Beyond respondent-level representation of multi-channel Nielsen ratings. A Bayesian networks approach

Master thesis
MSc in Econometrics and Management Science
Business Analytics & Quantitative Marketing

Armin Reinert
433814

Supervisors: Prof. dr. R. Paap, Harald Hoogstrate
Second assessor: Prof. dr. Dennis Fok

Erasmus School of Economics

Rotterdam/Wroclaw, 04/2018

Contents

1 Introduction

The importance of understanding patterns in media behaviour for planning of marketing campaigns is difficult to overestimate. It allows to target desirable audiences more accurately, avoid reaching the same people more times than companies wish, or efficiently use multiple advertising channels.

Scientific research offers useful insights into general trends or typical behaviours in TV watching or Internet browsing (such as Kim, 2002; Liu, 2010 and Mora, 2010). While valuable for high-level strategies and adjusting marketing plans to general changes in consumer habits, these methods are not well suited for answering very detailed questions regarding specific demographic groups and their consumption of particular TV programs or websites. Such in-detail investigation, based on as up-to-date data as possible, is often required for a day-to-day marketing planning. Published research is often too general or based on not the latest data for these needs.

This gap is filled by companies such as Nielsen, which offers tools, allowing its clients to perform detailed analysis of consumer media consumption adjusted to their needs. This is possible by utilization of respondent level data. Nielsen collects data on TV viewing and Internet usage patterns for an extensive set of respondents, along with very detailed demographic information about them. This data are then preprocessed and made accessible to the clients in a form, which allows for efficient exploration.

Most important metric of interest for companies is a reach of a TV program/website. Reach is defined as the percentage of people in certain (sub)population (e.g. males over 21 years old, or simply whole country population), which has been exposed to a particular media content. With use of respondent-level data reach is estimated by filtering a set of respondents belonging to a subpopluation of choice, and than calculating percentage of them exposed to a TV program/website.

Simply filtering respondents' level data to obtain insights is not flawless. Firstly, it offers no statistical verification of differences in reach between subpopulations, which may lead to perception of some randomly occurring dependencies of reach on other variables as true ones. Secondly, assessing an impact of multi-channel campaigns requires consumption patterns for all media being available for all respondents. This is often not the case as gathering such detailed information for every respondent would not be easy. That is why the Internet and traditional media behaviour are often collected on separate sets of respondents, typically only with a small overlap. Such set of respondents for which full media consumption is available is too small to be reliable, especially for very specific subpopulations. At the same time separate datasets for Internet and traditional media do not offer information on full media consumption of all respondents. To deal with this issue Nielsen performs a data fusion based on linking respondents by their demographic profiles - e.g. Internet behaviour of one respondent is assigned to a similar (the same age and gender) person's TV viewing pattern. Nevertheless, the company experience to-date shows limited effectiveness of this method. Strong assumption that Internet browsing and TV watching behaviour can be linked solely on the basis of a person's demographic profile seems to lower its accuracy. Last

but not least, storing and sharing with clients respondent-level datasets is becoming more and more difficult from the technical point of view, due to their increase in size. As a result, more and more powerful computational infrastructure is required, both on client's and Nielsen's side.

All these challenges motivate the search for an efficient method of representing media consumption patterns in a way which combines at least some statistical verification of the dependencies in the data and ease of their distribution among clients typical for econometric models, with flexibility with which respondent level data can be used to answer quickly even very specific questions of managers without econometric or statistical knowledge. To our best knowledge no research addressing this issues directly exists. Literature on recommender systems shows some similarities to our problem, but developed methods are not designed nor tested for delivering flexible insights on media reach among different subpopulations. Also, specific to our case data structure causes difficulties with the use of most of the existing methods. To address these shortcomings we assess Bayesian networks (Koller and Friedman (2009)) usefulness for this purpose.

Bayesian networks (BN) are probabilistic graphical models, which allow for effective representation and inference on joint distributions over a set of random variables. It is possible by utilizing graphical representation of relationships between random variables and exploiting conditional independencies between them. They are successfully used for a variety of purposes e.g. in bioinformatics (Friedman et al., 2000), medicine (Uebersax, 2004), image processing (Kolekar and Sengupta, 2015) and many others. They have also been widely adapted for recommender systems, which often requires data similar to ours.

Our goal is to create a network containing socio-demographic variables, as well as media content variables, which we define as ones indicating if a person has been "reached" by particular TV program or website. Such network can be used as an efficient inference engine for determining how socio-demographic characteristics or consumption of some media content influences chances of being exposed to any other TV program or website included in the model. The complexity of our problem, especially the fact of dealing with only partially overlapping datasets and a high number of variables means that existing in the literature solutions to learning Bayesian networks are inadequate. Also, little has been done so far to establish Bayesian networks ability to explain media consumption with socio-demographic variables. Connections between Internet usage and TV viewing are as well a sparsely researched topic.

In an attempt to fill this gap a new framework for learning the Bayesian networks on two, only partially overlapping datasets with a high number of variables is proposed. In our case one dataset consists of the TV viewing behaviour of some respondents, and second of Internet browsing behaviour of another set of respondents. Some subset of overlapping respondents can be found in both. We suggest a step-wise approach. It uses well-established methods of learning Bayesian networks on a single dataset, and aims to learn network with variables from both datasets while maximiz-

ing use of information contained both in an overlapping and non-overlapping parts of the data. At the same time, we propose a new method, based on earlier research, but tailored to our problem, to learn a network with high number of variables. We explore three alternative ways of conducting it. In the process, we also assess the general ability of Bayesian networks to explain consumers media consumption with their socio-demographic characteristics. We note that none of known to us methods can simultaneously learn model based on two only partially overlapping datasets using information contained in both overlapping and non-overlapping parts of data, and be computationally efficient on a dataset with high number of variables.

The thesis is structured as follows - firstly we review existing literature on use of Bayesian networks for capturing media consumption. Then, we describe data on TV and online media consumption, provided by Nielsen, which we use for empirical testing of our framework. In the next chapter we introduce necessary theoretical background on Bayesian networks, and present our proposed learning framework. Finally, we partially apply it to the reduced dataset of only most-watched TV programs and visited websites, to establish some best practices of learning Bayesian networks on our data and to compare three alternative approaches with which we propose to deal with high number of variables. Establishing best practices and testing alternative approaches is necessary taking into consideration limited literature on the topic. Use of reduced dataset makes it computationally feasible. We continue by presenting results of the framework application to the full data. We end with a summary of the findings and suggestions for further research.

2 Literature review

Bayesian networks were employed for representing consumer behaviour in various settings, among them, modelling consumer complaint process (Blodgett and Anderson, 2000), predicting a purchasing behaviour online (Kooti et al., 2016) and in-store (Zuo et al., 2015), or modelling consumers loyalty towards online retailers (Jaronski et al., 2001).

They are, however, most widely used for recommender systems, for both websites and viewing content. Naïve Bayes classifiers, which are restricted version of a Bayesian network, proved to be particularly effective and popular for this task (Catherine and Cohen, 2016). Recommender systems seek to predict a preference of an user regarding some item. Such predictions can be based either on the behaviour of similar users (collaborative filtering) or similarity of items to previously consumed content (content-based filtering). These approaches are also often combined in hybrid recommender systems (Adomavicius and Tuzhilin, 2005).

In one of the first applications of Bayesian networks to collaborative filtering they are shown to outperform several other methods (Breese et al., 1998). For comparison authors use independently three datasets – EachMovie containing users' ratings of films, MS Web composed of visits to various websites and Nielsen television data of ratings regarding network programs. The last two of them closely resemble ours

data. They use a variant of Bayesian network, that instead of explicit variables distributions contains a decision tree in each node, which is not a common solution. It seems, however, to perform well in this setting. Noteworthy, authors underline not only accuracy as important advantage of Bayesian networks but also the feasibility of delivering it to a user – many other collaborative filtering methods (mostly memory based ones) require making available not only a model but also at least part of the database. It is not the case for Bayesian networks. While it applies to any statistical models Bayesian networks seemed at the time to be the best suited one for the task.

Partially building on the previous paper, Heckerman with colleagues (Heckerman et al., 2000) introduces application of dependency networks (DN) to collaborative filtering task. DNs are models based on Bayesian networks, but allowing for cycles in the graph. Being less statistically efficient, which results in lower accuracy, DNs are less computationally expensive for training and inference. Their effectiveness is compared to similar variant of Bayesian networks as in (Breese et al., 1998) on three datasets – two based on websites visits (MS.COM and MSNBC), and one about whether or not users watched five or more minutes of network TV shows aired during a two-week period in 1995. The paper discusses as well usefulness of graphical models as BN or DN for visualization of predictive relationships on an example of internet-use data paired with demographic variables from Media Metrix.

Computational problems related to training and inference in fully connected networks, especially in business practice (Baldi et al., 2003), quickly motivated researchers to switch to restricted versions of Bayesian networks as Naïve Bayes classifiers or Tree-Augmented-Networks (Su and Khoshgoftaar, 2009). Strong restrictions on structure of a network in these models eliminate almost entirely a need of structure learning, which is the most complicated learning task. The downside of such approaches is that discovery of relationships between variables is lost on behalf of an imposed structure. Also, such models can typically explain a probability of exposure to only one media content variable at once, which creates a need to learn multiple separate models, one for each item. These simplified classifiers proved, however, to be very effective, which lead to their wide usage, such as in (Miyahara and Pazzani, 2000, 2002; Su and Khoshgoftaar, 2006).

The popularity of simplified classifiers does not mean that the usage of more complex networks has been abandoned. They have been successfully used e.g. to create hybrid recommender systems (De Campos et al., 2010) by representing both users and items as nodes in a net or to include time component into prediction (Lee et al., 2011). All these solutions do not address the main issues we are faced with - learning a network on only partially overlapping datasets with a high number of variables.

While the earliest uses of Bayesian networks for recommender systems closely reflect the problem we are concerned with, further research moves away from it, which is to be expected taking into consideration that its goals are different. However, even early work is insufficient in our case. First of all, it focuses only on evaluation of predictive accuracy, ignoring statistical validation of fit the distribution with means

of likelihood evaluation, which is suggested for purpose of domain knowledge discovery (Koller and Friedman, 2009), with which we are also concerned. Secondly, despite the fact that the usefulness of representing both TV and Web data is tested, representation of both medias in one network, which is one of ours primary goals, is not conducted. Also, demographic variables are included in only one case, and their exploratory power with respect to media consumption is not deeply explored [1]. Last but not least, the size of used datasets is also significantly smaller (the biggest reaching 1000 items and 41.000 users) in comparison to the dataset we are faced with (over 10.000 items and 180.000 users). All these shortcomings motivate the need for further research.

3 Data

The original dataset provided by Nielsen for U.S. households is used as the basis for this thesis. It consists of two panels of respondents – one with TV viewing behaviour and second with Internet sites visits. Each time a respondent turns on a TV precise date, channel, program and watching time is recorded. Each program is classified into one of 141 categories. Similarly, visits to websites, which are also categorized into 95 classes, are stored.

For the purpose of the analysis, we use data from the first week of October 2016. For this period there are around 80.000 respondents in the TV panel, and 180.000 in the Internet panel. Around 20.000 respondents are present in both panels. We aggregate viewing events to obtain the time spend watching a particular program or visiting a particular website during this week. The total number of programs aired during the considered period amounts to about 5000, while the number of sites is much bigger and is equal to around 100.000. The size of the Internet panel poses a serious difficulty, even for our proposed framework. However, since top 5.000 websites are responsible for around 96% time spent online, we decide to use only this subset for further analysis. Such reduction makes computations feasible. At the same time its impact on results is, in our opinion, limited. Number of websites kept in the data is still high enough to assess the effectiveness of the proposed solutons on big datasets. Also from business point of view all most important sites (ones with highest reach) are retained in the data.

In our analysis we are interested in reach of media content - percentage of the people in the population, who have been exposed to a particular TV program or website. We assume a person has been reached by TV program if she watched it at least for 2 minutes in the analyzed time period. For a website the threshold is set to 10 seconds. We note that the data are sparse. Only around 17% of TV programs and around 2% of websites have reach over 1% (ie. are watched by more then 1% of

[1] Usefulness of employing Bayesian networks along with other algorithms to model link between browsing behaviour, and demographic variables has been studied in (Pereira, 2015), with Bayesian networks reaching competitive results. However, small sample size causes the study to be inconclusive.

people in the USA). This poses challenges for the analysis, as, despite the fact that the sample size is big, number of respondents reached by certain TV programs or websites in some subpopulations can be low.

For each respondent in TV panel, 98 socio-demographic variables are available, covering a broad range of information on a respondent herself and her household. For the online panel this number is lower and equals to 34. Several variables are the same or very similar for both panels, however, differences in their definitions (e.g. differently specified education ranges) cause that only age and gender can be exactly matched. Full list of variables can be found in Appendix A. As a result of such data structure, we obtain, in fact, three datasets. TV data - containing socio-demographic variables from TV panel and information on each respondent if she has watched or not listed TV programs. Online dataset - principally similar, but based on the online panel and with information on visited websites. Finally, for an overlapping set of respondents present in both panels, we have socio-demographic variables from both and complete (TV and online) media consumption patterns. Values coding for even very similar variables is not exactly the same in TV and online panel. Because of this, in the joint data, we keep variables from both panels, rather than using only one from a selected dataset. We use all these three datasets in a network's learning process.

Respondents have weights assigned to them, expressing part of the population they are supposed to represent. Weights are adjusted daily and are independent for both panels. Assigning weights on a daily basis implies that we have more than one for each respondent. To obtain a single weight for the week-long period, we take an average of weights for each respondent. For overlapping set of respondents we use TV panel weights. Respondents are also being included (are "in tab") or excluded from the panels on a daily basis. As our analysis covers the span of a week it happens that respondent were not included for the whole time. In the analysis we consider only individuals, which were in tab for at least 4 days. Such restrictions exclude only several respondents, whose appearance in the panel can be attributed to some irregularities, and because of it has little to no impact on the results.

4 Methodology

4.1 Bayesian networks

Basic concepts Bayesian networks belong to the family of probabilistic graphical models, which employs graph representation of probabilistic relationships between variables to represent knowledge about some domain. We speak about a Bayesian network if the graph is directed and acyclic (DAG). Such graphs consist of nodes representing random variables and arcs which show probabilistic dependencies between them. Graphs of Bayesian networks can be understood in two strongly equivalent ways: as a structure providing compact, and factorized representation of a joint distribution as well as compact representation of conditional independence assumptions

about a distribution (Koller and Friedman, 2009).

Connections in a graph can be interpreted as follows: connecting random variable X_1 with another one X_2 by an arc directed from the first one to the latter is equivalent to stating that variable X_1 "influences" X_2. Following that variable X_1 is called a "parent" and variable X_2 becomes its "child". Extending this logic each variable which can be reached from X_i by the directed path in a graph (a set of arcs) is treated as its "descendant" (Pearl, 1998). If variables are not connected we assume they are not directly dependent on each other. They, however, still can influence each other through other variables. More formally it can be formulated as Markov property which states that each variable is independent of its nondescendants given their parents or, equivalently, every random variable depends directly only on its parents (Korb and Nicolas 2004, section 2.2.4).

To fully represent a joint probability distribution (PDF) we need not only set of independencies - a graph, but also its parameters. Using Markov property, we can specify a so-called chain rule (which is based on the Bayes theorem, whence the name of the networks) (Koller and Friedman, 2009).

$$P(\mathbf{X} = \mathbf{x}) = \prod_{i=1}^{n} P(X_i = x_i | \mathbf{Pa}(X_i) = \mathbf{pa}(X_i)) = \prod_{i=1}^{n} p(x_i | \mathbf{pa}(X_i)), \tag{1}$$

where $\mathbf{X} = (X_1, ..., X_n)$, $P(\mathbf{X} = \mathbf{x})$ denotes realization of global probability distribution, $\mathbf{Pa}(X_i)$ stands for vector of parents of variable X_i an $\mathbf{pa}(X_i)$ for its realizations.

Chain rule allows to factorize the joint PDF into a set of smaller local PDFs for each node in the network. Each of them involves only variables associated with a given node and its parents. Fundamentally any representation of probability distribution can be employed as local PDFs. In the thesis, we focus solely on discrete data and employ multinomial distributions represented with contingency tables or conditional probability tables (CPTs). We limit ourselves to use of discrete Bayesian networks as it is necessary for keeping computations feasible, because learning process of continuous networks, especially non-Gaussian ones (which would be needful due to the nature of our data) is far more complex. An example of a simple Bayesian network can be found in the Figure 1.

Although Bayesian networks are often constructed on the basis of expert's knowledge about a domain, learning them from data is also possible. Learning process reflects the dual nature of the BNs and consists of learning network's parameters (CPTs) and a structure (DAG).

Parameter learning Learning parameters provided that a structure is known can be done with maximum-likelihood estimation or within the Bayesian estimation framework. In both cases central to the parameter learning is the fact that thanks to assumed independencies and the chain rule we may decompose the global PDF to local ones, e.g. a likelihood function for a Bayesian network can be decomposed to a product of likelihoods for individual variables. (Koller and Friedman, 2009).

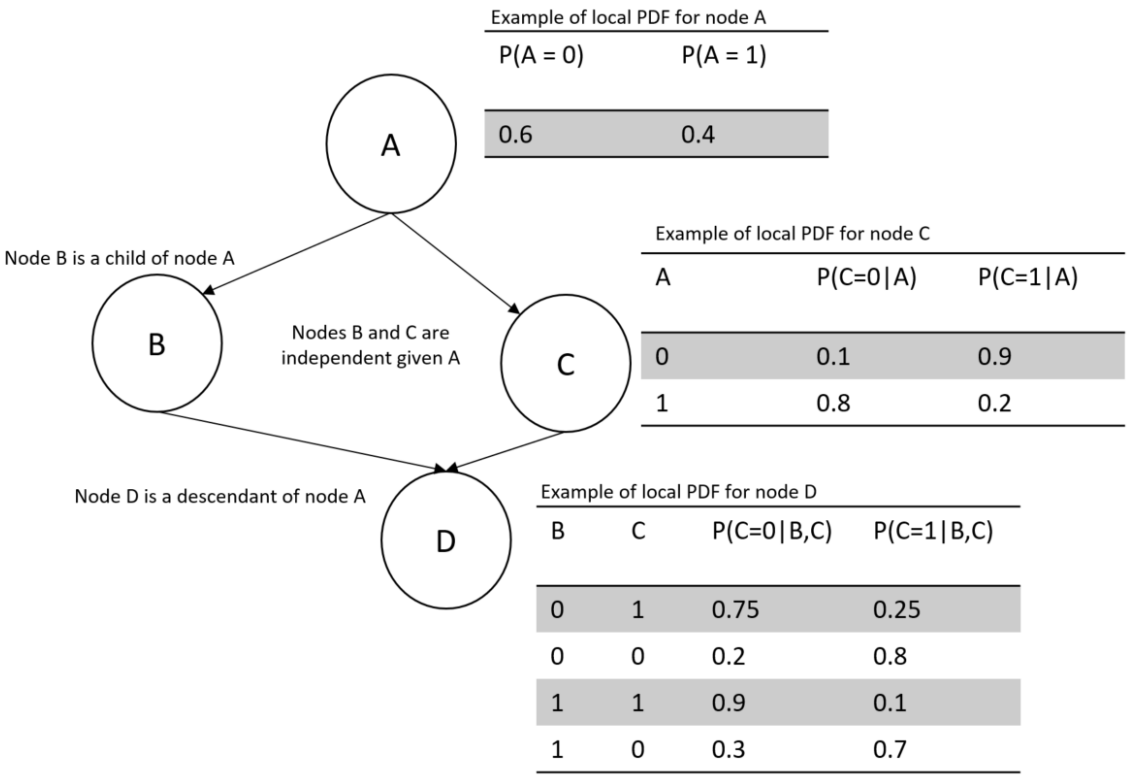

Figure 1: Example of a simple Bayesian network with crucial relationships described

In the thesis, we make use of Bayesian estimation as, this is one of techniques, that allows, thanks to priors, for non-zero probabilities for even zero occurrences of some sequences in the training data and adaptation of CPDs with new information available. In the analysis we assume uniform prior on the parameters, strength of which is typically expressed as an imaginary sample size (iss). Iss symbolizes the hypothetical number of observations, which are equally divided into all possible states of combinations of parents and a node variables for each node. For example, if node A can take values 1 and 2, and has one parent - node B (also binary), there are 4 possible combinations of states. If iss is chosen to be equal to 4, for each state 1 observation is added in the parameter learning process. Details on parameter learning as well as formal definitions can be found in Appendix B.1.

Structure learning Determining a network structure from data poses a much bigger challenge. The problem has been shown to be NP-hard (Chickering, 1996), which resulted in series of heuristic algorithms being developed. They can be roughly classified into the constraint-based and score-based approaches with some hybrid ones developed as well. Constrained-based algorithms attempt to construct a structure from series of independencies between variables established with the use of statistical tests, whereas score-based approaches tackle the task as optimization problem conducting a search through structures' space maximizing network score. In the thesis

we use tabu algorithm with BIC score as representation of score-based approaches and *fast.IAMB* for constrained-based ones. We compare their effectiveness within our proposed framework. For details on specific algorithms refer to Appendix B.2.

Concluding parts on Bayesian networks learning we note that respondents in our data have weights assigned to them. In order to adjust for that we implement a trivial change in formulas used for both parameter learning and scoring. Instead of using counts of observations we use sum of weights for respective respondents.

Inference Obtaining knowledge on variable distributions from a network requires conducting inference procedures. Probably the most common inference task performed on Bayesian networks is conditional probability query (CPQ). It allows us to learn a distribution of variable X_i under some evidence - conditioned on some subset of the rest of the variables in the network taking specific values. For example knowing that variable (node) A, in the network in Figure 1, takes value 1 (evidence), we want to know probability of variable (node) D taking value 0 (event). The problem of inference in a network is in general NP-hard - both exact and approximate (Dagum and Luby, 1993). Despite that, it often can be dealt with efficiently, especially in a simple network, or ones with imposed specific structure. In our case, as we deal with large networks with only some restrictions on their structure, exact inference still proves to be unfeasible, thus we utilise an approximate one. We employ likelihood weighting approach being a variant of importance sampling for Bayesian networks. For details we refer to (Koller and Friedman, 2009).

4.2 Proposed framework

The main aim of this thesis is to obtain a network capturing influence of socio-demographic variables and consumption of particular media content on probability of being exposed TV programs or websites. Simplified example of such network can be found in Figure 2.

Learning such network, considering our data, proves to be not easy. There are two main problems. First of all, we do not have a single dataset with complete socio-demographic variables set and TV/online media consumption for all respondents. Instead, we have two separate panels with only partial overlap in respondents. In order to obtain a network including both TV and Web behaviour, we might use only overlapping respondents. However, it seems unreasonable to exclude information contained in non-overlapping parts of datasets. To overcome this issue we propose to learn the network structure in steps - firstly learn structures of networks containing only TV panel and online panel variables, and then combine the structures performing structure learning on the joint dataset of overlapping respondents, using structures previously learned on single-datasets as starting points. Parameters of such network are learned based on joint datasets and then, calibrated using single TV and online panels, aiming to utilize as much of available information as possible.

Second problem concerns learning the structures itself. Complexity of all men-

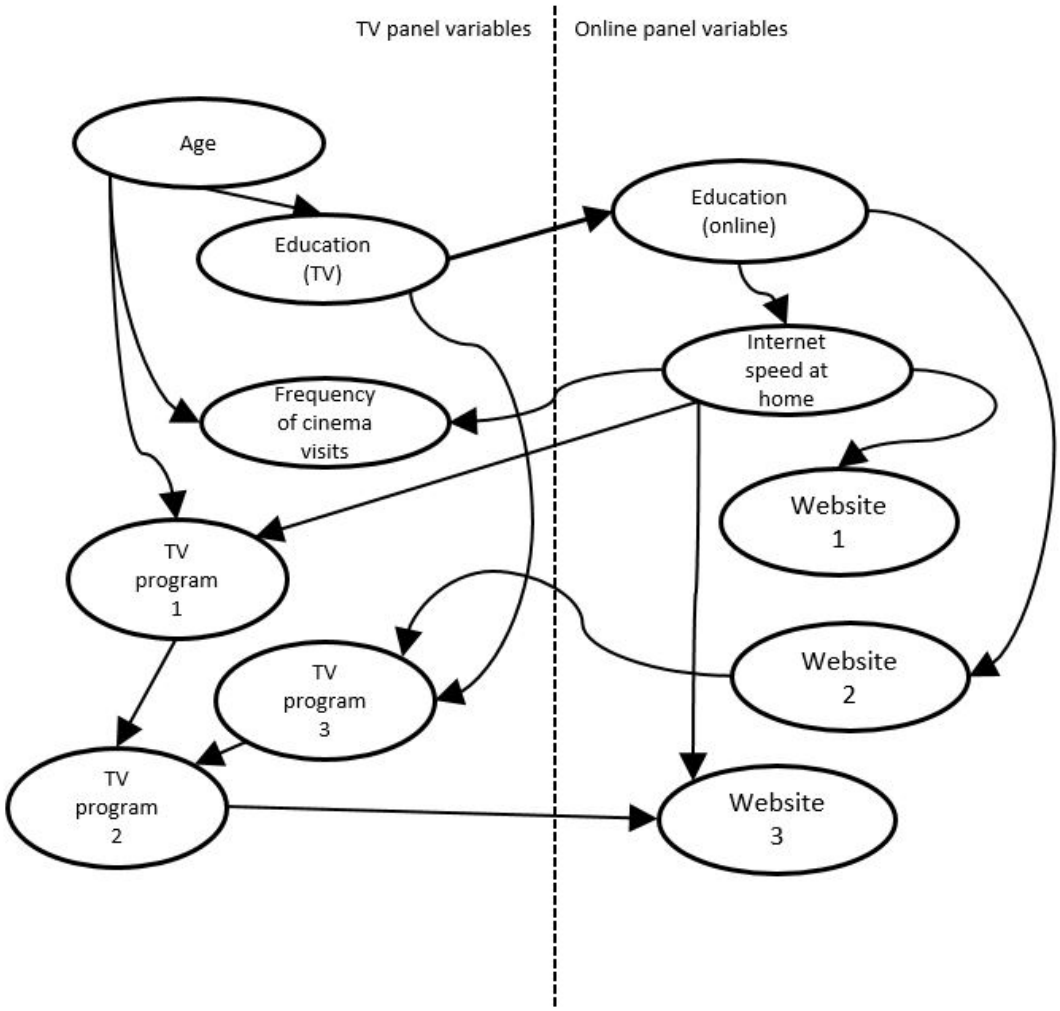

Figure 2: Simplified, artificially created example of desired network.

tioned structure learning algorithms increases drastically with growing number of variables in a network. Hence, simply learning a structure with over 10.000 or even 5.000 (considering only single dataset) nodes is not feasible.

Typically, such problem is solved by introducing additional "hidden" nodes and imposing a specific structure on a network. The idea is based on assumption, that we can summarize information contained in variables, by a smaller number of artificially created "summary" variables. In the next steps only they are used in structure learning, while the original nodes remain connected to respective summary nodes permanently. Example of simple structure, based on one presented above, with an additional "layer" of summary nodes, can be found in Figure 3.

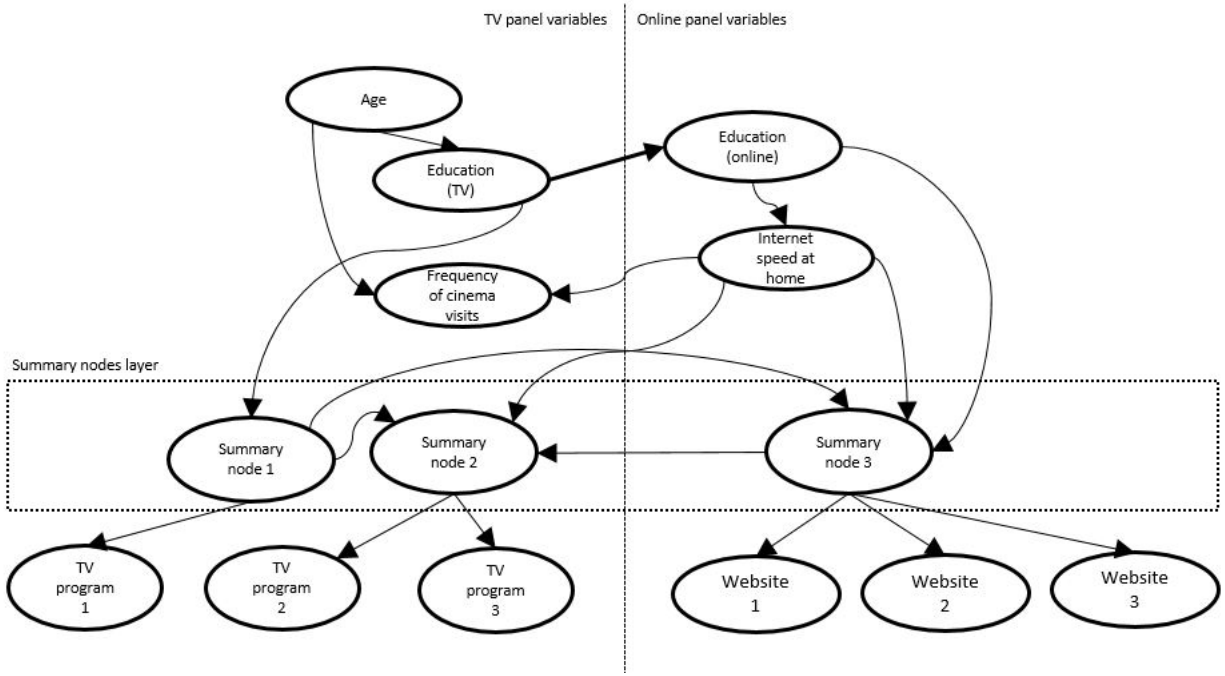

Figure 3: Simplified, artificially created example of desired network with extra layer of summary nodes.

We unify proposed solutions to both described above problems into one framework, steps of which can be seen in Figure 4. With respect to creating summary nodes we explore three alternative approaches - one based on categories assigned to TV programs and websites, one based on matrix factorization, and one based on Hierarchical Bayesian networks approach. More detailed description of each, as well as details on framework's steps can be found in following sections.

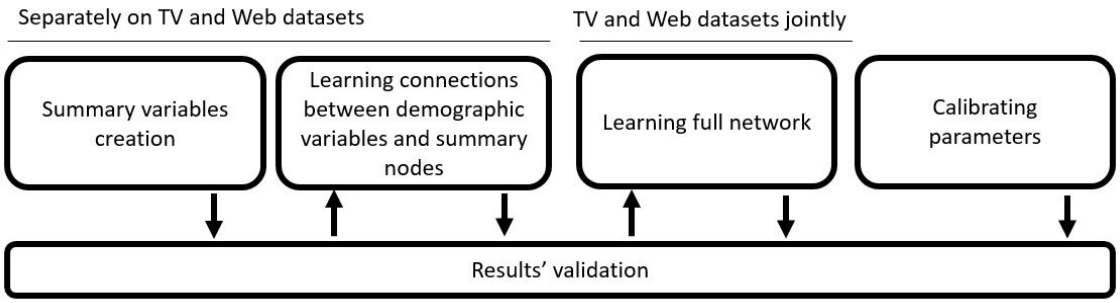

Figure 4: Schematic representation of the proposed algorithm.

In Nielsen's experience adding variables representing total media consumption to the models can be very beneficial for their quality. That is why we propose adding one additional summary node, regardless of used summary variable creation technique. Extra variables, which will be referred to as cumulative media consumption nodes (CC-nodes), are created both for TV and online panels, by summing total time spend, respectively watching TV or browsing Internet by respondent. We enforce links from CC-nodes to all TV programs/websites variables. Connections to other nodes are learned in the same fashion as for other summary nodes. Schematic representation of a joint network we aim to obtain, with framework steps and alternative addition of CC-nodes, can be found in Figure 5.

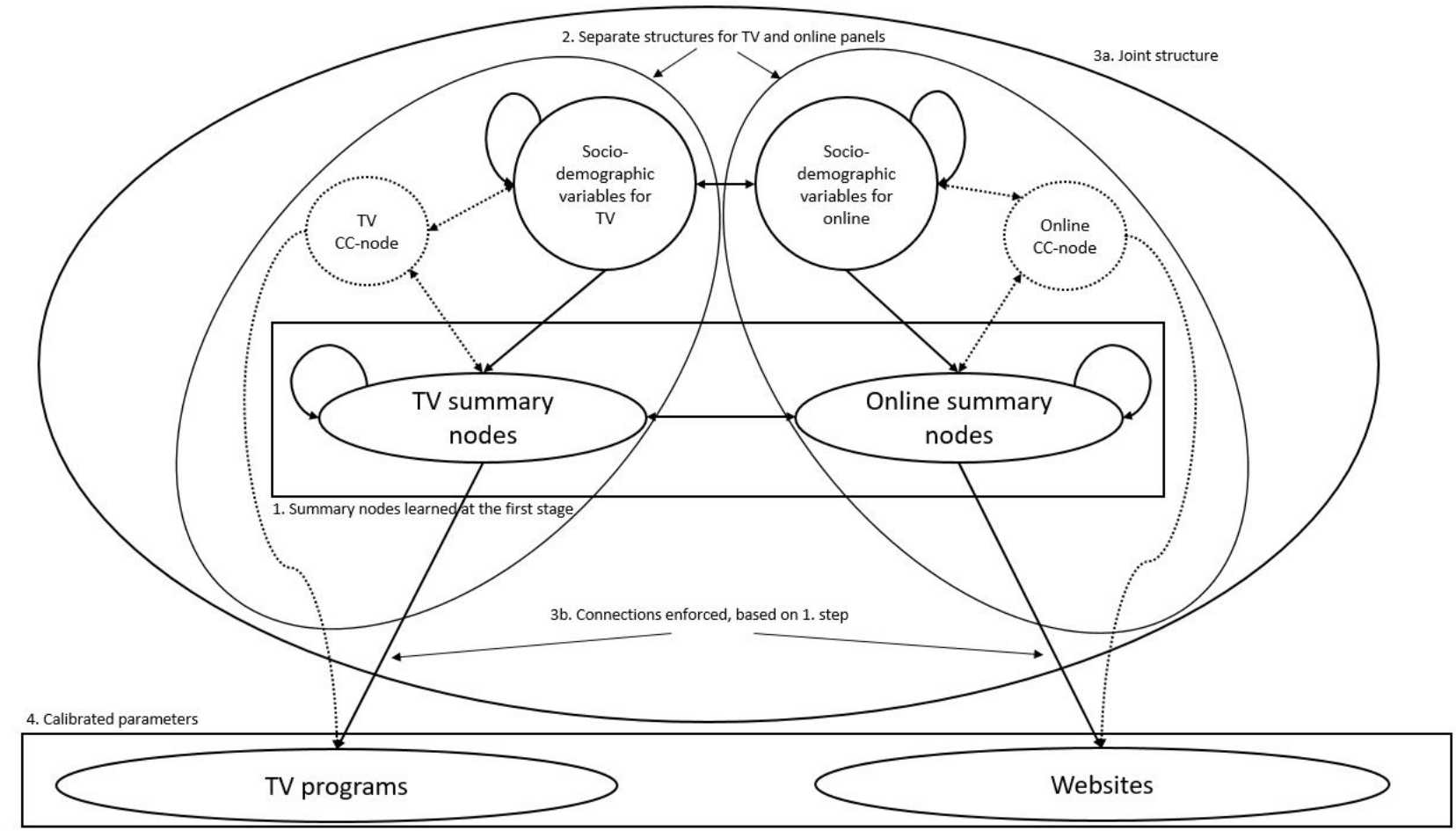

Figure 5: Schematic representation of the final network's structure along with steps of the proposed algorithm.

4.2.1 Summary variables creation

In the first step, we learn summary variables for TV and Web datasets separately. For creation of summary nodes only media content variables (TV programs or websites ones) are used, socio-demographic variables are added later in the process. We explore three alternative approaches of creating summary variables.

Aggregation based on content categories In the Nielsen database, each TV program or website is assigned to a category based on its subjects. We propose to create summary nodes based on this classification. In this setting, each basic media content node is connected to a node representing its category. A graphic representation of such structure can be found in Figure 6. Values of these summary nodes are created as the sum of exposure to all media variables, in a given category and then discretized, on the basis of deciles. Research shows that it is difficult to indicate one best approach for conducting the discretization process (Nojavan et al., 2017) in Bayesian networks. We decide for the decile-based method as it is computationally inexpensive and turns out to perform well in our setting.

A significant advantage of using content based summary nodes is simplicity of their creation and thus little time and computational power necessary to perform it. It also captures a structure, which almost certainly reflects some relationships in the data. It does not force us to infer it by ourselves. Last but not least, the structure created in such way is relatively stable over time. It allows for easier extension of the proposed framework by a time component. Naturally the simplicity of this method has its flaws. It assumes that reach of a particular TV program or a website is independent from other variables if a reach of its content category is known. It is unlikely to be completely true.

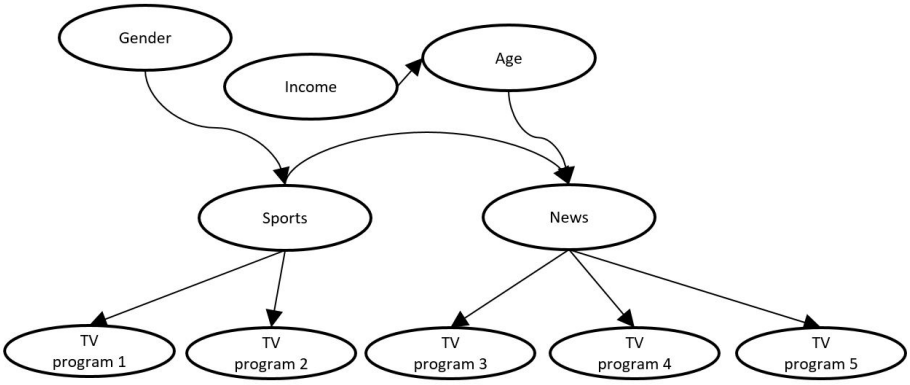

Figure 6: Example of network with categories as summary nodes.

Matrix factorization Output of various dimension reduction techniques has been previously used in Bayesian networks. Most often employed technique remains Pric-

ipal Component Analysis (PCA). It has been used mainly for image processing applications (Abbasnejad and Teney, 2015; Park and Aggarwal, 2004), but it was also employed in other contexts, often with good results (Sturlaugson and Sheppard, 2013). Nevertheless, it is vital to note, that in our context the PCAs variance reduction ratio may be limited, due to sparsity and size of the data. It will leave too many nodes, even though the assumption of Independence and thus no connections between principal components significantly restrains the search space in a structure learning task.

To address these issues we choose to use matrix factorization instead of PCA. It is a technique developed mainly for recommender systems. Its aim is to decompose ratings matrix $R_{m\times n}$, which contains, in our case, time spend by an n-th user being exposed to m-the TV program/website. As a result of matrix factorization such matrix is represented as a product of two lower dimensional ones - $P_{k\times m}$ and $Q_{k\times n}$. First one may be interpreted as a TV programs/websites hidden "qualities" matrix, second as a grid of consumer "attitudes" towards them. A number of latent features k is typically subject to a parameter tunning. Matrix factorization has proved to perform very well in numerous studies (Bell and Koren, 2007; Dror et al., 2012) and it can be a very computationally efficient technique. We believe that being designed for data closely resembling ours, it is the best suited for the task. It is useful also in the next proposed approach to dimensionality reduction. For details on the method itself, refer to Appendix C.1.

We use extracted latent features in the Q matrix as summary variables. We discretized their values. In a network structure we impose direct links from all summary nodes to all TV programs/websites variables. An example of such structure can be found in Figure 7.

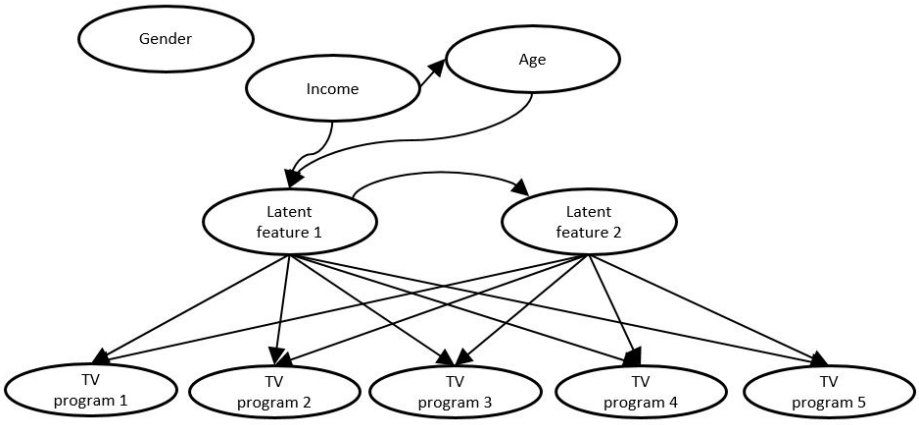

Figure 7: Example of network with 2 latent features identified by matrix factorization as summary nodes.

Matrix factorization at least partially overcomes shortcomings of the previous approach. It allows for more complex relationships between TV programs or websites and attempts to capture deeper patterns in the data. Nevertheless, we have to acknowledge the downsides of this method. Dense structure of network connections

limits number of latent nodes and their cardinality, as size of a conditional probability table (CPT) of a node in a network representing its local probability distribution (PDF) increases rapidly with number of parents and their cardinality. Assuming all parents and a variable itself can take 2 values, CPT for variable with k parents will have 2^{k+1} cells.

Hierarchical latent class Bayesian Networks Hierarchical latent class models have been developed extensively in bioinformatics to deal with highly dimensional genetic data. Examples of proposed algorithms are Daly et al. (2001); Kimmel and Shamir (2005); Greenspan and Geiger (2004); Gyftodimos and Flach (2004); Mourad et al. (2011). The idea is based on creating layers of latent nodes summarizing 2 or more nodes from the lower layer. Typically observed variables are treated as first layer. They are grouped by a criterion reflecting their similarity; then each group is being connected to one latent node. Learning observations values for a new latent node can be typically perceived as learning class-memberships in multinational mixture models, for which the Expectation-Maximization algorithm is employed. For details refer to Appendix C.3. Learned values are next treated as known, in the next step procedure is repeated for a new layer. At some point, the procedure is stopped. Then the nodes from the last layer are used for further analysis, e.g. to train connections between them (normally, creation of connections between nodes in a given layer is not permitted).

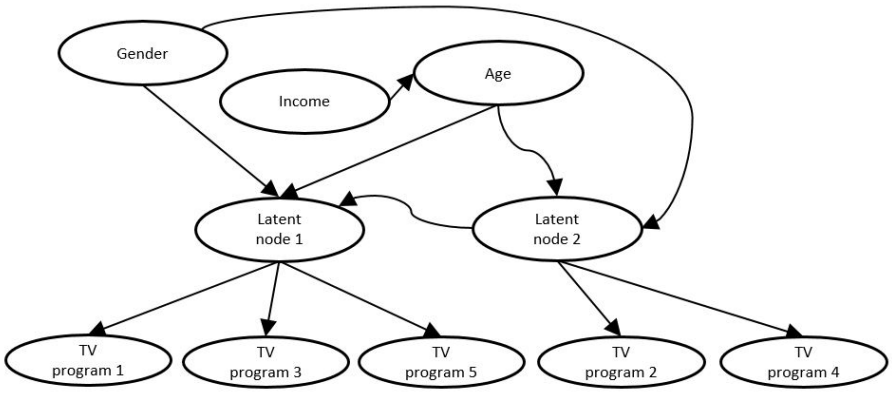

Figure 8: Example of hierarchical latent class network.

Algorithms successfully used in genomics cannot be directly employed in our case due to the differences in data structure. Researchers in this field typically face so-called wide data - a low number of observations and much higher number of variables. In our case, despite a high number of variables, the number of respondents surpasses it distinctly. This difference causes two main problems.

Firstly, with a low number of observations, separate EM-algorithms require relatively short time which allows running many of them and as a result learning substan-

tial number of latent nodes in a short time. Our situation is more challenging. We have high number of respondents and thus training one hidden variable can be very time-consuming. To deal with it, we limit ourselves to only one layer of latent variables. Each latent variable summarizes directly a group of TV programs or websites variables. Finding such groups is the second problem, which needs to be solved.

In mentioned algorithms, grouping of variables can be conducted easily by treating observations in the data as dimensions and finding clusters of variables in such space. In our case, such space would be higher-dimensional than number of TV programs/websites variables which can be clustered. We address this problem by first reducing dimensionality by matrix factorization, and then grouping variables by means of hierarchical clustering. Relevant theory overview on clustering can be found in Appendix C.2.

An example of such structure, with only one layer of 2 hidden variables can be seen in Figure 8.

Successful applications in genomics, postulated ability to capture complex relationships, and the fact of being embedded in the Bayesian networks framework are significant advantages of this approach. The proposed changes are necessary. Limited number of layers may harm the accuracy of the network, but it does not defy the underlying idea of the algorithm. Altered clustering method should not affect networks performance, as research on GWAS algorithm (Mourad et al., 2011) suggests, that as long as clusters identified resemble patterns in data, algorithms performance is not impaired (Phan et al., 2015). The biggest downside of this approach is its computational cost, which even with proposed restrictions can relatively be high.

4.2.2 Learning networks with demographic and summary variables

Obtained summary variables are used together with demographic variables, to train networks structures separately for TV and online panels.

We compare performance of tabu and fast.IAMB algorithms introduced in Section 4.1 as well as structures with and without CC-nodes. For practical implementation we employ the package *bnlearn* (Scutari, 2010) from R environment (R Core Team, 2016) along with some custom modifications. We use the same methods for both TV and web datasets.

For both trained structures we learn parameters as well to make their evaluation possible (more on evaluation techniques can be find in Section 4.2.5). In both structure and parameter learning, the formulas are adjusted to accommodate for the fact that each respondent has a weight assigned to her. Unless specified otherwise, for parameter learning we use imaginary sample size equal to 1. This assumption extends to all cases of parameter learning in the proposed approach.

4.2.3 Learning joint network

Having obtained networks for TV and Web variables we proceed with learning of a joint network. For this purpose we utilize the dataset of respondents present simul-

taneously in both panels, for whom a complete set of variables is known. The same learning algorithms as for single datasets structures are used. To take an advantage of earlier obtained knowledge we use networks for TV and online data as joint staring point, making sure that similar variables in both datasets are connected, however without determining direction of such arcs. We include CC-nodes only if they improve accuracy of a TV or online network. We then let algorithm learn new arcs or adjust existing ones.

In that way we obtain a structure containing socio-demographic variables, summary nodes and TV programs/website nodes from both panels. Then we learn parameters for such structure and evaluate networks quality using overlapping respondents set for both of these tasks.

4.2.4 Calibrating joint network's parameters

Although the overlapping dataset provides us with useful information on relationships between variables of both sets, it is less accurate with respect to distributions over variables included in each of them separately. As we are interested in obtaining as reliable estimates as possible, particularly concerning the reach of TV programs and websites, we propose to calibrate the parameters of the joint network using datsets with all respondents from TV and online panel respectively.

For the calibration, we make use of imposed independence structure of the network. We note that each TV program/website node is dependent only on the summary node(s). It means that in order to estimate parameter values for it we need observations of the variable and summary node(s) only. These, in each case, are available in separate datasets for both TV and online panel. Thus, we replace parameter values of media item's nodes, learned based on overlapping dataset, by these obtained from TV/online datasets.

We extend the above described calibration procedure to all nodes which parents set belongs exclusively to one separate dataset. It allows us to calibrate values not only for TV programs/websites nodes but also for multiple other ones, which in learning procedure had not been linked to variables from other panel.

Final network structure, along with marked algorithm steps can be found in Figure 5.

4.2.5 Evaluation

The evaluation of an obtained network is crucial at every step of the process. First of all, we want to able to assess if obtained network really captures the underlying distribution in general. Secondly, it is particularly important, from a business perspective, how accurately it reflects marginal distributions, which may be interesting from the point of view of Nielsen's clients.

For the sake of evaluation in both cases, we split all three used datasets (TV, online and one containing variables from both, with only subset of respondents) into training and testing set in a proportion 80/20.

Firstly we compare trained network structure for training and test datasets by means of its likelihood values per observation. The absolute difference between these values should be relatively small. We also perform goodness of fit test as described in (Koller and Friedman, 2009). We draw samples from the estimated network and calculate likelihoods of the structure using them for parameter learning. In the following way, we obtain the simulated distribution of likelihood values for the structure. The calculated likelihood on the training data should be within the 95% of probability mass of the simulated distribution. We perform such testing at each step of the network learning. We note that likelihood value for data given structure is dependent on the number of respondents. Thus to get a score comparable to the one for training set, for drawn data we would need to get a sets of samples of the size of the sum of all respondents weights - around 300 million. This is unfeasible because of potential computational cost. To deal with this problem, we draw a smaller sample and compare likelihood value per respondent instead.

To assess prediction performance of the final network, we propose our own approach. It is designed to resemble closely the actual situations in which the network can be used by Nielsen's clients. First, we sample set of TV programs/websites for which potential client can be interested in knowing the probability of reaching some group of people. Then for each of them, we sample a set of conditions to specify this subpopulation (e.g. gender: "male" and age: "between 18-19"). Then we estimate from the network the expected probability of respondent being reached by a TV program/website. We make here use of the fact, that not all values of variables in Bayesian network need to be specified to obtain predictions. It allows to take into consideration uncertainty in values of not specified variables and compere networks with slightly different set of variables. We compare probability inferred from a network to one calculated from training and test data. In both later cases we calculate reach by selecting only respondents belonging to subpopulation of interest, and then computing percentage of them, which were exposed to particular media content. Ideally, they all predictions (from a network and both datasets) should be the same. We note, however, that some randomness is impossible to avoid. At the same time, it is hard to assess how large variation should be acceptable. Thus, to get a benchmark, we compare differences between network and data results to differences between training and test set and treat them as such benchmark and baseline for randomness. Differences between reach inferred from a network and one calculated from data can be seen as prediction errors. The comparison is conducted based on root mean square error (RMSE), median average error (MAE) and Jensen-Shannon divergence (Lin, 1991) with natural logarithm as base, which can be expressed as:

$$JS(P,Q) = H(\frac{P+Q}{2}) - \frac{H(P)+H(Q)}{2}, \tag{2}$$

where $P(x)$ and $Q(x)$ are probability distributions and $H(P) = -\sum_x P(x) log P(x)$ is a Shannon entropy.

We also closely investigate prediction errors.

5 Application

We proceed with applying the proposed framework to real-life data. Firstly, we select a reduced dataset of top 100 watched TV programs and top 100 visited websites. Such number of variables allows us to establish the best approaches and practices while keeping computational time reasonable. We start with learning the networks based only on the observed variables themselves, without introducing any summary nodes. It helps to find general best guidelines in learning networks on our data and results may serve as reasonable baseline for approaches with summary variables. We continue by applying the first three steps of the framework on the reduced datasets in order to compare proposed summary nodes creation techniques and assess networks general quality. Using best solutions identified in earlier steps we test quality of joint network and effectiveness of its calibration. Finally, using best approach and earlier findings, we model full dataset and evaluate the results.

5.1 Baseline results for reduced dataset

We start with learning separate networks on "reduced" TV and online datasets consisting of 100 top consumed items in each as well as demographic variables. Then using the overlapping set of respondents and earlier obtained nets as a starting point we learn the network for both media, to which we refer as joint network.

For learning, we use and compare score-based and constraint-based algorithms mentioned in Section 4.1. Number of arcs in each network divided into several types can be seen in Table 1.

	Score-based			**Constraint-based**		
	TV	Online	Joint	TV	Online	Joint
Total	944	680	1375	30	28	63
Between socio-dem. variables	234	128	435	25	21	47
Between media content	402	261	495	5	7	14
From socio-dem. varbiables to media content	308	291	445	0	0	0
Between datasets	-	-	100	-	-	2

Table 1: Number of arcs by type in specified networks learned with use of score-based and constraint-based algorithms.

What becomes immediately clear is, that the choice of algorithm heavily impacts the resulting network. Score-based approach produces highly connected graphs with a lot of arcs within socio-demographic or content variables as well as between these sets. For a network on both datasets this method effectively "discovers" connections between relevant demographic variables in TV and online datasets, as, e.g. education ranges. Most of the learned arcs in the networks are reasonable from the common

sense point of view, like, e.g. dependence of income on age, working hours on occupation, visits to certain internet sites on internet speed or viewing of certain TV programs on access to paid channels. Such high number of connections also results in less obvious, and sometimes suspicious dependencies as, e.g. certain TV program viewing on having a dog. Nonetheless, these are the minority of connections learned. We note that viewing of TV programs is in vast majority influenced by variables related to age, place of living, income, the presence of children or access to cable or pay channels. Variables related to site visits in network on online data show similar dependencies with most often appearing visits dependencies on income or other socio-economic status variables. Variables that in the network for TV panel depended on TV access in this network depend on Internet speed. Multiple relationships directly between media content variables are shown as well. For network learned on both datasets also some connections between TV programs and websites are discovered.

Networks learned with use of constraint based algorithm provide a completely different picture. Within demographic variables only the most clearly existing dependencies are learned. Graph suggests as well that socio-demographic variables do not influence media viewing at all. What is alarming in case of network learned on both datasets multiple relationships, which we expect should exist are not discovered.

To some extend, sparsity of networks learned with constraint-based algorithm should be anticipated. Generally, score-based algorithms tend to produce less sparse networks. They assess a set of independencies at once rather them considering them separately as statistical tests conducted in constraint-based approaches do (Koller and Friedman, 2009). Constraint-based approaches are sensitive to failures of single independence tests - connections in a network are added based on sequentially conducted tests. Each test in the sequence is conditioned on earlier added connections, thus wrongly adding or omitting one can result in a reorganization of a whole network structure. This stops us from lowering significance level in order to obtain more connected networks. Such a notable difference between results between the two used algorithms is, nevertheless, surprising. On the one hand, it might suggest, that there is indeed little influence of socio-demographic variables on media consumption and score-based approach highly overfits the network. On the other hand, it is possible that constraint-based method simply fails to produce an accurate model for our data. We investigate it by looking at data and model likelihoods, goodness of a fit test and predictive results, as describe in Section 4.2.5.

Log-likelihoods values per person for both training and test datasets and boundaries of goodness of fit test (GoF) can be found in Table 2. Goodness of fit values are based on 100 samples, each equal to the number of respondents in the training set.

Firstly, we note that log-likelihoods for networks learned with the constraint-based approach are clearly lower. This is to be expected, as adding arcs to a network increases its likelihood. Interestingly, in all cases, log-likelihood for test data is higher than for training one. It is a good sign. Supposedly, test data, due to its smaller size has less observations in it which differ from standard patterns. Smaller log-likelihood for network on test data suggest that these true patterns are captured well. We note

	Score-based			Constraint-based		
	TV	Online	Joint	TV	Online	Joint
Training	-56.3	-28.0	-63.0	-106.2	-52.3	-136.9
Test	-49.5	-23.1	-52.0	-106.1	-51.7	-136.3
GoF - lower bound	-54.8	-27.8	-60.2	-108.2	-52.4	-139.1
GoF - upper bound	-55.0	-27.9	-60.5	-108.3	-52.4	-139.4

Table 2: Log-likelihood values for networks with parameters learned on specified datasets. Bootstrapped boundaries of 95% goodness of fit interval.

that the difference between log-likelihoods is bigger for networks learned using score-based algorithm. Ideally we would like both log-likelihood values to be contained within the GoF test boundaries. For both approaches, in all cases, training log-likelihood values are outside of them. However, while for score-based approach these GoF test boundaries are higher than training data log-likelihood, for constraint-based they are lower.

We also compare accuracy performance of the networks. In each case, we perform 5,000 conditional probability queries as described in Section 4.2.5. In each query, we evaluate the probability of a person being reached by chosen at random TV program/website, assuming that she has certain characteristics, e.g. has a dog, or has watched some particular TV show. Always two variables are randomly selected for characteristics of choice. As an impact of socio-demographic variables on reach is more interesting for us, we assign 70% probability of variable being chosen from socio-demographic variables set and respectively 30% for the choice of media content variable. Their values are chosen randomly as well, with equal probability assign to each possible state. We use only two restricted variables in each query as choosing more results often in too few or no respondents with desired characteristics. Such situation causes reached estimates based on data to be too volatile and untrustworthy.

Each query is performed on the network, and training and test data, to get a reach probability resulting from them. Resulting probabilities are compared, with training or test data serving as "true" values and network's results or test data as estimates. Prediction errors comparison based on such paradigm, by means of the root mean square error (RMSE), median absolute error (MAE), and average Jensen-Shannon divergence (JS), can be found in Table 3. The comparison is, as earlier, preformed for networks leaned on TV and web data, as well as the joint network learned using the overlapping set of respondents.

In all cases, for TV and joint datasets, networks learned using score-based clearly outperform constraint-based ones. Importantly, it is a case for comparison with both training and test dataset. It might suggest that the latter algorithm underfits. For online data, however, MAE and JS suggest that constraint-based algorithm performs slightly better. Closer examination of the results suggests that it is only illusory.

In the Figure 9 we can see scatter plot of estimated reach values for all performed

		Score-based			Constraint-based		
		TV	Online	Joint	TV	Online	Joint
RMSE	BN/Training	0.0289	0.0556	0.0303	0.0589	0.0860	0.0493
	BN/Test	0.0390	0.0682	0.0436	0.0645	0.0904	0.0573
	Test/Training	0.0311	0.0603	0.0423	0.0311	0.0603	0.0423
MAE	BN/Training	0.0182	0.0330	0.0160	0.0316	0.0328	0.0206
	BN/Test	0.0229	0.0392	0.0224	0.0345	0.0341	0.0258
	Test/Training	0.0147	0.0224	0.0184	0.0147	0.0224	0.0184
JS	BN/Training	0.0015	0.0060	0.0018	0.0039	0.0071	0.0031
	BN/Test	0.0025	0.0083	0.0036	0.0049	0.0080	0.0046
	Test/Training	0.0014	0.0044	0.0030	0.0014	0.0044	0.0030

Table 3: Reach predictions comparison based on differences between reach from specified datasets and one inferred from a Bayesian network. Networks learned on the reduced dataset.

conditional probability queries. values inferred from training dataset are marked on the x-axis. On y-axis we map values resulting from test data or from the network. In ideal situation all dots will lay on the $y = x$ line. It is, however, not the case and we observe some deviations. We can clearly see that the constraint-based network indeed might perform better if desirable reach value is small and similar to one for whole population (without any restrictions). Results from score-based algorithm show more variance in this region. If, however, the impact of restricted variables becomes apparent and reach differs from one for the population, more sparse network starts to become increasingly inadequate to the task.

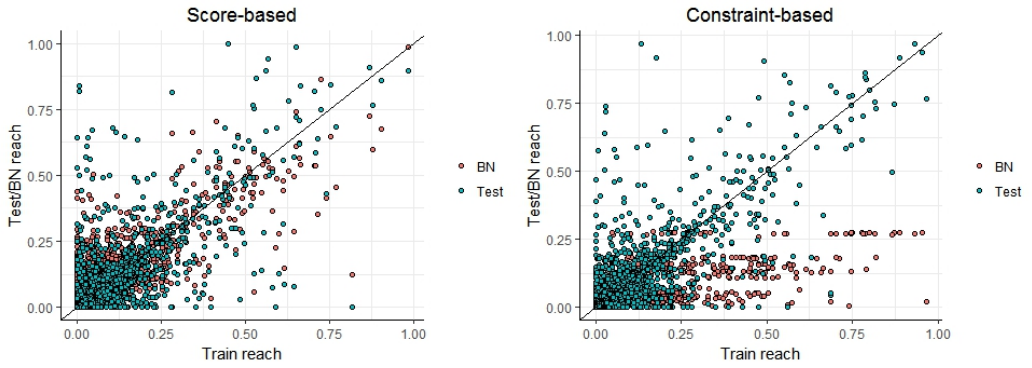

Figure 9: Scatter plot of reach inferred from BN and test data plotted against one from training dataset.

Based on results presented above we may conclude that score-based algorithm is better suited to learn networks in our case, even assuming that it might slightly overfit the models. We suspect the reason for this behavior can be found in the relationships structure in our domain. From networks learned by score based algorithm we can

see that several nodes are responsible for most of the connections, e.g. parents and children set of age variable in TV network consists of 122 nodes. Such situation is impossible to reconstruct in constraint-based algorithm, as number of observations per cell in conditional probability tables drops visibly with each variable added to its parents and children set and tests quickly becomes unreliable, preventing addition of new nodes. Pairing it with more strict requirements for adding an arc leads in our case to learning very sparse networks. It could be prevented by increasing the number of observations in the data, which is not feasible. Therefore, both empirical results and characteristics of the domain points towards using score-based algorithm.

Having decided to use the score-based algorithm, we assess general quality of networks learned using it. As a benchmark, we use differences between probabilities inferred from training and test datasets. With respect to reflecting training datasets networks perform relatively well, especially for the joint data. This is probably due to the fact, that using a reduced number of overlapping respondents means some extreme categories (e.g. people with high number of children or very specific, rarely-occurring education status), which are prone to be linked to untypical behaviour, are not being included in testing due to insufficient number of observations for them. Results obtained by comparison of predictions to training data are visibly worse. The difference is, however, not drastic and may be deemed acceptable for practical purposes. Nevertheless, it might indicate some overfitting of the model.

One more concern may be raised by failed goodness of fit tests. With respect to it, we observe that log-likelihood of a network (based on a data sampled from it depends directly on imaginary sample size (iss). Increasing it, we may lower a network's log-likelihood. As log-likelihood of a network calculated using data does not depend on iss, and network's log-likelihood calculated using samples from it is typically higher, by increasing iss we may "force" network's log-likelihood to be closer to data's one. A dependency of network's sampled log-likelihood on iss is shown in Figure 10.

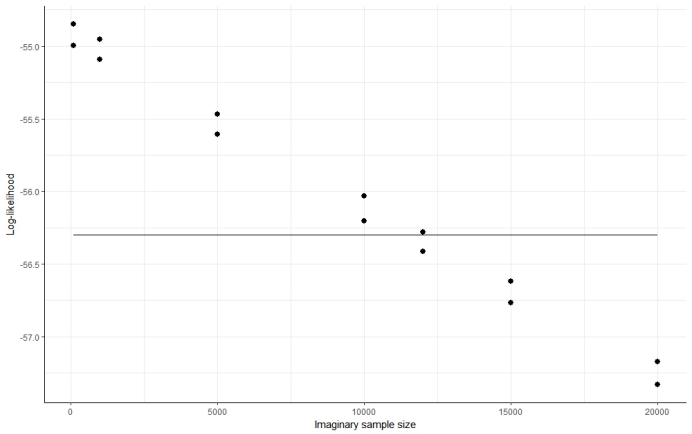

Figure 10: Dependence on iss of lower and upper bounds of 95% interval for log-likelihood of data sampled from a TV network with training data. Log-likelihood marked as horizontal line.

We can see that interval values drop linearly. In order to contain training data log-likelihood in their boundaries iss needs to be set to around 11,500. It may suggest that such network would actually be a better fit. To some extend it might be logical as higher iss results in higher probabilities of rare states of nodes, which due to sparsity of the data can be underestimated by us. Comparison of relevant metrics of predictive accuracy for TV networks, learned with score-based algorithm can be found in Table 4.

		RMSE	MAE	JS
BN (iss = 1)	- Training	0.0295	0.0185	0.0014
BN (iss = 11500)	- Training	0.0313	0.0201	0.0016
BN (iss = 1)	- Test	0.0381	0.0230	0.0024
BN (iss = 11500)	- Test	0.0391	0.0238	0.0026
Test	- Training	0.0315	0.0153	0.0015

Table 4: Reach predictions comparison of Bayesian networks learned with different imaginary sample sizes.

We observe that increasing iss worsens predictive results in all considered cases. For other networks similar relationship has been found. Thus, we decide to use iss=1 in all considered cases, despite goodness of fit test results, which may suggest otherwise.

5.2 Application to reduced dataset

We continue analysis by testing suggested methods of creating summary nodes and alternative network structures with cumulative media consumption nodes (CC-nodes). We apply it to the reduced dataset as describe above. For each summary nodes creation technique we train networks for TV and online data both with and without CC-nodes. Than we train networks for both jointly using best combinations of structures (eg. with CC-node for TV and without for online). We compare results for all obtained networks to better understand the overall ability of our models to capture patterns in the data. Finally using these findings we construct a joint network, calibrate its parameters and asses its final quality and effectiveness of calibration.

5.2.1 Category summary nodes

We begin with using category variables as summary nodes. Top 100 TV programs belong to 24 different categories, top 100 websites to 36 categories. In case of TV data, 5 categories contain only one program. For online data 17 categories account for only one website. A summary node for each respondent is created by adding exposition time to media content in every respective category and discretising by using deciles.

The learned networks are relatively dense, but with reasonable arcs, strongly resembling the ones learned earlier, The only difference is replacement of content

variables nodes by summary ones. This similarity extends to the full network for joint variables sets. Network on joint datasets is learned without using the CC-node for TV and online, as this configuration gave the best empirical results.

The goodness of fit tests results can be found in Table 5. We observe that in all cases networks log-likelihood calculated using simulated data is higher than one obtained using training dataset. Networks log-likelihoods on training and test data differ visibly. Networks with and without cumulative media consumption nodes behave similarly.

	TV		**Online**		**Joint**
	w/o CC	with CC	w/o CC	with CC	
Training	-55.26	-57.05	-31.23	-28.73	-78.92
Test	-48.90	-50.23	-26.26	-23.65	-67.40
GoF - lower bound	-53.85	-55.49	-30.54	-27.72	-75.78
GoF - upper bound	-53.98	-55.63	-30.69	-27.82	-76.42

Table 5: Log-likelihood values for networks with parameters learned on specified datasets. Bootstrapped boundaries of 95% goodness of fit interval. Networks learned on reduced dataset with category-based summary nodes.

Results of testing networks accuracies (Table 6) offer a clearer overview of models quality. Firstly, we observe that both in case of TV and online nets addition of CC-node worsens network's predictive quality. Secondly, we note that overall results are relatively good. It is a bit surprising considering simplicity of the approach. Probably, at least partially, it can be attributed to high ratio of summary nodes to summarized ones, and the fact that many of them represent actually only one TV program/website.

		TV		**Online**		**Joint**
		w/o CC	with CC	w/o CC	with CC	
RMSE	BN/Training	0.0365	0.0427	0.0499	0.0525	0.0342
	BN/Test	0.0447	0.0501	0.0629	0.0666	0.0463
	Test/Training	0.0311		0.0603		0.0423
MAE	BN/Training	0.0205	0.0259	0.0202	0.0221	0.0152
	BN/Test	0.0245	0.0295	0.0264	0.0287	0.0218
	Test/Training	0.0147		0.0224		0.0184
JS	BN/Training	0.0018	0.0025	0.0033	0.0038	0.0018
	BN/Test	0.0027	0.0034	0.0053	0.0060	0.0035
	Test/Training	0.0014		0.0044		0.0030

Table 6: Reach predictions comparison based on differences between reach from specified datasets and the one inferred from a Bayesian network. Networks learned on reduced dataset with category-based summary nodes.

5.2.2 Matrix factorization summary nodes

We continue by using matrix factorization output as summary nodes. Hyperparameters for matrix factorization are for both TV and online datasets determined with use of 5-fold cross-validation and extensive grid search over hyperparameters space with RMSE as loss function. A number of latent variables is determined to be equal to 15 for TV and 10 for the online dataset. We note that matrix factorization by itself, both in case of TV and online data, gives relatively good predictions, supporting the opinion on usefulness of this technique for media consumption modelling.

The output is discretized based on the median. Choosing other methods (mean or cut-off set at media consumption time equal or greater than 0) yielded similar final results. Unfortunately using more than two classes is not feasible. As we connect each summary variable to each node, size of conditional probability tables representing local probability distributions for content variables increases exponentially with number of classes in summary nodes. It causes a conditional probability table to quickly exceed reasonable size (for TV with 15 latent variables, and 2 classes in each it is equal to 2^{15}).

Resulting networks are similar to earlier mentioned ones, also when it comes to relatively high number of connections between summary nodes, which might complicate inference as each of them is then connected to all content nodes. Joint network for both datasets is learned without CC-nodes.

Results of goodness of fit test, which are presented in Table 7 seem to be promising. Difference between training and test log-likelihoods continues to be relatively big, however, TV networks without CC-nodes, online with CC-nodes, and the one for joint datasets variables pass goodness of fit test.

	TV		**Online**		**Joint**
	w/o CC	with CC	w/o CC	with CC	
Training	-60.25	-55.72	-31.07	-30.62	-66.35
Test	-51.60	-46.53	-26.68	-25.77	-53.56
GoF - lower bound	-60.13	-55.07	-30.31	-30.53	-65.62
GoF - upper bound	-60.40	-55.40	-30.41	-30.72	-66.41

Table 7: Log-likelihood values for networks with parameters learned on specified datasets. Bootstrapped boundaries of 95% goodness of fit interval. Networks learned on reduced dataset with matrix factorization-based summary nodes.

Results of prediction performance evaluation can be see in Table 8. Unfortunately, they clearly indicate that networks are badly fitted. All indicators, for all networks are much higher than thus for test/training, which we use for benchmark. Only the network for online variables is slightly better. Adding CC-nodes worsens accuracy even more both for TV and online data.

There can be multiple reasons for such results. Firstly, creating summary nodes in proposed way might not reflect any actual structure of the data. Secondly, matrix

factorization might not be a preferred summary nodes creation technique. Good results of prediction of matrix factorization itself and the fact that attempts to use the Principal Component Analysis yielded even worse results, suggests that, finding better technique may be difficult. General density of the network, resulting from a high number of connection between summary nodes, and connecting all summary nodes to each TV program/website node may be to blame as well. Nonetheless, attempts to restrict connections between summary nodes did not result in any notable improvements. Last, but not least, discretization of matrix factorization output into only two categories may be too drastic, but it is necessary.

		TV		**Online**		**Joint**
		w/o CC	with CC	w/o CC	with CC	
RMSE	BN/Training	0.1154	0.1907	0.0711	0.1529	0.1112
	BN/Test	0.1188	0.1927	0.0796	0.1584	0.1149
	Test/Training	0.0311		0.0603		0.0423
MAE	BN/Training	0.0873	0.1601	0.0312	0.0885	0.0743
	BN/Test	0.0893	0.1613	0.0363	0.0928	0.0771
	Test/Training	0.0147		0.0224		0.0184
JS	BN/Training	0.0137	0.0320	0.0065	0.0248	0.0147
	BN/Test	0.0149	0.0331	0.0085	0.0275	0.0168
	Test/Training	0.0014		0.0044		0.0030

Table 8: Reach predictions comparison based on differences between reach from specified datasets and the one inferred from a Bayesian network. Networks learned on reduced dataset with matrix-factorization based summary nodes.

The poor performance of networks constructed using matrix factorization summary nodes is surprising, especially taking into consideration relatively good results of matrix factorization itself. It can be probably attributed to to the big sizes of conditional probability tables for media content nodes. They translate to high number of parameters in the network and low number of observations used to learn each of them. Despite the significant size, our dataset turned out to be too small. Use of continuous Bayesian networks could improve models accuracy, as number of parameters in such networks would be lower. However, this is outside the scope of this thesis.

5.2.3 Hierarchical Bayesian networks

The third possible approach to creation of summary nodes is one based on hierarchical Bayesian networks framework. We use the obtained earlier output of matrix factorization in order to group TV program/website variables using hierarchical clustering. We utilize cosine distance as dissimilarity measure. Examination of dendrograms and elbow plots for clustering of both TV programs and websites data suggests that in both cases we can distinguish five clusters. We note that almost equally likely we could decide respectively for 11 and 9 groups, but we choose a lower number, to

better resemble ratio of summary nodes to TV programs/ websites variables which is feasible for the full datasets. For TV the grouping is rather balanced with three groups containing around 25 programs and two of them with around ten. For online dataset there is one visibly bigger group with almost half of the websites and remaining groups are balanced. Values of latent nodes are learned with E-M algorithm. Number of classes in each node is chosen as $0.1 \times q + 2$, where q is the number of nodes summarized by a latent variable. To avoid local minima we use multistart with so-called small EM - after random start 5 iterations of the algorithm are performed. The one, which delivered the best results measured by BIC criterion is used as starting point for actual EM run with convergence criterions set as 500 iterations or less than 0.001 difference between BIC of subsequent iterations.

We proceed with learning networks assuming the existence of hidden variables. Networks structures with respect to connections between demographic variables resemble closely ones learned earlier. Also, as before, main direct arcs to summary nodes come from age, gender, or income variables. Both TV and online latent nodes are densely interconnected as well. In the joint network connections between summary nodes from both datasets exist. Additionally, multiple reasonable connections between demographic variables can be found.

The goodness of fit test results, presented in Table 9, are relatively good. Differences between the training and tests data log-likelihoods are quite small, particularly for networks created on the online dataset. These are also almost in the boundaries of the goodness of fit test (for online with CC-node and for networks for both variables sets within).

	TV		**Online**		**Joint**
	w/o CC	with CC	w/o CC	with CC	
Training	-59.79	-60.85	-38.38	-38.45	-68.62
Test	-56.00	-56.81	-38.29	-37.85	-59.98
GoF - lower bound	-60.13	-61.17	-38.67	-38.54	-68.33
GoF - upper bound	-60.30	-61.38	-38.80	-38.69	-68.83

Table 9: Log-likelihood values for networks with parameters learned on specified datasets. Bootstrapped boundaries of 95% goodness of fit interval. Networks learned on reduced dataset with clustering-based summary nodes.

Conclusions on the quality of networks are supported by predictive accuracy results, which can be seen in Table 10. Both for TV and online networks introduction of CC-nodes improves their quality. For a TV data network, all metrics for the Bayesian network - training dataset comparison are close to our benchmark. Predictive performance on out-of-sample set is worse, but still reasonably good - it does not drastically differ from the differences in reach from test and train data. For network on joint dataset both in-sample and out of sample, performance is better than the chosen baseline of differences between training and test set. Model learned on the overlapping set of respondents shows good results against training set (better than

the train/test benchmark), and generalizes onto test set relatively well.

		TV		Online		Joint
		w/o CC	with CC	w/o CC	with CC	
RMSE	BN/Training	0.0317	0.0285	0.0490	0.0409	0.0320
	BN/Test	0.0409	0.0391	0.0610	0.0557	0.0448
	Test/Training	0.0311		0.0603		0.0423
MAE	BN/Training	0.0192	0.0179	0.0198	0.0177	0.0169
	BN/Test	0.0236	0.0228	0.0246	0.0235	0.0232
	Test/Training	0.0147		0.0224		0.0184
JS	BN/Training	0.0016	0.0013	0.0029	0.0024	0.0021
	BN/Test	0.0026	0.0024	0.0046	0.0042	0.0039
	Test/Training	0.0014		0.0044		0.0030

Table 10: Reach predictions comparison based on differences between reach obtained from specified datasets and one inferred from a Bayesian network. Networks learned on reduced dataset with clustering-based summary nodes.

5.2.4 Comparison of summary nodes creation techniques

Applying different summary nodes creation techniques to reduced datasets yields several interesting observations. First of all, networks with summary nodes based on matrix factorization clearly perform poorly. The difference between structures with latent variables based on clustering and the ones obtained using content categories are not that apparent. Good results for the category-based summary nodes structure are a bit surprising. For online data, approach with content categories clearly performs worse, however, for the rest of the networks the differences are not that clear. Their comparability to hierarchical Bayesian networks approach can be explained by much higher number of summary nodes used in the first method. This difference cannot be maintained for the full dataset, as the number of content categories is limited, whereas a number of latent variables can be increased relatively freely. That is why for application of the framework to the full dataset we chose hierarchical Bayesian networks approach. Nevertheless, we note that good results for category-based approach suggest that it can be an appealing alternative for analysis, in case the computational resources are very limited.

It is hard to assess if, in general, using a structure with cumulative media consumption nodes is beneficial. Results show that it depends both on an employed methodology and on datasets themselves. We decide to test both alternatives on full datasets and only after that conclude if introduction of CC-nodes is beneficial.

We also note that log-likelihood and goodness of fit tests seems to be not a good indicator of networks performance, and can be easily manipulated by changing the imaginary sample sizes. As performing the tests is computationally expensive and gains are limited we decide to not use them for assessing the full dataset application.

Finally, we observe that accuracy of models with summary nodes (trained using the approach based on hierarchical Bayesian networks) is not worse than for networks learned without them. For TV and joint networks the results are very close to the ones obtained without summary nodes. For online network we observe a visible improvement after introducing summary nodes. It suggests that introducing hidden variables is not only a necessary move used in order to make computations feasible, but it might actually reflect underlying structure in the data.

5.2.5 Full model performance

Having established general ability of Bayesian networks to represent media consumption behaviour on single datasets and best practices in learning networks with large number of variables, we assess the final quality of a joint network - one containing variables both from online and TV datasets. To create it we use networks created within hierarchical Bayesian networks framework, with CC-nodes, as they have proven to deliver the best results in earlier stages.

In earlier sections, we evaluate the quality of the joint network using only the dataset of overlapping respondents and without conducting the final step of the framework - calibration. In the calibration step we replace the parameters learned using only overlapping set of respondents, by these learned on full dataset (TV or online one) for all variables is it possible to do (variables which have all parents belonging to one dataset). We also looked at single networks (containing just TV and online variables) evaluated against datasets they were learned on. This is sufficient to assess the quality of summary nodes creation techniques and establish best practices for networks learning, as these steps are mainly performed only on single source networks. In order to draw final conclusions, we need to evaluate the ability of the joint network to represent marginal probabilities from full, single datasets, rather than only smaller one of the overlapping respondents. To do so, we follow evaluation procedures as described earlier, inferring probabilities of watching particular media content item from the network and single source dataset, respectively for TV and online. We compare the network's results before and after parameter calibration and with the performance of single networks as described in Section 4.2.5. The comparison of relevant metrics can be found in Table 11.

We observe that in comparison to both TV and online networks the joint one without calibration performs visibly worse. The difference is big enough to undermine conclusions about the usefulness of networks to represent media consumption behaviour. Fortunately, application of calibration procedure helps to improve the results. In all cases, single networks still perform better in terms of predictive accuracy on single dataset, but the results of the calibrated joint network are acceptable, and their deterioration seems to be worth obtaining the model with full media behaviour. We also note, that the calibrated, joint network shows smaller differences between performances against training and test set, which suggests better robustness against potential overfitting.

In addition to comparison of accuracy measures only, to better assess the model

		TV			Online		
		Single net	Joint net		Single net	Joint net	
			not cal.	cal.		not cal.	cal.
RMSE	BN/Training	0.0277	0.0300	0.0283	0.0401	0.0571	0.0514
	BN/Test	0.0361	0.0376	0.0362	0.0553	0.0662	0.0611
	Test/Training		0.0301			0.0582	
MAE	BN/Train	0.0173	0.0190	0.0176	0.0171	0.0247	0.0217
	BN/Test	0.0212	0.0227	0.0214	0.0231	0.0284	0.0259
	Test/Training		0.0145			0.0213	
JS	BN/Train	0.0014	0.0016	0.0014	0.0023	0.0044	0.0036
	BN/Test	0.0022	0.0024	0.0022	0.0042	0.0059	0.0050
	Test/Training		0.0015			0.0042	

Table 11: Reach predictions comparison based on differences between reach from specified datasets (TV or online) and ones inferred from the calibrated (cal.) and not calibrated (not cal.) joint Bayesian networks. Network learned on reduced dataset with clustering-based summary nodes.

quality, we also look at the behaviour of differences between reach inferred from the joint, calibrated BN and ones coming from data. In Figure 11 we can see scatter plots of reach inferred from the BN and calculated from test data plotted against one observed in the training set. Results show evaluation against respectively full TV and online datasets. Ideally, all results should be equal and thus lie on the black lines. Of course, some deviations are to be expected. Unfortunately, observed deviations for BNs in both cases show a systematical bias, which can be seen from fitted linear regressions. For higher reach values, both for TV and online variables networks slightly underestimate the reach. In case of TV, we also observe minimal overestimation for small reach values. The bias is visible, but not drastic and appears mainly for very small and very large reach values, which are rather extreme cases. It does not completely negate general advantages of the model, however, further research aimed to eliminate it, is recommended.

Both for online and TV variables the joint calibrated network performs slightly worse than the benchmark based on differences between reach in training and test sets. This advantage of the benchmark may be not stable in all cases. Size of the deviation between reach calculated from training and test dataset differs for different conditional probability queries. If a query is very specific, number of respondents in a subgroup defined by it may be low, e.g. there will be more males who are over 18 years old than men who have a dog and bought a car last year. It is important as the lower number of respondents the less precise reach estimates based on them are and thus variation of differences between reach from training and test set is higher.

In the Figure 12 we can see absolute differences between reach inferred from the network or test dataset and ones from training data for TV (for online variables we observe very similar behaviour), plotted against logarithm (base 10; applied in order

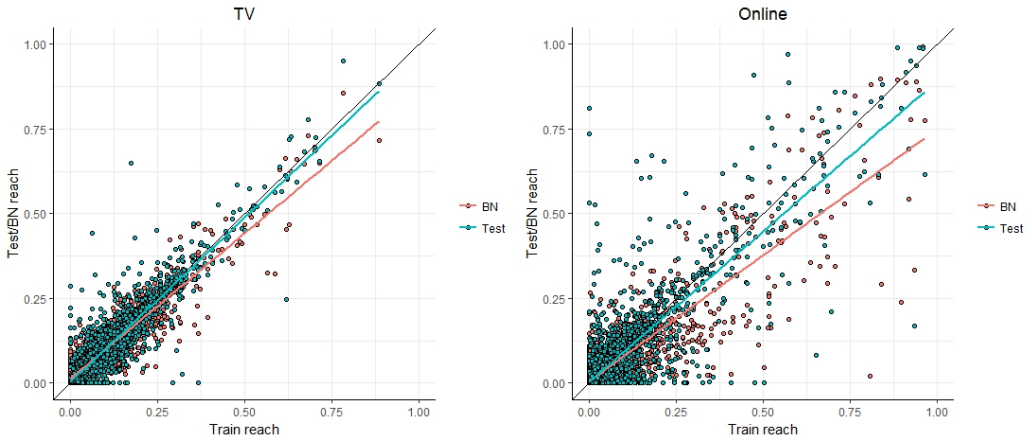

Figure 11: Scatter plot of reach inferred from the joint BN and test data plotted against reach calculated from training dataset.

to make discussed below differences more visible) of number of observations in test set for each query. Based on fitted linear trends we observe that both networks and test data follow the pattern described above - the smaller number of observations, the bigger average error. What is interesting, for smaller samples (up to around $10^{2.3}$ or 200 respondents) the network shows better average performance. Only above this value, its quality deteriorates. It suggests, that as long as errors for bigger subpopulations are on the acceptable level, Bayesian networks may be a useful tool for delivering more accurate reach estimates for very precise target groups, which so far had too small number of respondents in the panels for obtaining estimates of reasonable quality. We note, that in extreme cases with the use of BN we may even infer reach for people with characteristics which we do not observe in a panel - the testing of accuracy of such estimates is obviously impossible.

In summary, we conclude that the joint network trained using our framework performs relatively well. It shows worse results that single networks, however, parameters calibration helps to reduce the differences and proves to be beneficial. The results seem to be slightly biased, but the size of the bias is acceptable. Moreover, the model seems to perform relatively well for small subpopulations, what is promising for the planning of more precisely targeted campaigns. All in all there are shortcomings and room for improvement both of the concept and the learning framework exists. But initial results are promising.

5.3 Full dataset application

As the next step, we aim to create a model on full dataset, with all considered TV programs and websites, using findings and good practices from earlier sections. We follow the hierarchical Bayesian networks approach.

Similarly as earlier we perform a matrix factorization in order to obtain a basis for clustering. We repeat it for TV and online datasets. In both cases tuning of

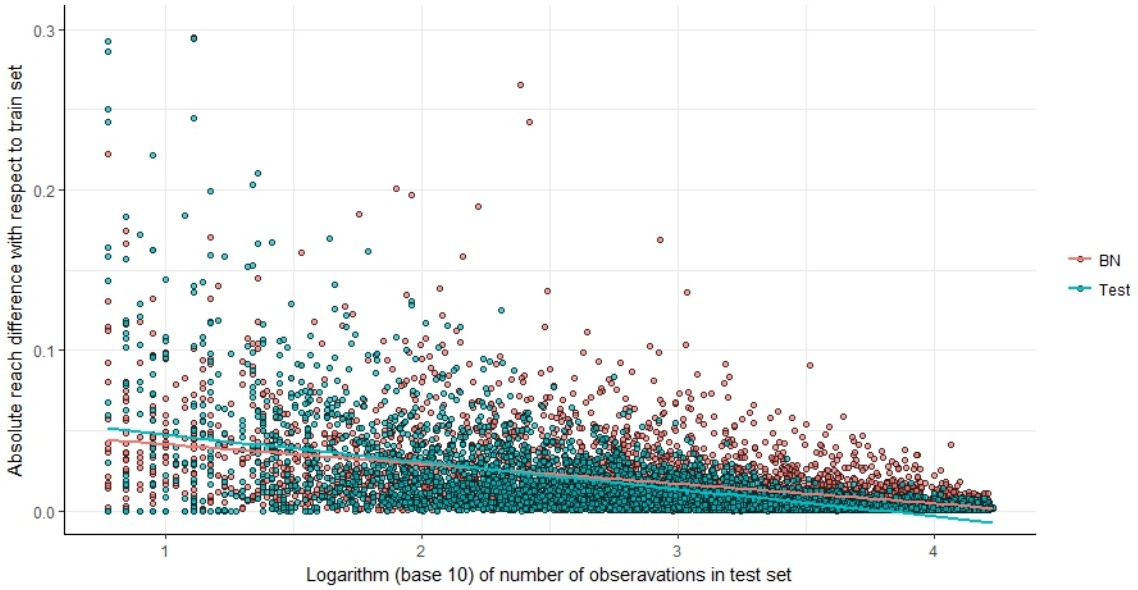

Figure 12: Scatter plot of absolute error with respect to test dataset of TV data plotted with logarithmic scale of number of observations in test dataset for restrictions in each conditional probability query.

hyperparameters is done by grid search in 5-fold cross-validation setting. Number of latent features for TV data is determined to be equal to 40 and for online to 35.

On the basis of the obtained latent features, we conduct clustering of media content items - TV programs and websites respectively. In both cases we aim to define 250 groups, as with demographic variables this amount adds up to a number of variables with which we can still learn a network in reasonable time. The grouping of websites is fairly unbalanced - the biggest cluster of websites contains 970 items (out of 5000), while in the smallest we can find only 4. Most of the clusters are small, with the median of number of websites in each equal to 12. For TV the cluster sizes are more balanced. The biggest one consists of 131 TV programs (out of 5136), the smallest one of 4, with the median equal to 15. In both cases, we observe existence of a big cluster containing rarely consumed items. For online data a similar phenomenon is also observed on the reduced dataset, while for TV it is something new.

For learning values of hidden nodes, the same approach is taken as for reduced datasets. We use EM algorithm with multistart based on "small EM" with BIC criterion. The number of classes of each latent variable is set to $0.1 \times q + 2$, with, q equal to a number of media content variables summarized by each latent one. Algorithms for all latent notes from both datasets converge. Unfortunately learning the networks with latent variables proves to be time-consuming - TV network takes around 50 hours to learn. Most of this time - 38 hours is taken by EM algorithms. For training we use the standard 36-cores Amazon AWS machine. Learning the online network takes even longer - around 87 hours, 72 of which is consumed by learning values of the hidden nodes.

		TV		Online	
		w/o CC	with CC	w/o CC	with CC
RMSE	BN/Training	0.0021	0.0071	0.0097	0.0106
	BN/Test	0.0027	0.0073	0.0257	0.0264
MAE	BN/Training	2.94E-04	9.75E-04	2.28E-03	2.64E-03
	BN/Test	3.85E-04	1.06E-03	3.72E-03	4.06E-03
JS	BN/Training	5.12E-05	2.86E-04	6.00E-04	7.04E-04
	BN/Test	7.34E-05	3.09E-04	1.10E-03	1.23E-03

Table 12: Accuracy comparison based on differences between reach from specified datasets (TV or online) and ones inferred from the single-source Bayesian networks. Networks learned on the full datasets with clustering-based summary nodes.

As discussed earlier we were not able to establish if introducing cumulative content consumption variable (CC-node representing is beneficial to a network structure. Because of this we learn, for both datasets, networks with and without it in order to compare results, using the same as earlier approach with 5000 randomly generated conditional probability queries. Results for accuracy comparison can be seen in Table 12.

We can clearly see that both for TV and online network introducing the CC- node worsens results for all considered metrics. Thus, we continue learning process without this extra variable.

Following the framework steps, we use single-source networks as starting points and the set of overlapping respondents as input data to learn a network containing all variables. Final network has 1768 nodes and 2839 arcs. For the sake of clarity we exclude in summary media content variables (TV programs and websites) and arcs between them and summarizing latent variables, as these connections are based on clustering, which was briefly discussed above. These are also not considered in structure learning, but enforced at the beginning of it. Structure learning applies directly to 632 variables - 250 latent ones for both TV and online media content and 132 socio-demographic variables, 98 from TV dataset and 34 from online one. Between these variables 1703 connections are discovered. In Table 13 we present a quantitative summary of them in a split into connections within TV and online data and in total in the network.

We observe that the network is not extremely dense, with each node being on average connected to 2.7 others. The density is unequal. Socio-demographic variables coming from TV dataset are relatively densely connected, while these from online data not. It is caused mainly by multiple connections from TV dataset's variables to online ones. Many variables from the latter dataset are connected only to TV ones - often very similar (e.g. age ranges or geographic locations). Latent variables in both datasets are equally well connected, interestingly there exists no direct connections between online and TV summary variables - all interactions between datasets are through socio-demographic nodes. It is different from what we observed on the

	TV	Online	Both
Total	1038	488	1703
Between socio-dem. variables	368	32	452
Between latent variables	395	349	744
From socio-dem. varbiables to latent ones	275	107	507
Between datasets	-	-	177

Table 13: Number of arcs by type within variables from specified datasets, excluding connections from summary variables to media content ones.

reduced dataset and might signalize some problems in the learning process - e.g. too high number of latent variables resulting in too many nodes in the network with respect to the number of observations.

In general discovered connections seem to be reasonable from common sense point of view. Most of them can be easily justified (as, e.g. relationships between Spanish background and language spoken in a household, or Internet speed and visits to certain groups of websites). With such a high number of arcs, we also observe some connections which cannot be easily explained (like, e.g. owning a foreign vehicle and visiting certain sites, not related to cars, or use of bottled water and number of TV sets), but these are in the vast minority.

After learning the structure of the joint network, we perform the calibration of parameters using full datasets of TV and online respondents in order to calibrate respective nodes. Such network is then evaluated in the same manner as the one in the Section 5.2.5 - we separately generate sets of conditional probability queries containing variables from TV dataset, online dataset and both datasets. Each query returns a reach of media content variable, which is then compared to value obtained from the training and tests sets. Differences between reach returned by network, and reach calculated from data are treated as Such oion errors.btained prepredictdiction errors are compared to the differences between reach calculated from test and training datasets. They serve as a randomness baseline, which we treat as a benchmark. Also, we include a naive prediction of a reach inferred from training datasets. We define the naive prediction as reach of a TV program/website in the whole population. It is compared to a reach from training dataset, calculated for a subpopulation defined by a query. Accuracy results can be seen in Table 14.

It can be seen that in all cases the network performs worse than the naive prediction. It is surprising taking into consideration relatively good results for smaller networks. However, we need to bear in mind that modelling of the full data - with significantly higher number of variables, and with only a few respondents actually consuming a media content in many of them, is a challenging task. We also observe that for online and joint comparisons naive prediction seems to perform better than the differences between test and training data treated as benchmark. It is partially illusory - bad accuracy of the benchmark is caused by high variance, while naive pre-

		TV	Online	Both
RMSE	BN/Training	0.0025	0.0080	0.0040
	BN/Test	0.0031	0.0262	0.0016
	Test/Training	0.0015	0.0253	0.0039
	Naive/Training	0.0024	0.0104	0.0013
MAE	BN/Training	3.35E-04	1.41E-03	5.06E-04
	BN/Test	4.20E-04	2.96E-03	8.03E-04
	Test/Training	2.51E-04	2.45E-03	5.17E-04
	Naive/Training	3.11E-04	1.47E-03	3.37E-04
JS	BN/Training	6.48E-05	3.16E-04	1.19E-04
	BN/Test	8.80E-05	9.27E-04	1.99E-04
	Test/Training	3.53E-05	7.48E-04	1.20E-04
	Naive/Training	5.69E-05	3.16E-04	7.56E-05

Table 14: Reach predictions comparison based on differences between reach from specified datasets (TV or online or joint) and ones inferred from the joint Bayesian network. Network learned on the full dataset with clustering-based summary nodes.

diction shows visible bias, with low dispersion for lower reach predictions (in cases when introducing a restriction on socio-demographic variables values changes the reach only slightly). The network results are less biased then naive ones, which is a good sign. We can see it on the scatter plot (Figure 13) showing test, the network's and naive prediction plotted against reach in the training set, for online data.

The high number of media content variables with low reach, causes additional difficulties in assessing the model. Low reach translates into low number of respondents consuming these TV programs/websites, and thus higher randomness of the results. To get an idea of network's performance on more stable target variables and compare the network accordingly to earlier models trained on the reduced dataset, we perform an additional evaluation, in which we select as target variables only media content variables present in reduced datasets. Rest of the procedure is identical to the ones used earlier. Results of such evaluation scheme are presented in Table 15.

After taking into consideration only most popular media content, we can see that network performance is still bad. Nevertheless, in all cases, the network (as well as the benchmark) is better than naive prediction. It shows (in combination with lower bias of network than naive predictions for a standard evaluation) that the model is not as bad as one might expect from results based on using all media content variables as target ones.

The accuracy of the network on the full dataset is disappointing. Despite the fact that we can reasonably conclude that it is better than naive predictions, high variance and a visible bias of the results deems it unusable for any practical purpose. To some extend it can be attributed to the general, randomness of consumption behaviour with respect to low-reach content. Bad results of the full model, also for only more popular TV programs/websites, in comparison to relatively good results obtained for

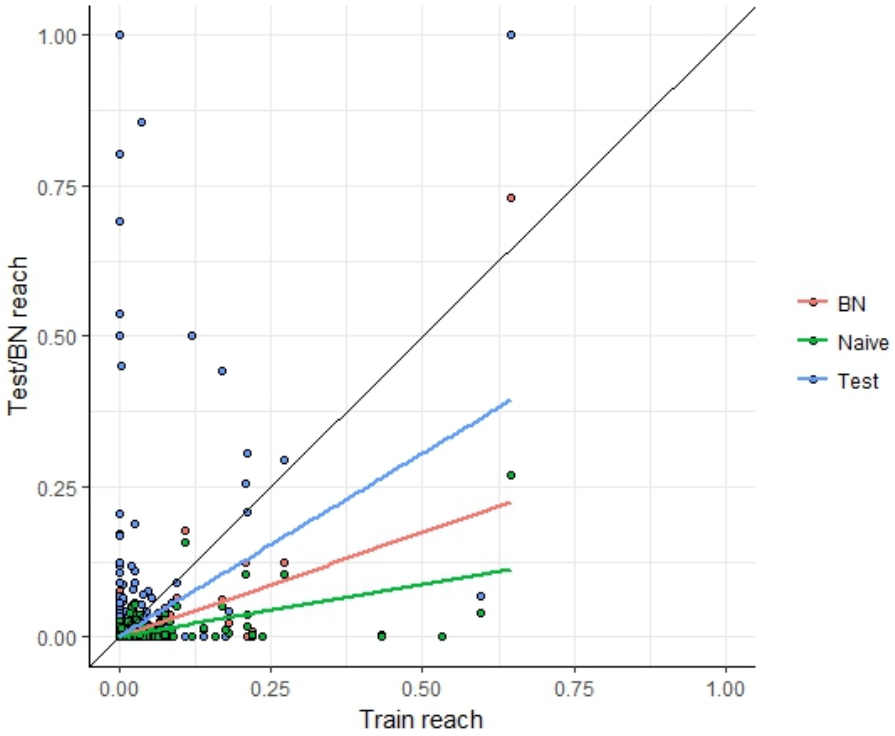

Figure 13: Scatter plot of reach inferred from the joint BN and test data plotted against the online training dataset.

the model on reduced data, suggest that proposed framework may be insufficient for the full data with such high number of variables. These experiences are not negating the validity of the general concept of using Bayesian networks for the analyzed task. Possibly some extensions of the proposed framework, like including more layers of hidden nodes, might help to improve accuracy of used model.

6 Conclusion

As the goal of this thesis we test the usefulness of Bayesian networks for efficient modelling of a reach of TV programs and websites. Such models should be able to replace typically used logic of simply filtering respondent-level data in delivering insights for advertisement planning and other related tasks. We believe, that the Bayesian networks combine the advantages of statistical models - low-memory usage and statistical verification of dependencies, with flexible inference characteristic for respondent-level approaches.

In order apply the Bayesian networks for the described task we need to overcome two main problems - learning networks on the basis of two only partially overlapping datasets, and learning networks with a high number of variables. In this thesis, we

		TV	Online	Both
RMSE	BN/Training	0,0329	0,0702	0,0389
	BN/Test	0,0408	0,0796	0,0337
	Test/Training	0,0267	0,0595	0,0300
	Naive/Training	0,0435	0,0902	0,0411
MAE	BN/Training	0,0126	0,0285	0,0127
	BN/Test	0,0159	0,0324	0,0154
	Test/Training	0,01	0,0215	0,0109
	Naive/Training	0,0144	0,0342	0,0122
JS	BN/Training	0,0021	0,0054	0,0023
	BN/Test	0,0029	0,0067	0,0029
	Test/Training	0,0014	0,0037	0,0017
	Naive/Training	0,0029	0,0075	0,0025

Table 15: Reach predictions comparison based on differences between reach from specified datasets (TV or online or joint) and the ones inferred from the joint Bayesian network. Evaluation using only top 100 watched TV programs and top 100 visited variables.

propose a consistent, based to some extent on earlier research in different domains, framework to deal with these issues. We also test its different variants and general best practices in modelling media content reach with Bayesian networks.

We test the concept of applying Bayesian networks for media reach modelling on reduced data of 100 most watched TV programs and 100 most vised websites, as well as on full dataset consisting of over 10.000 media content variables. Model accuracy on reduced dataset shows good performance. Importantly application of proposed framework helps not only to produce the joint model for all variables based on only partially overlapping data, but also improves the quality of individual models trained on single datasets. All of this suggest that proposed solutions could be successfully used in business practice.

Obtaining the network for both TV and online variables allows us to discover substantial evidence of the existence of relationships between socio-demographic variables and TV programs/websites reach. We also, observe direct (not explainable purely by socio-demographic characteristics) relationships between consumption of TV and online content.

Good results of the small-scale model on reduced datasets do not translate into a performance of the network containing all media content variables. Its accuracy being better than naive predictions remains insufficient for any practical use. This conclusion does not undermine usefulness of Bayesian networks for modelling data with lower number variables concluded above.

As our research are to big extend precursory, both with respect to analyzed data and proposed methodology our research is subject to various limitations. The relative simplicity of the approach expressed mainly by the use of discrete (not continuous)

Bayesian networks and only one layer of latent nodes based on clustering. We believe both simplifications were necessary at this stage of the research. Techniques of learning mixed, non-Gaussian continuous Bayesian networks (which would be necessary in our case) are far more complex than these used for discrete models. Given limited computational resources it would not be feasible for us to handle this additional complexity. Limitations in computational resources also influenced the decision to create only one layer of hidden variables, learning of which already took a significant amount of time.

Big difference in the quality of the results both on reduced as well as full dataset suggests that our solutions might indeed be too simplistic. Addressing suggested limitations might sufficiently improve the model performance. Especially increasing number of hidden layers may be the optimal model development strategy, as it can be done within our framework, even without complex adjustments (e.g. a structure of new layers can be built on the basis of already performed hierarchical clustering).

We also note that several other elements of the proposed approach could be improved as well. Nevertheless, we suspect improvement in these additional areas would not have as big impact as these mentioned earlier. We see most promising areas to be: an approach to networks structure learning, defining clusters of variables for latent nodes creation and improvements in learning parameters of latent nodes. Knowledge discovery by learning network's structure is always a difficult task. Optimizing structure learning, both with respect to choice of an algorithm and a score, as well as the incorporation of expert's knowledge could potentially have a beneficial impact on the model. The proposed method of clustering the media content variables based on the use of matrix factorization output could be modified as well, in order to perform dimensional reduction and clustering at once, minimizing the chances of potential errors. Two-mode clustering could be used for this purpose (e.g. (van Dijk et al., 2009)). Finally, improvements in EM algorithm, which is prone to finding local minima could be implemented, e.g. by use of smart staring points. It could also improve the convergence time. With respect to learning latent nodes, choice of the number of classes of latent variables could also be optimized.

Further research, in addition to focusing on improving suggested framework, could also focus on extending it for a wider problem formulation. Example of such directions might be inclusion of a time component in the model (possibly by using dynamic Bayesian networks) or developing techniques of tuning Bayesian networks parameters with use of additional information from extra data sources. We note that not necessarily the Bayesian networks, but also other probabilistic graphical models or generative models in general can be considered useful for the researched task.

We believe that, despite problems and shortcomings, which are hard to avoid in early stages of development, both postulated use of Bayesian networks for delivering insights into the reach of media content, and proposed framework for learning such models are promising. In our opinion, relatively good results on reduced datasets and existing visible directions for improvement of a model learning process prove it to be an interesting area for further research.

Appendix

A Socio-demographic variables used

Short name	Full name
agerange	Age
gendercode	Gender
principalshopper	Household principal shopper indicator
languageclasscode	Language used
workinghours	Working hours
ladyofhouseholdflag	Lady of the household indicator
headofhouseholdflag	Head of the household indicator
relationshiptoheadofhouseholdcode	Relationship status of head of the household
nielsenoccupationcode	Occupation
worksoutsideofhomeflag	Working outside home indicator
principalmoviegoerflag	Principal movie goer
educationranges	Education
internetusagehome	Internet usage at home
internetusagework	Internet usage at work
territorycode	Territory code
timezonecode	Timezone code
countysizecode	County size code
headofhouseholdrace	Head of the households race
householdlanguage	Main household language
children	Children indicator
childrenunder2	Children under 2 indicator
childrenunder3	Children under 3 indicator
children2to5	Number of children 2-5 years old
childrenunder6	Number of children under 6
children6to11	Number of children 6-11 years old
children12to17	Number of children 12-17 years old
numberofchildrenunder3	Number of children under 3
numberofchildren	Number of children
numberofadults	Number of adults in the household
householdsizecode	Household size code
householdincomecode	Household income
numberofincomes	Number of incomes
headofhouseholdagebreak	Head of the household age
headofhouseholdeducation	Head of the household education
headofhouseholdoccupation	Head of the household occupation

ladyofhousepresentflag	Lady of the household present indicator
ladyofhouseoccupationcode	Lady of the household occupation
wiredcable	Wired cable connected
paychannels	Access to pay-channels
cableplus	Cable plus
alternatedeliverysystem	Alternative TV provider
wireddigitalcable	Digital cable wired
dbsowner	DB owned
dvdowner	DVD reader owned
presenceofdvr	DVR owned
numberoftvsets	Number of TV sets
numberoftvsetswithpay	Number of TV sets with pay channels
numberoftvsetswithwiredcable	Number of TV sets with wired cable
numberoftvsetswithwiredcableandpay	Number of TV sets with wired cable and pay channels
numberofvcrs	Number of VCRs
videogameowner	Video games owner
headofhouseholdorigincode	Head of the household origin
headofhouseholdhispanic	Head of the household Hispanic ethnicity
numberofdvrs	Number of DVRs
householdwithcableservicesviatelco	Household with cable from TelCo
numberofcars	Number of cars
numberoftrucks	Number of trucks
newcarprospectlast3years	New car bought in last 3 years
newcarprospectlast5years	New car bought in last 5 years
newtruckprospectlast3years	New truck bought in last 3 years
newtruckprospectlast5years	New truck bought in last 5 years
domesticvehicleindicator	Domestic vehicle owned
foreignvehicleindicator	Foreign vehicle owned
dogindicator	Dog indicator
catindicator	Cat indicator
pcaccesshomeindicator	Access to PC at home
pcaccesswithinternetaccesshomeindicator	Access to PC with Internet at home
householdincomerangesdetailed	Household income range
householdincomeamount	Household income amount
householdincomenonworking	Household income from working
headofhouseholdgender	Gender of head of the household
headofhouseholdworksoutsidehome	Head of the household working outside home
meteredmarketflag	Metered market
homeownershipstatuscode	Homeownership status

homestructuretype	Home structure type
homeownershipsecondaryhomestatus	Homeownership status secondary home status
beverageusagebottledwater	Beverage usage bottled water
beverageusagecoffeeortea	Beverage usage bottled coffee/tea
beverageusagesoftdrinks	Beverage usage soft drinks
beverageusagetablewine	Beverage usage table wine
nsimarketrankranges	NSI market rank
telephonestatuscode	Telephone status
collegestudentaway	College student away
hdcapablehome	HD cable
hdcapablereceivablehome	HD cable receivable
hdtvdisplaycapable	HD TV
householdinternetconnectionspeed	Household Internet connection speed
householdtelephonecapability	Household telephone capacity
numberofoperablecomputerscode	Number of operable computers
numberofoperablelaptopscode	Number of operable laptops
numberofoperabledesktopscode	Number of operable desktops
numberofoperablecomputerswindowsoscode	Number of operable computers with Windows
numberofoperablecomputersmacoscode	Number of operable computers with Mac OS
numberofoperablecomputersotheroscode	Number of operable computers other
broadbandonlyhousehold	Broad band only indicator
asianhouseholdindicator	Asian household
workingwomenflag	Working women indicator
moviegoer	Movie goer

Table 16: Full list of socio-demographic variables for respondents in TV panel.

Short name	**Full name**
age	Age
gender_id	Gender
education_id	Education
race_id	Race
hispanic_origin_id	Hispanic origin indicator
web_access_locations	Location of web access
working_status_id	Working status
occupation_id	Occupation
industry_group_id	Employment industry
org_size_id	Size of employer
life_stage_id	Life stage
department_type_id	Department of employment
purchase_influence_id	Purchasing influence
members_2_11_count	Household members 2-11 years old
members_12_17_count	Household members 12-17 years old
household_size_id	Household size
income_group_id	Income group
property_type_id	Type of owned property
web_access_computers_id	Number of computers with web access
primary_isp_id	Primary Internet service provider
web_conn_speed_id	Internet speed - supposed
web_conn_speed_metered	Internet speed - measured
county_size_id	County size
census_region_id	Census region
census_division_id	Census division
usage_rank_id	Rank of Internet usage
hoh_flag	Head of the household indicator
prompt_status_id	Property status
spanish_lang_dominance	Spanish dominance
primary_language	Primary language
is_mobile_phone	Access to mobile phone
is_internet_mobile	Access to mobile Internet
use_internet_mobile	Usage of mobile Internet
hh_landline_ph	Phone landline in household

Table 17: Full list of socio-demographic variables for respondents in online panel.

B Bayesian networks - theory

B.1 Parameter learning

For parameter learning in the thesis we make use of Bayesian framework. We assume a multinomial distribution for our data. The Bayesian approach requires additionally to specify priors for parameters in the Bayesian network. To effectively do so we assume global and local parameter independence. These are widely used and not very restrictive assumptions, true for vast majority of the domains (Koller and Friedman, 2009). Global parameter independence states that parameters $\boldsymbol{\theta}_{X_i|\boldsymbol{Pa}_{X_i}}$ for each X_i are independent from each other. Local parameter independence extends it to parameter independence from variables parents as well, such that if variable parents set is $\boldsymbol{P} = \boldsymbol{Pa}_{Xi}$, and $P_j, P_k \in \boldsymbol{Pa}_{Xi}$ than $\boldsymbol{\theta}_{X_i|P_j}$ and $\boldsymbol{\theta}_{X_i|P_k}$ are also independent.

Under this assumption we can treat each parameter of local PDF of variable X_i, conditioned on its parents configuration and values, independently (Heckerman et al., 1995).

As Dirichlet distribution is conjugate prior for multinomial one, we assume our parameters follow it. Under this and previous assumptions the prior distribution over the set of parameters $\mathbf{p}$ of a Bayesian network can be written as:

$$P(\mathbf{p}|G) \propto \prod_{i=1}^{n} \prod_{j=1}^{q_i} \prod_{k=1}^{r_i} \frac{p_{ijk}^{\alpha_{ijk}-1}}{\Gamma(\alpha_{ijk})}, \tag{3}$$

where α_{ijk} stands for hyperparmeters, and it is equal to number of cells in local PDF table of X_i for parent set expressed by q_i, such that $\alpha_{ij} = \sum_{k=1}^{r_i} \alpha_{ijk}$, $k = 1, ..., r_i$.

Since we use conjugate prior, the posterior also follows the Dirichlet distribution. It is:

$$P(p|G, D) \sim Dir(N_{ij1} + \alpha_{ij1}, N_{ij1} + \alpha_{ij1}, ..., N_{ijr_i} + \alpha_{ijr_i}) \tag{4}$$

Typically, in applied setting for Bayesian networks, we do not investigate full posterior distributions of parameters but use only their modes. Based on that parameter estimate for i-th variable, for j-th parent variable taking h-th value is equal to:

$$\hat{p}_{ijk} = \frac{\alpha_{ijk} + N_{ijk}}{\alpha_{ij} + N_{ij}}. \tag{5}$$

In the analysis, we assume a uniform prior over each node's parameters. This means that in each case $\alpha_{ijk} = \alpha_{ij}/r_i$. Parameter α_{ij} in that setting is often referred to as imaginary sample sizes and determines the strength of the prior.

B.2 Structure learning

Scored-based learning Central for scored-base algorithms is choice of a score, which determines goodness of fit of the network. We focus on Bayesian information criterion (BIC) as an approximation of a networks Bayesian score, which can be expressed as:

$$BIC(G, D) = logP(D|\hat{\mathbf{p}}, G) - \frac{d}{2}logN \tag{6}$$

$$d = \sum_{i=1}^{n} q_i(r_i - 1), \tag{7}$$

with r_i equal to number of bins in local PDF of X_i for parent set expressed by q_i.

BIC has several desirable properties, most importantly it asymptotically (with number of observations going to infinity) favours structure, which exactly fits dependencies in the data (consistency), it gives the same values for all structures encoding the same set of independencies (score equivalence), and it is decomposable – an be computed for each variable separately (Koller and Friedman, 2009). The latter is particularly important in the learning process as it allows to assess a change in the score caused by change of one arc by recalculating only decomposed scores for variables under consideration without doing it for the whole network.

For the actual structure search, we utilise tabu algorithm (Scutari, 2010), which is essentially a greedy hill climbing search with additional restrictions. Beginning with some starting structure, it scores each possible operation on the graph. These operations are - between each pair of variables, if the arc exists – removing or reverting it; if it does not exist – adding one. Scoring operations separately is possible due to mentioned earlier score – decomposability. After obtaining the scores, it performs operation, which gives the biggest improvement in the network score and iterates further up to the point where the score cannot be improved. Tabu algorithm additionally prevents the operations performed in the last n iterations to be reversed, improving convergence speed. As every greedy approach tabu algorithm is prone to converge to the local optimum, which can be at least partially prevented by using multi-start.

Constraint-based learning The main focus in constraint-based algorithms is put on learning variables Markov blanket - a set of variables which render it independent from all others; in Bayesian networks, it is equivalent to variables children and parents of variables itself and its children. To efficiently discover Markov blanket for each variable we utilize fast-IAMB algorithm (Yaramakala and Margaritis, 2005).

The fast-IAMB algorithm consists of two phases - growing and shrinking one. In the growing phase, variables are added to the Markov blanket of given variable. Multiple variables can be added in each iteration. The first number of independence tests is conducted. Then variables are sorted based on tests significance, and being included in the Markov blanket as long as a number of instances in the contingency table is big enough for the test to be reliable (with suggested number equal to 5). When this limit is reached to move to shrinking phase, in which we remove from the Markov blanket all variables, which has become insignificant after adding new ones. The procedure is repeated until there are no variables are removed in the second phase.

Having Markov blanket for each variable roles (being a child or parent) of other ones in it are determined by comparing p-values of relevant tests. Then the network is constructed based on it.

As the independence test we use suggested by authors of the algorithm mutual information test, based on information-theoretic distance measure expressed as (Kullback, 1959):

$$MI(X, Y|\boldsymbol{Z}) = \sum_{i}^{R}\sum_{j}^{C}\sum_{k}^{L}\frac{n_{ijk}}{n}log\frac{n_{ijk}n_{..k}}{n_{i.k}n_{.jk}} \tag{8}$$

where X, Y are the variables we test independence of, $\boldsymbol{Z}$ is in our case current Markov blanket, and $\{n_{ijk}, i = 1, ..., R; j = 1, .., C; k = 1, ..., L\}$ are respective observed frequencies. MI is proportional to the log-likelihood ratio. For implementation details we refer to (Scutari, 2009). In the thesis we also learn networks with latent variables. The details on our approach to that cna be found in Section C.3.

C Supporting techniques

C.1 Matrix factorization

Matrix factorisation is a popular and efficient method used especially in fields where one has to deal with high-dimensional and sparse data, like text mining, image processing or recommender systems. It proved to be robust to noise in the data, computationally efficient and deliver sparse and meaningful latent features (Gillis, 2014). Discovered latent variables have also been successfully used for later clustering (Xu et al., 2003).

The main idea behind matrix factorization is decomposition of matrix $R_{m\times n}$, which consists of observations for n users for m variables. We aim to approximately represent such matrix as a product of two lower dimensional ones - $P_{k\times m}$ and $Q_{k\times n}$. In this representation value for user i of j-the variable is predicted by vectors product $\mathbf{p}_i'\mathbf{q}_j$ (Koren et al., 2009). In recommender systems or similar setting matrix, P can be interpreted as latent characteristics of media content and matrix Q as users preferences towards them.

The task of finding proper values for lower-dimensional matrices is typically posed as an optimization problem, expressed as (Chin et al., 2016):

$$\min_{P,Q} \sum_{i,j\in R} [(r_{i,j} - \mathbf{p}_i'\mathbf{q}_j)^2 + \mu_p \left\|\mathbf{p}_i\right\|_1 + \mu_q \left\|\mathbf{q}_j\right\|_1 + \frac{\lambda_p}{2} \left\|\mathbf{p}_i\right\|_2^2 + \frac{\lambda_q}{2} \left\|\mathbf{q}_j\right\|_2^2], \tag{9}$$

where vector norms are added as regularization terms and μ_p, μ_q, λ_p, λ_p are treated as hyperparameters. The number of latent dimensions k along with these hyperparameters is typically determined by standard parameter tuning.

Finding a solution for such stated optimization problem is not trivial. Many methods have been proposed to solve it, e.g. (Lee and Seung, 2001; Koren et al., 2009; Balan et al., 2011), probably most popular ones being based on stochastic gradient (SG) approach. Its main idea is randomly selecting one $r_{i,j}$, calculating gradient for it and updating corresponding $\mathbf{p}_i$, $\mathbf{q}_j$. The procedure is repeated until convergence. In the thesis we employ parallelized variation on this algorithm - Fast Parallel Stochastic Gradient (FPSG) (Chin et al., 2015) in the implementation by (Qiu et al., 2017).

As our data are non-negative we use a variant of the algorithm, which constraints search space to non-negative values.

C.2 Hierarchical clustering

The main aim of clustering is grouping objects into such groups that objects clustered together are more similar to each other than to ones in other groups. Many methods have been proposed to archive this objective, in the thesis, we make use of hierarchical clustering.

Hierarchical clustering is based on iterative approach. Firstly, we assume all objects are separate clusters. Then, in each step we merge two, the most similar to

each other, clusters into one. Similarity is based on distance matrix, with most often Euclidean distance used as dissimilarity measure. Cluster merging is followed by updating the distance matrix, replacing rows and columns of merged groups by joined one, with recalculated distances. General formula for updated distance can be expressed as follows (Lance and Williams, 1967):

$$d_{A+B,C} = \alpha \times d_{C,A} + \beta \times d_{C,B} + \gamma \times d_{A,B} + \delta \times |d_{C,A} - d_{C,B}|, \tag{10}$$

where A, B, C symbolize clusters, $d_{i,j}$ distance measure between clusters i and j, and α, β, γ, δ are parameters depending on choice of particular method. In the thesis we employ Ward's method, which aims to minimize inter-group variance. It tends to produce relatively balanced classes and is rather resistant to chaining (subsequently adding all variables to one class). Mentioned above parameters take values (Ward Jr, 1963):

$$\alpha = \frac{n_A + n_C}{n_A + n_B + n_C},\ \beta = \frac{n_B + n_C}{n_A + n_B + n_C},\ \gamma = \frac{-n_C}{n_A + n_B + n_C},\ \delta = 0, \tag{11}$$

where n_i stands for size of cluster i.

Hierarchical clustering algorithm ends with all observations being contained in one cluster. It is researchers role to decide into how many clusters split the data using algorithms output - list of subsequent merging steps and distances between merged clusters at each steps (heights). These are often visualized with so-called dendrogram. A number of clusters is then often chosen based on elbow plot of heights or by finding local maximum of ratios of subsequent heights. Alternatively measures of quality of obtained groups as Calinski-Harabasz index (Caliński and Harabasz, 1974) or Silhouette width (Rousseeuw, 1987) can be used.

C.3 Learning hidden variables

Introducing hidden variables to a network can substantially complicate already complex learning tasks, especially if a number of hidden nodes, their placement or cardinality are not known. Hierarchical latent class framework simplifies the problem. The network structure is imposed based on external evidence – in our case with the use of clustering. Nodes in each layer are assumed to be independent of each other and are treated as known variables for further learning steps. Also, observed variables connected to one node are assumed to be independent given this node.

All these assumptions allow us to treat task of learning each hidden variable values individually. Thus, for each hidden node, we are faced with Naïve Bayes-like structure with unknown class node. Distributions of all observed variables connected to it are conditioned on its value. It means we can perceive learning its values as classification task expressed as mixture model for multinomial distributions. The mixture model for respondent i can be expressed as (Everett, 2013):

$$f(\mathbf{x}_i|\boldsymbol{\theta}) = \sum_{k}^{K} \pi_k \times h(\mathbf{x}_i|\boldsymbol{\alpha}_k), \tag{12}$$

where $\mathbf{x}_i$ is set of observed values for respondent i, $\boldsymbol{\theta}$ is vector of parameters of mixture model, K equals to number of latent classes, π_k are latent classes proportions, with $\sum_k^K \pi_k = 1$, and $\boldsymbol{\alpha}_k$ is class specific vector of parameters. Function $h(\mathbf{x}_i|\boldsymbol{\alpha}_k)$ for mixture of multinomials becomes:

$$h(\mathbf{x}_i|\boldsymbol{\alpha}_k) = \prod_{j=1}^{m} \prod_{h=1}^{d_j} (\alpha_{kjh})^{\mathbf{x}_{ijh}}, \tag{13}$$

where m is the number of observed variables, and d_j the cardinality of j-th one.

Identifiability of such formulated model has to be considered. The number of parameters is equal to $K \times \sum_j (m_j - 1) + (K - 1)$ and it has to be lower then number of observations or the number of combinations of possible values for observed variables. We note the parameters number depends heavily on cardinality of latent variable, choice of which is a problem of its own. We decide on number of latent classes by using heuristic approach proposed by (Mourad et al., 2011). Cardinality of hidden node depends on m and is equal to $a \times m + b$, where $0 < a < 1$, $b > 1$ are parameters of our choice.

Parameters of the model are estimated by maximizing log-likelihood function:

$$f(\mathbf{x}_i|\boldsymbol{\theta}) = \sum_{i=1}^{n} \ln \sum_{k}^{K} \pi_k \prod_{j=1}^{m} \prod_{h=1}^{d_j} (\alpha_{kjh})^{\mathbf{x}_{ijh}}. \tag{14}$$

It proves to be difficult task. Popular solution for finding parameters estimates is Expectation-Maximization (EM) algorithm (Dempster et al., 1977). It is iterative method consisting of Expectation (E) and Maximization (M) steps. E-step creates an expectation function for log-likelihood under assumption of non-stochasticity of current parameter estimates (only class affiliations are treated stochastically). M-step switches the roles and maximizes obtained expected log-likelihood function to find parameter estimates.

In the thesis we use EM implementation by (Langrognet et al., 2016). Authors use slightly different formulation of the model function, introduced by (Celeux and Govaert, 1991). It aims to improve identifiability of the model by making it easier to impose restrictions on parameters. We note that for unrestricted models both formulations are equivalent. For details we refer to (Biernacki et al., 2008).

References

I. Abbasnejad and D. Teney. A hierarchical Bayesian network for face recognition using 2d and 3d facial data. In *Machine Learning for Signal Processing (MLSP), 2015 IEEE 25th International Workshop on*, pages 1–6. IEEE, 2015.

G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE transactions on knowledge and data engineering*, 17(6):734–749, 2005.

A. K. Balan, L. Boyles, M. Welling, J. Kim, and H. Park. Statistical optimization of non-negative matrix factorization. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 128–136, 2011.

P. Baldi, P. Frasconi, and P. Smyth. Modeling the internet and the web. *Probabilistic methods and algorithms*, 2003.

R. M. Bell and Y. Koren. Lessons from the netflix prize challenge. *Acm Sigkdd Explorations Newsletter*, 9(2):75–79, 2007.

C. Biernacki, G. Celeux, G. Govaert, F. Langrognet, G. Noulin, and Y. Vernaz. Mixmod-statistical documentation. downloadable from http://www. mixmod. org. *IMG/pdf/statdoc_2_1_1. pdf*, 2008.

J. G. Blodgett and R. D. Anderson. A Bayesian network model of the consumer complaint process. *Journal of Service Research*, 2(4):321–338, 2000.

J. S. Breese, D. Heckerman, and C. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*, pages 43–52. Morgan Kaufmann Publishers Inc., 1998.

T. Caliński and J. Harabasz. A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods*, 3(1):1–27, 1974.

R. Catherine and W. Cohen. Personalized recommendations using knowledge graphs: A probabilistic logic programming approach. In *Proceedings of the 10th ACM Conference on Recommender Systems*, pages 325–332. ACM, 2016.

G. Celeux and G. Govaert. Clustering criteria for discrete data and latent class models. *Journal of classification*, 8(2):157–176, 1991.

D. M. Chickering. Learning Bayesian networks is np-complete. In *Learning from data*, pages 121–130. Springer, 1996.

W.-S. Chin, Y. Zhuang, Y.-C. Juan, and C.-J. Lin. A fast parallel stochastic gradient method for matrix factorization in shared memory systems. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 6(1):2, 2015.

W.-S. Chin, B.-W. Yuan, M.-Y. Yang, Y. Zhuang, Y.-C. Juan, and C.-J. Lin. Libmf: a library for parallel matrix factorization in shared-memory systems. *The Journal of Machine Learning Research*, 17(1):2971–2975, 2016.

P. Dagum and M. Luby. Approximating probabilistic inference in Bayesian belief networks is np-hard. *Artificial intelligence*, 60(1):141–153, 1993.

M. J. Daly, J. D. Rioux, S. F. Schaffner, T. J. Hudson, and E. S. Lander. High-resolution haplotype structure in the human genome. *Nature genetics*, 29(2):229–232, 2001.

L. M. De Campos, J. M. Fernández-Luna, J. F. Huete, and M. A. Rueda-Morales. Combining content-based and collaborative recommendations: A hybrid approach based on Bayesian networks. *International Journal of Approximate Reasoning*, 51 (7):785–799, 2010.

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

G. Dror, N. Koenigstein, Y. Koren, and M. Weimer. The yahoo! music dataset and kdd-cup'11. In *Proceedings of KDD Cup 2011*, pages 3–18, 2012.

B. Everett. *An introduction to latent variable models*. Springer Science & Business Media, 2013.

N. Friedman, M. Linial, I. Nachman, and D. Pe'er. Using Bayesian networks to analyze expression data. *Journal of computational biology*, 7(3-4):601–620, 2000.

N. Gillis. The why and how of nonnegative matrix factorization. *Regularization, Optimization, Kernels, and Support Vector Machines*, 12(257), 2014.

G. Greenspan and D. Geiger. High density linkage disequilibrium mapping using models of haplotype block variation. *Bioinformatics*, 20(suppl 1):i137–i144, 2004.

E. Gyftodimos and P. A. Flach. Hierarchical Bayesian networks: an approach to classification and learning for structured data. In *Hellenic Conference on Artificial Intelligence*, pages 291–300. Springer, 2004.

D. Heckerman, D. Geiger, and D. M. Chickering. Learning Bayesian networks: The combination of knowledge and statistical data. *Machine learning*, 20(3):197–243, 1995.

D. Heckerman, D. M. Chickering, C. Meek, R. Rounthwaite, and C. Kadie. Dependency networks for inference, collaborative filtering, and data visualization. *Journal of Machine Learning Research*, 1(Oct):49–75, 2000.

W. Jaronski, J. Bloemer, K. Vanhoof, G. Wets, et al. Use of Bayesian belief networks to help understand online audience. In *Proc. Data Mining Marketing Appl. Workshop ECML/PKDD 2001*. Freiburg Germany, 2001.

C.-R. Kim. Identifying viewer segments for television programs. *Journal of Advertising Research*, 42(1):51–66, 2002.

G. Kimmel and R. Shamir. Gerbil: Genotype resolution and block identification using likelihood. *Proceedings of the National Academy of Sciences of the United States of America*, 102(1):158–162, 2005.

M. H. Kolekar and S. Sengupta. Bayesian network-based customized highlight generation for broadcast soccer videos. *IEEE Transactions on Broadcasting*, 61(2): 195–209, 2015.

D. Koller and N. Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

F. Kooti, K. Lerman, L. M. Aiello, M. Grbovic, N. Djuric, and V. Radosavljevic. Portrait of an online shopper: understanding and predicting consumer behavior. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*, pages 205–214. ACM, 2016.

Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8), 2009.

S. Kullback. Statistics and information theory. *J. Wiley and Sons, New York*, 1959.

G. N. Lance and W. T. Williams. A general theory of classificatory sorting strategies: Ii. clustering systems. *The computer journal*, 10(3):271–277, 1967.

F. Langrognet, R. Lebret, C. Poli, and S. Iovleff. *Rmixmod: Supervised, Unsupervised, Semi-Supervised Classification with MIXture MODelling (Interface of MIXMOD Software)*, 2016. URL `https://CRAN.R-project.org/package=Rmixmod`. R package version 2.1.1.

D. D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. In *Advances in neural information processing systems*, pages 556–562, 2001.

H.-k. Lee, J. Cha, and M. Kim. A Bayesian network model for user's preference estimation of personalized tv service. In *Advanced Communication Technology (ICACT), 2011 13th International Conference on*, pages 1555–1558. IEEE, 2011.

J. Lin. Divergence measures based on the shannon entropy. *IEEE Transactions on Information theory*, 37(1):145–151, 1991.

X. Liu. *Empirically modeling of audience behavior in the television industry*. PhD thesis, City University of Hong Kong, 2010.

K. Miyahara and M. Pazzani. Collaborative filtering with the simple Bayesian classifier. *PRICAI 2000 Topics in Artificial Intelligence*, pages 679–689, 2000.

K. Miyahara and M. J. Pazzani. Improvement of collaborative filtering with the simple Bayesian classifier. *Information Processing Society of Japan*, 43(11), 2002.

J. D. Mora. *Understanding the social structure of television audiences: Three essays*. PhD thesis, Business Administration: Faculty of Business Administration, 2010.

R. Mourad, C. Sinoquet, and P. Leray. A hierarchical Bayesian network approach for linkage disequilibrium modeling and data-dimensionality reduction prior to genome-wide association studies. *BMC bioinformatics*, 12(1):16, 2011.

F. Nojavan, S. S. Qian, and C. A. Stow. Comparative analysis of discretization methods in Bayesian networks. *Environmental Modelling & Software*, 87:64–71, 2017.

S. Park and J. K. Aggarwal. A hierarchical Bayesian network for event recognition of human actions and interactions. *Multimedia systems*, 10(2):164–179, 2004.

F. M. T. L. Pereira. *From user browsing behaviour to user demographics*. PhD thesis, 2015.

D.-T. Phan, P. Leray, and C. Sinoquet. Latent forests to model genetical data for the purpose of multilocus genome-wide association studies. which clustering should be chosen? In *International Joint Conference on Biomedical Engineering Systems and Technologies*, pages 169–189. Springer, 2015.

Y. Qiu, C.-J. Lin, Y.-C. Juan, W.-S. Chin, Y. Zhuang, B.-W. Yuan, M.-Y. Yang, and other contributors. See file AUTHORS for details. *recosystem: Recommender System using Matrix Factorization*, 2017. URL `https://CRAN.R-project.org/package=recosystem`. R package version 0.4.1.

R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2016. URL `https://www.R-project.org/`.

P. J. Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.

M. Scutari. Learning Bayesian networks with the bnlearn r package. *arXiv preprint arXiv:0908.3817*, 2009.

M. Scutari. Learning Bayesian networks with the bnlearn R package. *Journal of Statistical Software*, 35(3):1–22, 2010. doi: 10.18637/jss.v035.i03.

L. E. Sturlaugson and J. W. Sheppard. Principal component analysis preprocessing with Bayesian networks for battery capacity estimation. In *Instrumentation and Measurement Technology Conference (I2MTC), 2013 IEEE International*, pages 98–101. IEEE, 2013.

X. Su and T. M. Khoshgoftaar. Collaborative filtering for multi-class data using belief nets algorithms. In *Tools with Artificial Intelligence, 2006. ICTAI'06. 18th IEEE International Conference on*, pages 497–504. IEEE, 2006.

X. Su and T. M. Khoshgoftaar. A survey of collaborative filtering techniques. *Advances in artificial intelligence*, 2009:4, 2009.

J. Uebersax. Genetic counseling and cancer risk modeling: An application of bayes nets. *Marbella, Spain: Ravenpack International*, 2004.

B. van Dijk, J. van Rosmalen, and R. Paap. A bayesian approach to two-mode clustering. Technical report, 2009.

J. H. Ward Jr. Hierarchical grouping to optimize an objective function. *Journal of the American statistical association*, 58(301):236–244, 1963.

W. Xu, X. Liu, and Y. Gong. Document clustering based on non-negative matrix factorization. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 267–273. ACM, 2003.

S. Yaramakala and D. Margaritis. Speculative Markov blanket discovery for optimal feature selection. In *Data mining, fifth IEEE international conference on*, pages 4–pp. IEEE, 2005.

Y. Zuo, K. Yada, and E. Kita. *A Bayesian Network Approach for Predicting Purchase Behavior via Direct Observation of In-store Behavior*, pages 61–75. Springer International Publishing, Cham, 2015. ISBN 978-3-319-28379-1. doi: 10.1007/978-3-319-28379-1$_5$.