Summer Research

Random Sampling from Defect-d Preference Lists

Jasper Bown

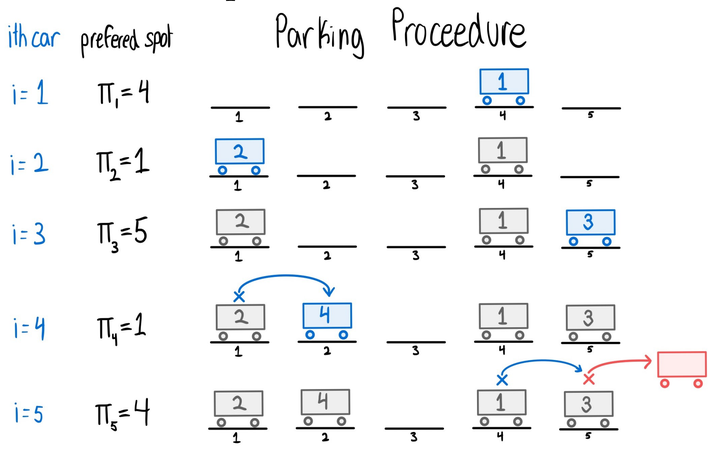
Introduction

Understanding how to sample uniformly often provides a useful starting point for understanding a space, enabling computation and highlighting structure. This poster describes the process of randomly sampling from defect-d preference lists.

Definitions

A **preference list**, $\pi \in [n]^m$, represents the parking preferences for m cars on a one way street with n spots.

For the preference list $\pi = 41514$



The **defect** of a preference list is the number of cars unable to park.

A **parking function**, $\pi \in PF_{n,m}$, is a preference list for n spots and m cars where all cars are able to park.

A **breakpoint** is an occupied spot where no other car attempts to park. For the above example, 2 is a break point.

A **prime parking function**, $\pi \in PPF_n$, is a parking function for n spots and cars with exactly 1 break point.

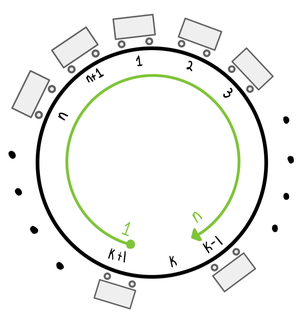
A **shuffle** interweaves two preference lists into a single preference list.
In the example above, $\pi = 41514$ is a shuffle of $\pi_1 = 11$ and $\pi_2 = 454$

Existing Methods

Parking Functions

Key mathematical insight:
Every coset of the subgroup the diagonal subgroup, $D = \langle (1, \cdots, 1) \rangle$, of C_{n+1}^m contains exactly 1 parking function.
To sample:

- Pick $\pi \in C_{n+1}^m$
- Pick $i \in [n-m+1]$
- Let k be the ith empty spot from circular parking
- Let $\pi_i' = \pi_i - k$

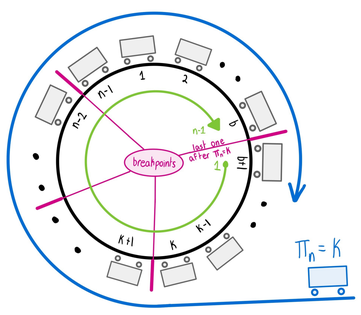


Enumeration:
$|PF_{n,m}| = (n-m+1)(n+1)^{m-1}$

Prime Parking Functions

Key mathematical insight:
Every coset of the diagonal subgroup of C_{n-1}^n contains exactly 1 prime parking function.
First seen in [2].
To sample:

- Pick $\pi \in C_{n-1}^n$
- Let b be the last breakpoint after π_n
- Let $\pi_i' = \pi_i - b$



Enumeration:
$|PPF_n| = (n-1)^{n-1}$

Sampling from Defect d

To sample from defect d preference lists, focus on s, the number of spots occupied after the last breakpoint. To sample:

- Sample s from the appropriate distribution given n and d
- Generate a parking function from $PF_{n-s,m-s-d}$
- Generate a prime parking function from PPF_{s+d} with no preferences for the last d spots. Let us call this a **d-prime parking function**, denoted $PPF_{n,d}$
- Shuffle the two preference lists

Distribution for s

Let $s(\pi)$ be the number of occupied spots after the last point and $d(\pi)$ the defect of π. Then the following generating function is computed by counting possible shuffles:

$$F(x,y) = \sum_{\pi \in [n]^m} x^{s(\pi)} y^{d(\pi)}$$

$$F(x,y) = \sum_{d=0}^{m-1} \sum_{s=1}^{m-d} \binom{m}{s+d} |PF_{n-s,m-s-d}||PPF_{s+d,d}| x^s y^d + (n-m) n^{m-1} x^0 d^0$$

Note that this generating function provides all of the necessary information to sample from the distribution for s given n and p.
Additionally, it provides a different method to enumerate defect d parking functions than [1].

p-prime Parking Function

Method 1

- Sample from $[s]^n$ where $s = n - p$ until you get a prime parking function

Removing the preference lists which have an empty first spot and those with their first break point at i gives the following count:

$$|PPF_{n,p}| = s^n - (s-1)^n - \sum_{i=1}^{s-1} \binom{n}{i} (i-1)^{i-1} (s-i)^{n-i}$$

Method 2

- Repeatedly sample from prime parking functions until you get a p-prime parking function

Following an inclusion exclusion argument based to remove prime parking functions which are not p-prime gives the following:

$$|PPF_{n,p}| = \sum_{i=0}^{p} \binom{n}{i} (n-i-1)^{n-i-1} (p-i)^i (-1)^i$$

Note that Abel's identity gives a computational way to show the two combinatorial expressions are the same.

When to use each method: When $p < n - (n-1)^{\frac{n-1}{n}}$, method 2 has a higher success rate. Otherwise the reverse is true.

Conclusions

- Break points and prime segments are basic building blocks of preference lists
- If you understand an attribute of interest for parking functions and p-prime parking functions, the distribution for s provides a method of extending this result to defect d preference lists.
- Defect 1 parking spaces can be understood with only prime parking functions and parking functions

References

[1] Peter J. Cameron, Daniel Johannsen, Thomas Prellberg, and Pascal Schweitzer. Counting defective parking functions, 2008.

[2] Richard P. Stanley and Sergey Fomin. *Enumerative Combinatorics*, volume 2 of *Cambridge Studies in Advanced Mathematics*. Cambridge University Press, 1999.

Acknowledgments

Thank you to Prof Orrison for numerous helpful and fun conversations in addition to the incredible support along the way. Thanks to the Harvey Mudd Math department for funding this research.

Advisor: Michael Orrison

Summer Research

Random Sampling from Defect-d Preference Lists

Jasper Bown

Draft of statement to submit to HMC poster session form (for a general audience)

Understanding how to sample uniformly at random from a space often provides a useful starting point for understanding that space. The process of sampling can provide insight into the structure of a space in addition to a computational starting point for checking future hypotheses. The goal of this poster is to describe the process of randomly sampling from defect-d preference lists. A defect-d preference lists is a generalization of parking functions, which originated in the study of hash tables and have connections ranging across different mathematical disciplines.

Questions about choices I made to leave some things off of the poster:

I have a couple of notes on how my definitions match up with the literature (or slight modifications), but took them out to save space on the poster. Is that OK or does it make it seem like I don't know how the terms are normally used in the literature?

- Parking function. Normally a parking function includes the constraint that the number of spots is the same as the number of cars. The definition I have listed generalizes the concept
- Breakpoint. The original definition of a breakpoint is a value i such that the increasing rearrangement $\pi'_i = i$. These points are the beginning of the prime segments in a parking function π. This simple formulation breaks down if there are any empty spots, so I started thinking of breakpoints as the definition above, which marks the end of a prime segment. I could change up my language to say endpoint maybe? This would not change the definition of a prime parking funciton
- Shuffle. In Diaconis and Hicks, a $\pi = 41514 \in sh(11, 121)$ rather than the example that I gave since the idea of shuffling

How broadly is $[n]$ used to mean $\{1, \cdots, n\}$? I got very used to that notation, and am wondering if I need to define it.

Are there other pictures which would be helpful/meaningful enough to include? Some ideas that I could include are the following

- A plot of what the distribution for s looks like
- a picture showing a defect d parking function broken into its two parts. That could be helpful, but I can also just point to the picture near the definitions
- A plot showing the probability of successfully getting a p-prime parking on the first try for Method 1 and Method 2 to illustrate the crossing point

One other thing that I didn't write explicitly on the poster: To me it is clear how a generating function gives you all of the counts that you need to be able to sample from the distribution for s. I could write out the probability as an expression in terms of the generating function, but that seems less important to me.

Should I make the pictures larger? If so, can I make the text smaller or do I need to edit?

There's definately more sources that I kind of refered to/based my work off of... Diaconis and Hicks for the importance of sampling/future direction, Meyles et al for the first time that I saw prime parking functions, Foata and Riordan for the paper that contains Pollak's circular argument. For references for a poster, how are you supposed to have space to include the mentions of relations to other work and all the space the references take up?

Advisor: Michael Orrison