Demographic Prediction of Web Requests from Labeled Aggregate Data

Brendan Kitts
PrecisionDemand
Seattle WA 98116 USA
bkitts@gmail.com

Garrett Badeau, Andrew Potter, Liang Wei, Sergey Tolkachov, Ethan Thornburg
Adap.tv
2 Waters Park Drive
San Mateo CA 94403 USA

Davood Shamsi
AOL Platforms
395 Page Mill Road
Palo Alto CA 94306 USA

ABSTRACT
Demographics are the currency of online advertising. But how does an advertiser know that they received the demographics that they requested? In online advertising it is routine for demographics to be audited by a "trusted" third party source such as Nielsen or Comscore. This third party is able to review a sample of online traffic, and then send back data on what percentage of that traffic has the demographic trait that the advertiser is targeting. Trusted Demographic Auditors can be thought of as "Oracles". Oracle data is typically only available on aggregated batches of requests. How could this data be used, then, to predict the demographics of individual requests? The paper discusses methods for predicting demographics from aggregated data. In particular we show results from sevcral algorithms on real ad server data.

Categories and Subject Descriptors
D.4.8 [**Performance (C.4, D.2.8, I.6)**]: Modeling and Prediction

General Terms
Measurement

Keywords
Advertising, targeting, demographics, prediction

1. INTRODUCTION
In order to reach customers, advertisers need a *lingua franca* to find their customers across multiple mediums. The currency would ideally be universal in that a descriptor's meaning is the same whether in television, websites, radio, billboards and other mediums. The currency should also enable advertisers to target their customers without being perceived as being overly intrusive.

Demographics have traditionally met all of the above criteria. For this reason most digital advertising products including Facebook, AOL, Google Adwords, Microsoft Bing, Yahoo and others, allow advertisers to target based on demographics.

The widespread use of demographics for targeting and billing introduces two problems for advertisers. Firstly, where should one advertise to reach an intended demographic? This could be termed the "Demographic Prediction Problem".

Secondly how does the advertiser know if they indeed reached that demographic? This could be termed the "Verification or Ground Truth Reporting Problem".

In order to solve both problems, measurement companies often maintain their own paid panels of individuals who allow their personal information to be reported anonymously. For example, Comscore's Validated Campaign Essentials (VCE) product comprises a 1 million sized panel of persons who allow their online behavior to be tracked and reported (Reagan, 2013). Nielsen Corporation maintains a similar panel with 200,000 US Panelists, and offers reporting through their Digital Ad Ratings product (Nielsen, 2015).

Because these companies are not selling media, but instead sell "ratings measurement" services – to both media sellers and buyers - they have some degree of independence and interest in providing the most accurate numbers possible. They also have a direct first party relationship with their panelists, who agree for their data to be used anonymously in aggregate. Many advertisers use these measurement companies as their ground truth for measuring campaign outcomes as well as billing. In the case of Adap.tv, approximately 25% of advertisers who are targeting demographics use third party measurement for verification purposes.

1.1 Mechanics of Demographic Assignment
Modern ad servers work by calling these data providers with the traffic they want audited. However rather than sending back demographics on the ad request right away, the data provider batches up the requests they have received, and then after reaching a sufficient threshold, for example, a batch of 10,000 requests, it can then be interrogated to reports back on the demographic percentages in the 10,000 batch of requests. This ensures panelist privacy and also is a security feature to ensure that the requesting party can't just develop a cache of the demographics for each user, and thereby replicate the panel that the data company is paying its panelists to maintain.

The requestor also sends a convenient label for the traffic to the data company – which is still batched to 10,000 – so that they can execute particular audits. For example, the requestor can audit a particular site (eg. cnn.com), ad campaign, user segment, or even time of day. In all cases, a batch of 10,000 requests is created.

The hard part is next: Each ad request is a billable event, and the ad server needs to ensure that the advertiser looking for the demographics get the opportunity to show their ad. Thus the ad

TargetAd 2016: 2nd International Workshop on Ad Targeting at Scale, WSDM, February 22--25, 2016, San Francisco, CA, USA.

server needs to predict the demographics of every individual *request*. However the measurement company reports aggregated demographics data in batches of 10,000. How can the ad server learn to predict the demographics of a request based on training data which has been purposely aggregated in order to avoid revealing the demographics of individual ad requests?

This will be the subject of this paper. We propose that predicting using aggregated information is a general class of problem with interesting properties that are worthy of study. Solving this problem is also of great economic importance, since advertiser ROI is directly proportional to the quality of demographic targeting.

2. PREVIOUS WORK

2.1 Demographic Prediction where individual case labels are available

Demographic prediction is a staple of web traffic analysis, and much previous work has been published on predicting demographics - *when individual labels are present*. This is a "standard" machine learning problem where a subset of users have a known class label, and others need to be inferred.

Some representative work includes Hua et. al. (2007), who used Windows live login data to build a set of users with known age, gender, and then try to predict probability of age-gender using keywords, webpages. Bi et. al. (2010) used a similar approach with Bing search queries. Weber and Castillo (2010) also predicted demographics using Yahoo! Registered users who have self-reported their demographics. Ulges et. al. (2012) used Youtube registration data to predict demographics based on video viewership. In some other domains there is a similar requirement to measure a variable based on samples. For example Geologists need to estimate the expected ore concentration from different geological deposits, based on sparse exploratory drill holes. (Everett, 2013).

Our work differs from previous work because our demographic data is on *an aggregated batches* – ie. there are no known demographics at the individual case level. Where-as the previous work is a straight-forward machine learning task with labeled cases, the problem that we describe in this paper is not – we don't have labeled cases – instead we have a labelled *aggregate*. When working with this kind of data, a variety of special techniques need to be used to cope with the loss of information.

2.2 Exact Demographic Re-Identification Given Sufficient Samples

It is possible to propose a "degenerate" solution to the problem, where individual user demographics can be re-identified exactly. Since the ad server can design the batches, it can then send samples of the set of users, with a different label each time, and observe the resulting probability distribution when each user is present in the sample.

Let *U* be a 0-1 matrix where columns are users, and rows refer to batches. *Y* is a vector of Oracle-provided audit scores with rows equal to batches. The unknown demographics *W* for each user are equal to:

$$W = U^{-1} \cdot Y$$

This has been used for inferring player contribution ratings in team games. Performance information is available at the team level, but we want to create rankings for individual team players. With enough repeated games it is possible to estimate the underlying player contribution scores. Huang et. al. (2006) and Menke and Martinez (2009) both describe systems that infer individual player performance using team outcome data.

This kind of solution is possible because the number of games (aggregate batches) is large, where-as the number of players in each game small. With 8 players, for example, one would expect the player's positive or negative contribution to the final score to be about 1/8 = 12.5%. In our domain, the batch sizes are very large (eg. 10,000). At this scale, one user's demographics has almost no bearing on the demographics of the aggregated batch; signal is about 1/10,000 = 0.01% - orders of magnitude weaker than in the game playing example. Thus specialized algorithms for selecting audits and using that audit information are needed to be effective with this problem.

Williams et. al. (2004) described a system used by Dstillery for predicting demographics from Oracle audits. The system assigned audit class labels to individual web requests, and then trained a model to predict the 0-1 class label based on requests. Dstillery's algorithm is the closest to our own work, and indeed they are a display ads server, where-as the authors of this paper describe work at a video ad server. The algorithm that we present has a significant computational complexity advantage over Dstillery's approach, which is a critical consideration for ad servers. We believe our algorithm also sheds some light on the nature of this problem and how it can be efficiently tackled.

3. PROBLEM DEFINITION

Let $X = (X_A..X_N)$ be an ordered pair of audit batches, each of which is batched by collecting together all web requests having property A. Their audited demographics are given by an ordered pair $Y = (Y_A..Y_N)$. In these audited results, the probability of demographic j is equal to $\Pr(d_j|x \in X_A)$. Let $x = (x_1..x_M)$ be an individual traffic request from batch X_A with properties $x_1..x_M$. Each property $x_i \in \{0,1\}$ is a 0-1 variable. The properties of the request could include browser, time of day, website from which the request is being made, and third party information about the cookie, and other HTTP headers. Our problem is to predict the probability of a demographic for a new request x using historical audit information.

3.1 User-Level Definition

Williams et. al. (2014) reported on a commercial approach to this problem in which user features were created, and then batch labels assigned, to requests to create training data. For example, given a batch of 10,000 users from the website.com, with age18to20 probability = 0.4, they create a training set with the same 10,000 users and their request attributes, each labeled with age18to20 probability equal to 0.4[1].

This approach will produce a huge data set. A typical ad server might have to process about 2 billion requests per day. Training complexity is at least $O(N \cdot M)$ where M are the number of properties and N the number of observations; in this case 2 billion. Figuring out how to operate a training algorithm over billions of requests would be an exciting Hadoop consulting problem - but is this volume of data really needed to solve this problem?

[1] Williams (2014) actually randomly assigns 40% of the underlying records to be 1, and 60% to be 0, so as to be able to re-use existing Dstillery code.

3.2 Equivalent Formulation in Audit Space

Let's consider what happens after the data above is sent to a training algorithm. We will use linear regression as an example, defining the problem in "request space", where X is a matrix with historical requests on rows and properties as columns, and where each element is 0 or 1, and Y are demographics inferred from the batch, and W is a vector with rows equal to number of properties.

$$Y = X \cdot W$$

The derivatives for the squared error of case i with respect to each weight j equals:

$$\frac{dE}{dw_{ji}} = (w_{ji}x_i - y_i)x_i$$

where x_i= 0 if the property i is not present, or 1 if it is present, and y_{iA} is the probability provided by the Oracle for batch X_A cases.

Since the historical web requests are actually "striped" in batches (Wikipedia, 2016), we now observe that y_i is the same for every row in the batch, and also x_i as 1 or 0 can be summed to create a probability which is measured for the batch. We can therefore introduce $\Pr(x_i|X_A)$ which is the probability of property i being present given that we are looking at results for batch X_A. The batch derivatives now become:

$$\frac{dE}{dw_{ji}} = \left(w_{ji}\Pr(x_i|X_A) - y_i\right) \cdot N \cdot \Pr(x_i|X_A)$$

where N are the number of impressions in the batch.

We can show that the derivatives for the above formula are the same as the derivatives for:

$$Y = P \cdot W$$

where $P = \Pr(x_i|X_A)$ is a matrix with properties across and audits down, Y is a vector of length equal to audits with demographics, and W is a vector of length number of audits and assuming equal audit batch sizes (if they are not equal then the error function becomes squared error weighted by the number of impressions in each audit).

We can now calculate derivatives in batches, assuming the existence of a new probability matrix P. This creates a significant computational complexity saving during training. Calculating P is an O(N) operation (assuming the property-audit probabilities are hashed in memory) and can be pre-computed in advance in distributed fashion. After this, the training algorithm only needs to operate on a matrix with size equal to the number of audits times properties $A \cdot M$, so a typical training complexity would then be $O(A \cdot M + N)$. The number of actual audits executed by ad servers tends to be small - a typical number might be A=10,000. Therefore training time will drop by a factor of around $2\text{x}10^5$ inclusive of the initial pass to calculate the probability matrix. The computational complexity reduction is significant – not only is training time lower, but many training algorithms need data to be loaded into main memory, and since the matrix size is reduced by the same factor, it is possible to run more complex algorithms in main memory.

4. DEMOGRAPHIC PREDICTION ALGORITHM

Adap.tv is one of the largest video ad servers in the United States in 2015, and is responsible for serving out about 13.2% of all US video ads. Google, in comparison, serves 11.1% of US video ads per month (Peterson, 2013; Comscore, 2014; Shields, 2015; Ember, 2015; Monica, 2015). We collected sampled Adap.tv data from July 2014 to July 2015. 39,546,119 observations were used to calculate the probability matrix. Oracle Audits were then collected at 1 month intervals between July 2014 and June 2015. Audit data was used for month m-1 to predict the audit results for month m.

When measuring hold out set accuracy, we found that properties that were rare, $\Pr(x_i) < \varepsilon$, yet had high probabilities of being present in some batches $\Pr(x_i|X_A) \approx 1$ would often produce predictions that would fail on the hold-out set. We gave these the colorful name of "sucker fish". Spurious variable associations have been noted by a variety of other authors to be a problem with large data sets (Anderson, et. al., 2001; Fan, 2014; Fan and Liang, 2014). We found that adding a threshold for the number of sites on which the property was expressed, was able to effectively remove the spurious "sucker fish" associations.

For smaller sites, the historical site audit demographics were very strong predictors of the future demographics of traffic on the site. This didn't work as well on larger sites such as Facebook. For larger sites, segments – i.e. properties that were specific to the user such as their interests – were better predictors.

We also found that when there were multiple segments, and the audits for those segments all tended to agree, then prediction using the average of segments tended to have good hold out set performance. If, on the other hand, there were multiple segments and their audits each disagreed, then it tended to indicate that the user/computer had mixed behavior and was hard to predict. In these situations we found that using the site was much more predictive.

We defined our predictor BAVG as follows:

$$\text{BAVG} = W \cdot \text{SAVG} + (1 - W) \cdot \text{U}$$

Where U was the historical audit for the URL or site. This provided a robust prediction if there was no segment information or the segment probabilities were contradictory (see below):

$$\text{U} = \Pr(d_j|\text{x} \in \text{X}_U)$$

SAVG were the average of audit results for segments on the web request, and only segments are averaged which appeared more than a threshold ε..

$$\text{SAVG} = \frac{1}{\#\text{X}_A}\sum_{\text{X}_A} \Pr(d_j|\text{x} \in \text{X}_A) : \Pr(\text{z} \in \text{X}_A) \geq \varepsilon$$

Weights W minimized the squared error between the predictor BAVG and actual demographic audit results. The weights determined how much emphasis to put on user-specific information (segments) versus the site URL. If the segments had high disagreement D, then more weight would be placed on the site.

$$W_T: \min \sum_{\text{X}_A} \left(\text{BAVG}(d_j|\text{x} \in \text{X}_A) - \Pr(d_j|x)\right)^2 : D(x) \in (L_T..H_T)$$

Each weight W_T was defined for a different level of "disagreement" between the segments, where disagreement was defined as the standard deviation of segment audit probabilities.

$$D(x) = \sqrt{\frac{1}{N}\sum_{X_A}\left(\Pr(d_j|x \in X_A) - \text{SAVG}\right)^2}$$

We also defined one other algorithm for analysis: segment at random "SEGR", which selected one of the web request properties, and returned its audit results without modification. This was a good diagnostic to see the effectiveness of "picking a segment at random".

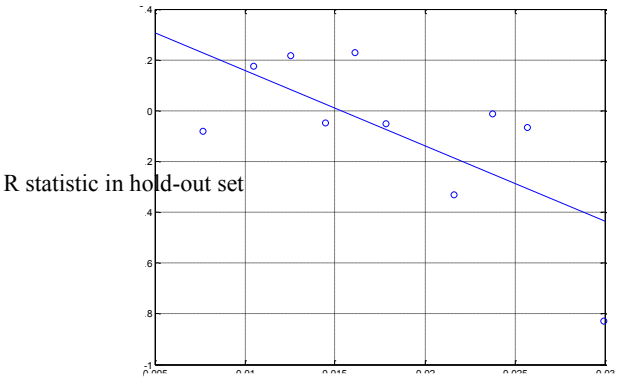

Figure 2. Standard deviation for audit probability from set of segments on incoming request (x-axis) versus hold-out set prediction quality (y-axis). The more that the segments disagreed, the worse became the prediction performance.

5. ACCURACY MEASUREMENT TECHNIQUES ON AGGREGATED DATA

5.1 Algorithm Audits

Measuring accuracy is also non-trivial since there is no independent source of truth other than the Oracle. Therefore we needed to devise a method for measuring algorithm accuracy. We called these "algorithm audits".

Let's say that a web request has been processed by the prediction algorithm above, and a demographic prediction created. This prediction could now itself be regarded as another 0-1 property of the web request. We can now create a batch label consisting of the combination of algorithm + demographic + score threshold for this particular web request (eg. Alg1 + M18to24 + 0.15..0.20).

In our experiments we used 5 algorithms, 6 score buckets and 24 demographics. For each algorithm audit the Oracle responded with another 24 demographics showing "actual" impressions across the demographics. This led to 17,280 combinations of algorithm + predicted demographic + score bucket + actual demographic and the resulting actual probability. Given an Oracle audited score tuple, s(a, di, sai, dj, sai)

5.2 Demographic Population Distributions

For each predicted demographic and probability (eg. Alg1 + Male18to24 + predicted probability=15%..20%), we could now show the distribution of actual returned demographics (eg. Male 18to24=12%, Male25to34=10%, ..., Female65+=4%). For an accurate demographic prediction algorithm – and a high prediction score for Male18to24 such as 20% - we would ideally see 100% of actuals in the true demographic, and then 0% in erroneous demographics. These graphs are shown in figure 3 and 4. This provides an easy to understand graphical picture of the quality of the prediction, and the match to ideal can be calculated using several methods (we will describe one method next).

5.3 In-Target Percentage

One method for summarizing the distribution match is to calculate the *in-target percentage*; which we can define as the percentage of the returned Oracle distribution that falls into the correct demographic. This also happens to be the key metric that advertisers use when planning their campaigns. In-Target Percentage is equal to the percentage of impressions bought which match the demographic that they are targeting – this rate should be proportional to advertiser revenue per impression.

Many reporters talk about achieving In-Target Percentages of 46% or better. However it is usually omitted that expected In-Target Percentages vary based on the particular age-gender range that is being targeted (Nielsen, 2014). For example, Adults 25 to 54 has a random In-Target rate of 56% - so actually a report of in-target at 50% is actually worse than random. Therefore in order to make the demographics comparable, we typically add the expected In-Target Rate at random, and then report on In-Target Lift as the In-Target Rate divided by the random In-Target Rate. This provides an apples-to-apples measure of quality.

Figure 5 shows that BAVG produces 5.1x In-Target *Lift* if an advertiser were to use predictions in the 20% or above prediction bucket.

5.4 In-Target Estimate of Bid Error

Although In-Target Percentage is used widely in business, it is not ideal for ad buying calculations. We need a measure that captures the quality of prediction across all traffic and predictions.

It is typical in machine learning to measure prediction quality using Area Under the ROC curves (AUC). However this can lead to misleading results in this particular domain. AUC is invariant to scale, shift and rank-preserving non-linearities. If the in-target prediction is consistently offset too high, or consistently scaled too low, then the resulting bid prices will be too high, revenue losses will accrue.

We therefore need another measure that captures the effectiveness of the in-target prediction for ad buying purposes, where the particular scale and bias in the numerical score matters. During bidding, the absolute difference between bid price placed b_i given the prediction provided, and optimal bid price b_i^*, had we had a predictor that exactly equaled actual is equal to:

$$err_t = \sum_i^N |b_i^* - b_i| \quad (6)$$

A bid that maximizes spend subject to a CPA_t constraint is to set bid price equal to the in-target rate multiplied with CPA_t

$$b_i = y_i \cdot CPA_t$$

where y_i is the in-target rate of impression *i*., Assuming y_i^* is the actual in-target rate (that the predictor should have predicted), we can now re-write our bid error formula (6) as follows:

$$err_t = \sum_i^I \left| y_i^* \cdot CPA_t - y_i \quad \cdot CPA_t \right|$$

$$= CPA_t \sum_{i}^{N} |y_i^* - y_i| \quad (7)$$

Therefore, the sum of difference of in-target predicted versus actual better captures the economics of how in-target rates are used in ad buying than some of more commonly used metrics such as AUC. Figure 11 shows squared differences by demographic for BAVG versus SAVG.

We can also show overall prediction quality graphically by showing a scatterplot of forecast versus actual using the prediction buckets as the forecasts. A perfect predictor would have predictions exactly on the diagonal of this chart. Any divergence is This is shown in figure 6.

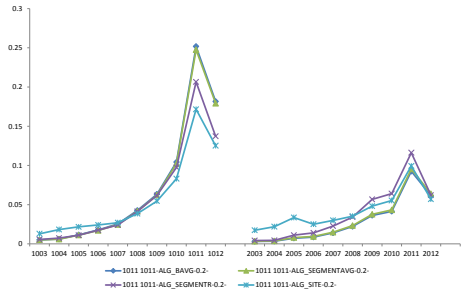

Figure 3. Distribution comparison for two algorithms; each algorithm is predicting high probability of demo=1011 (Female Age 55-64). One can see that BAVG has more Oracle impressions in the 1011 bucket, and fewer in other buckets.

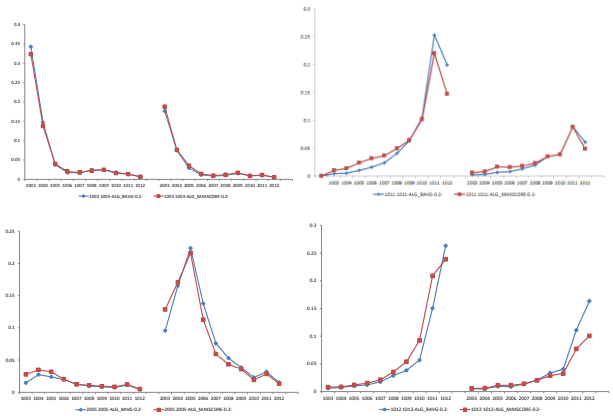

Figure 4. Distribution comparison for two algorithms across several demographics. In general BAVG produces more impressions in the correct demographic, and fewer in incorrect demographics.

Demographic	Expected In-Target (random)	Demographic	Expected In-Target (random)
FEMALE 18-20	2.54%	MALE 18-20	2.78%
FEMALE 21-24	2.99%	MALE 21-24	3.68%
FEMALE 25-29	3.72%	MALE 25-29	6.23%
FEMALE 30-34	3.77%	MALE 30-34	5.29%
FEMALE 35-39	3.57%	MALE 35-39	4.67%
FEMALE 40-44	4.09%	MALE 40-44	5.03%
FEMALE 45-49	4.26%	MALE 45-49	5.37%
FEMALE 50-54	5.31%	MALE 50-54	4.66%
FEMALE 55-64	9.48%	MALE 55-64	8.71%
FEMALE 65+	7.58%	MALE 65+	6.26%

Figure 4. Expected In-Target Rates at random for 20 mutually exclusive demographics. The expected rate for any combination can therefore be summed, eg. the expected In-Target Rate for F25to49, assuming random ad serving, is 19.4%.

lower	upper	BAVG	SITE	SAVG	SEGR
0.20	1.00	5.19	4.62	4.17	3.91
0.15	0.20	3.53	3.15	3.09	2.82
0.10	0.15	2.14	2.23	2.84	2.18
0.05	0.10	1.18	1.23	0.94	1.27
0.01	0.05	0.64	0.98	0.88	1.11
0.00	0.01	0.06	0.12	0.47	0.42

Figure 5. In-target lift for 6 probability thresholds. In the highest probability bucket, predictions have a lift that is

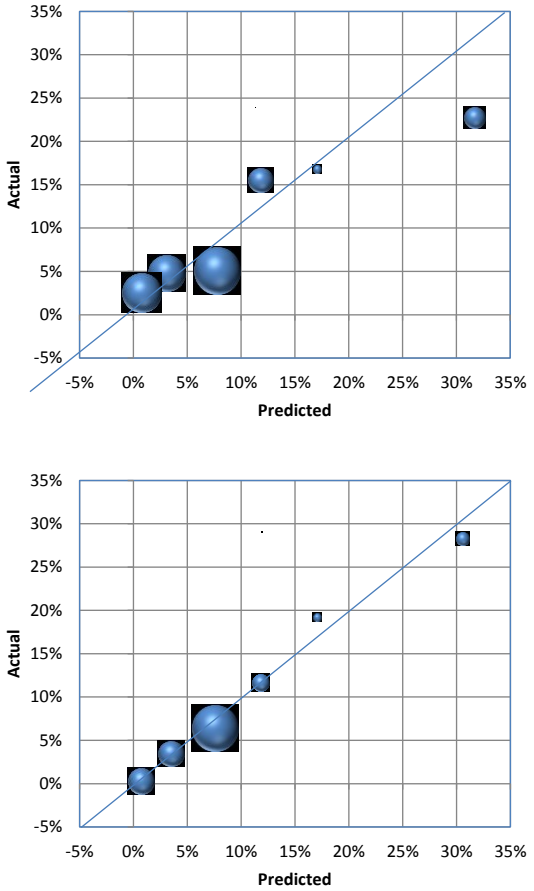

Figure 6. Prediction versus actual for SAVG (left) and BAVG (right). Bubble size is equal to the percentage of traffic in this prediction bucket. BAVG has much lower average error. In addition, SAVG even shows some signs of regression the mean phenomena – the lower bucket is too high and the highest bucket is too low. This can occur when spuriously high or low predictions are used. BAVG reverts to site information when there is disagreement amongst segments.

Low	Hi	BAVG mean pred	BAVG Oracle actual	BAVG traffic	SAVG mean pred	SAVG Oracle Actual	SAVG traffic
0.20	1.00	28.3%	30.5%	4.3%	22.7%	31.7%	6.8%
0.15	0.20	19.3%	17.0%	1.3%	16.8%	17.1%	1.0%
0.10	0.15	11.7%	11.8%	7.3%	15.5%	11.8%	9.8%
0.05	0.10	6.4%	7.6%	53.9%	5.1%	7.8%	35.8%
0.01	0.05	3.5%	3.5%	16.2%	4.8%	3.0%	21.9%
0.00	0.01	0.4%	0.7%	17.0%	2.5%	0.7%	24.6%

Figure 7. Predicted demographic probability versus actual demographic probability. For example, for traffic that was in bucket 0.20..1.00, the BAVG algorithm predicted the traffic's probability of having the demographic equal to 28.3%. Actual was 30.5%. SAVG predicted 22.7%, which was much lower than actual at 31.7%

Demo	BAVG squared error	SAVG squared error	Demo	BAVG squared error	SAVG squared error
F 18-20	0.033%	0.038%	M 2-11	0.010%	0.108%
F 21-24	0.084%	0.135%	M 12-17	0.074%	0.144%
F 25-29	0.017%	0.029%	M 18-20	0.062%	0.119%
F 30-34	0.044%	0.030%	M 21-24	0.013%	0.015%
F 35-39	0.039%	0.001%	M 25-29	0.038%	0.058%
F 40-44	0.016%	0.019%	M 30-34	0.038%	0.050%
F 45-49	0.028%	0.055%	M 35-39	0.037%	0.065%
F 50-54	0.010%	0.020%	M 40-44	0.079%	0.082%
F 55-64	0.094%	0.105%	M 45-49	0.057%	0.085%
F 65+	0.133%	0.091%	M 50-54	0.036%	0.042%

Figure 11. Mean absolute error across entire range of prediction scores, weighted by traffic, by demographic for two algorithms. Error is lower for BAVG in 17 out of 20 cases.

5.5 Controlled Test Ads

In order to test the demographic prediction systems in practice, they were deployed and used for 6 live advertiser campaigns. We created 3 ads using algorithm BAVG and another 2 using SAVG. Female 18to24 was chosen as the demographic target, because the background rate was low (5.37%) and so it would be easier to measure statistical significance. The ads each were to deliver 5,000 impressions over 30 days with a maxCPM of $10. We were only able to test two of the algorithms in this way. This provided an end-to-end test on a major ad server, with the only variation being the demographic prediction algorithm. The results suggest that BAVG delivers 3.5 lift over random, and SAVG delivers 2.3 (Figure 8).

5.6 Comparisons to Commonly-Used Commercial Demographic Providers

For comparison we also audited 12 demographic provider companies against the Oracle, and the very best of those showed an in-target lift of 2.4 over random (actually comparable to SAVG at 2.3). The average lift over random was 1.6 (Figure 10). These 12 particular provider companies were only selected because we had clear naming of age-gender and it matched our test demographics of W18to24 – we did not filter the list in any way, so it provides a useful picture of expected accuracy having randomly drawn 12 commercial demographic providers.

One possible reason why both SAVG and BAVG performed well compared to commercial data providers is the latter might set their services to *overly emphasize recall at the expense of precision*. Ultimately advertisers pay for these services only if they find them useful, these services may be inclined to provide a predicted positive in many more cases where the evidence for age or gender is slim.

For example, were we to buy the top 12.9% of traffic in BAVG, lift would drop to 2.1x. If we had to buy the top 66.7% of traffic, lift would drop further to 1.18x (Figure 7 and 5). Thus it is conceivable that the commercial services could be capable of higher lift than what we are seeing here, if they were to report back at a lower rate.

	Cell	Oracle reported random occurrence for W25to54	Oracle reported W25to54 in-target rate; test campaign	In-Target Lift	Imps
BAVG	All Ads	5.37%	19%	3.55	5,952
	Fresno	5.37%	20%	3.73	3,017
	Shreveport	5.37%	19%	3.60	1,566
	Wilmington	5.37%	17%	3.09	1,369
SAVG	All Ads	5.37%	12%	2.29	3,289
	Florence	5.37%	10%	1.79	1,443
	LittleRock	5.37%	14%	2.67	1,846

Figure 8. Demographics purchased in a Live Ad Campaign targeting W18to24, with Oracle reported In-Target Percentage and Lift reported.

Demo Company	In-target lift over random	Demo Company	In-target lift over random
A	2.39	H	1.42
B	2.26	I	1.36
C	1.89	J	1.08
D	1.82	K	1.02
E	1.81	L	1.01
F	1.67	M	1.00
G	1.43		

Figure 10: Twelve Demographic Provider companies (names withheld) and their in-target rates for W18to24 as measured by Oracle. In-target lift ranges from 1 (random) to 2.39 for the very best company. The companies listed are from a set including V12 Group, DLX Demographics, Dataline, BK Demographic, IXI, Media Source, Webbula, Experian, VisualDNA, I-Behavior, Lotame, Alliant, Relevate.

6. CONCLUSIONS

We have discussed the problem of predicting from aggregated labeled data. We have also shown how the problem can be decomposed to reduce training complexity by separating probability calculations from the training step and changing the problem to one in audit space instead of request space. We have also reported on some algorithms and shown how algorithm measurement can be performed.

7. ACKNOWLEDGMENTS

Thanks to Amir Cory and R. Luenberg. Umayr Hassan developed the weblogging infrastructure used for this paper (called Jambalaya).

8. REFERENCES

[1] http://www.adap.tv accessed July 12, 2015.

[2] Anderson, D., Burnham, K., Gould, W., Cherry, S. (2001), Concerns about finding effects that are actually spurious, Wildlife Society Bulletin, Vol. 29, No. 1. Spring 2001.

[3] Bi, B., Shokouhi, M., Kosinksi, M., Graepel, T. (2010), Inferring the Demographics of Search Users, Microsoft Report.

[4] Ding, W. and Marchionini, G. 1997. *A Study on Video Browsing Strategies*. Technical Report. University of Maryland at College Park.

[5] Ember, S. (2015), AOL Unveils ONE by AOL, an Advertising Platform, April 14, 2015. http://www.nytimes.com/2015/04/15/business/media/aol-unveils-one-by-aol-an-advertising-platform.html

[6] Computerworld (2013), http://www.computerworld.com/article/2921585/it-industry/verizons-buy-of-aol-would-offer-edge-against-google-facebook-on-mobile-ads.html

[7] Comscore (2014), Comscore Releases March 2014 US Online Video Rankings: Liverail serves the most video ads in march, takes top spot in ad ranking, April 18, 2014.http://www.comscore.com/Insights/Press-Releases/2014/4/comScore-Releases-March-2014-US-Online-Video-Rankings

[8] Everett, J. E. , (2013), Planning an Iron Ore Mine: From Exploration Data to Informed Mining Decisions, Issues in Informing Science and Information Technology Volume 10.

[9] Fan, J. (2014), Features of Big Data and Sparsest Solution in High Confidence Set, in Lin, X., Genest, C., Banks, D., Molenberghs, G., Scott, D. and Wang, J. Past, Present, and Future of Statistical Science, Chapman and Hall, pp. 507–523

[10] Fan, J. and Liao, Y. (2014), Endogeneity in high dimensions, Ann. Statist. Volume 42, Number 3 (2014), 872-917.

[11] Hua, J. et. al., (2007), Demographic Prediction Based on User's Browsing Behavior, WWW 2007. May 8–12, 2007, Banff, Alberta, Canada.

[12] Huang, T., Lin, C., Weng, R. (2006), Ranking Individuals by Group Comparisons, 23rd International Conference on Machine Learning, pp. 425-432. ACM Press.

[13] Kabbur, S. Han, E. Karypis, G. (20-0) Content-based methods for predicting web site demographic attributes,

[14] La Monica, P. (2015), Verizon wants to eat Google's and Facebook's lunch, CNN Money, May 12, 2015, http://money.cnn.com/2015/05/12/investing/verizon-aol-mobile-video-advertising/

[15] Menke, J., and Martinez, T. (2008), A Bradley-Terry Artificial Neural Network model for individual ratings in group competitions, Neural Computing and Applications, Vol. 17, No. 2, pp. 175-186.

[16] Menke, J., and Martinez, T. (2009), Artificial Neural Network reduction through oracle learning, Intelligent Data Analysis, Vol. 13, No. 1. Pp. 135-149.

[17] Nielsen Corporation (2014), Defining Online Ad Success: How Benchmarks are shifitng as advertisers take aim, Nielsen website, Sep 2014, http://www.nielsen.com/us/en/insights/news/2014/defining-online-ad-success-how-benchmarks-are-shifting-as-advertisers-take-aim.html

[18] Nielsen Corporation (2016), Digital Ad Ratings product information website, accessed Jan 31, 2016. http://www.nielsen.com/us/en/solutions/measurement/online.html

[19] Oracle website, accessed July 25, 2015 http://www.oracle.com/webfolder/assets/cloud-data-directory/index.html#/page/1

[20] Peterson, T. (2013), With Adap.tv, AOL Beats Google In Video Ads Served: Google Continues to Attaract Most Video Viewers, Ad Age, October 17, 2013

[21] Reagan, R. (2013), Nielsen's OCR & comScore's VCE Drive Spend From TV, ExchangeWire, June 13, 2013, https://www.exchangewire.com/blog/2013/06/13/nielsens-ocr-comscores-vce-drive-spend-from-tv-by-catherine-hallam-international-product-manager-videology-group/

[22] Relevate web site, accessed July 25, 2015, http://marketing.relevategroup.com/acton/attachment/12124/f-01f8/1/-/-/-/-/New%20Movers%20ID.pdf

[23] Shields, M. (2015), With "One," AOL Promises It Can Help Manage Every Dollar a Brand Spends, Wall Street Journal, April 14, 2015 http://blogs.wsj.com/cmo/2015/04/14/with-one-aol-promises-it-can-help-manage-every-dollar-a-brand-spends/

[24] Ulges, A., Koch, M., Borth, D. (2012), Linking Visual Concept Detection with Viewer Demographics, ICMR '12, June 5-8, Hong Kong, China

[25] Weber, I. and Castillo, C. (2010), The Demographics of Web Search, SIGIR 2010, July 19-23 2010, Geneva-, Switzerland.

[26] Wikipedia (2016a), Computational Complexity of Mathematical Operations, Wikipedia listing accessed 2/19/2016, https://en.wikipedia.org/wiki/Computational_complexity_of_mathematical_operations, accessed July 15, 2015.

[27] Wikipedia (2016b), Data Striping, Wikipedia listing accessed 2/19/2016, https://en.wikipedia.org/wiki/Data_striping,

[28] Williams, M., Perlich, C., Dalessandro, B., Provost, F. (2014), Pleasing the advertising oracle: Probabilistic prediction from sampled, aggregated ground truth, in Proceedings of the Eighth International Workshop on Data Mining for Online Advertising (ADKDD'14). ACM, New York, NY, USA.