Sequences of Sets

Austin R. Benson
Cornell University
Ithaca, NY
arb@cs.cornell.edu

Ravi Kumar
Google
Mountain View, CA
ravi.k53@gmail.com

Andrew Tomkins
Google
Mountain View, CA
atomkins@gmail.com

ABSTRACT

Sequential behavior such as sending emails, gathering in groups, tagging posts, or authoring academic papers may be characterized by a set of recipients, attendees, tags, or coauthors respectively. Such "sequences of sets" show complex repetition behavior, sometimes repeating prior sets wholesale, and sometimes creating new sets from partial copies or partial merges of earlier sets.

In this paper, we provide a stochastic model to capture these patterns. The model has two classes of parameters. First, a correlation parameter determines how much of an earlier set will contribute to a future set. Second, a vector of recency parameters captures the fact that a set in a sequence is more similar to recent sets than more distant ones. Comparing against a strong baseline, we find that modeling both correlation and recency structures are required for high accuracy. We also find that both parameter classes vary widely across domains, so must be optimized on a per-dataset basis. We present the model in detail, provide a theoretical examination of its asymptotic behavior, and perform a set of detailed experiments on its predictive performance.

ACM Reference Format:
Austin R. Benson, Ravi Kumar, and Andrew Tomkins. 2018. Sequences of Sets. In *KDD '18: The 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, August 19–23, 2018, London, United Kingdom.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3219819.3220100

1 INTRODUCTION

A significant fraction of the research in data mining and machine learning targets models of human behavior in pursuit of advantage in predicting which ad a user is likely to click on, which search result a user is interested in, which movie a user will enjoy, and so forth. Sometimes the event represents the first time a user has consumed a particular item, but sometimes it is the second or third time. The first time a user interacts with an item, numerous features about the item, the user, and their relationship have been studied to predict the identity of the item; this has largely been the focus of recommender systems [3]. To predict a repeated behavior, however, a new and powerful set of features emerges based on the nature and timing of past interactions of the item in question. Modeling of repeat behavior has a long history spanning psychology [12], marketing [15, 23], economics [10], and computer science [1, 28].

KDD'18, August 2018, London, UK

ACM ISBN 978-1-4503-5552-0/18/08…$15.00
https://doi.org/10.1145/3219819.3220100

Due to the powerful features available from past interactions, predictions of repeats are typically far more accurate than predictions of initial interaction.

Past study has focused almost exclusively on repeated behavior or interaction with a single item. In many settings, this is natural: the user will listen to a single song next, and the task is to predict this song [6]. However, in other common settings, the user will interact with several items simultaneously. For instance, online shopping carts may contain more than one item, emails may be sent to more than one person, people often meet in groups rather than pairs, and academic papers are typically produced through multi-way collaboration. All of these examples have strong repeat interaction effects, with subsets recurring either exactly or approximately. As an immediate example, the exact authors of this paper have written three earlier papers together, while a subset of two of the authors have written 68 earlier papers together.

With this idea of exact and approximate recurrence in mind, we now state our technical problem: given a sequence of past interactions, each of which is a set of items, predict the next set to be consumed. Our goal in studying this question is to capture natural behaviors effectively. Natural baseline approaches to this problem follow the techniques of single-item repeat consumption [4], taking into account the popularity and recency of individual items. However, we show that co-occurrence patterns within sets are not well-modeled by populating a new set via independent choices. We will present a model that out-performs such baselines by incorporating higher-order co-occurrence patterns.

We study the performance of our model based on eight datasets containing sets of paper authors, sets of email recipients, sets of tags applied to questions on stack exchange web sites, and sets of real-world co-occurrences of individuals. In Section 2.3 we establish that, as suggested by literature around group formation [13] and single-item repeat consumption [4, 6], repeated behavior is extremely common. The fraction of repeats varies, but in nearly all of our datasets, most set interactions are partial or exact repeats of already-seen sets. To see this, let us say that a set is an *extension* if it contains an already-seen set. In about half our datasets, half the sets are extensions. In the other half, almost every set is an extension. Hence, an understanding of re-use is critical to understanding behavior in these domains.

To complete the motivation for our problem, we consider whether the elements of each set may be viewed as occurring independently, in which case prior approaches to item-level repetition may suffice. We find strong evidence across all eight datasets that the constitution of elements within a set is not well-modeled by independent choices; hence, some set-aware process is required.

Before describing our model, we first define the boundaries of our repeat behavior modeling problem. We assume that the overall model is factored into a *size model* that determines how large a set

to produce at each timestep and a *membership model* that produces a set of the required size. We focus here exclusively on the membership model, and assume that the set sizes are given to us correctly. Additionally, following standard practice in repeat consumption and to focus the scope of the paper, we only model elements that are repeated. If a set contains four elements that have been seen before plus a novel fifth element, our process is responsible only for producing the four repeated elements. Our prior work provides some direction for how to jointly model novel and repeat consumption [6], although this was for sequences of single items (i.e., not of sets). We leave joint modeling of novel and repeat elements in set sequences for future work.

We now provide an overview of our most accurate model. At step $k+1$, the model must produce a set S_{k+1} of known size based on the k sets $S_1, \ldots, S_k$ seen so far. First, the set S_{k+1} is initialized to be empty. Next, the model selects a prototype set S_j from the past and randomly adds some elements from S_j to S_{k+1}. This step is repeated until S_{k+1} is of the appropriate size. We call our model the *Correlated Repeated Unions* (CRU) model, as it works by repeatedly taking the union of correlated subsets of prior sets.

The prototype from i timesteps in the past will be selected with probability proportional to some learned weight w_i, optimized to account for the particularities of recency in the dataset being trained. As we would expect, the optimized weights are roughly monotonically decreasing, but at different rates for different datasets.

The number of elements to copy from S_j to S_{k+1} is controlled by a correlation parameter p, which may be learned together with the w's (although in our experiments, we learn the w's by gradient descent and optimize p by grid search). Each element of S_j is copied to S_{k+1} with independent probability p, so on average a p fraction of the elements are copied from each prototype until the target set size is attained The complexity in fitting the model lies in computing the likelihood that a certain element from the past contributed to the formation of a new set; we perform this likelihood calculation via a trick that requires materializing all partitions of the new set. Details on the model and learning procedure are in Section 3.

We compare our model to a baseline where we flatten each set into a sequence of items, and then apply a standard single-item repeat consumption model. We show that our model significantly outperforms this baseline, providing a per-set mean likelihood improvement between 28% and 100% for an appropriate choice of p. We also show that correct modeling of the correlation likelihood for each dataset is essential for best performance. Some datasets, such as email recipients, perform best as $p \to 1$, whereas others show a significant likelihood drop as $p \to 1$. Most datasets show a clear mode, for which one regime of p provides clear best performance.

We also study the theoretical behavior of our process. If novel elements continue to arrive into the process, of course the behavior will continue to feature such elements. However, if eventually the new elements stop arriving, it is reasonable to ask whether the resulting fixed set of elements will all continue to occur forever, or whether a diminishing set of increasingly popular items will begin to dominate. In fact, we show that the outcome depends on the nature of the recency weights. If the infinite sum of the weights converges then with probability 1, the process will eventually repeat a single set forever. On the other hand, if the sum of the weights diverges, then every possible subset will occur infinitely often.

2 DATA ANALYSIS

The datasets we consider here are sequences of sets, where each sequence is a time-ordered list of subsets of elements from some universal set $\mathcal{U}$. We ignore the absolute value of the times and only consider the *ordering* of the sets in the sequence by time. Thus, by a "sequence of sets", we mean a list of sets $S_1, \ldots, S_n$, where $S_i \subseteq \mathcal{U}$, and a dataset consists of several such sequences of possibly varying lengths. In order to study sequences of sets, we collected datasets from a variety of domains. We briefly describe the datasets below. All of our data has been made publicly available.[1]

Email. In the email datasets, each sequence is derived from the recipients of emails sent by a particular email address. In the email-Enron-core dataset, a sequence of sets is the time-ordered sequence of sets of recipients of an email from a given sender email address in the Enron corpus [17]. We restrict the dataset to the "core" group of employees whose email was made public by the FERC investigation of the company—each sequence corresponds to one employee's emails. The email-Eu-core dataset is derived from the temporal network of email between employees at a European research institution [18, 30]. Timestamps were recorded at a resolution of one second, and we consider the set of all receivers of an email from a given sender at a given timestamp to be a set. Again, each sequences corresponds to one employee's emails.

Stack exchange tags. Stack exchange is a collection of question-and-answer web sites. Users post questions and annotate them with up to five tags. In our stack exchange tag datasets, each sequence is the time-ordered set of tags applied to questions asked by a user. The dataset tags-mathoverflow uses the complete history of MathOverflow,[2] a stack exchange site for research-level mathematics questions, and the dataset tags-math-sx uses the complete history of Mathematics Stack Exchange,[3] a stack exchange for general mathematics questions at any level.

Proximity-based contacts. The datasets contact-high-school and contact-prim-school are constructed from interactions recorded by wearable sensors in a high school [19] and a primary school [27]. The sensors record proximity-based contacts every 20 seconds. There is one sequence of sets per person, and we consider the set of individuals that a person comes into contact within each 20 second interval to be a set (only nonempty sets are considered—some intervals contain no interactions).

Coauthorship. Over time, researchers publish papers, often with other coauthors. In these datasets, each sequence corresponds to a researcher, and each set in the sequence is comprised of the coauthors on the paper (thus, a paper with k authors appears as part of k sequences—one for each author). The sequence is ordered by time of publication. Single-author papers are ignored, since these would correspond to an empty set in the sequence. We derive two datasets from the Microsoft Academic Graph—coauth-Geology and coauth-Business—corresponding to papers categorized as "Geology" or "Business" [5, 26].

We filter each dataset to only keep sequences of length at least 10 and sets of size at most five. The restriction to sets of size five is to provide uniformity across datasets. Stack exchange only allows

[1] http://www.cs.cornell.edu/~arb/data/
[2] https://mathoverflow.net
[3] https://math.stackexchange.com

Table 1: Summary statistics of datasets. Each dataset consists of sequences of sets, and each sequence is a time-ordered list of subsets from a universe of elements $\mathcal{U}$. The number of sets is the sum of the sequence lengths.

Dataset	# seqs.	$\lvert\mathcal{U}\rvert$	# sets	# unique sets
email-Enron-core	93	141	10,428	649
email-Eu-core	681	937	202,769	9,694
contact-prim-school	242	242	174,796	18,412
contact-high-school	325	327	308,990	9,785
tags-mathoverflow	1,594	1,399	44,950	24,157
tags-math-sx	15,726	1,650	517,810	122,099
coauth-Business	24,019	236,226	463,070	271,294
coauth-Geology	57,294	525,348	1,438,652	1,090,485

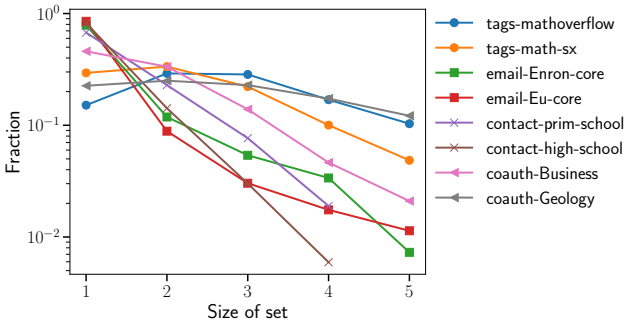

Figure 1: Distribution of set sizes.

up to five tags, so no data is lost there. Sets of size five capture most of the email, contact, and coauthorship datasets. Table 1 provides summary statistics of the datasets, and Figure 1 shows the distribution of set sizes in the sequences of our datasets. In the coauthorship and tag datasets, more than 50% of the sets have at least two elements.

Our subsequent data analysis gets across three points that will be used by our model. First, repeat behavior is common. Many sets appearing at some point in a sequence have appeared previously in the sequence. However, the repeats are not always the same set—we often see supersets or subsets of prior sets. Second, the elements appearing in sets are correlated. Specifically, pairs and triples appear more frequently than one would expect by chance given their total number of appearances in a sequence of sets. Third, there is a recency bias in set selection—a set is, on average, more similar to recent sets than older ones.

2.1 Repeat behavior

We first show that repeat behavior is common in our datasets. For each dataset, we measure the fraction over all sets in all sequences that are (i) exact repeats, i.e., the same set appeared earlier (Figure 2A); (ii) entirely novel, i.e., none of the set elements appeared earlier in the sequence (Figure 2B); (iii) subsets of a prior set (Figure 2C); and (iv) supersets of a prior set (Figure 2D). In last two cases, we do not require proper subsets or supersets. We measured these statistics as a function of the set size.

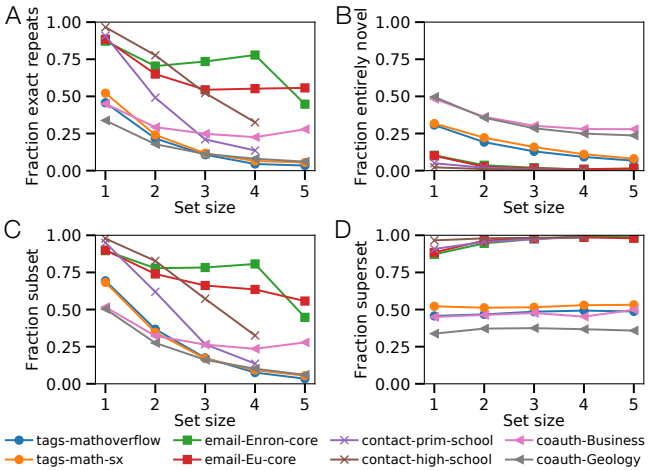

Figure 2: Repeat behavior in our datasets. (A) Fraction of sets in sequences that are exact repeats of a previous set as a function of set size. All datasets exhibit repeat behavior, although larger sets are typically less likely to be exact repeats. (B) Fraction of sets in sequences that are made up of completely novel items that have not appeared earlier in the sequence. Even when the set size is just 1 element, fewer than half the sets are comprised of entirely new elements. This fraction decreases with set size. (C) Fraction of sets in sequences that are (not necessarily proper) subsets of a previous set in the sequence. (D) Fraction of sets in sequences that are (not necessarily proper) supersets of a previous set in the sequence. In the email and contact networks, nearly all sets in a sequence of sets are supersets of a previous set.

We highlight a few key results. First, very few sets are comprised of entirely novel items (Figure 2B)—fewer than 50% for the coauthorship datasets, fewer than 35% for tags, and fewer than 10% for email and contact networks. These numbers decrease as the set size increases. Exact repeats (Figure 2A) and subsets (Figure 2C) exhibit similar behavior. We again see a large percentage of exact repeats or subsets of prior sets for small set sizes, although these percentages can decrease dramatically for large sets (especially for the tags datasets). Finally, many sets are supersets of prior sets. For email and contact networks, nearly all sets are supersets of earlier sets in the sequence, and in tags and coauthorship data, about half of the sets are supersets. Thus, our model should capture that new sets are in some sense constructed from elements appearing in (possibly several) prior sets in the sequence.

Finally, we examine the distribution of the number of repeated elements in sets that contain at least one element. Formally, for a sequence of sets $S_1, \ldots, S_n$, we measure $|(\cup_{j=1}^{r-1} S_j) \cap S_r|$, $r = 1, \ldots, n$, over all sequences in the dataset. Figure 3 shows the distribution of these values. We see that for several datasets, there is often more than one repeated element in the subset.

2.2 Subset correlation

Our second observation about our data is that subsets of consumed sets tend to be correlated. More specifically, the same subsets show up in multiple sets. This will play a key role in our model.

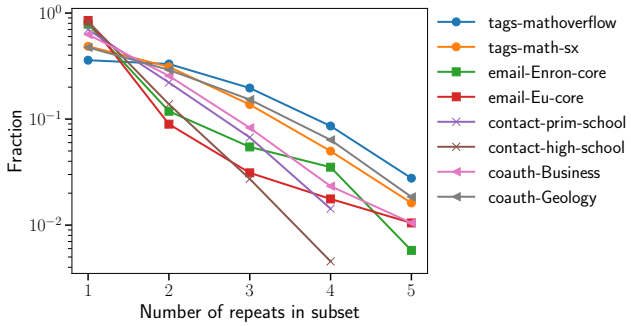

Figure 3: Distribution of the number of repeated elements in sets that contain at least one repeated item (c.f. Figure 1).

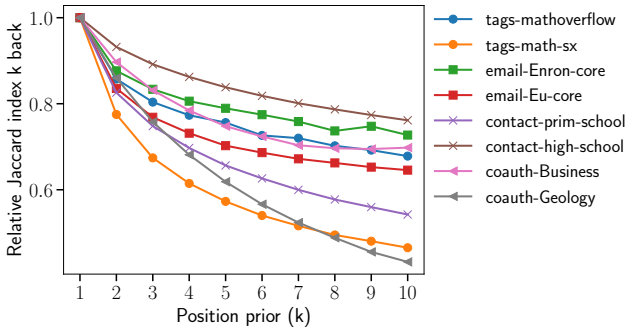

Figure 4: Evidence of recency bias in set selection. Average Jaccard index of a set in a sequence with the set appearing k steps prior in the sequence, normalized to $k = 1$. In all datasets, similarity is higher when k is small (all numbers are less than Jaccard index with the previous set, i.e., when $k = 1$). Similarity is roughly monotonically decreasing in k.

Table 2: Subset correlations. For each sequence in each dataset, we count the number of times each size-2 and size-3 subset appears. The mean counts are under the "data" columns. We then perform the same computation under a null model where elements are placed randomly into sets. We report the mean ± one standard deviation over 100 random samples. The real data has larger counts amongst size-2 and size-3 subsets compared to the null model.

	size-2 subset counts		size-3 subset counts	
Dataset	data	null model	data	null model
email-Enron-core	5.82	4.34 ± 0.043	4.23	2.67 ± 0.038
email-Eu-core	4.46	3.11 ± 0.008	3.23	2.08 ± 0.007
contact-prim-school	2.36	1.87 ± 0.003	1.35	1.09 ± 0.002
contact-high-school	4.49	3.26 ± 0.007	2.09	1.35 ± 0.004
tags-mathoverflow	1.49	1.41 ± 0.002	1.18	1.15 ± 0.002
tags-math-sx	1.49	1.31 ± 0.001	1.21	1.12 ± 0.001
coauth-Business	1.50	1.30 ± 0.001	1.40	1.24 ± 0.001
coauth-Geology	1.29	1.15 ± 0.000	1.15	1.07 ± 0.000

We quantify subset correlations as follows. For every size-2 and size-3 subset T of any set in a sequence, we count the number of times T appears in the sequence. This generates a count of each set, where larger counts indicate that T co-appears more often. Formally, in a sequence of sets $S_1, \ldots, S_n$, the count for a set T is

$$\text{count}(T) = \sum_{j=1}^{n} \text{Ind}[T \subseteq S_j], \tag{1}$$

where Ind[·] is the indicator function. We then aggregate such counts over all sequences in the dataset and compute the mean count. We note that the same subset T may appear in several sequences, and its counts would be considered separately for each sequence. The reasoning for this is that our goal is to capture sequence-level correlations rather than global correlations. Table 2 reports the mean counts over all size-2 and size-3 subsets.

If certain elements appear much more frequently than others, then the mean count can be large just from this structure. To provide evidence of subset correlations, we compare against a null model. The null model for a sequence keeps all of the set sizes the same but randomly assigns elements to sets. Thus, common items still appear frequently, but their correlations with other items are destroyed. We perform the same computations described above for 100 samples from the null model. Table 2 reports the mean and standard deviations of the mean counts over these instances of the null model. The size-2 and size-3 co-appearance counts are significantly larger than those in the null model. We conclude that our model should capture the fact that subsets of sets in a sequence tend to appear again as a subset of a set later in the sequence.

2.3 Recency bias

Our third observation about the data is that there is a recency bias in the sequences. More specifically, a given set is, on average, more similar to recent sets in the sequence. We measured the Jaccard index of sets in a sequence $S_1, \ldots, S_n$ by

$$J(S_r, S_{r-k}), \; r = 1, \ldots, n, \; k = 1, \ldots, \max(r-k, 0). \tag{2}$$

Here, k controls the recency, and $k = 1$ corresponds to the previous set in the sequence.

Figure 4 shows the average Jaccard index as a function of k, relative to the case of $k = 1$. The $k = 1$ case has the largest relative value in all datasets, meaning that similarity is largest with the most recent set. For all datasets, the similarities tend to decrease with k, providing further evidence that new sets are more related to the most recent sets in the sequence. This is consistent with prior work on repeat consumption on the Web [4, 6].

3 THE CORRELATED REPEATED UNIONS (CRU) MODEL FOR SEQUENCES OF SETS

We now propose our model for sequences of sets, incorporating the three ingredients observed in the previous section: repeat behavior, subset correlation, and recency bias. Our focus is specifically on modeling the repeat consumption, rather than the novel items that might appear in a sequence of sets, as we have identified this as a substantial feature of our sequences of sets data. Thus, our modeling framework takes the novel items and number of repeats as given and tries to reconstruct the repeats in a set from the history of the sequence up to that point. Modeling the novel items and sequence of set sizes is outside the scope of this paper, but certainly serves as an important avenue for future research. We anticipate that our model here will serve as the foundation for a more holistic modeling

Algorithm 1: Correlated Repeated Unions (CRU) model for repeat subset sampling.

Input: number of repeat elements r, recency weight vector w, correlation probability p, sequence of sets $S_1, \ldots, S_k$
Output: a repeat set $R \subseteq \cup_{j=1}^{k} S_j$ with $|T| = r$

```
R ← ∅
while true do
    if |R| = r then return R
    Sample set S_i with probability ∝ w_{k−i+1}
    Sample T ⊆ S_i by including each x ∈ S_i with probability p
    if |R ∪ T| > n then
        while |R| < n do
            Uniformly at random sample y ∈ T
            R ← R ∪ {y}
            T ← T \ {y}
    else
        R ← R ∪ T
```

framework. We call our model the *Correlated Repeated Unions* (CRU) model because it generates repeated elements of the next set in a sequence by taking the union of correlated subsets of sets in the history of the sequence.

In the next section, we formally describe the model. After, we show how to efficiently evaluate the likelihood of the data given the model parameters and learn the model parameters. Section 4 provides empirical evaluation of our model, showing that it outperforms a competitive baseline, while Section 5 is dedicated to theoretical analysis of the model.

3.1 Formal model description

Finally, we get to the model description. Recall that our data consists of sequences of sets. For simplicity of presentation, we only consider a single sequence of sets $S_1, \ldots, S_n$ for now.

Suppose that we have observed the sequence up to the kth set S_k. To reiterate our setup, we assume that an oracle has given us the following information about the next set S_{k+1}:

(1) the size of the new set: $|S_{k+1}|$
(2) the novel elements in the set: $N_{k+1} = S_{k+1} \setminus \cup_{i=1}^{k} S_i$.

Our goal is to determine the remainder of the set (i.e., $S_{k+1} \setminus N_{k+1}$), which are all repeated elements from the history of the sequence thus far $(S_1, \ldots, S_k)$.

The CRU model for constructing the repeated elements is really an algorithm that accumulates elements by sampling from the sequence thus far and taking unions (see Algorithm 1). There are two parameters of the algorithm: the recency weight vector w (of length $n-1$, where n is the length of the *entire* sequence) and the correlation probability p. The algorithm first initializes an empty set R and then samples a set S_i proportional to the recency weight w_{k-i+1}; for example, the most recent set S_k is sampled proportional to w_1. The algorithm then adds each element from S_i to R with probability p. Equivalently, a subset $T \subset S_i$ is sampled by including each element of S_i with probability p, and then R is updated by taking the union of itself with T. The algorithm then repeats until R has the correct number of elements (i.e., $|S_{k+1} \setminus N_{k+1}|$). If at some point the next sampled subset T would make R too large, then elements are uniformly at random dropped from T until R is the appropriate size and the algorithm terminates. The next set in the sequence is then $S_{k+1} = N_{k+1} \cup R$.

A key idea behind the CRU model is that it induces a probability distribution over repeat sets, making likelihood computation and parameter optimization tractable. We show this in the following two sections.

According to our findings of recency bias in Section 2.3, we should expect that earlier values (corresponding to smaller indices) in the optimal vector w should be larger than the later ones (corresponding to larger indices). This would imply that we are more likely to sample from more recent sets. Indeed, we will later find this to be the case across all datasets when we learn optimized model parameters from data (Figure 6).

We also expect the correlation probability p to have a role. The limit as $p \to 0$ means that only one item from a prior subset will be selected at a time. Our findings in Section 2.2 suggest that p should be somewhat larger than 0, in order to capture the correlation patterns of subsets. However, the optimal value of p is not obvious, and we will see that it is certainly greater than 0 but depends on the dataset (Figure 5). However, the optimal value of p tends to be roughly the same within each dataset domain.

3.2 Evaluating model likelihood

We now show how to evaluate the likelihood of a sequence of sets under our model. For simplicity of presentation, we consider the evaluation of the likelihood of one particular set in a sequence of sets. In the full model, the recency weights w and the correlation probability p are common across all sequences in a dataset. The log-likelihood of an entire dataset is then just the sum of the logs of the likelihoods on each sequence.

Again, let $S_1, \ldots, S_n$ be the observed sequence, and we will consider the likelihood of S_{k+1} under the CRU model, given the weight vector w and the correlation probability p. We introduce some additional notation. Let $\mathcal{P}(X)$ be the set of all *ordered partitions* of a set X, and let $E_{r,k}$ be the set of all size-r subsets of $\cup_{i=1}^{k} S_i$. For example, if $X = \{a, b\}$, then $\mathcal{P}(X) = \{(\{a\}, \{b\}), (\{b\}, \{a\}), (\{a, b\})\}$; and if $S_1 = \{a, b, c\}$, $S_2 = \{a\}$, and $S_3 = \{b, d\}$, then $E_{2,1} = E_{2,2} = \{\{a, b\}, \{a, c\}, \{b, c\}\}$ and $E_{2,3} = \{\{a, b\}, \{a, c\}, \{b, c\}, \{a, d\}, \{b, d\}, \{c, d\}\}$.

A key component of the CRU model is that there is a canonical surjective function from the output of Algorithm 1 with input $(r, w, p, S_i, \ldots, S_k)$ to the space $\Omega = \cup_{E \in E_{r,k}} \mathcal{P}(E)$. The output of Algorithm 1 can be interpreted as a set $E \in E_{r,k}$ as the incremental construction of R is equivalent to an ordered partition of the elements of R. Specifically, any execution of the outer **while** loop that changes R serves as the next subset in the ordered partition (i.e., when $T \setminus R \neq \emptyset$, there is a new subset that is added to the ordered partition). Since Algorithm 1 is random, it induces a probability distribution over Ω.

We illustrate the above process with an example. Suppose that $S_1 = \{a, b\}$, $S_2 = \{b, c\}$ and we are using the model to predict a repeat set R with $|R| = 2$. Let $w_i' = w_i/(w_1 + w_2)$ be normalized

recency weights for $i = 1, 2$. There are six possible samples T in each execution of the while loop: $T = \{a, b\}$ with probability $p^2 w'_2$; $T = \{b, c\}$ with probability $p^2 w'_1$; $T = \{a\}$ with probability pw'_2; $T = \{b\}$ with probability $pw'_2 + pw'_1$; $T = \{c\}$ with probability pw'_1; and $T = \varnothing$ with probability $1-p^2$. If $T \setminus R = \varnothing$, then the outer **while** loop of Algorithm 1 simply executes again with another sample of T. Otherwise R is updated, and we get the next set in the ordered partition. There are multiple ways in which, e.g., $R = \{b, c\}$ could be returned from Algorithm 1: the size-2 set $\{b, c\}$ is sampled from S_2; $\{b\}$ is sampled first from S_1 and then $\{c\}$ is sampled from S_2 (or in reverse order); or $\{b\}$ is sampled first from S_2 as a single item and then $\{c\}$ is sampled from S_2 (or in reverse order). Each case corresponds to an ordered partition of $\{b, c\}$.

Now we assume that we have observed the repeat elements R and want to evaluate the likelihood of the data given model parameters. Let $\mathcal{L}$ denote the likelihood and let $R_{k+1} \subseteq S_{k+1}$ be the set of repeat elements in S_{k+1}. Also let $\mathcal{A}(r, w, p, S_1, \ldots S_k)$ be a random variable over Ω denoting the probability of Algorithm 1 using a particular ordered partition. Then we have that

$$\begin{aligned}&\mathcal{L}(R_{k+1} \mid S_1, \ldots, S_k, w, p)\\ &\quad = \textstyle\sum_{X \in \mathcal{P}(R_{k+1})} \Pr(\mathcal{A}(|R_{k+1}|, w, p, S_1, \ldots S_k) = X). \end{aligned} \tag{3}$$

In other words, the likelihood of observing R_{k+1} is just the probability that the algorithm constructs R_{k+1} from some ordered partition $X \in \mathcal{P}(R_{k+1})$. Crucially, the CRU model is fashioned in a way that permits us to efficiently compute these probabilities.

Now we fix X and show how to evaluate the probability in Eq. (3). We will work through this computation algorithmically, following Algorithm 1. Suppose that we have accumulated X "correctly" thus far and that we are going to add the next subset B in the ordered partition X. Further suppose that B is not the last subset in the ordered partition X. Let T be the sample in a loop of the algorithm and let R be the accumulation of elements thus far in the execution of Algorithm 1. For the algorithm to succeed in producing X, one of two things must occur next

(1) $T \subseteq R$, in which case the **while** loop starts over
(2) $B \subseteq T \subseteq R \cup B$

Eventually, we need the second event to happen. Let q_r be the "restart probability" of the first case and let q_s be the "success probability" of the second case from one loop of the algorithm. Then the probability that the algorithm continues to succeed is

$$\textstyle\sum_{k=0}^{\infty} q_r^k q_s = q_s \sum_{k=0}^{\infty} q_r^k = \frac{q_s}{1-q_r}. \tag{4}$$

We can compute both q_s and q_r. Let $w'_i = w_i / \sum_{j=1}^{k} w_j$ be the normalized recency weights and $p_{T,S}$ be the probability of sampling $T \subseteq S$ under the model that elements of S are taken i.i.d. with probability p. If $|T| = t$ and $|S| = s$, then $p_{T,S} = p^t(1-p)^{s-t}$. Then

$$q_s = \textstyle\sum_{i=1}^{k} w'_{k-i+1} \sum_{T \subseteq S_i} p_{T,S_i} \cdot \mathrm{Ind}[B \subseteq T \subseteq R \cup B] \tag{5}$$

$$q_r = \textstyle\sum_{i=1}^{k} w'_{k-i+1} \sum_{T \subseteq S_i} p_{T,S_i} \cdot \mathrm{Ind}[T \subseteq R] \tag{6}$$

Now suppose that the next set B in the ordered partition X is the last one added to the set. In this case, we need to account for the fact that the sampled set T could make R "too big", in which case we randomly select elements from T to fill up R (the second **if** statement in the outer **while** loop of Algorithm 1). Equations (4) and (6) stay the same, but the value of q_s in Eq. (5) changes.

In this case, success of our algorithm means that $|T \setminus (R \cup B)| \geq |R \cup B|$ *and* only elements $y \in R \cup B$ are sampled before any element $y' \in T \setminus (R \cup B)$. Let $C = T \setminus (R \cup B)$. We claim that given the sample T, the probability that the algorithm successfully captures B is

$$z_{R,B,T} := \tfrac{|B|! \cdot |C|!}{(|B|+|C|)!},$$

To prove this, observe that the sampling procedure in the second **if** statement of Algorithm 1 is equivalent to first taking a random ordering of the elements of T and adding them in order, one by one to R, until $|R| = r$. Sampling $y \in R \cap T$ has no effect, so we only care about the relative ordering of elements in the disjoint sets B and C. There are $(|B| + |C|)!$ possible orderings, all equally likely by symmetry. The number that successfully capturing B have the first $|B|$ elements fixed to be B, and there are $|B|! \cdot |C|!$ such cases.

We now adjust Eq. (5) with an extra multiplier, using this result:

$$\tilde{q}_s = \textstyle\sum_{i=1}^{k} w'_{k-i+1} \sum_{T \subseteq S_i} p_{T,S_i} \cdot \mathrm{Ind}[B \subseteq R] \cdot z_{R,B,T}. \tag{7}$$

Finally, we put everything together. Denote the ordered partition by $X = (B_1, \ldots, B_t)$ and the repeat and success probabilities by $q_r(B_i)$ for $i = 1, \ldots, t$; $q_s(B_i)$ for $i = 1, \ldots t-1$; and $\tilde{q}_s(B_t)$. Then the likelihood contribution from X for R_{k+1} is

$$\left(\textstyle\prod_{i=1}^{t-1} \frac{q_s(B_i)}{1-q_r(B_i)}\right) \frac{\tilde{q}_s(B_t)}{1-q_r(B_t)}. \tag{8}$$

The total likelihood of a given repeat set R_{k+1} is then the sum of the above equation over all ordered partitions $X \in \mathcal{P}(R_{k+1})$. The log-likelihood takes the log of this sum, and then adds together other log-sums for $R_1, \ldots, R_n$ in the entire sequence of sets for all sequences in the entire dataset.

3.3 Learning model parameters

The log-likelihood function is not convex, due to the product form in Eq. (8). We learn p by a simple grid search, as our goal here is just to capture some macroscopic properties of the correlations. We learn the recency weights w from projected gradient descent, using a linear time (up to logarithmic factors) projection onto the probability simplex [11]. The remainder of this section sketches out the computation of the gradient, which can be done in the same time and space it takes to compute the likelihood. In practice, we simultaneously compute the likelihood and the gradient.

Following Eq. (3), the log-likelihood with respect to the parameters w for a particular repeat set is

$$LL(w) = \log\left[\textstyle\sum_{X \in \mathcal{P}(R_{k+1})} \Pr(\mathcal{A}(|R_{k+1}|, w, p, S_1, \ldots S_k) = X)\right].$$

Thus, applying the chain rule,

$$\nabla_w LL = \frac{\sum_{X \in \mathcal{P}(R_{k+1})} \nabla_w \Pr(\mathcal{A}(|R_{k+1}|, w, p, S_1, \ldots S_k) = X)}{\sum_{X \in \mathcal{P}(R_{k+1})} \Pr(\mathcal{A}(|R_{k+1}|, w, p, S_1, \ldots S_k) = X)}.$$

We now focus on a particular ordered partition $X = (B_1, \ldots, B_t)$ and the gradient $\nabla_w \Pr(\mathcal{A}(|R_{k+1}|, w, p, S_1, \ldots S_k) = X)$.

Let $W = \sum_{i=1}^{k} w_i$ be the weight normalization. We can rewrite Eq. (8) as

$$\left(\textstyle\prod_{a=1}^{t-1} \frac{W \cdot q_s(B_a)}{W - W \cdot q_r(B_a)}\right) \frac{W \cdot \tilde{q}_s(B_t)}{W - W \cdot q_r(B_t)} = \left(\textstyle\prod_{a=1}^{t-1} \frac{f_a(w)}{g_a(w)}\right) \frac{\tilde{f}_t(w)}{g_t(w)}. \tag{9}$$

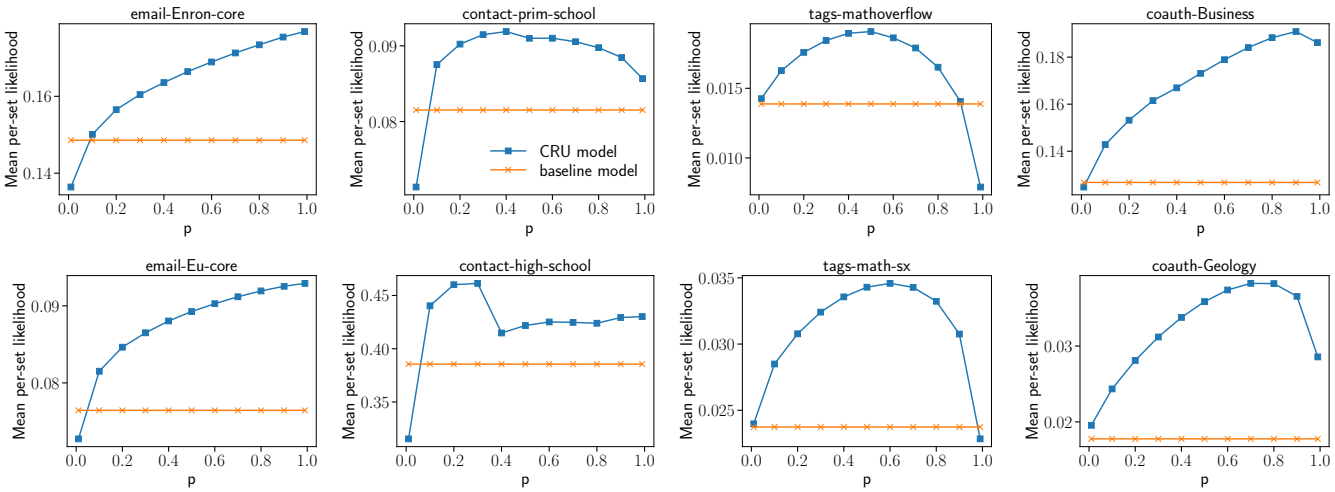

Figure 5: Mean per-repeat-set likelihood as a function of the correlation probability p. A larger p means more correlation in selecting items from the same set. We compare our CRU model against a "flat" baseline model, which has more model parameters but does not explicitly use set structure. Likelihood tends to be unimodal in p. In email, likelihood increases with p, suggesting that new sets are constructed by merging prior ones. Coauthorship has a maximum for large values of p but is not strictly increasing, suggesting that new sets are formed from sets close to—but not exactly the same as—prior sets.

We claim that f_a, g_a, and $\tilde{f}_t$ are linear in w. Following Eqs. (5) to (7):

$$f_a(w) = \sum_{i=1}^{k} w_{k-i+1} \sum_{T \subseteq S_i} p_{T,S_i} \cdot \text{Ind}[B \subseteq T \subseteq R \cup B_a];$$

$$g_a(w) = \sum_{i=1}^{k} w_{k-i+1} - \sum_{i=1}^{k} w_{k-i+1} \sum_{T \subseteq S_i} p_{T,S_i} \cdot \text{Ind}[T \subseteq R];$$

$$\tilde{f}_t(w) = \sum_{i=1}^{k} w_{k-i+1} \sum_{T \subseteq S_i} p_{T,S_i} \cdot \text{Ind}[B \subseteq R] \cdot z_{R,B_t,T}.$$

All of the weights on the linear functions in w are computed when computing the likelihood. Applying the product and quotient rules to Eq. (9) gives the final gradient.

4 EXPERIMENTAL RESULTS

We now analyze the CRU model after learning the recency weights w for each value of $p \in \{0.01, 0.1, 0.2, \ldots, 0.9, 0.99\}$ We compare against a baseline model (described below) and see that there are substantial likelihood gains for an appropriate correlation probability p. We then analyze the learned recency weights and confirm that they tend to decrease in the vector index, i.e., more weight is indeed placed on recent items. Under the assumption that recency weights monotonically decrease, we prove properties of the behavior of the model in Section 5.

4.1 Likelihood and performance

Figure 5 shows the mean per-set likelihood of the model on our datasets after having learned the recency weights for various values of the correlation probability p. Specifically, if LL_p is the log-likelihood with correlation probability p and optimized recency weights w, then we report $e^{LL_p/N}$, where N is the total number of sets in sequences of a dataset that contain at least one repeat.

The absolute value of the mean per-set likelihood may be small since there can be a large number of possible sets that contribute contribute to the likelihood. Thus, we compare against a baseline model that elucidate some of the likelihood gains that are possible by accounting for set structure. More specifically, we compare against a "flat model," which is similar to a prior model by Anderson et al. [4]; this model ignores the set structure and "flattens" the sequence of sets into a sequence of individual elements. We learn a set of recency weights (at the element level, instead of the set level), and draw elements proportional to learned recency weights. Essentially, this baseline ignores the set structure in the dataset; however, it also has more model parameters since there are a larger number of recency weights to learn.

We find that correlation probabilities p between 0 and 1 lead to substantial likelihood gains over the baseline. Furthermore, the likelihood gains tend to be unimodal in p with similar optima for datasets in the same domain. In the email datasets, likelihood simply increases with p, suggesting that many repeat sets are constructed from merging the entirety of prior subsets, or simply copying a single prior set in the sequence. This makes sense in context—there may be several emails sent by one person to the same set of people if, for instance, these individuals are working on a project together.

The contact and tags datasets have optimal correlation probabilities p at 0.3–0.4 (contact) and 0.5–0.6 (tags). Thus, new sets are formed via proper subsets of previous sets. With tags, this could be explained by the combined use of high-level concept tags and question-specific tags. An individual might explore the same general area of mathematics (e.g., algebra) and then ask questions on specific sub-areas (e.g., group theory). Finally, the coauthorship data has optimal likelihoods for large values of p (≥ 0.8), but not for $p = 1$. This suggests that coauthorship repeats are largely the same, but not exactly. This might be explained by individuals getting added or removed from a research collaboration over time.

4.2 Learned recency weights

Figure 6 shows the learned recency weights for all of the datasets and all of the correlation probabilities p. The weights tend to monotonically decrease, independent of p, which is consistent with our

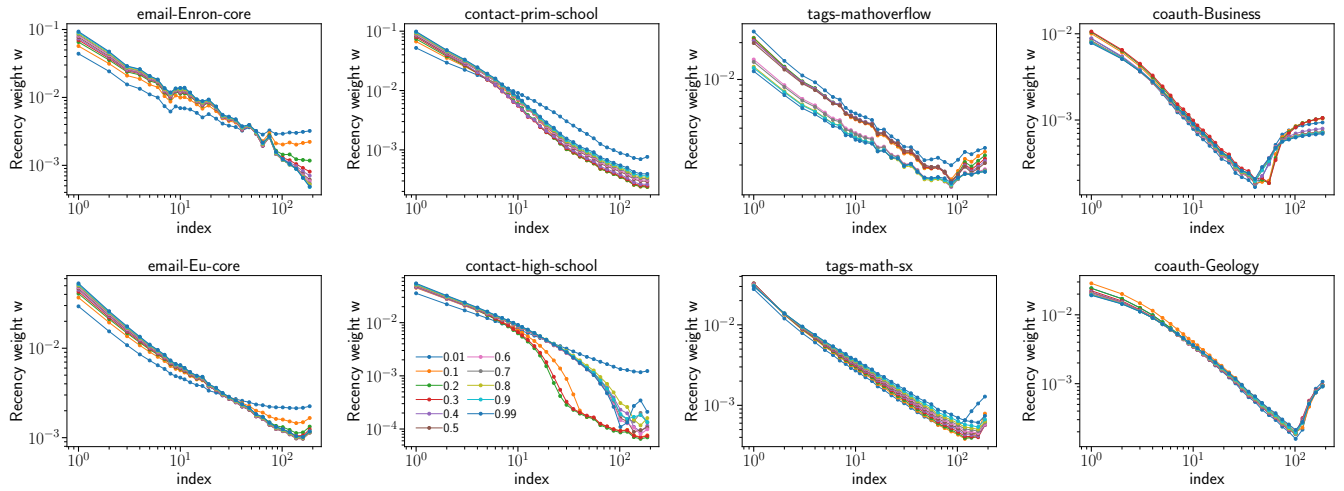

Figure 6: Learned recency weights w for several correlation probabilities p. Weights tend to monotonically decrease, which is consistent with our recency bias observations in Section 2.3. An exception is the coauthorship datasets which see weight increases for large indices. This exception is likely due to prolific individuals who publish many papers, as these tail weights would play no role for individuals without a large number of publications. We also see bifurcations in the recency weights in TAGS-MATHOVERFLOW and CONTACT-HIGH-SCHOOL, which align with different sides of the optimal value of p in Figure 5.

results in Section 2.3 on recency bias. This will also serve as a basis for some of our theoretical analysis in the following section. However, the coauthorship weights exhibit an increase at large indices (e.g., near index 100 for COAUTH-GEOLOGY). This is likely due to prolific authors in the dataset. Most authors in the dataset have fewer than 100 papers, so the weights above that index play no role in the likelihood of those sequences of sets. On the other hand, highly prolific authors could create long-term connections. This suggests that personalized weight parameters could be useful to develop better models.

Both TAGS-MATHOVERFLOW and CONTACT-HIGH-SCHOOL exhibit bifurcations in the learned recency weights. The two groups corresponds to the two sides of the optimal correlation probability p (see Figure 5). Thus, these datasets might be exhibiting two types of repeat behavior; exploring this is an avenue for future research.

5 ASYMPTOTIC TIPPING BEHAVIOR

In this section we study the asymptotic behavior of a simple instance of our process in which every set has size two. We study the event that at some time, a particular pair occurs at every future timestep; we will call this the *tipping event* after which no other pairs appear. Figure 7 illustrates this sequence of events. We will show that, similar to the single-item copying case [4], a strict dichotomy occurs: if $\sum_{i=1}^{\infty} w_i$ is bounded then eventually only a single pair will occur forever, and all other pairs will occur only finitely many times. On the other hand, if the weight sum is unbounded, then every pair occurs infinitely often. We begin by showing the first case.

Let h be the length of the history before a candidate tipping event. Assume that the same pair has occurred $j-1$ times consecutively since the candidate tipping event. We wish to lower bound the probability q_j that this pair will occur again for the jth time. Recall that the algorithm to generate a subset at this timestep will repeatedly perform a selection event until the correct size of subset

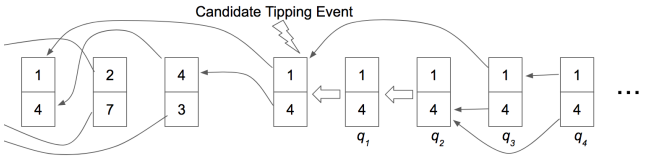

Figure 7: After a tipping event, a single pair occurs forevermore. Each new occurrence of this pair may result from copying individual elements from after the tipping point or by copying an entire pair from after the tipping point (indicated by block arrows). Theorem 5.3 shows that if the sum of the recency weights converges, every point has non-zero probability of becoming a tipping point, hence the process must eventually tip.

(in this case, size two) has been produced. Define $W_j = \sum_{i=1}^{j+h} w_i$ and $\Delta_j = W_{j+h} - W_j$. We now define three events on the outcome of a single selection event, with their probabilities, as follows:

Name	Meaning	Equation
pick1	the next choice selects a single item from after the tipping point	$p_1 = \frac{2p(1-p)W_j}{W_{j+h}}$
pick2	the next choice selects both of the target items from after the tipping point	$p_2 = \frac{p^2 W_j}{W_{j+h}}$
old	the next choice selects one or more elements from before the tipping point	$p_3 = \frac{(1-(1-p)^2)\Delta_j}{W_{j+h}}$

We may now write the probability q_j of successfully copying the same pair for the jth time. There are two paths to success: the process may copy the entire pair, or may copy each element independently. For example, in Figure 7, q_1 and q_2 both arise due to copies of an entire pair, while q_3 and q_4 arise due to copying of individual elements from after the candidate tipping point.

We consider the first time the process copies at least one element into the new pair; notice that the events pick1, pick2, and old are disjoint and cover all such cases. Hence, with probability $p_2/(p_1 + p_2 + p_3)$, the process succeeds in its first copy; with probability $p_3/(p_1 + p_2 + p_3)$, the process fails; and with remaining probability $p_1/(p_1+p_2+p_3)$, the process successfully copies a single element, and success is then dependent on copying the second element before copying an element from the h timesteps before the candidate tipping event. In the last case, a pick2 event must lead to success of the process, while a pick1 event will succeed only half the time (the other half, the process duplicates the already-chosen element, and leads to another round). Thus, the overall probability of event q_j may be written as: $q_j = \frac{p_2}{p_1+p_2+p_3} + \frac{p_1}{p_1+p_2+p_3}\frac{p_1/2+p_2}{p_1/2+p_2+p_3}$.

Note that $p_1+p_2+p_3 = p(2-p)$; this is expected, as it represents all events that copy at least one element, which occurs with probability $1-(1-p)^2 = p(2-p)$. We now show the following bound on q_j:

Lemma 5.1. $q_j \geq \left(\frac{W_j}{W_j+2\Delta_j}\right)^2$.

Proof. Using the expressions for p_1, p_2, and p_3, we get

$$\begin{aligned} q_j &= \frac{p_2}{p_1+p_2+p_3} + \left(\frac{p_1}{p_1+p_2+p_3}\right)\left(\frac{p_1/2+p_2}{p_1/2+p_2+p_3}\right) \\ &= \frac{p^2 W_j}{p(2-p)W_{j+h}} + \left(\frac{2p(1-p)W_j}{p(2-p)W_{j+h}}\right)\left(\frac{(p^2+p(1-p))W_j}{(pW_j+p(2-p)\Delta_j)}\right) \\ &= \frac{W_j}{W_{j+h}}\left[\frac{W_j}{W_j+(2-p)\Delta_j}\right] \\ &\geq \frac{W_j}{W_{j+h}}\left[\frac{W_j}{W_j+2\Delta_j}\right] \geq \left(\frac{W_j}{W_j+2\Delta_j}\right)^2. \quad \square \end{aligned}$$

For the remainder of the analysis, we require a technical bound:

Lemma 5.2. $\log(1-\frac{2\Delta_j}{W_j+2\Delta_j}) \geq -\frac{2W_\infty}{w_1}\frac{2\Delta_j}{W_j+2\Delta_j}$.

Proof. Let $x_j = \frac{2\Delta_j}{W_j+2\Delta_j}$. Observe that, as q_j values are non-increasing, x_j is maximized at $j=1$:

$$x_j \leq \frac{2\Delta_1}{w_1+2\Delta_1} \leq \frac{2(W_\infty-w_1)}{W_\infty+(W_\infty-w_1)} = 1-\frac{w_1}{W_\infty+(W_\infty-w_1)} \leq 1-\frac{w_1}{2W_\infty}.$$

Therefore, using the identity that $\log(1-x) \geq -\alpha x$ for $0 \leq x \leq 1-1/\alpha$, we conclude that $\log(1-x_j) \geq -\frac{2W_\infty}{w_1}x$ for all j. $\square$

We may now show that there is positive probability of tipping.

Theorem 5.3. *If $W_\infty < \infty$ then with probability 1, only a single pair will occur infinitely often.*

Proof. The probability that a candidate tipping point is a true tipping point is given by the product of the q_j's, which we now show is positive:

$$\begin{aligned} \log\prod_{j=1}^{\infty} q_j &= \sum_{j=1}^{\infty}\log(q_j) \\ &\geq 2\sum_j \log\left(\frac{W_j}{W_j+2\Delta_j}\right) \quad \text{(Lemma 5.1)} \\ &= 2\sum_j \log\left(1-\frac{2\Delta_j}{W_j+2\Delta_j}\right) \\ &\geq \frac{-2w_1}{W_\infty}\sum_j \frac{2\Delta_j}{W_j+2\Delta_j} \quad \text{(Lemma 5.2)} \\ &= \frac{-2w_1}{W_\infty}\sum_j \frac{2\sum_{i=j+1}^{j+h} w_i}{W_j+2\Delta_j} \\ &\geq \frac{-2w_1}{W_\infty}\sum_j \frac{2hw_j}{2\Delta_1} \\ &= \frac{-2w_1}{W_\infty}2h\frac{W_\infty}{2\Delta_1} = \frac{-2w_1 h}{\Delta_1} > -\infty. \end{aligned}$$

We have now shown that if $W_\infty < \infty$ then all but one pair will eventually disappear. The remaining part of the dichotomy requires us to show that for $W_\infty = \infty$, all items will occur infinitely often. This follows as an immediate consequence of Anderson et al. [4, Lemma 2]. This prior result applies to single-item copying, but the same proof holds for any bounded set size. $\square$

6 RELATED WORK

Repeat behavior has a long history in psychology and marketing science [9, 14, 15, 20, 23]. In those domains, repeat behavior might be purchasing the same product several times. However, this prior work focuses on individual items—rather than sets—and the datasets are nowhere near the scale of those analyzed here. However, it is not surprising that we also see repeat behavior with sets. For example, social groups are often formed from individuals that one is already familiar with [13]. Repeats in the email, contact, and coauthorship data are consistent with this phenomenon.

Repeat behavior has also been studied in the context of the Web, including repeat search queries [28, 29], Web browsing revisitation patterns [1, 2], short-term repeat consumption [8], and return times to user-item interactions [16]. Most closely related to this paper are prior models of consumption sequences that incorporate repeat behavior [4, 6]. This past work studied item-level (i.e., not set-level) consumption, and the datasets and models differ substantially.

Set-based techniques have also recently been used in a number of machine learning contexts, including embedding methods [24], deep learning [31], and discrete choice models [7]. While related in spirit, these techniques do not apply to the sequence data studied here. Finally, set evolution models appear in theoretical computer science and probability theory [21, 22, 25]. There is still a large gap between this theory and the practical data modeling applications, but the ideas provide interesting avenues for future research.

7 DISCUSSION

This paper proposes the Correlated Repeated Unions (CRU) model for repeat behavior in sequences of sets. The model was designed to capture three empirical findings: (i) exact and partial repeats of sets are extremely common in data, (ii) correlation of subsets in sequences of sets, and (iii) recency bias. A key property of the CRU model is that it uses a sampling algorithm which induces a probability distribution over repeat sets that makes likelihood computation and model parameter optimization tractable. After learning model parameters, we see substantial likelihood gains over a baseline model that does not explicitly incorporate set structure. We also found that the optimal correlation parameter p was different

across datasets but the same within domains. Our theoretical results demonstrate that the CRU model is amenable to analysis, and we envision that the CRU model will serve as a starting point for the mining, modeling, and analysis of sequences of sets data.

Code accompanying this paper is available at
https://github.com/arbenson/Sequences-Of-Sets.

ACKNOWLEDGMENTS

ARB supported in part by a Simons Investigator Award and NSF TRIPODS Award #1740822.

REFERENCES

[1] Eytan Adar, Jaime Teevan, and Susan T. Dumais. 2008. Large scale analysis of Web revisitation patterns. In *Proceeding of the twenty-sixth Annual CHI Conference on Human Factors in Computing Systems*. ACM Press, 1197–1206. https://doi.org/10.1145/1357054.1357241

[2] Eytan Adar, Jaime Teevan, and Susan T. Dumais. 2009. Resonance on the Web: Web dynamics and revisitation patterns. In *Proceedings of the 27th international conference on Human factors in computing systems*. ACM Press, 1381–1390. https://doi.org/10.1145/1518701.1518909

[3] Gediminas Adomavicius and Alexander Tuzhilin. 2005. Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering* 17, 6 (2005), 734–749. https://doi.org/10.1109/tkde.2005.99

[4] Ashton Anderson, Ravi Kumar, Andrew Tomkins, and Sergei Vassilvitskii. 2014. The dynamics of repeat consumption. In *Proceedings of the 23rd International Conference on World Wide Web*. ACM Press, 419–430. https://doi.org/10.1145/2566486.2568018

[5] Austin R Benson, Rediet Abebe, Michael T Schaub, Ali Jadbabaie, and Jon Kleinberg. 2018. Simplicial closure and higher-order link prediction. *arXiv:1802.06916* (2018). https://arxiv.org/abs/1802.06916

[6] Austin R. Benson, Ravi Kumar, and Andrew Tomkins. 2016. Modeling User Consumption Sequences. In *Proceedings of the 25th International Conference on World Wide Web*. ACM Press, 519–529. https://doi.org/10.1145/2872427.2883024

[7] Austin R. Benson, Ravi Kumar, and Andrew Tomkins. 2018. A Discrete Choice Model for Subset Selection. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*. ACM Press, 37–45. https://doi.org/10.1145/3159652.3159702

[8] Jun Chen, Chaokun Wang, and Jianmin Wang. 2015. Will You "Reconsume" the Near Past? Fast Prediction on Short-Term Reconsumption Behaviors. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*. 23–29. https://www.aaai.org/ocs/index.php/AAAI/AAAI15/paper/view/9491

[9] Chao-Min Chiu, Meng-Hsiang Hsu, Hsiangchu Lai, and Chun-Ming Chang. 2012. Re-examining the influence of trust on online repeat purchase intention: The moderating role of habit and its antecedents. *Decision Support Systems* 53, 4 (2012), 835–845. https://doi.org/10.1016/j.dss.2012.05.021

[10] Alan Collins, Chris Hand, and Maggie Linnell. 2008. Analyzing repeat consumption of identical cultural goods: some exploratory evidence from moviegoing. *Journal of Cultural Economics* 32, 3 (2008), 187–199. https://doi.org/10.1007/s10824-008-9072-0

[11] John Duchi, Shai Shalev-Shwartz, Yoram Singer, and Tushar Chandra. 2008. Efficient projections onto the ℓ_1-ball for learning in high dimensions. In *Proceedings of the 25th International Conference on Machine Learning*. ACM Press, 272–279. https://doi.org/10.1145/1390156.1390191

[12] Marion M. Hetherington, Ali Bell, and Barbara J. Rolls. 2000. Effects of repeat consumption on pleasantness, preference and intake. *British Food Journal* 102, 7 (2000), 507–521. https://doi.org/10.1108/00070700010336517

[13] Pamela J Hinds, Kathleen M Carley, David Krackhardt, and Doug Wholey. 2000. Choosing Work Group Members: Balancing Similarity, Competence, and Familiarity. *Organizational Behavior and Human Decision Processes* 81, 2 (2000), 226–251. https://doi.org/10.1006/obhd.1999.2875

[14] Jacob Jacoby and David B. Kyner. 1973. Brand Loyalty vs. Repeat Purchasing Behavior. *Journal of Marketing Research* 10, 1 (1973), 1–9. https://doi.org/10.2307/3149402

[15] Barbara E. Kahn, Manohar U. Kalwani, and Donald G. Morrison. 1986. Measuring Variety-Seeking and Reinforcement Behaviors Using Panel Data. *Journal of Marketing Research* 23, 2 (1986), 89–100. https://doi.org/10.2307/3151656

[16] Komal Kapoor, Mingxuan Sun, Jaideep Srivastava, and Tao Ye. 2014. A hazard based approach to user return time prediction. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM Press, 1719–1728. https://doi.org/10.1145/2623330.2623348

[17] Bryan Klimt and Yiming Yang. 2004. The Enron Corpus: A New Dataset for Email Classification Research. In *Machine Learning: ECML 2004*. Springer Berlin Heidelberg, 217–226. https://doi.org/10.1007/978-3-540-30115-8_22

[18] Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. 2007. Graph evolution: Densification and shrinking diameters. *ACM Transactions on Knowledge Discovery from Data* 1, 1 (2007). https://doi.org/10.1145/1217299.1217301

[19] Rossana Mastrandrea, Julie Fournet, and Alain Barrat. 2015. Contact Patterns in a High School: A Comparison between Data Collected Using Wearable Sensors, Contact Diaries and Friendship Surveys. *PLOS ONE* 10, 9 (2015), e0136497. https://doi.org/10.1371/journal.pone.0136497

[20] Leigh McAlister. 1982. A Dynamic Attribute Satiation Model of Variety-Seeking Behavior. *Journal of Consumer Research* 9, 2 (1982), 141–150. https://doi.org/10.1086/208907

[21] Ben Morris and Yuval Peres. 2003. Evolving sets and mixing. In *Proceedings of the thirty-fifth ACM Symposium on Theory of Computing*. ACM Press, 279–286. https://doi.org/10.1145/780542.780585

[22] Ben Morris and Yuval Peres. 2005. Evolving sets, mixing and heat kernel bounds. *Probability Theory and Related Fields* 133, 2 (2005), 245–266. https://doi.org/10.1007/s00440-005-0434-7

[23] Rebecca K. Ratner, Barbara E. Kahn, and Daniel Kahneman. 1999. Choosing Less-Preferred Experiences For the Sake of Variety. *Journal of Consumer Research* 26, 1 (1999), 1–15. https://doi.org/10.1086/209547

[24] Maja Rudolph, Francisco Ruiz, Stephan Mandt, and David Blei. 2016. Exponential family embeddings. In *Advances in Neural Information Processing Systems*. 478–486. https://papers.nips.cc/paper/6571-exponential-family-embeddings

[25] Laurent Saloff-Coste. 2004. Random Walks on Finite Groups. In *Probability on Discrete Structures*. Springer Berlin Heidelberg, 263–346. https://doi.org/10.1007/978-3-662-09444-0_5

[26] Arnab Sinha, Zhihong Shen, Yang Song, Hao Ma, Darrin Eide, Bo-June (Paul) Hsu, and Kuansan Wang. 2015. An Overview of Microsoft Academic Service (MAS) and Applications. In *Proceedings of the 24th International Conference on World Wide Web*. ACM Press, 243–246. https://doi.org/10.1145/2740908.2742839

[27] Juliette Stehlé, Nicolas Voirin, Alain Barrat, Ciro Cattuto, Lorenzo Isella, Jean-François Pinton, Marco Quaggiotto, Wouter Van den Broeck, Corinne Régis, Bruno Lina, and Philippe Vanhems. 2011. High-Resolution Measurements of Face-to-Face Contact Patterns in a Primary School. *PLOS ONE* 6, 8 (2011), e23176. https://doi.org/10.1371/journal.pone.0023176

[28] Jaime Teevan, Eytan Adar, Rosie Jones, and Michael Potts. 2006. History repeats itself: Repeat queries in Yahoo's logs. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM Press, 703–704. https://doi.org/10.1145/1148170.1148326

[29] Jaime Teevan, Eytan Adar, Rosie Jones, and Michael A. S. Potts. 2007. Information re-retrieval: Repeat queries in Yahoo's logs. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM Press, 151–158. https://doi.org/10.1145/1277741.1277770

[30] Hao Yin, Austin R. Benson, Jure Leskovec, and David F. Gleich. 2017. Local Higher-Order Graph Clustering. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM Press, 555–564. https://doi.org/10.1145/3097983.3098069

[31] Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Ruslan R Salakhutdinov, and Alexander J Smola. 2017. Deep sets. In *Advances in Neural Information Processing Systems*. 3394–3404. https://papers.nips.cc/paper/6931-deep-sets