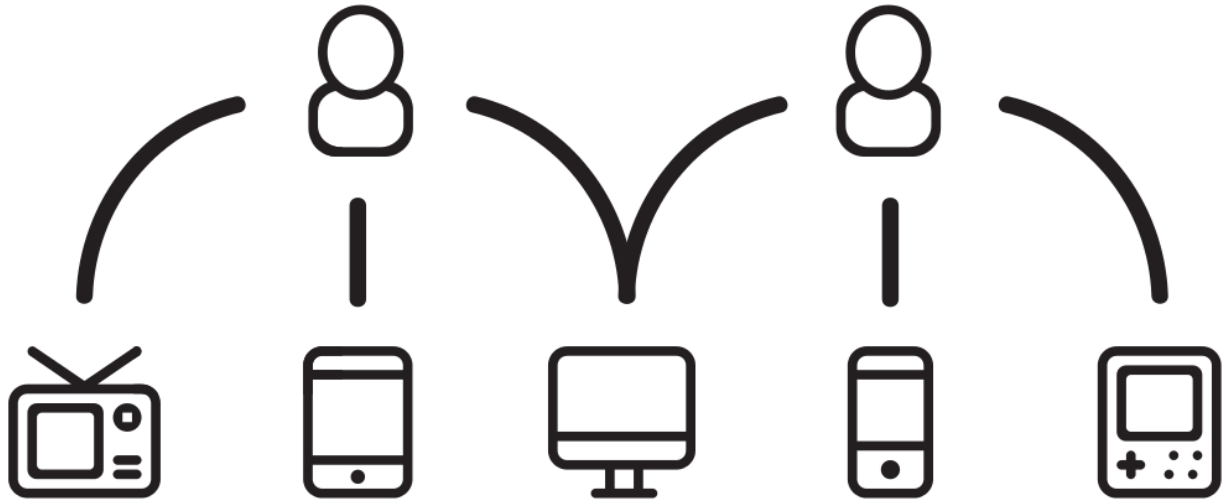
CHALMERS
UNIVERSITY OF TECHNOLOGY



Identity Bridging

Cluster Website Visits using Model-based Clustering

Master's thesis in Computer Science: Algorithm, Languages and Logic

Ásbjörn Hagalín Pétursson
Rúnar Kristinsson

Department of Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2015

Master's thesis 2015:NN

Identity Bridging

Cluster Website Visits using Model-based Clustering

ÁSBJÖRN HAGALÍN PÉTURSSON
RÚNAR KRISTINSSON

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Computer Science and Engineering
Chalmers University of Technology
Gothenburg, Sweden 2015

Identity Bridging
Cluster Website Visits using Model-based Clustering
ÁSBJÖRN HAGALÍN PÉTURSSON
RÚNAR KRISTINSSON

Examiner: Graham Kemp, Department of Computer Science

Supervisor: Christos Dimitrakakis, O. Docent, Department of Computer Science

Supervisor: Joel Segerlind

Master's Thesis 2015:NN
Department of Computer Science and Engineering
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Cross-device recognition.

Typeset in LaTeX
Gothenburg, Sweden 2015

Identity Bridging
Cluster Website Visits using Model-based Clustering
ÁSBJÖRN HAGALIN PÉTURSSON
RÚNAR KRISTINSSON
Department of Computer Science and Engineering
Chalmers University of Technology

Abstract

Model-based clustering is becoming increasingly popular with the rise in computational power. Cluster analysis is used in many disciplines, for example biology, geography, image analysis and marketing. In this thesis we developed a model-based unsupervised clustering method to cluster website visits into clusters that represent a unique Internet user.

As no ground truth exists we developed two evaluation methods to measure the quality of the clusters, one based on cluster content and size, and the other based on user behavior.

The model-based clustering method was compared with a simple deterministic clustering model, the results were very similar. With further development of the model-based clustering we believe that it can generate better clusters of website visits that likely represent a single user.

Keywords: Machine learning, Clustering, Model-based clustering, Website visits, Cross-device tracking, Probabilistic model.

Acknowledgements

We would like to express our deep gratitude to Christos Dimitrakakis, our academic supervisor, for guidance with the statistics and the probabilistic model. We would also like to thank our examiner Graham Kemp for advice on improving the language in the report. Our grateful thanks are also extended to our supervisor at the Company, Joel Segerlind for help with understanding the problem and guidance with implementation.

We would also like to extend our thanks to the employees at the Company, for their help and offering us office space.

Ásbjörn Hagalín Pétursson, Gothenburg, June 2015
Rúnar Kristinsson, Gothenburg, June 2015

Contents

List of Figures

List of Tables

1
Introduction

The company this thesis is done in collaboration with is a big data company that helps media publishing groups create new revenue streams through their audience, advertising and content.

One of their products involves building historical profiles of visitors across websites, devices and browsers. A historical profile consists of data explained in section 1.4. Being able to cluster website visits correctly to unique Internet users is crucial for this product. Classifying website visits incorrectly can lead to loss of revenue for the media publisher. A research from 2009 shows that the average price of behavioral targeting advertising is 2.68 times higher than price of untargeted ads. The research also showed that behavioral targeting is an important source of revenue (Beales, 2010).

The problem of clustering website visits to unique Internet users across multiple devices is not new. It is also known as *cross-device tracking*, *cross-device targeting* or *cross-device reporting* depending on the solution. The goal of cross-device tracking is to be able to tell if the person using mobile phone A is the same person that uses tablet B and desktop C, this then allows companies to re-target the person on all the devices. Two big tech companies; Google and Facebook offer cross-device tracking. Both these solution require the user to be signed in to their websites or apps on every device they use (Facebook, 2015; Google, 2015). This requirement could be a deal breaker for some companies that cannot rely on their visitors to be signed in on either of these services.

While working on this thesis there was no research found that directly focuses on the problem of cross-device tracking. One explanation could be that the market is highly competitive and research performed by companies remains in house.

1.1 Ethics Concerning Internet User Tracking

Charters (2002) and Rapp et al. (2009) describe the ethics surrounding the collection of information for advertising purposes and the implication for the Internet users privacy. They discuss the controversy over matching anonymous information with personal information. The company this thesis is done in collaboration with does not collect personal information in the sense a person can not be identified with the data collected. Charters (2002) discuss the importance of an *opt-out* feature. The company offers users the option to opt-out by placing an "opt-out cookie" on the users browser. This thesis does not discuss the ethics surrounding the collection of information but we acknowledges that data collection may never been done without total transparency. The users must be aware of the tracking and be able to have the option to deny information collection.

1.2 Problem Formulation

The method that is currently in use is deterministic clustering that clusters website visits into clusters by using device properties and user subscriptions. A cluster is supposed to represent a single user. These clusters are also referred as historical profiles. There are two main problems with this model; the first problem occurs when website visits from multiple users are clustered into one cluster, this occurs for example if users share devices or borrow subscription to a publisher's website from another user. The second problem is when website visits from one user is clustered into multiple clusters. The first problem is more undesirable behavior as it can lead to incorrect identification of a user, which in turn can lead to loss in revenue for the advertiser (Beales, 2010).

This thesis aims to design evaluation methods to evaluate the quality of the historical profiles and investigate alternative methods to cluster website visits to unique Internet users across multiple platforms and websites. We propose a probabilistic method that takes into account that users can share devices and borrow subscriptions. The method we propose is described in section 3.2.

A difficult part of the problem is being able to work with the data. The data set is large so any method has to be efficient, therefore excluding any methods that are not efficient. This thesis will use data from one media publisher that the Company collects data from approximately 150 websites. The users of these websites are a large part of the Scandinavian population resulting in large amount of data collected. The next challenge is to create a evaluation strategy, as no ground truth exist. The evaluation strategy will have to make use of known properties of the data collected. Another challenging aspect is to decide what part of the information available is possible to use due to the amount of data.

1.3 Definitions

This section describes specific terms which are used throughout the thesis.

1.3.1 The Company

The company at which the thesis work was conducted will throughout the thesis be referred to as *the Company*.

1.3.2 Company X

Company X is a media publisher that operates around 150 websites. X subscribes to analytic service provided by the Company.

1.3.3 The Websites

The websites that X operates will be referred to as *the Websites*.

1.4 The Data

The data set used in this thesis is supplied by the Company; it originates from X that operates around 150 websites. This media publisher subscribes to analytics services from the Company. The data set was collected over a seven day period.

When a user accesses one of the Websites his activity is sent to the Company. This activity is sent in the form of *device signals* which are sets of beacons. A beacon can have one of four flags; customer ID, parent ID, first party cookie and third party cookie, each of which will be described below. One day of data contains over 5 million device signals, resulting in the data set to contains at least 35 million device signals. One device signal contain two beacons on average.

The historical profiles mentioned above contain all the beacons that are believed to represent website visits from a single user.

1.4.1 Cookies

A cookie or a browser cookie is a small data file sent by a website that user visits and is stored on the user's browser. Every time the user visits the website again the browser sends the cookie to the website enabling the website to keep track of previous actions performed by the user.

The Company uses two kinds of cookies for providing analytics, first party cookie (FPC) and third party cookie (TPC). The word "party" refers to the domain as specified in the cookie (Barth, 2011).

If a user visits example.com and the domain of the cookie stored on the browser is example.com the cookie is referred to as first party cookie. However if the domain of the cookie set by example.com is for example not-example.com the cookie is referred to as third party cookie.

Third party cookies are used to identify the same user across multiple domains. However they are not perfectly reliable. Users tend to remove cookies and some browsers reject them all together (Lavin, 2006)(Apple, 2015).

1.4.2 Customer ID and Parent ID

A customer is a user that has registered on one of the Websites. By registering the user enters some personal information. When the user registers on one of the Websites, a customer ID is assigned to the customer.

Each customer can have one customer ID from a single website. If the customer is registered on multiple websites he will have multiple customer IDs, one from each of the websites he is registered on. Company X gathers all customer IDs belonging to single customer and links them to a parent ID. This is done by matching the account information across the Websites. Each customer can then have multiple customer IDs but should only have a single parent ID. However by experience there exist customers that have multiple parent IDs. These customers are an exception and can stem from error in Xs' process of linking multiple customer IDs to one parent ID.

An example of assigning customer IDs to users and link between parent IDs and customer IDs is shown in figure 1.1. In the figure a circle represents a customer and the number within the circle represents a unique customer. The same number under different websites represents the same customer. A link between a circle (customer) and a website represents use. A square represents a customer ID and a link to a customer represents that the customer has that customer ID. A diamond represents a parent ID and a link to a customer ID represents that the customer ID has this parent ID. The example shows four different customers across three websites. Each customer has one parent ID and multiple customer IDs.

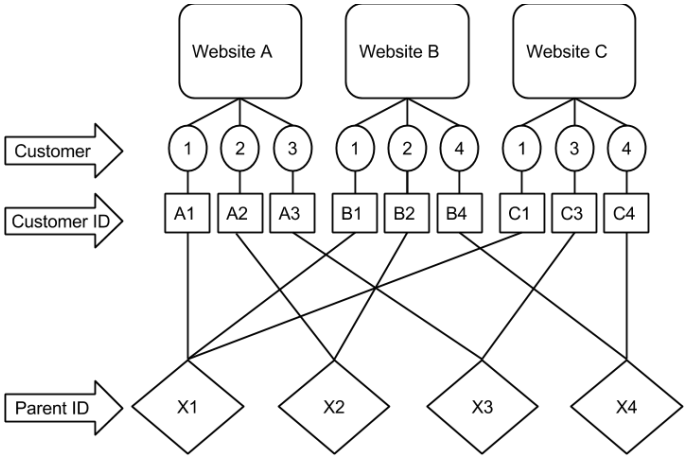


Figure 1.1: Example showing relation between customer IDs and parent IDs.

1.4.3 Input of device signals

To be able to perform evaluations of different methods for creating historical profiles with exactly the same order of observed device signals we programmed an input

method into the Company's system, giving the Company the opportunity to easily try out new methods. The input method simulates observation of device signals in the order they were observed previously in real time.

1.4.4 Preprocessing

The data set contains duplicates entries of device signals. Due to the amount of data it becomes necessary to remove the identical device signals, to speed up the simulation. One reason for duplicate entries of device signals could be that one of the Websites is reporting to the Company when the user is inactive, to remove these duplicates we remove duplicates in one hour time window at a time. An experiment was conducted using the current method in use by the Company to assess if the preprocessing affects the historical profiles created, the current method is explained in section 3.1.

First a part of the data set was used without the preprocessing, creating set of profiles A. Next, the same part of the data set was preprocessed removing duplicate device signals. The preprocessed data was then used to create set of profiles B.

The two resulting set of profiles were compared, for the preprocessing not to affect the data, A and B need to be identical. For every profile in A it was checked if identical profile in B existed, identical meaning that profiles contain the same beacons. The inverse of this check was also performed. After the comparisons of the two sets it was clear that sets A and B were identical.

1.5 Scope and limitation

The amount of data collected by the Company will require any method that clusters website visits to execute on a distributed system. The time frame of this thesis does not allow for distributed versions of the methods to be explored.

Another limitation to our approach is also related to the amount of data. The evaluation of the methods can only be executed on a limited amount of data. The initial goal is to be able to evaluate the methods on one week of data. Given that one day of data contains over 5 million device signals evaluating a method with a year of data or even few months becomes infeasible given the resources available for this thesis.

The probabilistic method we propose is computationally expensive. Therefore it's only feasible to evaluate the method using fraction of the data set. No ground truth exists for the historical profiles and to be able to evaluate the quality of methods that create historical profiles we have to design evaluation methods. Motivation for the evaluation methods is given in section 4.

The evaluation methods developed during this thesis will not address the number of profiles in the sense there is no higher bound on the number of profiles. This limitation can cause a method that creates historical profiles of minimum size to score high in the evaluation.

1.6 Thesis Outline

The thesis is divided into three main parts; designing an alternative method to build up the historical profiles, designing evaluation methods and evaluating the methods that build up the profiles. This is then followed by discussion section where we go over the most important results and future work. As well the thesis includes a theoretical chapter to assist the reader understand the methods used in the thesis, in that chapter we also go over related work.

Theory and Related Work

The foundation of the theory used in the thesis explained, we briefly mention probabilistic graphical models, Bayesian network and some probability theory. Likewise we briefly mention the sampling algorithms used and the statistical test used for comparing the evaluation methods.

Models

The method currently in use by the Company is explained, and with examples we show the pitfalls of this method. Furthermore we explain the method we designed, describe the probabilistic model and show results from the sampling algorithms on the model.

Designing the Evaluation Method

Two evaluation methods for evaluation of the quality of historical profiles are explained, Agile method and Additional data method.

Evaluation of the Models

The two evaluation methods; agile and additional data method are used to evaluate the quality of the historical profiles generated by the two models. To determine if the results from the evaluation methods give similar results a statistical test is used.

Conclusions

Results from the evaluations of the two models are analyzed and discussed. As well the comparison of the results from the evaluation methods and the limitations and flaws of the evaluation methods.

Future work

Discussions about what more can be done with the Probabilistic model, how one could distribute the model and how the evaluation methods could be improved. We mention another way of clustering, using Markov Cluster Algorithm (MCL).

2

Theory and related work

Using a probabilistic approach makes it possible to reason under uncertainty using the observed information (Poole and Mackworth, 2010). This chapter contains the main theory used in the thesis. First we will briefly go over *probabilistic graphical models* and *clustering*, then we will introduce the sampling algorithms used to sample from the probability distributions. Finally we introduce a statistical test we used to compare the results from the evaluation methods introduced in chapter 4.

2.1 Probabilistic graphical models

Probabilistic graphical models provides a mechanism to describe the structure of complex distributions in a compact way, in a graph the nodes represent random variables in the domain and edges represent the probabilistic dependencies between them. An example of the graphical representation is shown in figure 2.1

2.1.1 Bayesian network

Bayesian networks are a sub group of graphical models that represent the knowledge about an uncertain domain. For one to understand the Bayesian network a understanding of the following definitions and theorem is needed.

Definition 2.1.1 $\Pr(X|Y) = \Pr(X,Y)/\Pr(Y)$ *is the probability that* X *is true given evidence of other event* Y*, this is called* ***Conditional probability.***

Theorem 2.1.1 *The relation between* $\Pr(X|Y)$ *and* $\Pr(Y|X)$ *can be expressed with* ***Bayes' theorem***

$$\Pr(X|Y) = \frac{\Pr(Y|X) \cdot \Pr(X)}{\Pr(Y)} \tag{2.1}$$

where $\Pr(X|Y)$ *is the posterior probability of* X *being true and* $\Pr(Y|X)$ *is the prior knowledge.*

Definition 2.1.2 *When interested in a probability that involves several random variables we need to look at the* ***joint probability distribution*** *over these random variables. Let* X *and* Y *be discrete random variables, where*

$$f(x,y) = \Pr(X = x, Y = y) \tag{2.2}$$

For each (x,y) *within the probability space is called* ***joint distribution*** *of* X *and* Y*. When using more than two variables we talk about* ***multivariate distribution****.*

Definition 2.1.3 *In probability theory the* ***Chain rule*** *gives that any joint probability distribution of the random variables* $Z_1, Z_2, ..., Z_n$ *can be calculated with following equation*

$$\Pr(Z_n, ..., Z_1) = \Pr(Z_n | Z_{n-1}, ..., Z_1) \cdot \Pr(Z_{n-1}, ..., Z_1) \tag{2.3}$$

$$= \prod_{k=1}^{n} \Pr(Z_k | Z_1, \ldots, Z_{k-1}) \tag{2.4}$$

In figure 2.1 we have three random variables, A, B and C. A and B, A and C, and B and C are connected by an edge, this mean that they are not independent of each other. The joint distribution for this model is:

$$\Pr(C, B, A) = \Pr(C|B, A) \cdot \Pr(B|A) \cdot \Pr(A) \tag{2.5}$$

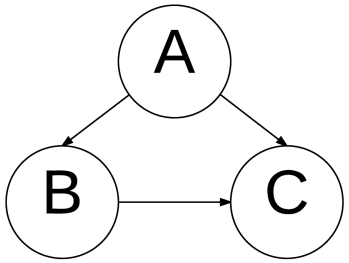


Figure 2.1: Directed graphical model, representing a Bayesian network, also known as belief network.

For more information on Probabilistic graphical models and Bayesian network, see (Koller and Friedman, 2009).

2.2 Clustering

Clustering is used to group sets with similarities into sub-sets, detecting underlying structure within a data-set (Stahl and Sallis, 2012). The clustering method used in this thesis is *model-based clustering*, "an increasingly popular area of cluster analysis that relies on probabilistic description of data by means of finite mixture models" (Melnykov, 2013). Since around the 1950s the mixture models for multivariate data, also called latent class models has been in development (Fraley and Raftery, 2002). The possibility of using Bayes nets to clustering problems was showed by (Chickering and Heckerman, 1997). Fraley and Raftery (2002) points out that using Bayes nets to clustering problems could be used for high-dimensional discrete data, for example, tracking visits to websites.

2.2.1 Model-based clustering

In model-based clustering it is assumed that the population is made up of multiple clusters (or subsets) where each cluster has its multivariate probability distribution. The model-based clustering uses probability distributions to calculate the posterior probability of observation being a part of a cluster. For more detailed information on model-based clustering, see (Fraley and Raftery, 2002; Melnykov, 2013; Stahl and Sallis, 2012).

2.3 Gibbs sampler

Gibbs sampler also called alternating conditional sampling is a Markov chain Monte Carlo (MCMC) algorithm which can be used to obtain a sequence of observations which approximate a multivariate probability distribution. The Gibbs sampler was introduced by the Geman brothers in 1984 (Geman and Geman, 1984) and has been found useful in multidimensional problems (Gelman et al., 2014). Gibbs sampler in it is basic implementation is a special case of the *Metropolis - Hastings* algorithm. The Gibbs sampler is useful because given a multivariate distribution it is easier to sample from conditional distribution than direct sampling. The Gibbs sampler is a randomized algorithm; it makes use of random numbers and hence can produce different results each time it is run. One iteration of the Gibbs algorithm for the multivariate distribution

$$\Pr(X_1, X_2, ..., X_{k-1}, X_k) \tag{2.6}$$

is shown below. The i-th sample is denoted by $X^{(i)} = (x_1^i, ..., x_k^i)$.

1. Begin with initial sample $X^{(0)} = (x_1^0, ..., x_k^0)$
2. Next, for all j the variable x_j^1 is sampled from the conditional distribution

   $$\Pr(X_j | x_{-j}^0)$$

   where x_{-j}^0 represents all the components of X, except for X_j, at their current values.

   $$x_{-j}^0 = (x_1^0, ..., x_{j-1}^0, x_{j+1}^1, ..., x_k^1)$$

   Thus each variable in X is updated conditional on the latest values of the other variables in X. The variables with 1 in the superscript are the ones that have been updated this iteration the others have not been updated.
3. When all the variable have been sampled a Gibbs sample has been constructed

   $$X^{(1)} = (x_1^1, ..., x_k^1)$$

4. The steps above are then repeated until convergence with the Gibbs sample from previous iteration as an initial sample.

For large enough iterations of the algorithm the simulated distribution converges to the multivariate distribution. It is common to ignore the first 1000 or so Gibbs samples, so called *burn-in period* and then only look at every n-th Gibbs sample when computing the expectation. For more detailed description on the Gibbs sampler we point to (Gelman et al., 2014).

When using the Gibbs sampler with a Bayesian network and model-based clustering we calculate the probabilities of observation belonging to a cluster from previous assignments that we have assigned and do not know with certainty if the assignments are correct. This can be modeled with a multinomial distribution, but in the problem of this thesis we do not know the multinomial and use a classical conjugate distribution to model our uncertainty about the multinomial, the Dirichlet distribution:

$$\Pr(X = x_i|\alpha_i) = \frac{\prod_i \Gamma(\alpha_i)}{\Gamma(\sum_i \alpha_i)} \prod_i x_i^{\alpha_i - 1} \tag{2.7}$$

2.4 Expectation-maximization

Expectation - maximization is an efficient iterative method for computing the maximum likelihood (ML) estimate when the model depends on latent variables or missing data. In ML estimation the goal is to estimate the parameters that give the highest expectation of the observed variables. The expectation-maximization (EM) algorithm is a deterministic algorithm hence it will produce the same result every time it is run.

The EM alternates between two steps: in the *expectation* (E step) the missing variables are estimated from the observed variables and the current estimate of the model parameters. This is done by calculating the expectation of the likelihood unconditioned of the missing variables. In the *maximization* (M step) the missing variables are assumed to be observed and the likelihood function is maximized given that assumption. The estimate from the E step is used instead of the missing variables. Both steps, E and M are straightforward for many standard models which makes the EM algorithm widely applicable (Gelman et al., 2014).

The EM algorithm guarantees convergence by increasing the likelihood every iteration until maximized. For more information on expectation-maximization, see (Dempster et al., 1977). For the more enthusiastic reader we point to a simple example of the EM algorithm (Do and Batzoglou, 2008).

2.5 McNemar's test

McNemar's test is a statistics test applied to paired nominal data. The test is applied to 2×2 contingency table which displays the results of two tests on the same n objects:

	Test 2 Positive	Test 2 Negative	Row total
Test 1 Positive	k	r	k + r
Test 1 Negative	s	m	s + m
Column total	k+s	r+m	n

Table 2.1: Contingency table.

Where:

- The number of times that both tests are positive, k
- The number of times that both tests are negative, m
- The number of time that test 1 is positive and test 2 is negative, r
- The number of time that test 1 is negative and test 2 is positive, s

The pairs with the same results are in agreement and they do not give any information on the tests. The pairs with different results are called discordant pairs. If there are no differences between the tests we expect $r \approx s$. One can use the McNemar's test to determine if the difference between the numbers of the discordant pairs is larger than expected by chance. To test the null hypothesis that there is no difference between numbers of discordant pairs the McNemar's test is the following:

$$\chi^2 = \frac{(r-s)^2}{r+s} \tag{2.8}$$

With sufficient number of discordant pairs the McNemar's test has a chi-squared distribution with one degree of freedom. If the result from the test is significant one can reject the hypothesis and conclude that there is difference between the tests.

2.6 Related Work

As mentioned in the introduction no research on the actual problem of classifying website visits to unique Internet users across multiple devices was found. Instead we will list related work to the methods used in this thesis.

(Joshi et al., 2008) use model-based clustering to cluster genes and conditions. The algorithm they introduce uses a Bayesian approach and a Gibbs sampler method to update the assignments of genes and conditions to clusters. They claim to address

the question about the convergence of the Gibbs sampler for large data sets. They do so by showing that for large data sets that after the burn in period the difference in likelihood between Gibbs samples is statistically insignificant.

(Cadez et al., 2000) use model-based clustering to cluster navigation patterns on a web site, where they cluster users with similar navigation paths through the site. Where a navigation path is the order in which users request web pages (Cadez et al., 2000) then use the clusters to visualize the navigation paths. For the model-based clustering they use mixture models using the EM algorithm.

3
Models

The two methods for creating profiles in this thesis are best explained as models. The method currently in use by the Company is explained as a Decision model and is described in section 3.1. The method we propose to improve the creation of profiles relies on a probabilistic model; it clusters device signals into groups, where a group is assumed to represent a single device. Then the model clusters the beacons of the device signal into groups with information from the device clusters, where a group of beacons is assumed to represent the websites visits from a unique Internet user. The model is described in section 3.2.

3.1 Decision model

For every observed device signal the method finds all candidate profiles for the device signal. The candidates are all the profiles that contain at least one of the beacons in the device signal. The oldest profile is selected from the candidate profiles and all the beacons in the device signal are added to the selected profile. For every beacon in the device signal that the selected profile did not contain before their oldest profile is found. All the beacons in those profiles are then added to the selected profile.

The steps in the decision model for an observed device signal are enumerated below:

1. All candidate profiles found
2. Oldest profile from the candidate profiles is selected
3. All beacons in the device signal that are not in the profile are selected
4. Their oldest profile is found
5. All beacons in those profiles are added to the selected profile
6. Every beacon in the device signal is added to the profile

3.1.1 The decision model described as a flowchart

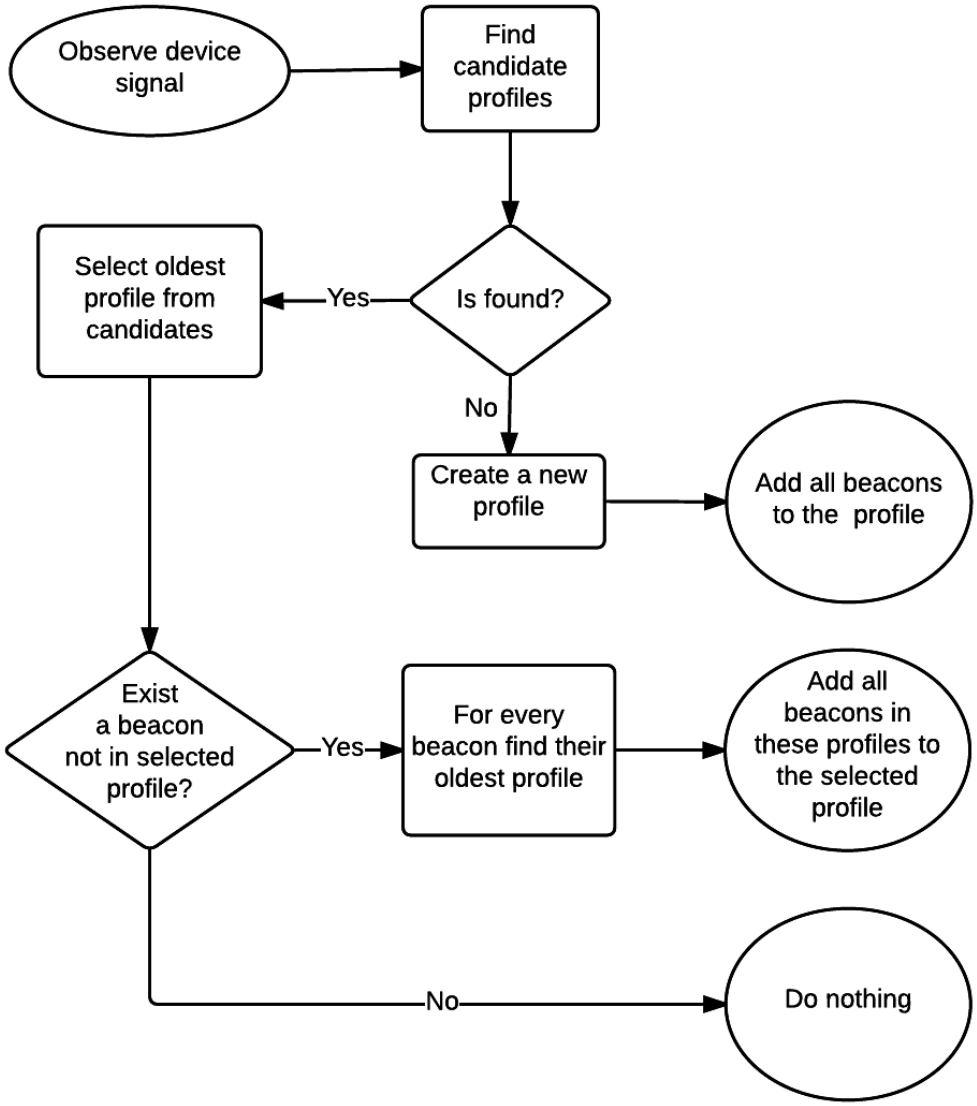


Figure 3.1: A flowchart representing the decision process for the Decision model.

The flowchart for the decision process of the Decision model is shown in figure 3.1.

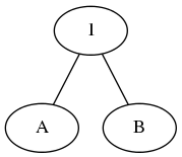
3.1.2 Example

Here is a simple example of how the Decision model works. We will describe beacons as letters and profiles as numbers.

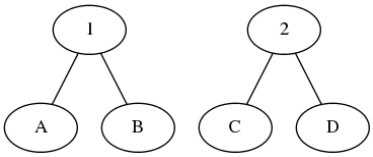
No.	Device signal	Resulting profile label
1	{A, B}	1
2	{A}	1
3	{C, D}	2
4	{A, D}	1
5	{D}	1

Table 3.1: Example on how beacons are added to profiles.

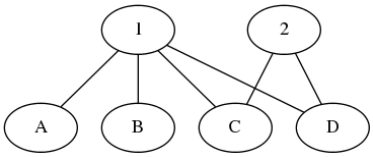
Observed device signals and resulting profiles are shown in table 3.1. First device signal contains A and B. No candidate profiles are found, a new profile is created and beacons A and B are added to the profile, see figure 3.2a. Second device signal only contains A. A has one candidate profile which is the oldest, profile 1. A is in profile 1 and nothing is done. Third device signal contains C and D. No candidate profiles are found, a new profile is created and beacons C and B are added to the profile, see figure 3.2b. Fourth device signal contains A and D. Two candidates profiles are found, 1 and 2. The oldest profile is profile 1. There exists a beacon that is not in profile 1, beacon D. The oldest profile that contains D is profile 2, all beacons in profile 2 are added to profile 1, see figure 3.2c. The last device signal contains beacon D. Two candidates profiles are found, 1 and 2. The oldest profile is profile 1. There exists no beacons in the device signal that are not in the oldest profile, nothing is done.



(a) Beacons A and B added to profile 1



(b) Beacons C and D added to profile 2



(c) Beacons C and D added to profile 1

Figure 3.2: Example showing how beacons are added to profiles

3.1.3 Model reaction to different user behavior

Below are few examples that show how the model reacts to different user behavior. In those examples the term session is used. Session refers to the period of time a user interacts with the website. The user session begins when the user accesses the website and ends when the user leaves the website.

What happens to:

1. Device signals prior to registration or logging in on a device that has never been used.

   When John visits one of the Websites without registering or logging in on a device that has never been used to access one of the Websites a new profile P_1 is created. The device signal $B = \{b_1, b_2, b_3\}$ from this session will be added to profile P_1. The resulting profile P_1 is shown in figure 3.3

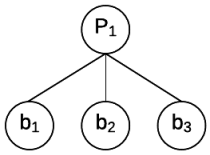


Figure 3.3: Resulting profile from device signal $B = \{b_1, b_2, b_3\}$

2. Device signals from previous sessions when a user uses the same device.

   John visits one of the Websites, performs some actions resulting in device signal $B_1 = \{b_1, b_2, b_3\}$ being observed, registers or logs in, logs out and then terminates the session. Because John logged in or registered, the device signal contains his customer ID beacon for that website, b_3. No profile contains a beacon from device signal B_1 and a new profile P_1 is created and the beacons in device signal B_1 are added to profile P_1, figure 3.4a. John then visits one of the Website, does not log back in or registers and the observed device signal is $B_2 = \{b_1, b_2\}$. Beacon b_1 and b_2 are only in profile P_1 and nothing is done, figure 3.4b. In this case b_1 and b_2 are cookies, they are the same as in his previous visit because John is still using the same device, cookie is stored on his device.

   An exception from this behavior is when John deletes his cookies before the second visit or the cookies expire. When that happens a new profile will be created for the second visit. As soon as John logs back in all the beacons from the second visit will be added to the same profile as the previous visit.

(a) Resulting profile from device signal $B_1 = \{b_1, b_2, b_3\}$

(b) No change in P_1 after observing device signal $B_2 = \{b_1, b_2\}$

Figure 3.4: Resulting profiles after device signal B_1 and B_2 are observed

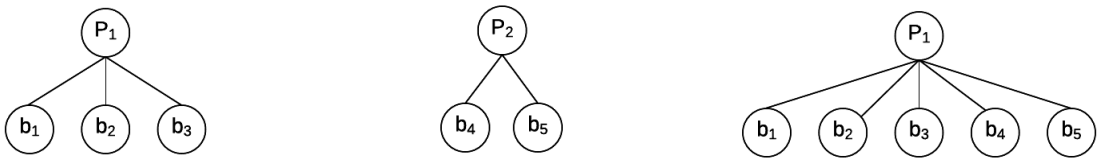
3. Device signals from several devices.

   (a) Vilma registers on one of the Websites on her computer.

   (b) Few days later she visits the website on her phone, does not log in.

   (c) Few days pass and she visits the website again on her phone, this time around she logs in on the website.

   The first time Vilma visits the website and registers, the device signal $B_1 = \{b_1, b_2, b_3\}$ is observed and contains her customer ID beacon for that website, b_3. None of the beacons in the device signal are in a profile and a new profile P_1 is created and the device signal is added to the profile P_1, figure 3.5a. When she visits the website on her phone for the first time, the device signal $B_2 = \{b_4, b_5\}$ is observed. None of the beacons in the device signal are in a profile and a new profile P_2 is created and the device signal is added to profile P_2, figure 3.5b. When she visits the website again on her phone and logs in, the device signal $B_3 = \{b_4, b_5, b_3\}$ is observed and contains her customer ID beacon for that website, b_3. Beacon b_3 is in a profile already, P_1 and beacons b_4 and b_5 are in the profile P_2 when Vilma visited the website on her phone without logging in. In this case the beacons in the newer profile, P_2 are added to the profile P_1 that contains beacon b_3,figure 3.5c.



(a) $B_1 = \{b_1, b_2, b_3\}$ **(b)** $B_2 = \{b_4, b_5\}$ **(c)** $B_3 = \{b_4, b_5, b_3\}$

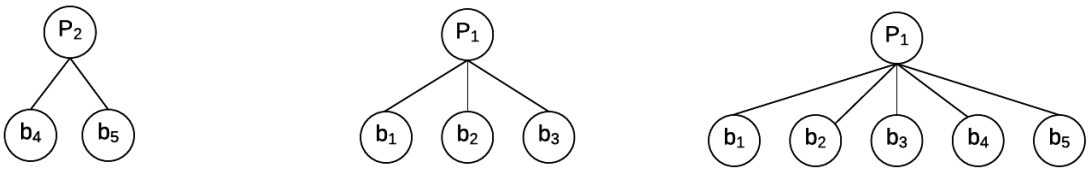
Figure 3.5: Resulting profiles after device signal B_1, B_2 and B_3 are observed

4. Device signals from multiple users using the same device.

   Vilma is browsing a website looking for knitting recipe to buy on a library computer; this computer has never been used before by anyone. She does not find a recipe to buy and never logs in, the device signal is $B_1 = \{b_4, b_5\}$ is observed. None of the beacons in the device signal B_1 are in a profile and a new profile P_2 is created and the device signal B_1 is added to profile P_2, figure 3.6a.

Now John goes on the library computer looking for airplane models to buy, he finds a model, logs in and purchases the model, the device signal $B_2 = \{b_4, b_5, b_3\}$ is observed and it contains his customer ID beacon for that website, b_3. When John logs in the device signal from his session is added to P_1 the oldest profile that contains b_3, figure 3.6b. The resulting profile is shown in figure 3.6c.

A side effect from John logging in is that all beacons from Vilma's session are added to the same profile.



(a) Resulting profile from device signal $B_1 = \{b_4, b_5\}$

(b) Previously created profile, with customer ID beacon b_3

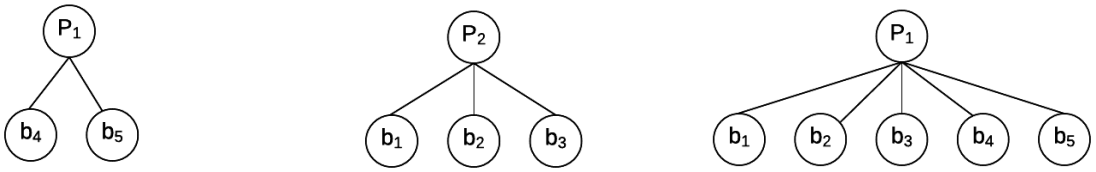
(c) Resulting profile from device signal $B_2 = \{b_4, b_5, b_3\}$

Figure 3.6: Resulting profiles after device signal B_1 and B_2 are observed

5. Device signals from multiple users logging in on the same device.

   Vilma lends John her phone; John logs Vilma out of the website and logs himself on the website.

   When John logs in on the website on Vilma's phone, the device signal $B_1 = \{b_4, b_3\}$ is observed and it contains his customer ID beacon for that website, b_3. Beacon b_3 is in profile P_2 that contains beacons from John's previous sessions, figure 3.7b. Beacon b_4 in profile P_1 that contains beacons from Vilma's sessions, figure 3.7a. Because beacon b_4 is in a profile that is older than profile P_2 the device signal B_1 is added to profile P_1, as well the beacons in profile P_2 are added to profile P_1, figure 3.7c. As a result the profile that contained beacons from John's session are now in the profile that contains beacons from Vilma's sessions.



(a) Previously created profile, with customer ID beacon b_5

(b) Previously created profile, with customer ID beacon b_3

(c) Resulting profile from device signal $B_1 = \{b_4, b_3\}$

Figure 3.7: Resulting profiles after device signal B_1 is observed

These examples show how the current model reacts to different user behavior. In examples 1, 2 and 3 the model reacts desirably. However in example 4 and 5 the method reacts undesirably.

In example 4 Vilma's session is added to the profile that represents John, this could for example indicate that John is interested in a topic that he has no interest in. This behavior is undesirable in the perspective that the accuracy of selling advertisements based on users previous actions can be affected.

In example 5 the beacons in profile P_2 are moved to profile P_1. This results in future sessions of Vilma and John to be added to the same profile. As a result profile P_1 is now representing two persons and not one person as is desired, making advertising less valuable (Beales, 2010).

3.2 Probabilistic model

The model represents the probability a single beacon belongs to a certain profile. The probability depends on the device signal observed and the device it was sent from. The dependencies are shown in the graphical model in figure 3.8, the variables of the model are explained below.

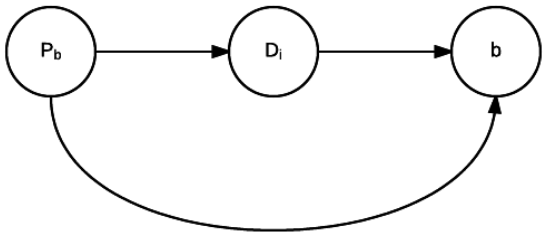


Figure 3.8: Graphical model showing dependencies between entities in the Probabilistic model

First part when observing a device signal is to create the probability distribution for the likelihood of the device signal being sent from any of the devices. How these probabilities are calculated is shown in section 3.2.1. This part does not depend on which profile the beacons in the device signal belong to, for that reason the devices that sent the device signals can be determined before going to the second part. In the second part a probability distribution for the likelihood of every unique beacon belonging to a particular profile is created. How these probabilities are calculated is shown in section 3.2.2. The variables and notation used in the model are:

- **Time step** i : It is a discrete time, at every time step i a device signal is observed.
- b is a **Beacon**: It is a pair (x, d) where x is the flag of the beacon and d is the data contained in this beacon. There are four different flags; first party cookie, third party cookie, customer ID and parent ID.
  - Parent ID - The data associated with this flag is a unique string.
  - Customer ID - The data associated with this flag is a unique string and the originating website.
  - Third party cookie (TPC) - The data associated with this flag is a unique string.
  - First party cookie (FPC) - The data associated with this flag is a unique string and the originating website.
- $B_i = \{b_{i,1}, \ldots, b_{i,n}\}$ is the **Device signal** observed at time step i: It is a set of n beacons that are sent from the same device at time step i.
- $B = B_i, B_{-i}$ is all the device signals, which can be separated into the device signal at time i and the device signals at other times.

- D_i is the device label at time step i.
- $D = D_i, D_{-i}$ is all the device labels, which can be separated into the device label at time i and the device labels at other times.
- P_b is the profile label for a specific beacon b.
- $P = P_b, P_{-b}$ is all the profile labels, which can be separated into the profile label for beacon b and all other profile labels P_{-b}

A device label is used to label the device that sent the device signal at every time step. But the profile label is used to label specific beacons, this is done because every beacon in a device signal does not need to have the same profile label.

In the following model the notations are used:

- $\Pr(X, Y|Z)$ is used to denote conditional distributions.
- $\Pr(X = k, Y|Z)$ is used to denote specific conditional probability of the event where X takes the value k.

3.2.1 Modeling the device

Equation 3.1 represents the probability that the device signal at time step i was sent by the device with label l given all the device signals B and all other device labels D_{-i}.

$$\Pr(D_i = l \mid B, D_{-i}) = \frac{Z_l}{\sum_k Z_k} \tag{3.1}$$

where

$$Z_l = \Pr(B_i \mid B_{-i}, D_i = l, D_{-i}) \Pr(D_i = l \mid B_{-i}, D_{-i})$$

The probability of device signal B_i being sent given all other device signals B_{-i}, $D_i = l$ and all other device labels D_{-i} is:

$$\Pr(B_i \mid B_{-i}, D_i = l, D_{-i}) = \prod_{b \in B_i} f(b, D_i = l, B_{-i}, D_{-i}) \tag{3.2}$$

where

$$f(b, D_i = l, B_{-i}, D_{-i}) = \begin{cases} \dfrac{I(b,l) + C}{\sum_{d \in D_{-i}} O(l,d) + C}, & \text{if b is in a device signal sent} \\ & \text{by a device with the device label } l. \\ \dfrac{C}{U(B_{-i})}, & \text{otherwise.} \end{cases} \tag{3.3}$$

$$\begin{aligned} I(b,l) &= \text{How often beacon } b \text{ is in a device signal sent by a device} \\ &\quad \text{with the device label } l \\ C &= \frac{1}{|D|} \\ U(B_{-i}) &= \text{Number of unique beacons in device signals } B_{-i} \\ O(l,d) &= \begin{cases} 1, & \text{if device label } l \text{ is the same as device label } d. \\ 0, & \text{otherwise.} \end{cases} \end{aligned}$$

The probability of $D_i = l$ given all other device signals B_{-i} and all other device labels D_{-i} is:

$$\begin{aligned} \Pr(D_i = l \mid B_{-i}, D_{-i}) &= \Pr(D_i \mid D_{-i}) \\ &= \frac{\sum_{d \in D} O(D_i, d)}{|D|} \end{aligned} \tag{3.4}$$

3.2.1.1 Gibbs Sampling

To be able to determine which device sent the device signal at every time step i, a Gibbs sampler can be used where we assume a Dirichlet-Multinomial model and then take the expectations. The algorithm for the Gibbs sampler is shown in algorithm 1.

Algorithm 1 Gibbs Sampler, Device Model

```
1: procedure GIBBS SAMPLER
2:     Initialize device labels :
3:     for max iterations do
4:         for i = 1...max time step do
5:             α = []
6:             for all device labels l do
7:                 α[l] ← Z_l
8:             end for
9:             p_l ~ Dirichlet(α), ∀l
10:            Set D_i = l with probability p_l
11:        end for
12:    end for
13: end procedure
```

A batch of device signals is processed at the same time, the output from the sampler are device labels for the device signals. In the initialization part of the sampler, the device signals that share a TPC or FPC are given the same device label. In literature it is common to execute at least 10000 iterations in a Gibbs sampler where the initial 1000 iterations are discarded (also called burn-in period) and every 10th or 20th iteration is stored (Raftery and Lewis, 1992). Our model implementation is very computational heavy which makes it infeasible to create 10000 Gibbs samples, for this reason and time restrictions of the thesis we will limit the number of iterations to 10 and the burn-in period will be 3 iterations. We believe that 10 iterations will be sufficient because of good initialization of the device labels and sparsity of the data.

Looking at one Gibbs sample is not correct, one should look long sequence of Gibbs samples, as was mention in section 2.3. We will assume that the Gibbs sampler has converged if three consecutive samples after the burn-in period show less than 10% decrease in change of device labels. If the label of the device that sent a device signal is different than from previous iteration it counts as a change.

Results from execution of the Gibbs sampler on different batch sizes are in table 3.2. All executions except the one with 20000 device signals converged within 10 iterations. The results are as expected as every second around 1000 device signals are collected which makes the data set sparse. Making it unlikely to observe device signal from the same device in the same batch of device signals. So the number of device labels are expected to be high, close to the number of device signals in the batch.

# Device signals	# Itr.	# Changes each Itr.	# labels	Time[s]/Itr.
100	6	[0, 0, 0, 0 ,0 , 0]	100	1.08
500	6	[1, 1, 0, 0, 0, 0]	496	1.85
1000	9	[0, 1, 0, 0, 1, 0, 0, 0, 0]	990	5.00
2000	6	[10, 5, 0, 0, 0, 0]	1958	20.78
4000	9	[15, 7, 1, 0, 1,0, 0, 0, 0]	3903	91.87
5000	7	[20, 10, 2, 0, 0, 0, 0]	4862	157.42
10000	8	[49, 26, 6, 4, 0, 0, 0, 0]	9634	724.65
20000	10	[125,61,18,4,1,1,1,2,3,1]	19111	4489.85

Table 3.2: Execution showing the convergence of the Gibbs Sampler on the Device model for different batch sizes of device signals.

3.2.1.2 Expectation-maximization sampler

Similar to the Gibbs sampler the Expectation-maximization (EM) algorithm is an iterative method for finding maximum likelihood. The EM was executed in the same manner as the Gibbs sampler. The initial guess in the EM is the same as the initialization for the Gibbs sampler. The sampler runs 10 iterations or until it converges, convergence is assumed if an iteration makes no changes. A change is defined in the same way as in the Gibbs sampler. The algorithm is shown in algorithm 2.

Algorithm 2 Expectation-maximization, Device Model

```
1:  procedure EM
2:      Initialize device labels :
3:      for max iterations do
4:          for i = 1...max time step do
5:              for all device labels l do
6:                  p_l = Pr(D_i = l | B, D_{-i})
7:              end for
8:              Select the label l that maximizes p_l
9:              Set D_i = l
10:         end for
11:     end for
12: end procedure
```

The EM algorithm was executed on the same batches as the Gibbs sampler. The results are in table 3.3. The EM algorithm computed similar results as the Gibbs sampler with fewer iterations and each iteration is faster. The # device labels is the same for all the batches except the final two. With 10000 device signals EM returns 9636 device labels but Gibbs returns 9634 and with 20000 device signals the EM returns 19127 device labels but Gibbs returns 19111.

# Device signals	# Itr.	# Changes each Itr.	# labels	Time[s]/Itr.
100	1	[0]	100	1.04
500	2	[2, 0]	496	3.26
1000	2	[2, 0]	990	4.89
2000	3	[13, 1, 0]	1958	16.80
4000	3	[21, 3, 0]	3903	82.59
5000	3	[29, 3, 0]	4862	125.39
10000	3	[76, 7, 0]	9636	525.39
20000	3	[179, 21, 0]	19127	1855.01

Table 3.3: Execution showing the convergence of the Expectation-maximization on the Device model for different batch sizes of device signals.

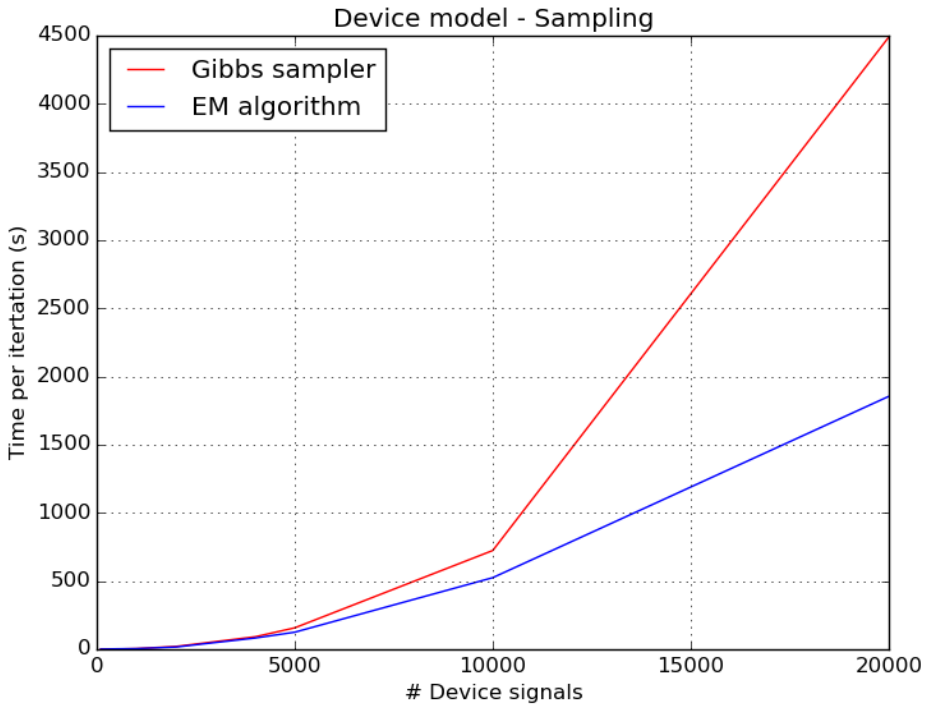


Figure 3.9: Execution showing the time per iterations for both sampling methods on the device model.

In figure 3.9 running time for both sampling methods on the device model is shown. When looking at the time per iteration for the 20000 device signal batch we can see that the time per iteration for the Gibbs sampler is twice as long as for the EM algorithm. From this figure it is clear that the sampling methods are computationally heavy.

3.2.2 Modeling the profile

Equation 3.5 represents the probability that a beacon $b_{i,h}$ which is beacon number h in B_i belongs to a profile with label j given the beacon $b_{i,h}$, all device signals B, all device labels D and all other profile labels $P_{-(i,h)}$.

$$\Pr(P_{b_{i,h}} = j | b_{i,h}, D, B, P_{-(i,h)}) = \frac{S_j}{\sum_{k \in P} S_k} \tag{3.5}$$

where

$$S_j = \sum_{d \in D} \Pr(b_{i,h} | P_{b_{i,h}} = j, B, P_{-b_{i,h}}) \cdot \Pr(d | P_{b_{i,h}} = j, P_{-b_{i,h}}, D, B) \cdot \Pr(P_{b_{i,h}} = j | P_{-b_{i,h}}) \tag{3.6}$$

$$\Pr(P_{b_{i,h}} = j | P_{-b_{i,h}}) = \frac{1}{|P|} \tag{3.7}$$

The probability a device with device label d sent a device signal containing $b_{i,h}$ given the profile label $P_{b_{i,h}} = j$, all other profile labels $P_{-b_{i,h}}$ and all device labels D is:

$$\Pr(d | P_{b_{i,h}} = j, P_{-b_{i,h}}, D, B) = \frac{\sum_{t=1}^{|D|} \sum_{k=1}^{|B_t|} G(P_{b_{i,h}}, P_{b_{t,k}}, d, D_t)}{\sum_{k \in P_{-b_{i,h}}} H(P_{b_{i,h}}, k)} \tag{3.8}$$

where

$$G(j,k,l,d) = \begin{cases} 1, & \text{if device label } l \text{ is the same as device label } d \\ & \text{and if profile label } j \text{ is the same as profile label } k. \\ 0, & \text{otherwise.} \end{cases}$$

$$H(j,k) = \begin{cases} 1, & \text{if profile label } j \text{ is the same as profile label } k. \\ 0, & \text{otherwise.} \end{cases}$$

The probability a beacon $b_{i,h}$ is sent in any device signal from any device given the profile label $P_{b_{i,h}} = j$, all other profile labels and all other device labels is:

$$\Pr(b_{i,h} | P_{b_{i,h}} = j, B, P_{-b_{i,h}}) = ... \tag{3.9}$$

... The number device signals that contain at least one beacon except $b_{i,h}$ with profile label j divided by the number of device signals containing beacon $b_{i,h}$.

3.2.2.1 Gibbs Sampling

To be able to determine which profile the beacons in the device signal time step i belong to, a Gibbs sampler can be used where we assume a Dirichlet-Multinomial model and then take the expectations. A batch of device signals is processed at the same time, the profile model depends on results from the device model, first the batch is processed using the device model and then the profile model. A beacon can only belong to a single profile at a time step i however a beacon can belong to different profiles at different time steps. For that reason we are only interested in the resulting profiles at the last time step. The output of the sampler is what profile each beacon belongs to at the last time step. The initialization step in the sampler assigns every unique beacon a unique profile label. The Gibbs sampler for the profile model uses the same number of maximum iterations and the burn-in period as the Gibbs sampler for the device model. As well the same assumption for convergence is used, where a change is if a profile contains different beacons from last iteration. The algorithm for the sampler is shown in algorithm 3.

Algorithm 3 Gibbs Sampler, Profile Model

```
1:  procedure GIBBS SAMPLER
2:      i = max time step
3:      Initialize profile labels :
4:      for max iterations do
5:          for all unique beacons b do
6:              α = []
7:              for all profile labels j do
8:                  α[j] ← S_j
9:              end for
10:             p_j ~ Dirichlet(α), ∀j
11:             Set P_b = j with probability p_j
12:         end for
13:     end for
14: end procedure
```

The Gibbs sampler for the profile model was executed following the execution of the device model shown in table 3.3. The results are in table 3.4. All executions converged within 10 iterations, except the final two, but likely would have with few more iterations. As was mentioned when the samplers were used on the device model, the data set is sparse and as a result the number of profiles is close to the number of device signals. The results are positive, the number of profiles is less or equal to the number of devices, the model should not be making more profiles than devices because the profile model depends on the device model.

# D. signals	# Itr.	Profile labels changed	# labels	Time[s]/Itr.
100	6	[80, 0, 0, 0, 0, 0]	100	0.23
500	7	[363, 13, 3, 0, 0, 0, 0]	496	2.27
1000	7	[708, 27, 3, 0, 0, 0, 0]	990	9.15
2000	7	[1384, 67, 9, 0, 0, 0, 0]	1957	35.12
4000	10	[2765, 149, 52, 15, 3, 1, 0, 0, 0, 0]	3902	195.26
5000	10	[3426, 184, 45, 10, 6, 1, 0, 0, 0, 0]	4859	298.05
10000	10	[6869, 331, 89, 25, 6, 2, 2, 1, 0, 0]	9626	1405.70
20000	10	[13755, 713, 179, 53, 14, 5, 5, 4, 3, 2]	19093	6899.55

Table 3.4: Execution of the Gibbs Sampler on the Profile model

3.2.2.2 Expectation-maximization sampler

The EM was executed in the same function as the Gibbs sampler. The initial guess in the EM is the same as in the Gibbs sampler above. The sampler runs 10 iterations or until it converges. A convergence is assumed if an iteration shows no change from previous iteration. The algorithm is shown in algorithm 4.

Algorithm 4 Expectationmaximization, Profile Model

```
1:  procedure EM
2:      i = max time step
3:      Initialize profile labels :
4:      for max iterations do
5:          for all unique beacons b ∈ B_1 ... B_i do
6:              for all profile labels j do
7:                  p_j = Pr(P_b = j|b, D, B, P_-b)
8:              end for
9:              Select the label j that maximizes p_j
10:             Set P_b = j
11:         end for
12:     end for
13: end procedure
```

For execution of EM the same batches of device signals were used as in the Gibbs sampler. The results are in table 3.5, where three dots in the column # Changes each Itr. represents the last number repeated until the last iteration. The EM does not converge within 10 iterations, it keeps moving beacons between profiles and does not find a local maximum. This is because a beacon can have equal probability for belonging to different profiles. A possible solution to this would be a different initial guess, but that has not been tested.

# Device signals	# Itr.	# Changes each Itr.	# labels	Time[s]/Itr.
100	10	[84, 4, ...]	104	0.15
500	10	[396, 40, ...]	536	1.97
1000	10	[790, 96, ...]	1086	10.55
2000	10	[1553, 199, ...]	2155	46.38
4000	10	[3088, 398, 397, ...]	4299	179.21
5000	10	[3836, 489, 487, ...]	5347	273.38
10000	10	[7703, 1004, 998, ...]	10625	1025.42
20000	10	[15409, 2004, 1981, ...]	21074	3903.80

Table 3.5: Execution of the Expectation-maximization on the Profile model

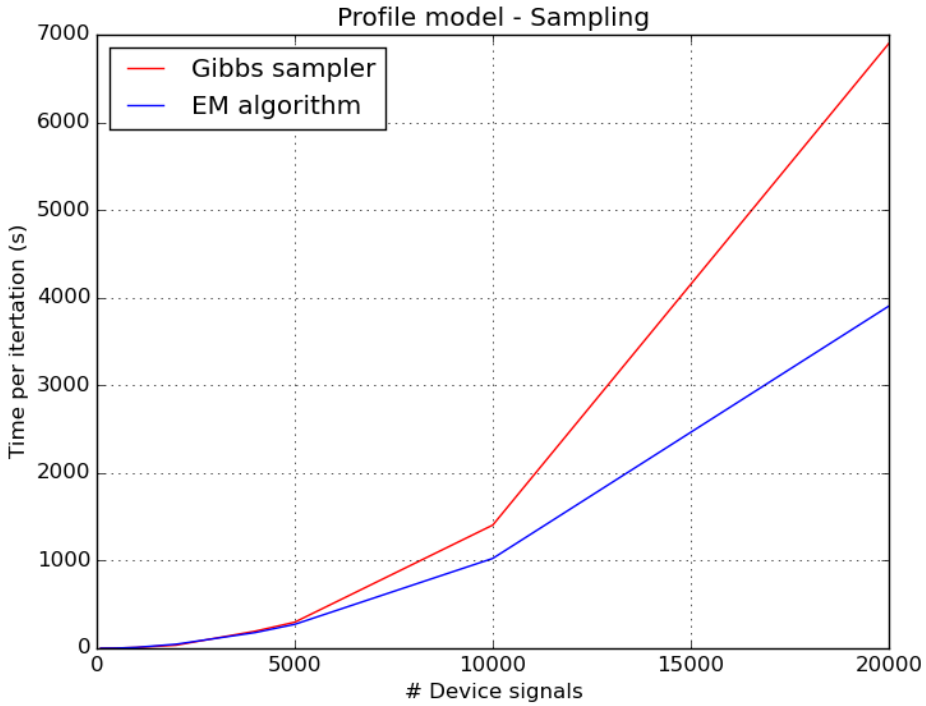


Figure 3.10: Execution showing the time per iterations for both sampling methods on the profile model.

In figure 3.10 running time for both sampling methods on the profile model is shown. When looking at the time per iteration for the 20000 device signal batch we can see that the time per iteration for the Gibbs sampler is about twice as long as for the EM algorithm. Also the time per iteration is longer for the profile model compered with the device model.

4
Designing the Evaluation Methods

The Company has room for improvements in ways to evaluate the quality of the historical profiles. Developing a good method for evaluating the historical profiles is a crucial step before exploring new methods to create historical profiles. In the following sections we will explain the design of two methods for evaluating the quality of historical profiles.

In section 4.1 we introduce a computationally efficient evaluation which could possibly be used for continuous evaluation of the historical profiles. This evaluation could enable the Company to monitor the status of the profiles in real time.

In section 4.2 we introduce an evaluation that uses additional data that is not available in real time. This additional data is available for every device signal and is, for example, IP address and user-agent. In the time-frame of this thesis this evaluation is only possible on a fraction of the profiles due to computational costs.

For the evaluations we will only use profiles that have a parent ID beacon. The profiles that have a parent ID are of most interest to the Company, more demographics are available for profiles with parent IDs than the ones without parent ID. These demographics make it possible to determine with some certainty the age, gender and the location of the user. This is then utilized by X to do target ads to specific group of users and monitor what user groups are using the Websites.

4.1 Agile evaluation

This evaluation uses the profile size and the number of beacons with the flag parent ID to identify profiles that likely represent more than one user, where profile size is the numbers of beacons a profile has.

Profiles that have a parent ID represent a user that is registered on at least one of the Websites. Profiles that have multiple parent IDs will be considered to represent multiple users, which is undesirable and is the first criterion for the evaluation. Although a user can have more than one parent ID caused by error on X's part as was mentioned in section 1.4.2.

Single parent ID in a profile is not a foolproof indicator of a single user, we can't rely solely on that information. For this reason we add a second criterion to the evaluation. We assume that large profile size is an indicator that a profile does not represent a single user. The profile size is expected to grow, user visit new sites, cookies expire and new cookies are set. It is essential to acquire a bound on the expected growth of profile size. The growth depends on the number of websites company X operates and their popularity, no information is available about this

growth and to acquire this bound we use data from the profiles created with the Decision model using the seven days of data in the data set as input. All the profiles that had one parent ID beacon were selected and the increase in profile size was found by comparing the profile size after day 1 and day 7. Profiles that did not increase in size were assumed to represent an inactive user during the time period and were discarded. The results are shown in table 4.1.

Period	1-7
Mean increase	2.97
Standard deviation	2.73
Mean increase per day	0.42

Table 4.1: Mean increase in profile size over a seven day period

The evaluation uses the information from table 4.1 to set the maximum profile size limit depending on the age of the profile.

Day	1
Mean size	4.39
Standard deviation	1.64
Ratio of profiles	0.99

Table 4.2: Mean profile size of one day old profiles with one parent ID

When a profile is created its size is expected to grow faster the first day because the first device signals are being observed. A unique beacon is only counted once in the profile size so when the beacons are being observed for the first time the profile size will grow faster. The same method was used as described to create results in table 4.1. The results are shown in table 4.2.

The criterion for maximum profile size for profiles that are one day old is equal to the results in table 4.2, the mean profile size and three standard deviations. For profiles of age more than one day the maximum profile size increases by the mean profile size increase and three standard deviations shown in table 4.1 multiplied with the profile age in days.

If one or both of the two criteria are violated the profile is classified as a profile that likely represents multiple users otherwise it is classified as likely to represent a single user.

The evaluation gives a rating that is calculated as the ratio of number of profiles classified as likely represented by single user and the number of profiles evaluated. The rating is between 0 and 1, with 1 being the best possible rating.

4.2 Additional data evaluation

Ground truth for which profiles represent a single user is not available. Using additional data gives us the ability to carry out evaluation based on user behavior. To identify unrealistic behavior of a single user we create criteria that we assume to represent unrealistic behavior in a single day. The criteria we assume are shown in table 4.3. We choose the criteria as close to the edge of what we assume to be realistic behavior of a single user. This is done to have higher probability of identifying profiles that likely show behavior of multiple users. If any of these criteria are broken on a single day the evaluation classifies the profile as unrealistic behavior of a single user.

Criterion	Value
Maximum speed	800 km/h
Maximum number of city visits	4
Maximum number of user-agents	6

Table 4.3: Criteria for unrealistic behavior of a single user

The Company utilizes a service that provides geolocation from IP address. The geolocation makes it possible to get the distance between observed device signals that contain beacons from particular profile. By assuming that the profile represents a single user how fast the user would have to be travelling the distance and what cities the user visited. To find the maximum speed of the user we look at all the locations in the time period, the time between observations of those locations and calculate the speed that would be needed to travel these distances. The number of city visits is found by looking at the city of the geolocation; cluster cities that are within 20 km of each other as the same city. Then in chronological order count how often the user changes city.

To set a value for maximum speed we assume a user can travel with an aircraft. As an example flight from Gothenburg to Stockholm takes 55 minutes and the total distance is about 400km. This gives an approximate average speed of about 400km/h. Although this is a relatively short flight and therefore the maximum speed will be set to the average speed of one the longest commercial flights, 800km/h (Flynn, 2013). For the maximum number of city visits we assume the average user unlikely to visit more than 4 cities in a single day and therefore the maximum number of city visits in a single day is set to 4.

Device signals that are sent from mobile networks do not have reliable geolocation and will be ignored in calculations based on geolocation.

The user-agent contains detailed information about a web browser and the OS. User-agent is not likely to change during a one day period (Flood and Karlsson, 2012). Flood and Karlsson (2012) found that less than 5 % of browsers show change in their user-agent over one day period. We assume that most users only use one browser on their device therefore we regard unique user-agent as a unique device. In 2012 less than 5 % of Internet users in Sweden used more than six devices to

connect to the Internet (Findahl, 2013). The maximum number of unique user-agents is therefore set to six.

This evaluation is computationally heavy, for each beacon in a profile it is needed to process the additional data from all the Websites for the period. The Websites gather around 200GB of compressed data in a single day. The number of profiles to evaluate is bound by the time wanted to spend on the evaluation and computational power.

The evaluation method uses the same rating system as the agile evaluation 4.1.

5
Evaluation of the models

For evaluation of the profiles generated by the models discussed in chapter 3 we use the evaluation methods designed in chapter 4. In the evaluation we use the data set described in section 1.4 that covers a seven day period. As was mentioned in chapter 4 we will only evaluate the profiles that have parent ID and in the additional data evaluation method we will only be able to evaluate a portion of the profiles due to limited computational resources. The sampling method used for the Probabilistic model for the evaluation is the Gibbs sampler, this method is computationally heavy as was shown in section 3.2. To work around this problem and create interesting results we select the profiles that were created by the Decision model and evaluated with the additional data evaluation method. Finally we do comparisons of the evaluation methods using McNemar's test, where the goal is to determine if both evaluation methods return similar results.

5.1 Decision model

The decision model is efficient and we were able to use the whole data set of device signals as input, for a period of seven days. In the following subsections the profiles generated by the decision model are evaluated using both evaluation methods.

5.1.1 Agile evaluation

The Agile evaluation was applied on the profiles created after each day of data that contain a parent ID beacon; the results are shown in table 5.1. The table shows the rating decreasing, number of profiles is increasing and the mean and standard deviation of beacons per profile are increasing. The is profile size is expected to increase because more users have been active and have used more websites as the days go by.

Day	1	2	3	4	5	6	7
Rating	0.981	0.975	0.976	0.976	0.974	0.977	0.974
# profiles	90168	129401	156131	176250	195098	212502	222984
Mean beacons/profile	4.427	4.776	4.999	5.144	5.293	5.452	5.566
Std beacons/profile	1.719	2.048	2.289	2.457	2.614	2.824	2.967

Table 5.1: Agile evaluation results of the profiles created by the Decision model.

Table 5.2 shows how many times each criterion of the Agile evaluation was broken by the profiles generated from the seven days of data. The table lists the results after each day.

Day	1	2	3	4	5	6	7
Max profile size	758	1447	1181	1249	973	1133	895
Max number of Parent IDs	1069	2054	2823	3439	4068	4816	5222
# profiles that violated either criterion	1680	3223	3727	4382	4787	5622	5841

Table 5.2: Number of profiles created by the Decision model that violate each criterion of the Agile evaluation.

Figure 5.1 and 5.2 show a scatter plot and a histogram of profiles with parent ID after day one and day seven. The scatter plot shows the distribution of the number of parent IDs versus profile size. The scatter plot alone does not show the number of profiles behind each point, this is shown with the histogram in the same figure. From the histogram we can see that the majority of the profiles lay well within the profile size limit. The zone that represents profile that likely is a single user is also shown, the dashed lines.

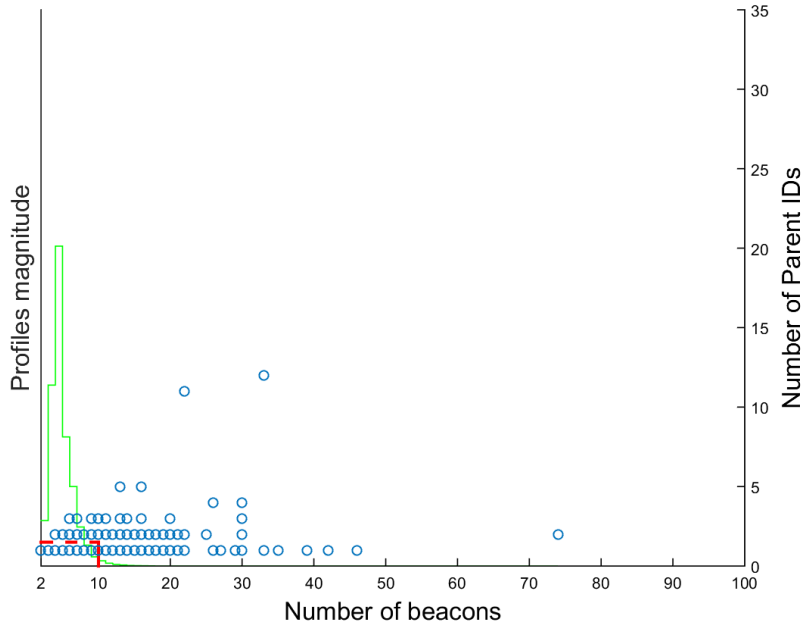


Figure 5.1: Day 1. Scatter plot: number of Parent ID vs Profile size. Histogram: Distribution of profile size. Area inside the dashed lines shows the zone where profiles likely represent a single user.

By comparing figures 5.1 and 5.2 it is clear that both the profile size and the number of parent IDs per profile is increasing. Although majority of them still lie within the zone representing the profiles that likely represent one user.

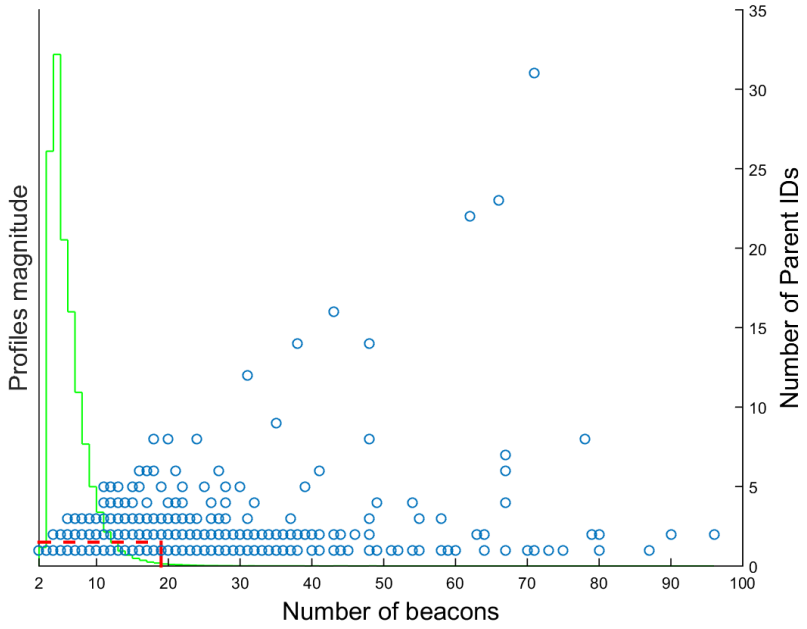


Figure 5.2: Day 7. Scatter plot: number of Parent ID vs Profile size. Histogram: Distribution of profile size. Area inside the dashed lines shows the zone where profiles likely represent a single user.

5.1.2 Additional data evaluation

Due to how computational heavy this evaluation is we are not able in the time frame of this thesis to apply the additional data evaluation on all profiles with a parent ID created by the Decision model. For this reason we will use the classification done by the Agile evaluation method and for each day of input randomly select 100 profiles that are classified as profiles that likely represent a single user and 100 profiles that likely represent multiple users. This will then allow us to compare the classifications between the Agile method and the Additional data method. The Additional data evaluation was applied on these profiles and the results are shown in table 5.3.

Day	1	2	3	4	5	6	7
Rating	0.985	0.980	0.985	0.990	0.990	0.965	0.995
# profiles	200	200	200	200	200	200	200
# device signals	2923	6315	8955	9800	12235	13924	14408

Table 5.3: Additional data evaluation results of selected profiles created by the Decision model, 200 profiles from each day.

An important thing to keep in mind when looking at table 5.3 is that half of the profiles evaluated were classified as likely representing multiple users by the Agile method. It shows that there is not the same decrease in the rating of the profiles as in the agile evaluation. In table 5.4 the number of times each criterion was broken by a profile each day is displayed.

Day	1	2	3	4	5	6	7
Maximum speed	1	4	3	1	1	4	1
Maximum number of city visits	0	0	0	0	0	0	0
Maximum number of user-agents	3	1	1	1	1	4	0
# profiles that violated at least one criterion	3	4	3	2	2	7	1

Table 5.4: Number of profiles created by the Decision model that violate each criterion of the Additional data evaluation

The results from table 5.4 show that only a fraction of the profiles break any criteria. Half of the profiles evaluated with the Additional data evaluation were classified as profiles likely to represent multiple users we therefore expect the rating from the evaluation to be 0.5. A detailed discussion about the reason for these differences between the two evaluation methods is in section 5.3.

5.2 Probabilistic model

The Probabilistic model is very computational heavy, as was shown in section 3.2, for that reason it is not feasible in the time frame of this thesis to evaluate the all the profiles evaluated for the decision model. Therefore a fraction of the profiles from the Decision model were selected and used for input to the Probabilistic model, the same profiles as were selected in the Additional data evaluation of the Decision model. This is done by locating all device signals that contain at least one beacon that is in the selected profiles. These device signals are then used as input to the Probabilistic model. This work around is possible because the Decision model adds all beacons that are contained in the same device signal to the same profile, see section 3.1. This gives a manageable amount of device signals that can be processed by the Probabilistic model and create profiles in the time frame of this thesis. The profiles created by the Probabilistic model are then evaluated with the same methods used for the Decision model.

5.2.1 Agile evaluation

The device signals from the selected profiles mentioned above were used as input to the Probabilistic model separately for each of the seven days. As the days increase the device signals accumulate, resulting in more device signals to feed to the probabilistic model each day. The results from the Additional data evaluation on the profiles are in table 5.5.

Day	1	2	3	4	5	6	7
Rating	0.507	0.503	0.500	0.542	0.531	0.560	0.536
# profiles	203	203	200	212	209	216	207
# device signals	2923	6315	8955	9800	12235	13924	14408
Mean beacons/profile	7.064	8.049	8.990	9.217	9.292	9.528	9.773
Std beacons/profile	4.973	4.306	5.520	6.253	5.655	6.990	6.746

Table 5.5: Agile evaluation results of the profiles created by the Probabilistic model.

A important thing to keep in mind when looking at table 5.5 is that the device signals used are from the selected profiles that were used in the Additional data evaluation method in section 5.1.2, where half of the profiles evaluated were classified as likely to represent multiple users by the Agile evaluation. If the Probabilistic model would perform the same as the Decision model we would expect the rating to be 0.5. So rating above 0.5 is a indicator that the Probabilistic model outperformed the Decision model.

5.2.2 Additional data evaluation

The same profiles that were evaluated by the Agile evaluation were evaluated with the Additional data evaluation. The results are shown in table 5.6.

Day	1	2	3	4	5	6	7
Rating	0.980	0.980	0.985	0.991	0.990	0.968	0.995
# profiles	203	203	200	212	209	216	207
# device signals	2923	6315	8955	9800	12235	13924	14408

Table 5.6: Additional data evaluation results on profiles created by the Probabilistic model.

As was mentioned for table 5.3 it is needed to keep in mind that around half of the profiles in table 5.6 were classified as profiles that likely represent multiple users by the Agile method. Table 5.7 shows the number of times each criterion was broken by a profile in the Additional data evaluation.

Day	1	2	3	4	5	6	7
Maximum speed	2	4	3	1	1	4	1
Maximum number of city visits	0	0	0	0	0	0	0
Maximum number of user-agents	3	1	1	2	1	4	0
# profiles that violated at least one criterion	4	4	3	2	2	7	1

Table 5.7: Number of profiles created by the Probabilistic model that violate each criterion of the Additional data evaluation.

The results from table 5.7 show that a fraction of the profiles break any criteria. The results show that there is great difference between the Agile and Additional data evaluation. A more detailed discussion about possible reason are in section 5.3.

5.3 Comparison of evaluation methods

To be able to determine if the results from the two evaluation methods are different a statistical test will be used. We chose the McNemar's test because we have two evaluation methods that use data from the same domain and McNemar's test is used to analyse matched pair data. The goal of the comparison is to see if the Additional data evaluation and the Agile evaluation are significantly different from each other. If they turn out not to be different the more efficient Agile evaluation could be used instead of the more computationally heavy Additional data evaluation to evaluate profiles in real time. The statistical test is applied on results from the evaluation methods for both the models and the results are in the following sections.
In the McNemar's test the following definitions are used:

$EAgile_N$:=The profiles that the Agile evaluation classified to be profiles that likely represent multiple users.

$EAgile_P$:=The profiles that the Agile evaluation classified to be profiles that likely represent a single user.

$EAdd_N$:=The profiles that the Additional data evaluation classified to be profiles that likely represent multiple users.

$EAdd_P$:=The profiles that the Additional data evaluation classified to be profiles that likely represent a single user.

$EAgile_N$ and $EAdd_N$:=The profiles that both the Agile and the Additional data evaluation classified to be profiles that likely represent multiple users.

$EAgile_N$ and $EAdd_N$:=The profiles that both the Agile and the Additional data evaluation classified to be profiles that likely represent a single user.

$EAgile_P$ and $EAdd_N$:=The profiles that the Agile evaluation classified to be profiles that likely represent a single user and the Additional data classified to be profiles that likely represent multiple users.

$EAgile_N$ and $EAdd_P$:=The profiles that the Agile evaluation classified to be profiles that likely represent multiple users and the Additional data classified to be profiles that likely represent a single user.

5.3.1 Using the Decision model

The results used to compare the two evaluation methods are the profiles created by the Decision model i.e. the 200 profiles that were evaluated using the Additional data evaluation. Table 5.8 lists the results for the seven day period from the two evaluation methods on the 200 profiles created by the Decision model.

Day	1	2	3	4	5	6	7
$EAgile_N$ and $EAdd_N$	2	3	2	2	2	7	0
$EAgile_P$ and $EAdd_N$	1	1	1	0	0	0	1
$EAgile_P$ and $EAdd_P$	99	99	99	100	100	100	99
$EAgile_N$ and $EAdd_P$	98	97	98	98	98	93	100

Table 5.8: Comparison of the results from the evaluation methods on the Decision model.

The results from table 5.8 are accumulated and displayed in table 5.9 that is the 2×2 contingency table for the McNemar's test.

	$EAdd_P$	$EAdd_N$	Row total
$EAgile_P$	696	4	700
$EAgile_N$	682	18	700
Column total	1378	22	1400

Table 5.9: Contingency table for results of two evaluation methods on the Decision model for the seven day period.

To determine if the difference between number of discordant pairs is higher than we would expect by chance the McNemar's test is applied. We select a p-value as $p^* = 0.001$.The result from the test is:

$$\chi^2 = \frac{(4 - 682)^2}{4 + 682} = 670.093 \quad \text{and } p < 0.00001$$

The result from the McNemar's test provides strong evidence for statistically significant difference between the evaluation methods.

5.3.2 Using the Probabilistic model

The results used to compare the two evaluation methods are the results from the evaluation of the profiles created by the Probabilistic model in section 5.2. Table 5.10 lists the results from the two evaluation methods on the profiles created by the Probabilistic model over the seven day period.

Day	1	2	3	4	5	6	7
$EAgile_N$ and $EAdd_N$	2	3	2	2	1	5	0
$EAgile_P$ and $EAdd_N$	2	1	1	0	1	2	1
$EAgile_P$ and $EAdd_P$	101	101	99	110	105	113	108
$EAgile_N$ and $EAdd_P$	98	98	98	95	97	90	96

Table 5.10: Comparison of the results from the evaluation methods on the Probabilistic model.

	$EAdd_P$	$EAdd_N$	Row total
$EAgile_P$	737	8	745
$EAgile_N$	672	15	687
Column total	1409	23	1432

Table 5.11: Contingency table for results of two evaluation methods on the Probabilistic model for the seven day period.

The results from table 5.10 are accumulated and displayed in table 5.11 that is the 2×2 contingency table for the McNemar's test. We select a p-value as $p^* = 0.001$

$$\chi^2 = \frac{(8 - 672)^2}{8 + 672} = 648.376 \quad \text{and } p < 0.00001$$

The results from the McNemar's test provides a strong evidence for statistically significant difference between the evaluation methods.

The results from the McNemar's tests tell us that there is a significant difference between the evaluation methods, which is what was expected. For the Additional data evaluation to be able to classify profiles that likely represent multiple users; both users have to be active within relatively short period of time or be located far apart. As mentioned in the designing of the Agile evaluation method an error in X's method of linking customerIDs to a parentID can cause a profile that represents a single user to have multiple parentID. However the Agile evaluation method will classify that profile as likely to represent multiple users. The McNemar's test can not tell us which evaluation method is more accurate but it tells us that the two evaluation methods can give different results.

6
Conclusions

To compare the quality of the historical profiles created by the Decision model and the Probabilistic model the two evaluation methods were used. Before comparing the quality of the historical profiles created by the models a discussion of the limitation of the evaluation methods is needed.

6.1 Limitation of the evaluation methods

The Agile evaluation is a rough evaluation that classifies many profiles as likely to represent multiple users. In table 5.2 we see that most of the profiles that were classified as likely to represent multiple users violated the criterion of having at most one parent ID. The Company assumes that multiple parent IDs is an indicator of multiple users, but we have no information on the quality of that assumption, as discussed in section 1.4.2 there can be some errors.

The Additional data evaluation is conservative and classifies few profiles as likely to represent multiple users as shown in table 5.4. The reason for this is because for a profile to violate for example the maximum speed criterion the users of the profile would have to be visiting websites within a short time period. As well the maximum user agent criterion is set very high to cover 95% of Swedish population as was mentioned in section 4.2, but in the authors experience it is uncommon to use so many devices in a single day. Another limitation of the Addition data evaluation is when users use a Virtual private network (VPN) service, this can cause the user to have different geolocation within a very short time period resulting in violation of the maximum speed criterion.

Both evaluation methods do not have a lower bound for the number of beacons within a profile, as a result the evaluation methods will favor a method that creates more profiles from the same input.

6.2 Comparing models

Comparing the quality of profiles created by the Decision model and the Probabilistic profile with results from the Agile evaluation we see from table 5.5 that the rating of the Probabilistic model is from 0.5 to 0.56. A rating of 0.5 means that the profiles are of the same quality as the ones created by the Decision model. Based on the results from the Agile evaluation method the Probabilistic model shows slight improvement over the Decision model.

Compering the quality of profiles created by the Decision model and the Probabilistic profile with results from the Additional data evaluation we see from table 5.3 and table 5.6 that there is no significant difference in the rating. Based on the results from the Additional data evaluation method the Probabilistic model shows similar results as the Decision model.

6.3 Probabilistic model

Even though the results from the evaluation methods do not show significant improvements of the quality of the profiles created by the Probabilistic model, we believe that the Probabilistic model can handle some of the pitfalls of the Decision model that were discussed in section 3.1.3 better because the model is based on probabilities that rely on the number of observation of beacons. For the Probabilistic model to function well it needs device signals over a long period. From the results of the evaluation we believe that using more than one week of device signals will improve the quality of the profiles.

7
Future work

The most important future work is to improve or find a better way of evaluating the quality of historical profiles. More collaboration could be done with X to filter out parent IDs that are not reliable which would improve the Agile evaluation. As well a criterion that limits the number of profiles created has to be designed to prevent false good rating of models that create too many profiles. To prevent this limitation we propose a penalty if FPC and TPC that were in the same device signal are added to different profiles.

The Additional data evaluation could also be improved by taking into account that a user can use VPN when visiting websites and also more research could be done on realistic behavior of a single user.

Even though the Probabilistic model is not showing significant increase in rating from the Agile evaluation method we believe that using model-based clustering to create historical profiles is in the right direction. With more time there could be done more testing with the Probabilistic model, using a data set that covers a longer time period and tune for example the likelihood of beacons moving between profiles or creating new profiles and test different assumptions for convergence when using the Gibbs sampler.

While doing the thesis we came across an interesting algorithm, Markov Cluster Algorithm(MCL), a fast and scalable unsupervised cluster algorithm for graphs (van Dongen and Abreu-Goodger, 2012). To use this algorithm the observed device signals could be setup as a graph, by creating a edge between beacons that are sent in the same device signal, where the edge has a weight that would increased each time beacons are observed in the same device signal. Due to the time-frame of the thesis work we did not have time to test this algorithm.

Doing the thesis we did not focus on making the Probabilistic model distributable, as was mention in section 1.5. During the process of creating the Probabilistic model we came up with a idea to parallelize the execution. It entails the use of the Decision model and regularly execute the Probabilistic model using the profiles created by the Decision model.

In detail the method we propose to parallelize the Probabilistic model will use the Decision model to create profiles in real-time because it is fast and in addition the Decision model would keep track of all device signals used to create each profile. Then at certain times the Agile evaluation method would be executed on the created profiles and all the profiles it classifies as likely to represent multiple users are selected. The device signals used to create these profiles would be used as input to the Probabilistic model. Given the nature of the Decision model a batch of device signals that created a single profile can be used as input separately making it

possible to execute multiple instances of the Probabilistic model in parallel where each input is a batch of device signals used to create single profile.

Bibliography

Apple. Safari, 2015. URL http://www.apple.com/safari/.

Adam Barth. HTTP State Management Mechanism. RFC 6265, RFC Editor, April 2011. URL http://tools.ietf.org/html/rfc6265.

Howard Beales. The Value of Behavioral Targeting. *Network Advertising Initiative*, 2010. URL http://www.networkadvertising.org/pdfs/Beales_NAI_Study.pdf.

Igor Cadez, David Heckerman, Christopher Meek, Padhraic Smyth, and Steven White. Visualization of Navigation Patterns on a Web Site Using Model-based Clustering. In *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '00, pages 280–284, New York, NY, USA, 2000. ACM. ISBN 1-58113-233-6. doi: 10.1145/347090.347151. URL http://doi.acm.org/10.1145/347090.347151.

Darren Charters. Electronic monitoring and privacy issues in business-marketing: The ethics of the doubleclick experience. *Journal of business ethics*, 35(4):243–254, 2002.

David Maxwell Chickering and David Heckerman. Efficient approximations for the marginal likelihood of Bayesian networks with hidden variables. *Machine Learning*, 29(2-3):181–212, 1997.

Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38, 1977.

Chuong B Do and Serafim Batzoglou. What is the expectation maximization algorithm? *Nature biotechnology*, 26(8):897–899, 2008.

Facebook. Measuring Conversions on Facebook, Across Devices and in Mobile Apps, 2015. URL https://www.facebook.com/business/news/cross-device-measurement.

O Findahl. Svenskar och internet. *.SE (Stiftelsen för internetinfrastruktur)*, 2013.

Erik Flood and Joel Karlsson. Browser fingerprinting. Master thesis, Chalmers University of Technology, 2012.

David Flynn. Qantas claims "world's longest flight" for Sydney-Dallas route, 2013. URL http://www.ausbt.com.au/qantas-claims-world-s-longest-flight-for-sydney-dallas-route.

Chris Fraley and Adrian E Raftery. Model-based clustering, discriminant analysis, and density estimation. *Journal of the American statistical Association*, 97(458): 611–631, 2002.

Andrew Gelman, John B Carlin, Hal S Stern, and Donald B Rubin. *Bayesian data analysis*, volume 2. Taylor & Francis, 2014.

Stuart Geman and Donald Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (6):721–741, 1984.

Google. Limits of User ID views Cross Device reports, 2015. URL https://support.google.com/analytics/answer/3223194?hl=en.

Anagha Joshi, Yves Van de Peer, and Tom Michoel. Analysis of a Gibbs sampler method for model-based clustering of gene expression data. *Bioinformatics*, 24 (2):176–183, 2008.

Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques*. MIT Press, 2009.

Marilyn Lavin. Cookies: What do consumers know and what can they learn? *Journal of Targeting, Measurement and Analysis for Marketing*, 14(4):279–288, 07 2006. URL http://search.proquest.com/docview/236966995.

Volodymyr Melnykov. Challenges in model-based clustering. *Wiley Interdisciplinary Reviews: Computational Statistics*, 5(2):135–148, 2013.

David L Poole and Alan K Mackworth. *Artificial Intelligence: foundations of computational agents*. Cambridge University Press, 2010.

Adrian E Raftery and Steven Lewis. How many iterations in the Gibbs sampler. *Bayesian statistics*, 4(2):763–773, 1992.

Justine Rapp, Ronald Paul Hill, Jeannie Gaines, and R Mark Wilson. Advertising and consumer privacy. *Journal of Advertising*, 38(4):51–61, 2009.

Daniel Stahl and Hannah Sallis. Model-based cluster analysis. *Wiley Interdisciplinary Reviews: Computational Statistics*, 4(4):341–358, 2012.

Stijn van Dongen and Cei Abreu-Goodger. Using MCL to extract clusters from networks. In *Bacterial Molecular Networks*, pages 281–295. Springer, 2012.