datasets?
- If subsets of items in a ranking dataset indeed satisfy the generalized independence assumption, or approximately so, how could we algorithmically determine what these subsets should be from samples?

We have shown that for riffled independence, the answer to each of the above questions lies in the affirmative. We next explored hierarchical riffle independent decompositions. Our model, in which riffle independent subsets are recursively chained together, leads to a simple, interpretable model whose structure we can estimate from data, and we have successfully applied our learning algorithms to several real datasets.

Currently, the success of our structure learning methods depends on the existence of a sizeable dataset of full rankings. However, ranking datasets are more typically composed of partial or incomplete rankings, which are easier to elicit from users. For example, top-k type rankings, or even rating data (in which a user/judge provides a rating of an item between, say, 1 and 5) are common. Extending our learning algorithms for handling such partially ranked data would be a valuable and practical extension of our work. For structure learning, our tripletwise mutual information measures can already potentially be estimated within a top-k ranking setting. It would be interesting to also develop methods for estimating these mutual information measures from other forms of partial rankings. Additionally, the effect of using partial rankings on structure learning sample complexity is not yet understood, and the field would benefit from a careful analysis.

Riffled independence is a new tool for analyzing rankings and has the potential to give new insights into ranking datasets. We believe that it will be crucial in developing fast and efficient inference and learning procedures for rankings, and perhaps other forms of permutation data.

Acknowledgements

This work is supported in part by the ONR under MURI N000140710747, and the Young Investigator Program grant N00014-08-1-0752. We thank Khalid El-Arini for feedback on initial drafts, and Brendan Murphy and Claire Gormley for providing the Irish voting datasets. Discussions with Marina Meila provided valuable initial ideas upon which this work is based. Finally, we thank our reviewers for thoughful comments and suggestions for improving the quality of our presentation.

Supplementary Material

Uncovering the Riffled Independence Structure of Ranked Data: Supplementary Material
(doi: 10.1214/12-EJS670SUPP; .pdf).

References

[1] Bach, F. R. and Jordan, M. I. (2001). Thin Junction Trees. In *Advances in Neural Information Processing Systems 14* 569–576. MIT Press.

[2] Bayer, D. and Diaconis, P. (1992). Trailing the Dovetail Shuffle to its Lair. *The Annals of Probability* **2** 294–313. MR1161056

[3] Chechetka, A. and Guestrin, C. (2008). Efficient Principled Learning of Thin Junction Trees. In *Advances in Neural Information Processing Systems 20* (J. Platt, D. Koller, Y. Singer and S. Roweis, eds.) 273–280. MIT Press, Cambridge, MA.

[4] Chen, H., Branavan, S. R. K., Barzilay, R. and Karger, D. R. (2009). Global models of document structure using latent permutations. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics. NAACL '09* 371–379. Association for Computational Linguistics, Stroudsburg, PA, USA.

[5] Clausen, M. and Baum, U. (1993). Fast Fourier Transforms for Symmetric Groups: Theory and Implementation. *Mathematics of Computations* **61** 833-847. MR1192969

[6] Diaconis, P. (1988). *Group Representations in Probability and Statistics.* Institute of Mathematical Sciences (Lecture Notes). MR0964069

[7] Diaconis, P. (1989). A Generalization of Spectral Analysis with Application to Ranked Data. *The Annals of Statistics* **17** 949-979. MR1015133

[8] Farias, V., Jagabathula, S. and Shah, D. (2009). A Data-Driven Approach to Modeling Choice. In *Advances in Neural Information Processing Systems 22* (Y. Bengio, D. Schuurmans, J. Lafferty, C. K. I. Williams and A. Culotta, eds.) 504–512.

[9] Fligner, M. and Verducci, J. (1986). Distance-based Ranking models. *Journal of the Royal Statistical Society, Series B* **83** 859-869. MR0876847

[10] Fligner, M. and Verducci, J. (1988). Multistage Ranking Models. *Journal of the American Statistical Association* **83**. MR0963820

[11] Gallo, G., Longo, G., Pallottino, S. and Nguyen, S. (1993). Directed hypergraphs and applications. *Discrete Applied Mathematics* **42** 177-201. MR1217096

[12] Gormley, I. C. and Murphy, T. B. (2007). A latent space model for rank data. In *Proceedings of the 24th Annual International Conference on Machine Learning. ICML'06* 90–102. ACM, New York, NY, USA.

[13] Guiver, J. and Snelson, E. (2009). Bayesian inference for Plackett-Luce ranking models. In *Proceedings of the 26th Annual International Conference on Machine Learning. ICML '09* 377–384. ACM, New York, NY, USA.

[14] Höffgen, K.-U. (1993). Learning and robust learning of product distributions. In *Proceedings of the sixth annual conference on Computational learning theory. COLT '93* 77–83. ACM, New York, NY, USA.

[15] Huang, J. and Guestrin, C. (2009a). Riffled Independence for Ranked Data. In *Advances in Neural Information Processing Systems 22* (Y. Bengio, D. Schuurmans, J. Lafferty, C. K. I. Williams and A. Culotta, eds.) 799–807.

[16] Huang, J., Guestrin, C. and Guibas, L. (2009b). Fourier Theoretic Probabilistic Inference over Permutations. *Journal of Machine Learning (JMLR)* **10** 997-1070. MR2520800

[17] Huang, J. and Guestrin, C. (2010). Learning Hierarchical Riffle Independent Groupings from Rankings. In *Proceedings of the 27th Annual International Conference on Machine Learning. ICML '10* 455–462.

[18] Huang, J., Guestrin, C., Jiang, X. and Guibas, L. J. (2009). Exploiting Probabilistic Independence for Permutations. *Journal of Machine Learning Research - Proceedings Track* **5** 248-255.

[19] Huang, J. and Guestrin, C. (2012). Uncovering the Riffled Independence Structure of Ranked Data: Supplementary Material. DOI: 10.1214/12-EJS670SUPP

[20] Jagabathula, S. and Shah, D. (2009). Inferring rankings under constrained sensing. In *Advances in Neural Information Processing Systems 21* (D. Koller, D. Schuurmans, Y. Bengio and L. Bottou, eds.) 753–760.

[21] Kamishima, T. (2003). Nantonac collaborative filtering: recommendation based on order responses. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining. KDD '03* 583–588. ACM, New York, NY, USA.

[22] Koller, D. and Friedman, N. (2009). *Probabilistic Graphical Models: Principles and Techniques*. MIT Press. MR2778120

[23] Kondor, R. (2008). Group Theoretical Methods in Machine Learning PhD thesis, Columbia University.

[24] Mallows, C. (1957). Non-null ranking models. *Biometrika* **44** 114-130. MR0087267

[25] Marden, J. I. (1995). *Analyzing and Modeling Rank Data*. Chapman & Hall. MR1346107

[26] Maslen, D. (1998). The efficient computation of Fourier transforms on the Symmetric group. *Mathematics of Computation* **67** 1121-1147. MR1468943

[27] Meila, M., Phadnis, K., Patterson, A. and Bilmes, J. (2007). Consensus ranking under the exponential model Technical Report No. UW TR-515.

[28] Motwani, R. and Raghavan, P. (1996). Randomized algorithms. *ACM Computational Surveys* **28**. MR1344451

[29] Plackett, R. (1975). The analysis of permutations. *Applied Statistics* **24** 193-202. MR0391338

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

[31] Rockmore, D. N. (2000). The FFT: An Algorithm the Whole Family Can Use. *Computing in Science and Engineering* **02** 60-64.

[32] Shahaf, D., Chechetka, A. and Guestrin, C. (2009). Learning Thin Junction Trees via Graph Cuts. *Journal of Machine Learning Research - Proceedings Track* **5** 113-120.

[33] Shin, J., Lee, N., Thrun, S. and Guibas, L. (2005). Lazy inference on object identities in wireless sensor networks. In *Proceedings of the 4th international symposium on Information processing in sensor networks. IPSN '05*. IEEE Press, Piscataway, NJ, USA.

[34] Sun, M., Lebanon, G. and Collins-Thompson, K. (2010). Visualizing differences in web search algorithms using the expected weighted hoeffding distance. In *Proceedings of the 19th international conference on World wide web. WWW '10* 931–940. ACM, New York, NY, USA.

[35] Terras, A. (1999). *Fourier Analysis on Finite Groups and Applications.* London Mathematical Society. MR1695775

[36] Thurstone, L. (1927). A law of comparative judgement. *Psychological Review* **34** 273-286.