UNIVERSITAT DE BARCELONA

FUNDAMENTAL PRINCIPLES OF DATA SCIENCE MASTER'S THESIS

Parametric learning of probabilistic graphical models from multi-sourced data

Author:
David CATALÁN CEREZO

Supervisor:
Dr. Jerónimo HERNÁNDEZ-GONZÁLEZ
Dr. Aritz PÉREZ MARTÍNEZ

A thesis submitted in partial fulfillment of the requirements for the degree of MSc in Fundamental Principles of Data Science

in the

Facultat de Matemàtiques i Informàtica

Github repository of the project

June 30, 2023

UNIVERSITAT DE BARCELONA

Abstract

Facultat de Matemàtiques i Informàtica

MSc

Parametric learning of probabilistic graphical models from multi-sourced data

by David CATALÁN CEREZO

In Machine Learning, it is common to encounter scenarios where learning a model from a scarce dataset may not be feasible. In these cases, data from multiple different sources have to be collected. When data from multiple sources is distributed differently, the benefit of a bigger sample size trades off with the difficulty to model together data sampled from different distributions. A similar framework is presented in fairness analysis, where subpopulations defined by the protected attributes might show different underlying distributios. In this work, we study the use of hierarchical Bayesian methods to learn Bayesian network (BN) models from all the available data while being aware of the presence of unequally distributed data sources. We propose a variation of a previous hierarchical Bayesian approach for learning BN parameters which naturally accommodates into the framework of BNs. The comparison with the state-of-the-art methods is done in two dimensions: the amount of samples available to train a model, and the divergence of the underlying distribution of the different data sources. Experimental results suggest that our model is competitive when data is scarce and the multiple sources are distributed differently.

Acknowledgements

First of all I would like to express my most sincere gratitude to my two supervisors, Aritz Pérez Martínez and Jerónimo Hernández-González for their continuous support and guidance during this project. Without their help this project would have been impossible.

Second, to my family, my girlfriend and friends, who have been an essential support during the project and along the master.

Contents

Chapter 1

Introduction

Data collection is a fundamental aspect that poses challenges to the application of machine learning techniques in various domains. In cases where the phenomena under study are limited in prevalence, it may not be feasible to obtain a sufficient number of cases from a single source. As a result, researchers often rely on data collected from multiple sources. However, there is a trade-off with this approach: the data collected may not always be standardised or collected in the same way in different sources. As a result, it may not be distributed in the same way or may come from different data generation processes. This makes it difficult to model and analyse data from multiple sources. The fact that data comes from different data generation processes means that the distribution of the data may vary between sources. This leads to complexity in developing a unified model that can effectively capture the inherent relationships and patterns within the data. When enough data comes from each data source, we could model them separately in an efficient way. However, in sparse data scenarios, where we have access to a few data samples from each source, it might be unfeasible to model them separately. In those cases, it can be more efficient to model them together using an alternative technique that allows to combine data coming from different sources.

A domain that often faces these challenges is healthcare research, where conducting robust studies often requires collecting data from different hospitals due to the cost of data collection. Incorporating data from multiple healthcare centers introduces additional complexity due to differences in data collection protocols, patient populations and institutional practices. Consequently, it is crucial to explore methodologies that can effectively address the inherent challenges of modeling multi-source data in scarce data scenarios.

It is interesting to note that a potentially important application of the multi-source learning approach is fairness analysis. When the problem involves a sensible attribute, the classification model should treat different sub-populations in a similar way, so that the algorithm does not discriminate on the basis of that protected attribute. In this scenario, we would ideally want to extract information from the whole population but adapt at a local level to each subpopulation so that the model is as robust as possible with any subgroup. In the spirit of data coming from multiple sources (like different hospitals), the sensible attribute can analogously be viewed as an auxiliary variable defining the different sources (subpopulations).

In this work, our aim is to explore the use of probabilistic graphical models (PGMs), specifically Bayesian networks (BN), to derive valuable insights from multi-source data. For the sake of simplicity, in this study we assume that all the variables in the datasets are categorical. The learning process in BN consists of a structural learning stage and a parametric learning stage. By assuming that the structure of the graph is already known, we focus on the parametric learning stage in the context of multi-sourced data. Our objective is to use Bayesian estimation to learn a BN classifier that can adapt well to multi-sourced data scarcity scenarios. The Maximum Likelihood Estimation (MLE) of the parameters of a BN has a closed form. However, when there is data scarcity the MLE estimation can be problematic due to overfitting. Alternatively, the Bayesian estimation of the parameters deals with data scarcity by integrating prior knowledge about the parameters of interest into the modeling process. This is particularly beneficial when data are limited, as the prior can serve as a valuable source of information and regularization. For categorical data, the most used prior is the Dirichlet distribution due to the conjugacy of the Dirichlet and the multinomial distributions. The conjugacy ensures that the maximum a posteriori is also a Dirichlet distribution, and allows to obtain a closed-form solution of the parameters of the categorical. Henceforth, the choice of the Dirichlet distribution as a prior ensures that we can do Bayesian estimation in an efficient way, as discussed in (Bernardo, 1994).

When dealing with multi-sourced data that may come from different data generation processes, we would ideally want a predictive model that learns from the full distribution of the whole population, but also takes into account the presence of data having different distributions, so that it can adapt locally to each subpopulation. The main methodologies that deal with this problem when learning BNs are the hierarchical-Dirichlet models proposed by Azzimonti, Corani, and Zaffalon (2019). The presented methodologies define a variation of the multinomial-Dirichlet approach in which there is a hierarchical hyperprior that performs together the estimation of the probability distributions of the same CPD, in the case of a single data source, or the estimation of the analogous distributions from different data sources, in the case of multiple data sets. According to (Azzimonti, Corani, and Zaffalon, 2019), these kinds of models will render better parameter estimations compared to other models in data scarce multi-source scenarios. We also hypothesize that they will give better parameter estimations compared to other models when data are differently distributed in the different sources. The model will be able to capture the differences in the underlying distributions of the sources while, at the same time, it will use statistical power from the other sources. This idea is referred to as “borrowing statistical strength” by Azzimonti, Corani, and Zaffalon (2019).

This project deals with the problem of the parametric learning of BNs from multi-sourced scarce data in two ways. First, we propose a method to learn the parameters of a BN from multi-sourced data complementary to the ones proposed in (Azzimonti, Corani, and Zaffalon, 2019). Second, we compare this new method with the state-of-the-art methods in two different axes: as data become scarce, and as different sources diverge in distribution. To achieve this goal, we assume that the structure of the BN is known and focus on the comparison of different methods for parameter estimation given a structure. The different methodologies are then compared in terms of the accuracy of the learned classifiers. Additionally, we design a synthetic data generation process that allows us to test these methods in different experimental conditions. We basically control for data sample size and divergence of the underlying distribution of the multiple sources.

The project is structured as follows: Chapter 2 reviews related works in the literature, Chapter 3 defines the methodology used, Chapter 4 presents the experimental setting where we explore our hypotheses, and finally Chapter 5 concludes our findings.

Chapter 2

Related Work

In this project, we treat the case of learning with data coming from multiple sources, assuming that all of them provide the same set of variables. These can potentially be distributed differently. Estimating models from multi-sourced datasets and overcoming the challenges associated with heterogeneity in data distribution have been topics of considerable interest in various research areas. For instance, Strømsø and Bråten (2009) evaluate classifiers' accuracies in the context of multi-sourced datasets.

In our project, we focus on learning Bayesian Network models for three reasons. First, they allow to obtain the value of the class we want to predict given the features, while reducing the number of parameters that have to be estimated. Second, they can be used as generative models, by modeling the joint distribution of the class and the features they can generate new instances. Bielza and Larranaga (2014) discuss different types of Bayesian Networks and their possible uses as generator models. Thirdly, in the context of databases composed of categorical variables, Bayesian Networks are one of the most used models for classification. Unfortunately, there are not so many works that make use of Bayesian Networks to learn from multi-sourced data.

Learning a general model from multi-sourced data has received considerable attention in the context of learning from distributed data, where data are distributed across a set of devices and aggregated in a central server. In the context of distributed data, McMahan et al. (2017) propose an estimation method called federated learning, where for each device (or source) a model is trained locally and only the model updates or aggregated model parameters are shared with the central server in a way that preserves privacy. Konstantinov and Lampert (2022) focus on fairness in federated learning. While this method is very useful in several applications, such as fairness awareness, it does not exactly serve our goal, where we want to learn a general model and then adapt it at the local level.

In the context of Bayesian networks learned from multi-sourced datasets, some works have focused on structure learning and others on parametric learning. For the former, Tillman (2009) use statistical tests to discover the general structure of a directed acyclic graph from IID data coming from different sources. Tillman and Spirtes (2011) extend the method to handle non-IID data by obtaining statistics for each data source and then obtaining a general common graph from them. Other works, such as (Oates et al., 2014) and (Oates et al., 2016), first pool the data and then learn a common structure. Azzimonti, Corani, and Scutari (2022) build on their previous work on parameter estimation using Hierarchical models (Azzimonti, Corani, and Zaffalon, 2019) and define a new scoring method, called BHD, for structural learning. It is a modification of the Bayesian Dirichlet equivalent uniform score that takes into account the Hierarchical structure of the model to learn a common structure from multi-sourced data. In our case, we assume that the structure of the network is already known to focus on the comparison of parametric learning methods.

Apart from hierarchical models, another interesting methodology proposed by Geiger and Heckerman (1996), called multinets, can be used to learn BN structures from multi-sourced data. This methodology, instead of using a single network structure, proposes a Hierarchical Bayesian network structure that allows for the consideration of a specific structure for each domain. Therefore, unlike hierarchical multinomial Dirichlet, they learn more than one structure and do not share conditional probability tables, as hierarchical models do by using the hyperprior. In our case, we focus on learning a single common structure (which we take as given) and estimate the parameters of the model being aware of the presence of multiple sources.

The literature on parameter learning of BNs from multi-sourced data is scarce. To the best of our knowledge, the first contribution to parameter learning that can be used given any Bayesian network structure was presented by Azzimonti, Corani, and Zaffalon (2019). They propose to add a hyperprior to the classical multinomial-Dirichlet so that distributions within a conditional probability table are linked by the hyperprior and they are drawn from a mixture of Dirichlet distributions. This approach leads to an exchange of information between all the available data, which they refer to as the ability of the distributions to "borrow statistical strength" from each other. This term is originally coined in (Teh et al., 2004) in the context of Hierarchical Dirichlet Processes. This concept can be adapted to learning from multi-sourced data. In this case, the distributions of the analogous conditional distributions obtained from the different sources are linked between them. The conditional distributions obtained from different sources should resemble each other because they represent the same probabilistic relationships. The only thing that differentiates them is that they pertain to different sub-populations defined by the auxiliary variable. A difficulty encountered in Hierarchical Bayesian Dirichlet estimation is that the posterior cannot be estimated directly due to the lack of a closed form and the costly exact inference process. It must be approximated using Markov Chain Monte Carlo (MCMC) methods or variational inference. Azzimonti, Corani, and Zaffalon (2019) propose their own method for performing efficient variational inference when applied to these hierarchical models.

Chapter 3

Methods

In this project, we build on top of the two different methodologies for parameter estimation of Bayesian networks that use a hierarchical multinomial-Dirichlet Bayesian network proposed by Azzimonti, Corani, and Zaffalon (2019) and propose a novel variation of these methodologies.

3.1 Probabilistic Graphical Models and Bayesian Networks

PGMs represent a factorization of a joint probability distribution according to a graph. The graph allows for reading the conditional independences that the encoded distribution fulfills. Thanks to these conditional independences, the joint distribution can be represented with a reduced number of parameters. In this work, we focus on Bayesian Networks, which are graphical models with a DAG structure that factorize the distribution into a product of conditional probability distributions.

Directed acyclic graphs: A DAG G is a pair (V, E), where $V = (v_1, ..., v_n)$ represents the set of vertices and E is a set of directed edges, where each of them is a pair (u, v) with $u \neq v$, meaning that $u \rightarrow v$. A DAG has no directed cycles.

Bayesian Networks: A Bayesian Network M is conformed by a DAG, G, and a set of parameters $\Theta = (\theta_{X_1}, ..., \theta_{X_n})$. From here on, we denote the set of random variables as $X = (X_1, ..., X_n)$, where each one has its own set of possible values $x_i \in \mathcal{X}_i$. Also denote $x = (x_1, ..., x_n)$ the instantiations of X, where $x \in \mathcal{X} = \mathcal{X}_1 * ... * \mathcal{X}_n$ and $(*)$ represents the Cartesian product. In this work, we focus on categorical variables, where $\mathcal{X}_i = (1, ..., r_i)$. We also denote the set of parents $Pa = (Pa_1, ..., Pa_n)$ and $pa = (pa_1, ..., pa_n)$ its possible values. The joint probability distribution of a Bayesian Network factorizes as:

$$p_M(X) = \prod_{i=1}^{n} p(x_i | pa_i, \theta_{X_i}) \tag{3.1}$$

Each X_i is associated with a node v_i in the graph G and the set Pa_i of parent variables of X_i are those connected through directed edges with the node v_i ($\forall v_j \in Pa_i, (v_j, v_i) \in E$). Each node X_i has an associated conditional probability distribution given its parents in the graph: $p(X_i | Pa_i, \theta_{X_i})$. The set of parameters Θ represents the parameters of all these conditional distributions. The factorization as a product of conditional probabilities allows for the reduction of the number of parameters that have to be estimated to represent the joint distribution of the data. The local distribution associated with each node only depends on the configuration of its parents.

Learning a Bayesian Network from data has two main steps: Structural learning and parametric learning.

Structural Learning consists of learning the graph of the BN from data. There are two main methodologies in order to find the best structure:

- **Constrained based methods**: they involve statistical independence tests that try to identify the structure that better represents the conditional independencies observed in the data.
- **Score-based methods**: they evaluate different network structures based on a scoring metric that measures its goodness-of-fit. Popular greedy approaches use an iterative method that evaluates the score of the structure once we add or subtract edges. Amongst the penalized scores, there is the family of Bayesian scores. One of the most popular scores is the Bayesian Dirichlet Equivalent Uniform score.

In this work, we assume that the structure of the model is given and focus on parametric learning. For the sake of simplicity, we will focus on Naive Bayes (NB) and Tree-augmented Naive Bayes (TAN) structures, two types of graphs of increasing complexity that are popular in supervised classification.

- **Naive Bayes (NB)**: it assumes conditional independence among the descriptive features given the class variable, which is the only parent. Figure 3.1 shows a general Naive Bayes structure:

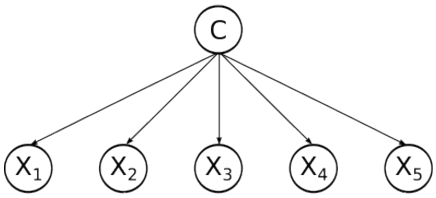


FIGURE 3.1: NB structure

- **Tree-Augmented Naive Bayes (TAN)** is an extension of the Naive Bayes that incorporates a tree structure to capture dependencies among the descriptive features. The class variable is still the root node. A TAN structure is shown in Figure 3.2:

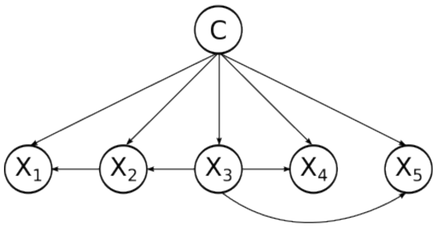


FIGURE 3.2: A TAN structure

Parametric Learning consists of finding the parameters Θ of the factorization defined by the structure. The parameters can be either estimated in closed form with

MLE or or Bayesian inference with a conjugate prior. Section 3.2 describes the two main methods used for parametric learning.

3.2 Parametric Learning with categorical data

As noted in section 3.1 parametric learning can be done either using Maximum Likelihood Estimation (MLE) or Bayesian Estimation. In this work, we assume that variables are categorical. Here, we present MLE and Bayesian estimation for categorical data.

3.2.1 Maximum Likelihood Estimation (MLE)

MLE is a statistical method used to estimate the parameters of a probability distribution based on observed data. The likelihood function $L(\theta|D)$ is defined as the probability of the data given the model with parameters $\Theta = (\theta_{X_1}, ..., \theta_{X_n})$. In BNs, the likelihood is given as:

$$L(\Theta|D) = p(D|\Theta) = \prod_{s=1}^{N}\prod_{i=1}^{n} p(x_i^{(s)}|pa_i^{(s)}, \theta_{X_i}) \tag{3.2}$$

Then, the log-likelihood function $l(\Theta; D)$ is:

$$l(\Theta|D) = \sum_{s=1}^{N}\sum_{i=1}^{n} log(p(x_i^{(s)}|pa_i^{(s)}, \theta_{X_i})) \tag{3.3}$$

Taking partial derivatives with respect to each θ_{X_i} and setting them to zero we can find the estimates of θ_{X_i} that maximize the log-likelihood function. That is:

$$\theta_i^{MLE} = \underset{\theta_{X_i}}{\operatorname{argmax}}\, l(\Theta|D) \tag{3.4}$$

where Θ are the parameters of all the CPDs of the Bayesian Network, including θ_{X_i}. Using a general notation, denote N_{xy} the counts in the data for $X = x$ and $Y = y$. Denote N_y the counts of $Y = y$ ($N_y = \sum_x N_{xy}$). The MLE of the parameter $\theta_{x|y}$ is:

$$\theta_{x|y}^{MLE} = \frac{N_{xy}}{N_y} \tag{3.5}$$

and the MLE of the distribution $P(X|Y = y)$ is:

$$\theta_{X|y}^{MLE} = (\theta_{x|y}^{MLE})_{x\in\mathcal{X}} \tag{3.6}$$

Therefore, to obtain the parameters using MLE we only need the counts for each variable given its parents.

3.2.2 Bayesian Estimation: Dirichlet-Multinomial conjugate prior (BMD)

Using a conjugate prior allows for obtaining a maximum a posteriori estimate in closed form. As it is well known, the Dirichlet prior is a conjugate prior of the multinomial distribution. Henceforth, using a Dirichlet prior ensures that the posterior distribution will also be a Dirichlet distribution if we model the parameters $\Theta_{X|pa}$ with a multinomial distribution. Formally, if the prior is:

$$\theta_{X_i|pa}|\alpha \sim Dirichlet(\alpha_1, ..., \alpha_r) \tag{3.7}$$

for $pa \in \mathcal{P}_i$ represents the set of all possible values for the parent variables, Pa_i, and the data is modeled with a Categorical distribution as:

$$X_i|pa, \theta_{X_i|pa} \sim Categorical(\theta_{X_i|pa}) \tag{3.8}$$

the maximum a posteriori is also a Dirichlet distribution as:

$$\theta_{X_i|pa}|\alpha \sim Dirichlet(\alpha_1 + N_{x_1 pa}, ..., \alpha_r + N_{x_r pa}) \tag{3.9}$$

where $N_{x_l pa}$ are the counts of $X_i = x_l$ and $Pa_i = pa$ as defined in Section 3.2.1. These counts are also known as sufficient statistics. The posterior of $\theta_{X_i|pa}$ has an expected value given by:

$$E[\theta_{X_i|pa}] = \left(\frac{\alpha_l + N_{x_l pa}}{\alpha + N_{pa}}\right)_{l\in\{1,...,r\}} \tag{3.10}$$

where $\alpha = \sum_{l=1}^{r} \alpha_l$ is known as the equivalent sample size and $N_{pa} = \sum_{l=1}^{r} N_{x_l pa}$. Note that all this procedure is carried out given a fixed value $pa \in \mathcal{P}_i$ for the parent variables. It actually holds for any $pa \in \mathcal{P}_i$.

3.3 Hierarchical Bayesian Estimation

The Hierarchical Dirichlet-Multinomial models proposed by Azzimonti, Corani, and Zaffalon (2019) can be used to pool data from distributions in the same CPD (first approach), but they have been proposed to learn from multi-sourced data too (second approach). In both methods they introduce a latent vector $\alpha^0 = (\alpha_1^0, ..., \alpha_r^0)$ as an hyperprior that models the α present in the Dirichlet prior on the Dirichlet-Multinomial. however,due to this hyperprior a closed form solution for the maximum a posteriori cannot be obtained, instead, it has to be approximated using Variational inference. Before giving an explanation of the different hierarchical models we give a brief introdduction to Variational inference, as it is a key part for the estimation procedure of hierarchical models.

Variational Inference:

Variational inference (VI) is a technique used in probabilistic modeling and Bayesian statistics to approximate complex probability distributions. It is used to compute the posterior distribution of the latent variables given the observed data when the exact posterior distribution is intractable. Variational inference offers an alternative approach by approximating the true posterior distribution with a simpler distribution from a predefined family of distributions.

In the context of HDM models, denote the posterior distribution as $p(\Theta, \alpha|D)$ where D denotes our dataset. Denote $q(\Theta, \alpha)$ a family of distributions used to approximate $p(\Theta, \alpha|D)$. Our goal is to find the optimal parameters ϕ^* for the variational distribution that minimizes the KL divergence between the variational distribution $q(\Theta, \alpha)$ and the true posterior $p(\Theta, \alpha|D)$, that is:

$$\phi^* = \underset{\phi}{\text{argmin}}\, KL(q_\phi(\Theta, \alpha)||p(\Theta, \alpha|D)) \tag{3.11}$$

where ϕ is the parametrization of the approximation q. The KL divergence measures the dissimilarity between distributions by comparing the relative entropies. Formally:

$$KL(q_\phi(\Theta,\alpha)||p(\Theta,\alpha|D)) = \mathbb{E}_{q_\phi(\Theta,\alpha)}[\log q_\phi(\Theta,\alpha)] - \mathbb{E}_{q_\phi(\Theta,\alpha)}[\log p(\Theta,\alpha|D)] \quad (3.12)$$

After some manipulations, it can be shown that minimizing the KL divergence is equivalent to maximizing the following expression, known as the Evidence Lower Bound (*ELBO*):

$$ELBO(q_\phi(\Theta,\alpha)) = \mathbb{E}_{q_\phi}[\log p(\Theta,\alpha,D)] - \mathbb{E}_{q_\phi}[\log q_\phi(\Theta,\alpha)] \quad (3.13)$$

Therefore, in order to obtain estimates for the posterior of the HDM model, we can maximize the *ELBO* with respect to the distribution parameters ϕ:

$$\phi^* = \operatorname*{argmax}_\phi ELBO(q_\phi(\Theta,\alpha)) \quad (3.14)$$

In this work, we use the Automatic Differentiation Variational Inference algorithm (ADVI) provided by Carpenter et al. (2017) in the STAN interface. The ADVI algorithm combines automatic differentiation techniques with variational inference to efficiently optimize the variational distribution. It does so by optimizing the *ELBO* using stochastic gradient ascent.

3.3.1 Hierarchical Dirichlet-Multinomial: single dataset (HDS)

As explained in section 3.3 the HDS presented in (Azzimonti, Corani, and Zaffalon, 2019) has the same form as the Dirichlet-Multinomial except that it assumes that there is also a Dirichlet hyper-prior with a latent vector of parameters $\alpha^0 = (\alpha^0_1, ..., \alpha^0_r)$ that models the α present in the Dirichlet prior. That is:

$$\alpha|s,\alpha^0 \sim s * Dirichlet(\alpha^0_1, ..., \alpha^0_r) \quad (3.15)$$

$$\theta_{X_i|pa}|\alpha \sim Dirichlet(\alpha_1, ..., \alpha_r), \forall pa \in \mathcal{P}_i \quad (3.16)$$

$$X_i|pa, \theta_{X_i|pa} \sim Categorical(\theta_{X_i|pa}) \quad (3.17)$$

This methodology assumes that all the data comes from the same underlying distribution. Hence, we apply the hierarchical model to the whole dataset in order to learn the parameters of a single Bayesian Network model. The hyperparameter α_0 serves as a hyperprior of the Dirichlet distribution. It is used to make all the conditional probability distributions within the same conditional probability table correlated, and thus all of them will tend toward each other.

This model assumes that the priors of all the distributions of the CPD $P(X_i|Pa_i)$ (for different values $pa \in \mathcal{P}_i$) come from the same hyper-prior α_0. The purpose of the vector α is to link the estimation procedure for the different distributions in the same conditional probability table. In fact, the α_0 hyperprior serves as a prior that shrinks the parameters of the different distributions towards each other, making them closer to the mean between the distributions. The difficulty of the HDM is that due to the correlation induced by the hyperprior α_0, the posterior does not have a closed-form solution, as it has the classical Multinomial-Dirichlet model and cannot be estimated directly. However, this posterior can be approximated by means of approximate inference techniques, such as Variational Inference.

The generative model is represented as a factor graph in Figure 3.3:

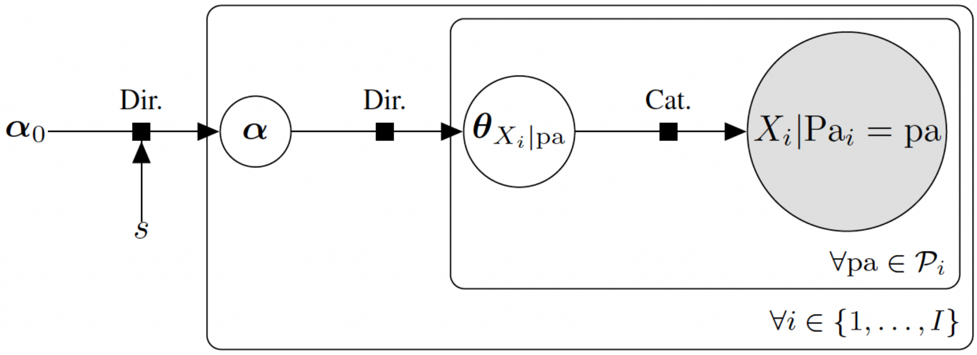

α_0
Dir.
s
α
Dir.
$\theta_{X_i|\mathrm{pa}}$
Cat.
$X_i|\mathrm{Pa}_i = \mathrm{pa}$
$\forall \mathrm{pa} \in \mathcal{P}_i$
$\forall i \in \{1, \dots, I\}$

FIGURE 3.3: Factor graph describing the hierarchical model for single-soruce data (HDS). Source Azzimonti, Corani, and Zaffalon, 2019

Implementation

Given a DAG structure, we apply the following procedure for each node X_i and its parents Pa_i. We first compute the sufficient statistic N_{xpa} and N_{pa} for all possible combinations $(x, pa) \in (\mathcal{X}_i, \mathcal{P}_i)$ as explained in Section 3.2.1 and apply Laplace smoothing, which adds artificial pseudo-counts to all the possible combinations, so that no combination has a probability of exactly zero. The mentioned statistics are the inputs of the variational inference algorithm, which returns the posterior distribution of the CPDs one node at a time. Then, we reconstruct the CPD of each node in our model with the new parameters provided by the VI method. Pseudocode in Table 3.1 summarizes the process.

```
HDS( $G$: structure)
for each node $X_i$ in model $G$:
    Compute $N_{xpa}$    $\forall x \in \mathcal{X}_i, pa \in \mathcal{P}_i$
    $N_{xpa} = N_{xpa} + 1$    $\forall x \in \mathcal{X}_i, pa \in \mathcal{P}_i$
    $N_{pa} = \sum_{x \in \mathcal{X}_i} N_{xpa}$    $\forall pa \in \mathcal{P}_i$
    $\theta_{X_i|Pa_i}$ = STAN-ADVI($\{N_{xpa}\}, \{N_{pa}\}$) (*Approximate posterior's mean*)
    Update CPD $P(X_i|Pa_i)$ with $\theta_{X_i|Pa_i}$
```

TABLE 3.1: HDS procedure to learn a Bayesian Network given a structure G

3.3.2 Hierarchical Dirichlet-multinomial multi data: joint states (HDMJS)

This methodology does assume the multi-sourced data scenario. A different model is learned for each subpopulation (defined by means of an auxiliary variable F), that is, each CPD is learned separately for each subpopulation or source. However, it is done in a way that the CPDs from each subpopulation are linked through a hyperprior, which allows the CPDs to "Borrow Statistical Strength" from each other.

The model in subsection 3.3.1 can be modified to estimate parameters from multiple datasets by liking the CPDs with an auxiliary variable F.

The generative model is based on on the transformation $X_i' = (X_i, Pa_i)$, with r' possible values as $\mathcal{X}_i * \mathcal{P}_i$. The generative process in this case is as follows:

$$\alpha | s, \alpha^0 \sim s * Dirichlet(\alpha_1^0, ..., \alpha_{r'}^0) \tag{3.18}$$

$$\theta_{X_i'|f} | \alpha \sim Dirichlet(\alpha_1, ..., \alpha_{r'}), \forall f \in \mathcal{F} \tag{3.19}$$

$$X_i' | f, \theta_{X_i'|f} \sim Categorical(\theta_{X_i'|f}) \tag{3.20}$$

where $\mathcal{F}$ includes all the possible values of the auxiliary variable F that identifies the sources. The auxiliary variable allow to link the columns of the analogous joint distribution tables. That is, the probabilities $p(X_i = x_i, Pa_i = pa_i)^f$ are linked between different datasets by means of f. The modified model presented in (Azzimonti, Corani, and Zaffalon, 2019) is presented as a factor graph in Figure 3.4:

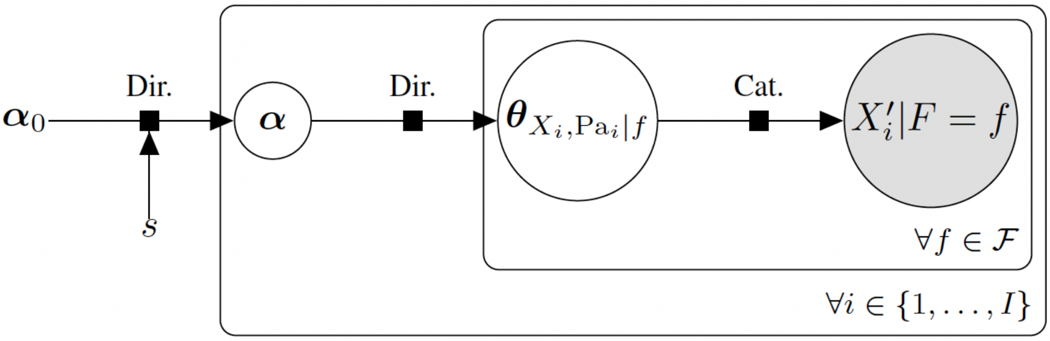

α_0
Dir.
s
α
Dir.
$\theta_{X_i,\mathrm{Pa}_i|f}$
Cat.
$X_i'|F = f$
$\forall f \in \mathcal{F}$
$\forall i \in \{1, \ldots, I\}$

FIGURE 3.4: Factor graph describing the hierarchical model for multi-sourced data (HDMJS). Source Azzimonti, Corani, and Zaffalon, 2019

The hyperprior α_0 now links the distributions analogous distributions from tables pertaining to the same variable but from differen sources.

Note the difference with the methodology presented in the 3.3.1. Before, in a single-source scenario, the conditional distributions of the same CPD were linked between them with the Hierarchical model. The assumption was that the distributions of the same CPD should be similar to each other. In this section, the Hierarchical model links the conditional probability distributions for the same node across models (sources). This induces a correlation between tables and makes the analogous distributions tend toward each other. This assumes that, even across the different sources, the CPD of the same node should be similar in the different models.

Implementation:

The VI estimation procedure, in this case, receives the joint states of all the possible combinations of the node X_i with its parents Pa_i, conditional on the auxiliary variable F. We first create a new variable X_i' that combines X_i with its parents Pa_i, with possible values $x' \in \mathcal{X}_i' = \mathcal{X}_i * \mathcal{P}_i$. Then, the sufficient statistic $N_{x'f}$ and N_f

for all possible combinations $(x', f) \in (\mathcal{X}_i', \mathcal{F})$ as explained in Section 3.2.1 and apply Laplace smoothing. These two elements are the inputs for STAN's ADVI algorithm. The output of the VI algorithm is the posterior $P(X'|f)$. We need to divide by the marginal probability $P(Pa_i)$ to obtain the CPD $P(X_i|Pa_i, f)$. Finally, the CPD $p(X_i|Pa_i, f)$ is recovered for each configuration of the auxiliary variable F on the corresponding model. These steps are summarised in Table 3.2.

```
HDMJS ($G$: structure, $F$: auxiliary variable)
for each node $X_i$ in model $G$:
    Define $X_i' = (X_i, Pa_i)$ with $\mathcal{X}_i' = \mathcal{X}_i * \mathcal{P}_i$
    Compute $N_{x'f} \quad \forall x' \in \mathcal{X}_i', f \in \mathcal{F}$
    $N_{x'f} = N_{x'f} + 1 \quad \forall x' \in \mathcal{X}_i', f \in \mathcal{F}$
    $N_f = \sum_{x' \in \mathcal{X}_i'} N_{x'pa} \quad \forall f \in \mathcal{F}$
    $\theta_{X_i'|F}$ = STAN-ADVI($\{N_{x'f}\}, \{N_f\}$)
    Compute $\theta_{X_i|Pa_i,F} = \frac{\theta_{X_i'|F}}{\sum_{x \in \mathcal{X}_i} \theta_{x,Pa_i|F}}$
    Update CPT $P(X_i|Pa_i)$ for model $F = f$ with $\Theta_{X_i|Pa_i,F=f}$
```

TABLE 3.2: HDMJS procedure to learn a Bayesian Network given a structure G

3.3.3 Hierarchical Model multi data: conditional states (HDMCS*)

The following methodology is our proposal, which was motivated by the observation that the previous method, proposed by Azzimonti, Corani, and Zaffalon (2019), might be failing to incorporate the auxiliary variable F as a parent in a natural way to the BN structure.

The methodology we propose is also expressed as a factor graph in Figure 3.5. The difference with the later method is that we use the auxiliary variable F as the parent of all the variables on the original BN graph, which allows us to model the conditional distribution $p(X_i|Pa_i, F)$.

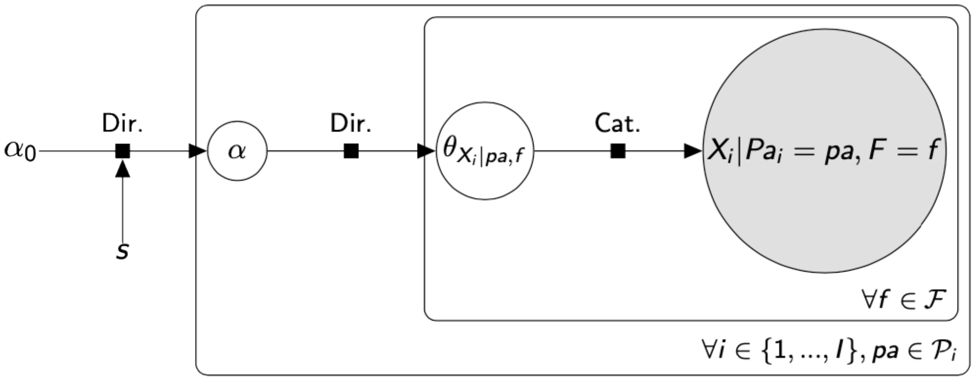

α_0
Dir.
s
α
Dir.
$\theta_{X_i|pa,f}$
Cat.
$X_i|Pa_i = pa, F = f$
$\forall f \in \mathcal{F}$
$\forall i \in \{1, ..., I\}, pa \in \mathcal{P}_i$

FIGURE 3.5: Factor graph: hierarchical model: conditional states (HDMCS*)

Our methodology applies the hierarchical model in the multi-sourced data scenario and explicitly adds the auxiliary variable F as a parent of all the other variables. We do not use the transformation $Z_i = X_i * Pa_i$ as Azzimonti, Corani, and Zaffalon (2019) to adapt the hierarchical model of Section 3.3.1 to multi-sourced data. In practice, this transformation joins the estimation of the parameters of all the distributions of the node X_i given any value of the parents Pa_i and the auxiliary F. It requires the subsequent use of the Bayes rule to obtain the conditional distributions of the BN. On the contrary, our methodology allows us to express the dependencies of the variables in a Bayesian Network structure, using explicitly F as a parent of all the variables.

Our methodology is actually more similar to that of Section 3.3.1, but the set of distributions that are affected by a common hyper-prior are those with the same parent values $pa \in \mathcal{P}_i$ but different source $f \in \mathcal{F}$. The generative process is:

$$\alpha|s, \alpha^0 \sim s * Dirichlet(\alpha_1^0, ..., \alpha_r^0) \tag{3.21}$$

$$\theta_{X_i|pa,f}|\alpha \sim Dirichlet(\alpha_1, ..., \alpha_r), \forall f \in \mathcal{F} \tag{3.22}$$

$$X_i|pa, f, \theta_{X_i|pa,f} \sim Categorical(\theta_{X_i|pa,f}) \tag{3.23}$$

Note that, as in Section 3.2.2, this procedure is carried out given a fixed variable X_i and value $pa \in \mathcal{P}_i$ for the parent variables. It actually holds for any X_i and $pa \in \mathcal{P}_i$.

Implementation:

The estimation of the CPDs of X_i is similar to that of the hierarchical model of Section 3.3.3 assuming that F is an extra parent, for each parent configuration $pa \in \mathcal{P}_i$. The steps are summarised in Table 3.3.

```
HDMCS* ($G$: structure)
Add $F$ as parent of each node.
for each node $X_i$ in model $G$:
  for each configuration $pa$ in $\mathcal{P}_i$:
    Compute $N_{xpaf} \quad \forall x \in \mathcal{X}_i, f \in \mathcal{F}$
    $N_{xpaf} = N_{xpaf} + 1 \quad \forall x \in \mathcal{X}_i, f \in \mathcal{F}$
    $N_{paf} = \sum_{x \in \mathcal{X}_i} N_{xpaf} \quad \forall f \in \mathcal{F}$
    $\theta_{X_i|pa,F}$ = STAN-ADVI($\{N_{xpaf}\}$, $\{N_{paf}\}$) (Approximate posterior's mean)
    Update distribution $P(X_i|pa)$ for model $F = f$ with $\Theta_{X_i|pa,F=f}$
```

TABLE 3.3: HDMCS* procedure to learn a Bayesian Network given a structure G

Chapter 4

Experiments

In this section, we carry out the experimental comparison of the methods presented in the previous chapter. We describe the experimental setting, including the used datasets and the synthetic data generation to control on two dimensions: the number of samples available for training and the divergence between the underlying distributions of the different subpopulations. Finally, we present and discuss our results.

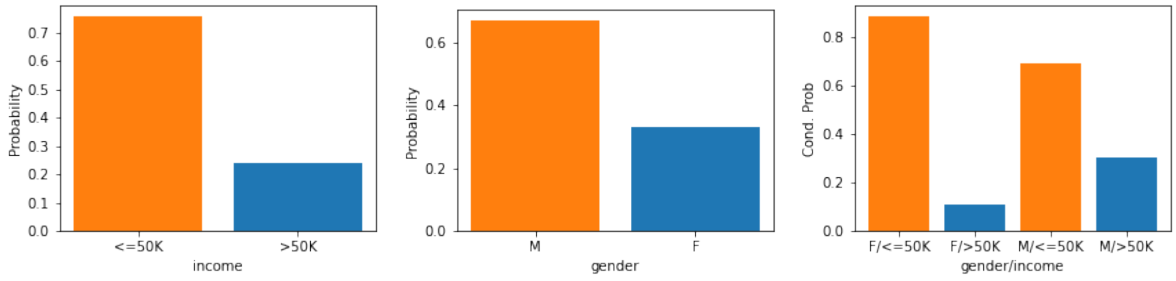
4.1 Experimental Setting

We use two real-world datasets, Adult and Diabetes, which are popular in the fairness analysis literature. We transform them synthetically to create the experimental conditions that we want to test.

Adult Dataset

The Adult dataset from the UCI Machine Learning Repository is a widely used dataset for classification tasks. It contains information about individuals from the 1994 US Census, and the goal is to predict whether an individual's income exceeds $50,000$$ per year or not based on their demographic and employment-related features. Table A.1 in Appendix A lists all the variables of the original dataset considered for this study. We use simple imputation to fill in missing values: for categorical variables, we use the mode. This decision is motivated by our intention of preserving the sample size, although it may result in a loss of variability.

The class variable we want to predict is "income" and the sensible attribute that we have used as the auxiliary variable F is the variable "gender". However, there are other potential sensible attributes that we could have used, for instance: race, native-country or education could have been potentially considered as protected attributes. The distributions of the class variable and the auxiliary variable, as well as the distribution of their combination, are shown in Figure 4.1. Both income and gender are unbalanced. Furthermore, when we look at the distribution of income given gender we see how the proportion of females with lower income with respect to the whole sample of females is much lower than that of men.



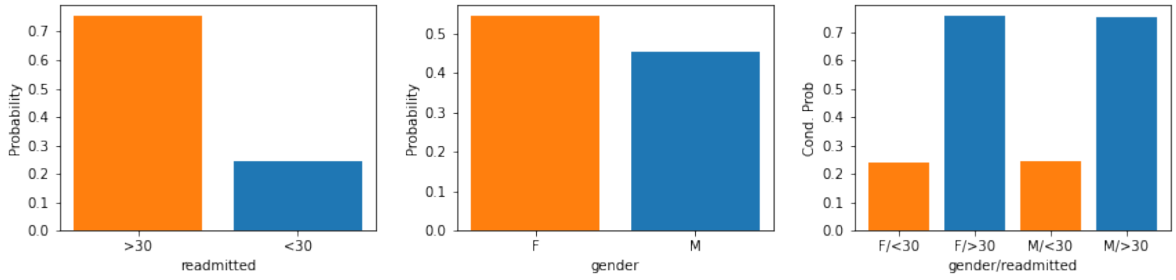
(A) (B) (C)

FIGURE 4.1: Distribution of Class variable (A), auxiliary variable (B) and combination of both variables (C) in the Adult dataset

Diabetes Dataset

The Diabetes dataset from the UCI Machine Learning Repository contains medical information about patients from 130 different hospitals. The classification task is to predict whether a patient will be readmitted within 30 days or if it will be readmitted in more than 30 days. Only the readmitted patients are kept. From among the 50 attributes contained on the original dataset, we keep the variables selected by Le Quy et al. (2022), which are summarized in Table A.2 from Appendix A.

We show the distribution of the class variable, "readmitted", and the auxiliary variable, "gender" in Figure 4.2. Both the class variable and the auxiliary variable are unbalanced, although the auxiliary variable is not as highly unbalanced as in the adult dataset. The proportion people readmitted within 30 days does not change much when we condition on gender.



(A) (B) (C)

FIGURE 4.2: Distribution of Class variable (A), auxiliary variable (B) and combination of both variables (C) in the Diabetes dataset

4.2 Simulation of multi-sourced data

Our experiments explore two dimensions: sample size and distribution divergence.

4.2.1 Simulation of different sample sizes

To simulate datasets with different numbers of samples, we simply generate random stratified subsamples. Stratification is carried out regarding both the class and the auxiliary variables.

4.2.2 Simulation of multi-sourced data with diverging distributions

In order to test the different estimation methods in datasets of the multi-source scenario, we have designed a simulation procedure to artificially generate data with the required characteristics. Basically, we exploit the ability of PGMs to serve as generative models. We first fit a model with the specified characteristics and then we use a sampling method like Forward Sampling to generate new data that reflects the probabilistic relationships specified by the model.

The procedure works as follows. We first pick a real-world dataset, a variable F as an auxiliary variable that defines subsamples of a dataset (or multiple-sources), and a fixed DAG structure for the model. We estimate the parameters of the model with pooled data, that is, all the data from all the multiple sources. We also estimate the parameters for each source (from its corresponding data subset) separately.

For the sake of simplicity, let us assume that there are only two sources, $F \in \{f_0, f_1\}$. Thus, we will obtain three CPDs for each node X_i: $\theta^{f_0}_{X_i}$, $\theta^{f_1}_{X_i}$ and $\theta^{Pooled}_{X_i}$. Then, for each node, we calculate the deviation $\theta^{f_1}_{X_i} - \theta^{Pooled}_{X_i}$. The sign of the element in this vector with the largest absolute difference $|\theta^{diff}_{X_i=x}|$ is used as the direction of the divergence, $s = \text{sign}(\max_x |\theta^{diff}_{X_i=x}|)$. Consequently, the divergence of $\theta^{f_0}_{X_i}$ from $\theta^{Pooled}_{X_i}$ will go in the opposite direction, $-s$, as the pooled estimates must lie between both.

The next step is to make the parameters of the model with $F = f_1$ diverge as much as we like. Let us define $\delta \in [0, 1]$ to denote the "amount of divergence". Given the original value, $\theta^0 = \theta^{f_1}_{X_i=x}$, the new value of the parameter is:

$$\begin{aligned} \theta^* &= (1-\theta^0) * \delta + \theta^0 && \text{if } s > 0 && (\theta^* \rightarrow 1) \\ \theta^* &= \theta^0 - \theta^0 * \delta && \text{if } s < 0 && (\theta^0 \rightarrow 0) \end{aligned}$$

However, when we change the value of some $\theta^*_{X_i=x}$, the other values $\theta_{X_i=y}$ in the same probability distribution have to be changed accordingly so that the probabilities add up to 1. We assume proportional covariation (Ballester-Ripoll and Leonelli, 2022), that is, the rest of values in the distribution change proportionally to the change of θ^* but in the opposite direction:

$$\theta^*_{X_i=y}(\theta^*_{X_i=x}) = \frac{1-\theta^*_{X_i=x}}{1-\theta^0_{X_i=x}} \theta^0_{X_i=y}, \forall y \in \mathcal{X}_i), y \neq x \tag{4.1}$$

We apply these steps for all the CPDs in the models, and obtain a BN model for each $F = f$, all with the same structure and diverging from each other according to δ.

The final step is to generate the data using each of the models. We use the standard Forward sampling algorithm for BNs to generate samples while maintaining the original proportions of samples according to the auxiliary attribute, F.

4.3 Evaluation

In these experiments, we measure and compare the performance of the methods (MLE,BMD,HDS,HDMJS and HDMCS*) in terms of accuracy. In the methods that involve, in practice, the use of different models for each source f, the class prediction for each test sample is carried out using the corresponding model. The results of each model are pooled to compute the global accuracy of the approach so that it can be compared with the other methodologies using the same amount of observations.

To analyze the sample size dimension in Section 4.4.1, we first carry out the training/test split stratifying them by the class variable and the auxiliary variable. The test set is left untouched and it is used for all the experiments. Second, to perform experiments with different sample sizes, we take the first N observations of the training subset, learn the model with these N samples, and test it against the test set. We repeat this step by taking training subsamples of increasing size, but the models are always tested against the same (complete) test set. We test different training sample sizes, $N \in \{20, , 60, 120, 260, 500\}$. This procedure is repeated five times and the mean accuracies and standard deviation are reported.

In Section 4.4.2, we generate a synthethic dataset following the procedure from Section 4.2.2, we generate it using pre-trained networks for multiple-sources (gender equal male or female). The generated data has the same sample size as the original data and the same proportions for the protected attribute.

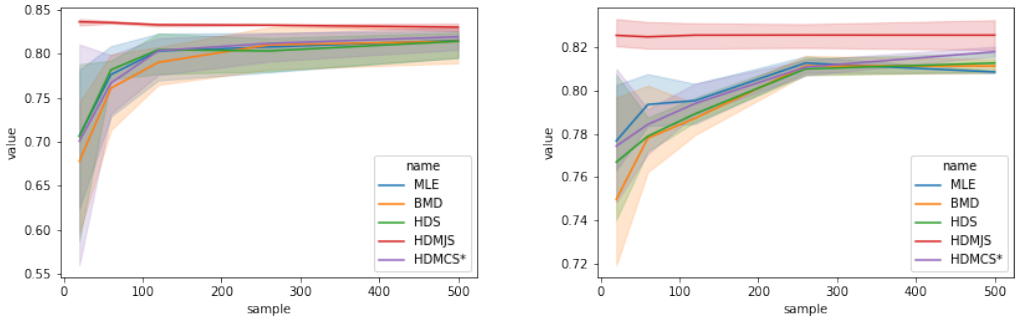
In Section 4.4.3, we first generate synthetic data following the procedure from Section 4.2.2 using a constant random seed and apply the same procedure explained above to do the analysis depending on the sample size. The amount of samples generated are$N \in \{20, , 60, 120, 260\}$. As above, they are generated so that the proportion of male and female is respected.

4.4 Results

In this section, we present the results obtained using the Adult and Diabetes datasets. First, we present the results along the sample size dimension, and then along the distribution divergence dimension. The last part explores a combination of both dimensions.

4.4.1 Experimental results along the first axis: sample dimension

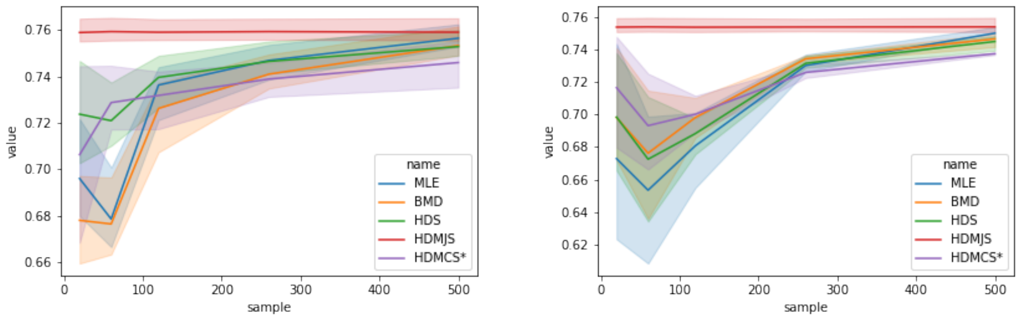
Following the experimental setup defined in Section 4.1, we present the results in terms of accuracy of the models depending on the number of samples of the training dataset. The models' comparison using a Naive Bayes structure are shown in Figures 4.3 and 4.4 for the Adult and Diabetes datasets, respectively. Two values of α are considered ($\alpha \in \{1, 10\}$). It represents the value s given to the hyperprior in the hierarchical models and the value of the equivalent sample size in the Multinomial-Dirichlet model. The larger α the more confidence we have in our prior and the less we let data move the estimation away from the prior. Throughout all the experiments, HDMJS is always the best-performing method. However, it shows a behaviour that could be considered as suspicious: it is almost constant and seems independent of the sample size or the structure of the model.[1]



(A) $\alpha = 1$

(B) $\alpha = 10$

FIGURE 4.3: Comparison of the methods on the first axis: sample dimension, for the Adult dataset (NB structure)



(A) $\alpha = 1$

(B) $\alpha = 10$

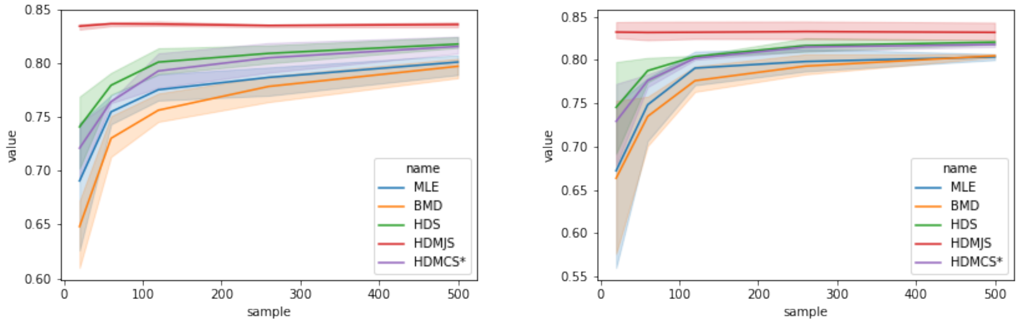
FIGURE 4.4: Comparison of the methods on the first axis: sample dimension, for the Diabetes dataset in terms of accuracy (NB structure)

In the Adult dataset, the HDMJS performs better than the other models. For an $\alpha = 1$, our proposal HDMCS* shows similar performance as the other models. When $\alpha = 10$ and the number of samples increases, HDMCS* becomes the second-best model. All methods tend to a similar solution as the sample sizes increase.

In the Diabetes dataset, the comparisons are more dependent on the number of samples. As expected, MLE performs worse when the sample size is small and is outperformed by Bayesian approaches. The HDMCS* model is the worse performing when the number of samples increases. Although all methods tend to the same solution as the sample size increases, the convergence seems to be slower in the case of the HDMCS*

Similar results when using a TAN structure are presented in Figures 4.5 and 4.6.

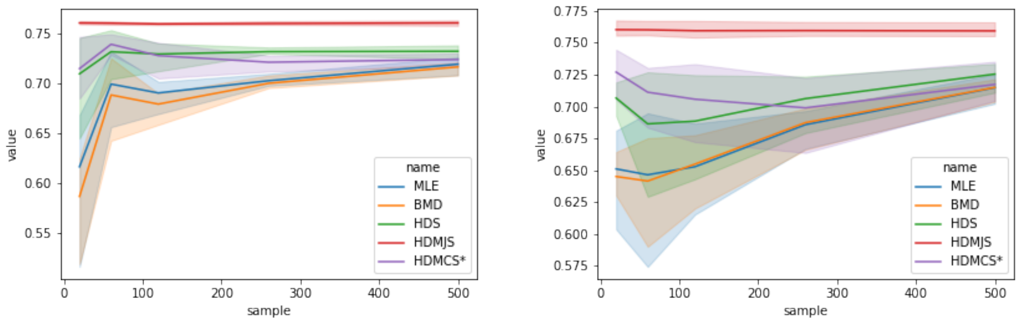
[1] This behaviour is consistent with a coding error. Unfortunately, we have not been able to find any mistakes.



(A) $\alpha = 1$

(B) $\alpha = 10$

FIGURE 4.5: Comparison of the models on the first axis: sample dimension, for the adult dataset in terms of accuracy (TAN structure)



(A) $\alpha = 1$

(B) $\alpha = 10$

FIGURE 4.6: Comparison of the models on the first axis: sample dimension, for the diabetes dataset (TAN structure)

Using a TAN instead of naive Bayes, the HDMJS still performs better than the other models. The HDS and the HDMCS* are close in performance, with HDS performing slightly better (mainly when the sample size is larger). Finally, non-hierarchical models are the ones that perform worse. Taking into account the results for both naive Bayes and TAN, it seems that the more complex the structure of the model is the better the hierarchical models perform compared to non-hierarchical models. As we add more complexity to the model the magnitude of the counts $N_{x,pa}$ for each feature decreases. The sharing of information between the CPDs that hierarchical models perform becomes more relevant when the magnitude of the counts $N_{x,pa}$ is low. That is, the "borrowed statistical strength" by the CPDs, as explained in (Azzimonti, Corani, and Zaffalon, 2019), is more effective.

As shown in Tables 4.1 and 4.2, when we do the same comparison but using all data available in the datasets, all the methods perform similarly, as expected.

Model	Accuracy	Precision	Recall	F1-score
BMD	0.815	0.612	0.723	0.663
HDS	0.815	0.612	0.725	0.652
HDMJS	0.821	0.626	0.716	0.669
HDMCS	0.820	0.625	0.709	0.664
MLE	0.805	0.593	0.725	0.652

TABLE 4.1: Comparison of the models on the first axis: sample dimension, for the adult dataset (TAN structure, full dataset)

Model	Accuracy	Precision	Recall	F1-score
BMD	0.759	0.76	0.999	0.863
HDS	0.759	0.76	0.999	0.863
HDMJS	0.76	0.76	0.999	0.863
HDMCS	0.759	0.76	0.998	0.863
MLE	0.759	0.76	0.998	0.863

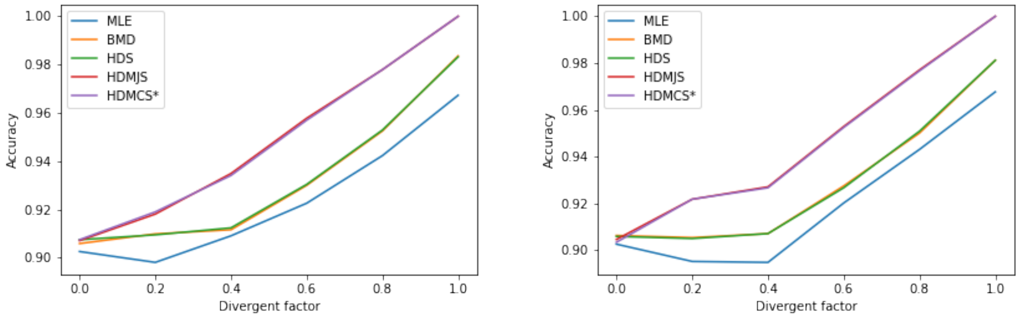
TABLE 4.2: Comparison of the models on the first axis: sample dimension, for the diabetes dataset (TAN structure, full dataset)

When the amount of samples is sufficiently large it is enough to estimate by means of MLE or simple Bayesian Estimation to capture the underlying distribution of the data. The intuition is that the counts $N_{x,pa}$ become sufficiently large and the effect of the priors loses relevance.

To better understand the comparisons between the different models we refer to Section 4.4.2, where we analyze how the difference in distributions of the multiple data sources affects the performance of the learned models.

4.4.2 Experimental results along the second axis: diverging multi-source distributions

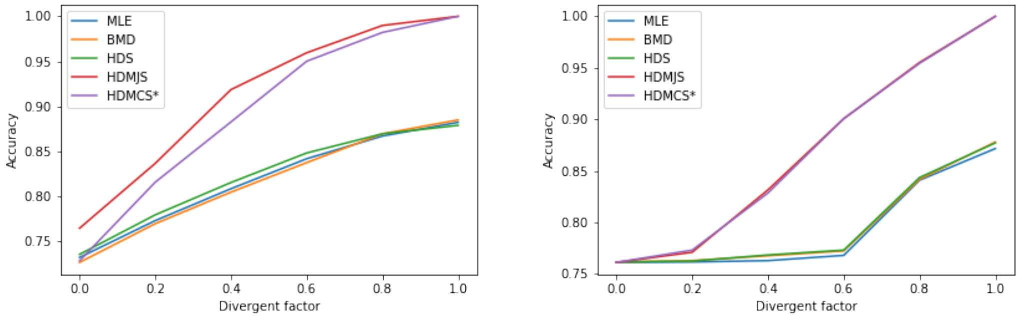
In this section, we apply the methodology from Section 4.2.2 and generate synthetic stratified data with the same sample size as the original data. Due to the way in which our experiment is conducted the distributions of all conditional probabilities of the explanatory variables diverge with δ. As we generate the samples using the diverged distributions, the models' accuracies will, in general, improve because, as δ increases, more unbalanced distributions are generated, which are easier to learn from data. Hence, it is important to analyze the distance between the accuracies of the models and not their magnitude. An important observation is that in real-world data, we would not expect data to diverge by a factor of δ close to 1. Therefore, we will focus our comparison on intermediate values of δ. Figures 4.7 and 4.8 show the results for the Adult dataset and the Diabetes dataset, respectively, using a naive Bayes structure. We can observe that HDMJS and HDMCS* perform similarly and better than the others when data starts to diverge, both in Adult and Diabetes datasets. The other three methods perform similarly.



(A) $\alpha = 1$

(B) $\alpha = 10$

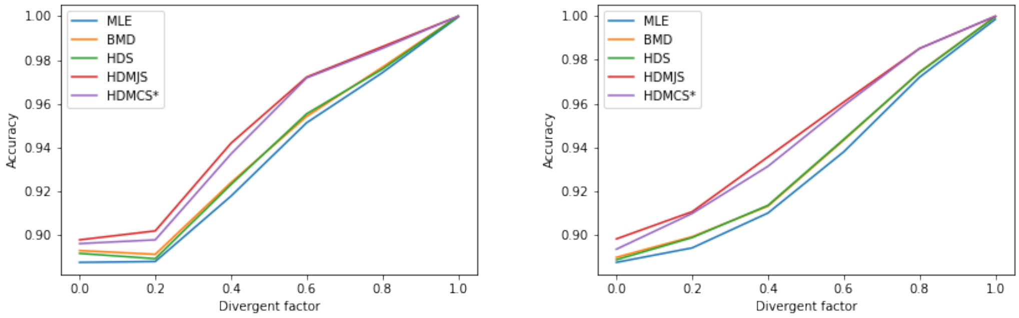
FIGURE 4.7: Comparison of the models on the second axis: diverging multi-source distributions, for the Adult dataset (NB structure)



(A) $\alpha = 1$

(B) $\alpha = 10$

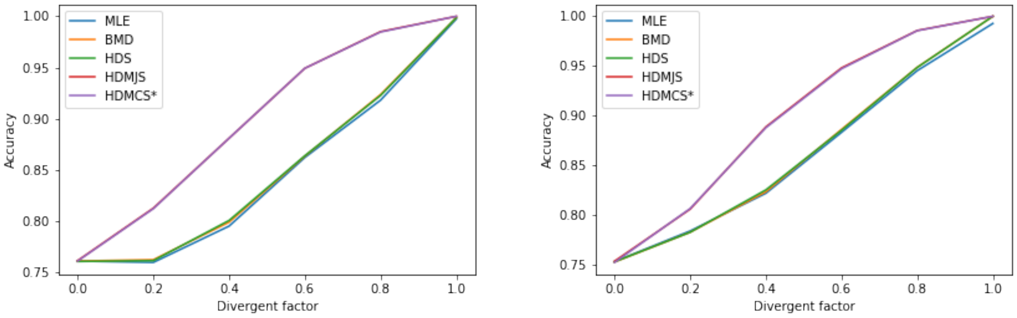
FIGURE 4.8: Comparison of the models on the second axis: diverging multi-source distributions, for the Diabetes dataset (NB structure)



(A) $\alpha = 1$

(B) $\alpha = 10$

FIGURE 4.9: Comparison of the models on the second axis: diverging multi-source distributions, for the adult dataset (TAN structure)



(A) $\alpha = 1$

(B) $\alpha = 10$

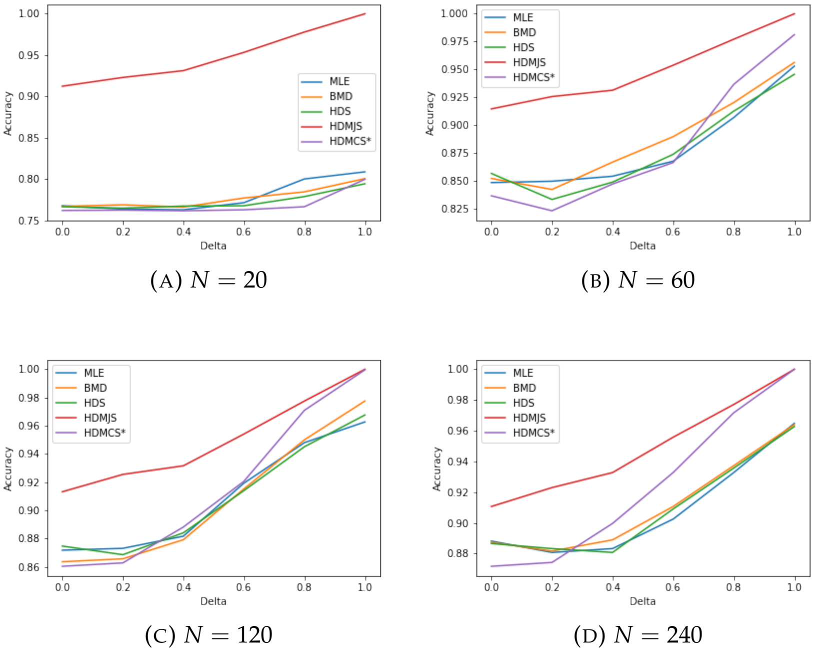
FIGURE 4.10: Comparison of the models on the second axis: diverging multi-source distributions, for the diabetes dataset (TAN structure)

Figures 4.9 and 4.10 show the same analysis but using a TAN instead of a Naive Bayes. Again, HDMJS and HDMCS* perform better when data starts to diverge. They perform almost equally, with minimal variations. This improvement in performance is driven by the fact that now data coming from multiple sources are differently distributed. When this happens, both HDMJS and HDMCS* are equipped with a hierarchical Bayesian prior that allows their models to adapt better to multiple sources that are differently distributed. The other three models do not incorporate this mechanism, or use it differently, and perform worse (BMD is not visible in the plot because it performs equally as the HDS). Another important observation is that models that do not incorporate the auxiliary variables do substantially worse when data starts to diverge if naive Bayes is used instead of TAN.

As explained above, from both datasets, our results suggest that our method, HDMCS*, and HDMJS are the ones that adapt better to divergences of distributions between multiple datasets, as expected. However, as we defined synthetic datasets with the same amount of samples as the real datasets, the number of samples used is very large. We would like to analyze how the divergence in distribution affects the results when the number of samples available from each data source is smaller. In the following section, we generate smaller synthetic samples and compare the models' performances along both axes.

4.4.3 Experimental results along both axes

In this section, we analyse how the performance of the models changes depending on both the sample size and the divergence in the distribution of the subpopulations. Figures 4.11 and 4.12 show, for Adult and Diabetes, respectively, the accuracy of the different methods as the amount of divergence δ increases for different samples sizes. In both cases, the structure is a TAN and $\alpha = 10$. Figures A.3, A.2 in Appendix A show the results with naive Bayes and $\alpha = 1$.



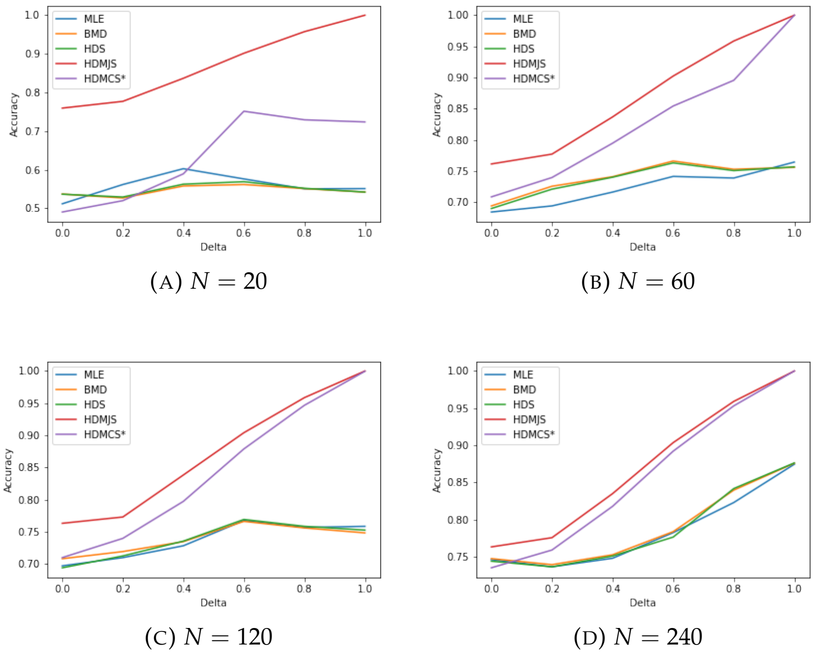
(A) $N = 20$

(B) $N = 60$

(C) $N = 120$

(D) $N = 240$

FIGURE 4.11: Comparison of the models on both axes: sample dimension and diverging multi-source distributions, for the Adult dataset (NB structure, $\alpha = 10$)



(A) $N = 20$

(B) $N = 60$

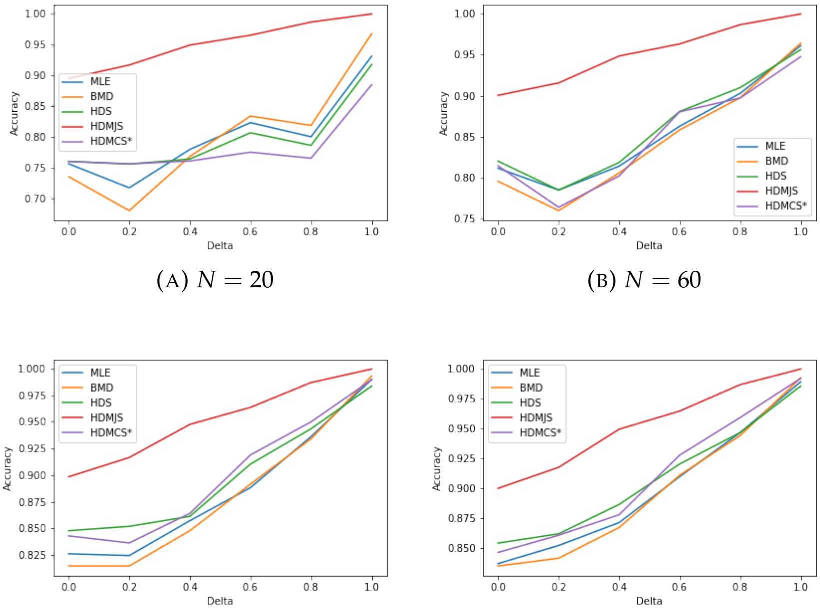
(C) $N = 120$

(D) $N = 240$

FIGURE 4.12: Comparison of the models on both axes: sample dimension and diverging multi-source distributions, for the Diabetes dataset (NB structure, $\alpha = 10$)

With the Adult dataset, the results for naive Bayes suggest that, for smaller sample sizes, even if the distribution of the multiple sources diverges by a lot, our method performs worse than non-hierarchical models. With the Diabetes dataset, even with smaller sample sizes our model performs better when distributions start to diverge. In both datasets, as the number of samples increases, smaller values of δ make the HDMCS* better compared to the methods that do not explicitly consider the existence of multiple sources.

Figures 4.13 and 4.14 show the same analysis but using TAN instead of naive Bayes, and $\alpha = 10$. Figures A.4 and A.5 in Appendix A show the results for TAN and $\alpha = 1$.



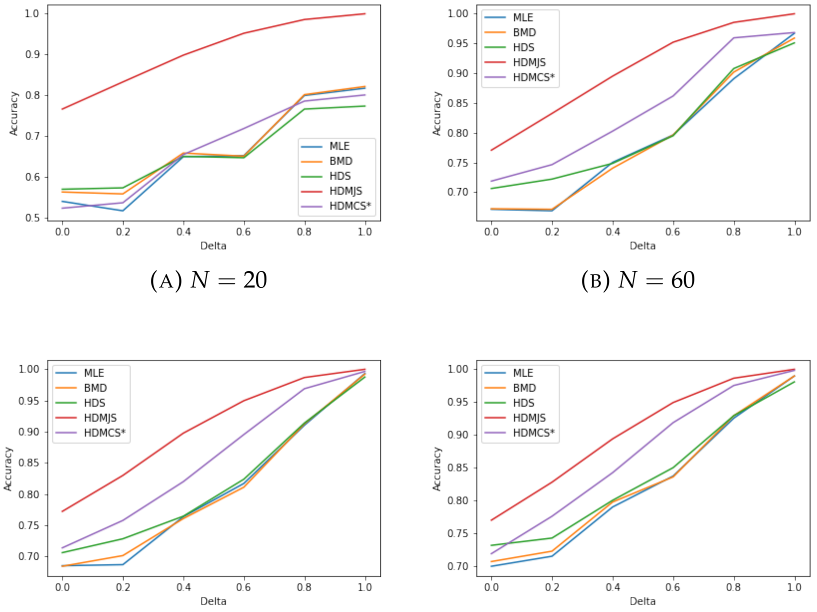
(A) $N = 20$

(B) $N = 60$

(C) $N = 120, \alpha = 10$

(D) $N = 240$

FIGURE 4.13: Comparison of the models on both axes: sample dimension and diverging multi-source distributions, for the adult dataset (TAN structure, $\alpha = 10$)



(A) $N = 20$

(B) $N = 60$

(C) $N = 120$

(D) $N = 240$

FIGURE 4.14: Comparison of the models on both axes: sample dimension and diverging multi-source distributions, for the diabetes dataset (TAN structure, $\alpha = 10$)

With the Adult dataset, our results suggest that when the underlying distributions diverge considerably, HDMJS is the best method. Our method, HDMCS*, performs very similarly as HDS. With Diabetes, the HDMCS is again the best model followed by HDMCS*. The results suggest that when data is not very scarce and the subsamples diverge from each other, our proposed methodology could be a reasonable method to learn a Bayesian network model. This result could be more exacerbated in the case where the auxiliary attribute had more than two possible categories.

Overall, when the distribution of the multiple data sources diverges, HDMJS and our method HDMCS* perform better, as they take into account the existence of the multiple sources represented by the auxiliary variable F. HDMJS and HDMCS* perform similarly but, when the number of samples reduces, the performance of our method moves away from that of HDMJS. A possible explanation is that, by definition, HDMJS uses all the available data to compute the **joint** distribution tables corresponding to each data source. In that sense, they "borrow statistical strength" from each other using all the available data. Our proposed methodology only links the conditional probability distributions column-wise, that is, for distributions with the same parent configuration, pa. Thus, the shared information is more limited. When the sample size is very small, i.e., the counts $N_{x,pa}$ can become very small, this lack of information to share is more relevant. When the sample size increases, this effect becomes less dominant and both HDMJS and HDMCS* perform similarly.

Chapter 5

Conclusions

In this project, we study the problem of learning the parameters of a Bayesian network model from data provided by multiple sources. The objective is to learn a model that can generalize well and is able to model together data sampled from different distributions. This problem links directly with the problem of fairness in machine learning, where a protected attribute can be viewed as an auxiliary variable that defines different data sources.

We propose a new method (HDMCS*) to learn from a dataset with multiple sources as a natural alternative to previous methods. We compare our methodology against the state-of-the-art on two dimensions: the sample size, and the divergence in the underlying distribution of each data source. We test our results on transformations of two popular datasets (Adult and Diabetes) using "gender" as the protected attribute. That is, we work in the fairness-analysis context.

A previous method, HDMJS, seems to systematically outperform our HDMCS*, although ours proposes an approach that is more natural for the BN framework. Our method generally outperforms other competitors. When the number of samples from each data source is small, our model performs better than non-hierarchical models, and very similarly to HDS. When the structure of the network is more complex our model also outperforms non-hierarchical models. Furthermore, when the underlying distributions of the multiple subsets diverge, HDMCS* performs even better when compared to those competitors. These results could be amplified if the number of sources would be higher.

5.1 Future Work

There are several open lines of research regarding our work. Firstly, it would be interesting to do a comparison with a higher number of sources as defined by the auxiliary attribute, as we limited our experiments to two subsets (according to the sensible attribute "gender"). Second, whereas we assumed two simple structures for the models (Naive Bayes and TAN), future research could try to find the best structure for the pooled dataset before conducting parameter estimation, and analyse how the results would change in that scenario. Third, we have used STAN's off-the-self variational inference algorithm. It would be interesting to use a specifically designed method, such as the one proposed by Azzimonti, Corani, and Zaffalon (2019), to better exploit the particularities of our method.

Future work could delve more in the effect of the hyper-parameters on the models comparisons. The performance of the HDMJS should be analysed in more detail. The accuracy of this method in some experiments is almost invariant to the number of samples used to train it. A possibility is that there exists a coding error, although our careful checking was not able to find any. Future work should also consider other performance measures, like the brier-score, and do a more extensive analysis with other measures, such as precision and recall.

Finally, it would be interesting to explore a method to generate synthetic data with diverging distributions that does not lead to models that are, on average, more accurate.

Appendix A

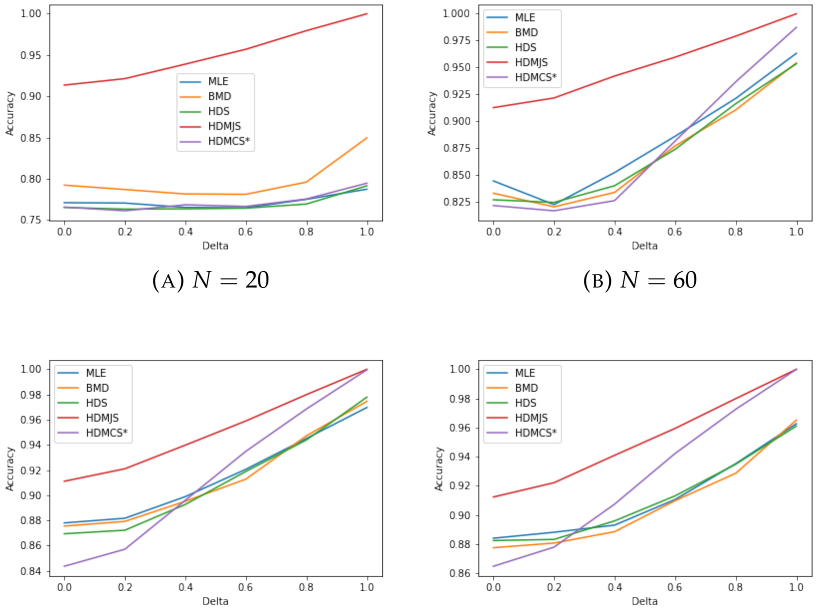
Appendix

Variable	Type	Missing Values	Values
age	numerical	0	[17-90]
workclass	categorical	2799	7
fnlwgt	Numerical	0	[13492-1490400]
education	categorical	0	16
educational-num	Numerical	0	[1-16]
marital-status	categorical	0	7
occupation	categorical	2809	14
relationship	categorical	0	6
race	categorical	0	5
gender	categorical	0	2
hours-per-week	categorical	0	[1-99]
native-country	categorical	857	41
income	categorical	0	2

TABLE A.1: Adult dataset: variables description

Variable	Type	Missing Values	Values
race	categorical	2273	6
gender	categorical	0	3
age	categorical	0	10
time_in_hospital	categorical	0	[1-14]
num_procedures	numerical	0	[0-6]
num_medications	numerical	0	[1-81]
number_outpatient	numerical	0	[0-42]
number_emergency	numerical	0	[0-76]
number_inpatient	numerical	0	[0-21]
A1Cresult	numerical	0	4
metformin	categorical	0	4
chlorpropamide	categorical	0	4
glipzide	categorical	0	4
rosiglitazone	categorical	0	4
acarbose	categorical	0	4
miglitol	categorical	0	4
diabetesMed	categorical	0	2
readmitted	categorical	0	3

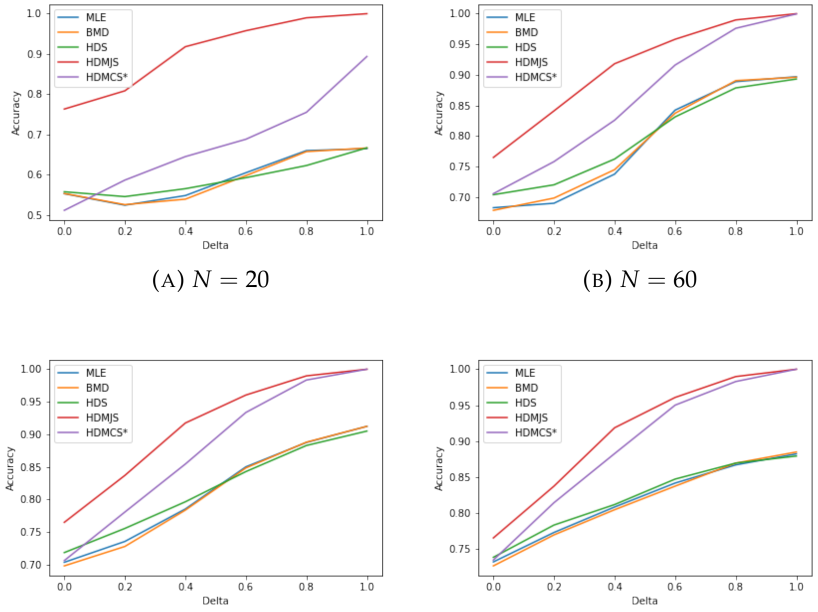
TABLE A.2: Diabetes dataset: : variables description



(A) $N = 20$ (B) $N = 60$

(C) $N = 120$ (D) $N = 240$

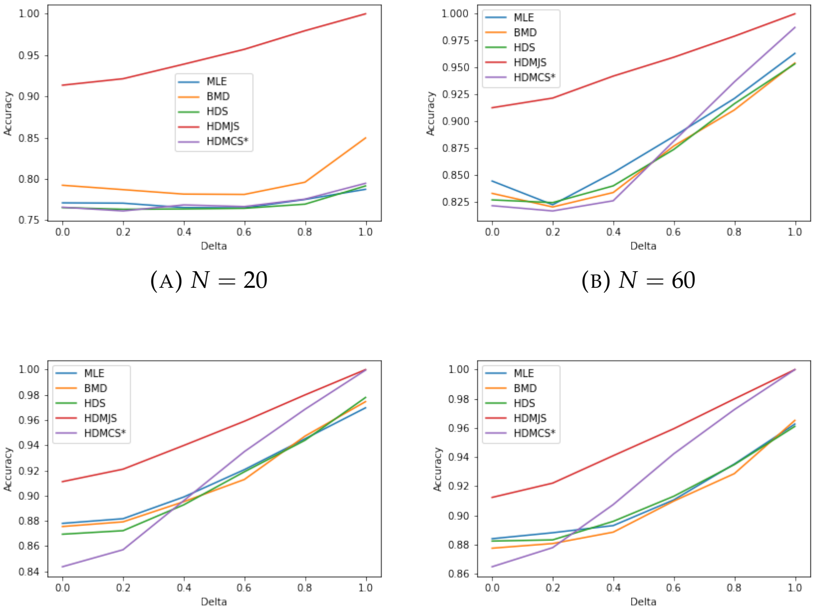
FIGURE A.1: Comparison of the models on both axes: sample dimension and diverging multi-source distributions, for the adult dataset (Naive Bayes structure, $\alpha = 1$)



(A) $N = 20$ (B) $N = 60$

(C) $N = 120$ (D) $N = 240$

FIGURE A.2: Comparison of the models on both axes: sample dimension and diverging multi-source distributions, for the diabetes dataset (Naive Bayes structure, $\alpha = 1$)



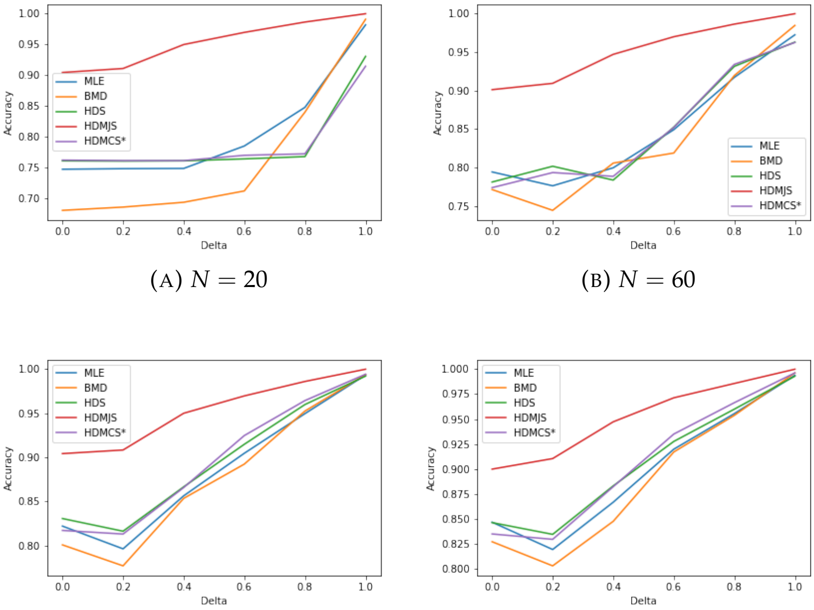
(A) $N = 20$

(B) $N = 60$

(C) $N = 120$

(D) $N = 240$

FIGURE A.3: Comparison of the models on both axes: sample dimension and diverging multi-source distributions, for the adult dataset (Naive Bayes structure, $\alpha = 1$)



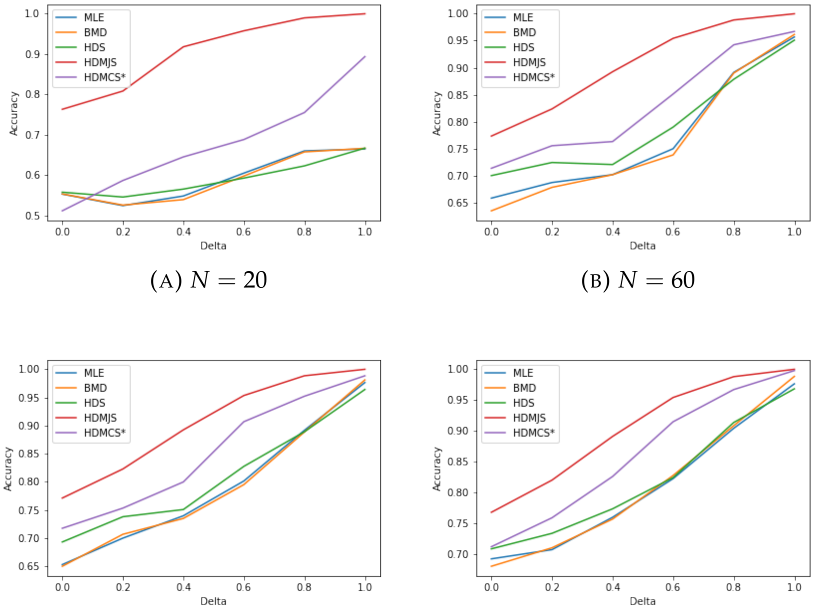
(A) $N = 20$

(B) $N = 60$

(C) $N = 120$

(D) $N = 240$

FIGURE A.4: Comparison of the models on both axes: sample dimension and diverging multi-source distributions, for the diabetes dataset (TAN structure, $\alpha = 1$)



(A) $N = 20$

(B) $N = 60$

(C) $N = 120$

(D) $N = 240$

FIGURE A.5: Comparison of the models on both axes: sample dimension and diverging multi-source distributions, for the diabetes dataset (TAN structure, $\alpha = 1$)

Bibliography

Azzimonti, Laura, Giorgio Corani, and Marco Scutari (2022). "A Bayesian hierarchical score for structure learning from related data sets". In: *International Journal of Approximate Reasoning* 142, pp. 248–265. ISSN: 0888613X. DOI: 10.1016/j.ijar.2021.11.013. URL: https://doi.org/10.1016/j.ijar.2021.11.013.

Azzimonti, Laura, Giorgio Corani, and Marco Zaffalon (2019). "Hierarchical estimation of parameters in Bayesian networks". In: *Computational Statistics and Data Analysis* 137, pp. 67–91. ISSN: 01679473. DOI: 10.1016/j.csda.2019.02.004.

Ballester-Ripoll, Rafael and Manuele Leonelli (2022). "You Only Derive Once (YODO): Automatic Differentiation for Efficient Sensitivity Analysis in Bayesian Networks". In: pp. 1–12. arXiv: 2206.08687. URL: http://arxiv.org/abs/2206.08687.

Bernardo, José M (1994). "Bayesian statistics". In: *Probability and Statistics, R. Viertl, Ed* 2, pp. 345–407.

Bielza, Concha and Pedro Larranaga (2014). "Discrete Bayesian network classifiers: A survey". In: *ACM Computing Surveys (CSUR)* 47.1, pp. 1–43.

Carpenter, Bob et al. (2017). "Stan: A probabilistic programming language". In: *Journal of Statistical Software* 76.1. ISSN: 15487660. DOI: 10.18637/jss.v076.i01.

Geiger, Dan and David Heckerman (1996). "Knowledge representation and inference in similarity networks and Bayesian multinets". In: *Artificial Intelligence* 82.1-2, pp. 45–74.

Konstantinov, Nikola and Christoph H. Lampert (2022). "Fairness-Aware PAC Learning from Corrupted Data". In: *Journal of Machine Learning Research* 23, pp. 1–60. ISSN: 15337928. arXiv: 2102.06004.

Le Quy, Tai et al. (2022). "A survey on datasets for fairness-aware machine learning". In: *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 12.3, pp. 1–59. ISSN: 19424795. DOI: 10.1002/widm.1452. arXiv: 2110.00530.

McMahan, Brendan et al. (2017). "Communication-efficient learning of deep networks from decentralized data". In: *Artificial intelligence and statistics*. PMLR, pp. 1273–1282.

Oates, Chris J et al. (2014). "Joint estimation of multiple related biological networks". In:

Oates, Chris J et al. (2016). "Exact estimation of multiple directed acyclic graphs". In: *Statistics and Computing* 26, pp. 797–811.

Strømsø, H. I. and I. Bråten (2009). "Learning from Multiple Information Sources". In: *International Encyclopedia of Education, Third Edition* 9, pp. 191–196. DOI: 10.1016/B978-0-08-044894-7.00496-6.

Teh, Yee et al. (2004). "Sharing clusters among related groups: Hierarchical Dirichlet processes". In: *Advances in neural information processing systems* 17.

Tillman, Robert E. (2009). "Structure learning with independent non-identically distributed data". In: *Proceedings of the 26th International Conference On Machine Learning, ICML 2009*, pp. 1041–1048.

Tillman, Robert E. and Peter Spirtes (2011). “Learning equivalence classes of acyclic models with latent and selection variables from multiple datasets with overlapping variables”. In: *Journal of Machine Learning Research* 15, pp. 3–15. ISSN: 15324435.