Riffled Independence for Ranked Data

Jonathan Huang and Carlos Guestrin

NIPS, 2009

Zachary Schutzman

University of Pennsylvania

February 3, 2017

Presentation Outline

Introduction

Introduction

Distributions over permutations are important!

Introduction

Distributions over permutations are important!

- Object tracking

Introduction

Distributions over permutations are important!

- Object tracking
- **Ranked data**

Introduction

Distributions over permutations are important!

- Object tracking
- **Ranked data**

But, there are computational challenges

- Storage complexity

Introduction

Distributions over permutations are important!

- Object tracking
- **Ranked data**

But, there are computational challenges

- Storage complexity
- Time complexity

Introduction

Distributions over permutations are important!

- Object tracking
- **Ranked data**

But, there are computational challenges

- Storage complexity
- Time complexity

Can we exploit properties of the distributions to reduce these difficulties?

Where we're going:

- Probabilistic vs Riffle Independence

Where we're going:

- Probabilistic vs Riffle Independence
- Riffle Distributions

Where we're going:

- Probabilistic vs Riffle Independence
- Riffle Distributions
- Fourier Domain Algorithms

Presentation Outline

Introduction

Some Math

Distributions on Permutations and Independence

Fourier Domain Algorithms

Questions and discussion

The Symmetric Group

The Symmetric Group

The set of permutations on n objects forms the group S_n

The Symmetric Group

The set of permutations on n objects forms the group S_n

- The composition of permutations is a permutation

The Symmetric Group

The set of permutations on n objects forms the group S_n

- The composition of permutations is a permutation
- Every permutation has a (unique) inverse permutation

The Symmetric Group

The set of permutations on n objects forms the group S_n

- The composition of permutations is a permutation
- Every permutation has a (unique) inverse permutation

$|S_n| = n!$

Composing permutations

This is a very straightforward process, but it's super important!

Composing permutations

This is a very straightforward process, but it's super important!

Let's take two permutations in S_5, $\sigma = [[34125]]$ and $\tau = [[54231]]$

Composing permutations

This is a very straightforward process, but it's super important!

Let's take two permutations in S_5, $\sigma = [[34125]]$ and $\tau = [[54231]]$

Writing '$\sigma\tau$' means 'first do τ, then do σ', like $g(f(x))$.

Composing permutations

This is a very straightforward process, but it's super important!

Let's take two permutations in S_5, $\sigma = [[34125]]$ and $\tau = [[54231]]$

Writing '$\sigma\tau$' means 'first do τ, then do σ', like $g(f(x))$.

Computing $\sigma\tau$ is really easy:

Composing permutations

This is a very straightforward process, but it's super important!

Let's take two permutations in S_5, $\sigma = [[34125]]$ and $\tau = [[54231]]$

Writing '$\sigma\tau$' means 'first do τ, then do σ', like $g(f(x))$.

Computing $\sigma\tau$ is really easy:

- $1 \xrightarrow{\tau} 5 \xrightarrow{\sigma} 5$

Composing permutations

This is a very straightforward process, but it's super important!

Let's take two permutations in S_5, $\sigma = [[34125]]$ and $\tau = [[54231]]$

Writing '$\sigma\tau$' means 'first do τ, then do σ', like $g(f(x))$.

Computing $\sigma\tau$ is really easy:

- $1 \xrightarrow{\tau} 5 \xrightarrow{\sigma} 5 \Rightarrow [[xxxx1]]$

Composing permutations

This is a very straightforward process, but it's super important!

Let's take two permutations in S_5, $\sigma = [[34125]]$ and $\tau = [[54231]]$

Writing '$\sigma\tau$' means 'first do τ, then do σ', like $g(f(x))$.

Computing $\sigma\tau$ is really easy:

- $1 \xrightarrow{\tau} 5 \xrightarrow{\sigma} 5 \Rightarrow [[xxxx1]]$
- $2 \xrightarrow{\tau} 3 \xrightarrow{\sigma} 1$

Composing permutations

This is a very straightforward process, but it's super important!

Let's take two permutations in S_5, $\sigma = [[34125]]$ and $\tau = [[54231]]$

Writing '$\sigma\tau$' means 'first do τ, then do σ', like $g(f(x))$.

Computing $\sigma\tau$ is really easy:

- $1 \xrightarrow{\tau} 5 \xrightarrow{\sigma} 5 \Rightarrow [[xxxx1]]$
- $2 \xrightarrow{\tau} 3 \xrightarrow{\sigma} 1 \Rightarrow [[2xxx1]]$

Composing permutations

This is a very straightforward process, but it's super important!

Let's take two permutations in S_5, $\sigma = [[34125]]$ and $\tau = [[54231]]$

Writing '$\sigma\tau$' means 'first do τ, then do σ', like $g(f(x))$.

Computing $\sigma\tau$ is really easy:

- $1 \xrightarrow{\tau} 5 \xrightarrow{\sigma} 5 \Rightarrow [[xxxx1]]$
- $2 \xrightarrow{\tau} 3 \xrightarrow{\sigma} 1 \Rightarrow [[2xxx1]]$
- $3 \xrightarrow{\tau} 4 \xrightarrow{\sigma} 2$

Composing permutations

This is a very straightforward process, but it's super important!

Let's take two permutations in S_5, $\sigma = [[34125]]$ and $\tau = [[54231]]$

Writing '$\sigma\tau$' means 'first do τ, then do σ', like $g(f(x))$.

Computing $\sigma\tau$ is really easy:

- $1 \xrightarrow{\tau} 5 \xrightarrow{\sigma} 5 \Rightarrow [[xxxx1]]$
- $2 \xrightarrow{\tau} 3 \xrightarrow{\sigma} 1 \Rightarrow [[2xxx1]]$
- $3 \xrightarrow{\tau} 4 \xrightarrow{\sigma} 2 \Rightarrow [[23xx1]]$

Composing permutations

This is a very straightforward process, but it's super important!

Let's take two permutations in S_5, $\sigma = [[34125]]$ and $\tau = [[54231]]$

Writing '$\sigma\tau$' means 'first do τ, then do σ', like $g(f(x))$.

Computing $\sigma\tau$ is really easy:

- $1 \xrightarrow{\tau} 5 \xrightarrow{\sigma} 5 \Rightarrow [[xxxx1]]$
- $2 \xrightarrow{\tau} 3 \xrightarrow{\sigma} 1 \Rightarrow [[2xxx1]]$
- $3 \xrightarrow{\tau} 4 \xrightarrow{\sigma} 2 \Rightarrow [[23xx1]]$
- $4 \xrightarrow{\tau} 2 \xrightarrow{\sigma} 4$

Composing permutations

This is a very straightforward process, but it's super important!

Let's take two permutations in S_5, $\sigma = [[34125]]$ and $\tau = [[54231]]$

Writing '$\sigma\tau$' means 'first do τ, then do σ', like $g(f(x))$.

Computing $\sigma\tau$ is really easy:

- $1 \xrightarrow{\tau} 5 \xrightarrow{\sigma} 5 \Rightarrow [[xxxx1]]$
- $2 \xrightarrow{\tau} 3 \xrightarrow{\sigma} 1 \Rightarrow [[2xxx1]]$
- $3 \xrightarrow{\tau} 4 \xrightarrow{\sigma} 2 \Rightarrow [[23xx1]]$
- $4 \xrightarrow{\tau} 2 \xrightarrow{\sigma} 4 \Rightarrow [[23x41]]$

Composing permutations

This is a very straightforward process, but it's super important!

Let's take two permutations in S_5, $\sigma = [[34125]]$ and $\tau = [[54231]]$

Writing '$\sigma\tau$' means 'first do τ, then do σ', like $g(f(x))$.

Computing $\sigma\tau$ is really easy:

- $1 \xrightarrow{\tau} 5 \xrightarrow{\sigma} 5 \Rightarrow [[xxxx1]]$
- $2 \xrightarrow{\tau} 3 \xrightarrow{\sigma} 1 \Rightarrow [[2xxx1]]$
- $3 \xrightarrow{\tau} 4 \xrightarrow{\sigma} 2 \Rightarrow [[23xx1]]$
- $4 \xrightarrow{\tau} 2 \xrightarrow{\sigma} 4 \Rightarrow [[23x41]]$
- $5 \xrightarrow{\tau} 1 \xrightarrow{\sigma} 3$

Composing permutations

This is a very straightforward process, but it's super important!

Let's take two permutations in S_5, $\sigma = [[34125]]$ and $\tau = [[54231]]$

Writing '$\sigma\tau$' means 'first do τ, then do σ', like $g(f(x))$.

Computing $\sigma\tau$ is really easy:

- $1 \xrightarrow{\tau} 5 \xrightarrow{\sigma} 5 \Rightarrow [[xxxx1]]$
- $2 \xrightarrow{\tau} 3 \xrightarrow{\sigma} 1 \Rightarrow [[2xxx1]]$
- $3 \xrightarrow{\tau} 4 \xrightarrow{\sigma} 2 \Rightarrow [[23xx1]]$
- $4 \xrightarrow{\tau} 2 \xrightarrow{\sigma} 4 \Rightarrow [[23x41]]$
- $5 \xrightarrow{\tau} 1 \xrightarrow{\sigma} 3 \Rightarrow [[23541]]$

Representations of the Symmetric Group

Group elements aren't easy to use in computations

Representations of the Symmetric Group

Group elements aren't easy to use in computations

But matrices (vector space elements) are!

Representations of the Symmetric Group

Group elements aren't easy to use in computations

But matrices (vector space elements) are!

A **group representation** is a map $\rho : G \to GL(k)$ which preserves the group structure
$\rho(g)\rho(h) = \rho(gh)$

Representations of the Symmetric Group

Group elements aren't easy to use in computations

But matrices (vector space elements) are!

A **group representation** is a map $\rho : G \to GL(k)$ which preserves the group structure
$\rho(g)\rho(h) = \rho(gh)$

We can always make a new representation by taking the direct sum of two other representations.

Representations of the Symmetric Group

Group elements aren't easy to use in computations

But matrices (vector space elements) are!

A **group representation** is a map $\rho : G \to GL(k)$ which preserves the group structure
$\rho(g)\rho(h) = \rho(gh)$

We can always make a new representation by taking the direct sum of two other representations.
An **irreducible representation** is one that cannot be written as the direct sum of two others.

Example: Representations of S_3

π

Example: Representations of S_3

π
123
213
132
321
231
312

Example: Representations of S_3

π	g
123	()
213	(12)
132	(23)
321	(13)
231	(132)
312	(123)

Example: Representations of S_3

π	g	ρ_{triv}
123	()	1
213	(12)	1
132	(23)	1
321	(13)	1
231	(132)	1
312	(123)	1

Example: Representations of S_3

π	g	ρ_{triv}	ρ_{sgn}
123	$()$	1	1
213	(12)	1	-1
132	(23)	1	-1
321	(13)	1	-1
231	(132)	1	1
312	(123)	1	1

Example: Representations of S_3

π	g	ρ_{triv}	ρ_{sgn}	ρ_{std}
123	$()$	1	1	$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$
213	(12)	1	-1	$\begin{pmatrix} -1 & 1 \\ 0 & 1 \end{pmatrix}$
132	(23)	1	-1	$\begin{pmatrix} 1 & 0 \\ 1 & -1 \end{pmatrix}$
321	(13)	1	-1	$\begin{pmatrix} 0 & -1 \\ -1 & 0 \end{pmatrix}$
231	(132)	1	1	$\begin{pmatrix} -1 & 1 \\ -1 & 0 \end{pmatrix}$
312	(123)	1	1	$\begin{pmatrix} 0 & -1 \\ 1 & -1 \end{pmatrix}$

Example: Representations of S_3

π	g	ρ_{triv}	ρ_{sgn}	ρ_{std}
123	$()$	1	1	$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$
213	(12)	1	-1	$\begin{pmatrix} -1 & 1 \\ 0 & 1 \end{pmatrix}$
132	(23)	1	-1	$\begin{pmatrix} 1 & 0 \\ 1 & -1 \end{pmatrix}$
321	(13)	1	-1	$\begin{pmatrix} 0 & -1 \\ -1 & 0 \end{pmatrix}$
231	(132)	1	1	$\begin{pmatrix} -1 & 1 \\ -1 & 0 \end{pmatrix}$
312	(123)	1	1	$\begin{pmatrix} 0 & -1 \\ 1 & -1 \end{pmatrix}$

Example: Representations of S_3

π	g	ρ_{triv}	ρ_{sgn}	ρ_{std}
123	()	1	1	$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$
213	(12)	1	-1	$\begin{pmatrix} -1 & 1 \\ 0 & 1 \end{pmatrix}$
132	(23)	1	-1	$\begin{pmatrix} 1 & 0 \\ 1 & -1 \end{pmatrix}$
321	(13)	1	-1	$\begin{pmatrix} 0 & -1 \\ -1 & 0 \end{pmatrix}$
231	(132)	1	1	$\begin{pmatrix} -1 & 1 \\ -1 & 0 \end{pmatrix}$
312	(123)	1	1	$\begin{pmatrix} 0 & -1 \\ 1 & -1 \end{pmatrix}$

Are there any other irreducible representations?

The Fourier Transform

Given a function $f : G \to \mathbb{R}$, the Fourier transform at a representation ρ is

The Fourier Transform

Given a function $f : G \to \mathbb{R}$, the Fourier transform at a representation ρ is

$$\hat{f}_\rho = \sum_{g \in G} f(g)\rho(g)$$

The Fourier Transform

Given a function $f : G \to \mathbb{R}$, the Fourier transform at a representation ρ is
$\hat{f}_\rho = \sum_{g \in G} f(g)\rho(g)$

$\hat{f}_\rho$ is a matrix of the same dimension as ρ.

The Fourier Transform

Given a function $f : G \to \mathbb{R}$, the Fourier transform at a representation ρ is
$\hat{f}_\rho = \sum_{g \in G} f(g)\rho(g)$

$\hat{f}_\rho$ is a matrix of the same dimension as ρ.

The family of $\hat{f}_\rho$ corresponding to the irreducible representations of G systematically encode f.

Presentation Outline

Distributions

Keeping the idea of the Fourier transform in the back of our minds...

Distributions

Keeping the idea of the Fourier transform in the back of our minds...

A distribution on S_n is a function $h : S_n \to \mathbb{R}^+$ such that $\sum_{\sigma \in S_n} h(\sigma) = 1$.

Distributions

Keeping the idea of the Fourier transform in the back of our minds...

A distribution on S_n is a function $h : S_n \to \mathbb{R}^+$ such that $\sum_{\sigma \in S_n} h(\sigma) = 1$.

To characterize an arbitrary distribution we need $n!$ parameters.

Distributions

Keeping the idea of the Fourier transform in the back of our minds...

A distribution on S_n is a function $h : S_n \to \mathbb{R}^+$ such that $\sum_{\sigma \in S_n} h(\sigma) = 1$.

To characterize an arbitrary distribution we need $n!$ parameters.

What if we have two distinct 'classes' of objects? Denote $X = \{1, 2, \ldots, p\}$ and $\overline{X} = \{p+1, \ldots, n\}$. Let's call $q = n - p$.

Probabilistic Independence

Let f be a distribution over S_p, corresponding to permutations on X and g the same for $\overline{X}$ over S_q.

Probabilistic Independence

Let f be a distribution over S_p, corresponding to permutations on X and g the same for $\overline{X}$ over S_q.

We say that h is **probabilistically independent** if we can write $h(\sigma) = f(\sigma_p) \cdot g(\sigma_q)$.

Probabilistic Independence

Let f be a distribution over S_p, corresponding to permutations on X and g the same for $\overline{X}$ over S_q.

We say that h is **probabilistically independent** if we can write $h(\sigma) = f(\sigma_p) \cdot g(\sigma_q)$.

If h does factor this way, why is that nice?

Probabilistic Independence

Let f be a distribution over S_p, corresponding to permutations on X and g the same for $\overline{X}$ over S_q.

We say that h is **probabilistically independent** if we can write $h(\sigma) = f(\sigma_p) \cdot g(\sigma_q)$.

If h does factor this way, why is that nice?
We can know h with only $p! + q!$ parameters.

Probabilistic Independence

Let f be a distribution over S_p, corresponding to permutations on X and g the same for $\overline{X}$ over S_q.

We say that h is **probabilistically independent** if we can write $h(\sigma) = f(\sigma_p) \cdot g(\sigma_q)$.

If h does factor this way, why is that nice?
We can know h with only $p! + q!$ parameters.

What is the problem with this?

Probabilistic Independence

Let f be a distribution over S_p, corresponding to permutations on X and g the same for $\overline{X}$ over S_q.

We say that h is **probabilistically independent** if we can write $h(\sigma) = f(\sigma_p) \cdot g(\sigma_q)$.

If h does factor this way, why is that nice?
We can know h with only $p! + q!$ parameters.

What is the problem with this?

It doesn't allow any expression of preferences between the two classes!

Interleavings

What if we allow an interleaving?

A **(p,q)-interleaving** is a permutation in S_n that preserves the relative ordering within the p- and q-subsets. We will call the set of all such interleavings $\Omega_{p,q}$

Interleavings

What if we allow an interleaving?

A **(p,q)-interleaving** is a permutation in S_n that preserves the relative ordering within the p- and q-subsets. We will call the set of all such interleavings $\Omega_{p,q}$

For example, if $\sigma_p = [[1342]]$ and $\sigma_q = [[6875]]$, then one possible interleaving is [[13648725]], corresponding to the (p,q)-interleaving $\tau = [[12536748]]$.

Interleavings

What if we allow an interleaving?

A **(p,q)-interleaving** is a permutation in S_n that preserves the relative ordering within the p- and q-subsets. We will call the set of all such interleavings $\Omega_{p,q}$

For example, if $\sigma_p = [[1342]]$ and $\sigma_q = [[6875]]$, then one possible interleaving is [[13648725]], corresponding to the (p,q)-interleaving $\tau = [[12536748]]$.

Does fixing an interleaving fix our problem?

Interleavings

What if we allow an interleaving?

A **(p,q)-interleaving** is a permutation in S_n that preserves the relative ordering within the p- and q-subsets. We will call the set of all such interleavings $\Omega_{p,q}$

For example, if $\sigma_p = [[1342]]$ and $\sigma_q = [[6875]]$, then one possible interleaving is [[13648725]], corresponding to the (p,q)-interleaving $\tau = [[12536748]]$.

Does fixing an interleaving fix our problem?
Not really.

Riffle Distributions

What if we allow a *distribution* over the interleavings?

Riffle Distributions

What if we allow a *distribution* over the interleavings?

Let $m_{p,q}$ be a distribution over $\Omega_{p,q}$. Equivalently, $m_{p,q}$ is a distribution over S_n which assigns probability zero to elements not in $\Omega_{p,q}$.

Riffle Distributions

What if we allow a *distribution* over the interleavings?

Let $m_{p,q}$ be a distribution over $\Omega_{p,q}$. Equivalently, $m_{p,q}$ is a distribution over S_n which assigns probability zero to elements not in $\Omega_{p,q}$.

The **uniform riffle distribution**, denoted $m_{p,q}^{unif}$ is the one which assigns probability $\frac{1}{|\Omega_{p,q}|}$ to each shuffle in $\Omega_{p,q}$.

Riffle Distributions

What if we allow a *distribution* over the interleavings?

Let $m_{p,q}$ be a distribution over $\Omega_{p,q}$. Equivalently, $m_{p,q}$ is a distribution over S_n which assigns probability zero to elements not in $\Omega_{p,q}$.

The **uniform riffle distribution**, denoted $m_{p,q}^{unif}$ is the one which assigns probability $\frac{1}{|\Omega_{p,q}|}$ to each shuffle in $\Omega_{p,q}$.

The uniform riffle distribution properly expresses indifference between X and $\overline{X}$ as classes, while still allowing for preferences within each class.

Riffle Independence

Define two f and g distributions as before to be **riffle independent** if h factors as $h(\sigma) = m_{p,q} * (f(\sigma_p) \cdot g(\sigma_q))$.

Riffle Independence

Define two f and g distributions as before to be **riffle independent** if h factors as $h(\sigma) = m_{p,q} * (f(\sigma_p) \cdot g(\sigma_q))$.

Let's note that the delta distribution on any one shuffle (i.e. fixing a shuffle) gets us back to the probabilistic independence case. We can see that riffle independence is a generalization of probabilistic independence.

Riffle Independence

Define two f and g distributions as before to be **riffle independent** if h factors as $h(\sigma) = m_{p,q} * (f(\sigma_p) \cdot g(\sigma_q))$.

Let's note that the delta distribution on any one shuffle (i.e. fixing a shuffle) gets us back to the probabilistic independence case. We can see that riffle independence is a generalization of probabilistic independence.

We need an additional $\binom{n}{p}$ parameters to characterize $m_{p,q}$, so storing a riffle independent distribution requires $\binom{n}{p} + p! + q!$ parameters.

Biased Riffle Shuffles

The authors describe a family of riffle distributions called **biased riffle distributions**.

Biased Riffle Shuffles

The authors describe a family of riffle distributions called **biased riffle distributions**.
These allow for expressing some weighted preference for one group over the other.

Biased Riffle Shuffles

The authors describe a family of riffle distributions called **biased riffle distributions**.
These allow for expressing some weighted preference for one group over the other.

For a bias parameter α, we can recursively construct a draw from $m^{\alpha}_{p,q}$ by appending to our final permutation the top item remaining in σ_p with probability proportional to α and the fraction of items remaining in σ_p.

Biased Riffle Shuffles

The authors describe a family of riffle distributions called **biased riffle distributions**.
These allow for expressing some weighted preference for one group over the other.

For a bias parameter α, we can recursively construct a draw from $m^{\alpha}_{p,q}$ by appending to our final permutation the top item remaining in σ_p with probability proportional to α and the fraction of items remaining in σ_p.

Note that $\alpha = .5$ gives us $m^{unif}_{p,q}$, $\alpha = 0$ says we prefer everything in $\overline{X}$ over anything in X, and $\alpha = 1$ is the opposite.

Conditional Independence

We might be inclined to wonder how riffle independence differs from conditional independence.

Conditional Independence

We might be inclined to wonder how riffle independence differs from conditional independence.

If instead we first draw a permutation of the ranks for the p items, then draw a σ_p and σ_q, we get something that feels like f and g being conditionally independent given the chosen ranks.

Conditional Independence

We might be inclined to wonder how riffle independence differs from conditional independence.

If instead we first draw a permutation of the ranks for the p items, then draw a σ_p and σ_q, we get something that feels like f and g being conditionally independent given the chosen ranks.

But conditional independence requires $\binom{n}{p}(p! + q! + 1)$ parameters, which is a further generalization of riffle independence.

Conditional Independence

We might be inclined to wonder how riffle independence differs from conditional independence.

If instead we first draw a permutation of the ranks for the p items, then draw a σ_p and σ_q, we get something that feels like f and g being conditionally independent given the chosen ranks.

But conditional independence requires $\binom{n}{p}(p! + q! + 1)$ parameters, which is a further generalization of riffle independence.

We can therefore say riffle independence lives somewhere between probabilistic and conditional independence.

Presentation Outline

Fourier Transform on Distribution

Let's observe that the Fourier transform of a distribution at the irreducible representations contains exactly enough information to fully reconstruct h.

Fourier Transform on Distribution

Let's observe that the Fourier transform of a distribution at the irreducible representations contains exactly enough information to fully reconstruct h.

It can be shown that, given an induced ordering on the representations, the Fourier matrices *systematically* encode the distribution.

Fourier Transform on Distribution

Let's observe that the Fourier transform of a distribution at the irreducible representations contains exactly enough information to fully reconstruct h.

It can be shown that, given an induced ordering on the representations, the Fourier matrices *systematically* encode the distribution.

Define a k**th order marginal** as the probability of observing a particular k-tuple at a certain position in a permutation.

Fourier Transform on Distribution

Let's observe that the Fourier transform of a distribution at the irreducible representations contains exactly enough information to fully reconstruct h.

It can be shown that, given an induced ordering on the representations, the Fourier matrices *systematically* encode the distribution.

Define a **kth order marginal** as the probability of observing a particular k-tuple at a certain position in a permutation.

The first two representations contain enough information to recover the first order marginals, the next three the second order marginals, and so on.

RiffleJoin and RiffleSplit

The authors present two algorithms which operate in the Fourier domain.

RiffleJoin and RiffleSplit

The authors present two algorithms which operate in the Fourier domain.

RiffleJoin takes the Fourier matrices for f, g, and m and computes the Fourier matrices for h.
RiffleSplit takes the Fourier matrix for h, factors out the uniform shuffle's dual's Fourier matrix, then splits f and g.

RiffleJoin and RiffleSplit

The authors present two algorithms which operate in the Fourier domain.

RiffleJoin takes the Fourier matrices for f, g, and m and computes the Fourier matrices for h.
RiffleSplit takes the Fourier matrix for h, factors out the uniform shuffle's dual's Fourier matrix, then splits f and g.

It can be shown that if f and g are not perfectly riffle independent, RiffleSplit returns estimates for factors which are as independent as possible.

Some experiments

The authors tested their algorithm on two data sets: the American Psychological Association election and the Sushi Data Set.

Experiment 1: APA Election

The data: 5738 ballots of ranked preferences across 5 candidates for president of the APA

Experiment 1: APA Election

The data: 5738 ballots of ranked preferences across 5 candidates for president of the APA

The hypothesis: C1,C3 and C4,C5 fall on opposite ends of a political spectrum. Voters may express a preference for one group over the other, then express preferences within the group.

Experiment 1: APA Election

The data: 5738 ballots of ranked preferences across 5 candidates for president of the APA

The hypothesis: C1,C3 and C4,C5 fall on opposite ends of a political spectrum. Voters may express a preference for one group over the other, then express preferences within the group.

The experiment: ignoring C2, decompose the full distribution into riffle factors.

Experiment 1: APA Election

The data: 5738 ballots of ranked preferences across 5 candidates for president of the APA
The hypothesis: C1,C3 and C4,C5 fall on opposite ends of a political spectrum. Voters may express a preference for one group over the other, then express preferences within the group.
The experiment: ignoring C2, decompose the full distribution into riffle factors.

The results: The authors find that C1,C3 and C4,C5 are nearly riffle independent, and that fitting a mixture-of-riffles model yields bias parameters that strongly suggest their hypothesis is correct.

Experiement 2: Sushi

The data: 5000 full rankings of 10 different kinds of sushi (note, WAY fewer than 10! observations). The authors divide this into two groups of five.

Experiement 2: Sushi

The data: 5000 full rankings of 10 different kinds of sushi (note, WAY fewer than 10! observations). The authors divide this into two groups of five.

The hypothesis: the full distribution can be approximated by combining an optimal m with two factor subsets and/or by finding an optimal bias parameter α.

Experiement 2: Sushi

The data: 5000 full rankings of 10 different kinds of sushi (note, WAY fewer than 10! observations). The authors divide this into two groups of five.
The hypothesis: the full distribution can be approximated by combining an optimal *m* with two factor subsets and/or by finding an optimal bias parameter α.
The experiment: Try to learn the true distribution using these two methods

Experiement 2: Sushi

The data: 5000 full rankings of 10 different kinds of sushi (note, WAY fewer than 10! observations). The authors divide this into two groups of five.
The hypothesis: the full distribution can be approximated by combining an optimal m with two factor subsets and/or by finding an optimal bias parameter α.
The experiment: Try to learn the true distribution using these two methods

The results: Assuming riffle independence significantly lowers the sample complexity required to learn the distribution.

Experiement 2: Sushi

The data: 5000 full rankings of 10 different kinds of sushi (note, WAY fewer than 10! observations). The authors divide this into two groups of five.
The hypothesis: the full distribution can be approximated by combining an optimal m with two factor subsets and/or by finding an optimal bias parameter α.
The experiment: Try to learn the true distribution using these two methods

The results: Assuming riffle independence significantly lowers the sample complexity required to learn the distribution.
Biased riffle shuffles are a useful tool for learning on small samples.

Presentation Outline

Discussion

What are some problems where riffle independence might lend some new insight?

What if the riffle factors are non-obvious?

How about if there are 3+ subsets we want to study?

Anything else?