Virtual People: Actionable Reach Modeling

Evgeny Skvortsov, Jim Koehler

Google Inc.

June 21, 2019

Abstract

We introduce a method for serving models that estimate reach and demographics of cross-device online audiences. The method assigns *virtual people* identifiers to events. The reach of a set of events is estimated as a simple count of distinct virtual people assigned to these events. This allows efficient serving of reach models at large scale. We formalize what it means for a reach model to be actionable and prove that any actionable reach model is equivalent to some virtual people model. We present algorithms for encoding reach models with virtual people and show that a wide variety of modeling techniques can be implemented with this approach.

1 Introduction

Koehler, Skvortsov, and Vos (2013) [1] (KSV) presents a method for measuring reach and frequency of online ad campaigns by audience attributes for one device (or cookie) type. The method combines ad server logs, publisher provided user data (PPD), census data, and a representative panel to produce corrected cookie and impression counts by these audience attributes. The method corrects for cookie issues such as deletion and sharing, and for PPD issues such as non-representativeness and poor quality of demographic labels. It also proposes a model that converts cookie counts to user counts.

Koehler, Skvortsov, Ma, Liu (2016) [2] (KSML) extends the method to today's world of multiple device types such as desktop, smartphone, and tablet. A formulation for converting multiple cookie counts to people counts was proposed. The article introduced the concept of an Activity Distribution Function (ADF), which describes the probability of a person generating cookies of each type. A theory relating ADFs to matching cross-device reach functions was presented and it was shown that ADFs can be approximated by a mixture of Dirac delta functions [3] which can be estimated empirically using panel data. A natural extension of the demographic correction to multiple devices was presented as well.

This paper presents a technology that implements the measuring methodologies [1] and [2] in large scale production systems efficiently. We convert reach and demographic correction models into assignments of virtual people to each of the events in the logs. Each virtual person has demographic attributes (age and gender) assigned to them. Total reach of an audience (ad campaign, web site, YouTube channel etc) can be estimated as a simple count of unique virtual people assigned to the corresponding set of events. Demographic composition of an audience is estimated as the demographic composition of the set of virtual people.

Encoding a reach and demo correction model as virtual people assignment comes with multiple benefits. The most important benefits are

1. Protection of user privacy.
2. Simplicity of engineering implementation and maintenance.
3. Guarantee of reach and demographics model self-consistency.

Privacy. Attempting to identify the set of cookies that belong to a particular person would entail privacy risks, even if only aggregate level reach is reported. Techniques outlined here assign cookies to people according to random distributions with large domains, thus the probabilty that cookies of a real person are assigned to the same virtual person is low. For almost all virtual people no two cookies assigned to them belong to the same real person. Yet through these assignments, large aggregate counts of virtual people closely approximate counts of real people.

Simplicity. To understand the simplicity of the virtual people model we provide a brief overview of an infrastructure required for an aggregate model application.

Aggregate methods usually require unique user id (cookie, logged-in id) counting for each audience report (campaign, site etc). The counts need to be obtained for each modelled sub-dimension. For example, to estimate the reach of demographic categories with an aggregate model, counts of unique identifiers needs to be obtained for each demographic label separately. Then once these aggregate counts are accumulated a call to the aggregate model needs to be made. This call can be done via a remote call to a server containing the model (e.g. via HTTP request), or via a library function call. Both approaches have pros and cons and require engineering resources to maintain, require workflows to be developed for model rollout, change of the labeling algorithms etc. For instance if the source of demographic labels changes then the aggregate model likely needs to change and these two changes need to be synchronized.

In contrast, virtual people model requires a single unique counting to be done for each audience and the count can immediately be reported. For years the Internet industry built systems to report impression and unique cookie counts. Any such system can be updated to unique people reach reporting by simply changing the cookie identifier to virtual person identifier.

Consistency. It is important that a reach measurement is *actionable*, i.e. an advertiser should be able to inspect slices of traffic and assess whether these slices are beneficial for reach, or should be turned off. For a model, actionabilty requires that the model should be applicable to an arbitrary set of events and should produce numbers for those sets that are consistent with each other. For instance, the number of people reached via desktop must not be greater than the total number of people reached by campaign. While this expectation is highly intuitive it is violated (at least for some audiences) by many off-the-shelf machine learning models. For example, a multi-layer perceptron would have no guarantee that its output reach for the first day of the week would always be smaller than its output reach for the whole week.

Virutal people models are inherently consistent for arbitrary slices and are thus actionable. Moreover, in Section 3 we show (under a reasonable formal definition) that any actionable model is equivalent to a virtual people model.

The rest of the paper is organized as follows. In Section 2 we review the cross-device measurement methods presented in [1] and [2]. In Section 3 we give a formal definition of actionable models, we then formulate Theorem 1 that any actionable model is equivalent to a virtual people model (the proof of the theorem is given in the Appendix). Section 4 presents three algorithms for virtual people assignment. We start with an algorithm required to implement a multi-device reach function with no demo reporting. The second algorithm allows implementation of a reach function with demo correction via a stochastic redistribution matrix. Algorithm 3 is the most general and allows implementation of a variety of modeling methodologies. Next in Section 5 we illustrate the generality of Algorithm 3 discussing three modeling approaches and how they can be encoded as virtual people by the algorithm. In particular in subsection 5.1 we show how the demographic correction methodology [2] can be implemented via virtual people. We conclude with Section 6 running simulations verifying on two examples that virtual people indeed produces results that closely follow the aggregate model application.

Appendix contains proof of Theorem 1 and Notation Glossary.

2 Review of Cross-Device Measurement

2.1 Demographic Correction Model

In [1] it was proposed to estimate the distribution (in form of a vector) of true demographics of cookies belonging to audience $\hat{\boldsymbol{y}}$ from distribution of publisher provided labels of the cookies $\boldsymbol{x}$ as

$$\hat{\boldsymbol{y}} = (1-\alpha)A\boldsymbol{x}/||A\boldsymbol{x}||_1 + \alpha B\boldsymbol{x} \tag{1}$$

where A is a $D \times D$ "correction" matrix, B is a D x D left-stochastic matrix[1], and α represents the fraction of cookies (impressions) for the given audience either served on the publisher's site or via cookie targeting using the PPD. Hence, $1-\alpha$ represents the fraction of unlabeled cookies (or unlabeled impressions) for that audience. Matrices A and B are trained from the panel data and are fixed in the model. Parameters $\boldsymbol{x}$ and α are computed from logs data for each audience.

This approach was generalized in [2] for multiple devices. The generalization amounts to an application of Equation 1 for each device, i.e. for device b we would have

$$\hat{\boldsymbol{y}}^b = (1-\alpha^b)A\boldsymbol{x}^b/||A\boldsymbol{x}^b||_1 + \alpha B\boldsymbol{x}^b \tag{2}$$

where $\boldsymbol{x}$ is the distribution of labels of the audience on device b, α is the fraction of cookies for the given audience on device b that has an assigned PPD label and $\hat{\boldsymbol{y}}^b$ is the estimated distribution of cookies with respect to true demographics.

2.2 Mapping Cookies to Users

It was shown in [2] that reach functions can be modeled via distributions describing activity of users generating cookies of different types. Assume that there is an underlying population of people, and each person has a certain probability of generating a cookie of each type. Let the (multivariate)

[1] A left-stochastic matrix is a square matrix with non-negative entries and columns that sum to one.

probability distribution $\mathcal{A}$ model the heterogeneity of these probabilities. Distribution $\mathcal{A}$ can be converted to a cross-device reach surface using the formula

$$R_{\mathcal{A}}(\boldsymbol{t}) = \int_{\boldsymbol{x} \in (\mathbb{R}^{+})^{\dim \boldsymbol{x}}} \mathcal{A}(\boldsymbol{x}) \cdot (1 - e^{-\boldsymbol{t}\boldsymbol{x}}) d\boldsymbol{x}. \tag{3}$$

Here $\boldsymbol{x}$ and $\boldsymbol{t}$ are vectors of equal dimensionality and i-th coordinate of $\boldsymbol{x}$ corresponds to activity of users generating the i-th cookie while i-th coordinate of *cookie reach vector* $\boldsymbol{t}$ corresponds to the number of cookies reached by a web audience, scaled by the total number of people in the modeled population. Distribution $\mathcal{A}$ is called the *Activity Distribution Function* (ADF). Function R is called a *reach function* and it maps the vector cookie reach vector to the total number of unique people reached.

Function $1 - R_{\mathcal{A}}(\boldsymbol{t})$ is known as the *(multivariate) moment generating function* (mgf) of distribution $\mathcal{A}$. Moments and moment generating functions are useful tools of modern statistics and have found applications in multiple areas including climate science [5] and astrophysics [4]. While moment generating functions usually play a role in an aggregate abstract description of a studied distribution, in reach modeling the moment generating function and distribution switch roles. The reach curve (mgf) is fit directly to panel data and the distribution (ADF) is used as an abstract representaiton of it.

A few forms of ADF were discussed in [2]. In this paper we will use the Dirac Mixture and Exponential Bow.

Dirac Mixture is a linear combination of Dirac Deltas, i.e. functions of finite volume that are zero everywhere except for an infinitely small area around a certain point. A Dirac Mixture Activity Density Function has an interpretation that the population consists of a finite number of sub-populations. Each sub-population has a different cookie generation behavior, but people within a sub-population have identical cookie generation behavior. Sub-population k is characterized by a vector $\boldsymbol{x}_k$ and its coordinate $\boldsymbol{x}_k^{\tau}$ is the activity of people in the pool generating user identifiers of type τ. For instance, $\boldsymbol{x}_1^{mobile}$ could be activity of people in the first pool generating mobile cookies.

Dirac Mixture Activity Density Function is defined as

$$\mathcal{D}(\boldsymbol{x}) = \sum_{k} \alpha_k \delta(\boldsymbol{x} - \boldsymbol{x_k}). \tag{4}$$

Substituting Dirac Mixture in Equation 3 and doing integration we obtain the formula for reach

$$R_{\mathcal{D}}(\boldsymbol{t}) = \sum_{k=1}^{n} \alpha_k (1 - e^{-\boldsymbol{x_k} \cdot \boldsymbol{t}}). \tag{5}$$

Dirac Mixtures are of interest to us because they can approximate reach of an arbitrary ADF with arbitrary precision and can be efficiently trained to the observed data. Moreover it turns out that Dirac Mixture reach functions can be naturally implemented as virtual people models.

Exponential Bow is a reach curve corresponding to an ADF equal to the exponential distribution. That is

$$\mathcal{A}_{ExponentialBow}(x) = \frac{e^{-\frac{x}{\kappa}}}{\kappa}$$

and consequently

$$ExponentialBow(t) = \mathcal{R}_{\mathcal{A}_{ExponentialBow}}(t) = \frac{\kappa t}{\kappa t + 1}.$$

Coefficient κ is the only parameter of an Exponential Bow and it happens to be equal to the derivative of the corresponding reach function at 0.

Exponential Bow is of interest because it is observed to match Desktop reach curves reasonably closely. In this paper we will use it as an example of a non-Dirac Mixture reach function when encoding models with virtual people.

3 Actionable Reach Models

In practice it is not enough to simply report after the fact on a campaign, the model needs to be *actionable*. That is the model should give an advertiser the power to plan and monitor reach performance of the campaign. This makes an initial campaign plan more efficient and enables the advertiser to do real-time adjustments to maximize their goals. To enable this planning the model should support slicing on arbitrary dimensions (e.g. device type, time intervals, time of day etc). The model should also produce internally consistent numbers for these slices.

We propose the following definition.

Definition 1. *A model is* actionable *if and only if*

- *The model is applicable to any set of events. (In industry language: "The model allows arbitrary slicing.")*
- *The model is self-consistent.*

Self-consistency is expected by any agent/analyst working with the model. For example, it is expected that the model over a month reports reach of more people than it reports over the first week of the month. It is also expected that reach over a week is smaller than the sum of the reach over all days of the week. Yet most standard machine-learning techniques (neural networks for instance) have no a priori guarantee of self-consistency.

How to achieve self-consistency of a model? One way is to encode the model as a *virtual people* assignment. For each event we assign a virtual person identifier and the model simply reports the number of unique identifiers corresponding to the set of events. It is obvious that such model is *actionable.*

It may seem that constraining modeling techniques to virtual people assignment is too much of a burden on model design. But it turns out that any actionable model is equivalent to a model based on virtual people.

Theorem 1. *Let $\mathcal{E}$ be a set of events and a function $F : 2^{\mathcal{E}} \to \mathbb{Z}$ mapping sets of events to integers satisfy the following properties for any sets of events A, B, C:*

1. Monotonicity*: $F(A \cup B) \geq F(A)$.*

2. Convexity*:* $F(A \cup B) \leq F(A) + F(B)$*.*

3. Attribution of incremental reach*: If* $F(A \cup B \cup C) > F(A)$ *then* $F(A \cup B) > F(A)$ *or* $F(A \cup C) > F(A)$*.*

4. Disjoint sets property*: If* $F(A \cup B) = F(A) + F(B)$ *and* $F(A \cup C) = F(A) + F(C)$ *and* $F(B \cup C) = F(B) + F(C)$ *then* $F(A \cup B \cup C) = F(A) + F(B) + F(C)$*.*

Let also for any individual impression i *we have* $F(\{i\}) \leq 1$*.*

Then there exists a set $\mathcal{V}$ *(which we call* a set of virtual people*) and a mapping* $V : \mathcal{E} \to \mathcal{V}$*, such that for any set of events* A *we have* $F(A) = |V(A)|$*. Recall that by definition* $V(A) = \{V(i)|i \in A\}$*. In other words* $F(A)$ *is equal to the count of virtual people corresponding to elements of* A*.*

The proof of Theorem 1 is given in Appendix.

Fortunately virtual people assignments can be efficiently obtained and a wide variety of modeling techniques can be encoded via virtual people, such algorithms are discussed in Section 4.

4 Virtual People Assignment

In this section we introduce algorithms of virtual people assignment. These algorithms assign virtual people identifiers to events in the logs in such a way that reach of a set of events can be estimated by counting the number of unique identifiers assigned to these events. We start with algorithms that handle simple intuitive models and finally introduce Algorithm 3 that can implement a wide variety of modeling techniques, which will be illustrated in Section 5.

Formally, for a set of all events in the logs $\mathcal{E}$ and an abstract set of virtual people $\mathcal{V}$, of cardinality $|\mathcal{V}|$ equal to the size of the modeled population, a virtual people assignment function $V : \mathcal{E} \to \mathcal{V}$ defines reach of set of events $E \subseteq \mathcal{E}$ as

$$\mathcal{R}_V(E) = |\{V(e)|e \in E\}|. \tag{6}$$

We allow V to be undefined for some events, then these events bring no incremental reach. This is required to implement reach curves that have partial derviative at zero less than one.

4.1 Assignment for Dirac Mixture

An assignment that matches a reach surface defined by an Dirac Mixture ADF (see Equation 5) can be obtained as follows.

Recall that $\mathcal{D}(\boldsymbol{x}) = \sum_k \alpha_k \delta(\boldsymbol{x} - \boldsymbol{x}_k)$ and let $\boldsymbol{t}(C)$ denote the vector of counts of cookies in C for each cookie type. Then a virtual people assignment V follows a surface of Dirac Mixture $\mathcal{D}$ if and only if

$$\frac{\mathcal{R}_V(E)}{|\mathcal{V}|} \approx R_{\mathcal{D}}(\boldsymbol{t}(C(E))), \tag{7}$$

where $C(E)$ is the set of user identifiers (cookies, log-in identifiers etc) corresponding to the set of events E.

Without loss of generality we will assume that we model a single country of population m. For simplicity we shall also assume that for any i the number $m \cdot \alpha_i$ is an integer. Algorithm 1 performs the assignment of virtual people to cookies.

```
input  : total population per census data $m$, set of events $\mathcal{E}$, correspondence of
         events to user identifiers $C : \mathcal{E} \to \mathcal{C}$, correspondence of user identifiers to
         types of cookies $T : \mathcal{C} \to \mathcal{T}$, Dirac Mixture Activity Distribution
         $\mathcal{D}(\boldsymbol{x}) = \sum_{k=1}^{n} \alpha_k \delta(\boldsymbol{x} - \boldsymbol{x}_k)$,
output : a set of virtual people identifiers $\mathcal{V}$ and a mapping $V : \mathcal{C} \to \mathcal{V}$
// Allocating pools of virtual people per Dirac Delta.
let $r = 1$;
for $i \in \{1 \ldots n\}$ do
    let $P_i$ be the set of $m \cdot \alpha_i$ integers $\{r, \ldots, r + m \cdot \alpha_i - 1\}$;
    let $r = r + m \cdot \alpha_i$;
end
let $\mathcal{V} = \bigcup_{i=1}^{n} P_i$;
assert $\mathcal{V}$ is a set of integers $\{1 \ldots m\}$;
// Assigning virtual people to log events.
for $e \in \mathcal{E}$ do
    let $c = C(e)$;
    let $\tau = T(c)$;
    let $\kappa_\tau = \sum_k \alpha_k \cdot \boldsymbol{x}_k^\tau$;
    with probability $1 - \kappa_\tau$ assign no virtual person to $e$ and continue to next iteration of
     the for loop;
    sample $i$ randomly, according to distribution $P(i) = \frac{\alpha_i \cdot \boldsymbol{x}_i^\tau}{\kappa_\tau}$ ;
    sample $p$ uniformly at random from $P_i$ ;
    let $V(e) = p$
end
```

Algorithm 1: Assignment of virtual people following a Dirac Mixture reach surface

Each cookie should be mapped to the same virtual person for all events, so both random samplings in Algorithm 1 should be done deterministically, e.g. using hash of the cookie id as a source of randomness.

From the definition of Dirac Mixture follows

Observation 1. *Assignment produced by Algorithm 1 follows Dirac Mixture $\mathcal{D}$.*

It may be required that more than 1 virtual person is assigned to an event. In particular connected TV may have $\kappa > 1$, because it it is often located in a living room and is exposed to co-viewing. Extension of the algorithms to mutliple virtual people assignment can be done, but is beyond the scope of this paper.

4.2 Assignment for Dirac Mixture into People Categories

In practice the reach surface is not the only constraint that a reach function has to fullfill. The model may be enriched with information about the degree of reach penetration into certain categories (e.g. age, gender, location) of the population. Formally speaking we may be able to estimate the probability of a given cookie reaching certain segments of the population conditional on information that can be directly observed about the cookie, i.e.

$$\{P(\text{event reaches population category } \mathbb{c} \mid \text{cookie belongs to class } \ell)\}_{\ell \in \mathcal{L}, \mathbb{c} \in \mathbb{C}},$$

where $\mathbb{C}$ are some categories of population and $\mathcal{L}$ is the set of user identifier labels. These estimations can be made from panel data if categories of panelists are known.

To provide accurate consistent estimates per population category, virtual person assignment needs to be done per category. Or, equivalently, each virtual person has to have a category assigned to it. Otherwise by Theorem 1 we would be at risk of inconsistent reach numbers per some categories.

For example, a distribution of the audience over demographic buckets may be required to be reported.

Another example is that we can estimate a mapping from information available for the event (e.g. IP-address approximate location) to the distribution over some geographic regions. In important practical cases this mapping can in fact be virtually deterministic. For instance if we need to estimate the number of users reached in each state of the USA then approximate IP address location can be used to assign states directly. Errors due to inaccuracy of IP to geo mapping at such a coarse level would be insignificant.

Generally each category may have its own reach surface.

Algorithm 2 does the assignment assuming that we have training data that is granular enough to fit the mapping from cookie attributes to probability distribution over people categories as well as reach surfaces per category. This mapping is given to the algorithm via a matrix of conditional probabilities

$$\{r_{\mathbb{c},\ell} = P\left(person(c) \in \mathbb{c} | L(c) = \ell\right)\}_{\mathbb{c} \in \mathbb{C}, \ell \in \mathcal{L}}.$$

Each $r_{\mathbb{c},\ell}$ is the probability of a person who generated a cookie c to belong to the category $\mathbb{c}$ given than the cookie c has label $\ell \in \mathcal{L}$. In the simplest case the set of all labels $\mathcal{L}$ could have a similar structure to the set of categories, e.g. $\ell \in \mathcal{L}$ could be a demographic that the user declared when they logged-in while $\mathbb{c} \in \mathbb{C}$ could be their true demographic. But in general sets $\mathcal{L}$ and $\mathbb{C}$ could have a very different structure, for instance $\ell \in \mathcal{L}$ could be a combination of user's declared and inferred demographic, as well as a mask of their IP-address and $\mathbb{c} \in \mathbb{C}$ be a combination of their true demographic and geographic region.

Without loss of generality we assume that ADFs of all categories have the same dimensionality (as the max of dimensionalities can be used otherwise).

4.3 Generic framework: Dirac Mixture of People Categories

Algorithm 2 allows to do the estimate of reach in an arbitrary set of categories. Yet in practice the number of categories may be too large to divide the population into. Panel data is also too sparse for the analysis of large number of categories.

```
input   : set of events $\mathcal{E}$ and a set of categories of people $\mathbb{C}$,
          total population per category from census data $\{m_{\mathbb{c}}\}_{\mathbb{c}\in\mathbb{C}}$, correspondence of events to user identifiers $C : \mathcal{E} \to \mathcal{C}$,
          correspondence of user identifiers to labels $L : \mathcal{C} \to \mathcal{L}$,
          correspondence of user identifiers to types of identifiers $T : \mathcal{C} \to \mathcal{T}$, a matrix of probabilities $\{r_{\mathbb{c},\ell} = P(p(c) \in \mathbb{c} | L(c) = \ell)\}_{\mathbb{c}\in\mathbb{C},\ell\in\mathcal{L}}$,
          Dirac Mixture Activity Distribution per category
          $\{\mathcal{D}_{\mathbb{c}}(\boldsymbol{x}) = \sum_{k=1}^{n} \alpha_{k,\mathbb{c}} \delta(\boldsymbol{x} - \boldsymbol{x}_{k,\mathbb{c}})\}_{\mathbb{c}\in\mathbb{C}}$
output  : a set of virtual people identifiers partitioned by people categories $\mathcal{V} = \bigcup_{\mathbb{c}\in\mathbb{C}} \mathcal{V}_{\mathbb{c}}$
          and a mapping $V : \mathcal{C} \to \mathcal{V}$
// Allocating pools of virtual people per people categories and Dirac
   Deltas.
let $r = 1$;
for $\mathbb{c} \in \mathbb{C}$ do
    for $i \in \{1 \ldots n\}$ do
        let $P_{i,\mathbb{c}}$ be the set of $m_{\mathbb{c}} \cdot \alpha_{i,\mathbb{c}}$ integers $\{r, \ldots, r + m \cdot \alpha_{i,\mathbb{c}} - 1\}$;
        let $r = r + m \cdot \alpha_i$;
    end
    let $\mathcal{V}_{\mathbb{c}} = \bigcup_{i=1}^{n} P_{i,\mathbb{c}}$;
end
let $\mathcal{V} = \bigcup_{i=1}^{n} \bigcup_{\mathbb{c}\in\mathbb{C}} P_{i,\mathbb{c}}$;
assert $\mathcal{V}$ is a set of integers $\{1 \cdots \sum_{\mathbb{c}} m_{\mathbb{c}}\}$;
// Assigning virtual people to log events.
for $e \in \mathcal{E}$ do
    let $c = C(e)$;
    let $\ell = L(c)$;
    sample $\mathbb{c}$ randomly, according to distribution $P(\mathbb{c}) = r_{\mathbb{c},\ell}$;
    let $\tau = T(c)$;
    let $\kappa_{\tau,\mathbb{c}} = \sum_k \alpha_k \cdot \boldsymbol{x}^{\tau}_{k,\mathbb{c}}$;
    with probability $1 - \kappa_{\tau,\mathbb{c}}$ assign no virtual person to $e$ and continue to next iteration
     of the for loop;
    sample $i$ randomly, according to distribution $P(i) = \frac{\alpha_{i,\mathbb{c}} \cdot \boldsymbol{x}^{\tau}_{i,\mathbb{c}}}{\kappa_{\tau,\mathbb{c}}}$ ;
    sample $p$ uniformly at random from $P_i$;
    let $V(e) = p$
end
```

Algorithm 2: Assignment of Virtual People Per Category

Consider the situation of estimating of reach across 30 interest categories (e.g. "Sports Enthusiasts", "Movie Lovers" etc), where each interest category contains around half of the population. If we assume independence of categories we end up with 2^{30} individual combinatorial categories (for example one such category could be people interested in cars and movies, but nothing else), which means that we have to split the population into around one billion pools. Even in this example with only a few categories this approach becomes impractical, as small pools reduce the accuracy of the model and poses a risk to the privacy of users.

Instead of dividing the population into a set of discrete categories we can introduce a random distribution over categories for subsets of the population. Moreover, instead of having two partitions of the population – one for categories and one for Dirac Deltas – we can re-use the Dirac Deltas for category classification and fold-in the categorical features into user identifier types.

Formally, we assume that we have a Dirac Mixture $\mathcal{D}$, whose deltas are indexed by elements of some set Θ. We also have a mapping from this set to the set Ω of probability distributions over $\mathbb{C}$, i.e. $\omega : \Theta \to \Omega$ such that for a given dirac delta θ we have $\omega : \theta \mapsto \{P(\mathbb{c}|\theta)\}_{\mathbb{c}\in\mathbb{C}}$. Thus we arrive at Algorithm 3.

Assignment of category to a virtual person must be consistent across events, so sampling of $\mathbb{c}$ should be done deterministically, using hash of virtual person id as a source of randomness.

5 Examples of Modeling with Virtual People

Virtual people assignment (Algorithm 3 presented in the previous section) is generic and allows implementation of a variety of modeling approaches. We illustrate this flexibility in this section via examples of modeling techniques and how they fit the paradigm of Algorithm 3.

5.1 Cookie-level Demographic Correction

If we can infer the true demo distribution of users from their input demographic signals, e.g. combinations of declared age and gender, then such demographic correction technique can be encoded directly into matrix $r_{\mathbb{c},\ell}$ of Algorithm 2 or function T of Algorithm 3. But to implement the models in [1] and [2] we need to extrapolate the distribution from cookies with PPD demographics to unlabelled cookies via Equation (2).

In this case Algorithm 3 can be used by placing an arbitrary demo inference outside of the reach modeling framework and then taking output of this demo inference as an input signal for the reach model.

Let $L : \mathcal{C} \to \mathcal{L}$ be a correspondence of the set of all user identifiers to an arbitrary set of labels, containing information that we want to use for training a demographic model. For instance exposure to an ad campaign can be part of the label. First we build a model $M : \mathcal{L} \times \mathbb{D} \to \mathbb{R}$ that maps each user identifier label to a probability distribution over demographic categories $M : (\ell, \mathbb{d}) \mapsto P(\mathbb{d}|\ell)$.

Note that there are no limits on the techniques for this model, artificial neural networks can be used for it. In particular the set of labels may be infinite, e.g. labels could be real valued vectors.

We finally convert the model into a form $\tilde{L} : \mathcal{C} \to \mathbb{D}$ that can be digested by Algorithms 2 and 3. We set

```
input     : total population per census data $m$, set of events $\mathcal{E}$,
            correspondence of events to user identifiers $C : \mathcal{E} \to \mathcal{C}$,
            correspondence of user identifiers to types of identifiers $T : \mathcal{E} \to \mathcal{T}$,
            Dirac Mixture Activity Distribution $\mathcal{D}(\boldsymbol{x}) = \sum_{\theta \in \Theta}^{n} \alpha_\theta \delta(\boldsymbol{x} - \boldsymbol{x}_\theta)$,
            correspondence of Dirac Deltas to distributions over people categories:
            $\omega : \Theta \to \Omega$
output    : a set of virtual people identifiers $\mathcal{V}$:
            a mapping of events to virtual people $V : \mathcal{E} \to \mathcal{V}$,
            a mapping of virtual people to people categories: $\sigma : \mathcal{V} \to \mathbb{C}$
// Allocating pools of virtual people per Dirac Delta.
let $r = 1$ for $\theta \in \Theta$ do
    let $P_\theta$ be the set of $m \cdot \alpha_\theta$ integers $\{r, \dots, r + m \cdot \alpha_\theta - 1\}$;
    let $r = r + m \cdot \alpha_i$;
end
let $\mathcal{V} = \bigcup_{\theta \in \Theta} P_i$;
assert $\mathcal{V}$ is a set of integers $\{1 \dots m\}$;
// Assigning virtual people to log events.
for $e \in \mathcal{E}$ do
    let $c = C(e)$;
    let $\tau = T(e)$;
    let $\kappa_\tau = \sum_{\theta \in \Theta} \alpha_\theta \cdot \boldsymbol{x}_k^\tau$;
    with probability $1 - \kappa_\tau$ assign no virtual person to $e$ and continue to next iteration of
      the for loop;
    sample $\theta$ randomly, according to distribution $P(\theta) = \frac{\alpha_\theta \cdot \boldsymbol{x}_\theta^\tau}{\kappa_\tau}$;
    sample $p$ uniformly at random from $P_\theta$;
    let $V(e) = p$;
    sample $\mathbb{c}$ uniformly at random from distribution $P(\mathbb{c}) = \omega(\theta)$;
    let $\sigma(p) = \mathbb{c}$;
end
```

Algorithm 3: Assignment of Virtual People following a Dirac Mixture of People Categories

$$\tilde{L} : c \mapsto \mathbb{d},\ \text{with probability } M(L(c), \mathbb{d}),$$

where the random assignment is deterministically made, e.g. applying an appropriate hash function to the value of c.

The range of function $\tilde{L}$ is equal to $\mathbb{D}$ and it can be used as a finite set of labels in Algorithm 2 along with an identity matrix r.

Note that matricies A and B from Equation (2) can be used as a method to infer the demographics of cookies. Specifically matrix A can be used to determine the distributions of cookies with PPD labels and matrix B to determine distributions of cookies without labels for each advertising campaign. Thus we can implement models described by [1] and [2] via virtual people.

5.2 Independence assumption

An independence assumption is often used in modeling the interaction of different features if panel data is too sparse to measure interaction coefficients. Possibly because independence is the simplest version of maximal entropy it often matches observation once enough data is collected to actually perform the measurement.

In this section we consider modeling

- geo location with known $L_g : \mathcal{C} \to \mathcal{L}_g$ and r_g stochastically mapping $\mathcal{L}_g$ to $\mathbb{G}$ (see Algorithm 2),
- demographics with a known $L_d : \mathcal{C} \to \mathcal{L}_d$ and $r_d : \mathcal{L}_d \to \mathbb{D}$,
- cross-device reach surface

via the independence assumption.

We assume that cross device reach $R_{\text{x-device}}$ can be computed from reach on desktop R_{desktop} and reach on mobile R_{mobile} via the formula

$$R_{\text{x-device}} = R_{\text{desktop}} + R_{\text{mobile}} - R_{\text{desktop}} \cdot R_{\text{mobile}}.$$

We use the following activity distributions to define reach within each device

$$\mathcal{D}_{\text{desktop}} = \sum_{\theta \in \Theta_d}^{n} \alpha_\theta \delta(\boldsymbol{x} - \boldsymbol{x}_\theta) \ \text{ and } \ \mathcal{D}_{\text{mobile}} = \sum_{\theta \in \Theta_m}^{n} \alpha_\theta \delta(\boldsymbol{x} - \boldsymbol{x}_\theta).$$

We build T, $\mathcal{D}$ and ω to implement this model with Algorithm 3.

First we build an auxiliary set of categories of events as a simple Cartesian product:

$$\mathcal{L} = \mathcal{L}_g \times \mathcal{L}_d.$$

The set of cookie types is then defined by taking a product of the set of event categories onto the set of devices:

$$\mathcal{T} = \mathcal{L} \times \{\text{desktop}, \text{mobile}\}.$$

Thus each cookie type corresponds to a triple (g, d, b), where g is geo signal, d is demo signal and b is a device.

The set of Dirac Deltas is a similar Cartesian product

$$\Theta = \mathbb{G} \times \mathbb{D} \times \Theta_d \times \Theta_m.$$

Let $|\mathbb{c}|$ be the fraction of the population belonging to category $\mathbb{c}$.

Reach surface parameters α are computed from independence assumption as

$$\alpha_{\mathbb{g},\mathbb{d},\theta_d,\theta_m} = |\mathbb{g}| \cdot |\mathbb{d}| \cdot \alpha_d \cdot \alpha_m.$$

Reach surface activities $\boldsymbol{x}_{\mathbb{g},\mathbb{d},\theta_d,\theta_m}$ can also be computed from the independence assumption. Coordinate of $\boldsymbol{x}_{\mathbb{g},\mathbb{d},\theta_d,\theta_m}$ corresponding to cookie type (g, d, b) is equal to

$$r_g^{g,\mathbb{g}} \cdot r_d^{d,\mathbb{d}} \cdot \boldsymbol{x}_{\theta_b}.$$

5.3 Interest reach modeling

It is common to target users via their inferred or declared interests. For example a climbing gear retailer may be interested in reaching people who are classified as “Outdoor Enthusiasts”, while an organizer of a music band tour may want to target “Music Lovers”.

Even a small number of interest groups gives rise to large number of combinations, therefore modeling interests with categories of all interest combinations requires an impractically high number of Dirac deltas.

To deal with this dimensionality explosion we can find a collection of people categories corresponding to certain fixed distributions over interests. The sizes of categories and corresponding distributions over interests are to be fit to match measured marginal precision of the interest labels and total populations of each interest.

Similarly to the previous subsection we search for a direct assignment of cookies to people categories. But this time each category will have a distribution over measurable features (interests) assigned to it.

If $\mathcal{A}$ is the set of all interests then each people category $\mathbb{a} \in \mathbb{A}$ would correspond to some distribution over interests $\omega : \mathbb{a} \mapsto \{P(a|\mathbb{a})\}_{a \in \mathcal{A}}$ that people in this category follow.

Recall that $2^{\mathcal{A}}$ denotes the set of all subsets of $\mathcal{A}$. Let $L : C \to 2^{\mathcal{A}}$ be the correspondence between user identifiers and inferred interests.

Let ϵ_a be measured precision of the labels of interest a and $|a|$ be the fraction of population known to have this interest. For a given set of people categories $\mathbb{A}$ and an assignment of cookies to categories $\tilde{L} : C \to \mathbb{A}$ we can observe the modeled precision as

$$\hat{\epsilon}(a) = \frac{\sum_{c \in C, a \in L(c)} P(a|\omega(\tilde{L}(c)))}{|\{c \in C | a \in L(c)\}|}$$

and total audience as

$$\widehat{|a|} = \sum_{\mathbb{a}\in\mathbb{A}, a\in\mathbb{a}} |\mathbb{a}| \cdot P(a|\omega(\mathbb{a}))$$

In practice the modeling can be simplified by also assuming that cookie generation patterns are the same across categories of people. This is, for instance, required to model the interaction of interests with demo and geo via the independence assumption. We can express this constraint as

$$\forall \mathbb{a} \in \mathbb{A}\ |\mathbb{a}| \approx \frac{|\{c \in C|\mathbb{a} = \tilde{L}(c)|}{|C|} \tag{8}$$

Of course, the set of categories has to be of bounded size, otherwise we may end up with each category having a single person and will have no room to model any other feature. We can determine some threshold K and search for the set of categories that does not exceed K.

We can use a neural network to find a set of people categories $\mathbb{A}$, assignment $\tilde{L}$ and distribution map ω that minimizes errors

$$\sum_{a\in\mathcal{A}} \left\|\widehat{|a|} - |a|\right\|_2 \text{ and } \sum_{a\in\mathcal{A}} \|\hat{\epsilon}_a - \epsilon_a\|_2$$

under the constraints (8) and $|\mathbb{A}| \leq K$.

Relative weights of the errors and strictness of the constraints is part of the art of model training and may highly depend on specifics of the training data.

6 Simulations

In this section we present results of simulations verifying that the virtual person assignment matches aggregate level models. In the first subsection we look at a simple Exponential Bow reach curve, in the second subsection we look at model that calculates reach into two abstract categories.

6.1 Reach curve approximation

We ran an optimization algorithm that searched for an optimal Dirac Mixture consisting of 4 deltas that approximates the Exponential Bow model. The resulting mixture is presented in the Table 1.

α	$\boldsymbol{x}$
0.164086	0.0650141
0.388211	0.427442
0.312285	1.27499229
0.135418	3.14015914

Table 1: 4-delta Dirac Mixture parameters approximating Exponential Bow reach curve.

This dirac mixture defines a curve that approximates Exponential Bow with maximal relative error of 0.14%.

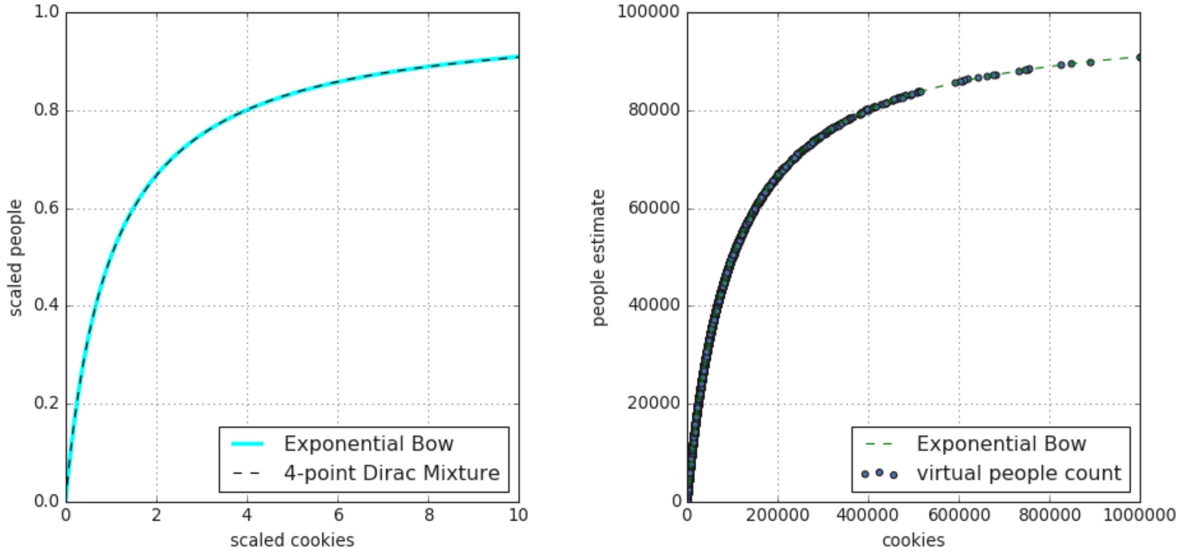

Figure 1: *Left*: Exponential Bow and 4-point Dirac Mixture approximation. *Right*: Exponential Bow estimates and counts of virtual people for simulated campaigns.

In Simulation 1 we generate random campaign cookies/people reach data that follows a Exponential Bow of $\kappa = 1$ and limit 100000. We sample cookie counts from Pareto type II distribution with expectation of 40000 and shape parameter equal to 3.0.

Recall that PDF of Pareto type II distribution has the form

$$\frac{\alpha}{\lambda}\left(1 + \frac{x}{\lambda}\right)^{-(\alpha+1)}$$

and in general many classes of online audiences follow a similar distribution.

```
for $i \in \{1 \dots 10000\}$ do
    sample number of cookies visiting $i$-th campaign $s_i$ from Pareto type II distribution
      with scale 3 and expectation 40000 ;
    let $r_i = \frac{100000 \cdot s_i}{100000 + s_i}$ ;
end
```

Simulation 1: Synthesising per campaign cookie reach s_i and aggregate estimates of people reach r_i.

The results of applying Algorithm 1 to the data are given at Figure 1. The absolute relative error for measurement $\hat{r}$ of quantity r is defined as $\epsilon = |\frac{\hat{r}}{r} - 1|$. Average absolute relative error of virtual people estimates in this simulation is 0.3%.

6.2 Per category reach

We implemented an aggregate model that estimates reach into two abstract categories of people A and B. The estimate is done based on labels. Each event has a label a, b, the label distribution is mapped to people category distribution via a stochastic matrix $r = \begin{bmatrix} 0.9 & 0.2 \\ 0.1 & 0.8 \end{bmatrix}$.

Similarily to the previous section we assume that campaign cookie reach is following a Pareto Type II distribution. We assume that logarithms of fractions of labels in campaigns for each category is normally distributed with $\mu = 0, \sigma = 1$.

Each of the categories A and B is assumed to have 50 thousand people and the cookie to people curve in each of the categories is assumed to be Exponential Bow .

```
for $i \in \{1 \ldots 10000\}$ do
    sample number of cookies visiting $i$-th campaign $s_i$ from Pareto type II distribution
      with scale 3 and expectation 40000 ;
    sample $z_i$ from Normal distribution with mean 0 and variance 1;
    let $f_i = \frac{e^{z}_{i}}{1+e^{z_i}}$;
    let $s^a_i = \lfloor f_i \cdot s_i \rfloor$;
    let $s^b_i = s_i - s^a_i$ ;
    let $\begin{bmatrix} s^A_i \\ s^B_i \end{bmatrix} = \begin{bmatrix} 0.9 & 0.2 \\ 0.1 & 0.8 \end{bmatrix} \cdot \begin{bmatrix} s^a_i \\ s^b_i \end{bmatrix}$;
    let $r^A_i = \frac{50000 \cdot s^A_i}{50000 + s^A_i}$ ;
    let $r^B_i = \frac{50000 \cdot s^B_i}{50000 + s^B_i}$ ;
end
```

Simulation 2: Synthesising per campaign per label cookie reach s^a_i, s^b_i and aggregate estimates of per category people reach r^A_i.

The code for generating data for this simulation is given in Simulation 2. Algorithm 2 is used for the virtual people assignment.

Figure 2 shows the comparison between aggregate estimates and estimates based on virtual people counts. The estimates for reach into category A, reach into category B and fraction of reach belonging to category A are compared. In pratice A and B could be men and women, or "Movie Lovers" and "not Movie Lovers", etc. Only campaigns with reach of more than 1000 people total are used for evaluating fraction of reach into category A.

Average absolute errors for compared quantities are given at Table 2.

quantity	error
reach into A	0.0061
reach into B	0.0091
fraction of A	0.0061

Table 2: Average relative absolute errors for the simulation of per-category reach modeling with virtual people.

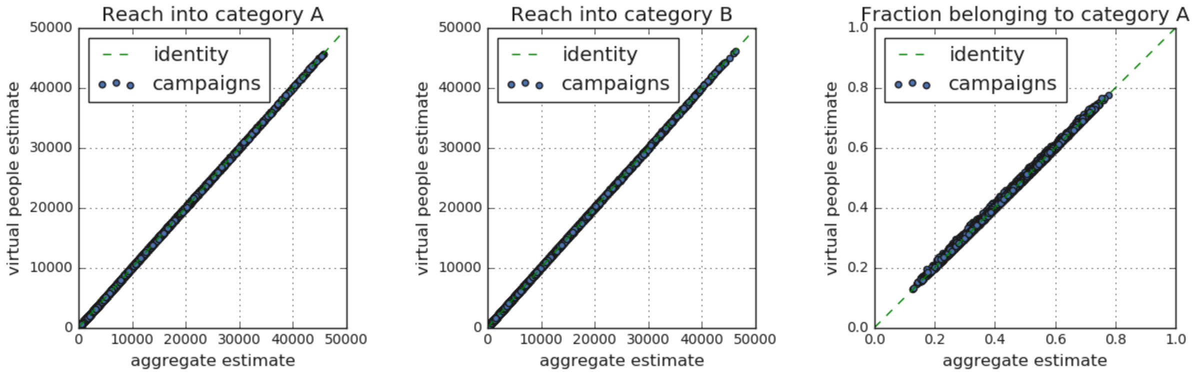

Figure 2: Exponential Bow estimates and counts of virtual people for simulated campaigns.

References

[1] Koehler, J, E Skvortsov, and W Vos. A method for measuring online audiences. Technical report, Google Inc.
http://static.googleusercontent.com/media/research.google.com/en/us/pubs/archive/41089.pdf

[2] Koehler, J, E Skvortsov, S Ma and S Liu. Measuring Cross-Device Online Audiences. Technical report, Google Inc.
http://static.googleusercontent.com/media/research.google.com/en/us/pubs/archive/45353.pdf

[3] Dirac, P. (1958), The Principles of Quantum Mechanics (4th ed.), Oxford at the Clarendon Press, ISBN 978-0-19-852011-5.

[4] J. N. Fry, S. Colombi, Pablo Fosalba, Anand Balaraman, István Szapudi, R. Teyssier; Cell count moments in the halo model, Monthly Notices of the Royal Astronomical Society, Volume 415, Issue 1, 21 July 2011, Pages 153–167

[5] Evin, Guillaume, Favre, Anne-Catherine. (2008). A new rainfall model based on the Neyman-Scott process using cubic copulas. Water Resources Research - WATER RESOUR RES. 44. 10.1029/2007WR006054.

7 APPENDIX

7.1 Proof of consistency theorem

Theorem 1. *Let $\mathcal{E}$ be a set of events and a function $F : \mathcal{E} \to \mathbb{Z}$ from the sets of events to integers satisfy the following properties for any sets of events A, B, C:*

1. Monotonicity*: $F(A \cup B) \geq F(A)$.*
2. Convexity*: $F(A \cup B) \leq F(A) + F(B)$.*
3. Attribution of incremental reach*: If $F(A \cup B \cup C) > F(A)$ then $F(A \cup B) > F(A)$ or $F(A \cup C) > F(A)$.*

4. Disjoint sets property*: If $F(A \cup B) = F(A) + F(B)$ and $F(A \cup C) = F(A) + F(C)$ and $F(B \cup C) = F(B) + F(C)$ then $F(A \cup B \cup C) = F(A) + F(B) + F(C)$.*

Let also for any individual impression i we have $F(\{i\}) \leq 1$.

Then there exists a set $\mathcal{V}$ (which we call a set of virtual people*) and a mapping $V : \mathcal{E} \to \mathcal{V}$, such that for any set of events A we have $F(A) = |V(A)|$. Recall that by definition $V(A) = \{V(i)|i \in A\}$. In other words $F(A)$ is equal to the count of virtual people corresponding to elements of A.*

Proof. First note that any impression with reach 0 can not impact reach of any set due to *convexity*. So we drop all such events from consideration and assume without loss of generality that for any impression i we have $F(\{i\}) = 1$.

We introduce the relation over events $i \sim j <=> F(\{i, j\}) = 1$. This relationship is equivalence, i.e. it is reflexive, symmetric and transitive. Reflexivity and symmetry are trivial.

To show transitivity consider events i, j, k and assume that $i \sim j$ and $j \sim k$. Setting $A = \{j\}, B = \{i\}, C = \{k\}$, then $F(A \cup B) = 1$ and $F(A \cup C) = 1$, and by contraposition of *attribution of incremental reach* property we arrive to $F(A \cup B \cup C) = 1$. Which by *monotonicity* leads to $F(B \cup C) = 1$ and $i \sim k$.

Thus we have proven that $\sim$ is equivalence. Let $[i]$ be the class of equivalence of i with respect to relation $\sim$.

We define $\mathcal{V}$ to be the set of classes of equivalence of $\sim$ and define $V(i) = [i]$.

Now we prove by induction on the size of A that $F(A) = |\{V(i)|i \in A\}|$.

If $|A| = 1$ then $F(A) = 1$ by the condition of the theorem and $|\{V(i)|i \in A\}| = 1$ because it contains the only class of equivalence of the single element of A.

To show the step of induction we consider set M and assume that any set of events of size strictly less than $|M|$ satisfies the inductive assumption. Let i be an arbitrary element of M and $M = N \cup \{i\}$, where $|N| = |M| - 1$.

Consider two cases: 1. $[i] \in |\{V(j)|j \in N\}|$ and 2. $[i] \notin |\{V(j)|j \in N\}|$.

Case 1: If $[i] \in |\{V(j)|j \in N\}|$ then $|V(M)| = |V(N)|$ and we need to prove that F(M) = F(N).

By convexity and monotonicity properties we have $F(N) \leq F(M) \leq F(N) + 1$.

There exists $j \in N$ such that $i \sim j$ because $[i] \in |\{V(j)|j \in N\}|$.

By inductive assumption $F(N) = F(N \setminus [j]) + 1$. From convexity we get that $F(N \cup [j]) \leq F(N \setminus [j]) + 1$. Also by definition we have $M \subseteq N \cup [j]$ and thus $F(M) \leq F(N \cup [j])$. Therefore we have $F(M) \leq F(N \cup [j]) \leq F(N \setminus [j]) + 1 = F(N)$.

But due to monotonicity $F(N) \leq F(M)$ and therefore we conclude that $F(N) = F(M)$.

Case 2: If $[i] \notin \{V(j)|j \in N\}$ then $|V(M)| = |V(N)| + 1$ and we need to prove that $F(M) = F(N) + 1$.

Pick an arbitrary $j \in N$. To apply disjoint sets property consider $A = [j] \cap N, B = N \setminus [j], C = \{i\}$. By inductive assumption we have $F(A) = 1$, $F(B) = F(N) - 1$ and $F(C) = 1$.

Thus we have $F(A \cup B) = F(A) + F(B)$ and $F(B \cup C) = F(B) + F(C)$. Since $i \not\sim j$, then by definition of $\sim$, monotonicity and then convexity we have $2 \leq F(\{i, j\}) \leq F(A \cup C) \leq F(A) + F(C) = 2$, therefore $F(A \cup C) = F(A) + F(C)$.

Finally by disjoint sets property we conclude that

$$F(M) = F(A \cup B \cup C) = F(A) + F(B) + F(C) = 1 + F(N) - 1 + 1 = F(N) + 1,$$

which finishes the proof. □

Notation Glossary

C a function mapping an event to the user identifier (e.g. a cookie) corresponding to this event. 7

L a function mapping a user identifier to its label (e.g. stating that a cookie has declared age range of 25 to 34). 8

R a reach function. 4

T a function mapping a user identifier to its type, (e.g. desktop cookie to "Desktop"). 7

V a function mapping each event to a virtual person identifier. 6

Ω the set of all random distributions over the set of people categories. 10

α a coefficient of Dirac Delta in Dirac Mixture. 4

$\boldsymbol{t}$ a cookie reach vector. 4

$\boldsymbol{x}$ an activity vector. 4

δ a Dirac Delta function. 4

ℓ a class of cookies. 8

κ a partial derivative of a reach function at 0. 5

$\mathbb{C}$ the set of all categories of people. 8

$\mathbb{c}$ a category of people. 8

$\mathcal{A}$ Activity Density Function. 4

$\mathcal{C}$ the set of all user identifiers. 7

$\mathcal{D}$ Dirac Mixture. 4

$\mathcal{E}$ the set of all events. 7

$\mathcal{L}$ the set of user identifier labels. 8

$\mathcal{T}$ the set of types of user identifiers. 7

$\mathcal{V}$ a set of virtual people identifiers. 6

ω a function mapping a Dirac Delta to a distribution over people categories. 10

σ a function mapping a virtual person to a people category. 11