# Algebra of Ground Truth Inference for Web Unique Identifiers

Andrés Corrada-Emmanuel
BlueCava
131 Innovation Drive
Irvine, CA, United States
andres.corrada@bluecava.com

## ABSTRACT

The accuracy of a proprietary unique ID web service (BlueCava's BC ID) can be measured in the absence of knowledge of the true unique labels for web browsers. We do this by comparing its labels to those of the most common web unique id service - cookies. This is an example of ground truth inference, estimating a statistic of true labels without actually knowing the label of any single data point. In this case, we create a system of polynomial equations for the accuracy parameters we want to measure. An asymmetry in the errors made by our web service and those by cookies allows us to solve the problem exactly. We obtain point estimates for the accuracies of the BC ID and cookies. For example, we can measure the cookie churn rate. We compare the results obtained during an online advertising campaign, where ground truth is not available, to those obtained on a surrogate data set, for which ground truth is available.

## Categories and Subject Descriptors

10002951.10003260 [**Information Systems**]: World Wide Web—*online advertising, crowdsourcing*; 10002978.10003029 [**Security and Privacy**]: Human and societal aspects—*usability*

## General Terms

Measurement

## Keywords

Ground truth inference, unique ID services

## 1. INTRODUCTION

The term *ground truth inference* is used to denote algorithms that measure the performance of machine learning workers. Crowdsourcing sites like Amazon Turk have highlighted a problem we face with many algorithms also. How do we judge their performance when we do not know the correct job has been done? In other words, we do not know the ground truth for the decisions made by the workers or algorithms so how can we know they are performing adequately?

Ground truth inference algorithms have been developed for Crowdsourcing settings [5, 7, 6]. In this paper we consider the ground truth inference problem in the setting of unique ID services on the Web. Instead of trying to judge human workers, we want to judge algorithms. Specifically, we want to know the accuracies of our unique ID web service. The service produces a branded ID, the BlueCava ID. We refer to it in this paper as the BC ID. The central conclusion of this paper is that we can judge the quality of one worker, the BlueCava ID service, using an inferior worker - web browser cookies.

We use the phrase *unique ID service* rather than just plain ID service to make a subtle distinction - the purpose of the ID service is not to provide identity but to provide uniqueness for web devices. The goal of the service is to guarantee uniqueness of its IDs without knowing the identity of the entities it IDs. The central claim of this paper is that we can measure the performance of such a privacy-respecting service without knowing the true identity of the devices for any portion of our data.

The term *ground truth* in this paper refers to the true identity of the devices. We assume it exists but is unknown to us. A second party cookie could act as such ground truth. Our operational reality is that the true identity of the devices that appear in our data stream are unknown to us. And yet, even though we do not have those labels, we still would like to measure the average accuracy of BC IDs for any data set we extract from the service's data stream.

This hidden knowledge aspect of ground truth inference problems intersects nicely with our research into privacy preserving measurement algorithms. We do not know, nor do we want to know, the true identity of the devices we tag with our service. But, we still would like to know the average performance of our service. Ground truth inference algorithms satisfy this privacy concern. They are designed to measure statistics of ground truth when the ground truth itself is absent. This paper demonstrates one such algorithm - we measure the accuracy of our service without actually knowing the identity of the devices we ID.

Our data input is a time aligned record of our BC ID, and a 3rd party cookie for the web browsers seen during an ad-

| timestamp | cookie ID | BC ID |
|---|---|---|
| 10 | c1234 | b2323 |
| 12 | c4321 | b4532 |
| ⋮ | ⋮ | ⋮ |

**Table 1: Format for the ID data set.**

vertising campaign. Table 1 shows a fictional sample of this data. The unknown ground truth would be a fourth column on this table. The core idea of this paper is that by looking at events in the first three columns (timestamp, BC ID, cookie) we can calculate the average accuracies and error rates of the BC ID service. That is, we can estimate statistics of the unseen fourth column on Table 1.

We create a system of polynomial equations relating observables from the data (which can be measured without knowing ground truth) to the unknown ground truth statistics we are trying to estimate. The cookie churn, $p_{\text{cookie}}(\text{new} \mid \text{prev})$, is one such statistic. For a given data set it is defined as the number of times a previously seen device was given a new cookie. We shall refer to this and other in-sample statistics as just statistics in the rest of this paper.

We shall show how a small asymmetry between the errors our BC ID service makes versus those made by cookies allows us to solve the polynomial system if we assume that our ID service makes errors independently of those by a 3rd party cookie. The independence assumption is necessary and perhaps more justifiable in the context of crowdsourcing. We discuss this point further in the Conclusions section of the paper.

Nowadays, the abundance of lots of data is common to many sciences. However, most data collected is missing the final labels engineers care about. A web document does not carry its true relevance score for every user's query. A photograph does not contain the names of the people that are in it. Most of our data remains, and will always remain, largely devoid of true labels for the tasks carried out by our algorithms. That does not stop our companies from building web services to carry out the tasks, albeit noisily. But how do you measure the performance of such services when you don't have the true labels for your task?

Ground truth inference algorithms are meant to solve this measurement problem. This paper details one such measurement algorithm - we measure the statistical performance of our BC ID service without actually knowing how correct each individual decision was. This novel way to make a measurement on web data should be of broad interest in this workshop.

## 1.1 The basic idea behind ground truth inference

The use of ensemble systems for better decisions is well known to the academic and industrial community of machine learning researchers and developers. The idea of majority voting dates back to the work of Condorcet during the French Revolution. Less well know is the use of multiple decisions not for *decisioning* as is done in majority voting ensemble, but for *inference*. As far as we can tell, that subtle distinction was first discovered by Dawid and Skene in 1979([3]) with their likelihood maximization method that relied on the EM algorithm. One can see why the EM algorithm would be good choice for ground truth inference problems. The EM algorithm was invented to deal with data containing hidden labels. In its ground truth application, Dawid and Skene used it to maximize a likelihood function based on the unknown performance statistics of individual human medical judges. Crucially, they had to assume the judges were independent in their errors. Although the EM algorithm has been extended to correlated judges by Raykar et al. [5].

We can thus state the basic question for *ground truth inference* - how do we use the noisy decisions of an ensemble of detectors to measure their individual performance when we don't know the true prediction for any data point? This paper considers the case where we have two noisy unique ID services and no knowledge of the devices' true IDs. By comparing their decisions, we will be able to estimate their average performance and simultaneously estimate the true number of unique entities in the data stream. In the appendix we prove that any single service would not be enough for solving the measurement problem. This makes evident the ensemble nature of the algorithm for *inferring* statistical accuracies of the services. We don't combine them to decide better, we combine them to infer their statistical performance. Ensembles are being used for *inference*, not *decisions.*

## 1.2 Algebras for ground truth inference

Readers may be familiar with the topic of *ground truth inference* in the context of judging or ranking workers in Crowdsourcing contexts. Starting around 2010, a revival of the likelihood method proposed by Dawid and Skene has occurred. The citations yearly frequency of their 1979 paper in Google Scholar [4] tells the tale. Two-thirds of its 370 citations occurred after 2010. The core idea in their paper is that you can set-up a likelihood function for the performance statistics of judges based on just their agreements or disagreements on the labels for a data set. The EM algorithm is used to maximize the likelihood of their observed decisions. Since EM algorithms are not guaranteed to provide global maximas, this method could return incorrect estimates. Most recently, Zhang et. al. [6] have shown how to guarantee a provably optimal solution with this approach.

In contrast, this paper introduces to the academic literature an alternative method for setting up a ground truth inference algorithm. Moments of the data can be used to define a polynomial system of equations for the statistics of ground truth one is trying to estimate. This methodology has been previously used in the context of measuring the performance of labelers in a two label task (document relevancy) and a five label task (DNA sequencing).[1]

A polynomial system of equations has several advantages.

- It can be easily formulated and solved numerically as a convex minimization problem. In contrast, the log likelihood method is a non-convex optimization problem.

- It can be exactly solved, and has been exactly solved, given assumptions about how the algorithms or workers correlate in their decisions. The solutions are point estimates unlike the run estimates provided by the EM algorithm.

- The point estimates solution is formulated in terms of a descending chain of polynomials of the frequency moments for the detectors' decisions. These polynomials are essentially instantaneously solvable with today's technology. The 100M records data set in this paper is solved in about a minute.

- The algebraic solution of the polynomial system of equations makes clear the number of degenerate solutions possible. The problem solved in this paper results in a linear solution which in turn implies that there is only one explanation for the accuracies of the BC ID and cookies for the observed campaign.

Nonetheless, the algebraic method has one weakness as befits a method based on moments of the data - it requires a large number of decisions where the same workers can be compared to each other. In other words, the data has to be aligned for all judges, and no records with missing decisions are allowed. Luckily, this is the operational theater of the algorithm in our company. An advertising campaign typically involves millions of devices. The primary data set considered in this paper has about 6 million putative devices to which were served 107 million impressions. We conclude the problem of measuring a web ID service's accuracy is amenable to the algebraic method.

## 2. STATISTICS OF UNIQUE ID SERVICES

The development of an algebraic ground truth inference algorithm proceeds through four stages. This section details the first of these stages - defining the statistics of ground truth one wants to estimate. To define these statistics we must first define the task and the ground truth that would allow one to see how correctly the task was done.

### 2.1 The task of unique identification

We begin by defining the machine learning task we are considering - giving unique IDs to web browsers in an online advertising campaign. The task definition can be simplified by considering an event table for the perfect algorithm or worker as given in Definition 1

*Definition 1.* The task of unique web browser identification.

- Web browsers arrive sequentially into the unique ID service.

- A perfect unique ID service assigns a new ID to a browser never seen before.

- A perfect unique ID service assigns a previous browser the same ID it assigned in its previous appearance.

The practical problem we seek to solve in this paper is not how to make the unique identification service better, but how to measure its performance on any particular data set we care to focus on. No single algorithm or worker is going to be able to perform this task perfectly. And we want to be able to measure its errors even when we don't have access to the true labels.

The practical utility of a measurement algorithm should be immediately obvious. It would allow one, for example, to leverage the large amount of data that the service receives without having to incur the cost of obtaining the true labels for the data. This, in turn, could then allow one to select a best of breed algorithm.

### 2.2 The ground truth

We assume that if we wanted, we could implement a ground truth oracle for the data stream generated by our service . In this paper, we will use this unknown oracle to define the statistics we want to estimate. The oracle would consist of a mapping between the device and the IDs the service gives to each appearance of the device in the data stream. If such an oracle existed (equivalent to saying a ground truth exists), we would be able to count all decision events by the id service. For example, all the times it was right and wrong conditioned on the true ID labels.

While we don't require the actual existence of the ground truth, in this particular case it is easy to construct it. We could, for example, compare our data to the 2nd party ID of the browsers. Web companies like Google, Yahoo, and Facebook have the equivalent of the ground truth for the identity of the browsers of their users. But even if that was possible, such a ground truth would be applicable to only one subset of our data - that which overlaps with the data of the 2nd party. A ground truth inference algorithm allows you to make measurements on all subsets of your data you choose to look at.

To reiterate, the true identity of each device that appears in the data stream exists. We just don't know what it is. This assumption is to be compared with other circumstances where a label may be ambiguous and many human observers could disagree on its correct value. For example, the task could be transcribing noisy audio files and certain portions could contain words so badly jumbled that no two transcribers agree in their decisions.

### 2.3 Prevalence of stream events

Using the task definition, we can now enumerate the states of the data stream that will allow us to measure average performance. Out of many possible choices for partitioning the stream, we choose a two state partition for the state of each browser appearing at the ID service. We do this because we only care to estimate average performance for our service. We are not interested in how well our service works on any single browser. We are interested in average performance. We can accomplish this by defining just two states for a browser when it appears at our service: the *new device* and *previous device* states. For conciseness, we refer to them as the *new* and *previous* event as shown in Definition 2

*Definition 2.* States of a unique ID data stream event.

- The first appearance of a device is a *new* event.
- Every other appearance of a device is a *previous* event.

With this event typology, we can define statistics of performance such as the number of times that the service correctly assigns an ID to a previously seen device. If we had the true labels, we can define this count exactly in terms of individual ID event counts,

$$\#(\text{previous device assigned previous ID}) = \sum_{i=1}^{I} \#(\text{previous } d_i \text{ to previous } ID_i), \quad (1)$$

where $I$ denotes the number of distinct devices. Our goal is to estimate the left hand side without needing to estimate the right hand side terms.

Note that these two states are easily computed if we had the ground truth of web browser unique ids. If we knew which browsers were different, we would be able to tag the first appearance of each one as a 'new' event and also its subsequent appearances, if any, as 'previous' events. In addition, these states are not intrinsic properties of the devices. A device is in the *new* state only in its first appearance. In the *previous* state in all subsequent appearances. Nonetheless, these unknown true events are unambiguously measurable from a ground truth oracle.

The prevalence of these two events is the first two statistics of ground truth that we want to calculate in this paper. They are defined as a statistic of the unknown ground truth for the sample defined by the data set. By construction, our two state definition of the data stream events is disjoint. A device's appearance is either its first one or not. This leads to the following easy derivation of a polynomial relating these two statistics,

Lemma 1. *Let $p_{new}$ be the relative count of new events in a ID service data set,*

$$p_{new} = \frac{\#(\textit{new device events})}{\#(\textit{total events})}. \quad (2)$$

*Let $p_{prev}$ be the relative count of previous events in the same data set,*

$$p_{prev} = \frac{\#(\textit{previous device events})}{\#(\textit{total events})}. \quad (3)$$

*Then they are related by a linear equation*

$$p_{new} + p_{prev} = 1. \quad (4)$$

Proof. By definition, the data stream has been partitioned into a disjoint set of events. If we had the ground truth, we would be able to count the number of times a device first appeared and otherwise. The sum of these two events would be exactly equal to that of the total events in the data stream

$$\#(\text{total events}) = \#(\text{new events}) + \#(\text{previous events})$$

Dividing both sides of this equation by #(total events) leads to the lemma's conclusion. □

This derivation may seem trivial or elementary (and it is intended to be so!) but it is meant to illustrate an algebraic approach we can also follow for the other statistics of ground truth we want to measure - how accurate is the unique ID service?

## 2.4 Errors in unique identification

There are many ways that one can choose to understand the performance of a unique identifier algorithm. The most exhaustive one would be to specify the whole confusion matrix for all the $E$ distinct entities that are actually present in the sampled data stream. For example, we may want to know how often entity A is confused with Entity B or viceversa. This is impractical in our case. The confusion matrix for an L label detector is of size $L^2$. In a typical advertising campaign one sees millions of distinct browsers. Deducing the full confusion matrix for such a data set is impossible with just two unique id services.

The approach we take here is to consider aggregate errors similar to how speech recognition system descriptions talk about the average *word error rate* rather than detailing the confusion between any two words in a full 50K vocabulary, say, the words `the` and `tea`. This motivated our choice of just two state labels for the data stream as detailed in the previous section. Similar to this, we focus on dividing the decisions of the unique ID services into five possible decision events, two of which correspond to correct decisions by the unique IDer and three to errors. They align with the two state partition of the data stream defined in the previous section.

The first set of parameters correspond to the possible decisions made by the unique ID service when it encounters a new device. We detail them in Definition 3. Each of these statistics involves the decision made by the unique ID service and the true state of the data stream.

*Definition 3.* Decision events for new devices.

- A new device is correctly identified as new.
- A new device is incorrectly identified as any previous entity.

A similar set can be constructed for the decisions made by the service when it encounters a device it had seen previously. They are defined in Definition 4.

*Definition 4.* Decision events for previous devices.

- A previous device is correctly identified and given the same ID as in its previous appearance.
- A previous device is incorrectly issued a new ID.
- A previous device is incorrectly identified as any other previous device and given that incorrect ID.

These quantities, along with the prevalences of new and previous states are the only statistics of ground truth we solve for in this paper. They also satisfy simple normalization linear polynomial equations. We state them here as two unproven lemmas. It should be clear to the reader that these statistics have a single value under the assumption that a ground truth for browser identity exists. In addition, these statistics are in-sample values. The same service could have radically different counts for a different data set.

Lemma 2. *Let $p(new \mid new)$ be the relative count of new devices correctly identified as new,*

$$p(new \mid new) = \frac{\#(new\ ID,\ new\ device)}{\#(new\ device)}. \quad (5)$$

*Let $p(prev \mid new)$ be the relative count of new devices incorrectly assigned the ID of a previous device,*

$$p(prev \mid new) = \frac{\#(previous\ ID,\ new\ device)}{\#(new\ device)}. \quad (6)$$

*Then they are related by a linear equation*

$$p(new \mid new) + p(prev \mid new) = 1.$$

Lemma 3. *Let $p(new \mid new)$ be the relative count of previous devices correctly mapped to their previous ID,*

$$p(prev \mid prev) = \frac{\#(prev\ ID,\ prev\ device)}{\#(prev\ devices)}. \quad (7)$$

*Let $p(new \mid prev)$ be the relative count of previous devices incorrectly assigned a new ID,*

$$p(new \mid prev) = \frac{\#(new\ ID,\ prev\ device)}{\#(prev\ devices)}. \quad (8)$$

*Let $p(other\ prev \mid prev)$ be the relative count of previous devices incorrectly assigned the ID of some other previous device,*

$$p(other\ prev \mid prev) = \frac{\#(other\ prev\ ID,\ prev\ device)}{\#(prev\ devices)}. \quad (9)$$

*Then they are related by a linear equation*

$$p(prev \mid prev) + p(new \mid prev) p(other\ prev \mid prev) = 1.$$

## 3. ALGEBRA OF DECISION EVENTS

This section details the statistical identities that are the core construct in algebraic ground truth inference algorithms. The basic idea is that the decisions by a unique ID service can be partitioned into unknown events involving the actual decision and the unknown true label. An example demonstrates the principle.

Consider the case of the first arrival of entities into the unique ID service. By definition, these events correspond to entities that should be assigned a new ID by the service. Consider, then, all events where the ID service assigned a new ID. Given our error classification scheme, these events can be exactly partitioned into two event types. All events where the unique ID service assigns a new ID must be one of two disjoint stream event classes,

- The entity is actually new.
- The entity was previously seen but is mistaken as a new one.

This can be expressed as a mathematical identity for the counts

$$\#(\text{new ID events}) = \#(\text{new ID, new entity}) + \\ \#(\text{new ID, previous entity}). \quad (10)$$

Even though this equation is unsolvable as stated, it is worth pointing out the difference between its left and right hand sides. The quantity on the left, the number of times that the service assigned a new ID, is observable for any data set from the unique ID data stream. We can always tell when the service assigns a new ID even when we don't know whether it did it correctly or not. In contrast, the quantities on the right side of equation 10 are unknown to us. Indeed, the goal of this paper is to estimate those right side terms when we can only observe the left hand side. That is the primary goal of ground truth inference algorithms: to estimate statistics of the decisions and true labels of the data when we don't know the true labels of any single data point.

We can continue to manipulate equation 10 to obtain further identities. In particular, we can break up the joint decision event counts on its right hand side into conditional counts and true event label counts. The correct identification subevent can be rewritten identically as,

$$\#(\text{new ID, new entity}) == \\ \frac{\#(\text{new ID and new entity})}{\#(\text{new entities})} \#(\text{new entities}) \quad (11)$$

The incorrect identification sub-event can also be rewritten identically as,

$$\#(\text{new ID, previous entity}) == \\ \frac{\#(\text{new ID and previous entity})}{\#(\text{prev. entities})} \#(\text{prev. entities}) \quad (12)$$

We can combine all these identities into a single statistical identity for a single unique ID service

$$f_{\text{new IDs}} = p(\text{new} \mid \text{new})\, p(\text{new}) + \\ p(\text{new} \mid \text{previous})\, p(\text{previous}) \quad (13)$$

This identity is true for any data set we choose to sample from the data stream of the unique ID service. But it is unsolvable in the case of a single ID service (see Appendix B). Ground truth inference requires that we bring in other noisy unique ID services. We continue the paper by showing how we use the unique identifier workhorse of the Web, cookies, to accomplish this.

## 3.1 Cookies as a unique ID service

Web browser cookies are also a unique ID service. They are noisy, to be sure. But they can also be characterized by the same error typology we stated previously. The one thing that is different for cookies is that they have, assuming they are properly implemented, only one error mode. The only mistake made by a cookie is that it can assign a new ID to a previously seen device. This is commonly known as the problem of *cookie churn*, $p_{\text{cookie}}(\text{new} \mid \text{prev})$ is nonzero. This paper assumes that, for all practical purposes, the following two error modes for a cookie are zero,

$$p_{\text{cookie}}(\text{prev} \mid \text{new}) = 0 \tag{14}$$

$$p_{\text{cookie}}(\text{other prev} \mid \text{prev}) = 0 \tag{15}$$

## 3.2 Aligning unique ID decisions

Now that we have introduced a second service, we need to state explicitly how we compare the two ID labels. It should be clear to the reader that the arbitrary nature of these IDs means that they cannot be compared directly. Nonetheless, we must have a protocol for deciding when the two algorithms agree and disagree on the uniqueness of a device.

We do it by defining the data set we will use for measuring their performance. Our data set consists of a time ordered record of the ID given by each service. We will re-arrange this data set in different ways to create the set of polynomial equations that relate all the statistical parameters we want to measure. Each polynomial in that system defines a way to compare the decisions of the two ID services.

# 4. THE MEASUREMENT ALGORITHM

## 4.1 Exact definition of the problem

We can now state exactly our ground truth inference problem.

*Definition 5.* The ground truth inference problem for two ID services is to estimate twelve ground truth statistics (2 prevalences plus 5 performance statistics from each service) using only a time ordered data set of the ID each service assigns to each device's appearance. The two prevalences are

- $p_{\text{new}}$
- $p_{\text{previous}}$

and the five conditional performance statistics for each of the two services are

- Encountering a new device: $p(\text{new} \mid \text{new})$, $p(\text{other prev} \mid \text{new})$.
- Encountering a previous device: $p(\text{prev} \mid \text{prev})$, $p(\text{new} \mid \text{prev})$, $p(\text{other prev} \mid \text{prev})$.

It so happens the implementation of cookies makes it an ID service that does not make two of the three possible errors we defined. This makes it possible to solve this problem. Instead of twelve statistics, we only have to estimate ten. The definitions of the statistics gives us five normalization equations. If we can find another five independent polynomials, the complete problem would be solved. We begin by enumerating the five normalization equations.

## 4.2 Normalization equations

We already have five polynomial equations for our ten unknown statistics of the ground truth. They correspond to the fact that our definition of events is disjoint. They can be viewed as normalization equations. They are,

$$p_{\text{new}} + p_{\text{prev}} = 1 \tag{16}$$

$$p_{\text{cookie}}(\text{new} \mid \text{new}) = 1 \tag{17}$$

$$p_{\text{cookie}}(\text{prev} \mid \text{prev}) + p_{\text{cookie}}(\text{new} \mid \text{prev}) = 1 \tag{18}$$

$$p_{\text{BC}}(\text{new} \mid \text{new}) + p_{\text{BC}}(\text{prev} \mid \text{new}) = 1 \tag{19}$$

$$p_{\text{BC}}(\text{prev} \mid \text{prev}) + p_{\text{BC}}(\text{new} \mid \text{prev}) = 1 \tag{20}$$

If we can find more equations relating the unknown parameters to each other, we would be able to solve our problem. To define new polynomials we look at joint events in the decisions made by the two ID services. These are the measurement polynomials that may not be as obvious to the reader as the normalization ones here.

## 4.3 Joint ID decision polynomials

We can get three new polynomials by counting the number of times the two ID services agree or disagree on assigning a new ID. These correspond to three observables from our time ordered data set. They are

- #(BC new, cookie new): The number of times both services assigned a new ID.
- #(BC prev, cookie new): The number of times the cookie was new but the BC ID had been assigned before.
- #(BC new, cookie prev): The number of times the BC ID was new but the cookie had been seen before.

Each of these events leads to a new polynomial. We prove Lemma 4 in the Appendix. The rest are easy to prove in a similar manner.

LEMMA 4. *The relative frequency of events where both services give a new ID obeys the mathematical identity*

$$f_{BC\ new,\ cookie\ new} = p(BC\ new,\ cookie\ new \mid new)\, p_{new} + p(BC\ new,\ cookie\ new \mid prev)\, p_{prev} \tag{21}$$

*In the case that their errors are independent, this becomes a polynomial in the desired individual performance statistics,*

$$f_{BC\ new,\ cookie\ new} = p_{BC}(new \mid new)\, p_{cookie}(new \mid new)\, p_{new} + p_{BC}(new \mid prev)\, p_{cookie}(new \mid prev)\, p_{prev} \tag{22}$$

LEMMA 5. *The relative frequency of events where the BC ID assigns a new ID but the cookie was previously seen obeys* *the mathematical identity*

$$f_{BC\ new,\ cookie\ prev} = p(BC\ new,\ cookie\ prev \mid prev)\, p_{prev} \tag{23}$$

*In the case that their errors are independent, this becomes a polynomial in the desired individual performance statistics,*

$$f_{BC\ new,\ cookie\ prev} = p_{BC}(new \mid prev)\, p_{cookie}(prev \mid prev)\, p_{prev} \tag{24}$$

The fact that (24) is a monomial immediately implies that these are events we can unequivocally count as errors in the BC ID service.

Lemma 6. *The relative frequency of events where the BC ID is not new but the cookie is new obeys the mathematical identity*

$$\begin{aligned} f_{BC\ prev,\ cookie\ new} = {} & p(BC\ prev,\ cookie\ new \mid new)\, p_{new} + \\ & p(BC\ prev,\ cookie\ new \mid prev)\, p_{prev} \end{aligned} \tag{25}$$

*In the case that their errors are independent, this becomes a polynomial in the desired individual performance statistics,*

$$\begin{aligned} & f_{BC\ new,\ cookie\ new} = p_{BC}(new \mid new)\, p_{cookie}(new \mid new)\, p_{new} + \\ & (p_{BC}(new \mid prev)\ + p_{BC}(new \mid prev))\ p_{cookie}(new \mid prev)\, p_{prev} \end{aligned} \tag{26}$$

Note that these polynomials are cubic in the unknown statistics. This came about because we are assuming the two ID services make independent errors. Our final system of polynomials will thus consist of a mixture of linear and cubic equations. It will become necessary to use computer algebra systems to solve it.

Why don't we write a fourth polynomial for the events where both IDs assign a previous ID? We could. But our formulation in terms of event frequencies makes this fourth polynomial redundant. The sum of the four possible joint events must sum to the total number of events.

## 4.4 ID trail polynomials

Our final set of polynomials is going to come from comparing decisions when we align the data set into what we call *ID trails.*

*Definition 6.* ID trails

- An ID trails data set consists of one record per unique ID seen by a service.
- The single ID record consists of the time-ordered sequence of the IDs of the second service.

So, for example, we could align the data stream using the BC ID. A BC ID trail record would then consist of the cookies assigned to the same events where the trail's ID was also assigned. Since we are considering two services in this paper, we can do this in two different ways. We can re-arrange the data set into ID trails using the BC ID or we can create the trails using the cookie ID.

Assume that you have re-arranged the data stream into cookie trails. Each cookie trail now has a first record. We define this first record to have agreement between the two ID services. This an arbitrary but perfectly possible definition since what we care about is the errors on consecutive appearances of devices. Now look at the second event in a cookie trail if it exists. If the two IDs agree, we count it. We continue for all events in that cookie's trail and over all the cookies. The importance of the cookie trails is that they show sucessive events where the cookie is correct since cookies never mix trails. The following lemma illustrates that point by having a single term.

Lemma 7. *The relative frequency of events where the BC ID and cookie agree on successive impression events in the cookie trail data set obeys the mathematical identity*

$$f_{need} = p(BC\ prev,\ cookie\ prev \mid prev)\, p_{prev} \tag{27}$$

*In the case where their errors are independent, this becomes a polynomial in the desired individual performance statistics,*

$$f_{need} = p_{BC}(prev \mid prev)\, p_{cookie}(prev \mid prev)\, p_{prev} \tag{28}$$

This polynomial (Eq. 28) should be compared to the polynomial 26. It is here that the asymmetry between the errors our unique ID service makes and those of cookies reveals itself. Cookie trails will never be wrong by our assumptions. Thus, if our unique ID changes during a cookie trail we know unequivocally that we have made a mistake. The appearance of the term $p_{\mathrm{BC}}(\text{other prev} \mid \text{prev})$ in Equation 26 but not in Equation 28 means that we can unambigiously measure each separately. To speak geometrically, the system of polynomials can find point estimates rather than a single curve relating $p_{\mathrm{BC}}(\text{prev} \mid \text{prev})$ and $p_{\mathrm{BC}}(\text{other prev} \mid \text{prev})$.

For our final polynomial equation, we create the BC ID trails data set. How many records do we get in this re-arrangement of the data stream? We should get as many records as the number of times the BC ID service was correct plus the number of times it was wrong. We state the lemma for the polynomial without proof.

Lemma 8. *The ratio of BC IDs to impressions served during the campaign is equal to*

$$\begin{aligned} f_{new\ BC\ ID\ impressions} = {} & p_{BC}(new \mid new)\, p_{new} + \\ & p_{BC}(new \mid prev)\, p_{prev} \end{aligned} \tag{29}$$

# 5. A SOLUTION FOR AN ADVERTISING CAMPAIGN

We now apply the formalism developed in the previous sections to an actual advertising campaign. The data set consists of $107,187,864$ impressions served during a one month period in 2013. As required, we can create the requisite time ordered data set from the timestamps of the impressions. For this campaign's data set, there are $6,142,935$ distinct BC IDs but $14,525,610$ distinct cookies. The cookie came

**Table 2: Joint decision counts**

| joint decision event | observed count |
|---|---|
| BC ID new, cookie new | $5,484,113$ |
| BC ID prev, cookie new | $9,041,497$ |
| BC ID new, cookie prev | $658,822$ |

**Table 3: ID trails counts**

| id trail event | observed count |
|---|---|
| Cookie impressions with same BC IDs | $91,748,960$ |
| unique BC IDs | $6,142,935$ |

from the advertiser itself and is thus implemented and managed separately from the BC ID.

The observed counts for our decision events polynomials were tallied. We present them in the two tables below. The joint decision counts are presented in Table 2. The counts for statistics of the ID trails are shown in Table 3.

We used the *Mathematica* software system to carry out the solution of the resulting system of 10 polynomial equations. Five linear equations come from normalization constraints on our unknown statistics. These are given by Equations (16) to (20). The five other equations come from the observable decision events: (22), (24), (26), (28), and (29).

The `GroebnerBasis` function in *Mathematica* implements Buchberger's algorithm. This algorithm can be coaxed into yielding what is called an elimination system of polynomials. For the problem we defined in this paper, that elimination system starts with a linear equation for the unknown prevalence of new devices into the service. Subsequent polynomial equations in the elimination system are also linear. This means there is only one possible solution to the independent ID model we are assuming in this paper. The numerical values so obtained are tabulated in Table 4.

Systems like Mathematica conveniently offer functions that implement algorithms from algebraic geometry. Readers interested in the mathematics behind solving a system of polynomial equations should consult the excellent introduction by Cox et al. [2]. In the lingo of algebraic geometry, our point estimate is a *variety.* Less equations would result in lines or other geometric objects. The results in this paper demonstrate that single point solutions are possible in the unique ID task.

## 5.1 A surrogate confirmation

At this time, we do not have a data set that can separately confirm the applicability of the independence assumption to the observed data. If we did, we would not have developed the algorithm presented in this paper. But we can carry out an independent measurement on part of our data. This section describes that measurement, why it can only partially measure the parameters we care to know, and how it agrees with the ground truth inference algorithm of this paper.

The BC ID service also uses a cookie associated with the `bluecava.com` domain. We hasten to add that this cookie cannot be used with our method since it patently violates our core assumption - the ID services are independent in their errors. The use of the BC cookie is partially for convenience reasons. If a device arrives at our service with a previously seen BC cookie, we can quickly find its previous ID and move on. We expend computational power only on devices for which we do not have a previous BC ID cookie.

**Table 4: Ground truth inference estimates**

| parameter | estimated value |
|---|---|
| $p_{\text{prev}}$ | 0.949076 |
| $p_{\text{new}}$ | 0.050924 |
| $p_{\text{cookie}}(\text{new} \mid \text{new})$ | 1 |
| $p_{\text{cookie}}(\text{prev} \mid \text{prev})$ | 0.91087 |
| $p_{\text{cookie}}(\text{new} \mid \text{prev})$ | 0.0891301 |
| $p_{\text{BC}}(\text{new} \mid \text{new})$ | 0.99289 |
| $p_{\text{BC}}(\text{prev} \mid \text{new})$ | 0.00711 |
| $p_{\text{BC}}(\text{prev} \mid \text{prev})$ | 0.990144 |
| $p_{\text{BC}}(\text{new} \mid \text{prev})$ | 0.00710993 |
| $p_{\text{BC}}(\text{other prev} \mid \text{prev})$ | 0.00274623 |

**Table 5: Surrogate comparison**

| parameter | campaign estimate | surrogate ratio |
|---|---|---|
| $p_{\text{BC}}(\text{prev} \mid \text{prev})$ | 0.990144 | 0.982 |
| $p_{\text{BC}}(\text{new} \mid \text{prev})$ | 0.00710993 | 0.016 |
| $p_{\text{BC}}(\text{other prev} \mid \text{prev})$ | 0.00274623 | 0.00198 |

The devices for which we see the same BC cookie on successive appearances can be used to measure the performance of our system on previous devices. That is, we can pretend we don't know it has the same cookie as before and ask our service to give it its correct ID. Most of the time it can give a correct answer, sometimes it cannot. In essence, we are able to measure all the performance parameters for the condition when the device has been seen previously by our system. The measurement is not exactly what we want even for this subset since it is performed on the devices that have the same BC ID cookie. This measurement must be viewed as a *surrogate* measurement. It is the same statistic being measured but it is on a different data set.

We offer the comparison between this surrogate set and the values in Table 4 in Table 5. We freely acknowledge that the comparison is only suggestive that our independence assumption holds for the practical data set shown here. The surrogate data set consisted of 94.4 million records from our daily log. The performance on the surrogate data is slightly worse but in the same range as the one deduced with the polynomials. Tellingly, the rate of mistaken devices for each other is small. This agrees with the design of the service - our customers prefer us making the $p(\text{new} \mid \text{prev})$ errors rather than the $p(\text{other prev} \mid \text{prev})$ errors.

# 6. CONCLUSIONS

An algebraic method for inferring the statistical performance of two web unique ID services was presented. The polynomial system of equations becomes solvable for two services when we assume they make independent errors. Empirical results on an actual advertising campaign are suggestive that it is able to measure the performance without knowledge of the true labels for the data set.

How appropriate is our assumption of independence of errors between the two services? We cannot answer that conclusively at this time. Note, however, that since cookies make only one of the three errors the BC ID service makes, two of their errors are already uncorrelated by definition. If our BC ID is correlated with other cookies it would have to be only in the $p(\text{new} \mid \text{prev})$ error.

Do we have to assume independence of errors? Suppose, for example, that you wanted to measure the correlation between two algorithms without having to assume they are independent. This is possible if you bring in more algorithms to compare against each other. Just as one service cannot measure itself, two services cannot measure their correlation.

A possible line of future research is to determine how many services would be needed to fit a pair correlated decision model. It is clear from the mathematics done here that including new terms to model the pair correlation would lead to higher order polynomials than the cubic ones that result from the independence model. This, in turn, could make the problem unsolvable in practice.

Solving the correlated pair model would make the independence assumption more plausible. The claim that a linear model is the right model for data is more believable when a quadratic model fit shows a small value for the square term. Likewise, a solution with a correlated pair model that shows a small value for the pair correlation parameters would give us more confidence the services are statistically independent. As we stated previously, we presently do not know how many services would be required to measure the pair correlation between any two of them.

# APPENDIX

## A. PROOF OF THE NEW-NEW ID DECISION EVENT POLYNOMIAL

In this section we prove how we construct a cubic polynomial for the counts of data stream events where both ID services assign a new ID as stated in Lemma 4.

Proof. Our proof proceeds in a manner similar to the one done for Lemma 1. We avail ourselves of the disjoint nature of true labels and label decision events. The added complication is that we must prove that the observable event can be tallied for our data set when we don't have the true label. We begin with proving that the observable statistics of the data are completely defined in this case.

Our observable event is the number of device arrival events where both ID services assigned a new ID. Since our data set preserves the time ordering of device arrivals, the question - "Is this the first time you assign this ID in this data set?" - is well-defined and has an unambiguous answer for either service. The quantity $f_{\text{BC ID new, cookie new}}$ is thus equal to the ratio of two counts we can deduce from the data set,

$$f_{\text{BC ID, cookie new}} = \\ \#(\text{BC ID, cookie new})/\#(\text{all arrival events}) \quad (30)$$

For the campaign data set discussed in the paper, this ratio happens to have the value

$$f_{\text{BC ID, cookie new}} = \frac{5484113}{107187864} \quad (31)$$

The count of new IDs for both services can also be decomposed into two unknown counts, the times the new IDs were given for truly new devices and the times they were given to previously appearing devices.

$$\#(\text{BC ID, cookie new}) = \#(\text{BC ID, cookie new, dev. new}) + \\ \#(\text{BC ID, cookie new, dev. previous}) \quad (32)$$

As in Lemma 1, we can transform each of the terms on the right hand side using mathematical identities. We write this out explicitly for the first right hand term. The mathematical identity is

$$1 = \#(\text{new devices})/\#(\text{new devices}). \quad (33)$$

Applied to the first term of equation 32 we get,

$$\#(\text{both IDs new, dev. new}) = \frac{\#(\text{IDs new, new})}{\#(\text{new})}\#(\text{new}) \quad (34)$$

$$= p(\text{IDs new} \mid \text{new})\#(\text{new}) \quad (35)$$

So equation 32 is identical to

$$\#(\text{BC ID, cookie new}) = p(\text{IDs new} \mid \text{new})\#(\text{new}) + \\ p(\text{IDs new} \mid \text{prev})\#(\text{prev}) \quad (36)$$

Dividing both sides of this equation by #(total events) leads to the first claim of the lemma

$$f_{\text{BC ID, cookie new}} = p(\text{IDs new} \mid \text{new})\, p_{\text{new}} + \\ p(\text{IDs new} \mid \text{prev})\, p_{\text{prev}}. \quad (37)$$

This statistical identity is exact. But it is unsolvable in the context of just two ID services. To set-up a set of polynomials that can be solved with just two ID services, we need to assume statistical independence for both conditional terms on the right hand side of equation 37. This leads to the final claim of the lemma,

$$f_{\text{BC ID, cookie new}} = p_{\text{BC}}(\text{new} \mid \text{new})\, p_{\text{cookie}}(\text{new} \mid \text{new})\, p_{\text{new}} + \\ p_{\text{BC}}(\text{new} \mid \text{prev})\, p_{\text{cookie}}(\text{new} \mid \text{prev})\, p_{\text{prev}}. \quad (38)$$

□

## B. A COOKIE CANNOT MEASURE ITS OWN PERFORMANCE

This appendix contains a proof to the claim that ground truth inference requires at least two ID services. Suppose the contrary and attempt to build a polynomial system for the cookie service. Four statistics of ground truth are unknown to us. The two prevalences, $p_{\text{new}}$ and $p_{\text{prev}}$, and the two cookie performance statistics, $p_{\text{cookie}}(\text{prev} \mid \text{prev})$ and $p_{\text{cookie}}(\text{new} \mid \text{prev})$. We get two equations for the normalization constraints on the statistics. Since only one service is being used, there are only two possible events we can deduce from the data stream. This only gives one polynomial based on the performance of the cookie. Three equations cannot give a point solution for four variables.

Contrast this with the case of the two services. There we had 3 joint ID decision events. The crucial point is that for N services the number of decisions events grows as $2^N$. But the number of parameters that we need to deduce grows linearly with the services if we assume they are independent.

## C. REFERENCES

[1] A. Corrada-Emmanuel. Method for inferring attributes of a data set and recognizers used thereon, Dec. 1 2010. WO Patent App. PCT/US2011/062,772.

[2] D. A. Cox, J. Little, and D. O'Shea. *Ideals, Varieties, and Algorithms*. Springer-Verlag, New York, 3rd edition, 2007.

[3] A. P. Dawid and A. M. Skene. Maximum likelihood estimation of observer error rates using the em algorithm. *Journal of the Royal Statistical Society, Series C*, 28(1):20–28, November 1979.

[4] A. P. Dawid and A. M. skene. Google scholar page for dawid and skene's 1979 paper, Nov. 2014.

[5] V. C. Raykar, S. Yu, L. H. Zhao, G. H. Valadez, C. Florin, L. Bogoni, and L. Moy. Learning from crowds. *The Journal of Machine Learning Research*, 11(3):1297–1322, March 2010.

[6] Y. Zhang, X. Chen, D. Zhou, and M. I. Jordan. Spectral methods meet em: A provably optimal algorithm for crowdsourcing. In *Advances in Neural Information Processing Systems*, December 2014.

[7] D. Zhou, J. Platt, S. Basu, and Y. Mao. Learning from the wisdom of crowds by minimax entropy. In *Advances in Neural Information Processing Systems 25*, pages 2204–2212, 2012.