2010

Modeling Conversions in Online Advertising

John Winston Chandler-Pepelnjak
The University of Montana

MODELING CONVERSIONS IN ONLINE ADVERTISING

by

John Chandler-Pepelnjak

B.A. Middlebury College, USA 1996

M.S. University of Washington, USA 1999

presented in partial fulfillment of the requirements

for the degree of

Doctor of Philosophy

The University of Montana

May 2010

Approved by:

_____

Committee Chair

_____

Dean, Graduate School

_____

Date

Chandler-Pepelnjak, John W. Ph.D., May 2010 Mathematics

Modeling Conversions in Online Advertising

Committee Chairs: David Patterson, Ph.D. and Brian Steele, Ph.D.

This work investigates online purchasers and how to predict such sales. Advertising as a field has long been required to pay for itself—money spent reaching potential consumers will evaporate if that potential is not realized. Academic marketers look at advertising through a traditional lens, measuring input (advertising) and output (purchases) with methods from TV and print advertising. Online advertising practitioners have developed their own models for predicting purchases. Moreover, online advertising generates an enormous amount of data, long the province of statisticians. My work sits at the intersection of these three areas: marketing, statistics and computer science. Academic statisticians have approached the modeling of response to advertising through a proportional hazard framework. We extend that work and modify the underlying software to allow estimation of voluminous online data sets. We investigate a data visualization technique that allows online advertising histories to be compared easily. We also provide a framework to use existing clustering algorithms to better understand the paths to conversion taken by consumers. We modify an existing solution to the number-of-clusters problem to allow application to mixed-variable data sets. Finally, we marry the leading edge of online advertising conversion attribution (Engagement Mapping) to the proportional hazard model, showing how this tool can be used to find optimal settings for advertiser models of conversion attribution.

Acknowledgements

First and foremost I would like to thank the co-chairs of my committee, David Patterson and Brian Steele. Their patience and perseverance allowed me to survive this process. I have been their student for seven years now and have learned a tremendous amount about both statistics and mentorship. I would particularly like to thank Brian for his encouragement to pursue a topic related to my work in online advertising—that might be the single most valuable piece of advice from any source.

I would like to thank my other committee members Jon Graham and Solomon Harrar for their support and encouragement. I would especially like to thank Jakki Mohr. She went far above and beyond the typical duties of an external committee member, and both I and my dissertation are much improved from her assistance.

I would like to thank my colleagues for their encouragement and support. I would not have begun this second trip through academia without my boss of nine years, Young-bean Song. I certainly would not have finished the dissertation without the help of his successor, Esco Strong. The genesis of Chapter 4 came from a conversation with Erik Hanson and I appreciate his contributions through my work. Andy Martin initiated the project that became Chapter 6 and helped me clarify my thoughts. I appreciate the assistance from my other colleagues Matt Butcher, Jed Fowler, Morris Martin and Iqbal Nijjar. Thanks to Mark Smucker for getting me to use R back before it was popular and encouraging me to pursue statistics.

I would like to thank my partners in crime in the statistics department from 2003 to 2008, Joran Elias and Cindy Scavarda. They made learning lots of fun.

Thanks to my parents for their continual interest in my progress, even if I chafed under the steady questioning. Thanks also to my friend John Adams for his much-better-calibrated interest in my work.

Finally and most importantly, I'd like to thank my wife, Cori Chandler-Pepelnjak, for her steady support, shocking tolerance, and general understanding of this academic pursuit. I promise that in the future, when I enter into projects that will take seven years and thousands of hours, I will not frame it as an exciting surprise. Never a dull day, indeed.

Contents

Abstract ii

Acknowledgements iii

List of Tables vii

List of Figures viii

1 Introduction 1

2 Introduction to Online Advertising and Conversion Attribution 3

2.1 Online Advertising 4

2.2 Conversion Attribution 9

2.2.1 Conversion Attribution pre-2007 9

2.2.2 An Emerging Standard: Engagement Mapping 11

2.3 E-Map Definition 13

3 Academic Results 18

3.1 General Advertising Research 19

3.2 Response Modeling 25

4 Finding Drivers of Conversions with Hazard Models 29

4.1 Proportional Hazard Models 30

4.2 A Click Only Model 40

4.3 Data for PHM with Time-varying Covariates 41

4.4 Example: Retail Advertiser 48

4.4.1 Model Fitting 49

4.4.2 Assessing Model Fit 52

4.4.3 Translation to E-Map 55

4.5 The Case Against Last-Ad 56

4.6 Computational Statistical Results 60

5 Visualizing Cookie Histories 69

5.1 Predicting and Plotting Conversion Probabilities 70

5.2 The Tie to Engagement Mapping 76

5.3 A Statistical View 83

6 Finding Common Cookie Histories with Clustering 85

6.1 Clustering Based on User Histories 87

6.1.1 Data for Clustering Cookies 87

6.1.2 Calculating a Distance Matrix 90

6.1.3 Partitioning-based Clustering 92

6.1.4 Hierarchical Clustering Approach 97

6.1.5 The Gap Statistic 100

6.2 Clustering the Hotel Data with PAM 102

6.2.1 Estimating the Hotel-Data Clusters with GAP 113

6.3 A Statistical View 114

7 Conclusion **118**

Glossary **121**

Code **126**

Bibliography **156**

List of Tables

2.1 Example of ad-serving log records 6

2.2 Credit sharing across six exposures 14

4.1 Distribution of clicks in the retail data set. 40

4.2 Retail data model estimates 51

6.1 An extensive summary of the hotel advertiser data. 89

6.2 The number of observations per cluster with $k = 8$ for four different clustering algorithms. 100

6.3 The summary of cluster membership stability on the hotel data across a variety of k values. 112

List of Figures

2.1 E-Map scores across seven exposures with three models 16

3.1 An illustration of the framework for understanding the mechanism of advertiser from Vakratsas and Ambler. 20

4.1 Kaplan-Meier Survival Curve 32

4.2 Kaplan-Meier Survival Curve for Click-only Model 42

4.3 Estimates for retail-data PHM 50

4.4 Data for the Hosmer-Lemeshow test of fit 54

4.5 Example Figure for E-Map Order Parameter 57

4.6 An illustration of conversion percentages in groups. Each dot represents a collection of 50 cookies. As we move from left to right the estimated conversion probability in the groups increases. The horizontal line represents the normalized conversion probability in the sample. The imposed curve is a lowess smooth. It appears for the top third of cookies we are making predictions with some accuracy. The bottom two-thirds show no real prediction pattern. 59

5.1 PHM conversion probability plot, high volume 72

5.2 PHM conversion probability plot, low volume 73

5.3 PHM conversion probability plot, ten cookies compared 77

5.4 Example of E-Map Cookie Plot 81

5.5 Example of E-Map Cumulative Cookie Plot 82

6.1 Data summarizing a three cluster solution for the hotel data. 95

6.2 Data summarizing a three aggolmerative cluster solution for the hotel data. . . 99

6.3 A summary of the hotel data, displayed in one "cluster." 104

6.4 Data summarizing a two-cluster solution for the hotel data. 105

6.5 Data summarizing a three cluster solution for the hotel data. 106

6.6 Data summarizing a four cluster solution for the hotel data. 108

6.7 Data summarizing an eight cluster solution for the hotel data. 111

6.8 The scree plot for the hotel data—a diagnostic for the number of clusters. . . . 113

6.9 The GAP plot for the hotel data. 115

Chapter 1

Introduction

This dissertation sits at the intersection of online advertising and statistics. Fundamentally I seek to address the question, What makes people respond to online advertising and how can we measure that response? Online advertising is a new domain within modern advertising, itself young when compared to mathematics. I begin with a thorough introduction to the mechanics and practice of online advertising, paying particular attention to the data that are available for further modeling.

We then immerse ourselves in a topic called "conversion attribution". A conversion is an online action like a sale or online sign-up. We wish to attribute this conversion to the marketing that led to it, appropriately sharing the credit across all consumer advertising events that drive conversions. Many media channels are not trackable in this sense (TV, magazines, etc) and thus we confine our studies to online marketing. With a response variable of a conversion and explanatory variables of all aspects of trackable media, we seek to build models.

Before building models, however, Chapter 3 discusses current marketing research. We begin by looking at marketing research in general. Since online advertising is so new, the literature only extends back about 15 years. By embedding online advertising within the larger field of

marketing we can draw on closer to 60 years of research as well as take advantage of a series of survey articles published in the last 10 years. We then turn our attention to modeling of response and conversion attribution in the literature, bringing us up to the the state-of-the-art, proportional hazard models.

The next two chapters are complementary. Chapter 4, assimilates ideas from the academic literature, statistics, and the work of earlier chapters. There is a smattering of computer science topics—any work with data in the volumes created by the web requires intensive computer work to even initiate. From a statistical perspective, the principal contribution to the state-of-the-art is improvements to the proportional hazard model fitting software that allows work with much larger data sets than before. Estimating Engagement Mapping (E-Map) parameters using proportional hazard models with time-varying covariates, shows great promise and is the launch pad for future research, discussed in the conclusion. Chapter 5 expands on the proportional hazard model visualization techniques. Rather than using traditional survival plots, we explore a method of plotting E-Map scores directly. E-Map is an unusual approach from a statistical standpoint, not being based on distributional theory but instead on marketing expertise and a collaboration between leading advertisers, agencies and publishers. My employer, Microsoft Advertising, has been instrumental at driving E-Map adoption. This chapter discusses the role of the survival function in visualizing a user's history. We then extend the visualization approach so that it can be applied to any set of data with any E-Map model, independent of the survival model framework.

The final chapter, 6, attempts to better understand why people convert online through a series of cluster analyses. In this chapter we modify an existing statistical technique to determine the number of clusters, for the first time allowing the Gap statistic to be used for categorical or mixed data sets. We provide an in-depth discussion of the a clustering solution with the suggested number of clusters (eight), illustrating how these ideas can be applied.

Chapter 2

Introduction to Online Advertising and Conversion Attribution

The growth of the web and the pervasiveness of internet access has created unprecedented information sharing and opportunities for people to engage with each other, with corporations, and with ideas. Concurrent with this growth have been business opportunities both in terms of selling via the web and sites using advertising to provide their content free to consumers. Consumers typically pay for access to the internet (often to the phone company or cable company) but once someone is online most websites freely share their content. The cost of creating and distributing that content is paid by advertisers. These days every major corporation has a website and corporations that sell goods to consumers often use those websites for commerce. To encourage customers to visit their site (think eBay getting people to come bid on auctions), the sites use advertising. In this chapter I am going to cover two aspects of this business. The first section covers the basics of the online advertising business and the kind of data that is collected. The second section covers "conversion attribution", the process by which statistics is used to determine why people respond to advertising. In the third section we define, at a granular level, the conversion attribution algorithm known

as "Engagement Mapping".

2.1 Online Advertising

In the last 15 years, a number of different approaches to advertising online have been devised. The current incarnation is that a business purchases either ad space from an online publisher or purchases search keywords, usually with an advertising agency as intermediary. Search keywords are bought based on a particular piece of text (e.g., EBay might buy the keyword "used DVD") and the search engines display an ad if the keyword bid is high enough and the ad content deemed relevant enough. If the ad is clicked on the advertiser pays the search engine a certain amount of money, called the cost-per-click (typically 20 cents to a few dollars, but ranging up to a hundred dollars for very valuable keywords[1]).

Non-search advertising is typically called "display advertising" and its business model is more varied and quite different from search. Typically an advertiser or agency will purchase space for ads through an online publisher such as Yahoo! or ESPN. The unit of measurement for this advertising in the online space is an impression—the viewing of one advertisement by one person[2]. The advertiser will pay a fee for the impressions to be shown. Typically the deal will be structured so that the advertiser is paying a rate (typically in the $3 to $50 range) for 1,000 ads, known as cost-per-thousand (CPM[3]). Advertisers typically employ a technological intermediary as part of this process. Rather than maintaining the files for the ads and serving those ads to the publishers at the time of the request, advertisers employ a third-party ad server (TPAS) to perform this role. The TPAS are responsible for the technological infrastructure that enables the ad-serving relationship as well as functioning

[1]The most lucrative keyword I've ever heard of is the keyword "mesothelioma" a lung cancer caused by, among other things, asbestos fibers. At the peak of the asbestos litigation fervor, this keyword was selling for over $100 on Google[3].

[2]There is an incredible amount of work that has actually gone into defining an "impression". Typically it is defined as the request for and subsequent attempt at delivery of the file associated with the ad space. From here the definition spins off into mind-numbing technology minutia.

[3]There is a glossary appended to this document defining advertising terms such as "CPM".

as a trusted third-party maintaining the accounting of the advertising system. Microsoft Advertising owns and operates a TPAS (called Atlas) and that is the source of the data I discuss throughout this research.

The mechanics of how a TPAS works is not that germane to the conversion attribution problem we will be discussing. The gist is that as ads are served, certain information is recorded in log files. Additionally, on advertiser's websites (e.g., Best Buy selling electronics) Atlas has tags that allow data to be gathered when someone visits a webpage or purchases. This data is collected anonymously using cookies[4]. In order to make this example concrete, Table 2.1 contains example log records.

This table contains the typical fields captured by Atlas in the course of third-party ad serving on behalf of advertisers. The data represented here are four records for one cookie. The first record is a click on an ad (denoted by the click=1 field). The second record is an action, in this case it happens to be a user signing up for an online service. Action records have non-zero values in the action column and zero values for the Placement and Ad ID columns. The last two records are impressions–views of ads. These are distinguished by non-zero Placement and Ad ID columns with zeros in the Action and Click columns. Here is a brief discussion of the various fields in these log records.

- Cookie: This is the unique identifier for a computer. Multiple people can use one computer (and hence have one cookie) and cookies can be deleted. These records are imperfect but they are the best, easily-accessible method of identifying people [5].
- Tentative: This field is 1 if we have one and only one record for the cookie in their entire history. Typically this happens when someone has set their browser security settings

[4] "Cookies" are small text files stored on a user's computer. Typically these are simply long somewhat random numbers that identify the computer and browser for the purposes of anonymous tracking. Cookies are, for instance, the way Amazon remembers who you are when you come back to the site and can thereby build a custom homepage for you.

[5] Technically a cookie is a text file that resides on the web surfer's computer. Typically this file contains a unique number identifying the computer. The server that creates the cookie—and only that server—can read the cookie on subsequent visits, thereby recognizing that computer.

Row	Cookie	Tentative	IP Address	DateTime	PlacementID	AdID	Click	ActionID
1	9798741-1597548	0	206.148.160.0	2007-08-06 19:01:03.0	18736771	17487593	1	0
2	9798741-1597548	0	206.148.160.0	2007-08-10 00:52:08.0	0	0	0	55395
3	9798741-1597548	0	206.148.160.0	2007-08-12 10:29:23.0	18736771	29266992	0	0
4	9798741-1597548	0	206.148.160.0	2007-08-30 10:25:54.0	10810596	29266997	0	0

Table 2.1: This table contains the typical fields captured by Atlas in the course of third-party ad serving on behalf of advertisers. The data represented here are four records for one cookie. The first record is a click on an ad (denoted by the click=1 field). The second record is an action, in this case it happens to be a user signing up for an online service. Action records have non-zero values in the action column and zero values for the Placement and Ad ID columns. The last two records are impressions–views of ads. These are distinguished by non-zero Placement and Ad ID columns with zeros in the Action and Click columns.

to reject cookies and so every time we see the person it is as though for the first time. In most cases cookies with Tentative=1 are excluded from analyses and we will exclude them from our future analyses.

- IP Address: This is the internet address from which the request[6] for an impression, click, or action came. Using this field we can determine (in most cases) geographically where the user is located, their connection speed, and whether they are surfing from home or work.

- Datetime: This is the date and time (down to milliseconds) when the request generating the log record was received by the ad server.

- PlacementID and AdID: Broadly speaking, there are two types of records: those recording advertising (both views and interactions) and those recording user actions. For records that describe advertising, these two fields will be non-zero and will contain identifying numbers that allow us to determine where the media was shown and which advertisement was shown. The PlacementID defines the place where the media was purchased. In the traditional world this might be a specific ad in a magazine or newspaper or the first commercial in a television show. Online this typically refers to a place for ad content on a webpage. Basically, the placement is the "physical" location where the ad ran. The AdID field gives us the unique identifier for the specific creative that was shown on the placement. This is what people think of as the "ad".

- Click: This field is set to 1 if the request is a click. This happens when someone clicks on an ad.

- ActionID: This field is non-zero if someone hits an "action tag". An action tag is essentially a marker on a page that keeps track of people hitting the page. The most popular use of action tags is by advertisers who want to measure how advertising has

[6]The term "request" seems somewhat odd since advertisements seem foisted upon us. Nevertheless, as a webpage is loading, each element of that page is requested from the various servers that supply parts of the page. The same is true for clicks (in which case a redirect from the current website to another website is requested) and actions (which are really just impressions in disguise.)

influenced people hitting a given page. For instance, Nordstrom could set up an action tag on the page that thanks people for purchasing or Best Buy could put an action tag on the last step of their shopping process. The action records allow us to make online advertising accountable. In table 2.1 we see that on August 6, 2007 the user clicked on an ad. Then on August 10 they hit the action tag.

I've shown four log records (for one cookie and one advertiser in one month). On a typical day in 2009 Atlas creates about 15 billion log records with a peak throughput of over 250,000 ads per second. This number comprises billions of impressions, hundreds of millions of clicks, and millions of actions. (Some actions, such as users hitting the homepage of a website, are relatively common.) These data form the underlying currency of the online economy because they are used to justify advertising expenditures—a $20 billion annual business. The typical use of these data is in summary form. For instance, Atlas will sum up the number of ads shown for an advertiser on a particular publisher's website over the course of a month so that the former will pay the latter appropriately. Atlas also adds up the number of actions so that advertisers can measure their performance. An advertiser might see that a certain site is bringing in too little business relative to the cost of the ads. In this case the advertiser might stop advertising on the site or maybe they would ask for some free inventory in order to bring the site's performance in line with other sites on the media plan.

But summary data only scratches the surface of what is possible. Within these cookie histories are rich stories about how people are using the internet, how they are interacting with sites and advertisers, and the mechanisms that are encouraging them to transact online. Using this detailed record to learn about online marketing is the goal of my dissertation. In particular, the next section will lay the foundation for the meat of this work: investigating why people respond to online advertisements.

2.2 Conversion Attribution

2.2.1 Conversion Attribution pre-2007

How did online advertising grow from nothing to a $20 billion per year business in 15 years? The central answer revolves around the concept of "accountability". For most media (television, radio, newspaper, out-of-home) there is no direct way to tie advertising to sales. There are approximate schemes (such as only advertising on TV in certain regions or using focus groups to ask about ad recall and intent-to-purchase) that can provide estimates of the effects of advertising, but these are fraught with error and approximation. We will detail some of these methods in Section 3.1 and discuss the limitations and results. Online advertising is different because both the delivery of advertising and the transactions can be tracked through a cookie.

The analog for this cookie-level tracking of ads and actions would be if, for example, Nike could assign a number to track each viewer of their running shoe ads on television. Then, at the time of sale, Nike would have some way to read the number. Then they could go back through the delivery records and see which ads and programs brought in the most buyers. Since every ad view and click is tracked, along with any online purchases, this advertising utopia (or dystopia, depending on your philosophical bent) basically exists online.

So what mechanism should the advertisers use to determine whether or not an ad has influenced a subsequent sale? Reader, you could probably rattle off a number of useful formulas more sophisticated than the one the industry currently uses: the last-ad model. The last-ad model is dead simple. The most recent ad seen gets 100% of the credit for an action unless there is also a click in the user history, in which case the click gets all the credit [7].

[7] There's just one wrinkle that I've glossed over here. Advertisers are allowed to set up "conversion windows" which determine how far back to look for ads. Typically the lookback window for clicks will be set at 30 days and the window for impressions will be set at 7 days. Then the algorithm first looks for a click within the click window, giving 100% of credit to the most recent one. If there are no clicks, then the algorithm looks for an impression without a click but within the window, again giving the most recent one all the credit.

Surprisingly, when this approach was adopted it actually made sense and delivered meaningful results. Since then, the internet and internet marketing have changed, leading to a number of shortcomings of the last-ad model. Some of them include the following:

- Advertisers are spending much more online, meaning that users are reached more often than before. If someone is being reached only once, then the last-ad model is perfectly fine. If someone is being reached dozens of times, then the approximation that the last-ad model becomes untenable. Influence accretes and it seems myopic to give all credit to the most recent ad.

- There are a profusion of ways to advertise online. Ten years ago, the only way to advertise online was basically through display advertising. Today there are many channels including display, search, video and rich media (a form of interactive advertising online). These media fit at different places in the purchase cycle. In particular, search is typically found very near the bottom of the funnel and this allows search to take sole credit for a number of actions that involved multiple ads over the purchase funnel.

- Internet usage has increased and, along with it, ad consumption. The penetration of broadband access and the resulting increase in surfing has led to a hundred-fold growth of ads consumed. Since users are surfing more and receiving more ads the last-ad model has become increasingly happenstance. If two ads are shown just seconds apart, the binary nature of the last-ad model (all credit to one or the other of the ads) is capricious and overly-sharp rather than a smooth average of credit between the two ads as one would hope.

- As we shall see Chapter 3, the last-ad model fails to take into account important intermediate effects of advertising. The last ad model ties conversions to ad exposures, ignoring potential sources of influence such as frequency, creative type, interaction with rich media ads, and ad size.

For several years, dissatisfaction with the last-ad model has grown in the advertiser community. Although many do not complain about the attribution model directly, agency teams commonly make optimization decisions based on more than simply the conversion[8] data because they are aware of the biases inherent in the methodology. The next section will indicate how the conversion attribution models are evolving. It's undeniably odd to spend this much time setting up the last-ad straw man, but it's worthwhile to understand how far from state-of-the-art the business currently is.

2.2.2 An Emerging Standard: Engagement Mapping

A sea change is underway in online advertising conversion attribution. Given the shortcomings listed above, the industry has been interested in finding more comprehensive attribution algorithms. Atlas has, in fact, been a pioneer in the area with the creation of "Engagement Mapping" (E-Map). E-Map is a flexible framework that allows advertisers to define a custom conversion attribution model.

The E-Map model is defined by two sets of parameters. The first set of parameters defines the relative value of different ways of interacting with a cookie. These parameters (called base-weights) are positive real numbers. They are indexed to the weight of the most basic online impression—say, a standard JPEG or GIF display ad— which is set to have the value one. Then all other types of marketing have relative weights. Advertisers typically set the weight for text-link impressions, which are considered lower quality impressions, to a value around 0.1. (Setting the weight to any number less than 1 captures the idea that the value of the text-link is lower than the standard GIF image.) On the other hand, a view of 15 seconds of online video is considered a very high quality impression and this typically gets a weight between five and ten times greater than the reference weight. Finally, clicks are typically given

[8]One additional point of terminology. The word "action" is used to denote someone hitting a page that is tagged as in the description above. The term "conversion" means an action that has an impression or click beforehand so that an association can be made between the marketing and the user behavior on the website.

a high weight, perhaps 25 to 50 times greater than the reference weight. Again, all ways of interacting with a potential consumer online have their own weights [9].

The second set of parameters is designed to modify the base-weight set of parameters. For instance, there is a recency score that diminishes the weight as the length of time between the impression and the action grows. There is also an ad size score that allows advertisers to give greater weight to larger ads. These parameters are designed so that they range from 0 to 1 although at some point they may be adjusted to go above 1. In Section 2.3 we provide more detail on the form and implementation of the E-Map model.

Once a model has been defined, what is done with it? If we have an action (a consumer performing a desired action on an advertiser's webpage) and a set of impressions and clicks in the associated conversion window, a score can be determined for every event. This score is calculated based on the base-weight and then discounted by things like recency and ad size. Then these scores are normalized so that each impression and click has a number between 0 and 1 associated with it, the sum of the scores is 1 and each score is interpreted as the share of conversion credit that impression or click "deserves". Essentially, E-Map models allow the sharing of credit for a conversion between all of the ways someone is reached by advertising.

It is natural to ask why these models are used instead of something more sophisticated and, frankly, more statistical. The academic literature, for instance, does not employ last-ad or E-Map models as we will see in Chapter 4. The answer is that conversions are central to online businesses—they are the coin of the realm. As such, marketers are extremely reluctant to modify the way in which conversion credit is calculated. The E-Map model outlined above was chosen in conjunction with advertisers so that it could be transparent and include all aspects of the conversion process they deemed important. The marketing community is not, overall, particularly sophisticated with regard to statistical modeling. Switching the currency

[9]There's one important aspect of marketing that is being ignored by this approach: ad wear-out. The idea is that ads are more effective on initial viewing and that people get tired of seeing the same commercial or ad repeatedly. The phenomenon of ad wear-out is well-documented, if not well-understood in the context of online advertising. At this point E-Map models do not take into account ad wear-out.

of the business from the lousy Last-Ad standard to a highly complicated proportional hazards model, for instance, would require a level of trust in analytics not currently seen. The E-Map model is seen as a useful compromise in the appropriate direction.

2.3 E-Map Definition

Let us now spend a moment to firm up the definition of an Engagement Mapping model. Broadly speaking the E-Map model is a combination of base weights and variables that are applied to data to share conversion credit. In all there are 22 different base weights pertaining to different types of media. These media fall into broader categories such as text links, display, flash ads, rich media, and video[10]. These media are given weights. By convention the display weight is set to one and all other media are set relative to that. Higher weights indicate media that are estimated to be of greater importance in modifying conversion probability and should therefore receive a greater portion of the shared credit. Typically text links have a weight of around 0.1, Flash and Java have weights about the same as display, rich media ads tend to have weights from 2 to 4 and video ads are in the range of 5 to 10. Note that these choices are based purely on convention. Additionally, the ways that consumers interact with ads (clicks on most types of ads and interactions with rich media ads) carry their own weights. Click weights tend to be between 10 and 50. The data we will be working with do not have rich media interactions, so we will not concern ourselves with those weights.

At this point, let us imagine an individual who buys something online, receiving seven total exposures beforehand: two display ads, three flash ads, and 2 text links with clicks on the

[10]Text links are simple clickable text placed on web pages that can be used by consumers to navigate to an advertiser's website. Text links are also the typical advertisement consumer's see when they use search. Display advertising in this context refers to images served on web pages. These come in a variety of sizes although the most common pixel dimensions are 468 by 60, 728 by 90, 88 by 31 and 125 by 125. Flash ads are similar to display ads in size, but the underlying technology and customization are much richer for Flash and Java ads. Rich media is an omnibus term referring to ads that use advanced, interactive technology other than Flash and Java. Finally, video ads are either stand-alone or run adjacent to online video, similar to how commercials are used on television.

second one. Table 2.2 represents the absolute and relative scores for the various events. The

Event		
Type	Base Weight	Credit (%)
Flash	1.25	4.1
Flash	1.25	4.1
Display	1	3.2
Display	1	3.2
Text Link	0.1	0.3
Flash	1.25	4.1
Text Click	25	81
Total	30.85	100

Table 2.2: A simple example of credit sharing among six marketing exposures. Note the percentage credit is simply the normalized base weight value in this case.

key idea is that when exposures occur in a consumer's history, these base weights give us the ability to apportion credit based on simple rules.

As mentioned before, the E-Map model includes both base weights and what we are calling variables. These variables affect the scores of all base weights and, in the current incarnation, are related to the ad size, the time between the conversion and the exposure, and order in which clicks or other active events take place. All variables are constrained to be between 0 and 1 although the specific meaning of the value is variable-dependent. Assume that we have an exposure within our view conversion window (called win_v) with base weight b taking place at time t_e and a conversion at time t_i. Further assume that the exposure is an ad size of s_e. The reference ad size, against which other ads are measured, is s_r and varies by creative type. Then, if v_s and v_r are the parameters for ad size and recency respectively, the final score s for an exposure is as follows:

$$s = b \cdot \underbrace{\left(\left(v_s \cdot \frac{s_e}{s_r}\right) + (1 - v_s)\right)}_{\text{Size}} \cdot \underbrace{\left(\frac{t_i - t_e}{\text{win}_v}\right)^{v_r}}_{\text{Recency}} \tag{2.1}$$

In Equation 2.1, the first component is the raw base weight discussed above. The size com-

ponent gives us linear changes in the size effect. If $v_s = 1$ then an ad that is twice the size of the reference ad will receive twice the credit, whereas at $v_s = 0$ they will receive the same credit. The recency component of the score represents an exponential decay in the effect of ads that were more distant with the parameter v_r tuning the steepness of the decay. There is an additional variable, called the "order" variable, that determines how credit is shared among clicks or other active events. This is a relatively rare case; suffice it to say that clicks after the first one receive lower weights and the value of v_r determines the steepness of the drop-off.

Currently most advertisers using E-Map employ one of three default models, determined using data analysis and experience across many advertisers. The default models are called Brand, Balanced, and Direct Response (DR). The Brand model discounts recency and considers size very important. There is a large amount of spread in the base weights for passive ads and clicks are somewhat less important. The DR model considers clicks and recency to be very important (the click base weight is 50 and recency is set to 0.20) whereas size receives a medium value. The Balanced model essentially represents an averaging of the DR and Brand models.

This figure is a graphical representation of the E-Map scores for seven exposures using raw base weights, the modified scores based on the Brand model, and the modified scores based on the DR model. These actual events took place very close to each other in time so I have evenly spaced them across the x-axis for readability. The time of exposure and ad size (not shown) in conjunction with the model parameters cause large changes in the credit allocated to certain exposures. In particular, notice that the final click has more than twice the credit under the DR model compared to the brand model. This is due to the very high weight that the DR model gives to clicks.

As it currently stands, these 22 base weights and the three variables are decided upon in an *ad hoc* way. Our goals in these next sections are to formalize the ideas of E-Map as a model

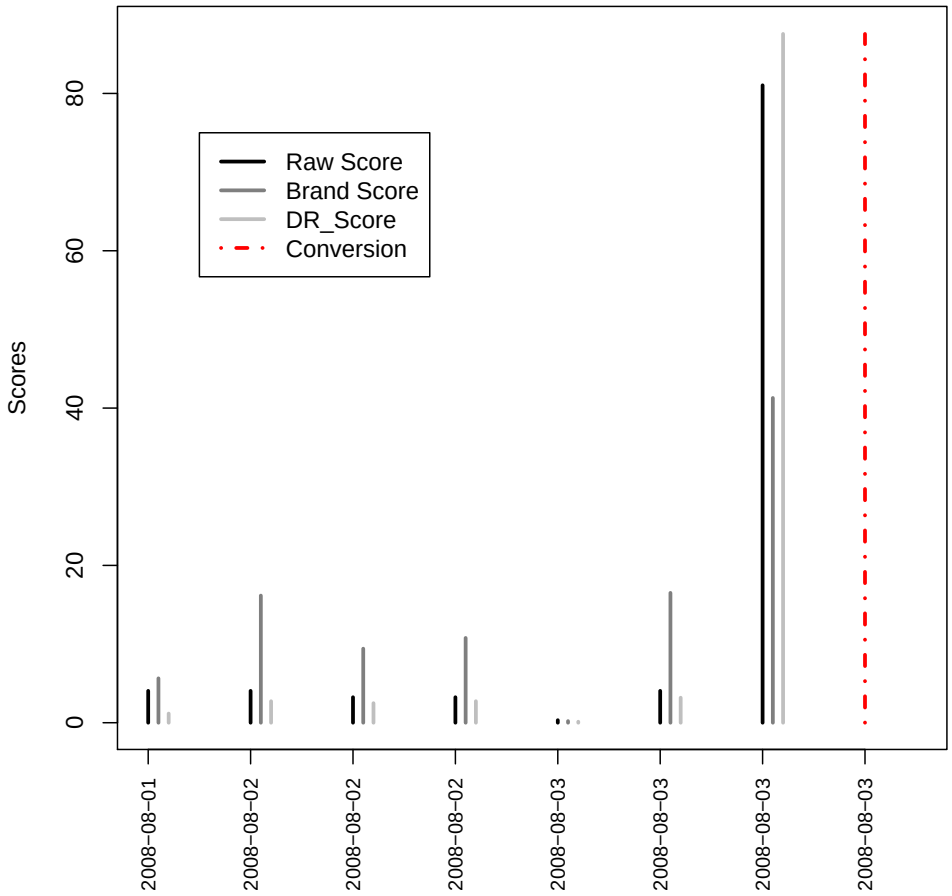

Figure 2.1: A graphical representation of the E-Map scores for seven exposures using raw base weights, the modified scores based on the Brand model, and the modified scores based on the DR model. The time of exposure (stylized for readability) and ad size (not shown) in conjunction with the model parameters cause large changes in the credit allocated to certain exposures. In particular, notice that the final click has more than twice the credit under the DR model compared to the brand model. The actual conversion is illustrated as a dash-dot red line.

and allow us to think about which model fits a given data set the best. For instance, for this particular cookie, how can we say whether the brand model parameters or the DR model parameters provide a better fit? That answer is to come.

Chapter 3

Academic Results

Advertising and marketing have a rich tradition in academic literature. The money spent on advertising annually (about $300 billion dollars in the United States and across all media in 2008 [14]) attracts researchers interested in understanding the consumer response to advertising. This response is measured across many different dimensions such as affect (the feeling or emotion created by advertising) or cognition (what advertising makes people think about a product or brand). A small subset of the overall literature focuses on actual purchases and the return-on-investment (ROI) from advertising. I have focused my research primarily in the *Journal of Marketing*, *Marketing Science*, the *Journal of Marketing Research* and the *Journal of Advertising Research*.

Television, dominating advertising expenditures for the last 40 years, sensibly occupies the lion's share of purchase modeling. Online advertising has existed only since about 1995 and therefore the research in this area is much more limited. On the other hand, the richness of the data explored above draws researchers like miners to a gold strike and the past ten years have seen an explosion of research into the consumption of online advertising and the sales resulting from such advertising. Our research follows this rich vein. In the next two sections we will

provide an overview of relevant research in both advertising in general and online advertising in particular. The next section will treat advertising broadly, detailing the models that have been developed and the methodological approaches taken. It will prove useful to embed online advertising research in this greater corpus. After discussing the overall advertising research we will provide an overview for response modeling in online advertising. There is a straightforward path in the past ten years, culminating in some relatively sophisticated and accurate models predicting response. Chapter 4 is an extension of the state-of-the-art described in the response modeling section.

3.1 General Advertising Research

To focus the discussion, I begin with the Vakratsas and Ambler paper providing a synthesis of more than 250 journal articles in an attempt to determine how advertising works[34]. The first (and arguably most useful) insight from the paper is a distillation of the framework for advertising study, reproduced in Figure 3.1. It is worthwhile to spend a moment understanding the ideas in this diagram as these concepts are foundational for the overall understanding of advertising models discussed. Advertising input is passed through various filters (based on audience and medium) and reaches the consumer. These advertising messages drive response: cognition, how the consumer thinks about the brand or product; affect, how the consumer feels about the brand or product; and experience, an interaction between the advertising and the user's previous experience with the brand or product. These in turn drive behavior. Virtually every piece of advertising research fits within this framework and, conceptually, papers are notable for the aspects they attempt to address. All research begins in the first box, Advertising Input. Input is typically measured via some "tonnage" of advertising, typically Gross Ratings Points (GRPS) (a measure of audience coverage) or gross impressions (a measure of advertising viewership). This is the explanatory variable of greatest interest. There are really several different levels of response variables. Many retrospective studies jump directly

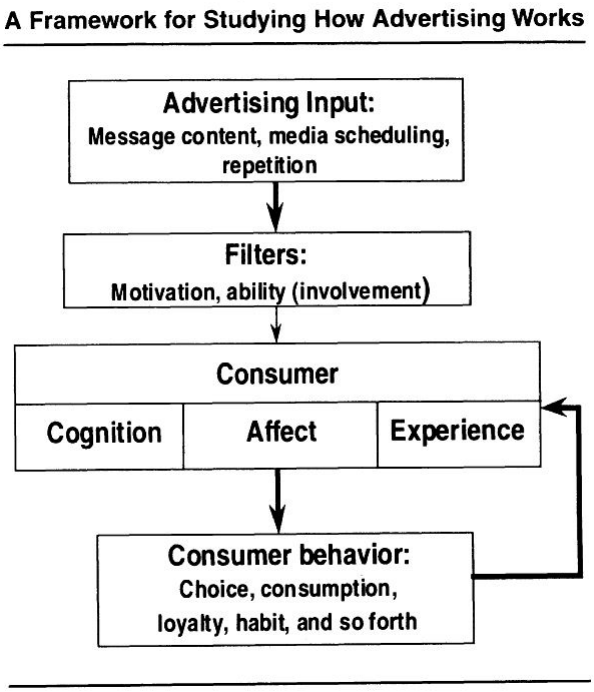

Figure 3.1: This figure, reproduced from Vakratsas and Ambler's paper [34], gives a framework for studying how advertising works. Advertising input is passed through various filters (based on audience and medium) and reaches the consumer. These advertising messages drive response: cognition, how the consumer thinks about the brand or product; affect, how the consumer feels about the brand or product; and experience, an interaction between the advertising and the user's previous experience with the brand or product. These in turn drive behavior.

to the final box, consumer behavior. If advertising works, then companies should see results in overall sales or profits. For offline media, if there is no survey component measuring cognition (C), affect (A) or experience (E), then sales are all that can be measured. Consumer surveys can be used at this level as well, measuring how advertising has affected cognitive response such as brand loyalty or product preference.

Vakratsas and Ambler build their article on empirical research that can be shoe-horned into this framework[1]. They go on to detail individual models that are created with the elements of their framework. Market response research, denoted "(-)" in their notation indicating the empty model, looks at how advertising inputs explain consumer behavior. This empty model posits no framework within which advertising works. Instead advertising is viewed as a black box into which a media buy is put and out of which comes new consumer behavior. Offline, this research tends to be aggregate (e.g., we bought this much TV in this market and this is how same-store sales changed) with the exception of certain data sets that require consumers to report purchases in some way. Online advertising data provides an unfettered view of consumer response and hence much research (and virtually all practical applications of the data) are market response models in this sense. In subsequent sections, this paper discusses a number of other types of models implied by existing research. (For instance, quite a bit of time is spent on a hierarchical model, called "(CA)" in their nomenclature, in which advertising influences cognition which in turn influences affect.)

One of the most intriguing models discussed is called (C)(E)(A), attempting to imply that advertising affects all three mental areas but that these areas are not organized into a hierarchy. While the notion of a hierarchy of effects is persuasive and attractive, the results of the authors' analysis of the extant research indicate that such a hierarchy is not supported in the literature. One of the most significant contributions from the paper are five "generalizations": consistent, objective conclusions that are supported (or not contradicted) throughout the research. The

[1]But this is a broad framework and it appears the only major body of work they do not consider are questions of how the overall social and economic climate affect advertising.

generalizations are quoted below:

G_1 Experience, affect, and cognition are three key intermediate advertising effects, and the omission of any one can lead to overestimation of the effect of the others.

G_2 Short-term advertising elasticities are small and decrease during the product life cycle.

G_3 In mature, frequently purchased packaged goods markets, returns to advertising diminish fast. A small frequency, therefore (one to three reminders per purchase cycle), is sufficient for advertising an established brand.

G_4 The concept of a space of intermediate effects is supported, but a hierarchy (sequence) is not.

G_5 Cognitive bias interferes with affect measurement.

Generalizations one and four (called G_1 and G_4) are particularly germane to my research. The principle articulated in G_1 supports the idea that a model of advertising effectiveness that fails to take into account experience, affect, and cognition, is likely to overestimate the effect of the others. In particular, the last-ad model seems to take into account the effect on cognition while ignoring touchpoints higher up the marketing funnel. In G_4 we see an indictment of a hierarchical model which, typically, places a causal chain on advertising. Advertising affects either cognition or affect or experience which in turn affects one of the other two attributes. Finally, consumer action is affected. If this hierarchy were true, it seems possible that a sufficiently intelligent last-ad model might have some hope of capturing advertising effect (by essentially acting as a tollbooth, capturing the journey to the action). The absence of this hierarchy undermines any case for last-ad effectiveness. When a single user history may contain both the viewing of a 30 second web video spot and a click on a paid search link, online advertising can affect all of cognition, experience, and affect. By giving all credit to a single advertisement, the last-ad model is insufficient to capture this richness.

The previous chapter discussed the E-Map and last-ad models for conversion attribution. How do they fit into the research framework? In some sense the last-ad model is an implementation of the null model, albeit a meticulous and flawed implementation. In the Vakratsas and Ambler framework, the empty model takes as input advertising delivery and tries to determine how that input affects consumer behavior without regard for any intermediate mechanisms. Whereas often the empty model is used with aggregate data, indicating the overall application of advertising and the community's response, the last-ad model measures both input and response at the individual level. The last-ad model is meticulous in this sense: measuring response at the user level. The most recent ad a consumer is exposed to within a certain window receives credit for the consumer behavior. No information about the ad, other than the binary variable indicating delivery, is used. Additionally, multiple ad exposures (the true measure of advertising input) are disregarded as all credit shifts to the last-ad. It is in this sense that the last-ad model is a flawed implementation of a null model, since we fail to account for all advertising input to the user. When empty models are applied to offline advertising, the totality of the advertising exposures are measured rather than just the most recent. To the extent that last-ad conversion attribution represents the empty model, it does a poor job capturing the ability of advertising to affect the three causal mechanisms outlined by Vakratsas and Ambler.

Engagement Mapping, on the other hand, is an attempt to locate the advertising message within the general cognition-experience-affect space, without regard to a hierarchy between them. For instance, ad size and duration has been shown to have an effect on both cognition and affect [30]. By including variables like ad size, Engagement Mapping takes into account additional variables other than exposure that determine the effect of advertising on the intermediate mental variables of cognition, affect, and experience.

A final generalization from this paper is notable. Generalization G_2 states that "short-term advertising elasticities are small and decrease during the product life cycle." In this context, elasticity refers to the concept that a given percentage change in advertising creates a smaller

percentage change in customer response. We call attention to this generalization to note that the effect we are trying to find by modeling advertising response is a small effect and varies by advertiser and product. There are many untracked advertising media and a myriad of reasons why a consumer responds or does not respond. Since we are incapable of measuring many of these covariates, we should expect that our models may explain only a small amount of the variance in behavior and attempts to predict when conversions actually happen will probably not be that accurate.

Hu, Lodish and Krieger have written a useful paper detailing their attempts to synthesize 241 TV advertising tests measuring effectiveness and estimating these elasticities [19]. This paper is a partial update of the Lodish et al. paper from 1995 [24] that analyzes 389 TV campaigns. These studies are quite different from those we will undertake in this research; my goal in mentioning the paper is to highlight the small effects they found with their model[2]. Overall, they found that advertising elasticities are between 0 and 0.2 although these were significantly different from 0. Their data are a result of two different pools of results. One set, from a product called Behavior Scan®, allows advertising input in a market to be compared with individual purchases scanned by panel members. The other type of data comes from matching two markets, one receiving advertising and the other not receiving. Essentially this is the analog of the medical-study idea of matching cases and controls but at a broader level. The data we will be working with are more similar to the Behavior Scan or IRI data, except we can also measure the consumption of individual advertising.

We now turn our attention from the general area of advertising effectiveness research into what I am calling response modeling: the effort to determine how internet users respond to advertising within that medium.

[2]Incidentally, the model Lodish et al. employ is the empty model from the Vakratsas and Ambler synthesis.

3.2 Response Modeling

When the problem is to determine the response of one variable to a host of additional explanatory variables, typically regression analysis is the answer. Advertising response research is no exception and the short history of modeling response closely mirrors the recent advances in regression modeling. As mentioned above, offline advertising measurement typically begins with the weight of advertising input and then some measure of either behavior (purchases, loyalty, etc.) or a measure of an intermediate effect such as attitudes toward a brand or message recall. Online advertising, with its richer data, allows for a much more direct modeling of the correlation between advertising and response[3]. The building of models for online advertising response begins in 1998 soon after advertising data started rolling in with the Ph.D. dissertation of Chatterjee [8]. This research in turn led to additional joint work in clickstream modeling in 2003 [7]. Clickstream modeling attempts to understand the drivers of clicks in a user's history. The term "clickstream" refers to the advertiser's view of their data (an incoming stream of clickers). In 2006 Manchanda, Dubé, Goh and Chintagunta [25] wrote a seminal paper, combining the basic ideas of Chatterjee, et al. [7] but using a much more sophisticated modeling approach: hierarchical Bayesian models. This section provides a short introduction to the academic research into response modeling. Chapter 4 builds on this tradition and represents an addition to the state-of-the-art.

The first papers, "Modeling the Clickstream: Implications for Web-Based Advertising Effects" [7] and the related dissertation by Chatterjee, are notable for being the first academic papers to provide an "empirical analysis of behavioral outcomes at the microlevel of *each ad exposure occasion*" (emphasis theirs). In other media, these data have been unavailable; in online these authors were the first to do the research. The authors create a model where the response variable is clicks on ads and they study the following effects: the effect of repeated exposures

[3]With very few exceptions, advertising studies "in the field" are not experiments. As the old statistical saw goes, there is no causation without manipulation and the lack of experiments in advertising hamstrings us to talk about concepts like correlation between spend and response instead of talking about causality. Wherever actual experiments have been done, I will take care to highlight them in the text.

to ads; consumer click proneness; consumer heterogeneity with regard to click rates; the effect of inter-visit time on click proneness; and the effect of navigation path. Their emphasis on modeling at the level of the advertising impression, what they call the exposure occasion, is notable because it is only within that context that we can begin to model the actual advertising mechanism. Ads are not delivered to markets or households, they are delivered to people who do or do not respond to them. An additional finding from this research, supported throughout the literature, is the heterogeneity of consumers. The same advertising delivered within the same context to different people will result in very different response. As such, any model that does not include consumer-level terms is bound to have large error; a model that estimates this heterogeneity will see a large amount of variance explained by the consumer-level parameters. While these qualities are strong, this research has shortcomings. First, as the title proclaims, clicks are being modeled. At best clicks are considered an intermediate response in the medium: consumers exposed to ads have the opportunity to click in order to navigate from the publisher's page to the advertiser's page. It is not, however, those clicks that are the desired result. Purchases, which affect the advertiser's bottom line, are the preëminent response variable. Therefore, Chatterjee, et al. are introducing an undesirable level of misdirection into their analysis. Briggs [5, 6] makes a cogent case that focus on clicks may actually distract advertisers from their principal goal of sales or registrations. Additionally, a recent study by Starcom (an advertising agency), Tacoda (an advertising network), and Comscore (a provider of internet data) casts considerable doubt on the value of clickers and the reasons why they click [20]. This research focuses on an intermediate outcome of dubious value and the model employed is a bit more cumbersome and less flexible than the hierarchical Bayesian framework used in subsequent research. This body of work is laudable for tackling the data opportunities head-on, but is, I think, quickly surpassed by the work of Manchanda, et al [25].

The work of Allenby in several different contexts has laid the foundation for the application of hierarchical Bayesian modeling to consumer response. Although he begins his exploration

of modeling response in a Bayesian context as early as 1990, the first paper relevant to our research is "On the Heterogeneity of Demand"[1], in which the authors call into question the supposition, long-held in marketing practice, that one can usefully think of segments of consumers as homogeneous. This research is based on an attempt to model segment behavior as a mixture of multivariate normals and finds that the within-component heterogeneity is substantial and unaccounted for otherwise. The data are based on offline consumer preference data for outboard motors, ketchup and tuna, the first time those three items have ever appeared together in literature, we suppose. The first reference we see to the use of hierarchical Bayesian modeling to consumer purchase (as opposed to descriptive concepts like consumer heterogeneity) is the 1999 paper modeling purchase timing [2]. Notably, this paper was published in the *Journal of the American Statistical Association*, marking the integration of the marketing concepts with the statistical concepts at a time when computer-intensive methods such as Markov chain Monte Carlo (MCMC) were first gaining wide currency in applied statistics. The application to theoretical statistics happened about a decade earlier [12]. Ultimately all of this work led to the 2003 paper [28] and 2006 book [29] (uncreatively having the same name!). Allenby's joint work with Rossi, a basic introduction to and marketing applications of Bayesian statistics and decision theory, is a useful reference for a non-statistical audience. They develop the theory of both maximum likelihood and Bayesian estimators in the marketing context.

The highest refinement of these ideas is the 2006 paper by Manchanda et al. [25], synthesizing many of the ideas discussed in the preceding paragraphs. This research models purchases instead of clicks, an improvement on the Chatterjee, et al. work. Additionally, rather than using the more traditional logit-based model, Manchanda, et al. use a proportional hazard model. Hazard models, typically seen in survivor analysis, are natural choices for data that are censored in some way. Consumer purchase data conforms to the censored data framework since, at the time data collection ends, some future purchasers likely remain in the data classified as non-purchasers. The proportional hazard model is essentially a synthesis of this hazard

model with the logit transform and used to model probabilities of conversion. The authors found positive advertising effects for number of exposures, websites visited, and number of pages visited (a proxy for surfing activity). There was a negative effect associated with the number of different ad message types that were seen. Additionally, they were able to establish a difference in the customer response for new customers versus repeat customers. Chapter 4 extends this work. I introduce time-varying covariates to model the likelihood of conversion. Manchanda, et al. also restricted their analysis to estimating inter-purchase times using summary-level data for the advertiser. I believe modeling individual exposures (as is possible with the time-varying covariates) is a novel and informative addition to our understanding of response.

Chapter 4

Finding Drivers of Conversions with Hazard Models

This chapter estimates the impact of advertising on conversion probabilities using proportional hazard models (PHM). Within this framework we will see it is possible to nearly fit the exact E-Map function in a highly rigorous way.

A brief note on Microsoft Advertising's history of this sort of conversion modeling is warranted. When Engagement Mapping was first being developed, in early 2006, we cast about for a set of models that would allow us to determine the drivers of conversions. Initially we settled on logistic regression, only to abandon it when we found the issue of multiple records per cookie to be insurmountable. As we have discussed, in our data it is commonplace for one cookie to have hundreds of records for an advertiser and another cookie to have only one or two. The first solution we explored was collapsing all of the records into a summary row. Unfortunately, this approach eliminates order and time-dependence, making recency almost impossible to estimate. Attributes that vary at the ad level (e.g. size) are lost. The second approach is to create replicates of exposure-level columns going out five or ten exposures.

Unfortunately this truncates the cookie's data and the records that are included could span only a few seconds when the cookie's entire data set might span months. As far as I know, there is still no good solution for these problems within logistic regression. Fundamentally, the summary data that would be required for logistic regression, where a cookie is an observation and a record summarizes that entire cookie's history, reduces the available information too much for accurate modeling.

4.1 Proportional Hazard Models

We first describe the basic PHM framework, then discuss the time-varying covariate addition. PHMs are well-described in the literature, with the foundational work being in Cox's *Analysis of Survival Data* [10] with algorithms for R/S-Plus found in Therneau and Grambsch's *Modeling Survival Data* [32]. It seems as though the modern applied work in the area, particularly the work that uses the excellent `survival` package from R/S-Plus, is based on Therneau and Grambsch. A useful shorter overview is found in the comprehensive regression book by Harrell [15]. Smith [31] has a good overview of basic survival analysis. I used Hosmer [18] as a reference, but the treatment there lacks the clarity of these others.

Let T denote survival time where "death" in this context is defined to be a conversion with a cumulative distribution function of $F(t) = \Pr(T < t)$. Now define $S(t) = 1 - F(t) = \Pr(T \geq t)$. This is the survival function: the probability of a cookie remaining a non-converter at least as long as t. Note that $S(0) = 1$ as there are no instantaneous conversions. (This matches the business logic—the definition of a "conversion" is an action preceded by advertising.) We next define the hazard function. In plain terms the hazard function, $\lambda(t)$, is the instantaneous rate of conversion. More formally,

$$\lambda(t) = \lim_{\delta \to 0} \frac{\Pr(t < T \leq t + \delta | T > t)}{\delta}.$$

We can derive an identity that is necessary in Chapter 5 by using the rules of conditional probability.

$$\begin{aligned}
\lambda(t) &= \lim_{\delta \to 0} \frac{\Pr(t < T \leq t + \delta | T > t)}{\delta} && (4.1)\\
&= \lim_{\delta \to 0} \frac{\Pr(t < T \leq t + \delta) / \Pr(T > t)}{\delta} && (4.2)\\
&= \lim_{\delta \to 0} \frac{(F(t + \delta) - F(t))/S(t)}{\delta} && (4.3)\\
&= \frac{dF/dt}{S(t)} && (4.4)\\
&= \frac{f(t)}{S(t)} && (4.5)
\end{aligned}$$

where $f(t)$ is the density function of T and,

$$\frac{dS(t)}{dt} = \frac{d(1 - F(t))}{dt} = -f(t).$$

Therefore, since $\frac{d(\log g(t))}{dt} = \frac{g'(t)}{g(t)}$, where g is differentiable, we have the following identity:

$$\lambda(t) = \frac{-\partial \log S(t)}{\partial t}$$

or

$$\begin{aligned}
S(t) &= \exp[-\Lambda(t)] && (4.6)\\
&= \exp\left(-\int_0^t \lambda(u) du\right). && (4.7)
\end{aligned}$$

I take a brief digression detailing how the baseline survival function is estimated. Although this is not strictly needed for the proportional hazard models that form the bulk of this chapter, one of my goals is to estimate the probability that a given cookie will convert. To do

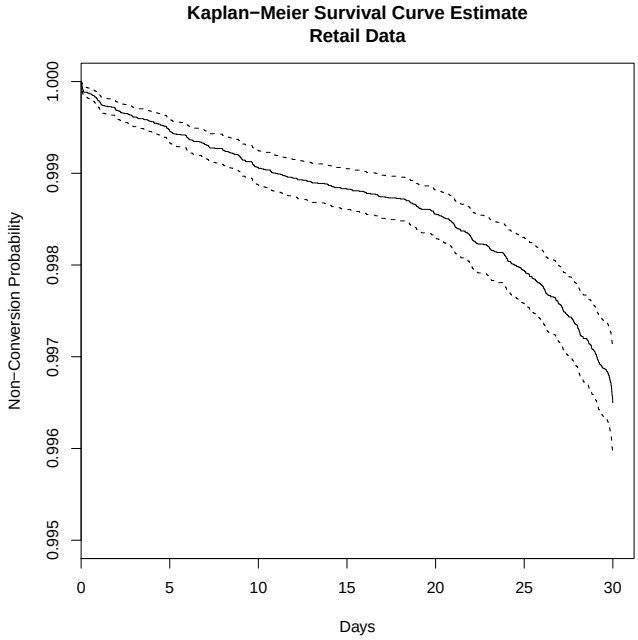

Figure 4.1: An illustration of the Kaplan-Meier survival function estimate for the retail data set. The sampling in the data set resulted in about 0.5% of cookies being converters. Consequently, after 30 days the probability of conversion is approximately 0.5%. For this advertiser, a number of cookies convert more than 15 days from their initial exposure, leading to a surprising drop in the curve from day 15 onwards.

this, $S(t)$ must be estimated. We use the Kaplan-Meier survival function estimate

$$\hat{S}(t) = \prod_{t_i \leq t} \frac{n_i - d_i}{n_i} \tag{4.8}$$

where n_i is the number at risk of conversion at time t_i and d_i are the number of conversions at t_i. Throughout this chapter I illustrate concepts with the data from a retail advertiser. The retail data set contains 1.3 million records across 108,392 cookies. The data belong to an advertiser selling products online using search and display advertising. I will describe the data in more detail as it becomes necessary for the exposition. In Figure 4.1 we see the Kaplan-Meier survival estimate for the retailer data.

This figure illustrates the Kaplan-Meier survival function estimate for the retail data set. The

full potential data set is too large to be practical (terabytes of data over several months). This data is sampled taking all converters and a subset of non-converters, resulting in about 0.5% of cookies being converters. Thus, after 30 days the probability of non-conversion from the KM estimate is approximately 99.5%. For this advertiser, a number of cookies convert more than 15 days from their initial exposure, leading to a surprising drop in the curve from day 15 onwards. Given a cookie with no other covariate information, this curve would give us an estimate of the probability of conversion at any time $t \leq 30$.

The Cox Proportional Hazard Model (PHM) formulates the hazard function as

$$\lambda(t, \mathbf{X}|\beta) = \lambda_0(t) \exp(\mathbf{X}\beta) \tag{4.9}$$

where $\mathbf{X}$ is our data matrix and β is the vector of coefficients. The model is semi-parametric because, while the covariates enter the model through a series of parameters, the baseline hazard, $\lambda_0(t)$, is left undefined. The hazard function is defined for all $t \geq 0$. Let X_{ji} denote covariate i for observation j. This model is called the proportional hazard model because, if two subjects differ with respect to a single covariate, $(X_{\cdot i})$, then the ratio of their hazards has a simple form:

$$\frac{\lambda(t|\mathbf{X_1})}{\lambda(t|\mathbf{X_2})} = \frac{\lambda_0(t) \exp(\mathbf{X_1}\beta)}{\lambda_0(t) \exp(\mathbf{X_2}\beta)} \tag{4.10}$$

$$= \frac{\exp(X_{1i}\beta_i)}{\exp(X_{2i}\beta_i)} \tag{4.11}$$

$$= \exp\{\beta_i(X_{1i} - X_{2i})\}. \tag{4.12}$$

In words, the hazard for subject 1 is $\exp\{\beta_i((X_{1i} - X_{2i})\}$ greater than subject 2. This parallels the role of coefficients in a logistic regression model. In logistic regression, if two subjects differ in one covariate then the odds for subject 1 is the odds for subject 2 multiplied by the anti-log of the cofficient times the covariate difference. Here the hazard ratio is simply the anti-log of the coefficient of interest times the difference in the measurements on the subject. Note

that this hazard ratio is independent of time—this is where the term "proportional hazard" comes from—and this is nearly exactly what we require for the base weights derived from the creative type covariate in E-Map. Engagement Mapping is founded on the premise that exposure to and interactions with marketing multiply the probability of conversion by some factor, independently of time except for recency and order. We will see below that these two covariates, which *do* vary with time, require special treatment in the proportional hazard model framework.

Equation 4.6 gives us the relationship between the survival function and the cumulative hazard function. If we integrate Equation 4.9 then we have

$$\int_0^t \lambda(u)du = \int_0^t \lambda_0(u)\exp(\mathbf{X}\beta)du \tag{4.13}$$

$$= \Lambda_0(t)\cdot\exp(\mathbf{X}\beta) \tag{4.14}$$

We can now substitute the result 4.14 (Cox formulation) into Equation 4.6, yielding

$$S(t) = \exp[-\Lambda(t)] \tag{4.15}$$

$$= \exp[-\Lambda_0(t)\cdot\exp(\mathbf{X}\beta)] \tag{4.16}$$

$$= [S_0(t)]^{\exp(\mathbf{X}\beta)} \tag{4.17}$$

where $S_0(t) = e^{-\Lambda_0(t)}$ is the baseline survival function.

We now discuss how to estimate Equation 4.17. Let $(t_i, \mathbf{X_i}, c_i)$ be a triplet of time, covariate vector and censoring variable where $c_i = 1$ if an observation converts at time t_i and $c_i = 0$ if an observation does not convert at t_i. In this case we assume that $i \in (1, 2, \ldots, n)$ where there are n cookies that form our observations[1]. For any observation that fails at t_i, the contribution to the likelihood function is defined by $f(t_i, \mathbf{X_i}|\beta)$, the density of the failure distribution. An observation that survives at least as long as t_i contributes $S(t_i, \mathbf{X_i}|\beta)$ to the

[1]Below we make the extension to time-varying covariates where we will have $N = \sum n_i$ total records with each cookie having n_i records in the data set

likelihood function. We can concisely write the total contribution as

$$[f(t_i, \mathbf{X_i}|\beta)]^{c_i} \cdot [S(t_i, \mathbf{X_i}|\beta)]^{1-c_i} \tag{4.18}$$

We assume independence between the observations and can therefore combine our observations to form the traditional likelihood and log-likelihood:

$$L(\beta) = \prod_{i=1}^{n} \left\{ [f(t_i, \mathbf{X_i}|\beta)]^{c_i} \cdot [S(t_i, \mathbf{X_i}|\beta)]^{1-c_i} \right\} \tag{4.19}$$

$$l(\beta) = \sum_{i=1}^{n} \{c_i \ln [f(t_i, \mathbf{X_i}|\beta)] + (1 - c_i) \ln [S(t_i, \mathbf{X_i}|\beta)]\} \tag{4.20}$$

where the product and sum are taken over all cookies in the data. From Equation 4.5 we know that $f(t, \mathbf{X}|\beta) = \lambda(t, \mathbf{X}|\beta) \cdot S(t, \mathbf{X}|\beta)$ and we can substitute this expression for the failure PDF into Equation 4.20 and use the parametrization for the proportional hazard model from Equation 4.16 to get

$$l(\beta) = \sum_{i=1}^{n} \left\{ c_i \ln [\lambda_0(t_i)] + c_i \mathbf{X_i}\beta + e^{\mathbf{X_i}\beta} \ln [S_0(t_i)] \right\} \tag{4.21}$$

It is not possible to maximize Equation 4.21 as written. To do so would require maximization with respect to the baseline hazard function (and baseline survival function derived from this hazard function) as well as the censoring times.

Instead we proceed in a different direction. Let R_i denote the number of cookies at risk of conversion at time t_i[2]. We wish to determine the conditional probability that cookie i converts

[2]Technically the cookies in R_i are at risk at time $t_i - \epsilon$.

at t_i given that cookie i is in R_i and given that there is exactly one conversion[3] at time t_i.

$$\begin{aligned}&\Pr(\text{cookie } i \text{ converts at } t_i | R_i \text{ and one failure at } t_i) \\ &\qquad = \frac{\Pr(\text{subject } i \text{ converts at } t_i | R_i)}{\Pr(\text{one conversion at } t_i | R_i)}. \end{aligned} \tag{4.22}$$

The quantity in the numerator of Equation 4.22 is proportional to $\lambda(t_i, \mathbf{X_i}|\beta)$. Similarly, the denominator is proportional to the sum of the individual hazards of every cookie in the risk set: $\sum_{j \in R_i} \lambda(t_i, \mathbf{X_j}|\beta)$. There are a few subtle points about this summation. We index the terms of the sum by $j \in R_i$, where the index on R indicates that we considering the risk set at time t_i. Similarly, this is why we evaluate $\lambda(\cdot)$ at t_i but for covariate vector $\mathbf{X_j}$. Building on these observations and Equation 4.22,

$$\frac{\Pr(\text{subject } i \text{ converts at } t_i | R_i)}{\Pr(\text{one conversion at } t_i | R_i)} \approx \frac{\lambda(t_i, \mathbf{X_i}|\beta)}{\sum_{j \in R_i} \lambda(t_i, \mathbf{X_j}|\beta)} \tag{4.23}$$

$$= \frac{\lambda_0(t_i) \exp(\mathbf{X_i}\beta)}{\sum_{j \in R_i} \lambda_0(t_i) \exp(\mathbf{X_j}\beta)} \tag{4.24}$$

$$= \frac{\exp(\mathbf{X_i}\beta)}{\sum_{j \in R_i} \exp(\mathbf{X_j}\beta)} \tag{4.25}$$

$$\tag{4.26}$$

From here we can extend to a likelihood taking into account all observations,

$$l_p(\beta) = \prod_{i=1}^{n} \left[\frac{\exp(\mathbf{X_i}\beta)}{\sum_{j \in R_i} \exp(\mathbf{X_j}\beta)} \right]^{c_i} \tag{4.27}$$

$$= \prod_{i=1}^{m} \frac{\exp(\mathbf{X_i}\beta)}{\sum_{j \in R_i} \exp(\mathbf{X_j}\beta)} \tag{4.28}$$

Equation 4.27 takes the product over all observations, assuming independence of the observations. The second step (4.28) simply excludes all terms where $c_i = 0$. This final expression

[3]Exactly one conversion at time t_i assumes that there are no ties in the data set. The times in our data are resolved to within a second, so ties are infrequent. In the literature, a great deal of energy is expended dealing with ties. Throughout this chapter when ties arise we use the Efron estimator for the risk values. This is the default ties-method in the software R/S-Plus.

forms the *Cox partial likelihood*. Cox speculated in 1972 that the value of β that maximizes this expression would have the same distributional properties of a full likelihood solution, rigorous proof of this came later (see Andersen et al., [26]).

The partial likelihood is analogous to the KM survival function estimator seen above (Equation 4.8). There we looked at the proportion of the data set that converted at time t_i compared to the total number at risk of conversion. The calculation is similar here. The conventional likelihood function gives the likelihood of a given data set as a function of the parameter vector. Maximizing $l_p(\beta)$ gives us a plausible estimator for our coefficients and Cox [9] showed that the usual large sample properties of likelihood estimators apply to the partial likelihood estimators. The estimators are asymptotically normal, and their variance found in the estimate of the information matrix (used during maximization). Rigorous proof of the properties of the partial likelihood came in Andersen, et al. [26]. As is standard practice, the log-partial-likelihood is maximized in the software implementation.

So far we have not touched on the unique feature of PHMs that modeling of cookies with vastly different numbers of records. Recall, first, the point of this exercise. Our cookies' have such extreme variation in histories (number of records, length of exposure, patterns of behavior, etc.) that summarizing the data distorts our understanding of the drivers of conversions. Thus far in the discussion, I have treated each cookie as an observation with n total cookies. I now split each cookie into n_i separate records—one for each ad impression, ad click, or conversion. Let $N = \sum n_i$ be the total number of records in the full data set. If we assume that we have p covariates (which will be specified below), then for every cookie, we form a $n_i \times (p + 3)$ matrix. In addition to the p covariates we have the start and stop times and a conversion indicator column. For every cookie we renormalize the times so that the first start time is 0 and this is the time of the first media exposure[4]. Let $0 = e_1, e_2, \ldots, e_{n_i}$ be the event times for cookie i. The first time is set to 0, the subsequent times are the elapsed time after that

[4]In practice we require the first exposure for the consumer to fall within our consideration period to avoid the confounding effect of exposures before the beginning of data collection

event. The first start-stop time interval is $(0, e_2]$, the second is $(e_2, e_3]$ and the final interval is $(e_{n_i}, t_i]$. The half-open intervals are intentional and important. Say event one is a display impression without a click. From that moment until the next event the cookie is "under the influence" of that impression and in the risk-of-conversion set for all cookies that have one display impression without a click. The final time interval either ends at a conversion or the end of the data collection period.

There is a side-effect of the time-varying covariates that requires consideration. When cookie i generates n_i records with start and stop times that partition the lifetime of cookie i, each one of those records enters the model, and the partial-likelihood, independently. To give an example, if record 1 is a display impression and record 2 is a search click at day 4, then two records are created. The first record has a start of 0 and a stop at 4. The second record has a start at 4 and some stop time greater than 4. These two records, in the course of formation of the partial likelihood, are treated independently and we do not make use of the fact that they were from the same cookie. In the discussion of the time-varying covariate data structure, a quotation from the seminal work on this topic, Therneau and Grambsch [32], is germane. On page 70, they discuss the splitting of a single summary record into a set of three records. The analog in our case would be creating three records for a cookie who had received three impressions.

> One concern that often arises is that observations 2 and 3 are "correlated," and would thus not be handled by standard methods. This is not actually an issue. The internal computations for a Cox model have a term for each unique death or event time; a given term involves sums over those observations that are available or "at risk" at the select event date. Since the intervals for a particular subject, "Jones" say, do not overlap (assuming of course that Jones does not have a time machine, and *could* meet himself on the street), any given internal sum will involve at most one of the observations that represent Mr. Jones; that is, the sum

> will still be over a set of independent observations. For time-dependent covariates, the use of (start,stop] intervals is just a mechanism, a trick almost, that allows the program to select the correct x values for Jones at a given time.

In other words, as long as the data simply record what has happened most recently to a cookie, correlation between records is not a concern. This benefit has a drawback, however. Take, for example, the cookie above with an impression at day 4. In the calculation of the partial likelihood, this record represents the cookie completely for all calculations done after day 4. If the cookie had one impression beforehand or 100, this record on day 4 appears the same in the calculations. This is a problem, because I wish to model cumulative effects of advertising. This requires the introduction of new covariates that span multiple records. In particular, I introduce two covariates (the one dealing with recency and the one dealing with previous clicks) that model the history the cookie has experienced up to a certain point. For instance, the previous click covariate is the number of previous records in a cookie's history that are clicks. Clearly the existence of a previous click value of, say, 4 requires previous click values of 0, 1, 2, and 3.

In the calculation of the partial likelihood, we ignore the covariance structure between the n_i records originating from the same cookie. Ignoring this structure produces an observed partial-likelihood that is actually a first-order approximation to the correct partial likelihood. The effect of this covariance structure on our estimators ($\hat{\beta}$) should be negligible and the estimators themselves should be reliable. What is less unreliable, at least without further research, are the estimators of the standard errors based on the observed information matrix. In the case of multivariate linear regression, correlation between covariates inflates the standard error estimates. (See for example Fox [11], page 120-121.) Symmetrically, failing to model correlation between covariates creates overly optimistic (small) standard errors. My assumption is that the same result holds for proportional hazard models and that the standard errors discussed in Subsection 4.4.1 are too small. On the other hand, the volume of data available

for these analyses should diminish concern about parameter estimation uncertainty.

By partitioning the time domain in this fashion, the time-varying covariate PHM has sidestepped the issue that holds back logistic regression. Each cookie exists in a given state over a period of time and is in the risk set over the same period of time.

4.2 A Click Only Model

Instead of diving deeper into the counting process that underpins the theory, we turn to a simple applied example. I work with the retail data set and one covariate predicting conversions. Note that in this section I am using summary data at the cookie level. For simplicity, I am not using the time-varying covariates in this section. Of the 108 thousand cookies, 0.5% convert and 0.7% of cookies have any clicks. For cookies that have clicks, 13% are converters. Table 4.2 holds the information regarding the distribution of clicks in this data set.

Clicks	0	1	2	3	4	5	6	7	8	11
Frequency	107630	601	99	31	12	5	5	3	4	2
%	99	1	0	0	0	0	0	0	0	0

Table 4.1: Distribution of clicks in the retail data set.

The click-only model is

$$\lambda(t) = \lambda_0(t) \cdot \exp\left(\beta_{cl} \cdot X_{cl}\right). \tag{4.29}$$

where X_{cl} is the count of clicks in the user's history. Referring to Table 4.2, there will be 107630 cookies with a value of 0 for X_{cl}, 601 cookies with a value of 1, 99 with a value of 3, et cetera. Fitting the model, I obtain a parameter estimate for β_{cl} of 0.64 (standard error = 0.02). Therefore each additional click increases the relative odds ratio of a conversion by $\exp(\beta_{cl}) = \exp(0.64) = 1.909$. The 95% confidence interval for the exponentiated parameter, formed using asymptotic results via the information matrix, is (1.823, 1.998). We are highly

confident that the presence of additional clicks increases the probability of conversion for this data set. (As we will see, the click effect is likely *much* stronger than this—something we will only learn with the addition of more covariates.) This model does not require the use of time-varying covariates, as we are simply using the total number of clicks the cookie has. As such, this model does not run afoul of the considerations of modeling the covariance structure discussed in the previous section.

In Figure 4.2 we see the estimated survival function for three hypothetical cookies based on the click-only model. The highest curve, with an estimated non-conversion rate of 99.9%, is for a cookie with zero clicks. The next curve is for a cookie with 1 click. The lowest curve, with an estimated non-conversion rate of 97.46%, estimates the conversion probabilities by time for a cookie with two clicks. The general shape of the curve closely follows the Kaplan-Meier estimate for the baseline hazard function seen in Figure 4.1.

There is an apparent contradiction to resolve between this model and the results from the literature review suggesting that clicks are an overrated measure of media performance. The answer to resolving the contradiction lies in the term "overrated". Although clicks have an important role in the conversion path (as seen in the results of this section), clicks do not account for all of the variability in conversions, as we will see in the coming section. Therefore, we recommend that practitioners use both clicks and other covariates in assessing drivers of conversions and in optimizing their media.

4.3 Data for PHM with Time-varying Covariates

If cookie i has n_i events or records at times $0 = e_1, e_2, \ldots, e_{n_i}$, we produce a data set with n_i rows containing the start time, stop time, conversion status, and the covariates we need to estimate an E-Map model. For each ad impression, the data are the creative type (Flash, Display, Text, etc.) and the creative size relative to the default size for that creative type. The

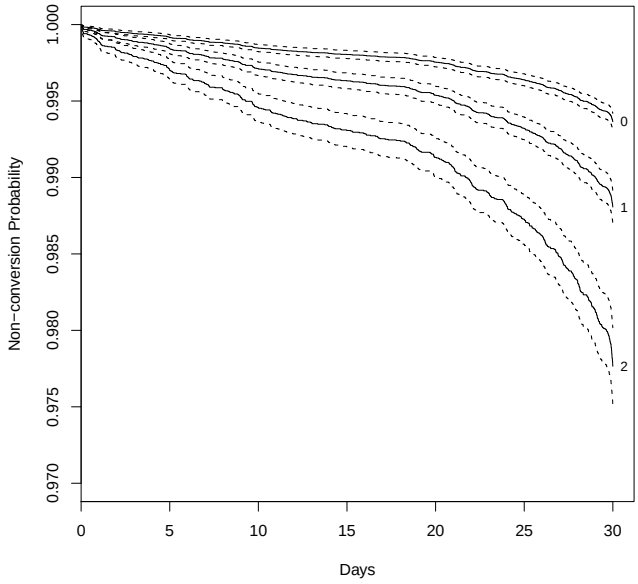

Figure 4.2: Estimated survival function for three hypothetical cookies based on the click-only model. The highest curve, with an estimated non-conversion rate of 99.9%, is for a cookie with zero clicks. The next curve is for a cookie with 1 click. The lowest curve, with an estimated non-conversion rate of 97.46%, estimates the conversion probabilities by time for a cookie with two clicks. The general shape of the curve closely follows the Kaplan-Meier estimate for the baseline hazard function seen in Figure 4.1. The curves are labeled with the number of clicks the hypothetical cookie has.

data must include an indicator of whether the ad was clicked on and, if rich media interactions are being tracked, the type of interaction.

Recall the definition of baseweights from Section 2.3: when credit is shared between exposures, the baseweight is the foundation from which we allocate credit. (The baseweight is then modified based on recency, size, and order.) Baseweights are evaluated against each other based on the question, How much more valuable is an exposure of type A versus one of type B? The relative power at influencing conversions will be expressed as the ratio of these two values—the same interpretation we give to our creative-type base weights. In a PHM, the coefficients for the creative types will become the base weights by exponentiation. If a given creative type has a model coefficient of β_t, then $\exp(\beta_t)$ is the modification to the hazard function by the exposure. That is, it is the odds ratio when comparing the probabilities of conversion. The quantity $\exp(\beta_t)$ also tells us how much an exposure changes the probability of conversion. If we have two cookies with identical histories except for one flash impression and we let β_f be the coefficient of the flash indicator variable in the model, then the probability of conversion for the cookie that receives the flash ad will be increased by a multiplicative power of $\exp(\beta_f)$. The click base weight will have some similar coefficient of the form $\exp(\beta_c)$. The click value will work in exactly the same fashion. Clicks on different types of creatives tend to have very different effects. Text link clicks are often on search ads and many of these are navigational in nature. As such we would expect their value in increasing conversion probability to be smaller than a click on display or flash. Therefore wherever possible we should estimate the interaction effect between clicks and creative types. (In some cases this is not possible because such a disproportionate share of clicks will be on one creative type, typically text links.)

The interpretation of the creative size ratio seems to be very different, although the conclusion ends up being similar. Since creative size is expected to follow diminishing returns and varies over an exceptionally large range (from a ratio of 0.01 all the way up to 10 or higher) we model the log of the size ratio. Let β_s be the coefficient we estimate in our PHM for the log

of the size ratio, a variable we call X_s. Note that this means that the actual size ratio is e^{X_s}. Also assume we are going to compare two cookies whose histories are identical in every respect except that they have one pair of ads where the difference in the log size ratios is X_s. Equation 4.10 tells us that the probability that cookie i converts relative to cookie j is $\exp(\beta_s X_s)$. As we saw back in Equation 2.1, the Microsoft Advertising use of the size variable is complicated. If we translate the size portion of Equation 2.1 using this notation, the new formula for the size multiplying factor is

$$v_s \cdot \exp(X_s) + (1 - v_s) = 1 + v_s \cdot (\exp(X_s) - 1) \tag{4.30}$$

where X_s is the size ratio. (Note that the E-Map model uses size ratio directly, not the log we use in our modeling, hence the factor of $\exp(X_s)$ in the above equation.)

Let us take stock. With the PHM formulation, we introduce a term for creative size into the model. We propose to introduce X_s into Equation 2.1 directly for the $\frac{s_e}{s_r}$ term. The remaining question, however, is that if we proceed as in Equation 4.30, what should we set the size variable v_s to? Our next paragraph attempts to settle that question.

Our goal is to try to match the influence that $\exp(\beta_s X_s)$ has on the proportional hazard model to the influence that v_s has to the E-Map model. In other words, we want these values to produce the same change in conversion probability. Thus we begin our search for a v_s value by setting Equations 4.30 and $\exp(\beta_s X_s)$ equal and solving for v_s, the size variable.

$$1 + v_s \cdot (\exp(X_s) - 1) = \exp(\beta_s X_s) \tag{4.31}$$

$$v_s \cdot (\exp(X_s) - 1) = \exp(\beta_s X_s) - 1 \tag{4.32}$$

$$v_s = \frac{e^{\beta_s X_s} - 1}{e^{X_s} - 1} \tag{4.33}$$

We wish for our single number summary (v_s) to be equal to a function of the log size ratio, X_s. Clearly this is impossible since the function is not constant. Notice, however, that creative

size ratios tend to be close to 1 (after all, the reference size is one of the most common sizes). Typically, then, the log of the size ratio is often 0. Therefore, what value of v_s agrees with the influence of β at $X_s = 0$? We cannot plug in that value since we get an undefined fraction of 0/0. We can however, use L'Hospital's rule (!) and determine

$$v_s = \lim_{X_s \to 0} \frac{e^{\beta_s X_s} - 1}{e^{X_s} - 1} \overset{L.H.}{=} \lim_{X_s \to 0} \frac{\beta_s e^{\beta_s X_s}}{e^{X_s}} = \beta_s. \tag{4.34}$$

Hence, if we simply set $v_s = \beta_s$ then we have exact agreement of behavior at a creative size ratio for ads that are exactly equal to the reference size.

The order variable in Engagement Mapping is important because of the marketing belief that all clicks are not created equal. If we take a cookie who has seen three impressions and compare them to another cookie who has seen three impressions and clicked on the final one, we might see an increase in conversion probability of 10. (Incidentally, this would imply that $\exp(\beta_{\mathrm{C}}) = 10$.). If we now compare to a cookie that has a click on the final *two* impressions, we need some way to allow the conversion probability change to be something other than $10^2 = 100$. It is possible that this change is actually 50, indicating that subsequent clicks, after the first, have lower weight (in this case half the influence on conversions). Alternatively, we might find that this cookie with two clicks has an estimated conversion probability 200 times higher, indicating a positive interaction between multiple clicks. How can we model this? We introduce a variable, X_{pc}, that counts the number of previous clicks before a given record. We have $X_{pc} = 0$ for any record not preceded by a click. Thus, if we have a record at e_j, then X_{pc} will be greater than 0 if and only if there is a record with $e_i < e_j$ such that the record is a click. With this definition X_{pc} is simply an count of preceding clicks and the interaction between click and X_{pc} details how effective a click is if it is not the first. There are other potential covariates that could be used to model order. One easy example would be to change X_{pc} to a binary variable instead of the sum of previous clicks. Another possible approach would be to create different categorical variables for various click combinations believed to be important. An example could be to have a factor with levels for first click, last click, and

intermediate click. In my initial testing, the X_{pc} indicator seemed to perform well.

Recency is, perhaps, trickier than order and is certainly more important since cookies with multiple clicks are much less common than cookies with multiple impressions. The goal of recency is to measure how the impact of ads diminishes over time. Typically recency is modeled with an exponential decay curve (called in the literature the "forgetting curve"). In Equation 2.1, the recency term is

$$\left(\frac{t_i - t_e}{\text{win}_v}\right)^{v_r} \tag{4.35}$$

where t_i is the time of conversion (or the end of the study), t_e is the time of an event, win_v represents the view conversion window[5], and v_r is the recency parameter in the E-Map model. Notice that $0 \leq (\frac{t_i - t_e}{\text{win}_v}) \leq 1$, so the overall recency effect is always less than 1 and v_r defines the rate of decay. Our initial approach is to follow the ideas of the order variable. I created a set of variables of the form `ad_X_to_Y` that were defined by the number of ads served to this cookie between X and Y days ago where $X < Y$. An example is the following variable:

$$\texttt{ad\_1\_to\_2}_i = \sum_j I\{e_i - 2 \leq e_j < e_i - 1\}.$$

Thus, unlike the order variable these `ad_X_to_Y` variables are not just indicators but also keep a count of the number of historical ads. Originally X and Y were chosen to partition the range of exposure times. This strategy was eliminated because it constrained the variables to sum to n_i thus creating a singular design matrix.

A second version of the recency parameter was suggested by my colleague Andrew Martin. Our term for it is "Random Follow Recency" (RFR) (whereas the above is called "Grid Recency"). In RFR we randomly choose one event for cookie i from the n_i events in its history. Begin by indexing the cookie events by $j \in (1, 2, \ldots, n_i)$ and denote the randomly chosen event by the

[5]The view conversion window is the amount of time that an ad server looks back for impressions in a cookie's history, assuming there are no clicks. For instance, if the view conversion window was set to 30 days and a cookie triggered an action tag on December 31, the ad server would look back until December 1 to see if any advertising preceded the action.

subscript r. Then we set our RFR covariate, X_{RFR} based on the following rule:

$$X_{RFR} = \begin{cases} \text{Undefined} & \text{if } j < r \\ t_j - t_r & \text{if } j \geq r \end{cases}$$

To put this variable in plain words, we choose an ad to follow and measure the time from that ad to all subsequent ads.

If we have estimated a coefficient, β_r, associated with random-follow recency, how can we translate that into a parameter in the E-Map model (v_r)? Given the formulation of the PHM, $\exp(\beta_r)$ is the change in the odds of conversion associated with increasing the temporal distance to a preceding ad by one day. If we use Equation 4.35, keeping t_i constant but replacing t_e with $t_e - 1$, we move the time of the event one day further from the time of conversion. Therefore, the ratio of more distant to less distant should be equal to our change in conversion odds. We arrive at the following relationship:

$$\begin{aligned} \exp(\beta_r) &= \left(\frac{t_i - t_e}{\text{win}_v}\right)^{v_r} / \left(\frac{t_i - (t_e - 1)}{\text{win}_v}\right)^{v_r} \\ \exp(\beta_r) &= \left(\frac{t_i - t_e}{t_i - (t_e - 1)}\right)^{v_r} \\ \beta_r &= v_r \cdot \log\left(\frac{t_i - t_e}{t_i - t_e + 1}\right) \\ v_r &= \frac{\beta_r}{\log\left(\frac{t_i - t_e}{t_i - t_e + 1}\right)} \end{aligned} \tag{4.36}$$

As with our size relationship, Equation 4.36 is not constant as a function of t_e. In this case, we simply plug in the average value for the quantity $t_i - t_e$ in the data set. For the retail data set, this quantity is, interestingly, almost exactly seven days.

Why do the time-varying covariates not cover the idea of recency? The answer is because the records comprising the cookie are treated as is independent since the start and stop times partition the cookie's life within the data set. As such, variables such as order and recency,

that measure how ads influence each other, are required in order to account for the "memory" of other behavior in the data set.

4.4 Example: Retail Advertiser

In this section we apply our modeling to a retail advertiser. We first detail the fields that are in the data set. These are calculated for every exposure to every cookie. Therefore, every one of these variables are time-varying.

- `start`: The time of the exposure measure relative to the first exposure time.
- `stop` If this is not the last record for the cookie then this variable is set to the time of the next record. If this is the last record then this is the time of the conversion or the time the cookie is lost to follow-up (relative to the first time).
- `conversion` 0 or 1 indicating if the record terminates in conversion. Each converter only creates one record where `conversion` = 1.
- `creative_type` A categorical variable defining creative type such as display, flash, rich media, text or video.
- `creative_size_ratio` The ratio of the ad to the reference size for its type.
- `click` 0 or 1 indicating if the record is a click.
- `previous_click` The sum of previous clicks or zero if this is the first click for the cookie.
- `time_to_followed_ad` The parameter associated with the "Random Follow Recency" above. It is the time since the randomly followed ad or is set to `NA` for ads preceding the followed ad.

4.4.1 Model Fitting

I fit the models of this section using the `survival` library in the statistical software R, specifically the `coxph` function. Our original model includes a three-way interaction between creative_type, click and previous_click. The response variable is a "Survival object", created by combining the information held within start and stop times and then the event variable indicating if a time interval ended with a conversion or not. The covariates are discussed above in the list that begins this section. The model form we discuss is

$$\lambda(t) = \lambda_0(t) \cdot \exp\left(\beta_{ct} \cdot X_{ct} + \beta_c \cdot X_c + \beta_{pc} \cdot X_{pc} + \beta_s \cdot \log(X_s) + \beta_r \cdot X_r\right). \tag{4.37}$$

We have the following parameter definitions:

β_{ct} Model coefficient associated with creative type parameter.

β_c Model coefficient associated with click presences or absence.

β_{pc} Model coefficient associated with the count of previous clicks.

β_s Model coefficient associated with the creative size ratio.

β_r Model coefficient associated with our recency variables. Using random-follow recency this is a single covariate and associated parameter.

Using the retail data set with 108K cookies (only 550 of which are converters) and 1.3M events in the cookie histories, we fit the above model. Table 4.4.1 contains the estimates and standard errors for the above model. Figure 4.3 provides a graphical representation of the estimates and confidence intervals. It is important to echo the concerns expressed at the end of Section 4.1. Two covariates used in the model, β_{pc} and β_r, are correlated within a given cookie. Since we are not modeling this covariance, our stardard errors are probably smaller than they should be and should be used with caution. The estimates should be accurate.

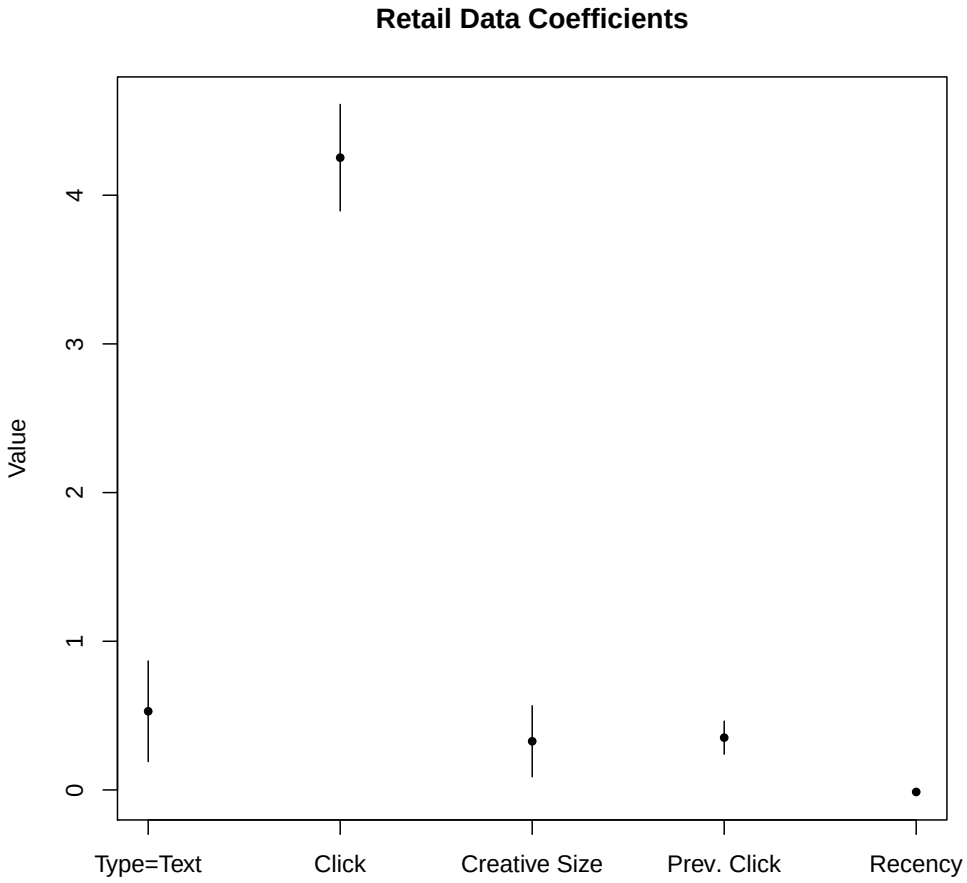

Figure 4.3: A graphical representation of the estimates and 95% confidence intervals for the retail data model coefficients. These are on the untransformed scale, so we must exponentiate to see the effect these coefficients have on conversion probabilities. The recency confidence intervals are too small to plot, though the value of recency is significantly different from zero ($p < 4 \cdot 10^{-16}$).

Covariate	Coef.	Estimate	St. Err.	exp(Coef)	Lower 0.95	Upper 0.95
Creative Type	β_{ct} (Text)	0.530	0.169	1.70	1.21	2.36
Click	β_c	4.253	0.179	70.3	49.5	99.8
Size	β_s	0.327	0.120	1.39	1.09	1.75
Previous Click	β_{pc}	0.351	0.055	1.42	1.28	1.58
Recency	β_r	-0.013	0.0016	0.986	0.983	0.990

Table 4.2: This table holds the model estimates for the retail model detailed in Equation 4.37. The estimate column holds the estimate formed by maximizing the log-partial-likelihood. The standard errors are derived from the asymptotic theory estimates. The confidence intervals are based on adding or subtracting two standard errors from the estimate and the exponentiating. The random-follow recency was used for X_r.

All covariates have a positive effect on the probability of conversion with the exception of the recency variable. By far the most influential covariate is clicks—the presence of a click increases the probability of conversion by a factor of $e^{4.253} \approx 70$. The click coefficient is also by far the most statistically significant coefficient. The next most important factor, in terms of changes to the odds of conversion and which is vastly less important than clicks, is creative type. Going from flash to text increases conversion probability by a factor of 1.7. This result is surprising as flash typically shows much higher conversion rates than text impressions. This advertiser, however, uses text impressions across a wide variety of their buys, leading to the hypothesis (untestable with privacy restrictions for this document) that the placements where text links run are substantially different from the Flash placements. Next in importance is the role of a previous click. Increasing the number of previous clicks by one increases the odds of conversion by 42%. Finally, increasing the log of creative size ratio by 1 increases conversion odds by 39%. Of course, this translates into an ad that is ten times bigger, something that is impossible in the data set. The most common size ratio in the data set is 2.3. Increasing from that size to 3.41, another common size, increases the conversion probability by 13%. This is a more realistic effect size. In this discussion I have ordered the covariates not in terms of their statistical significance (in which case Recency would be deemed the second most important) but rather by the practical significance. Increasing the number of clicks in an user's history will have a greater impact on conversion rates than simply collapsing the

user's history (which is the equivalent of moving the recency covariate in a direction that positively influences conversion odds.)

The final parameter to discuss is recency. In this model, the recency covariate used was the random-follow recency described above. The estimate for $\exp(\beta_r)$, 0.986, does not seem large. The associated covariate, X_r, varies over a wider range than the other variables in the model. The difference between seeing an ad 1 day ago versus 15 days ago is $\exp(-0.013 \cdot 14) = 0.83$. In other words, the probability of conversion goes down by 17% if the previous is ad is moved two weeks further away from a given time t. In this sense, the recency effect is similar to that of the size effect.

4.4.2 Assessing Model Fit

To assess the fit of our PHM, I borrow the Hosmer-Lemeshow test from logistic regression. A full discussion of this test can be found in Section 5.2.2. of Hosmer and Lemeshow [17]. I began by subsampling the data, keeping all 500 or so converters and randomly sampling the non-converters to reduce our total data set to approximately 5500 cookies. I did this to ease the computational burden and allow experimentation on a compressed timeline.

Conceptually, to perform the Hosmer-Lemeshow test, the data are sorted by estimated conversion probabilities in g groups. Traditionally, $g = 10$ although since the data are so large I also tested $g = 25$. There are two methods for grouping the data. The first, based on percentiles of the estimated conversion probabilities, splits the data into g groups of equal size. The first group has the n/g cookies with the smallest estimated conversion probabilities. The second group, also with approximately n/g members, has the next smallest set of estimated conversion probabilities, and so on. The second method, which Hosmer and Lemeshow showed is less effective in cases like mine where estimated probabilities are small, is to divide the probability range (0 to 1) into g equal-length groups. For instance, cut-points could be

defined at 0.1, 0.2, etc. I use the first approach, per their recommendation.

After splitting the data into g equal-sized groups, we form the test statistic. Index the groups by k for $k \in (1, 2, \ldots, g)$ and let c_k be the measured number of conversions among the n_k cookies in group k. Note that $\sum_{k=1}^{g} n_k = n$, the total number of cookies. Define

$$\bar{\pi}_k = \frac{\sum_{j=1}^{n_k} \hat{\pi}_j}{n_k}$$

as the average conversion probability in the group, where $\hat{\pi}_j$ is the estimated conversion probability for the j^{th} cookie in group k. Our test statistic is a close analog of the Pearson chi-square test statistic and compares the observed conversions to the expected using the following formula:

$$\hat{C} = \sum_{k=1}^{g} \frac{(c_k - \bar{\pi}_k n_k)^2}{n_k \bar{\pi}_k (1 - \bar{\pi}_k)}. \tag{4.38}$$

Through a series of simulation results, Hosmer and Lemeshow showed that $\hat{C} \sim \chi^2(g-2)$.

I begin by showing the data, found in Figure 4.4. This figure shows the predicted conversion proportion and actual conversion proportion across 10 groups. The range of estimated conversion probabilities that define the group is used for the x-axis labels. For instance, the group with the largest conversion probabilities (the right-most plotted values) includes all cookies with estimated conversion probabilities in the range of (0.142,1]. It is worth reiterating that 0.142 is the 90th percentile of the estimated conversion probabilities. The data are split into 10 groups. The measured proportion of converters within the group is illustrated by the black dots. The predicted proportion is illustrated by the hollow circle.

I will begin with the test statistic, described in Equation 4.38. The value of the test statistic is $\hat{C} = 13.66$. As a critical value in $\chi^2(8)$, 13.66 gives a value of $p < 0.091$. While this is not significant, our data illustrates one of the shortcomings of this method (discussed in greater detail in Harrell's *Regression Modeling Strategies* [15]). These data *do* illustrate some lack-of-fit. For the cookies in the groups with lower estimated conversion probabilities, the model of

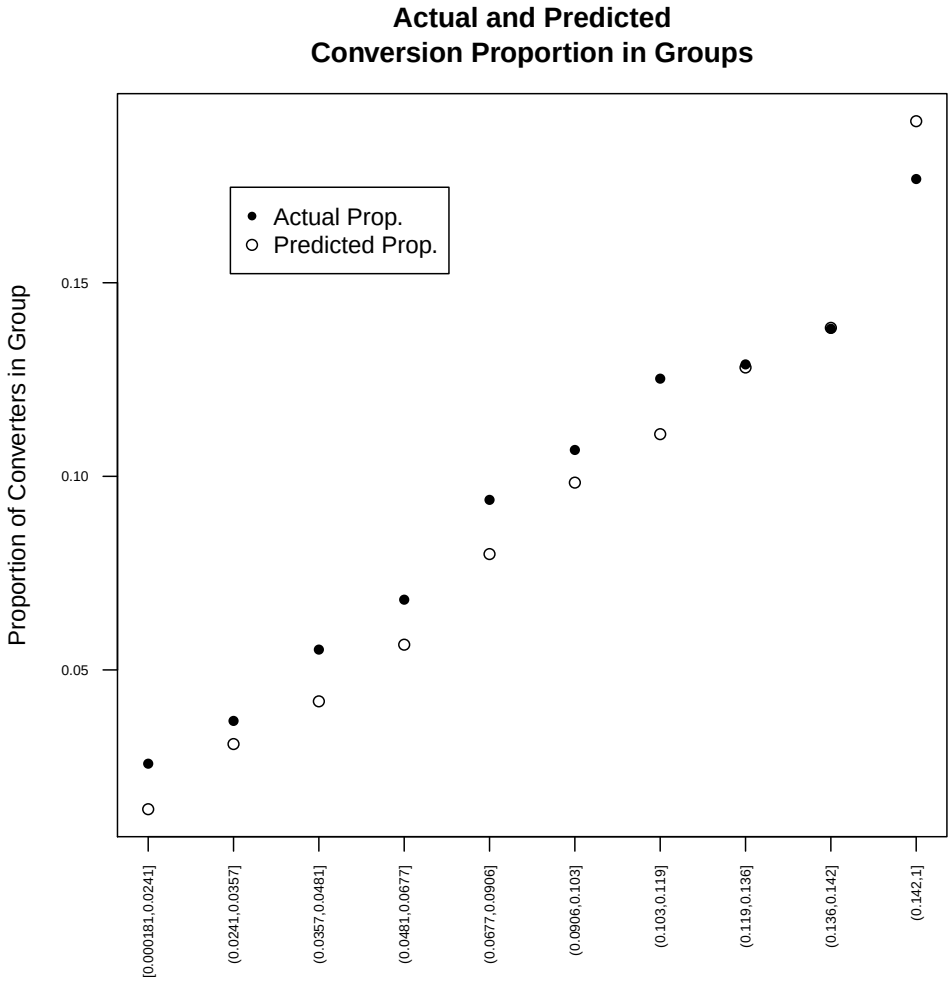

Figure 4.4: Data for the Hosmer-Lemeshow test of the goodness of fit of our PHM for the retail data. The data are split into 10 groups. The measured proportion of converters within the group is illustrated by the black dots. The predicted proportion is illustrated by the hollow circle. Although the test statistic is not significant (see text), there is evidence of lack of fit—the model underpredicts conversion rates for cookies with low estimated conversion probabilities and overpredicts for the cookies with the highest estimated conversion rates.

the retail data estimates fewer conversions than were actually measured. This effect is most pronounced (as a percentage of conversions) for the groups with low estimated probabilities. The effect diminishes as the probabilities increase. For the second- and third-highest groups, there is near perfect agreement between the predicted and actual conversion proportions. The largest group shows a significant overestimation of the number of conversions. In summary, while the test statistic does not highlight lack-of-fit, the graphical summary indicates that further refinements to the model may be necessary in the future[6].

4.4.3 Translation to E-Map

In Subsection 4.4.1 we fit a PHM model to the retail data. How can we translate those parameters into a suggested E-Map model? We handle each parameter in turn in the subsequent paragraphs.

For creative baseweights, as we discussed earlier in this section, the correct approach is to simply translate the exponentiated coefficients over to E-Map. As such, the flash ad creative type (which was the baseline variable in the contrasts for the creative type categorical variable) will be set to 1. The text impression base weight will be set to $\exp(0.52) = 1.70$. The click baseweight will be set to 70, reflecting the enormous improvement in conversion rates resulting from clicks in user histories.

The size variable for this advertiser should be $v_s = 0.32$ as described in Equation 4.34. From an advertising perspective, this number defines the diminishing returns of size. Doubling the size of an ad will result in an additional 25% of credit for that ad (since the formula reduces

[6] I experimented briefly with changing g from 10 to 25. Surprisingly, this change now gave a significant p value ($p < 0.005$). I looked at the graphical summary and it appears that two groups had wildly divergent actual conversion values. (For one, the predicted proportion of converters was about 12% and the actual proportion was 3%. Then the next group, also with a predicted proportion of around 12%, had an actual proportion of nearly 20%. These two groups together show an average actual proportion of converters that is nearly correct (11.5% versus the correct 12%). It appears there is some lack of smoothness in the data that was exposed in this grouping. This aside illustrates the sensitivity of the Hosmer-Lemeshow test to the groupings chosen for the data.

to $2^{v_s} = 2^{0.32} = 1.25$).

The random-follow recency model was discussed above and, as we saw in the model fitting subsection, moving a previous ad back one day results in a conversion odds 98.67% as large. We use Equation 4.36 and set $t_i - t_e \equiv 7$, the mean number of days between censoring/conversion and the preceding events in the retail data set. Using this equation we see that $v_r = \beta_r / \log(7/8) = 0.10$.

The previous click parameter from the PHM is related to the order variable from Engagement Mapping. Recall the definition of the order parameter in E-Map: order is set between 0 and 1 and defines the degree to which subsequent clicks diminish in importance from the first click. A value of 1 indicates that subsequent clicks share no credit for a conversion. A value of 0 indicates that each click is equally important. The interpretation of the order parameter can be complicated, though Figure 4.5 illustrates the correct answer. In this figure we see four different scenarios for a cookie with two records. This cookie has two flash ads with creative size ratio set to 0. In order of conversion probability, the four possible scenarios are zero clicks, a click on the second event only, a click on the first event only, and a click on both events. From analyzing the endpoints of the four curves, we see there is very little difference in conversion probability if a second click is added on to a first click, versus the second record becoming a click. The conversion probabilities change, though the relative conversion probabilities are nearly identical.

4.5 The Case Against Last-Ad

Throughout these first four chapters I have set up the last-ad model as a straw-man. Academic research indicates that the last-ad model is insufficient as it fails to address the three key intermediate variables of cognition, affect, and experience. The results of this chapter also illustrate the shortcomings of the last-ad model. In the last-ad model, the only variable that

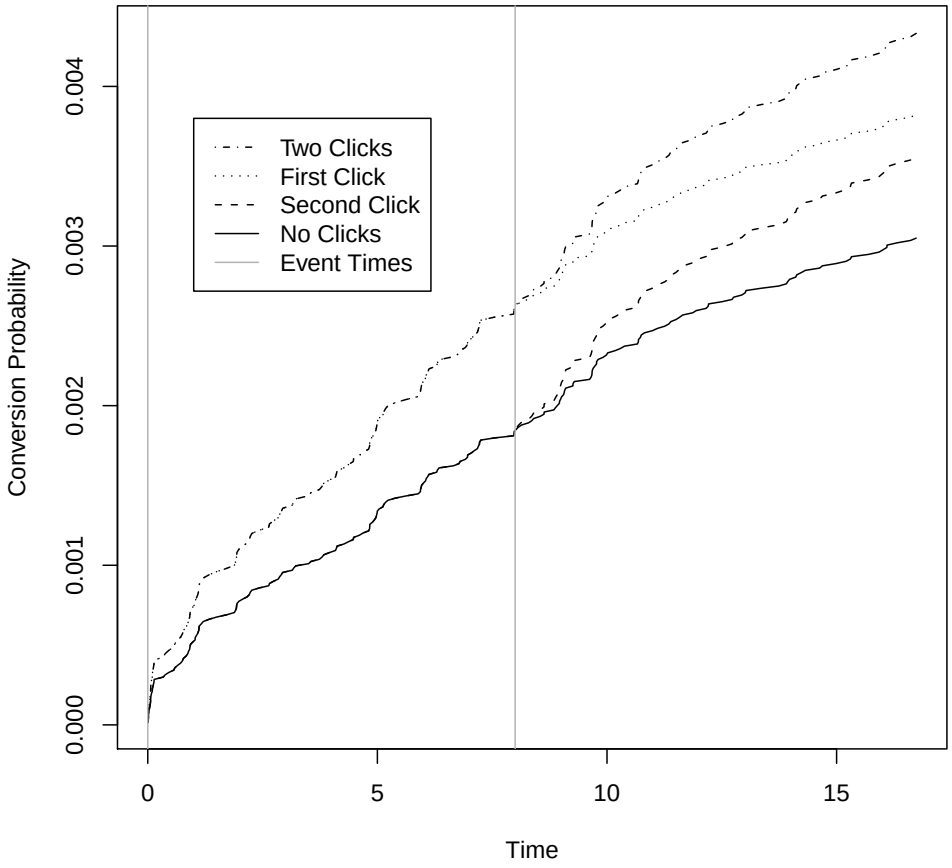

Figure 4.5: This figure illustrates a hypothetical cookie with two events. Event times are denoted by solid gray lines. The possible scenarios are no clicks, a click on the first record, a click on the second record, and a click on both records.

is influential is recency—yet we have seen that creative type, creative size, clicks and previous clicks all have an important role in conversion attribution.

There are several potential analytical approaches to illustrating the shortcomings of the last-ad model. At a fundamental marketing level, a more sophisticated conversion attribution approach is useful only to the extent that it helps someone make better decisions. At Microsoft, one of the ways we illustrate the utility of E-Map when compared to the last-ad model is to estimate the change in conversion credit under the different models. Typically we see search advertising (which is over-represented by the last-ad model) lose between 10% and 35% of its conversion credit. Display advertising gains a similar amount, as this is a zero sum game.

Since this dissertation is prepared in the presence of both a great deal of cookie-level data and a fitted proportional hazard model, we can take advantage of another interesting way to assess last-ad efficacy. My approach is to create cookie-level scores according to our model. (To ease the computation burden we perform this calculation on 100% of converters and a group of 2500 randomly chosen non-converters.) We can then rank the cookies by conversion probability and partition the population. If the model is doing its job, we will expect to see an increasing proportion of converters in each group as we move from lower scores to higher scores. Figure 4.6 holds the results of the analysis. This figure is an illustration of conversion percentages in groups of cookies, separated by estimated conversion probability. Each dot represents a collection of 50 cookies. As we move from left to right the estimated conversion probability in the groups increases. The horizontal line represents the normalized conversion probability in the sample The imposed curve is a lowess smooth and it shows that for the top third of cookies we are making predictions with some accuracy. The bottom two-thirds show no real prediction pattern. In the top 33% of predicted probabilities we have 72% chance of conversions. I did a quick permutation test (permuting actual conversions relative to the predicted probability). The sampling distribution was basically normally distributed with a mean of 47% and a largest sampled value of 51%. So there is no real question in the statistical significance of our ability to order the cookies and have a disproportionate share of converters

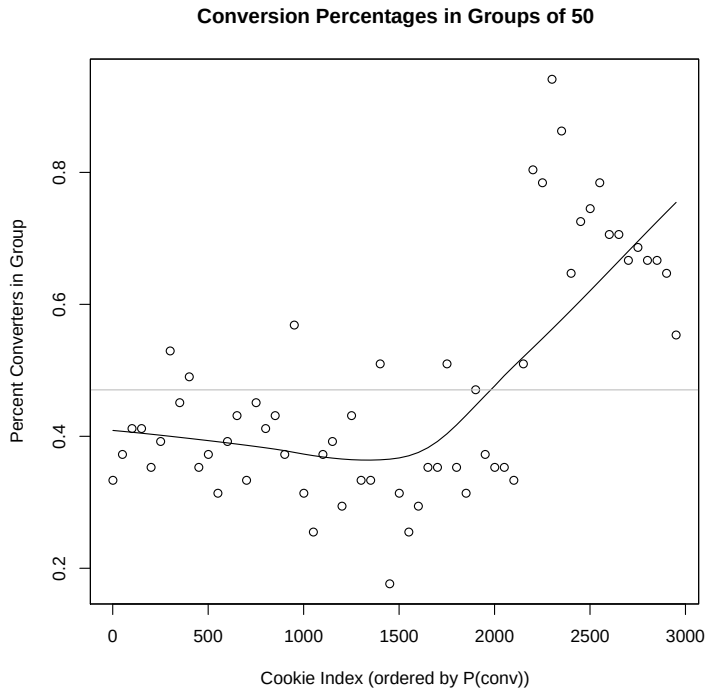

Figure 4.6: An illustration of conversion percentages in groups. Each dot represents a collection of 50 cookies. As we move from left to right the estimated conversion probability in the groups increases. The horizontal line represents the normalized conversion probability in the sample. The imposed curve is a lowess smooth. It appears for the top third of cookies we are making predictions with some accuracy. The bottom two-thirds show no real prediction pattern.

rise to the top.

The last-ad model cannot distinguish between any of these cookies—its only criterion for separation is the presence of advertising in a user's history. Our ability to distinguish between converters and non-converters here is evidence of the usefulness of our new approach (both E-Map and PHM-based coefficient estimation). This final result is arguably the most useful and has the most implications for those who manage advertising campaigns, particularly as it applies to targeting. Targeting is the ability for an advertiser (in this case, our retailer) to pay a network or publisher to deliver impressions to a particular target audience. A good example of targeting is an advertiser paying to have ads served to people who have visited the

client's website. Targeting is effective in so far as those cookies reached are a large enough group to matter *and* convert at a high enough rate to justify the premium paid for targeting. Using a technique such as that we have outlined in this chapter, a large cookie pool could be quickly scored against the survival fit. These cookies could then be ordered and a cut-off point could be chosen. (Alternatively, and more profitably, cookies could be stratified with prices modified depending on the expected conversion rate.) The advertiser could then enjoy the higher conversion rates indicated in Figure 4.6.

4.6 Computational Statistical Results

It is impossible to work with online advertising data without encountering computational constraints. The cookie records are stored in snapshots that are several petabytes in uncompressed form. Record extraction is only accessible through a distributed computing environment comprising thousands of of machines and a proprietary data processing language based on C Sharp. Additional data extraction takes place via SQL queries against databases containing additional metadata. Data cleaning and processing is done with Python—this is the language used to create both the summary data and the record-level data sets used throughout this dissertation.

The actual analysis we discuss (fitting of proportional hazard models seen here, visualizing of user histories in Chapter 5, and the clustering analysis in Chapter 6) is performed using the R language [27]. R is essentially a procedural programing language, based on the S language that also forms the core of the commercial software package S-Plus, with the greatest statistical library ever assembled. Despite its many benefits, R is memory-constrained for many of the tasks required with online advertising data. Running 32-bit Windows requires that the total memory size for R be kept under 4 gigabytes.

The analyses of this chapter were carried out using the `survival` package in R. Model fitting,

through the `coxph` function in R, actually proceeds with few issues given these data. In truth, however, the size of the data sets were somewhat chosen in order to fit within the memory constraints of `coxph`.

Fitting models is only half the battle. In order to use these models marketers must be able estimate a survival probability at a given time. Estimating survival probability also forms the foundation for the visualization techniques of Chapter 5 since the most important plot in that chapter is the estimated conversion probability over time. R includes a function that creates survival curves, `survfit`. The function takes as input a fitted proportional hazard model and an individual cookie's data with the necessary covariates (as well as some convenience variables indicating whether or not estimates such as confidence intervals and standard errors should be returned). The return value is extensive and includes survival estimates for the new cookie at all censoring and conversion times. The baseline survival estimate is based on the Kalbfleisch-Prentice estimator.

The function `survfit` provides necessary functionality, but unfortunately the implementation is incredibly inefficient. In order to estimate survival probabilities a number of modifications were required to the survfit function. These modifications were written in R and are in my function `get_survival_estimates`, included in the code appendix. For ease of replication, I walk through this code in detail here, providing commentary on key points. The input for `get_survival_estimates` are

- `cookie_event_data`: the set of record-level (or event-level, as it is called in the code) for the cookie.
- `hazard_times`: a vector of all censoring or conversion times in the data set.
- `hazard_surv`: a vector of baseline survival probabilities coinciding with the times in `hazard_times`. Resulting from the Kaplan-Meier estimate discussed above.
- `model`: A proportional hazard model fit from R.

- `times`: A vector of times at which to estimate the survival probability. If this is a single time the function will return a single probability. If this is a vector of length greater than one then survival estimates will be made for all these times.

- `return_survival` A Boolean indicating whether conversion probabilities (the default) or survival (i.e., non-conversion probabilities) are desired as output.

The principal innovation that makes `get_survival_estimates` work where `survfit` fails is the inclusion of `hazard_times` and `hazard_surv`, the baseline survival estimate. The native R function, `survfit`, estimates the baseline hazard every time the function is called. The addition of this baseline survival estimate to the quantities calculated during the function call immediately outstrips the memory allocated to R. Pre-calculating this baseline hazard is one of the keys that allows the code below to function.

We now walk through the function. We take a "literate programming" approach advocated by Knuth [23]. As such, the following paragraphs intersperse the actual working code with my explanation. Marginal comments from the original code remain.

We begin with a series of relatively common R statements to extract the necessary information from our PHM, `model`.

```
# gather model information we need
mod_coef <- ifelse(is.na(model$coefficients), 0, model$coefficients)

mod_asgn  <- model$assign # gives us the look-up between model and data
mod_terms_obj <- terms(model) #gigantic terms object, used for other stuff
mod_terms <- names(mod_asgn)

mod_frame <- model.frame(mod_terms_obj, data=cookie_event_data)
```

```
        # pulls out the relevant data for cookie
mod_mat   <- model.matrix(delete.response(mod_terms_obj), mod_frame,
              contr=model$contrasts)[,-1,drop=FALSE]
                  # a design matrix based on cookie. Handles coding of categorical
                  # drop the intercept in a PHM model
nterms    <- length(mod_terms)
pred      <- matrix(0,ncol=nterms,nrow=nrow(cookie_event_data))

mean_pred <- mod_coef * model$means
```

A number of important and complicated objects are created in this sequence of statements. The variable `mod_coef` holds the estimated coefficients of the fitted model. The variable `mod_frame` takes the cookie event data and produces a design matrix based on the data and the `mod_terms_obj`, the object that defines the model. Then `mod_mat` holds a reduced design matrix based on the contrasts in the model (for categorical data) and dropping the intercept term. The intercept term is dropped in the sixth line because the proportional hazard model assumes the existence of a baseline hazard. We create a matrix, `pred`, where the columns are the p predictors in our model and the rows correspond to the individual cookie records. (Recall that our input data, `cookie_event_data`, is made up of multiple records for a given cookie.) The vector `mean_pred` holds the mean hazard multiplier (defined as the estimated model coefficient times the covariate values.

We now fill our `pred` matrix.

```
for (i in 1:nterms) {
    ii <- mod_asgn[[ mod_terms[i] ]]
    pred[,i] <- mod_mat[,ii,drop=FALSE] %*% (mod_coef[ii])
```

```
}
# pred is now an n x p matrix where n = records and p = variables in model
```

This for loop walks through the p terms in the model (here called `nterms` for number of terms). We determine where in the design matrix these terms are used and then multiply the design matrix by the model coefficients to estimate the hazard multiplier, $\mathbf{X_i}'\beta$, for the individual record.

At this point we take the anti-log of the predicted hazard multiplier and subtract off the overall mean predicted risk. This is only necessary so our output coincides with the R output.

```
risks <- exp(apply(pred,1,sum) - sum(mean_pred))
```

We have now extracted most of the information from the cookie's history and the model. It is time to integrate these data with the baseline survival model to estimate survival probabilities for all times in the `times` variable.

```
num_events <- dim(cookie_event_data)[1]

# first just build the full survival curve

#Select down the baseline survival data because
# our raw data can have hundreds of thousands of
# points that we don't need to carry through for the calculation
this_haz_idx <- hazard_times < max(cookie_event_data$stop)

if(sum(this_haz_idx) == 0) {
    warning("Last stop time for cookie is less than minimum
                survival time.\nReturning full survival curve.")
```

```
    this_haz_idx <- rep(TRUE,length(hazard_times))
}

this_times <- hazard_times[this_haz_idx]
this_surv <- hazard_surv[this_haz_idx]
```

This previous section of code cuts down our baseline survival estimates to only times that correspond to our cookie. We throw out any survival estimates corresponding to times greater than the max time for the cookie.

We next work from the baseline survival estimates to get the instantaneous hazard estimates.

```
# create a product based version of this_surv
temp <- c(1,this_surv)
this_surv_prod <- temp[2:(length(this_surv)+1)]/temp[1:length(this_surv)]
# to generate any entry in this_surv[i] just take
# cumprod(this_surv_prod[1:i])[i]
# to the right point.

adj_surv <- this_surv_prod
```

At this point, we are ready to modify our baseline survival estimates by the variable `risks` that holds the anti-log of the estimated coefficient from the model times the covariates.

```
# we use idx to determine what parts of this_surv_prod we need to raise
# to which power. Then we make adj_surv the cumprod
for (i in 1:num_events) {
    idx <- cookie_event_data$start[i] <= this_times &
        this_times <= cookie_event_data$stop[i]
```

```
        adj_surv[idx] <- adj_surv[idx]^risks[i]
    }
```

This for loop has two steps. The first determines the index within the times vector that corresponds to a given cookie record. The second step raises the baseline survival estimate for that length of time to the appropriate power to estimate the survival probability. All of the work to create this line is the programmatic necessity to realize the ideas in Equation 4.17. At this point, however, we've worked not with the actual survival probabilities but with the hazards, so we must take the cumulative product to generate the actual estimated survival curve.

```
    adj_surv <- cumprod(adj_surv)
    # that last bit took about a week of work. Although to be fair,
    # the better part of the
    # week was spent trying to figure out exactly what the
    # differences were between
    # S(t_i), \Lambda(t_i), and \lambda(t_i). This is what happens when
    # you don't have a class in suvival analysis!
    #
    # Anyway, the key insight, that took me forever to reach,
    # was that
    # the critcal component was the multiplicative aspect
    # of the survival curve. At
    # any time, t_i, you could derive, from the fit, a
    # number $a$ that obeyed
    # the relationship a \cdot t_i = t_{i+1}. That's what's
    # in this_surv_prod.
    if(!return_survival) adj_surv <- 1-adj_surv
```

The remaining code from the function is, essentially, bookkeeping. If no times are sent in at the beginning of the function then we return the full survival curve in the following code. Otherwise we create a survival object so that we can extract the pieces we need for things like cookie targeting or plotting.

```
if(missing(times)) {
    # we want the full curve
    return(data.frame(time=this_times,surv=adj_surv))
} else {
    surv_holder <- numeric(length(times))

    for (i in 1:length(times)) {
        if (times[i]==0) { #handling an edge case
            surv_holder[i] <- as.numeric(return_survival)
        } else { # handling the normal case, some time in the middle
            t_idx <- max(which(this_times < times[i]))
            if(t_idx < 0) {
                surv_holder[i] <- 1-as.numeric(return_survival)
            } else {
                surv_holder[i] <- adj_surv[t_idx]
            }
        }
    }
    return(data.frame(time=times,surv=surv_holder))
}
```

In summary, there are a handful of tricks in this code to enable estimation of survival probabilities. The key idea is to separate baseline survival curve estimation from the further

estimation based on the proportional hazard model. The second key idea is to only use in the calculation the portions of the survival curve that are relevant to the estimation in hand. Finally, there are a number of tricks employed throughout to match the R functionality that exists in `survfit`.

Chapter 5

Visualizing Cookie Histories

Web surfing data can be baffling. A large advertiser can easily produce a billion impression records in a month with millions of clicks and associated actions. Visualizing the data provides statisticians and marketers alike with the opportunity to quickly assess patterns and trends. Industry tools have been developed to visualize data in aggregate form. Nothing, however, exists to visualize individual user histories.

This lacuna in the visualization literature makes sense. Users have enormously variable histories. Some people are exposed to one display advertisement and then purchase. Many of these conversions may be happenstance, where the marketing did not directly influence the conversion. Some people may get online knowing exactly the transaction they wish to make, performing a search, clicking on a paid keyword, and purchasing. These users create just one search record. Yet other users may be influenced by a variety of marketing messages, spooling out a history in the log records that spans several months and hundreds or thousands of marketing messages. Moreover, there is a rich diversity in the types of interactions (clicks, impressions, rich media, video, etc.), the volume of interactions (in our case studies we see some users with one record, others with more than ten thousand), and chronology (for some

behavior is tightly packed into a few hours, for others marketing events range over weeks or months).

How then can we create a visual summary of a user or a group of users? This chapter attempts to answer that question by focusing on the following critical concept: we can use probability of conversion as a yardstick against which we can measure the cookie's history over time. This enables us to plot the duration of the cookie's exposure against the changing probability of its conversion. This is particularly useful as a way to see the histories of cookies identified as cluster centers in Chapter 6. This solution will not prove perfect, but will give us a powerful tool to understand users and their histories.

5.1 Predicting and Plotting Conversion Probabilities

For the purposes of this section we will assume that we have a set of user records to which we have fit a proportional hazard model (PHM) of the type in Chapter 4. More formally, the population of interest is online users and here the observations are the individual log records belonging to the user, as discussed in the time-varying covariate discussion of the previous chapter. Each observation is a sequence of log records, ordered by time. From that data set we estimate coefficients for a proportional hazard model with the following estimators:

- Creative Type: An estimator of the amount by which the odds of conversion change with an additional exposure to an advertisement of a given creative type. The creative types we analyze in this research are typically Text, Display, and Flash.
- Click: An estimator of the amount by which the odds of conversion change when an exposure is changed from an impression to a click.
- Previous Click: This parameter gives the effect of a previous click on conversion odds. There is typically an important interaction effect between clicks and previous clicks; the

more clicks a user has in their history, the less influence any one click tends to exert.

- Creative Size: Modifies the conversion probability based on the size of the ads the user is exposed to.
- Recency: In Chapter 4 we explore how to model recency. This estimator serves two masters; providing better PHM fit via the time-to-followed-ad covariate and giving a way to estimate the E-Map recency parameter.

The goal of fitting proportional hazard models is twofold in our case. Principally we would like to estimate the effect of various kinds of cookie behavior on the cookie's probability to convert. Additionally we are interested in simply estimating the conversion probability by cookie. This latter desire makes sense when viewed from the perspective of a marketing manager. If one has a single ad to serve but has two available cookies to receive the ad, the ad should be "spent" on the cookie who will have the highest conversion probability after seeing the ad.

The mathematical details of PHM were covered in Chapter 4, so we begin with an example to illustrate the objectives. Figures 5.1 and 5.2 illustrate the estimated probability of conversion for the cookie with the highest volume in the hotel data set and one of the lowest, respectively. (Throughout this section we use the hotel data to illustrate our concepts. This data set is discussed in exhaustive detail in Chapter 6.)

The high-volume cookie figure (5.1) shows the estimated conversion probability over a 30 day time period for a cookie with a large amount of activity. The estimated conversion probabilities come from a PHM similar to that of Chapter 4. This cookie was exposed to 3734 ads over the 30 days, with 42 clicks, 16 of which took place on search advertisements. The large jumps at days 14 and 23 correspond to periods where the cookie clicked on multiple ads. At the end of the period we estimate a conversion probability (within this data set) of almost 42%. (The cookie *did* ultimately convert.) The estimated conversion probability is simply one minus the estimated survival probability discussed in Chapter 4. In contrast to this estimated conversion

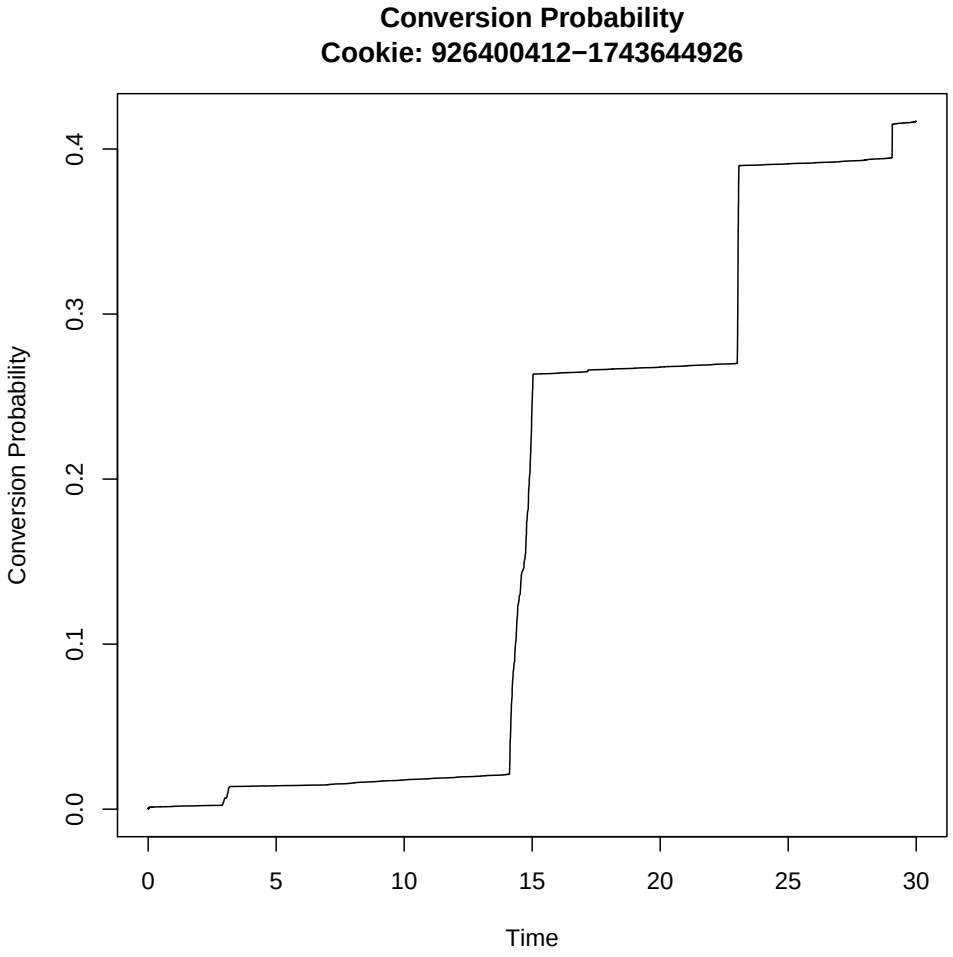

Figure 5.1: This figure shows the estimated conversion probability over a 30 day time period for a cookie with a large amount of activity. This cookie was exposed to 3734 ads over the 30 days, with 42 clicks, 16 of which took place on search advertisements. The large jumps at days 14 and 23 correspond to periods where the cookie clicked on multiple ads. At the end of the period we estimate a conversion probability (within this data set) of almost 42%. The cookie ultimately converted.

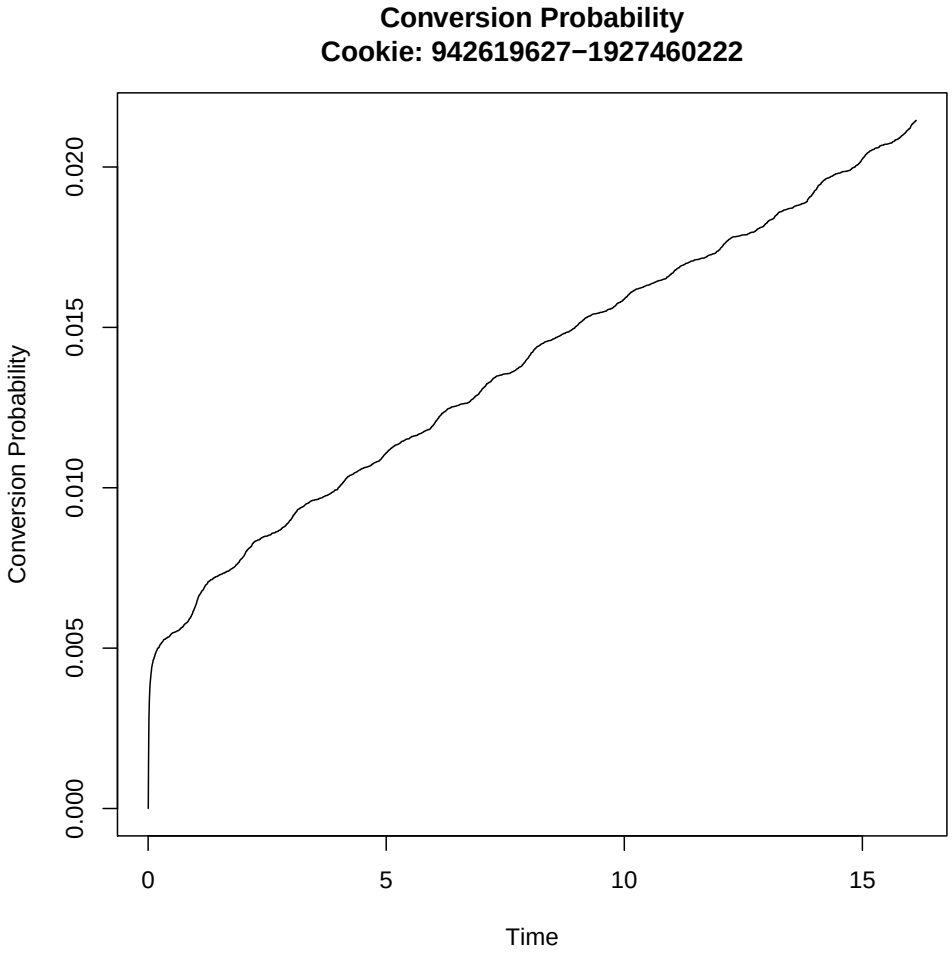

Figure 5.2: This figure shows the estimated conversion probability over a 16 day time period for a cookie with only one record, a display impression. With no changes to the covariates over the 16 days, what we see here is simply the basic Kaplan-Meier conversion probability estimate. Note the steep initial increase (a relatively large percentage of converters act almost immediately after their first exposure) and a daily periodicity to the curve. At the end of the period we estimate a conversion probability (within this data set) of only 2%.

probability, Figure 5.2 shows the estimated conversion probability over a 16 day time period for a cookie with only one record, a display impression. With no changes to the covariates over the 16 days, what we see here is simply the basic Kaplan-Meier conversion probability estimate. Note the steep initial increase (a relatively large percentage of converters act almost immediately after their first exposure) and a daily periodicity to the curve. This periodicity is an outgrowth of how people surf the internet—many light-volume surfers get online at around the same time each day. At the end of the period we estimate a conversion probability (within this data set) of only 2%.

In theory this plot is simple to extract from the R function `survfit` in the package `survival`. Unfortunately, that code is written in such a fashion that it is unable to plot survival curves (or compute survival estimates) for data sets as large as ours. (In fact, on a data set of this size the survival curve estimation fails for any cookie with more than one record.) I have rewritten the relevant code of the `survival` package to be more efficient, enabling both the creation of these charts and also quick estimation of conversion probability from a cookie history. Section 4.6 details the code changes necessary to create these estimates and the resulting figures. All code can be found in the code appendix.

How is this conversion probability calculated? The hazard function, $\lambda(t)$, gives the instantaneous rate of conversion, conditional on non-conversion to a given time t. The survival function, $S(t)$, gives the cumulative probability of non-conversion at time t. The two functions are linked through the identity

$$\begin{aligned} S(t) &= \exp[-\Lambda(t)] && (5.1) \\ &= \exp\left(-\int_0^t \lambda(u)du\right). && (5.2) \end{aligned}$$

Within the PHM framework, covered in detail in Chapter 4, our parametrization of the hazard function is

$$\lambda(t, \mathbf{X}|\beta) = \lambda_0(t) \cdot e^{\mathbf{X}\beta}$$

where λ is the hazard function, λ_0 is the baseline hazard function, and $e^{\mathbf{X}\beta}$ is our proportional hazard model for how additional covariates affect our hazard. From this we can derive the survival function as

$$S(t, \mathbf{X}|\beta) = [S_0(t)]^{\mathbf{X}\beta}.$$

Therefore, to estimate conversion probabilities (which are $1 - S(t, \mathbf{X}|\beta)$), we must estimate the baseline survival function $S(t_0)$ and then our additional risks enter into our estimate as the exponent. We estimate the baseline survival using the Kalbfleisch-Prentice estimate [21]. This estimate reduces to the more familiar Kaplan-Meier estimate when we weight all cookies equally[1]. The Kaplan-Meier estimator (which we detail here because it is much simpler) is

$$\hat{\lambda}(t) = \prod_{t_i \leq t} \frac{n_i - d_i}{n_i}$$

where n_i is the number at risk of conversion at time t_i and d_i are the number of conversions. These quantities are determined at every unique conversion or censoring time in the data. For instance, in the hotel data set, the product ranges over $1.1 \cdot 10^6$ values. Once we have computed $\hat{\lambda}$, we can then estimate $S(t)$ using the identity 5.2.

Assume now that a given cookie has events at t_i for $i \in (1, 2, \ldots, n)$. Since we are using a proportional hazard model with time-varying covariates, at each time t_i we have a covariate vector $\mathbf{X}_i$. Similarly, we have a risk multiplier based on that covariate vector of $\mathbf{X}_i\hat{\beta}$, obtained from on our partial-MLE fit of β. We can therefore estimate the survival probability at any time t_k as

$$S(t_k) = \left[\hat{S}(t_k)\right]^{\mathbf{X_i}\hat{\beta}} \quad \text{where } t_i \leq t_k < t_{i+1}.$$

[1]The concept of cookie weighting is interesting within its own right, although I do not delve into the issue in this work.

Recall that we have sampled our data so that converters are over-represented relative to reality. For instance, the hotel data contains about seven thousand converters and fifty thousand non-converters. The converter sample represents 20% of the total converter population during this time period. In contrast, the non-converters represent 0.08% of the total non-converter history. As such, if were going to produce accurate conversion probabilities we should weight non-converters 250 (equal to 20/0.08) times greater than the converters. Instead of carrying these weights along through all the calculations in the text, we ignore the issue to all clearer exposition of our central topics and simpler notation.

It is one minus this quantity that is plotted in Figure 5.1.

One other interesting view of these data, illustrating one potential use of the visualization techniques I am discussing, is seen in Figure 5.3. This figure shows the conversion probability over a 30 day time period for ten different cookies, all with ten records. The y-axis has been truncated to show the nine cookies with relatively small estimated conversion probabilities. The topmost line, representing a cookie with two search clicks and one display click, has a maximum probability of conversion of 44%. (If we change the y-axis to cover this full range the remaining nine cookies are much harder to distinguish.) There are two short lines, representing cookies with 2 and 14 days of history. The rest of the cookies have 30 days of history. Again we return to the overarching goal of this chapter. A marketing manager, faced with a choice of whom to show an advertisement to, would like to choose the cookie where an additional ad results in the highest change in conversion probability. This quantity can be estimated from the data underlying these curves. Moreover, this plot gives an easy way to illustrate the different experiences of ten cookies, all of which may appear similar when summary statistics are looked at.

5.2 The Tie to Engagement Mapping

In the previous section, we described the mechanism by which we can plot conversion probabilities over time to investigate the probability that a cookie will convert. In this section we extend the concept of plotting survival probabilities to plotting Engagement Mapping (E-Map) models.

There are important reasons why we are making this extension. E-Map is ubiquitous for major advertisers using Atlas. Nearly all advertisers have implemented an E-Map model, typically one of the default models. On the other hand, for many day-to-day marketers concepts such as of proportional hazard models and baseline survival estimates are esoteric. Additionally,

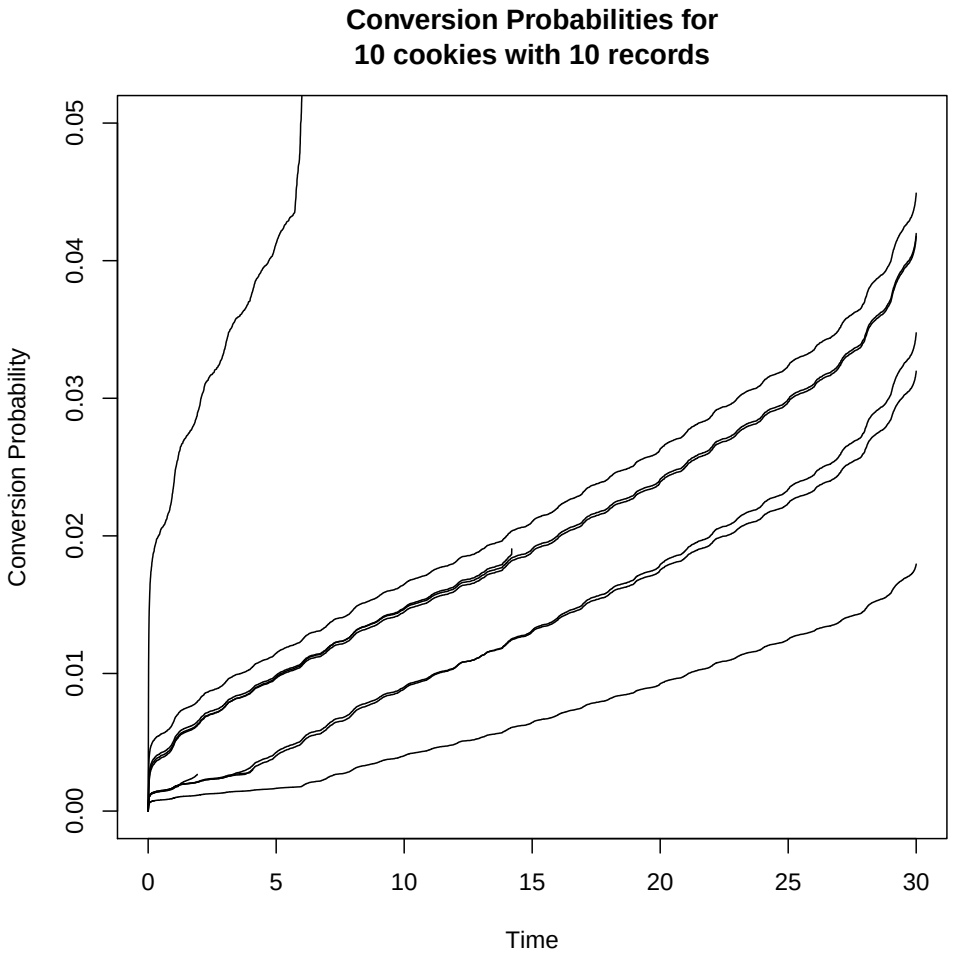

Figure 5.3: This figure shows the estimated conversion probability over a 30 day time period for ten different cookies, all with ten records. The y-axis has been truncated to show the nine cookies with relatively small conversion probabilities. The topmost line, representing a cookie with two search clicks and one display click, has a maximum probability of conversion of 44%. Shorter lines indicate short cookie histories.

few of these companies have the ability to gather their log records and perform the necessary analyses to produce the estimates of the last section. If the only route to visualizing cookie paths runs through expensive and complicated analyses, the majority of advertisers will not be interested. Therefore, I seek to develop a methodology that allows visualization of cookie histories based on E-Map parameters that agrees in some sense with the PHM model estimates, allowing one tool (Engagement Mapping) to serve as a conversion attribution methodology *and* a visualization technique. In short, if all marketers had access to the analysis of Chapter 4 then the results of the previous section would suffice. These analyses are unlikely to become widely available in the near term and so I will show that we can use E-Map directly to plot cookie histories.

That said, how can one use a conversion attribution methodology to produce plots? Whereas with PHM the solution is obvious (plotting conversion probabilities over time) with E-Map we will require some additional motivation. We will build on the ideas of Section 2.3 and we suggest the reader be familiar with the concepts found there, particularly the concept of how E-Map distributes conversion credit across marketing events.

Fundamentally E-Map is designed to take a given conversion and share credit across all media exposures leading up to that conversion. In order to conceptualize E-Map as a visualization tool, I must first expand the definition. First, some notation: E-Map can be thought of as a function, f, taking as input the following: a set of marketing data, $\mathbf{X}$; a vector of parameters, θ, that define the E-Map function; and time, t, measured as time elapsed since the first marketing exposure. Assume further that $\mathbf{X}$ is an $n \times p$ matrix where n is the number of marketing exposures to a person before their conversion and that there are p attributes determining the conversion score via E-Map. Further assume that $\mathbf{X}$ is sorted in ascending time order so that row i precedes row j in time if $i < j$. (This will just be a notational convenience for what follows.) The mechanics of this conversion attribution function, $f(\mathbf{X}, \theta, t)$, now become important. The final function value is calculated in two steps, returning first raw scores for each exposure in $\mathbf{X}$ and then normalizing these so they sum to 1. The two columns of

Table 2.2 illustrate this concept. The base weights are the starting point for the raw scores, the column giving the percent of credit shows the normalization at work. Although this normalized score is what is required for E-Map to be useful to advertisers as a conversion attribution methodology, in the discussion below I will work with raw scores. Furthermore, we will add the parameter of time to the function.

Let $f(\mathbf{X}, \theta, t) = \mathbf{s}$ be an E-Map function and let us define this function at any time t for a set of exposures to a cookie $\mathbf{X}$ and given a set of accompanying parameters θ. When used as a conversion attribution methodology, f just maps marketing data ($\mathbf{X}$) onto scores ($\mathbf{s_c}$) at the time of the conversion (t_c). Write f as $f(\mathbf{X}, \theta, t_c) = \mathbf{s_c}$ where t_c is the time of a conversion (referred to as t_i in Chapter 4) and s_c is the resulting score vector. The vector $\mathbf{s_c}/\sum \mathbf{s_c}$ is the resulting normalized score vector for the conversion. Contrary to this intended use, we will now consider E-Map functions evaluated at any time $t > 0$. If t predates the time of any marketing event in $\mathbf{X}$ then $f(\mathbf{X}, \theta, t) \equiv 0$, since no credit is given for events yet to take place. Ff t is so far past the events of $\mathbf{X}$ that there are no more events in the advertiser's conversion window (again, typically 30 days if there is a click in $\mathbf{X}$ and 7 days if there are only ad views) then we will have $f(\mathbf{X}, \theta, t) \equiv 0$ [2]. The case between these two extremes is more interesting. Index the events in $\mathbf{X}$ by $k \in (1, 2, \ldots, n)$. Note that if $k_1 < k_2$ in our index then $t_{k_1} < t_{k_2}$ since $\mathbf{X}$ is ordered by time. Then, if $t_k \leq t < t_{k+1}$ we have $f(\mathbf{X}, \theta, t) > 0$. One way to think about this generalization of the E-Map function is that $f(\mathbf{X}, \theta, t)$ assumes (counter-factually) that a conversion exists at time t and then applies the E-Map model using only the preceding events in $\mathbf{X}$.

Since we are not normalizing the scores in $f(\mathbf{X}, \theta, t) = \mathbf{s}(t)$, the sum, $\sum \mathbf{s}$, will fluctuate with time. Figure 5.4 gives an example of a score vector. The stair-step nature of the function is the result of additional messages being delivered, the drop in credit after the peaks represent the role of the recency decay parameter. Three times are highlighted, indicated by a dark

[2] The same results hold for the PHM formulation. $S(0) = 1$, since no cookies have converted at time 0 and the hazard rate drops to 0 once we have gone more than 30 days from an ad exposure.

line, at $t = 10, 17,$ and 24.6 days.

There are important differences between the E-Map plotted values and the PHM conversion probability plot. First, the PHM plot is strictly non-decreasing whereas the E-Map plot decreases because of the recency function. Second, the E-Map plot is simply plotting $\sum \mathbf{s}(t)$ over time—it is not plotting probabilities. In some sense this E-Map function is the analog of $\lambda(t)$ rather than $S(t)$, measuring the moment-by-moment propensity to convert. We can create an analog of the cumulative hazard function, $\Lambda(t)$, by taking the cumulative sum of the function in Figure 5.4. Figure 5.5 shows the same data with the score summed cumulatively rather than displayed instantaneously.

We have done simulation analyses on the correlation between these two sets of scores, as well as tried to prove a lower bound on the correlation. Typically the correlation between the cumulative E-Map score and $S(t)$ is quite good (the mean correlation across 1000 hotel cookies was 0.46) although there is a great deal of variability (the quartiles Q_1 and Q_3 are 0.11 and 0.97 respectively). In fact, the distribution of correlations was bimodal: cookies with one record all had correlations under 0.22 with a mean of 0.13; cookies with more than one record had correlations greater than 0.56 with a mean of 0.93. In order to understand this disparity, we take a closer look at the respective scores.

For cookies with one record, the E-map score at any time after the advertisement is shown is simply $b \cdot s \cdot r$ where b is the baseweight, s is the size multiplier, and r is the recency or order effect. See Equation 2.1 for the details. The PHM estimate of conversion probability is $\left[\hat{S}(t)\right]^{\mathbf{X}\hat{\beta}}$. The expression $\mathbf{X}\hat{\beta}$ is the dot product of the coefficients with the individual event covariates. The covariates in the models of Chapter 4 are chosen to mirror the variables in the E-map model. Thus, increasing values of $b \cdot s \cdot r$ translate into larger values of $\mathbf{X}\beta$ and decreased survival estimates. Conversion probabilities increase as E-Map scores increase.

What accounts for the wide variety of correlation coefficients seen for cookies with only one

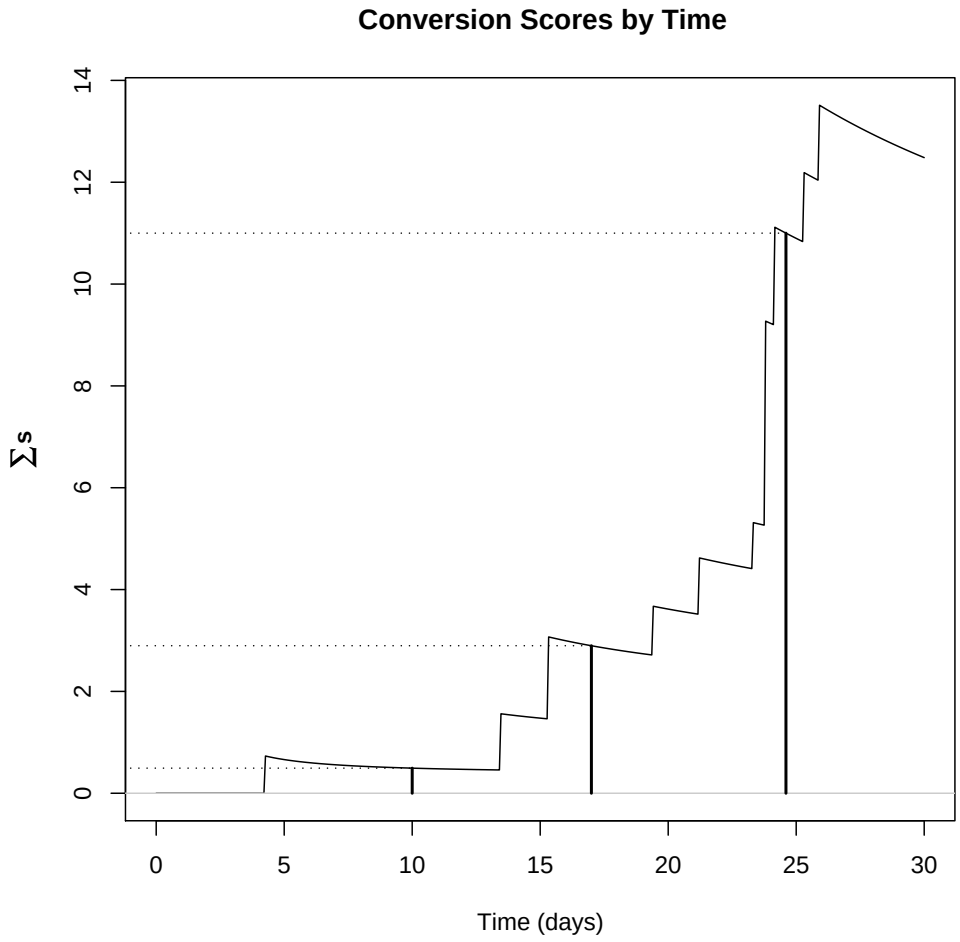

Figure 5.4: The thin continuous line represents the value of $\sum \mathbf{s}(t)$ over time. The function $f(t|\mathbf{X}, \theta) = \mathbf{s}(t)$ gives a vector of scores for the individual marketing messages that have been delivered. The stair-step nature of the function is the result of additional messages being delivered, the drop in credit after the peaks represent the role of the recency decay parameter. Three time values are highlighted, indicated by a dark line, at $t = 10, 17,$ and 24.6 days.

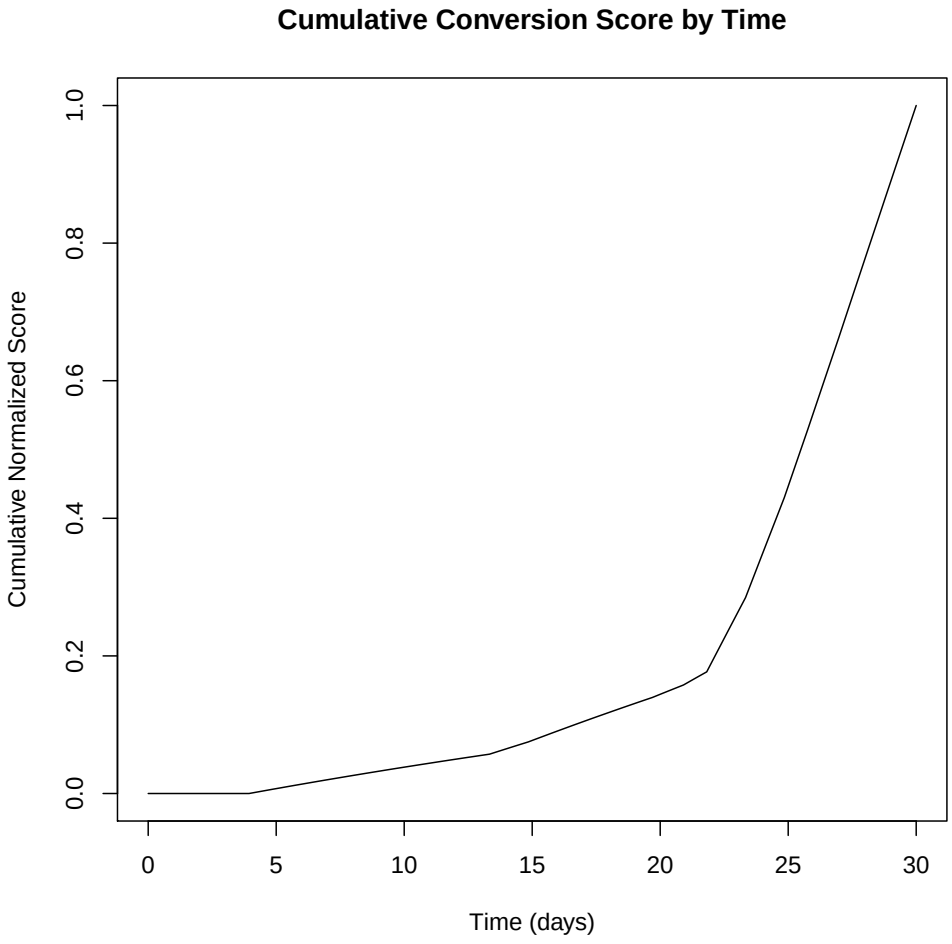

Figure 5.5: A cumulative plot version of Figure 5.4.

record? The answer is, surprisingly, time of event. Through an artifact of the data collection process, $\lambda(t) = 0$ for $t > 30$. (We only collected 30 days of converter history for computational convenience.) Cookies that do not convert, however, can have longer histories and, when the E-Map curve is built over longer periods of time, the flat stretch of conversion probability from day 30 onwards causes lower correlations. When we restrict our analysis of the correlation between E-Map scores and conversion probabilities from PHM, the mean correlation for cookies with one record goes from 0.13 to 0.17 (still not great). The overall mean correlation goes from 0.46 to 0.61. One of the ongoing projects from this work is to derive a mathematical relationship between conversion probabilities and E-Map scores.

5.3 A Statistical View

This section briefly reprises the key statistical points from this chapter, particularly for a non-marketing audience. In this chapter we describe a technique that can be used to visualize cookie histories as they vary over time. I begin with the estimated survival function using the proportional hazard model:

$$\lambda(t, \mathbf{X}|\beta) = \lambda_0(t) \cdot e^{\mathbf{X}\beta}$$

where λ is our hazard function, λ_0 is the baseline hazard function, and $e^{\mathbf{X}\beta}$ is the term expressing the effects of additional covariates on the hazard. From this we can derive the survival function as

$$S(t, \mathbf{X}|\beta) = [S_0(t)]^{\mathbf{X}\beta}.$$

This survival function is illustrated in Section 5.1. Plotting these survival functions on data as large as ours required a full rewrite of the code in R's `survival` package[3].

I then formulated an approach to plotting cookie histories using Engagement Mapping models

[3]The code appendix contains the code and Section 4.6 details the changes required to create the estimates necessary for the figures in this chapter

instead of proportional hazard models. I do this because many advertisers have access to E-Map but not to PHM (or, more accurately, not to statisticians to perform PHM analyses on their behalf). Moreover, the correspondence between E-Map and the PHM requires additional investigation and explication before PHM will be accepted beyond the field of statistics. I then illustrated how E-Map scores can be used to plot a cookie history and carried out a brief analysis looking into the correlation between E-Map scores and PHM estimated conversion probabilities. The correlation is always positive (and we provide a heuristic argument as to why this makes sense) and, when the time domain is restricted so that the functions are evaluated over the same period, has a mean correlation of 0.61 on a sample of 1000 cookies from the hotel data set.

Chapter 6

Finding Common Cookie Histories with Clustering

A principal challenge with applying conversion attribution models is the inability to apply the models to representative case histories, measure the resulting attribution, and visualize the results. A further challenge with models such as those in Chapter 4 is that marketing managers require good agreement between the fitted models and measured data. Practitioner adoption will be minimal if a model predicts that users receiving three ads and clicking on one will convert with probability 0.01 and these types of users actually have a measured conversion rate much higher or lower than this. Therefore, it is desirable to calculate model fits and summary statistics (from the actual data) for a variety of cookie scenarios. In traditional modeling, this can often be done rather simply: fit the model to representative values for the explanatory variables. In conversion attribution modeling, this approach will be insufficient as the model is fit to a cookie history with multiple events over a range of times.

One solution is to create mock data sets. The downside of this approach is the modeler runs the risk of creating a user history that does not exist. As Therneau and Grambsch say,

"creating such a covariate path is difficult; it is all too easy to create baseline hazards that correspond to a subject who is either uninteresting or impossible." [32] Although in our data impossibility is not a great concern [1], we do worry that our fitted models will be applied to implausible user histories. A fictitious path showing one display impression every day for seven days may be interesting from a modeling perspective but may not occur "in the wild."

Moreover, the data sets we work with are quite large: tens of thousands of users with millions of records. How can one discover patterns when faced with such a volume of data? Traditional summary statistics are almost useless at this scale. While one may simply calculate the average number of impressions or clicks across the users, these numbers may bear little relation to an actual history experienced by a user. We wish to understand converters and therefore finding common experiences is a useful undertaking.

In this chapter, we explore an approach that is different from the reasoning-by- summary-statistic tack taken by most practitioners: clustering cookies into groups. There are two immediate benefits of such an approach. The first is that we can discover patterns in converter histories. Each cluster should represent a set of common user experiences and from those we can derive a better understanding of the types of users that convert, their patterns of behavior, and the probability that these types of users will ultimately convert. The second benefit lies in the "center" of these clusters. We can visualize (using the techniques of Chapter 5) the users that form the medoids of the clusters. The difficulty in working with the details for thousands of cookies can be avoided by using these representatives. Finally, we can score the clusters overall with both the model and the data and determine how good a job in aggregate we are doing estimating the probability of conversions.

The first section of this chapter describes the theoretical underpinnings for our clustering approach. The second section is a detailed example using data from a hotel advertiser.

[1] The canonical example of the impossible subject is using a value of 0.5 for a variable such as `male` where 0 corresponds to female and 1 to male.

6.1 Clustering Based on User Histories

We begin by discussing the data set we will use to cluster cookie histories. We then turn to the primary clustering algorithm we will use, `pam`, found in the software package `cluster` within R. Next, we focus on the distance matrix that lies at the foundation of any clustering technique and ours in particular. We then compare the results of the partitioning-based clustering algorithms with a hierarchical clustering algorithm. Finally, we leverage the gap statistic, a "number of clusters" statistic, to compare our results with some more objectively derived cluster results.

6.1.1 Data for Clustering Cookies

Recall the data structure in Table 2.1 from our chapter on online advertising. This table represents a set of log records (captured by the server) for a single cookie. To cluster these data we must create a dissimilarity matrix that tells us the distance between observations. Since we wish to cluster cookies rather than records, we must summarize the data at the record level and create a new data set at the cookie level.

Certain metrics for cookie experience immediately suggest themselves. The first is simply the number of records for the cookie. Users with very little Internet activity are less likely to transact online and probably require different marketing approaches. We can further subdivide the total number of advertising events into counts of impressions and clicks. Again, users who have very few impressions are typically less active Internet users. Similarly, recent research from comScore [20] indicates that the presence or absence of clicks as well as click volume tells us more about Internet behavior than might have been previous suspected. In addition to these basic metrics, there are natural subdivisions within them. For the impression count, it seems worthwhile to keep count of the type of impressions delivered across our three principal types of events: text links, standard JPEG or GIF display ads, and Flash or java rich media

ads. These three types will be denoted "Text", "Display", and "Flash". For clicks, we add an additional variable that denotes whether or not the click happened on a search placement. The behavior of people who click on search ads is different from the behavior of those clicking on display ads. Typically search events appear closer to a conversion after the consumer has decided to take action to make a transaction. Display clicks, by contrast, tend to be more spur-of-the-moment and often indicate greater engagement with the advertiser's ads. Finally, we suspect that users fall into certain basic types based on the type of media they have been exposed to. There are those who are exposed only to display advertising, those who are exposed to only search, and those who have both search and display in their histories. We create binary variables for this partition of the data set.

In this chapter, we make extensive use of data from a hotel advertiser. These data are summarized in Table 6.1.1. Since the data are displayed so comprehensively, we will not belabor the data description except for these few points:

- **RecordCount** is the total number of records (both clicks and impressions) in a user's history and shows a great deal of variability with a median of three records, a mean of 32, a 95% percentile of 156 and a maximum over 3000.
- **Text**, **Display**, **Flash**, and **Search Events** show little variability except for Flash which composes over 95% of the display ad views. There are a variety of search events considering the typically low volumes we see for that medium.
- **Clicks** and **Display Clicks** show a similar distribution as search events, unsurprising since clicks and search events are nearly synonymous. Display clicks, the subset of clicks that occur on display ads as opposed to text ads, show an expected lower volume.
- **DisplayOnly**, **SearchOnly**, and **SearchAndDisplay** partition the data set with 72% of converters being exposed only to display, 17% having only search events, and 11% showing exposure to both media.

Hotel Advertiser Conversion Data
10 Variables 7318 Observations

	n	unique	$\bar{x}$	.05	.10	.25	.50	.75	.90	.95
RecordCount	7318	393	31.04	1	1	1	3	16	74	156.1
Text	7318	20	0.5827	0	0	0	0	1	2	3

Display (JPEG and GIF)

n	unique	Mean
7318	13	0.0466

	0	1	2	3	4	5	7	9	10	13	17	32	34
Frequency	7166	100	30	7	4	3	2	1	1	1	1	1	1
%	98	1	0	0	0	0	0	0	0	0	0	0	0

	n	unique	Mean	.05	.10	.25	.50	.75	.90	.95
Flash	7318	4	0.01	0	0	0	0	0	0	0
SearchEvents	7318	18	0.5295	0	0	0	0	1	2	3

	0	1	2	3	4	5	6	7	8	9	10+
Frequency	5253	1232	437	189	86	49	26	10	13	5	18
%	72	17	6	3	1	1	0	0	0	0	0

	n	unique	Mean	.05	.10	.25	.50	.75	.90	.95
Clicks	7318	22	0.0259	0	0	0	0	0	0	1
DisplayClicks	7318	14	0.09634	0	0	0	0	0	0	1

	0	1	2	3	4+
Frequency	6951	238	66	26	37
%	95	3	1	0	0

	n	unique	Mean
DisplayOnly	7318	2	0.7178
SearchOnly	7318	2	0.1727
SearchAndDisplay	7318	2	0.1095

Table 6.1: This table has an extensive description of the hotel advertiser data used for clustering in this chapter. There are 10 variables total and this table gives summary statistics and the distribution for all.

6.1.2 Calculating a Distance Matrix

The goal in cluster analysis is to segment the data in such a way that the cookies within a cluster are more similar to each other than they are to cookies in other clusters. In order to achieve this goal we must define some notion of dissimilarity between observations. Let $\mathbf{D}$ be an $N \times N$ distance matrix where, for the hotel data, $N = 7318$. Nearly all algorithms require a symmetric distance matrix ($\mathbf{D} = \mathbf{D}^T$) and that will be the case for these data. Additionally, we define $d_{ii} = 0$ for all i. In nearly all clustering applications the choice for distance is simple Euclidean distance where, if x_i and x_j are two vectors of observations of length n then

$$d_{ij} = L_2(x_i, x_j) = \frac{1}{n} \sum_{k=1}^{n} (x_{ki} - x_{kj})^2$$

where d_{ki} denotes the k element of observation i.

With our data set the Euclidean norm is inappropriate. The first objection is that the variables are of different scales with `RecordCount` $\in (1, \ldots, 3129)$ whereas the partitioning variables are binary. This problem can be remedied by scaling the variables but nevertheless problems persist. Following the discussion in Kaufman and Rousseeuw [22], who in turn derive their work from Gower [13], certain of our binary variables can be considered "asymmetric". Whereas symmetric binary variables carry equal weight with agreement and disagreement (e.g., a variable called `male` coded as 0 or 1), asymmetric variables convey greater information in agreement than in disagreement. A useful example of such variables could be diseases. If we had a health data set that included the presence or absence of various diseases, we would put much greater weight on the mutual presence of a disease than on the mutual absence. Similarly, if two cookies share a 1 for `SearchOnly`, we wish that agreement to carry disproportionate weight. This could be accomplished with scaling although the R function `daisy` gives a convenient alternative option. The `daisy` variant of the Gower similarity computes

distances as

$$d(x_i, x_j) = \frac{\sum_{k=1}^{n} \delta_{ij}^{(k)} d_{ij}^{(k)}}{\sum_{k=1}^{n} \delta_{ij}^{(k)}}$$

where $\delta_{ij}^{(k)}$ is an indicator variable for the k^{th} element in our data vector between observations i and j and $d_{ij}^{(k)}$ is an, as yet unspecified, element-level distance calculation. Typically $\delta_{ij}^{(k)} = 1$ in the case where both observations have non-missing values in the k variable. The only exception to this rule is that $\delta_{ij}^{(k)} = 0$ when k is an asymmetric binary variable and $x_i^{(k)} = x_j^{(k)} = 0$. In other words, the mutual absence of trait measured by the k^{th} variable is not evidence of similarity between observations i and j. This distance metric requires the classification of variables into different types. For each of these types, a different distance function is defined.

The first variable type is the common interval-scaled variable where the distance function is simply the L_1 norm divided by the variable range or

$$d_{ij}^{(k)} = \frac{|x_i^{(k)} - x_j^{(k)}|}{R_k} \tag{6.1}$$

$$d(x_i, x_j) = \sum_{k=1}^{p} d_{ij}^{(k)} \tag{6.2}$$

where R_k is the range taken over all observations on the k^{th} variable. If all variables are interval-scaled, then this distance metric yields the Manhattan distance metric. For categorical or binary variables our distance metric is

$$d_{ij}^{(k)} = \begin{cases} 1 & \text{if } x_i^{(k)} \neq x_j^{(k)} \\ 0 & \text{if } x_i^{(k)} = x_j^{(k)} \end{cases}$$

This more elaborate distance metric has nice features for our hotel advertiser data set. Differences in the number of events and records are scaled to 0-1 and carry equal weight. Our binary variables, all considered asymmetric here, do not pull observations together when neither observation falls within a particular group.

The distance matrix defined above justifies the difficulty in its assembly by mirroring our intuition about what distinguishes users. For thoroughness, I tested clusters based on the standard distance matrix using the Gower dissimilarity measure and also using standardized variables with Euclidean distance. (The variables were standardized by subtracting the mean and dividing by the mean absolute deviation.) The Gower clustering without custom variable definition, treating all variables as interval scaled, showed relatively good agreement with the more elaborate model. Without proper treatment of the binary variables, there were higher rates of these variables being split across clusters. For instance, with three clusters it was common to see one cluster containing search, display, and search and display cookies. In contrast, using the more complicated approach, we see strict separation between `SearchOnly`, `DisplayOnly`, and `SearchAndDisplay` in the three cluster solution. With the interval-scaled Gower we see mixing between the two categories involving search. The scaled Euclidean distance matrix yields similar results with the blending even more pronounced. As such we proceed with the custom Gower distance matrix for the clustering algorithms discussed in the next two sections.

6.1.3 Partitioning-based Clustering

We focus on partitioning-based clustering algorithms that assign each observation directly to a cluster without regard for probability models. By far the most popular version of these types of algorithms is the K-means algorithm. The algorithm is simple if we assume a random starting configuration: every observation is assigned to the cluster pertaining to the cluster center to which it is closest, followed by a recalculation of the cluster centers. The analyst must define K, the number of clusters into which the data will be grouped. The process repeats until assignments cease to change. With well-chosen cluster center initialization or a lucky random start, this algorithm converges quite quickly. The best practice is to start from several random starting locations and compare the output. Comparing the resulting clustering allocations can be pretty tedious, so it is unclear how many people carry out this

step in practice.

We use a slightly more robust version of K-means known as Partitioning Around Medoids (PAM)[22] and implemented in the R function `pam`. PAM has a number of advantages in our case over K-means: we can work from our custom dissimilarity matrix; we are not constrained by Euclidean distance; and the cluster centers are actual observations in the data set rather than average observations. (In most cases this last advantage is slight since, if there is an observation close to the center, that can always be used instead to summarize the data.) The algorithm for PAM is very similar to K-means:

1. Choose a k, the number of clusters we will create.

2. Randomly choose k observations from the N total observations in the data set. Call these medoids m_k for $k \in (1, \ldots, K)$.

3. Assign each observation to one of the clusters, $C(1), C(2), \ldots, C(K)$. This is based on the formula

$$C(i) = \underset{1 \leq k \leq K}{\operatorname{argmin}} \; d(x_i, m_k)$$

   where $d(x_i, m_k)$ is the distance between observation x_i and cluster medoid m_k.

4. For each m_k and each x_i with $C(i) = k$, try switching m_k and x_i and compute the total distance in the resulting configuration. This total distance is defined in terms of having observation i as a medoid:

$$D(m_i) = \sum_{k=1}^{K} \sum_{C(j)=k} d(m_i, x_j).$$

   In other words, we sum the distance between m_i and x_j for every x_j with $C(j) = k$ as in step 3, and then sum over all K clusters.

5. Select the m_k giving the lowest total distance.

6. Halt when there are no more medoid changes.

As in the K-means algorithm, it behooves the analyst to start several times and compare results. Fortunately, in our data set, there is almost never any variation on the clusters ultimately chosen. (This is not the case when we depart from the Gower distance with specific variable types. In the other cases the blending that happens between groups introduces a stochastic element to the final cluster memberships.)

Figure 6.1 illustrates an example of the PAM applied to the hotel data with $k = 3$. The choice of $k = 3$ is made here to give us a concrete example—later we will explore approaches to estimating the "true" number of clusters in the data set. We also take this opportunity to introduce a custom chart summarizing clusters for these data. While we will defer a full exposition of the clustering of the hotel data until Section 6.2, it is worthwhile to spend some time orienting ourselves in the visualization of these clusters so that we can evaluate our future cluster decisions. Ultimately, three clusters may not be optimal to understand these data, but that number gives more interesting results than two clusters and allows easier decoding than a large number of clusters.

Figure 6.1 summarizes cluster membership across all the dimensions in our data. The figure is a complicated "dashboard" of the clustering results, divided into four sections. Note that this chart is best viewed in color, although it is possible to distinguish many of the features based on the different plotting points and slight grayscale differences in black and white.[2]

The topmost section indicates the number of total records per cluster with a stylized box plot. The thicker line denotes the inter-quartile range (IQR $= Q_3 - Q_1$). The thin line denotes the whiskers of the box plot ($Q_i \pm 1.5 * \text{IQR}$ where $i = 1$ or 3). The marked point is the median. We see, for instance, that Cluster 1, denoted by a hollow circle and a green color, has a relatively large number of records per cluster member although the spread is quite wide. Note also that

[2]The excellent Color Brewer [4] with attendant R package `RColorBrewer` makes it easy to choose good colors for both color and BW printing.

Figure 6.1: Summary data derived from the three cluster solution using the PAM algorithm. The figure is divided into four sections. The topmost indicates the number of total records per cluster with a stylized box plot. See the text for a description. The next section gives point estimates for the mean number of events by cluster for four different types of events. The third section summarizes clicks, again with means. Note the differing x-axes in these middle two sections. Finally, the last section depicts four values on a 0-1 scale: the percent of observations in the cluster, the portion that are display and search only, and the portion that have both search and display.

the x-axis for the records is indicated below the last cluster box plot and ranges from 1 to 58 records.

The next section gives point estimates for the mean number of events by cluster for four different types of events: Text, Display, Flash and Search. The vertical position of the points has been jittered to enhance separation between points. Again there is a custom x-axis for this section (ranging from 0 to 38.6 events). Cluster 1 shows some Text/Search events and a large number of Flash events. Search events are a subset of Text events, so the agreement in volume between the two is expected. No clusters show any Display events, unsurprising given their scarcity in the data set.

The third section summarizes clicks, again using the mean value for the cluster as the summary statistic. Here the x-axis ranges from 0 to 2.35. Clicks are highly correlated with Search and thus we see high values for Cluster 2. Interestingly, Cluster 3 (comprising search and display converters) shows the highest average number of display clicks of and overall clicks.

Finally, the last section depicts four values on a 0-1 scale: the proportion of observations in the cluster, the proportion that are display and search only, and the pro portion that have both search and display. Cluster 1 is the smallest cluster, representing about 11% of converters. Cluster 3 is the largest at 72% of all converters. The final three variables partition the data set and it is notable that each cluster comprises just one group. Our example cluster, Cluster 1, is made up of those converters who both clicked on a search term and were exposed to display advertising. It is at this point that our custom distance matrix bears fruit. Without treating these final three variables as asymmetric binary variables, we find that these groups blend within our clusters. Marketing research, however, indicates that converters coming from different online media (search versus display, principally) are both quantitatively and qualitatively different. It is satisfying to see the clusters respect this natural relationship and it seems our partitioning algorithm gives a practically useful result to be discussed in greater detail below.

6.1.4 Hierarchical Clustering Approach

A class of clustering algorithms provides an alternative to partitioning methods. These are hierarchical clustering algorithms and, as the name implies, they enforce a hierarchy on the data, attempting to form the equivalent of a taxonomic tree for the data in hand. Hierarchical clustering algorithms produce clusters of every size from 1 to N, the number of observations in the data set. The analyst must decide where to "cut" the tree and form clusters. This process is the equivalent of choosing the number of clusters in a partitioning algorithm. Hierarchical clustering algorithms come in two flavors depending on whether one splits clusters (divisive clustering) or starts with every observation in a cluster of one and begin joining the clusters (agglomerative). Agglomerative clustering has been studied much more extensively and thus we focus on that approach in this section.[3]

In agglomerative clustering, we begin with N clusters. Over the next $N-1$ steps, clusters are successively merged. The algorithm, therefore, depends critically on the ability to define distances between clusters. Let K_1 and K_2 denote two clusters (with K total clusters). We wish to define a distance metric $d(K_1, K_2)$ to represent the distance between the two groups. There are a profusion of techniques; we will mention only a few. The single linkage method defines the distance between the clusters to be the minimum distance between any two members:

$$d_s(K_1, K_2) = \min_{x_i \in K_1 \ x_j \in K_2} d(x_i, x_j).$$

The opposite of the single linkage is the complete linkage which takes the maximum of the distances.

$$d_c(K_1, K_2) = \max_{x_i \in K_1 \ x_j \in K_2} d(x_i, x_j).$$

In the parlance of the clustering literature, single linkage tends to form long clusters and is

[3]We tested divisive clustering, using R's function `diana`. The results were not useful. The algorithm begins with every observation in the same cluster and then splits to create new clusters. With the hotel data splitting produced new clusters of only one observation, so that when $k = 8$ we had one cluster with over 99% of the observations.

prone to "chaining". Complete linkage tends to form compact circular clusters. The final distance method we try, average cluster linkage, is defined as

$$d_a(K_1, K_2) = \frac{1}{N_{K_1} N_{K_2}} \sum_{x_i \in K_1} \sum_{x_j \in K_2} d(x_i, x_j)$$

and attempts to compromise between the single and complete linkage. Average linkage is the most popular choice for hierarchical clustering (although there has been a rise of "flexible" techniques that attempt to choose a linkage method based on ones data). The typical concern in the literature, echoed, for instance, in Hastie and Tibshirani [16], is that bad clusters can be formed if the data are not put on a common scale.

All three of these methods give results that exactly match the PAM algorithm for $k = 3$. Figure 6.2 displays the summary data for the three cluster solution using single linkage. It is satisfying that aggolmerative clustering with three clusters exactly matches the PAM results (this was not the case with divisive clustering).

At $k = 3$, complete and average linkage give clusters that match our previous two approaches. For $k > 3$, however, we see a weakness in these methods. In table 6.1.4 we summarize the number of observations by cluster for the eight cluster solution for single linkage, complete linkage and average linkage, a method we discuss below. Percentages for PAM are also included as a basis for comparison. The table is organized so that the largest cluster has the smallest number, so the cluster size decreases as we move down the table. PAM yields reasonably-sized clusters; the smallest two (10.5% and 6.25% of converters respectively) are not particularly small. The hierarchical methods have only 5 or fewer observations in the smallest two clusters for all three methods. In short, we see nice results for a small number of clusters such as $k = 3$. Once the number of clusters grows, however, we see that all methods make very small clusters beyond the initial three. This is unsatisfactory since the goal of clustering analysis in this case is to build useful groups of users from which we can infer interesting behavioral information. Going forward we will focus on partitioning-based methods.

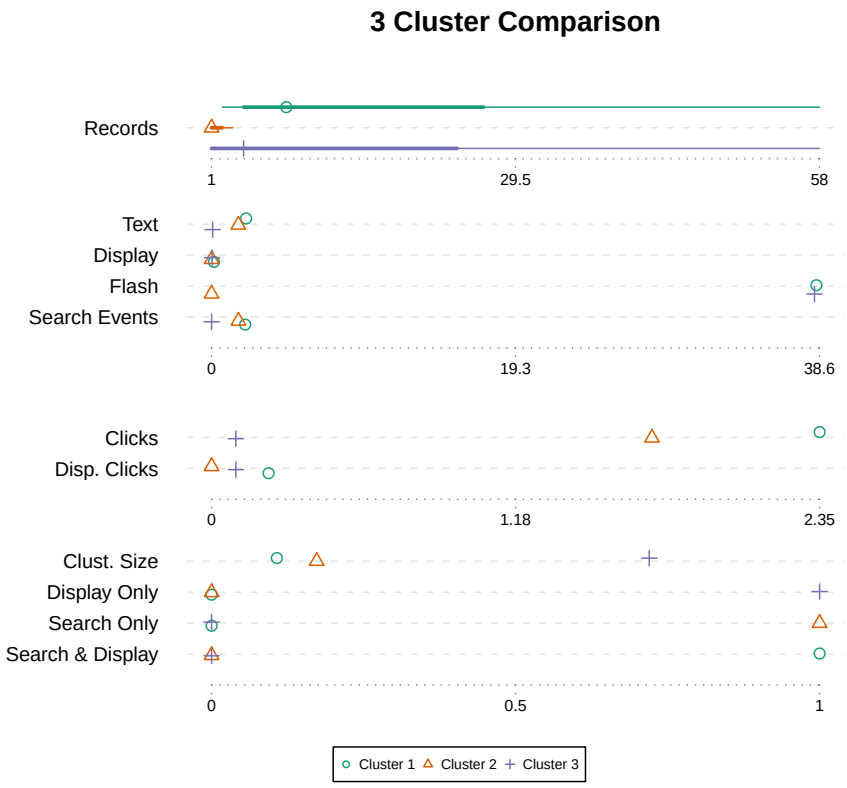

Figure 6.2: Summary data derived from the three cluster solution using aggolmerative clustering. See Section 6.1.3 for a description of this chart type. Results are an exact match with those of Figure 6.1 with the display only, search only, and search and display variables acting as markers for the three clusters.

	Percent of Observations			
	Hierarchical Aggolmerative			Partitioning-based
Cluster Number	Single	Complete	Average	PAM
1	71.950	71.733	71.733	18.900
2	17.250	17.250	17.250	15.350
3	10.700	10.517	10.517	13.950
4	0.033	0.217	0.217	13.333
5	0.017	0.133	0.133	11.033
6	0.017	0.083	0.083	10.733
7	0.017	0.033	0.033	10.450
8	0.017	0.033	0.033	6.250

Table 6.2: This table gives the number of observations per cluster with $k = 8$ for four different clustering algorithms. The clusters are sorted so that larger clusters have smaller cluster numbers. For the hierarchical clustering allocation, only the first three clusters have any meaningful size. In contrast, the partitioning method gives a reasonable allocation of users to all eight clusters. In fact, for PAM, the ratio of the largest cluster size to the smallest is 3. For complete-linkage, that ratio is 2152.

6.1.5 The Gap Statistic

The Gap statistic [33] was developed to allow an estimation of the number of clusters in a data set by comparing the clustering of reference data sets to the clustering of the actual data set. Here we will give the heuristic argument so that the reader may understand the method. The critical idea is to modify an *ad hoc* approach to estimating the number of clusters: the scree plot. The scree plot is based on the within-cluster dispersion, called W_k for k clusters. The statistic W_k is easy to calculate: within each cluster, add the squared distances between each observation and the centroid and then sum the results for every cluster. Although we cannot directly apply this method to the PAM clustering, the obvious extension is to sum the within-cluster dissimilarities. Both the within-cluster sum of squares and the within-cluster sum of dissimilarities decrease monotonically as k increases—similar to the reduction in residual sum of squares in regression with an increase in the number of explanatory variables—although the curve tends to flatten out for some k and in some contexts. As stated in [33], "statistical folklore has it that the location of such an 'elbow' indicates the appropriate number

of clusters." The Gap statistic attempts to formalize that heuristic.

The Gap statistic attempts to normalize W_k by comparing it to a reference distribution based on the (strongly) null hypothesis of a uniform distribution of the variables across their range[4]. The following is the algorithm for the computation of the Gap statistic for interval-scaled data:

1. Cluster the observed data, varying the total number of clusters from $k = 1, 2, \ldots, K$, giving within-cluster dispersion measures $W_k, k = 1, 2, \ldots, K$.

2. Generate B reference data sets using one of two methods described below. For each of these, cluster the generated data into $k = 1, 2, \ldots, K$ clusters and derive dispersion measures $W^*_{kb}, k = 1, 2, \ldots, K$ $b = 1, 2, \ldots, B$ and compute the estimated Gap statistic

$$\text{Gap}\,(k) = (1/B)\sum_b \log(W^*_{kb}) - \log(W_k).$$

3. Calculate the standard deviation of the Gap statistics and call this number sd_k. Define $s_k = \sqrt{1 + 1/B}sd_k$. Finally, choose the number of clusters via

$$\hat{k} = \text{smallest } k \text{ such that } \text{Gap}\,(k) \geq \text{Gap}\,(k+1) - s_{k+1}.$$

   In other words, choose the number of clusters such that the Gap for k is larger than the Gap for $k + 1$ minus one standard deviation.

Tibshirani, et al. [33] propose two ways of generating the reference data sets. The first, simpler way is to draw new values uniformly from a p-dimensional box bounding the observations. The second method is similar but aligns the box with the principal components of the data. The performance of the methods is similar although the second method performs somewhat

[4]This null hypothesis is particularly strong because it ignores any distributional information about the data except for the range and additionally ignores any covariance information between the variables.

better when the clusters are ellipsoidal in shape.

One benefit of the Gap statistic approach to the number of clusters problem is that it can be applied with a variety of clustering algorithms. The original paper makes use of both k-means and hierarchical clustering methods, though an extension to PAM is immediate if we modify our definition for W_k and W^*_{kb} from the within-cluster sum of squares to the within cluster sum of dissimilarities.

Calculating the Gap statistic is very computationally costly on data sets of the size we deal with. With the hotel data, our matrix of converters is 7318×11. Each reference data set is the same size and requires the creation of a distance matrix so that clusters can be made of all k that we test. Nevertheless, as we see in the next section, we apply the Gap approach to the hotel data with interesting, albeit mixed, results.

6.2 Clustering the Hotel Data with PAM

In this section we apply the approaches outlined in the previous section to the hotel advertiser data summarized in section 6.1.1. We first explore a partitioning clustering algorithm using the traditional heuristic approaches to determining the number of clusters.

We investigate a number of clustering solutions depicted in Figures 6.3,6.4, 6.5, 6.6, and 6.7. Each figure illustrates the results from a different number of clusters with $K \in (1, 2, 3, 4, 8)$ respectively. (The use of $k = 1$ is non-traditional but carried out to afford us a view into the baseline measurement for the entire data set.) The next set of clusters build upon each other and we will talk about each in turn.

Figure 6.3 calibrates our understanding of the data. A detailed explanation of the structure of this chart appears with Figure 6.1 and in Section 6.1.3. As we see from the top box plot, there

is a wide variety in the number of records. At least 25% of converters have only 1 record, the median is 3, and the upper hinge ranges to 38. As seen in Table 6.1.1, the mean is 32 and the maximum value is in the thousands. The vast majority of events are Flash with the average number of search events being 0.5 though the median is 0. Display clicks are rare—only 5% of converters show non-zero values. Finally, we see that 72% of converters are display only, 17% are search only and the remaining 11% have both search and display in their histories. The vast majority of display converters do not click on an advertisement. These last three groups will form the foundation for our discussion of the clusters to come.

The two cluster solution, Figure 6.4, seems imperfect. We see a split between low-volume search converters, Cluster 1, and high-volume display converters. Those converters with both search-and-display are split between clusters with two-thirds falling into Cluster 2. Investigating further, we find that those search-and-display converters who fall into cluster 1 have either 2, 3, or 4 records with fewer clicks and almost no display clicks. Converters in cluster 2 have a minimum of 5 records, a median of 17 records and a mean of 65 records. Since search-only converters tend to have many fewer records than display-only converters, we see a bifurcation of those sharing both traits into high and low volume.

The three-cluster solution is the first partition that seems to satisfy the natural grouping in the data, placing the three types of converters in separate clusters. The split between groups is as follows: cluster 1 holds display-only converters; search-only converters are in cluster 2; and the converters with both search and display activity are in cluster 3. Since these groups define the clusters exclusively, the rest of Figure 6.5 simply provides descriptive statistics on these groups. Notice, however, the relatively large spread in the number of records for clusters 1 and 3. We will see cluster 1 divide in subsequent clusterings. Unsurprisingly, clusters 2 and 3 show the most clicks. Interestingly, cluster 3 has the largest number of display clicks—cookies who click on search ads appear to be more likely to click on display ads as well.

The four-cluster solution (Figure 6.6) splits the display-only cluster from Figure 6.5 into low-

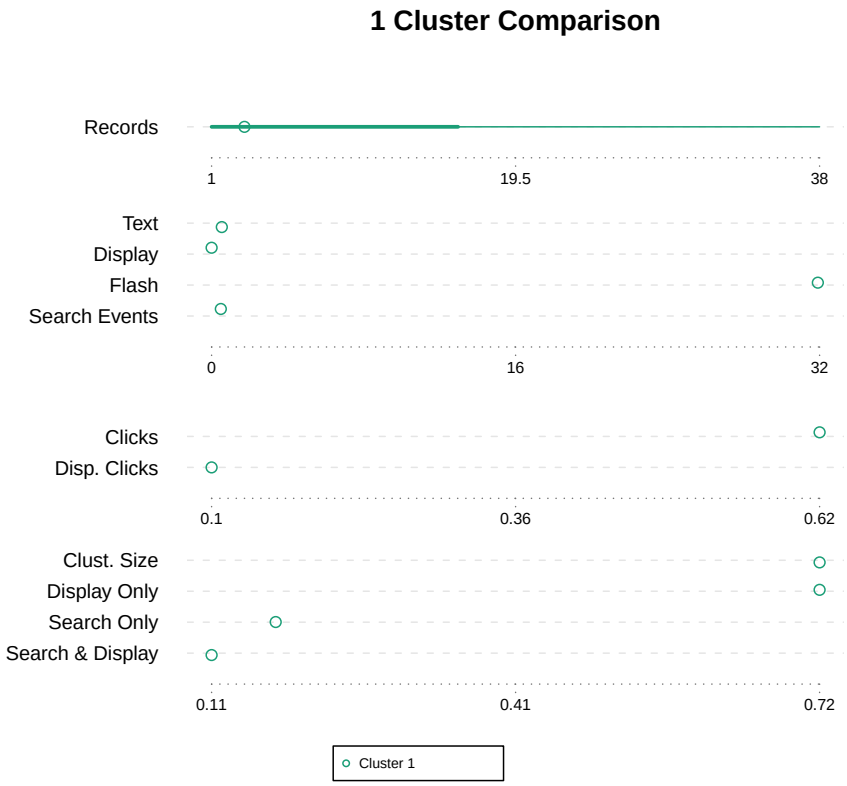

Figure 6.3: We reprise the data display in Figure 6.1. This graph depicts the data in one "cluster", affording us a view into the baseline measurement for the entire converter data set. We see a wide range of records in the boxplot at the top of the chart; most of the records are Flash as we see in the next section's dot plot. There are 0.6 search clicks on average and 0.1 display clicks across the converters. Most converters (72%) are "display only" converters with search only composing the next 17.5%.

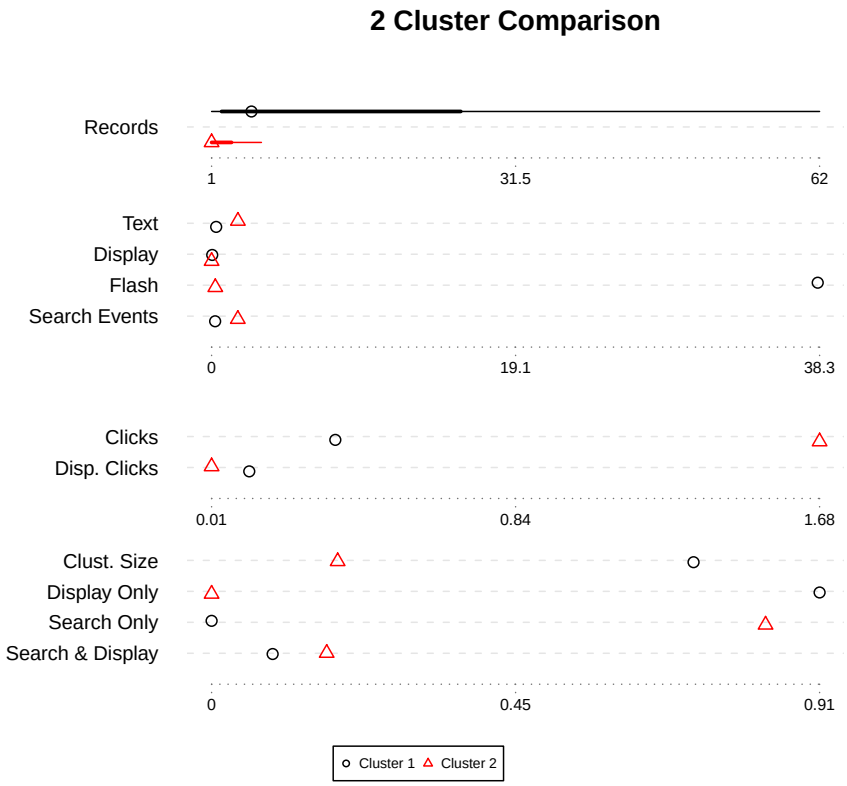

Figure 6.4: This figure depicts the two-cluster solution. We see a split between low-volume search converters in Cluster 1 and high-volume display converters. Those converters with both search and display are split between clusters, with converters having higher volume or display clicks being assigned to Cluster 2.

Figure 6.5: The three-cluster data summary shows a substantial refinement on Figure 6.4. Now the three principal groups of converters—search only, display only, and search and display—are in separate clusters. The box plots indicate a wide range of records within the clusters, notably for the display only and search and display groups.

and high-volume converters. The low-volume, display-only cluster, Cluster 1, is defined by an average record count of 2.5, a maximum number of records of 8, and only 2.5% of converters having clicks. Cluster 2, in contrast, has a mean number of records at 96 (the median is 37). In this group, 9% of converters have clicks in their history and, because of the right skew of that distribution, the average number of clicks is 0.18 (compared to 0.04 for Cluster 1).

Although we do not display the figures for $k = 5$, $k = 6$, and $k = 7$, the results follow the pattern established with the four-cluster solution. Namely, we see a continual splitting of the clusters established in the three-cluster solution. With $k = 5$, the display-only group splits into three clusters, again based on volume. At $k = 6$, the search-only cluster splits into converters with only one search click and converters with multiple search clicks. This distinction is useful from a marketing perspective. Converters with only one search click in their history are very likely to use search in a purely navigational fashion, going to the search engine, typing in their term, clicking the paid link and converting in short order. Those with multiple search clicks are more likely to use search engines for research, and as such, the quality of the ad copy and the keyword penetration becomes more important. At $k = 7$, we see the display-only group splits into four groups. There is substantial agreement between this result and the eight-cluster solution. As we will see below, the GAP statistic supports the seven-cluster solution, so we will discuss this solution below.

Clustering the converters into eight clusters seems to strike a nice balance between a tractable solution with well-defined clusters and enough clusters to capture some of the richness of the data. We will describe each cluster in turn. The graphical representation of the clustering can be found in Figure 6.7.

Cluster 1: This is the highest volume display-only cluster. 12.6% of converters fall into this cluster, which is notable for the large number of records (mean of 176, median of 101) amongst the members. A large percentage (12%) of the group has a display click and the mean number of clicks is 0.22. The converter with the fewest records in the group has 40 and

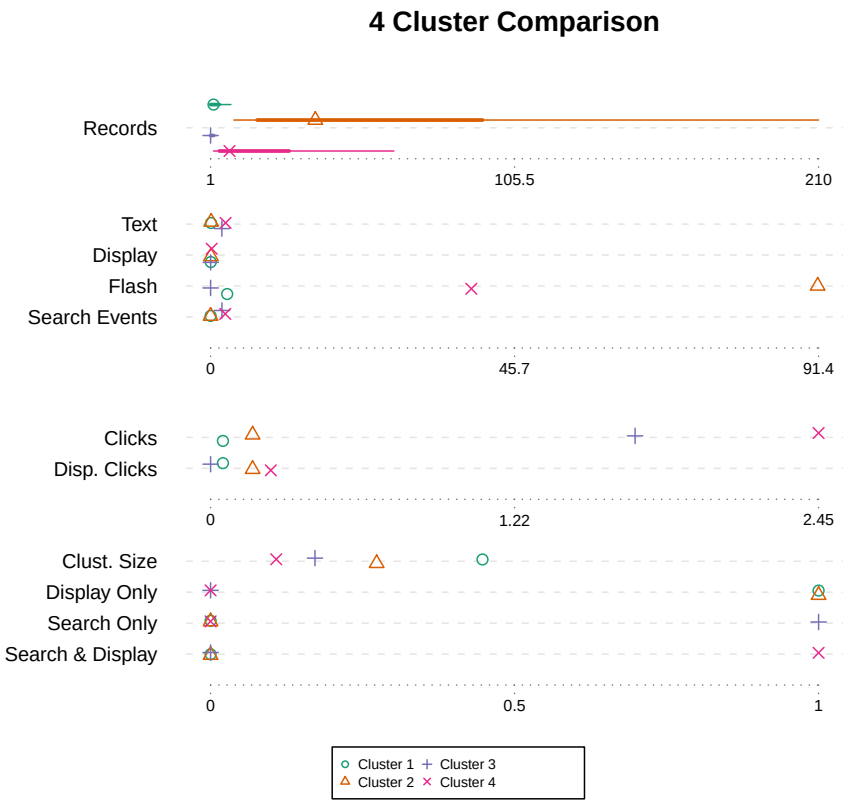

Figure 6.6: The four-cluster solution bifurcates the display-only cluster into low- and high-volume converters. The low-volume group, depicted in green with a circle, shows between 1 and 3 display views and very few clicks. The high volume group has an average frequency of almost 40 ads.

the middle 90% of the data ranges from 43 to 579.

Cluster 2: This is a lower volume display-only cluster with some click activity and with multiple records. Converters in cluster 2 have between 2 and 6 records in percentages of 40%, 21%, 17%, 12% and 11% respectively. The mean number of records is 3.3. Only 6% of the cluster have click (all display, naturally). 22.8% of our converters are in this cluster, making it the largest overall cluster.

Cluster 3: This is the low-volume search-only cluster. All the converters in this cluster, representing 11% of total converters, have exactly one search record. This is the larger of the two search clusters.

Cluster 4: This is the high-volume search-only cluster. Every search only converter with more than one record is in this cluster and the mean number of records is 3. The range is 2 to 20, though 90% of the cluster has fewer than 6 records. This group composes 6% of our total converters, making it half the size of the low-volume search only cluster.

Cluster 5: This cluster is one of the two search-and-display clusters. Whereas the search only and Display only clusters are primarily split by number of records, the search and display clusters are divided principally by the number of clicks by converters in the cluster. Cluster 5, with 8% of converters, averages 1.4 clicks. Most of these clicks are on Search (85%).

Cluster 6: Cluster 6 holds the converters who are display-only with a "medium high" number of records, 18 on average. The low is 6 records and the highest number of records in the cluster is 39. This cluster composes 17% of our total converters; 94% of these converters have no clicks.

Cluster 7: This cluster, composing 19% of our converters (making it the second largest cluster), is made up entirely of display-only converters with exactly one record. Similar to cluster 3, the data are very homogeneous with only 16 total clicks in the cluster. Unsurprisingly,

there seems to be a relationship between the number of ads that someone sees and the probability they will have a click in their history.

Cluster 8: This cluster is the second search-and-display cluster, for converters with an increased number of clicks. Interestingly, the average number of records (41) is almost the same as the average number of records for Cluster 5 (42) but the number of clicks is much higher. Cluster 8 has an average of 5 clicks whereas Cluster 5 had 1.4. The primary difference is in the number of search events: Cluster 8 seems to comprise converters who do multiple searches before arriving at a booking decision. Although this is the smallest cluster, with only 2.7% of converters, it seems to be one of the most amenable to marketing efforts—intelligent display advertising coupled with search marketing could move these users through the purchase funnel more quickly.

What happens beyond 8 clusters? To answer this question I investigated the results of clustering subsets of the data numerous times [5]. As we can see from the above discussion, the critical feature of clusters is the number of clusters in the three major groups of search only, display only, and search and display. Twenty data sets with 2000 observations were made from the full converter data. These sub-samples were put into clusters of with $k \in (2, 3, \ldots, 15)$. For each value of k, I calculated the percentage of observations in the three binary categories. I then took the average of the sum of percentages across the 20 replications. Table 6.2 holds the results.

For example, in all 20 trials when three clusters are formed, the data partition into search only, display only, and search and display. On the other hand, when seven clusters are formed, there are always two search-only clusters. 60% of cases have two search-and-display clusters. The other 40% of the sub-samples have four display-only clusters. (These percentages are taken from the decimal part of the seven cluster row in the table.) The "stability" of a clustering

[5]The entire data set that we have, which is itself a sample of 20% of converters, is too large to allow a full distance matrix to be held in memory. Therefore, we perform our clustering on the largest possible sub-samples and then assign additional observations to the clusters.

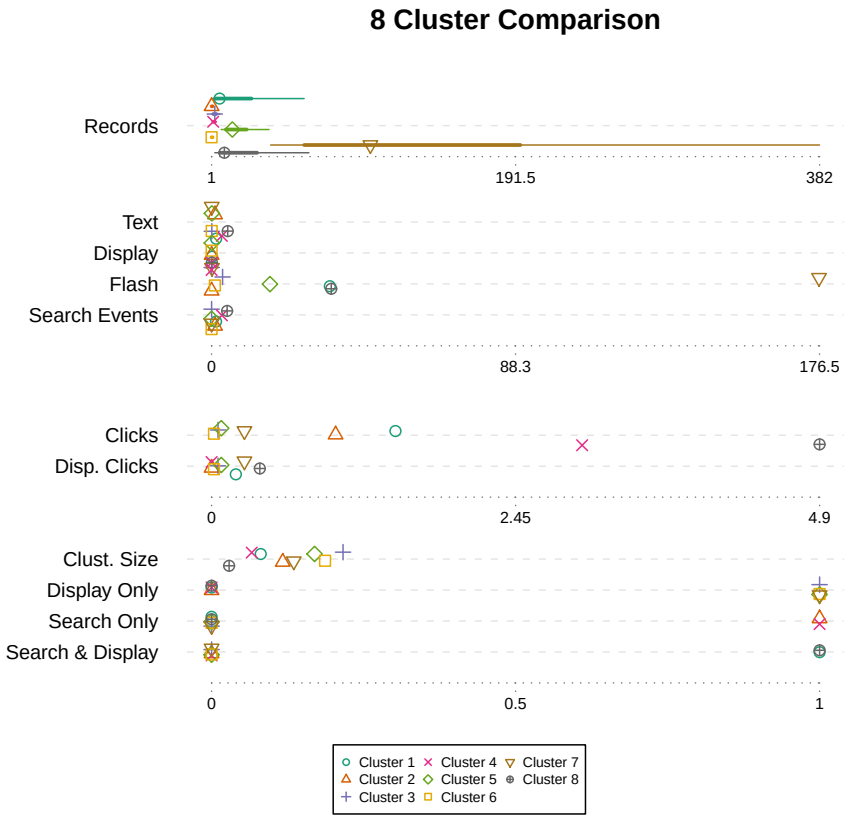

Figure 6.7: The result for 8 clusters appears substantially more complicated than the 3 cluster solution although the two are closely related. From our original 3 clusters we that the search-and-display group has split into two groups based on number of clicks. The search-only cluster has split into high- and low-volume groups, with the low-volume group having just one click. The display-only cluster has split into 4 smaller clusters based on volume of advertising consumed.

Num. Clusters	Search Only	Display Only	Search and Display
2	0.67	0.97	0.36
3	1.00	1.00	1.00
4	1.00	2.00	1.00
5	1.00	3.00	1.00
6	2.00	3.00	1.00
7	2.00	3.40	1.60
8	2.00	4.10	1.90
9	2.15	4.80	2.05
10	2.80	5.05	2.15
11	2.95	5.20	2.85
12	3.05	5.55	3.40
13	3.05	6.00	3.95
14	3.60	6.25	4.15
15	3.70	6.95	4.35

Table 6.3: Twenty data sets with 2000 observations were made from the full converter data. These sub-samples were put into clusters of with $k \in (2, 3, \ldots, 15)$. For each value of k, I calculated the percentage of observations in our three binary categories. I then take the average of the sum of percentages across the 20 replications. For example, with three clusters always yields the same separation. With 13 clusters there are six display-only clusters and 95% of the time we get four search-and-display clusters and three search-only clusters. In 5% of cases, though, there is an additional search-only cluster instead.

solution can be seen by looking at the decimal values. For instance, the eight-cluster solution seems more stable than the nine-cluster solution. Eight clusters always has two search-only clusters and then 90% of the time we have two search-and display clusters. Nine clusters always has four display-only clusters and two of the other categories. The final cluster in the nine cluster solution comes from an additional display cluster in 80% of cases. The remaining cases are largely search only (15% to 5%).

Table 6.2 shows what happens beyond eight clusters. Nine clusters introduces a fifth display-only cluster and the number of display clusters remains the same until 13 total clusters are formed. A third search-only cluster is formed at around 10 or 11 total clusters. Search and display seems more fluid, as we have a smoother transition from two (at eight total clusters) to three (between 11 and 12) and four (around 13 total clusters).

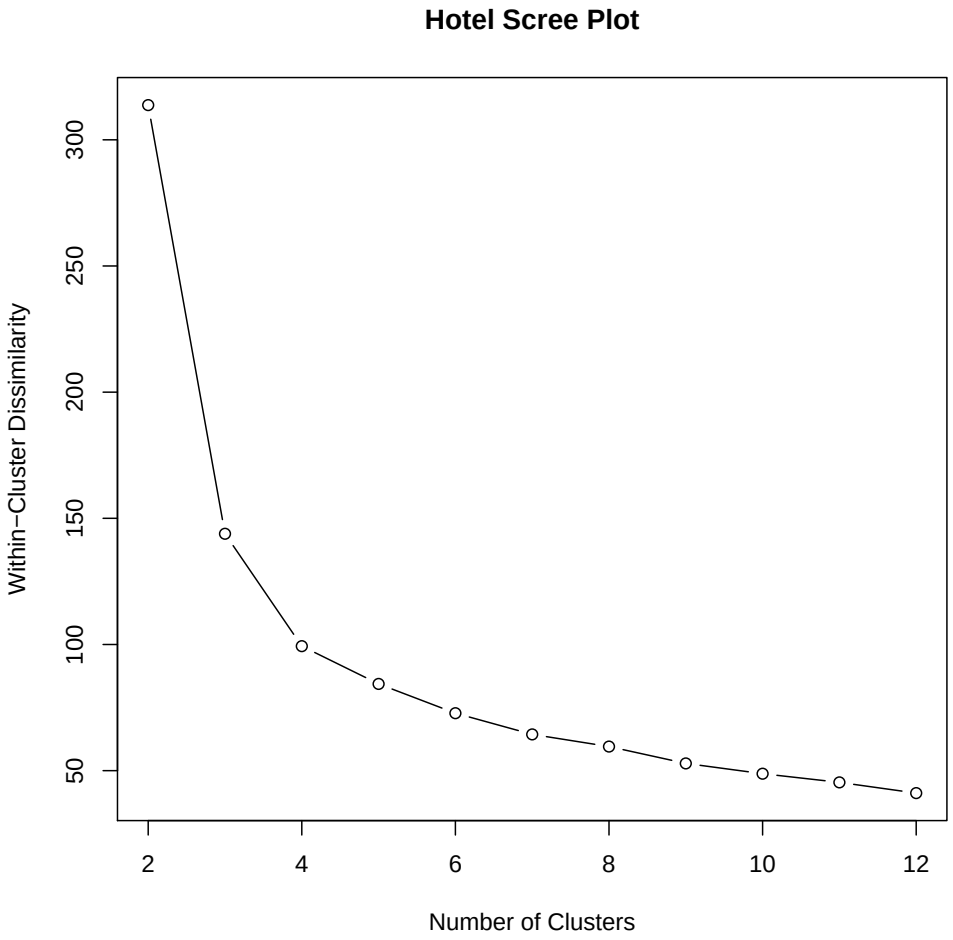

Figure 6.8: The scree plot for the hotel data. Subsection 6.1.5 contains a description of how this plot is formed. A potential elbow appears at $k = 3$ or $k = 4$.

6.2.1 Estimating the Hotel-Data Clusters with GAP

In Subsection 6.1.5 we discussed both the concept of the scree plot (which we hope will have an "elbow," indicating the appropriate number of clusters) and also the GAP statistic, a resampling approach to estimating the number of clusters by comparing the scree plot to a reference scree plot formed on resampled versions of the data set. We have created both types of plots for the hotel data. They are found in Figures 6.8 and 6.9.

The scree plot shows a potential elbow at $k = 3$ or $k = 4$ although it is not clear. This figure illustrates the shortcomings of the scree plot heuristic that the gap statistic strives to overcome. We present this figure largely for illustrative purposes and will not devote any time to divining the mysteries in its tea leaves.

The GAP statistic values are illustrated in Figure 6.9. The criterion from the original paper [33] for the "optimal" number of clusters is the first value along the curve that is higher than the subsequent point's single standard error lower bound. The first value where this occurs is $k = 7$, indicating support for the seven-cluster solution using the GAP statistic. When we consider the data in Table 6.2, however, we see that forming seven clusters results in instability in the formation of the seventh cluster. In approximately 40% of cases the seventh cluster is an additional display cluster. The remaining 60% of cases create the seventh cluster by forming a second search and display cluster. As mentioned above, moving to eight clusters allows us to side-step this dilemma.

6.3 A Statistical View

This chapter is a relatively straightforward application of a number of clustering concepts to online advertising data, with some extensions made to the GAP statistic to allow estimation of the optimal number of clusters in a data set with mixed variable types. Our goal is to understand converter behavior through clustering. As discussed in Chapter 5, the data created by online advertising can be difficult to summarize. There is a great deal of variability in both the number of records and the types of records that users generate. Whereas a traditional approach to understanding a subset of observations might involve plotting some basic data and calculating summary statistics, in marketing that is insufficient. Knowing that the average converter is exposed to 32.09 marketing messages does not increase understanding of the path that the converter takes to a conversion. Nor does it give those who manage campaigns an

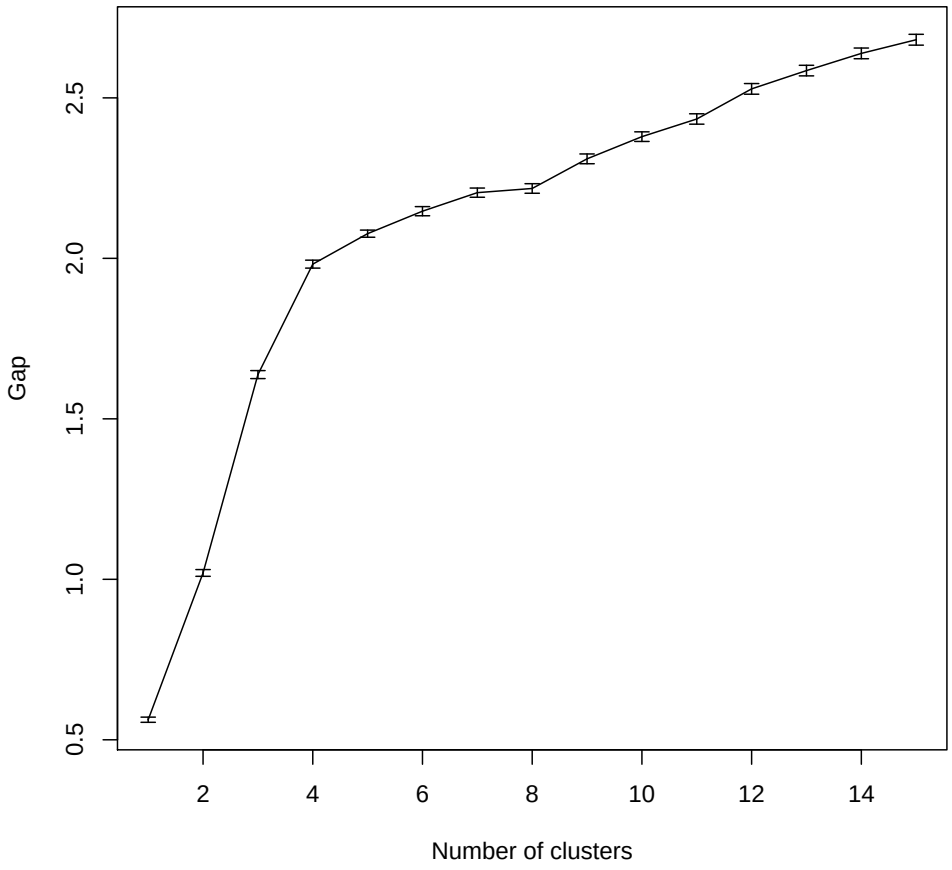

Figure 6.9: The GAP plot for the hotel data. The criterion for the "optimal" number of clusters is the first value along the curve that is higher than the subsequent point's single standard error lower bound. The first value where this occurs is $k = 7$, indicating support for the seven-cluster solution using the GAP statistic.

indication of how to turn more non-converters into converters. Throughout this chapter we work with data called the "hotel data set", based on marketing data from late 2008 for a hotel advertiser.

Our first step is data reduction. Since cookies vary in the number of records they generate, we summarize the data using a variety of simple measures. A group of three measurements, search only, display only, and search and display are important. Converters in the search only group are exposed only to search advertising. Similarly, the display only group sees only display advertising, occasionally clicking. The search and display group has both types of advertising in their history. Since, by definition, the term "converter" means those who are exposed to advertising before taking a desired action, these three groups partition the converter record set. The creation of the data set in Subsection 6.1.1 requires a concomitant creation of a distance matrix. This is described in Subsection 6.1.2. This section is notable because we depart from a traditional L_1 or L_2 norm, instead making use of the Gower distance matrix that attempts to give all variables equal weight and that makes use of the important distinction between symmetric and asymmetric binary variables.

The next two subsections discuss partitioning- and hierarchical-based clustering algorithms and apply them to the representative data set from this chapter. The partitioning-based clustering subsection gives an in-depth summary statistical visualization for the clusters in the chapter. This type of figure is used extensively throughout the chapter. Section 6.1 closes by discussing the GAP statistic, a resampling technique that allows us to estimate the number of clusters in a data set. The original 2001 paper [33] uses a variation of parametric bootstrap resampling (of a multivariate uniform distribution). Here we modify that technique to use a bootstrap resampling approach that is appropriate for both the continuous data in the original paper and for categorical or mixed data. This is the first time this extension of the GAP statistic has been developed in any context. For social scientists this modification represents a substantial improvement on the state-of-the-art for the number of clusters question. Typically social science data includes both continuous data (e.g., the length of time between exposure

and conversion) and categorical data (e.g., our search and display binary categories). The GAP statistic cannot be used with these data sets without the modifications proposed here.

Section 6.2 provides a lengthy application of the concepts of the previous chapter to the hotel data. We find structure in the converter data, determining eight clusters that show both good stability when subsamples of the data are re-clustered and tractable interpretations that can aid future marketing. In particular, we find that converters clustering into four groups receiving display impressions only (spread out based on volume of impressions and the presence of clicking behavior), two groups receiving search only advertising (split based on whether the converter had only one search or multiple searches), and two groups of converters with both search and display advertising. These last two clusters have a similar number of overall records, but are divided based on the number of clicks in their history, indicating that one of the groups is doing a great deal more research before purchasing. By identifying these types of users before they convert, marketers could message to them differently and theoretically move them through the purchase funnel more quickly.

Chapter 7

Conclusion

The online advertising and literature reviews we began with are necessary but not hard going from a research perspective. The journey to arrive at our central three chapters on visualization, PHMs, and clustering has been arduous. In an original draft I devised an *ad hoc* way of measuring the prediction error of a fitted E-Map model. The PHM results replace those. Before this research there were no methods for fitting conversion attribution models and as I began the dissertation I was driving towards the goal of putting the methodology of Chapter 5 on firm footing. The technique of applying PHMs with our time-varying covariates for every media exposure finally captures the ideas of E-Map in a fully rigorous setting. Survival analysis is typically employed to estimate the time of an event. The beauty of the PHM methodology is that it does not become fixated on predicting the actual time but instead gives us a way to measure the impact of covariates.

The primary enhancements we derive with these PHMs as applied are as follows:

- We improve on the methods of Manchanda and Dubé through the use of the time-varying covariate framework. Unconstrained from the shackles of summary-only data, we allow

each media exposure to influence conversion, building these into a coherent model of response.

- PHMs are relatively fast. Although some of the model fitting still requires the use of 64-bit machines, typical execution time for estimating a cookie's conversion probability is in the realm of a few seconds. With the E-Map prediction methodology of earlier chapters this estimation took tens of minutes. This increase in speed has important implications for the ultimate application of conversion attribution to targeting of ads. In Section 4.6 we extensively detail the modifications necessary to the statistical software package R§-Plus in order to estimate survival probabilities on extensive data sets.

- PHMs allow rigorous fitting of *ad hoc* models determined by analysts. With a well-fit PHM we can estimate the impact of adding additional ads in a user's history or changing all ads from a cheaper format like display to a more expensive format like rich media. At last an accurate cost-benefit analysis of media types can be performed.

My future work in this area will continue down this line. First, there are gaps in the statistical PHM implementation in both S-Plus and SAS. For instance, the use of case-weights is not supported. This is relatively unimportant in medical studies, since sample sizes are often small and study analysis time is freely available. Online, however, we need the ability to generate model estimates across many advertisers based on a relatively small sample of total converters and non-converters. In this context, using case weights would be advantageous to synchronize the model estimates with true conversion rates. Additionally, the fitting techniques in the `survival` package are ill-suited for extremely large sample sizes. I plan to pursue an implementation of the functions in this package that divorces them from the standard memory constraints.

From a marketing standpoint, an obvious area for further work involves a more rigorous exploration of the recency results, particularly across multiple advertisers. The concept of recency is well-documented in observational studies in labs: people remember almost anything

better if they have seen it more recently.

One of my advisors, Brian Steele, has inculcated in me the idea that for the dissertation, the journey is the destination. Having been through the process, he is undoubtedly correct, although I am gratified that the actual destination seems to be worthwhile as well.

Glossary of Advertising Terms

Action (also Action Tag) An action is behavior on an advertiser's website that the advertiser wishes to track. This tracking is done using cookies and is carried out using an Action Tag. Tags are placed on pages that the advertiser wishes to track. Actions are used to define Conversions. An Action is any triggering of an Action Tag, whereas a Conversion is the triggering of an Action Tag by a Cookie that has previously been exposed to Advertising.

Advertiser In the context of this dissertation, the term Advertiser stands for companies who conduct business online and advertise online. Advertisers are responsible for paying publishers to run media. Advertiser's maintain websites where consumers transact and the advertiser will typically use a third-party ad server to track online behaviors. Examples of advertisers would be Best Buy, Nike, and Bank of America.

Ad (Advertisement) The combination of the creative (the actual pixels that constitute a display ad or the copy that makes up a search ad) and the click-through URL for the ad (the web address the consumer is taken to after clicking on the ad.)

Atlas Atlas is a division of Microsoft Advertising and my employer. Atlas functions within the online advertising industry as a third-party ad server.

Attribution Model A model that marketers to assign credit for a conversion to the advertising a user is exposed to.

Baseweight Baseweights are parameters within Engagement Mapping models. Baseweights determine the amount of credit that goes to different types of media or to different actions. Examples of advertising events that might have different baseweights include: Display, Text, and Video impressions; clicks; and interactions with Rich Media ads.

Click For our purposes, a click is considered the act of a user interacting with an advertiser's ad in a way that directs the user from the publisher's website to the advertiser's website. (The technical definition of a click is surprisingly complicated.)

Conversion An action that can be tied to advertising. Actions are typically behavior on a website that an advertiser wishes to measure such as sales or registrations. If these actions are preceded by advertising within the conversion window then the action is considered a conversion.

Conversion Rate The rate of conversions per unit of advertising. Typically this is expressed as a decimal and defined as conversions per impressions.

Conversion Window The conversion window is the length of time, from an action, a third-party ad server will look backwards for media that will turn the action into a conversion. Typically conversion windows for ad views are shorter than those for ad clicks. Common conversion windows are seven days for views and 30 days for clicks. Media that fall outside the conversion window will not receive credit for causing the conversion.

Cost-Per-Mille (CPM) The pricing structure for most display advertising. Advertisers pay for impressions by paying a given dollar amount per 1000 impressions served on their behalf by the publisher.

Cost-Per-Click (CPC) The pricing structure for all search advertising and for some display advertising. The advertiser pays the publisher a flat rate for every click of an ad.

Cookie Small text files stored on a user's computer. Typically these are simply long somewhat random numbers that identify the computer and browser for the purposes of anonymous tracking. Cookies are, for instance, the way Amazon remembers who you are when

you come back to the site and can thereby build a custom homepage for you. Cookies are the unique identifier for a computer. Multiple people can use one computer (and hence have one cookie) and cookies can be deleted. These records are imperfect but they are the best, easily-accessible method of identifying people

Creative Type The type of ad shown to a user. Plain JPEG or GIF ads are called display. Other creative types include Video, Rich Media, Flash, Java, and Text ads.

Display Advertising A type of advertising that involves showing images on publisher's websites. Defined in contrast to Search Advertising.

Engagement Mapping (E-Map) E-Map is a flexible conversion attribution framework, developed by Atlas, that allows advertisers to define a custom model that shares credit across all marketing messages delivered within the conversion window.

Flash (also FlashJava or Java) A creative type based on the languages Flash or Java. Typically these ads are more interesting to consumers than basic GIF or JPEG ads.

Impression The display of a single advertisement on a publisher's website. Often an impression is defined as an "opportunity to see" a given ad. Impressions are the basic unit of display advertising.

Gross Ratings Points (GRPs) A measure of ad viewership. GRPs are defined as reach (the number of people who see an ad) times frequency (the number of times people see an ad) divided by audience size and multiplied by 100. For example, 100 GRPs mean that there were exactly as many ad impressions as there were people in the target audience (although we are not ensured that each person received exactly one ad).

IP Address This is the internet address of where the record request came from. Using this information we can determine (in most cases) geographically where the user is located, their connection speed, and whether they are surfing from home or work.

Last-ad Model A conversion attribution methodology where the most recent ad seen gets 100% of the credit for an action unless there is a click, in which case the click gets all the credit. All clicks or impressions must fall within the conversion window.

Log Records These are the data we record for a cookie. Log records include both web surfing data (such as impressions and clicks) and data from the advertiser's website: actions.

Order The order parameter in an Engagement Mapping model attempts to measure the influence between clicks. Specifically, it modifies the value of a click based on the presence of a previous clicks.

Placement The location where an ad runs. For display advertising placements are typically locations on a page, although some placements span multiple locations (e.g. , a placement that allows a banner to be shown anywhere within the Finance section of Yahoo!). For search advertising we can think of placements as being keywords that are purchased by the advertiser. For instance, eBay might purchase "used DVDs" as a keyword.

Publisher The counter-party to advertisers, publishers create websites. These websites attract people and publishers charge advertisers for the opportunity to show ads to the people on the publisher's site.

Recency Measures how the impact of ads diminishes over time. Typically recency is modeled with an exponential decay curve (called in the literature the "forgetting curve"). Ads that are more recent have a larger influence over consumer behavior.

Rich Media A creative type. Rich Media refers to advertisements made with a variety of technologies. Rich Media ads are substantially more interactive than Flash, Java, JPEG or GIF ads and may include games, homepage takeovers, and ads that interact with the elements of a publisher's webpage.

Search Advertising A type of advertising that involves buying keywords from a search

provider. Advertisers pay a fee to have their text ads shown on the page when the keyword is searched on.

Size (also, Ad Size) The size of an advertisement, measured in square pixels. The most common ad size on the internet is the 468x60 banner which is 468 pixels wide and 60 pixels tall for a total ad size of 28080 pixels squared.

Third Party Ad Serving (TPAS) Advertisers employ a third-party ad servers to deliver and track ads across the internet. The TPAS are responsible for the technological infrastructure that enables the ad-serving relationship as well as functioning as a trusted third-party maintaining the accounting of the advertising system.

Tentative A field in the cookie record. This field is 1 if we have one and only one record for the cookie in their entire history. Typically this happens when someone rejects our cookies and so every time we see the person it is for the first time. In most cases cookies with Tentative=1 are excluded from analyses and that is the case for every analysis we discuss henceforth.

Text Link A type of advertisement that involves turning text on a webpage into a hyperlink that can take the consumer to an advertiser's website. Typically text link impressions have the lowest impact on conversion rates, although they can generate a large number of conversions through clicks. Search ads are a particular type of text links.

Video A type of ad that involves showing a video ad (similar to a television chimerical) online. Ads can run by themselves or adjacent to other video content.

Code

```
////////////////////////////////////////////////////////
//              CLUSTERING FUNCTIONS                  //
////////////////////////////////////////////////////////

require(cluster)
require(gtools)
require(Hmisc)
require(RColorBrewer)

plot_clusters = function(the_data,the_clusters) {
    # This plots the relevant summary stats
    # for data with clusters split out.
    # Fields:
    #   "cookie"          "RecordCount"     "Text"            "Clicks"
    #   "IsConverter"     "DisplayClicks"   "DisplayOnly"     "NonSearch"
    #   "FlashJava"       "SearchOnly"      "SearchEvents"    "Display"

    # Idea. Plot means for everything first.
```

```
# We're going to do a carefully constructed semi-dot plot

uni_clusters = sort(unique(the_clusters))
num_clusters = length(uni_clusters)

pch_vals = c(1,2,3,4,5,0,6,10,19,15)

if (num_clusters == 2) {
    col_vals = c("black","red")
} else if (num_clusters <= 8) {
    col_vals = brewer.pal(num_clusters,"Dark2")
} else {
    col_vals = brewer.pal(num_clusters,"Set3")
}

# Calculate the values to plot

# cluster sizes
cl_sizes = numeric(num_clusters)

for (i in 1:num_clusters) {
    cl_sizes[i] = mean(the_clusters==uni_clusters[i])
}

# boxplot of recs
recs_box_data = boxplot(the_data$RecordCount ~ the_clusters,plot=F)$stats
recs_means = numeric(num_clusters)
```

```
for (i in 1:num_clusters){
    recs_means[i] = mean(the_data$RecordCount[
        the_clusters==uni_clusters[i]])
}



# means for different types of events
event_means = data.frame(type=c("Text","Display","Flash","Search",
    "NonSearch"))

for (i in 1:num_clusters){
    event_means = cbind(event_means,0)
}
names(event_means) = c("type",uni_clusters)

event_means[event_means$type=="Text",2:(num_clusters+1)] =
    aggregate(the_data$Text,by=list(the_clusters),mean)$x
event_means[event_means$type=="Display",2:(num_clusters+1)] =
    aggregate(the_data$Display,by=list(the_clusters),mean)$x
event_means[event_means$type=="Flash",2:(num_clusters+1)] =
    aggregate(the_data$FlashJava,by=list(the_clusters),mean)$x
event_means[event_means$type=="Search",2:(num_clusters+1)] =
    aggregate(the_data$SearchEvents,by=list(the_clusters),mean)$x
event_means[event_means$type=="NonSearch",2:(num_clusters+1)] =
    aggregate(the_data$NonSearchEvents,by=list(the_clusters),mean)$x

# click data
```

```
click_means = data.frame(type=c("Clicks","DisplayClicks"))

for (i in 1:num_clusters){
    click_means = cbind(click_means,0)
}
names(click_means) = c("type",uni_clusters)

click_means[click_means$type=="Clicks",2:(num_clusters+1)]        =
    aggregate(the_data$Clicks,by=list(the_clusters),mean)$x
click_means[click_means$type=="DisplayClicks",2:(num_clusters+1)] =
    aggregate(the_data$DisplayClicks,by=list(the_clusters),mean)$x

disp_only_means = aggregate(the_data$DisplayOnly,by=list(the_clusters),
    mean)$x
search_only_means = aggregate(the_data$SearchOnly,by=list(the_clusters),
    mean)$x
s_and_d_means =
    aggregate(the_data$SearchAndDisplay, by=list(the_clusters),
    mean)$x

### Define the plotting area and plot
op = par(mar=c(5,10,4,2) + 0.1,cex.axis=0.8,las=2)

#axis labels
labs = c("Clust. Size","Records","Text","Display","Flash",
               "Search Events",
               "Clicks","Disp. Clicks",
               "Display Only","Search Only","Search & Display")
```

```
# Setting the heights manually
rc_h = 0.97

in_group_offset = 0.045

text_height = 0.83
disp_height = text_height - in_group_offset
flash_height = text_height - 2*in_group_offset
search_height = text_height - 3*in_group_offset

click_height = 0.52
dclick_height = click_height - in_group_offset



sz_h = 0.34
d_only_height = sz_h - in_group_offset
s_only_height = sz_h - 2*in_group_offset
s_and_d_height = sz_h - 3*in_group_offset

all_heights = c(sz_h,rc_h,text_height,disp_height,flash_height,
                search_height,
                click_height, dclick_height,
                d_only_height, s_only_height,s_and_d_height)
```

```
    # Some other vals that we use throughout
    reference_line_offset = -0.045
    reference_line_tick_length = -0.015

    if (num_clusters <= 4) {
        jitter_factor = 0.8
    } else {
        jitter_factor = 1.5
    }

    # Plotting
    plot(c(0,1),c(0,1),type="n",main=paste(num_clusters,
        "Cluster Comparison"),
        xlab="",ylab="",axes=F)

    # axes
    axis(side=2,at=all_heights,
        labels = labs,
        tick=F)

    abline(h=all_heights,lty=2,col="gray90")

    # size
    points(cl_sizes,jitter(rep(sz_h,num_clusters),factor=jitter_factor),
        col=col_vals,pch=pch_vals)

    # recs
    # Going to do a variation of the boxplot
```

```
    recs_box_range = range(recs_box_data)
    recs_box_scale = diff(recs_box_range)

    # horizontal reference line
    lines(x=c(0,1),y=c(rc_h,rc_h)+reference_line_offset,lty=3,col="gray45")
    lines(x=c(0,0),y=c(rc_h,rc_h+reference_line_tick_length)+
        reference_line_offset,
        lty=3,col="gray45")
    lines(x=c(1,1),y=c(rc_h,rc_h+reference_line_tick_length)+
        reference_line_offset,
        lty=3,col="gray45")
    lines(x=c(0.5,0.5),y=c(rc_h,rc_h+reference_line_tick_length)+
        reference_line_offset,
        lty=3,col="gray45")

    text(x=0,  y=rc_h+reference_line_offset+reference_line_tick_length*2,
        labels=paste(recs_box_range[1],sep=""),cex=0.65)
    text(x=0.5,y=rc_h+reference_line_offset+reference_line_tick_length*2,
        labels=paste(round(mean(recs_box_range),1),sep=""),cex=0.65)
    text(x=1,  y=rc_h+reference_line_offset+reference_line_tick_length*2,
        labels=paste(recs_box_range[2],sep=""), cex=0.65)

    recs_line_heights = seq(from=rc_h-reference_line_offset*
        (1-1/num_clusters),
        to=rc_h+reference_line_offset*(1-1/num_clusters),
        length.out=num_clusters)

    for (i in 1:num_clusters) {
```

```
    # Doing stylized tufte boxplots.
    low_whisker     = recs_box_data[1,i]
    twenfive_pct    = recs_box_data[2,i]
    med_val         = recs_box_data[3,i]
    sevenfive_pct   = recs_box_data[4,i]
    high_whisker    = recs_box_data[5,i]

    lines((c(low_whisker,high_whisker)-recs_box_range[1])/
          recs_box_scale,y=rep(recs_line_heights[i],2),
          lty=1,col=col_vals[i],lwd=1)
    lines((c(twenfive_pct,sevenfive_pct)-recs_box_range[1])/
          recs_box_scale,y=rep(recs_line_heights[i],2),
          lty=1,col=col_vals[i],lwd=2.5)
    points((med_val-recs_box_range[1])/recs_box_scale,
        recs_line_heights[i],
        col=col_vals[i],pch=pch_vals[i]) # was pch = 20

}

#Events
event_range = range(event_means[,-1])
event_scale = diff(event_range)

# Text
points((event_means[event_means$type=="Text",-1]-event_range[1])/
    event_scale,
    jitter(rep(text_height,num_clusters),factor=jitter_factor),
    col=col_vals,pch=pch_vals)
```

```
# Display
points((event_means[event_means$type=="Display",-1]-event_range[1])/
    event_scale,
    jitter(rep(disp_height,num_clusters),factor=jitter_factor),
    col=col_vals,pch=pch_vals)

# Flash
points((event_means[event_means$type=="Flash",-1]-event_range[1])/
    event_scale,
    jitter(rep(flash_height,num_clusters),factor=jitter_factor),
    col=col_vals,pch=pch_vals)

# Search Events
points((event_means[event_means$type=="Search",-1]-event_range[1])/
    event_scale,
    jitter(rep(search_height,num_clusters),factor=jitter_factor),
    col=col_vals,pch=pch_vals)

# Event reference line
event_ref_line_height = search_height + reference_line_offset

lines(x=c(0,1),     y=rep(event_ref_line_height,2),lty=3,col="gray45")
lines(x=c(0,0),     y=c(event_ref_line_height,event_ref_line_height+
    reference_line_tick_length),
    lty=3,col="gray45")
lines(x=c(1,1),     y=c(event_ref_line_height,event_ref_line_height+
    reference_line_tick_length),
```

```
    lty=3,col="gray45")
lines(x=c(0.5,0.5), y=c(event_ref_line_height,event_ref_line_height+
    reference_line_tick_length),
    lty=3,col="gray45")

text(x=0,  y=event_ref_line_height+reference_line_tick_length*2,
    labels=paste(round(event_range[1] ,1),sep=""), cex=0.65)
text(x=0.5,y=event_ref_line_height+reference_line_tick_length*2,
    labels=paste(round(mean(event_range) ,1),sep=""), cex=0.65)
text(x=1,  y=event_ref_line_height+reference_line_tick_length*2,
    labels=paste(round(event_range[2] ,1),sep=""), cex=0.65)



#CLICKS
click_range = range(click_means[,-1])
click_scale = diff(click_range)

# Clicks
points((click_means[click_means$type=="Clicks",-1]-click_range[1])/
    click_scale,jitter(rep(click_height,num_clusters),
    factor=jitter_factor),
    col=col_vals,pch=pch_vals)

# Display Clicks
points((click_means[click_means$type=="DisplayClicks",-1]-
    click_range[1])/
    click_scale,jitter(rep(dclick_height,num_clusters),
    factor=jitter_factor),
```

```
    col=col_vals,pch=pch_vals)

# Click reference line
click_ref_line_height = dclick_height + reference_line_offset

lines(x=c(0,1), y=rep(click_ref_line_height,2),lty=3,col="gray45")
lines(x=c(0,0), y=c(click_ref_line_height,click_ref_line_height+
    reference_line_tick_length),
    lty=3,col="gray45")
lines(x=c(1,1), y=c(click_ref_line_height,click_ref_line_height+
    reference_line_tick_length),
    lty=3,col="gray45")
lines(x=c(0.5,0.5),y=c(click_ref_line_height,click_ref_line_height+
    reference_line_tick_length),
    lty=3,col="gray45")

clk_range = range(click_means[,-1])

text(x=0,  y=click_ref_line_height+reference_line_tick_length*2,
    labels=paste(round(clk_range[1]    ,2),sep=""), cex=0.65)
text(x=0.5,y=click_ref_line_height+reference_line_tick_length*2,
    labels=paste(round(mean(clk_range) ,2),sep=""), cex=0.65)
text(x=1,  y=click_ref_line_height+reference_line_tick_length*2,
    labels=paste(round(clk_range[2]    ,2),sep=""), cex=0.65)


# display and search breakdown
ds_range = range(c(disp_only_means,search_only_means,s_and_d_means))
```

```
ds_scale = diff(ds_range)

# Display only
points((disp_only_means-ds_range[1])/ds_scale,
    jitter(rep(d_only_height,num_clusters),factor=jitter_factor),
    col=col_vals,pch=pch_vals)

# Search only
points((search_only_means-ds_range[1])/ds_scale,
    jitter(rep(s_only_height,num_clusters),factor=jitter_factor),
    col=col_vals,pch=pch_vals)

# S & D
points((s_and_d_means-ds_range[1])/ds_scale,
    jitter(rep(s_and_d_height,num_clusters),factor=jitter_factor),
    col=col_vals,pch=pch_vals)


# ds reference line
ds_ref_line_height = s_and_d_height + reference_line_offset

lines(x=c(0,1),     y=rep(ds_ref_line_height,2),lty=3,col="gray45")
lines(x=c(0,0),     y=c(ds_ref_line_height,ds_ref_line_height+
    reference_line_tick_length),
    lty=3,col="gray45")
lines(x=c(1,1),     y=c(ds_ref_line_height,ds_ref_line_height+
    reference_line_tick_length),
    lty=3,col="gray45")
```

```
lines(x=c(0.5,0.5), y=c(ds_ref_line_height,ds_ref_line_height+
    reference_line_tick_length),
    lty=3,col="gray45")

text(x=0,  y=ds_ref_line_height+reference_line_tick_length*2,
    labels=paste(round(ds_range[1] ,2),sep=""),    cex=0.65)
text(x=0.5,y=ds_ref_line_height+reference_line_tick_length*2,
    labels=paste(round(mean(ds_range) ,2),sep=""), cex=0.65)
text(x=1,  y=ds_ref_line_height+reference_line_tick_length*2,
    labels=paste(round(ds_range[2] ,2),sep=""),    cex=0.65)



# Legend
leg_x = 0.2
leg_y = 0.07

if (num_clusters < 3) {
    leg_ncol = 2
} else {
    leg_ncol = 3
}

legend(x=leg_x,y=leg_y,legend=paste("Cluster",1:num_clusters),
    col=col_vals,pch=pch_vals,cex=0.6,ncol=leg_ncol)

par(op)
```

```
}

AssignNonConverters <- function(non_conv_data, cluster_centers, daisy_list) {
    # This function takes non_converters, cluster centers, and the
    # daisy_list for distance (maybe this should be optional). The output
    # is a vector of length dim(non_conv_data)[1] that indicates
    # which cluster a given row is assigned to, where the number is
    # based on the index of the cluster center. Ties are broken
    # randomly.

}

ShowMedoids <- function(clus_obj, the_data){
    # prints the data for each medoid in the clustering object.

    cluster_medoids <- clus_obj$medoids

    for (i in 1:length(cluster_medoids)) {
        print(paste("Printing Cluster",i,sep=" "))
        print(subset(the_data, rownames(the_data)== cluster_medoids[i]))
    }
}

SummarizeClusters <- function(the_data,the_clusters) {
    # goal is to give us an easy way to summarize
    # a set of clusters. The key statistics are Search Only,
    # Display Only, S&D percentages, average record size and
    # average number of clicks. These are returned in a
```

```
    # data frame where the first column is the number of
    # clusters.
    #
    # The results will be sorted by S Only, D Only, then S & D
    # to try to get some uniformity and get away from the arbitrary
    # cluster numbers.

    uni_clusters <- sort(unique(the_clusters))
    num_clusters <- length(uni_clusters)
    num_obs <- dim(the_data)[1]

    results <- data.frame(
        cluster = uni_clusters,
        avg_records = 0,
        search_only = 0,
        display_only = 0,
        s_and_d = 0,
        avg_clicks=0,
        pct_convs=0)

    for (i in 1:num_clusters){
        results[i,2] =
            mean(the_data$RecordCount[the_clusters==uni_clusters[i]]) #
        results[i,3] =
            mean(the_data$SearchOnly[the_clusters==uni_clusters[i]]) # S
        results[i,4] =
            mean(the_data$DisplayOnly[the_clusters==uni_clusters[i]]) #D
        results[i,5] =
```

```
            mean(the_data$SearchAndDisplay[the_clusters==uni_clusters[i]])
        results[i,6] =
            mean(the_data$Clicks[the_clusters==uni_clusters[i]]) # Clicks
        results[i,7] =
            sum(the_clusters==uni_clusters[i])/num_obs*100 # pct conv
    }

    order_idx = order(results$search_only,results$display_only,
        results$s_and_d,results$avg_records,decreasing=TRUE)

    return(results[order_idx,])
}
# SummarizeClusters(small_conv,cl10_daisy$clustering)

FindClusterSplits <- function(the_data,daisy_list,K=12, B=10,N=1000) {
    # This function finds the splits in the clustering solution.
    # The basic concept is that we split the data into a group of
    # size N. We form clusters from 2 to K clusters and keep
    # track of the number of clusters that are search only, display
    # only and S&D.
    # We do this (splitting, clustering, counting) B times and capture
    # the results. What we're looking for is 0 variation so we summarize
    # our resulting data frame appropriately. (Ie, I haven't figured
    # out the summary as I write my comments.)

    holder <- data.frame(trial = seq(1,(K-1)*B),
                         clusters = rep(seq(2,K),B),
                         num_s_only = 0,
```

```
                        num_d_only = 0,
                        num_s_and_d = 0)
    row_counter <- 1
    for (i in 1:B) {
        small_data_set <- some(the_data,n=N)

        small_diss <- daisy(small_data_set, type = daisy_list)

        for (j in 2:K) {
            the_clusters = pam(small_diss,k=j)

            cluster_results = SummarizeClusters(small_data_set,
                the_clusters$clustering)

            holder[row_counter,3] <- sum(cluster_results$search_only)
            holder[row_counter,4] <- sum(cluster_results$display_only)
            holder[row_counter,5] <- sum(cluster_results$s_and_d)

            row_counter = row_counter + 1
        }

    }

    # create our summary results. For each number of clusters
    # we'll calculate the mean and sd for the s, d and s&d
    results <- data.frame(clusters=2:K,
                          s_only_mean = 0,
                          s_only_sd = 0,
```

```
                          d_only_mean = 0,
                          d_only_sd = 0,
                          s_and_d_mean = 0,
                          s_and_d_sd = 0)
    for (i in 2:K) {
        results$s_only_mean[i-1]  <-
            mean(holder[holder$clusters==i,"num_s_only"])
        results$s_only_sd[i-1]    <-
            sd(holder[holder$clusters==i,"num_s_only"])
        results$d_only_mean[i-1]  <-
            mean(holder[holder$clusters==i,"num_d_only"])
        results$d_only_sd[i-1]    <-
            sd(holder[holder$clusters==i,"num_d_only"])
        results$s_and_d_mean[i-1] <-
            mean(holder[holder$clusters==i,"num_s_and_d"])
        results$s_and_d_sd[i-1]   <-
            sd(holder[holder$clusters==i,"num_s_and_d"])
    }
    return(results)
}

#FindClusterSplits(conv_data,daisy_type_list,K=10,B=2,N=1000)



.ls.objects = function (pos = 1, pattern, order.by,
                        decreasing=FALSE, head=FALSE, n=5) {
    napply <- function(names, fn) sapply(names, function(x)
                    fn(get(x, pos = pos)))
```

```
    names <- ls(pos = pos, pattern = pattern)
    obj.class <- napply(names, function(x) as.character(class(x))[1])
    obj.mode <- napply(names, mode)
    obj.type <- ifelse(is.na(obj.class), obj.mode, obj.class)
    obj.size <- napply(names, object.size)
    obj.dim <- t(napply(names, function(x)
        as.numeric(dim(x))[1:2]))
    vec <- is.na(obj.dim)[, 1] & (obj.type != "function")
    obj.dim[vec, 1] <- napply(names, length)[vec]
    out <- data.frame(obj.type, obj.size, obj.dim)
    names(out) <- c("Type", "Size", "Rows", "Columns")
    if (!missing(order.by)) out <- out[order(out[[order.by]],
        decreasing=decreasing), ]
    if (head) out <- head(out, n)

    out
}

////////////////////////////////////////////////////
//           VISUALIZATION FUNCTIONS                //
////////////////////////////////////////////////////

# This file holds the functions needed to do the visualzation described
# in the "Conversion Attribution Visualization" chapter of the dissertation.
#
# John Chandler-Pepelnjak, January 2010
#
# TODO: Handle dates appropriately on y-axis
```

```
#


emap_code_directory <-
    "C:\\Analytics\\EngagementMapping\\EM_as_Model\\Code\\"


the_wd <- setwd(emap_code_directory)


source('Definitions_EventBased.r') # needs to precede data_input.r
#source('CA_Model_Functions_Evt.r')
#source('Support_Functions.r')


setwd(the_wd)




refresh_code <- function() {
    # refreshes this code if I make a change
    the_wd <- setwd(code_dir)
    source("VisualizationFunctions.r")
    setwd(the_wd)
}



plot_cookie_survival_curve <- function(cookie_event_data,
    hazard_times, hazard_surv, model,
        just_lines=F,plot_survival=F,main_title,...) {


    # just makes use of the utility function.
```

```
    surv_estimates <- get_survival_estimates(cookie_event_data, hazard_times,
        hazard_surv, model,return_survival=plot_survival)

    this_times <- surv_estimates$time
    plot_y <- surv_estimates$surv

    if(plot_survival) {
        y_lab <- "Non-conversion Probability"
        if(missing(main_title)) main_title =
            paste("Non-conversion Probability\nCookie:",
                cookie_event_data$cookie[1])
    } else {
        if(missing(main_title)) main_title =
            paste("Conversion Probability\nCookie:",
                cookie_event_data$cookie[1])
        y_lab = "Conversion Probability"
    }

    if(just_lines) {
        lines(this_times,plot_y,...)
    } else {
        plot(this_times,plot_y,type="l",
            main=main_title,
            xlab = "Time",
            ylab = y_lab,...)
    }
}
```

```
get_survival_estimates <- function(cookie_event_data, hazard_times,
        hazard_surv, model,times,return_survival=F) {
    # This function produces survival estimates and returns a
    # data frame where
    # the first column is the times and the second column is the times.

    # gather model information we need
    mod_coef <- ifelse(is.na(model$coefficients), 0, model$coefficients)

    mod_asgn  <- model$assign # gives us the look-up between model and data
    mod_terms_obj <- terms(model) #gigantic terms object, used
        # for other stuff
    mod_terms <- names(mod_asgn)

    # pulls out the relevant data for cookie
    mod_frame <- model.frame(mod_terms_obj, data=cookie_event_data)
    mod_mat   <- model.matrix(delete.response(mod_terms_obj), mod_frame,
                    contr=model$contrasts)[,-1,drop=FALSE]
                        # a design matrix based on cookie.
                        # Handles coding of categorical
                        # drop the intercept in a PHM model
    nterms    <- length(mod_terms)
    pred      <- matrix(0,ncol=nterms,nrow=nrow(cookie_event_data))

    mean_pred <- mod_coef * model$means

    for (i in 1:nterms) {
```

```
        ii <- mod_asgn[[ mod_terms[i] ]]
        pred[,i] <- mod_mat[,ii,drop=FALSE] %*% (mod_coef[ii])
    }

    # pred is now an n x m matrix where n = records and m =
    # variables in model
    risks <- exp(apply(pred,1,sum) - sum(mean_pred))
    num_events <- dim(cookie_event_data)[1]

    # first just build the full survival curve,
    # don't know another way to do this accurately

    #Select down the baseline hazard data because
    # our raw data can have hundreds of thousands of
    # points that we don't need to carry through for the calculation
    this_haz_idx <- hazard_times < max(cookie_event_data$stop)

    if(sum(this_haz_idx) == 0) {
        warning("Last stop time for cookie is less than
                    minimum survival time.
                    \nReturning full survival curve.")
        this_haz_idx <- rep(TRUE,length(hazard_times))
    }

    this_times <- hazard_times[this_haz_idx]
    this_surv <- hazard_surv[this_haz_idx]

    # create a product based version of this_surv
```

```
temp <- c(1,this_surv)
this_surv_prod <- temp[2:(length(this_surv)+1)]/
    temp[1:length(this_surv)]
# to generate any entry in this_surv[i] just take
# cumprod(this_surv_prod[1:i])[i]
# to the right point.

adj_surv <- this_surv_prod
# we use idx to determine what parts of this_surv_prod we need to raise
# to which power. Then we make adj_surv the cumprod
for (i in 1:num_events) {
    idx <- (cookie_event_data$start[i] <= this_times &
        this_times <= cookie_event_data$stop[i])
    adj_surv[idx] <- adj_surv[idx]^risks[i]
}
adj_surv <- cumprod(adj_surv)
# that last bit took about a week of work. Although to be fair,
# the better part of the
# week was spent trying to figure out exactly what the
# differences were between
# S(t_i), \Lambda(t_i), and \lambda(t_i). This is what happens when
# you don't have a class in suvival analysis!
#
# Anyway, the key insight, that took me forever to reach, was that
# the critcal component was the multiplicative
# aspect of the survival curve. At
# any time, t_i, you could derive, from the fit,
# a number $a$ that obeyed
```

```
    # the relationship a \cdot t_i = t_{i+1}.
    # That's what's in this_surv_prod.
    if(!return_survival) adj_surv <- 1-adj_surv

    if(missing(times)) {
        # we want the full curve
        return(data.frame(time=this_times,surv=adj_surv))
    } else {
        surv_holder <- numeric(length(times))

        for (i in 1:length(times)) {
            if (times[i]==0) { #handling an edge case
                surv_holder[i] <- as.numeric(return_survival)
            } else { # handling the normal case, some time in the middle
                t_idx <- max(which(this_times < times[i]))
                if(t_idx < 0) {
                    surv_holder[i] <- 1-as.numeric(return_survival)
                } else {
                    surv_holder[i] <- adj_surv[t_idx]
                }
            }
        }
        return(data.frame(time=times,surv=surv_holder))
    }
}


get_emap_scores <- function(x,time,theta,convWind=convWindow,
```

```
    col_sum=T,normalize=T,cume=T) {
    # this function takes a cookie_event_data record (gulp),
    # an emap function,
    # and optionally some times. It returns a data frame where
    # the first column
    # holds the times and the second column holds the emap scores.

# This comes up with the share of credit
# scores for a set of data, assuming an EM model.
# If there is no t supplied then we use the conversion time.
# Otherwise we just come up with the score
# that would have resulted if there was a conversion
# at time t (where t could be a vector)
#
# Note: a lot of the really tricky programming (like "outer" usage)
# is required to support time vectors. Seems worthwhile, just harder.

    clicks_vector = x[,colClick]
    event_time_vector = x[,colEventTime]
    act_time_vector = rep(max(x$stop),len=length(clicks_vector))
    creative_type_vector = x[,colCreativeType,drop=TRUE]
    ad_size_vector = x[,colAdSize,drop=TRUE]

if(missing(time)){
    time <- min(act_time_vector)
}

# The conversion window needs to be defined relative to
```

```
# the time and the events in the window.
if (!any(clicks_vector==1)){
        # this is the easy case. No clicks means view window everywhere
    conv_window <- convWind$view
        in_window_idx = outer(event_time_vector,time,FUN=function(evt,t)
            t-conv_window <= evt & evt <= t)
} else {
    # this case is harder. If clicks exist we have to use the view window
    # until the click happens, then switch to the click window
    # for as long as the click is in.
    # And I'd like to avoid using a loop because I think
    # it's going to be slower. (and this function gets called
    # millions of times.)
    # But I can't figure it out right now so i'm going to loop

        # start with the windows
        view_in_window_idx =
            outer(event_time_vector,time,FUN=function(evt,t)
                t-convWind$view <= evt & evt <= t)
        click_in_window_idx =
            outer(event_time_vector,time,FUN=function(evt,t)
                t-convWind$click <= evt & evt <= t)

        in_window_idx = view_in_window_idx

        #Iterate over the times, replacing the view with click
        # columns if needed.
    for (i in 1:length(time)) {
```

```
        if(any(clicks_vector[click_in_window_idx[,i]]==1)){
            in_window_idx[,i] = click_in_window_idx[,i]
        }
    }
    }
# First get the baseweights.
bw = theta$baseweights$bw[match(creative_type_vector,
    theta$baseweights$creative_type_names)]
bw[clicks_vector==1] = theta$vars$bw[theta$vars$var_names=="Click"]

    # determine size multiplier
size = ad_size_vector
size[clicks_vector==1] = 1 # No size and recency on clicks

#Recency is more complicated since t can be a vector
rec_exp = outer(event_time_vector,time,FUN=function(x,y)
    (y-x)/convWind$view) #I'm amazed that worked on the first try
                                            # even with vectorized time
# any events that happen after the action time will get a score of 0.
# You need this when time is a vector
rec_exp[!in_window_idx] = 0

# no recency on clicks
rec_exp[clicks_vector==1,] = 0

rec = (1-theta$vars$bw[theta$vars$var_names=="Recency"])^rec_exp

#Order: Only for active events which means just clicks at this point.
```

```
# It's a vector of 1s unless you are something other than the first
# click in which case it decrements according to
# Order = (1-amount used)^(n-1) for order n in (1,2,3,4)
    ord_exp = ifelse(clicks_vector==1,1,0)
    ord_exp = cumsum(ord_exp)-1
    ord_exp = ifelse(clicks_vector==1,ord_exp,0)
    ord = (1-theta$vars$bw[theta$vars$var_names=="Order"])^ord_exp

# Multiple values to get scores
# this is component-wise multiplication, not R's %*% matrix multiplication
scores = matrix(rep(bw,length(time)),ncol=length(time)) * rec *
    matrix(rep(size,length(time)),ncol=length(time)) *
        matrix(rep(ord,length(time)),
        ncol=length(time))
scores[!in_window_idx] = 0

if(col_sum){
scores = apply(scores,2,sum)
}

if(col_sum & cume) {
    scores <- cumsum(scores)
}

if(normalize & col_sum) {
    scores = scores/max(scores)
}
```

```
return(data.frame(time=time,score=scores))



}
```

Bibliography

[1] Greg M. Allenby, Neeraj Arora, and James L. Ginter, *On the heterogeneity of demand*, Journal of Marketing Research **35** (1998), no. 3, 384–389.

[2] Greg M. Allenby, Robert P. Leone, and Lichung Jen, *A dynamic model of purchase timing with application to direct marketing*, Journal of the American Statistical Association **94** (1999), no. 446, 365–374.

[3] John Battelle, *The search: How google and its rivals rewrote the rules of business and transformed our culture*, Portfolio Hardcover, 2005.

[4] Cynthia A. Brewer, *Color brewer*, 2009.

[5] R. Briggs, *Abolish clickthrough now!*, Digitrends, Fall (1999).

[6] R. Briggs and N. Hollis, *Advertising on the web: Is there response before click-through?*, Journal of Advertising Research **37** (1997), no. 2.

[7] P. Chatterjee, D. L. Hoffman, and T. P. Novak, *Modeling the clickstream: Implications for Web-Based advertising efforts*, Marketing Science **22** (2003), no. 4, 520–541.

[8] Patrali Amal Chatterjee, *Modeling consumer network navigation in world wide web sites: Implications for advertising*, Ph.D. Thesis, 1998.

[9] D. R. Cox, *Partial likelihood*, Biometrika **62** (1975), no. 2, 269–276.

[10] D.R. Cox and David Oakes, *Analysis of survival data*, 1st ed., Chapman & Hall/CRC, 1984.

[11] John Fox, *Applied regression analysis, linear models, and related methods*, Sage Publications, 1997.

[12] A. E. Gelfand and A. F. M. Smith, *Sampling-based approaches to calculating marginal densities*, Journal of the American Statistical Association **85** (1990), no. 410, 398–409.

[13] J. C. Gower, *A general coefficient of similarity and some of its properties*, Biometrics **27** (1971), no. 4, 857–871.

[14] David Hallerman, *US online advertising: Resilient in a rough economy*, eMarketer, 2008.

[15] Frank E. Jr. Harrell, *Regression modeling strategies*, Corrected, Springer, 2001.

[16] Trevor Hastie, Robert Tibshirani, and Jerome Friedman, *The elements of statistical learning: Data mining, inference, and prediction*, 2nd ed., Springer, 2009.

[17] David W. Hosmer and Stanley Lemeshow, *Applied logistic regression*, John Wiley & Sons, Inc., 2000.

[18] David W. Hosmer, Stanley Lemeshow, and Susanne May, *Applied survival analysis: Regression modeling of time to event data*, 2nd ed., Wiley-Interscience, 2008.

[19] Y. Hu, L. M. Lodish, and A. M. Krieger, *An analysis of real world TV advertising tests: A 15-Year update*, Journal of Advertising Research **47** (2007), no. 3, 341.

[20] Erin Hunter, *New study shows that heavy clickers distort reality of display advertising Click-Through metrics.*

[21] J. D. Kablfleisch and R. L. Prentice, *The statistical analysis of failure time data*, New York:Wiley, 1980.

[22] Leonard Kaufman and Peter J. Rousseeuw, *Finding groups in data: An introduction to cluster analysis*, 1st ed., Wiley-Interscience, 2005.

[23] Donald E. Knuth, *Literate programming*, The Computer Journal **27** (1984), no. 2, 97–111.

[24] L. M. Lodish, M. Abraham, S. Kalmenson, J. Livelsberger, B. Lubetkin, B. Richardson, and M. E. Stevens, *How TV advertising works: A Meta-Analysis of 389 real world split cable TV advertising experiments*, Journal of Marketing Research **32** (1995), 125–139.

[25] P. Manchanda, J. P. Dubé, K. Y. Goh, and P. K. Chintagunta, *The effect of banner advertising on internet purchasing*, Journal of Marketing Research **43** (2006), no. 1, 98–108.

[26] R.D. Gill P.K. Andersen O. Borgan and N. Keiding, *Statistical models based on counting processes*, Springer-Verlag, 1993.

[27] R Development Core Team, *R: A language and environment for statistical computing*, R Foundation for Statistical Computing, Vienna, Austria, 2009. ISBN 3-900051-07-0.

[28] P. E. Rossi and G. M. Allenby, *Bayesian statistics and marketing*, Marketing Science (2003), 304–328.

[29] Peter E. Rossi, Greg M. Allenby, and Rob McCulloch, *Bayesian statistics and marketing*, 1st ed., Wiley, 2006.

[30] S. N. Singh and C. A. Cole, *The effects of length, content, and repetition on television commercial effectiveness*, Journal of Marketing Research (1993), 91–104.

[31] Peter J. Smith, *Analysis of failure and survival data*, Chapman and Hall, 2002.

[32] Terry M. Therneau and Patricia M. Grambsch, *Modeling survival data: Extending the cox model*, Springer, 2001.

[33] Robert Tibshirani, Guenther Walther, and Trevor Hastie, *Estimating the number of clusters in a data set via the gap statistic*, Journal of the Royal Statistical Society. Series B (Statistical Methodology) **63** (2001), no. 2, 411–423.

[34] D. Vakratsas and T. Ambler, *How advertising works: What do we really know*, Journal of Marketing **63** (1999), 26–43.