Probabilistic Multi-Dimensional Classification

Abstract

Multi-dimensional classification (MDC) can be employed in a range of applications where one needs to predict multiple class variables for each given instance. Arguably, probabilistic MDC has been studied seldom when compared to its single class variable counterpart. Existing MDC methods often suffer from at least one of inaccuracy, scalability, limited use to certain types of data, hardness of interpretation or lack of probabilistic (uncertainty) estimations. To our best knowledge, this paper is a first attempt to address all these disadvantages simultaneously. We propose a formal framework for probabilistic MDC in which learning an optimal multi-dimensional classifier can be decomposed, without loss of generality, into learning a set of (smaller) single-variable multi-class probabilistic classifiers and a directed acyclic graph. Current and future developments of both probabilistic classification and graphical model learning can directly enhance our framework, which is flexible and provably optimal. A collection of experiments is conducted to highlight the usefulness of this MDC framework.

1 INTRODUCTION

In (multi-class) classification, a predictive system makes use of a training data set (consisting of input-output pairs which specify individuals of a population) and a hypothesis space (consisting of the possible classifiers), and seeks for a classifier that optimizes its chance of making accurate predictions with respect to some given evaluation criterion (such as a loss function or an accuracy metric). Numerous studies on classification have been devoted to learning probabilistic classifiers which predict, for each observation of the input space, a univariate probability distribution over the output space. The intention of probabilistic classification is not only to provide the end user with all necessary information about the optimal predictions of different loss functions [Elkan, 2001], but also information about the uncertainty associated with the possible predictions.

To overcome the assumption that the output space must be fully characterized by a single class variable, multi-dimensional classification (MDC) has been proposed in which the output space is characterized by multiple class variables which can be correlated. MDC appears in important applications. An example of MDC is predicting subtypes/stages of diseases associated with each patient given his/her medical image and/or demographic information. Few other examples of MDC tasks are classification of biomedical text [Shatkay et al., 2008], vehicle classification [Jia and Zhang, 2021] and others [Gil-Begue et al., 2021, Jia and Zhang, 2022].

Existing multi-dimensional classifiers are often non-probabilistic [Jia and Zhang, 2021], inaccurate [Jia and Zhang, 2022][Section II & III], or unscalable [Gil-Begue et al., 2021]. To the best of our knowledge, no existing method specific for MDC is capable of directly handling mixed data, i.e., continuous and discrete features coexisting (without preprocessing or other external manipulations). Problem transformation methods [Jia and Zhang, 2021] which transform the original MDC problem into either a huge multi-class classification (MCC) problem, for example using the class powerset (CP) classifier, or a set of independent MCC problems, for example the Binary relevance (BR) classifier, can be combined with deep multimodal learning [Kline et al., 2022, Ramachandram and Taylor, 2017, Xu et al., 2021] to handle mixed data and other complex types of input. They are often either inaccurate or unscalable. More importantly, they are arguably hard to interpret. The set of marginal probability distributions provided by BR can be associated to (infinitely) many joint distributions over the class variables and does not inform much about the (true) joint distribution, while the joint distribution provided by CP contains an exponential number of masses and is not

Submitted to the 39th Conference on Uncertainty in Artificial Intelligence (UAI 2023). **To be used for reviewing only**.

easily interpretative for end users.

We present a framework to learn probabilistic multi-dimensional classifiers which are interpretable, accurate, scalable and are capable of handling mixed data. This formal framework for probabilistic MDC allows us to learn an optimal multi-dimensional classifier, without loss of generality/optimality, by decomposing the task into learning a set of probabilistic MCC models plus a directed acyclic graph (DAG). Notably, the framework inherits the interpretability of Bayesian networks (BNs), which is a compact representation of quantitative and qualitative probabilistic relationships among class variables, and the scalability and flexibility of deep (multimodal) learning, i.e., handling complex types of data. Moreover, the probabilistic nature allows the framework, among other characteristics, to optimize different loss functions by only learning a single probabilistic model. We prove that the probabilistic model learned by this framework is universal and the learning procedure is globally optimal whenever MCC is universal and can be solved optimally too. This means no other approach to solve MDC, using a collection of MCCs, can do a better job.

We formalize the probabilistic MDC problem in Section 2, present formal results on the optimality of the framework in Section 3, followed by a practical algorithm and properties of the learning framework in Sections 3.1 to 3.4. Section 4 discusses the inference task, and Section 5 further motivates the framework by presenting a collection of experiments indicating the advantages of the framework against existing MDC approaches. Section 6 concludes this paper. All formal results in this paper (propositions) are stated without proofs, which are deferred to Appendix B.

2 PROBABILISTIC MDC

Let $\mathbf{X} = \{X^1, \ldots, X^Q\}$ be a finite set of features, let $\mathcal{X} := \mathcal{X}^1 \times \ldots \times \mathcal{X}^Q$ denote an input space, and let $\mathbf{Y} = \{Y^1, \ldots, Y^K\}$ be a finite set of class variables. Let $\mathcal{Y}^k = \{y^{k,1}, \ldots, y^{k,M_k}\}$ be the set of M_k possible outcomes for the kth class variable Y^k, $k \in [K] := \{1, \ldots, K\}$. We define $\mathbf{Z} := \mathbf{Y} \cup \mathbf{X}$. We denote by $\mathbf{X}_d$ and $\mathbf{X}_c$ the discrete feature set and continuous feature set, respectively. We also define $\mathbf{Z}_d := \mathbf{Y} \cup \mathbf{X}_d$. For each instance $\boldsymbol{x} \in \mathcal{X}$, we say it is associated with a (vector)class $\boldsymbol{y} \in \mathcal{Y} = \mathcal{Y}^1 \times \ldots \times \mathcal{Y}^K$.

We assume observations to be realizations of independently and identically distributed (i.i.d.) random variables generated according to a probability distribution on $\mathcal{X} \times \mathcal{Y}$, i.e., an observation $\boldsymbol{y} = (y^1, \ldots, y^K)$ is the realization of a corresponding random vector $\mathbf{Y} = (Y^1, \ldots, Y^K)$. Let $\boldsymbol{p}(\mathbf{X}, \mathbf{Y})$ be a (mixed) joint density function. We denote by $\boldsymbol{p}(\mathbf{Y} \mid \boldsymbol{x})$ the conditional joint distribution of $\mathbf{Y}$ given $\mathbf{X} = \boldsymbol{x}$, whose probability mass function is given by

$$\boldsymbol{p}(\boldsymbol{y} \mid \boldsymbol{x}) := \frac{\boldsymbol{p}(\boldsymbol{x}, \boldsymbol{y})}{\sum_{\boldsymbol{y}' \in \mathcal{Y}} \boldsymbol{p}(\boldsymbol{x}, \boldsymbol{y}')}, \forall (\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{X} \times \mathcal{Y}. \quad (1)$$

We denote by $\boldsymbol{p}(Y^k \mid \boldsymbol{x})$, $k \in [K]$, the marginal distribution of Y^k, whose probability mass function is given by

$$\boldsymbol{p}(y^k \mid \boldsymbol{x}) := \sum_{\boldsymbol{y} \in \mathcal{Y}: Y^k = y^k} \boldsymbol{p}(\boldsymbol{y} \mid \boldsymbol{x}), \forall y^k \in \mathcal{Y}^k. \quad (2)$$

Given training data in the form of a finite set of observations

$$\mathcal{D} = \left\{(\boldsymbol{x}_n, \boldsymbol{y}_n)\right\}_{n=1}^N \subset \mathcal{X} \times \mathcal{Y}, \quad (3)$$

drawn independently from a distribution, the goal in MDC is to learn a predictive classifier model $\boldsymbol{h} : \mathcal{X} \longrightarrow \mathcal{Y}$ that assigns a $\boldsymbol{y} \in \mathcal{Y}$ to each instance $\boldsymbol{x} \in \mathcal{X}$. Thus, the output of a classifier $\boldsymbol{h}$ is a vector

$$\hat{\boldsymbol{y}} := \boldsymbol{h}(\boldsymbol{x}) = \left(h^1(\boldsymbol{x}), \ldots, h^K(\boldsymbol{x})\right) \in \mathcal{Y}. \quad (4)$$

In a probabilistic setting, a classification task can be viewed as a two-stage problem, in which a mapping $\boldsymbol{h} : \mathcal{X} \longrightarrow \mathcal{Y}$ is not learned directly, but in a more indirect way. Roughly speaking, one can split a probabilistic classification into two tasks: learning a function $\boldsymbol{p} : \mathcal{X} \longrightarrow \boldsymbol{p}(\mathcal{Y}|\mathcal{X})$ (with abuse of notation) and constructing an efficient inference operator $o : \boldsymbol{p}(\mathcal{Y}|\mathcal{X}) \longrightarrow \mathcal{Y}$ (we will deal with o in Section 4).

Motivated by the observations that discriminative models can perform better than generative models in many classification tasks [Bouchard and Triggs, 2004, Ng and Jordan, 2001, Ulusoy and Bishop, 2006], and by the fact that in M-open cases [Bernardo and Smith, 2000], maximizing the (log) likelihood function may not converge to a best possible distribution as maximizing the conditional (log) likelihood function does [Roos et al., 2005], we learn a multi-dimensional classifier encoding $\boldsymbol{p}$ which maximizes the conditional log likelihood (CLL) function $C(\boldsymbol{p} \mid \mathcal{D})$:

$$C(\boldsymbol{p} \mid \mathcal{D}) := \log \prod_{n=1}^N \boldsymbol{p}(\boldsymbol{y}_n \mid \boldsymbol{x}_n). \quad (5)$$

Let $\mathcal{P}^0$ be a hypothesis space which consists of the possible functions $\boldsymbol{p}$. The learning problem can be defined as finding

$$\boldsymbol{p}^* \in \arg\max_{\boldsymbol{p} \in \mathcal{P}^0} C(\boldsymbol{p} \mid \mathcal{D}). \quad (6)$$

3 A LEARNING FRAMEWORK

The optimization problem (6) is very generic and its complexity highly depends on the given hypothesis space $\mathcal{P}^0$. We present reformulations of this problem that are more suitable to be optimized based on some assumptions about the hypothesis space. We proceed under the assumption that the features X^q, $q \in [Q]$, are always made available. This means we neither assume missing values at the training time nor assume missing features at the prediction time [Saar-Tsechansky and Provost, 2007]. This is not a limitation of the approach and missing data can be tackled smartly within

the same overall ideas, but the discussion and presentation would go beyond the scope of this manuscript.

Throughout, we assume the chain rule of probability holds. Using the concept of conditional independence, we can assume without loss of generality that any $\boldsymbol{p}(\mathbf{X}, \mathbf{Y})$ can be fully encoded by a DAG G and a parameter set θ inducing the factorization

$$\boldsymbol{p}_\theta^G(\boldsymbol{x}, \boldsymbol{y}) = \prod_{X \in \mathbf{X}_c} \boldsymbol{p}_\theta(x \,|\, \pi_x) \prod_{Z \in \mathbf{Z}_d} \boldsymbol{p}_\theta(z \,|\, \pi_z)\,, \quad (7)$$

where π_x and π_z are (with abuse of notation) called configurations (compatible with $(\boldsymbol{x}, \boldsymbol{y})$) of the parent sets Δ_G^X and Δ_G^Z (for easiness, we assume that discrete parts of configurations are dictionaries with pairs (variable, value), and continuous parts are given via the appropriate functionals). The complexity of this factorization depends on G.

Therefore, the hypothesis space of any probabilistic MDC can be defined as $\mathcal{P} := \mathcal{G} \times \Theta$, where $\mathcal{G}$ and Θ are respectively the set of possible DAGs and the set of possible parameter sets, and the problem (6) becomes

$$\boldsymbol{p}_{\theta^*}^{G^*} \,:\, (G^*, \theta^*) \in \operatorname*{arg\,max}_{(G,\theta) \in \mathcal{P}} C(\boldsymbol{p}_\theta^G \,|\, \mathcal{D})\,. \quad (8)$$

A learning procedure is optimal if it can find an optimal pair (G^*, θ^*). Parameter learning is optimally solved if we can find θ^* in (8) for a given $G \in \mathcal{G}$. In the following, we show that the factorization in (7) can lead to a great simplification of the learning problem (8).

Proposition 3.1. *Assume the parameter learning problem is optimally solved. We have*

$$\max_{\boldsymbol{p} \in \mathcal{P}^0} C(\boldsymbol{p} \,|\, \mathcal{D}) = \max_{(G,\theta) \in \mathcal{P}} C(\boldsymbol{p}_\theta^G \,|\, \mathcal{D}) = \max_{(G,\theta) \in \mathcal{P}^1} C(\boldsymbol{p}_\theta^G \,|\, \mathcal{D})\,, \quad (9)$$

where $\mathcal{P}^1 := \mathcal{G}^1 \times \Theta$ and $\mathcal{G}^1 \subsetneq \mathcal{G}$ is the set of DAGs which contain no edge of the form $Y \longrightarrow X$.

We will assume a few times in this document that parameter learning can be optimally solved. In general, this is a strong assumption. However, we almost always deal with factorizations of $\boldsymbol{p}$ where each factor involves a small number of variables. In these cases, we hope one can learn the parameters well (certainly much better than in a global model). This is a condition we expect from local models in the factorization in order to prove the optimality of the framework. Note that the cardinality $|\mathcal{G}^1| = R(K)2^{KQ}R(Q)$ can be much smaller than $|\mathcal{G}| = R(K+Q)$, where $R(\cdot)$ is Robinson's formula [Bielza et al., 2011]. Thus, looking for the best (G, θ) over $\mathcal{P}^1$ can be much more practical than doing so over $\mathcal{P}$. The next proposition shows that finding an optimal pair $(G, \theta) \in \mathcal{P}^1$ is equivalent to finding an optimal pair whose G contains no edge between features.

Proposition 3.2. *For any $G \in \mathcal{G}^1$, the joint conditional distribution* (1) *can be factorized (according to G):*

$$\boldsymbol{p}_\theta^G(\boldsymbol{y} \,|\, \boldsymbol{x}) = \prod_{Y \in \mathbf{Y}} \boldsymbol{p}_\theta\left(y \,|\, \pi_y\right)\,, \forall (\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{X} \times \mathcal{Y}\,, \quad (10)$$

where π_y is the configuration for the parents of Y (according to G) that is compatible with $(\boldsymbol{x}, \boldsymbol{y})$. Moreover, the following relation holds:

$$\max_{(G,\theta) \in \mathcal{P}^1} C(\boldsymbol{p}_\theta^G \,|\, \mathcal{D}) = \max_{(G,\theta) \in \mathcal{P}^2} C(\boldsymbol{p}_\theta^G \,|\, \mathcal{D})\,, \quad (11)$$

where $\mathcal{P}^2 := \mathcal{G}^2 \times \Theta$ and $\mathcal{G}^2 \subsetneq \mathcal{G}^1$ consists of $R(K)2^{KQ}$ DAGs with no edges between any two elements of $\mathbf{X}$.

Thus, we formulate the new optimization problem:

$$\boldsymbol{p}_{\theta^*}^{G^*} \,:\, (G^*, \theta^*) \in \operatorname*{arg\,max}_{(G,\theta) \in \mathcal{P}^2} \log \prod_{n=1}^{N} \prod_{Y \in \mathbf{Y}} \boldsymbol{p}_\theta\left(y_n \,|\, \pi_{y_n}\right)\,. \quad (12)$$

It is clear that solving (12) may lead to sub-optimal solutions, compared to solving (8) if the assumption that the parameter learning problem is optimally solved does not hold, and in that case the relation $\mathcal{G}^2 \subsetneq \mathcal{G}$ implies that the best CLL score attained over $\mathcal{G}^2$ is at best the one attained over $\mathcal{G}$. However, there are strong motivations for why one should solve (12) in practice, instead of (8). First, the optimality of (12) can be reachable under milder conditions, while the optimality of (8) is often unreachable. In fact, solving (8) is often impractical because optimizing the CLL function can be impractical even if $G \in \mathcal{G}$ is given [Friedman et al., 1997]. However, one can be much more optimistic about solving (12). As will be shown in Section 3.1, solving (12) is possible as long as one can learn a set of (independent) probabilistic classifiers, plus learning an optimal DAG over the class variables. So one can use all current/future developments of both probabilistic classification and graphical model learning towards solving (12), which also opens doors for further use of research in these topics. Second, as will be shown in Section 3.1, $\forall G \in \mathcal{G}^2$ and $\forall \boldsymbol{x} \in \mathcal{X}$, $\boldsymbol{p}_\theta^G(\mathbf{Y} \,|\, \boldsymbol{x})$ can be factorized as a product of conditional probability distributions whose conditional part is always specified by a multivariate continuous variable. This provides us with a rich representational capacity as discussed in Section 3.2. In particular, any probabilistic classifier can be directly employed to model conditional probability distributions without requiring any data preprocessing transformation, leading to a rich framework for the employment of sophisticated techniques. The representational capacity would be much weaker if one had to parameterize $G \in \mathcal{G} \setminus \mathcal{G}^1 \supsetneq \mathcal{G} \setminus \mathcal{G}^2$ because it would be needed to find some parametric model to encode all conditional density functions $\boldsymbol{p}_\theta^G(x \,|\, \pi_x)$ whose conditional part would be specified by a mixture of discrete and continuous variables. This would be a challenging problem by itself, especially if one

does not want to use any data preprocessing transformation either before or during the training phase.

Our final simplification of the optimization problem while keeping optimality is to realize that we can seek for an optimal G where all continuous variables are parents of every class variable, that is, $\mathbf{X}_c \subset \Delta_G^Y,\ \forall Y \in \mathbf{Y}$. Besides being non-restrictive (we are forcing arcs to stay put, hence we can always fit any "simpler" distribution which would have dropped some connections by the appropriate parameter learning), this condition has also a positive consequence, as it allows us to use methods which are not able to handle mixed setups of continuous and discrete variables.

Therefore, we introduce an updated version of (12) in which we only force the global learning algorithm to explicitly handle the discrete features, while assuming all continuous ones are passed on to the learning of local models. More formally, let $\mathcal{G}^3 \subsetneq \mathcal{G}^2$ be the set of $R(K)2^{K|\mathbf{X}_d|}$ DAGs such that, $\forall G \in \mathcal{G}^3$ and $\forall Y \in \mathbf{Y}$, we have $\mathbf{X}_c \subset \Delta_G^Y$. We formulate the optimization problem as:

$$\boldsymbol{p}_{\theta^*}^{G^*} \ :\ (G^*, \theta^*) \in \underset{(G,\theta)\in\mathcal{P}^3}{\arg\max} \log \prod_{n=1}^{N} \prod_{Y\in\mathbf{Y}} \boldsymbol{p}_\theta\left(y_n \,|\, \pi_{y_n}\right)\,, \tag{13}$$

where $\mathcal{P}^3 = \mathcal{G}^3 \times \Theta$.

Proposition 3.3. *Assume the parameter learning problem is optimally solved. The following relation holds*

$$\max_{(G,\theta)\in\mathcal{P}^2} C(\boldsymbol{p}_\theta^G \,|\, \mathcal{D}) = \max_{(G,\theta)\in\mathcal{P}^3} C(\boldsymbol{p}_\theta^G \,|\, \mathcal{D})\,. \tag{14}$$

The conclusion here is that we can have a globally optimal probabilistic MDC whose optimization is done via (13), potentially saving significant time and data requirements for training. One needs "only" to learn the local conditional models (factors) of the expression, so long as we have an efficient solver to find the DAG G inducing a good factorization. Moreover, we hope for a valid (in terms of being an I-map for the true distribution [Koller and Friedman, 2009]) yet simple G. Hence, in the next section, we show that solving (13) can be optimally decomposed into learning a set of probabilistic classifiers and learning an optimal DAG.

3.1 ALGORITHMIC SOLUTION

In order to solve (13), we first need to model the local conditional probability distributions:

$$\boldsymbol{p}_\theta\left(Y \,|\, \Delta_G^Y\right)\,, \forall G \in \mathcal{G}^3\,, \forall Y \in \mathbf{Y}\,. \tag{15}$$

Given G, for any $Y \in \mathbf{Y}$, let $\Delta_d^Y = \Delta_G^Y \setminus \mathbf{X}_c$ be the set of all discrete variables in Δ_G^Y. Let Π_d^Y be the set of all configurations of Δ_d^Y. Hence, each local distribution (15) is represented by $|\Pi_d^Y|$ distributions

$$\boldsymbol{p}_\theta\left(Y \,|\, \pi, \mathbf{X}_c\right)\,, \forall \pi \in \Pi_d^Y\,. \tag{16}$$

Thus, the optimization problem (13) becomes

$$(G^*, \theta^*) \in \underset{(G,\theta)\in\mathcal{P}^3}{\arg\max} \sum_{Y\in\mathbf{Y}} \sum_{\pi\in\Pi_d^Y} \log \prod_{(\boldsymbol{x},\boldsymbol{y})\in\mathcal{D}_\pi} \boldsymbol{p}_\theta\left(y \,|\, \pi, \boldsymbol{x}^c\right),$$

with $\mathcal{D}_\pi := \{(\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{D} | \pi_y^d = \pi\}$. A key point is the separation of discrete conditionals π and continuous conditionals $\boldsymbol{x}^c$. Moreover, we have $\max_{(G,\theta)\in\mathcal{P}^3} C(\boldsymbol{p}_\theta^G \,|\, \mathcal{D})$

$$= \max_{G\in\mathcal{G}^3} \sum_{Y\in\mathbf{Y}} \sum_{\pi\in\Pi_d^Y} \max_{\theta\in\Theta} C(\boldsymbol{p}_\theta \,|\, Y, \pi, \mathcal{D})\,, \tag{17}$$

where $C(\boldsymbol{p}_\theta \,|\, Y, \pi, \mathcal{D}) = \log \prod_{(\boldsymbol{x},\boldsymbol{y})\in\mathcal{D}_\pi} \boldsymbol{p}_\theta\left(y \,|\, \pi, \boldsymbol{x}^c\right)$. This means that we can reformulate the optimization problem (13) as a two-phase optimization problem: (P1) for any tuple $(Y, \pi) \in \mathbf{Y} \times \Pi_d^Y$ (considering the possible Δ_p^Y), learn the optimal parameter set θ^* of each distribution (16) which optimizes the local CLL function, i.e.,

$$\theta_{Y,\pi}^* \in \underset{\theta\in\Theta}{\arg\max}\, C(\boldsymbol{p}_\theta \,|\, Y, \pi, \mathcal{D}), \tag{18}$$

and then (P2) learn the best DAG $G^* \in \mathcal{G}^3$ which maximizes the CLL function: $G^* = \arg\max_G C(\boldsymbol{p}_{\theta^*}^G \,|\, \mathcal{D})$ and

$$C(\boldsymbol{p}_{\theta^*}^G \,|\, \mathcal{D}) = \sum_{Y\in\mathbf{Y}} \sum_{\pi\in\Pi_d^Y} C(\boldsymbol{p}_{\theta_{Y,\pi}^*} \,|\, Y, \pi, \mathcal{D}). \tag{19}$$

Problem (P1) can be solved for each tuple $(Y, \pi) \in \mathbf{Y} \times \Pi_d^Y$, for each possible $\Delta_p^Y \in \mathcal{F}^Y$ independently (where $\mathcal{F}^Y$ is a set of candidate parent sets for Y). (P2) can be cast as the structure learning for BNs, so we can leverage the research on that topic. The elephant in the room here is the size of $\mathcal{F}^Y$ (for each Y), which will be discussed in Section 3.4.

In this paper, we solve (19) using GOBNILP [Bartlett and Cussens, 2017, Cussens et al., 2017] which is a state-of-the-art globally optimal algorithm and can be easily adapted to handle regularized variants of CLL function as presented in Section 3.4. Intuitively, GOBNILP, which was designed for generative learning of Bayesian networks, can be instead used to reformulate the problem (P2) as learning a collection of parent sets $\{\Delta_d^Y : Y \in \mathbf{Y}\}$ which optimizes the CLL function (19) and together satisfy the DAG properties. It uses the local scores: $\forall Y \in \mathbf{Y}, \forall \Delta_p^Y \in \mathcal{F}^Y$:

$$C(Y, \Delta_d^Y) = \sum_{\pi\in\Pi_d^Y} C(\boldsymbol{p}_{\theta_{Y,\pi}^*} \,|\, Y, \pi, \mathcal{D})\,, \tag{20}$$

where we simplified the notation by removing θ and $\mathcal{D}$, since parameters have been already learned via (18) and data are fixed. Problem (P2) can be expressed as an Integer

Programming (IP) problem:

$$\text{Maximize} \sum_{Y \in \mathbf{Y}} \sum_{\Delta_d^Y \in \mathcal{F}^Y} \gamma(\Delta_d^Y) \cdot C(Y, \Delta_d^Y)\,, \tag{21}$$

$$\text{Subject to} \sum_{\Delta_d^Y \in \mathcal{F}^Y} \gamma(\Delta_d^Y) = 1\,, \forall Y \in \mathbf{Y}\,,$$

$$\sum_{Y \in \mathbf{Y}'} \sum_{\substack{\Delta_d^Y \in \mathcal{F}^Y \\ \Delta_d^Y \cap \mathbf{Y}' = \emptyset}} \gamma(\Delta_d^Y) > 1\,, \forall \mathbf{Y}' \subseteq \mathbf{Y}\,, |\mathbf{Y}'| > 1\,,$$

$$\gamma(\Delta_d^Y) \in \{0, 1\}\,, \forall Y \in \mathbf{Y}, \forall, \Delta_d^Y \in \mathcal{F}^Y\,.$$

The implementation is given in Algorithm 1, which returns a $(G^*, \theta^*) \in \mathcal{P}^3$ of (13). We call this type of model defined by (G^*, θ^*) a generalized Bayesian Network classifier (GBNC). Note that the loops starting in lines 2 and 3 can be easily parallelized since the local distributions (16) can be learned independently.

Algorithm 1 Learning a GBNC of (13)

1: **Input:** Data $\mathcal{D}$, Probabilistic hypothesis spaces.
2: **for** $Y \in \mathbf{Y}$ **do**
3: **for** $\Delta_d^Y \in \mathcal{F}^Y$ **do**
4: **for** $\pi \in \Pi_d^Y$ **do**
5: Solve (18) and store it in a proper data structure
6: **end for**
7: Compute $C(Y, \Delta_d^Y)$ by (20) using stored values
8: **end for**
9: **end for**
10: Find a best collection $\{\Delta_d^Y : Y \in \mathbf{Y}\}$ which optimizes (21) using GOBNILP
11: **Output:** A GBNC $(G^*, \theta^*) \in \mathcal{P}^3$ of (13)

3.2 REPRESENTATIONAL CAPACITY

To represent the joint conditional probability distribution $\boldsymbol{p}(\mathbf{Y} \mid \mathbf{X})$, we need a set of probabilistic classifiers $\boldsymbol{p}' : \mathcal{X}_c \longrightarrow \mathcal{Y}^k$ to estimate the local conditional probability distributions (16). Local models $\boldsymbol{p}'$ are trained with what we call base learners. Note that discrete variables are not included in the input for $\boldsymbol{p}'$ (they are dealt with through the DAG optimization), which also facilitates learning and representational capacity. First, it allows us to represent the distribution $\boldsymbol{p}(\mathbf{Y} \mid \mathbf{X})$ where $\mathbf{X}$ can contain both continuous features and discrete features without requiring any preprocessing transformation either before or during the training phase. We never face the problem of representing qualitative data for use as input as deep learning does [Hancock and Khoshgoftaar, 2020]. Besides, representing qualitative data for use as input is arguably the most critical obstacle for generalizing Classifier Chains (CCs) [Dembczyński et al., 2010, Read et al., 2021], which is a state-of-the-art multi-label classification framework, to cope with MDC. Moreover, we naturally overcome a bottleneck in the development of Multi-dimensional Bayesian network classifiers (MDBNCs) [Gil-Begue et al., 2021] that is a shortage of classifiers for the cases of continuous features, and mixed features.

Second, the probabilistic classifier inducing $\boldsymbol{p}'$ can be freely chosen according to our needs. It can be as intuitive as k-NN classifiers [Cover and Hart, 1967] and can be as counter-intuitive as ensembles of deep networks [Ganaie et al., 2021]. This allows us to employ sophisticated probabilistic classifiers to encode complex probabilistic relationships within $\boldsymbol{p}'_{Y,\pi} := \boldsymbol{p}_\theta\,(Y \mid \pi, \mathbf{X}_c), \forall \pi \in \Pi_d^Y$. For example, when each image is encoded using an $\boldsymbol{x}$, a convolutional network [LeCun et al., 2015] can be employed to encode $\boldsymbol{p}'_{Y,\pi}$. If one seeks for more accurate GBNCs, there should be no restriction on the use of ensemble learning methods, except the availability of computational resources. This flexibility of the framework is remarkably different from existing probabilistic MDC approaches [Gil-Begue et al., 2021, Jia and Zhang, 2022]. Roughly speaking, so long as you train good local models $\boldsymbol{p}'_{Y,\pi} : \mathcal{X}_c \longrightarrow \mathcal{Y}^k$ (for which you can use all toolsets available in the literature for "standard" single-class-variable classification), the framework in this paper does the rest to combine them optimally into an MDC solution.

3.3 INTERPRETABILITY

GBNCs are interpretable at both the population and individual levels. At the population level, the structure G provides a compact representation of the qualitative probabilistic relationships among feature and class variables. This graph representation is easy to interpret to end users when compared to an exponential number of masses provided by CP [Jia and Zhang, 2021] and the (infinitely) many joint conditional distributions associated with the set of marginal probability distributions provided by BR [Jia and Zhang, 2021]. At the individual level, the structure G and its parameters specified by θ under the particular value of an individual $\boldsymbol{x}$ form a compact representation of the qualitative and quantitative probabilistic relationships within $\boldsymbol{p}(\mathcal{Y} \mid \boldsymbol{x})$, which can be seen as a BN over the class variables.

3.4 REGULARIZATION

While Algorithm 1 helps to find an optimal GBNC which maximizes the CLL function, the next proposition suggests that this best GBNC may not always be the one we want, especially with regard to overfitting.

Proposition 3.4. *Assume local models have parameters optimally learned. Then* $\forall Y \in \mathbf{Y}$ *and* $\forall \Delta, \Delta' \in \mathcal{F}^Y$ *such that* $\Delta \subset \Delta'$*, we have*

$$C(Y, \Delta) \leq C(Y, \Delta')\,. \tag{22}$$

Therefore, at least one optimal solution of the Algorithm 1 is a fully connected DAG G*.*

Over-complex DAGs can happen frequently, especially when the local classifiers are learned without enforcing regularization terms. To seek for a better generalization, we propose a regularized variant of the CLL function:

$$S(\boldsymbol{p}_\theta^G \mid \mathcal{D}) = C(\boldsymbol{p}_\theta^G \mid \mathcal{D}) - \sum_{Y \in \mathbf{Y}} \text{pen}(\Delta_d^Y, |\mathcal{D}|)\,, \quad (23)$$

where $\text{pen}(\Delta_d^Y, |\mathcal{D}|)$ can be the penalty term of any decomposable scoring function [Liu et al., 2012]. Even a mild penalty can already help to reduce model complexity.

Algorithm 1 can be revised to learn GBNCs of regularized variants (23) as presented in Appendix C.1 and C.2. Moreover, as shown in Appendix C.2, pruning rules [de Campos et al., 2018] can be employed to find GBNCs which optimize regularized variants (23) without losing any optimality. This helps to greatly reduce the learning time because for each $Y \in \mathbf{Y}$, large candidate parent sets $\Delta_d^Y \in \mathcal{F}^Y$ are often pruned due to high penalties [de Campos et al., 2018]. Finally, for a very large number of class variables, it is not unreasonable to expect the treewidth of the true distribution to be limited, so that one can bound the size of $\mathcal{F}^Y$ and even employ bounded-treewidth learning [Nie et al., 2017].

4 INFERENCE

The learned function $\boldsymbol{p}$ (defined via G and θ) provides, given an $\boldsymbol{x} \in \mathcal{X}$, a conditional joint probability distribution $\boldsymbol{p}(\mathcal{Y}|\boldsymbol{x})$ which is used to find the Bayes-optimal prediction (BOP) $\hat{\boldsymbol{y}}$ w.r.t a target loss function $\ell : \mathcal{Y} \times \mathcal{Y} \longrightarrow \mathbb{R}_+$:

$$\hat{\boldsymbol{y}} := o(\boldsymbol{p}(\mathcal{Y}|\boldsymbol{x})) \in \operatorname*{argmin}_{\overline{\boldsymbol{y}} \in \mathcal{Y}} \sum_{\boldsymbol{y} \in \mathcal{Y}} \ell(\boldsymbol{y}, \overline{\boldsymbol{y}}) \boldsymbol{p}(\boldsymbol{y} \mid \boldsymbol{x})\,. \quad (24)$$

Yet, different loss functions may call for different BOPs (24) [Dembczyński et al., 2012, Gil-Begue et al., 2021, Nguyen and Hüllermeier, 2021, Waegeman et al., 2014]. Knowledge about the probability distribution $\boldsymbol{p}(\mathcal{Y}|\boldsymbol{x})$ is necessary for finding BOP (24) of any loss function. The complexity of finding BOP can greatly depend on the nature of the chosen loss function. This problem has been studied rarely in the MDC setting. An exception is [Bielza et al., 2011, Gil-Begue et al., 2021]. Notably, in these works, finding BOP (24) of some commonly used loss functions is shown to be equivalent to computing the most probable explanations (MPEs) of class variables when the classifier is an MDBNC. This is an interesting finding because it implies that the complexity of finding BOP (24) depends on the nature of both the chosen loss function and the classifier. While this finding allows us to directly employ any current/future developments on exact/approximate MPE inference [Gil-Begue et al., 2021] to find BOP (24) of some loss functions, one cannot get rid of the computational burden introduced by large numbers of features when working with MDBNCs.

In our framework, we can also show that finding BOP (24) of some loss functions is computing the MPEs of class variables. In the following, we describe the problem of finding BOP (24) of two commonly used loss functions[1] which are the *Hamming loss* (25) and the *subset 0/1 loss* (26):

$$\ell_H(\boldsymbol{y}, \hat{\boldsymbol{y}}) := \frac{1}{K} \sum_{k=1}^{K} [\![y^k \neq \hat{y}^k]\!]\,, \quad (25)$$

$$\ell_S(\boldsymbol{y}, \hat{\boldsymbol{y}}) := [\![\boldsymbol{y} \neq \hat{\boldsymbol{y}}]\!]\,. \quad (26)$$

The indicator $[\![A]\!]$ equals 1 if the A is true and 0 otherwise. Thus, both losses generalize the standard $0/1$ loss in binary classification. As noted in [Bielza et al., 2011], finding a BOP of ℓ_H and ℓ_S are respectively equivalent to finding K marginals (27) and equivalent to finding one MPE (28):

$$\hat{y}^k \in \operatorname*{argmax}_{\overline{y}^k \in \mathcal{Y}^k} \boldsymbol{p}(\overline{y}^k \mid \boldsymbol{x})\,, \forall k \in [K]\,, \quad (27)$$

$$\hat{\boldsymbol{y}} \in \operatorname*{argmax}_{\overline{\boldsymbol{y}} \in \mathcal{Y}} \boldsymbol{p}(\overline{\boldsymbol{y}} \mid \boldsymbol{x})\,. \quad (28)$$

Hence, the model does not require retraining to allow for different BOP. Exact MPE and marginal inferences are NP-hard problems [de Campos, 2020, Roth, 1996, Shimony, 1994]. However, in our framework, the complexity of MPE and marginal inferences only depend on the number of class variables. Thus, we do not encounter the computational burden introduced by large numbers of features, making the framework usable in practice in spite of that. Moreover, we can control the graph complexity among class variables by employing bounded-treewidth learning [Nie et al., 2017].

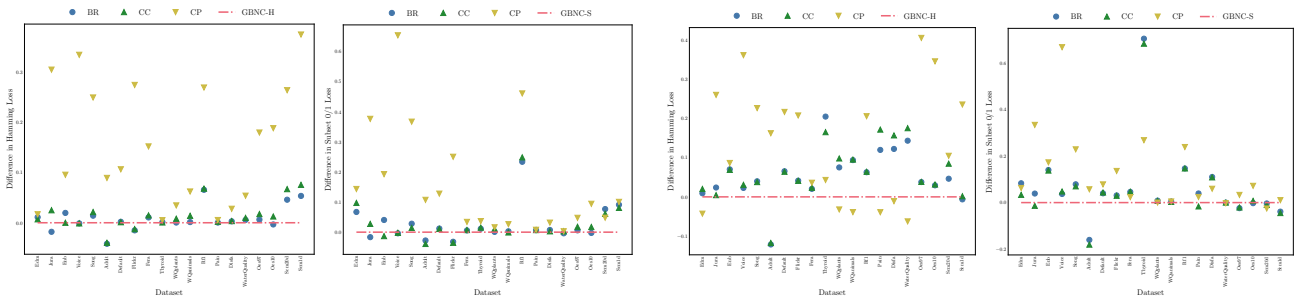
5 EXPERIMENTS

We compare two instantiations of GBNCs (GBNC-S which optimizes (23) and produces BOP (28) of ℓ_S, and GBNC-H which optimizes (23) and produces BOP (27) of ℓ_H) with three probabilistic competitors found in the literature on 20 tabular data sets [Jia and Zhang, 2022] and one image data set. To our best knowledge, this is the largest set of MDC data sets used in a single paper in the literature: the number of instances varies from 154 to 28779, the number of features varies from 10 to 1536, and the number of class variables varies from 2 to 16. It also contains 3 data sets with mixed discrete and continuous features.

We utilize an MDC version of the PASCAL VOC 2007 data set [Everingham et al., 2010]. We encode the objects found in that data set using 4 class variables: Person (Yes and No), Animal (No animal, Bird, Cat, Cow, Dog, Horse and Sheep), Vehicle (No vehicle, Aeroplane, Bicycle, Boat, Bus, Car, Motorbike, Train) and Indoor (No indoor object, Bottle, Chair, Dining table, Potted plant, Sofa, TV/Monitor).

For tabular data sets, we compare GBNCs with BR and PC [Jia and Zhang, 2022][Section II], and CC [Jia and Zhang,

[1] We defer intensive studies on finding BOPs of other loss functions [Gil-Begue et al., 2021][Section 4] to future work.



(a) Base learner: Logistic Regression.

(b) Base learner: Naive Bayes.

Figure 1: Performance differences to GBNCs (negative means better than GBNCs). Data sets (x-axis) are ordered by number of class variables.

2022][Section III]. Because of the limitations of competitors to deal with mixed data, we follow the suggestion of [Jia and Zhang, 2022] and convert discrete features/variables into continuous variables using one-hot encoding whenever they appear as parts of input of local classifiers of BR, PC and CC. Because we are not aware of any refinement of CC which can handle image data sets, we eliminate it from our comparison on the PASCAL VOC 2007. For tabular data sets, we use logistic regression (LR) [Menard, 2002] and Naive Bayes (NB) classifiers [Domingos and Pazzani, 1996, Zhang, 2004] to estimate the local distributions (16) (one can use more complex models, but as we see in the remainder, these choices already yield state-of-the-art results, so we decided that further tuning would go beyond our scope). For the image data set, distributions (16) are estimated using ResNet-18 [He et al., 2016] with the weights pre-trained on ImageNet [Deng et al., 2009], which are calibrated using temperature scaling [Guo et al., 2017]. Following the suggestion of [Zhang et al., 2017], we also employ *mixup* to improve the generalization of ResNet-18. Details of the experimental setting are given in Appendix D.1.

Overall, the results suggest the superiority of our framework against existing probabilistic MDC frameworks (See Table 1–3, and Figure 1). On the image data set, GBNCs indeed provide the most promising ℓ_H and ℓ_S (See Table 1).

Table 1: Results (mean $\pm$ std.) on the image data set.

Hamming loss (ℓ_H)		
GBNC-H	BR	CP
11.41 $\pm$ 0.35	12.51 $\pm$ 1.71	21.81 $\pm$ 7.62
Subset 0/1 loss (ℓ_S)		
GBNC-S	BR	CP
37.31 $\pm$ 0.84	41.57 $\pm$ 5.16	56.57 $\pm$ 13.28

GBNCs yield the best average ranks over the 20 tabular data sets, both for ℓ_H and ℓ_S. Furthermore, Friedman tests [Demšar, 2006] on the ranks yield small p-values, and strongly suggest performance differences between the classifiers. We also conduct Nemenyi post-hoc test [Nemenyi, 1963] and Conover post-hoc test [Conover, 1999, Conover and Iman, 1979] (see Table 3) to see if there are significant differences between pairs of classifiers. For each combination (among the 12 combinations) of competitor, loss and local models, we find at least one test where GBNCs is significantly better than that competitor in almost all cases.

Table 2: Average ranks and p-values of Friedman tests.

The cases of Hamming loss (ℓ_H)					
Learner	GBNC-H	BR	CC	CP	p-value
LR	**1.43**	1.98	2.60	4.00	**1.1e-09**
NB	**1.40**	2.70	2.95	2.95	**1.8e-04**
The cases of Subset 0/1 loss (ℓ_S)					
Learner	GBNC-S	BR	CC	CP	p-value
LR	**1.55**	2.20	2.38	3.88	**1.2e-07**
NB	**1.73**	2.80	2.28	3.20	**1.6e-03**

Even if the Nemenyi post-hoc test may be too conservative, has low power, and may not detect existing differences when Friedman's test rejects the null hypothesis (as elaborated in [Ulaş et al., 2012] and also elsewhere), it already informs significant differences. Table 3 suggests that the use of both LR and NB as local models (i.e. base learners) leads to improvements with respect to other approaches. Actually, LR performs better with more class variables, while NB with fewer (these differences can be appreciated in the Appendices). Yet, it is not the goal of this work to answer this question. The experiments with two different local models (LR and NB) have the purpose of demonstrating the capabilities of the overall idea.

Our experimental results are in agreement with the results found in literature. First, CC can hardly be a state-of-the-art MDC approach [Jia and Zhang, 2022]. Second, BR may provide competitive performance, especially when the number of class variables is not large [Wu and Zhu, 2020]. On the other hand, our experiments suggest a very interesting result that GBNC-H which estimates the joint conditional distribution and extracts marginal distributions using Defini-

Table 3: Post-hoc tests: p-values.

The cases of ℓ_H: p-values < 0.05 are given in bold				
H_0	Nemenyi		Conover	
	LR	NB	LR	NB
GBNC-H = BR	0.529	**0.008**	0.184	**0.002**
GBNC-H = CC	**0.021**	**0.001**	**0.006**	**3.9e-04**
GBNC-H = CP	**0.001**	**0.001**	**4.6e-08**	**3.9e-04**
BR = CP	**0.001**	0.9	**7.0e-07**	0.545
CC = CP	**0.003**	0.9	**0.001**	1
BR = CC	0.42	0.9	0.132	0.545
The cases of ℓ_S: p-values < 0.05 are given in bold				
H_0	Nemenyi		Conover	
	LR	NB	LR	NB
GBNC-S = BR	0.384	**0.042**	0.118	**0.01**
GBNC-S = CC	0.180	0.528	**0.049**	0.178
GBNC-S = CP	**0.001**	**0.002**	**4.9e-07**	**5.6e-04**
BR = CP	**0.001**	0.735	**1.4e-04**	0.326
CC = CP	**0.001**	0.106	**5.5e-04**	**0.026**
BR = CC	0.9	0.563	0.671	0.198

tions (2) often outperforms BR which directly estimates the marginal distributions. This suggests that capturing the dependency relationships can lead to more accurate estimates of the marginal probability distributions.

Although comparing ranks [Demšar, 2006] of classifiers is a common practice when one seeks short summaries of the performances, there is no golden rule about how the classifiers should be ranked. In this case, ranking the losses can not tell us whether there is any visible gain/loss. To gain more insights into the differences between classifiers, we make scatter plots for the losses provided by pairs of classifiers (See Figure 2–5 in Appendix D.2). In all cases, GBNC-H and GBNC-S are rarely worse than others with visible differences, and visible gains of GBNC-H and GBNC-S are observed in all cases. Again, those figures suggest that GBNC-H and GBNC-S can consistently provide promising performance. In practice, we would expect to see approaches which take into account dependencies among the class variables brings more advantages when the number of class variables K increases and the base learner is accurate. To show this ability of GBNCs, we make scatter plots for the losses provided by pairs of classifiers on 11 data sets with $K \geq 7$ with LR as the base learner (which is often more accurate than NB on these data sets). Figure 2 confirms that GBNCs indeed provide visible gains on these data sets.

Finally, we acknowledge that one can devise creative ideas to tackle MDC indirectly via other approaches, so one might ask to which extent our experiments yield state-of-the-art performance in a broader sense. We emphasize that our goal is to improve on MDC itself and to demonstrate the usefulness of this framework which has proven optimality properties and is very flexible to work with many other

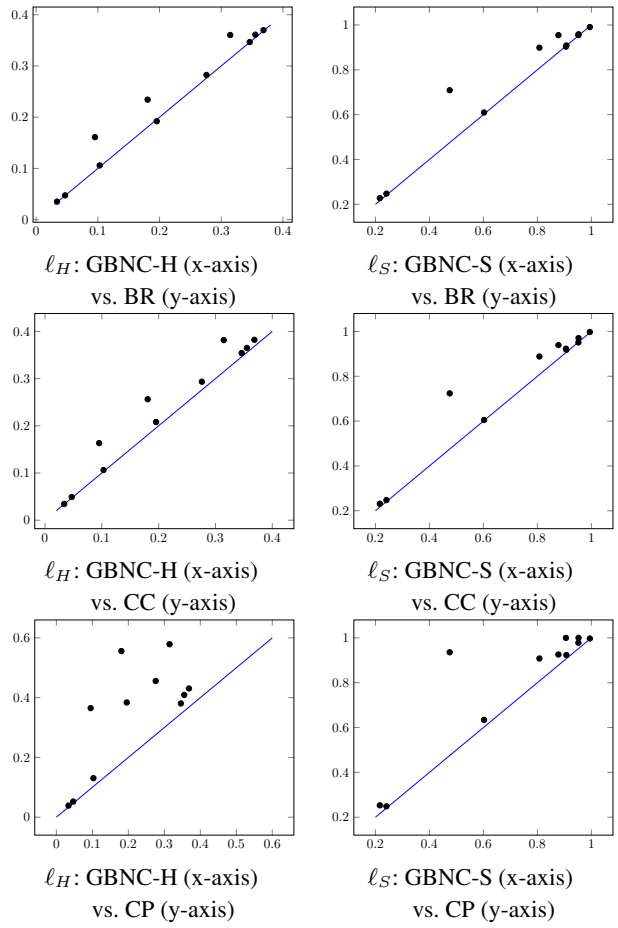


Figure 2: ℓ_H and ℓ_S with $K \geq 7$ (**base learner: *LR***)

(off-the-shelf) classifiers as internal local models (i.e. base learners). If one embraces the framework and chooses strong local models, this is likely (based on the theoretical results) to outperform any state of the art for MDC.

6 CONCLUSION

We propose a formal framework for probabilistic multi-dimensional classification (MDC) in which learning an optimal multi-dimensional classifier can be decomposed into learning a set of probabilistic classifiers and learning an optimal Bayesian network (BN) structure. We discuss how single-class-variable probabilistic classification and BN learning can be directly integrated into the framework with respect to optimality, representational capacity and scalability. We present algorithmic solutions for the learning and inference problems and discuss on their complexity. Finally, extensive experiments highlight the usefulness of the MDC framework. We hope that this paper can open doors for further research on all these strongly related topics.

References

Mark Bartlett and James Cussens. Integer linear programming for the bayesian network structure learning problem. *Artificial Intelligence*, 244:258–271, 2017.

J. M. Bernardo and A. F. M. Smith. *Bayesian Theory*. Chichester: Wiley, 1 edition, 2000.

Concha Bielza, Guangdi Li, and Pedro Larranaga. Multi-dimensional classification with bayesian networks. *International Journal of Approximate Reasoning*, 52(6): 705–727, 2011.

Guillaume Bouchard and Bill Triggs. The tradeoff between generative and discriminative classifiers. In *Proceedings of the 16th IASC International Symposium on Computational Statistics (COMPSTAT)*, pages 721–728, 2004.

William Jay Conover. *Practical nonparametric statistics*, volume 350. john wiley & sons, 1999.

William Jay Conover and Ronald L Iman. On multiple-comparisons procedures. *Los Alamos Sci. Lab. Tech. Rep. LA-7677-MS*, 1:14, 1979.

Thomas Cover and Peter Hart. Nearest neighbor pattern classification. *IEEE transactions on information theory*, 13(1):21–27, 1967.

James Cussens, Matti Järvisalo, Janne H Korhonen, and Mark Bartlett. Bayesian network structure learning with integer programming: Polytopes, facets and complexity. *Journal of Artificial Intelligence Research*, 58:185–229, 2017.

Cassio P de Campos. Almost no news on the complexity of map in bayesian networks. In *Proceedings of the 10th International Conference on Probabilistic Graphical Models (PGM)*, pages 149–160. PMLR, 2020.

Cassio P de Campos, Mauro Scanagatta, Giorgio Corani, and Marco Zaffalon. Entropy-based pruning for learning bayesian networks using bic. *Artificial Intelligence*, 260: 42–50, 2018.

Krzysztof Dembczyński, Weiwei Cheng, and Eyke Hüllermeier. Bayes optimal multilabel classification via probabilistic classifier chains. In *Proceedings of the 27th International Conference on International Conference on Machine Learning (ICML)*, pages 279–286, 2010.

Krzysztof Dembczyński, Willem Waegeman, Weiwei Cheng, and Eyke Hüllermeier. On label dependence and loss minimization in multi-label classification. *Machine Learning*, 88(1):5–45, 2012.

Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine learning research*, 7:1–30, 2006.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the 2009 IEEE conference on computer vision and pattern recognition (CVPR)*, pages 248–255. Ieee, 2009.

Pedro M Domingos and Michael J Pazzani. Beyond independence: conditions for the optimality of the simple bayesian classifier. In *Proceedings of the Thirteenth International Conference on Machine Learning (ICML)*, pages 105–112, 1996.

Charles Elkan. The foundations of cost-sensitive learning. In *Proceedings of the 17th international joint conference on Artificial intelligence (IJCAI)*, pages 973–978, 2001.

Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010.

Nir Friedman, Dan Geiger, and Moises Goldszmidt. Bayesian network classifiers. *Machine learning*, 29(2): 131–163, 1997.

Mudasir A Ganaie, Minghui Hu, et al. Ensemble deep learning: A review. *arXiv preprint arXiv:2104.02395*, 2021.

Santiago Gil-Begue, Concha Bielza, and Pedro Larrañaga. Multi-dimensional bayesian network classifiers: A survey. *Artificial Intelligence Review*, 54(1):519–559, 2021.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, pages 1321–1330, 2017.

John T Hancock and Taghi M Khoshgoftaar. Survey on categorical data for neural networks. *Journal of Big Data*, 7(1):1–41, 2020.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the 29th IEEE conference on computer vision and pattern recognition (CVPR)*, pages 770–778, 2016.

Bin-Bin Jia and Min-Ling Zhang. Decomposition-based classifier chains for multi-dimensional classification. *IEEE Transactions on Artificial Intelligence*, 3(2):176–191, 2021.

Bin-Bin Jia and Min-Ling Zhang. Multi-dimensional classification via selective feature augmentation. *Machine Intelligence Research*, 19(1):38–51, 2022.

Adrienne Kline, Hanyin Wang, Yikuan Li, Saya Dennis, Meghan Hutch, Zhenxing Xu, Fei Wang, Feixiong Cheng, and Yuan Luo. Multimodal machine learning in precision health: A scoping review. *npj Digital Medicine*, 5(1): 1–14, 2022.

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

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

Zhifa Liu, Brandon Malone, and Changhe Yuan. Empirical evaluation of scoring functions for bayesian network model selection. In *BMC bioinformatics*, volume 13, pages 1–16. Springer, 2012.

Scott Menard. *Applied logistic regression analysis*. Number 106. Sage, 2002.

Peter Bjorn Nemenyi. Distribution-free multiple comparisons. Master's thesis, Princeton University, United States, 1963.

Andrew Y Ng and Michael I Jordan. On discriminative vs. generative classifiers: a comparison of logistic regression and naive bayes. In *Proceedings of the 14th International Conference on Neural Information Processing Systems (NIPS)*, pages 841–848, 2001.

Vu-Linh Nguyen and Eyke Hüllermeier. Multilabel classification with partial abstention: Bayes-optimal prediction under label independence. *Journal of Artificial Intelligence Research*, 72:613–665, 2021.

Siqi Nie, Cassio P. de Campos, and Qiang Ji. Efficient learning of bayesian networks with bounded tree-width. *International Journal of Approximate Reasoning*, 80:412–427, 2017.

Dhanesh Ramachandram and Graham W Taylor. Deep multimodal learning: A survey on recent advances and trends. *IEEE signal processing magazine*, 34(6):96–108, 2017.

Jesse Read, Bernhard Pfahringer, Geoffrey Holmes, and Eibe Frank. Classifier chains: a review and perspectives. *Journal of Artificial Intelligence Research*, 70:683–718, 2021.

Teemu Roos, Hannes Wettig, Peter Grünwald, Petri Myllymäki, and Henry Tirri. On discriminative bayesian network classifiers and logistic regression. *Machine Learning*, 59(3):267–296, 2005.

Dan Roth. On the hardness of approximate reasoning. *Artificial Intelligence*, 82(1):273–302, 1996.

Maytal Saar-Tsechansky and Foster Provost. Handling missing values when applying classification models. *Journal of Machine Learning Research*, 8:1625–1657, 2007.

Hagit Shatkay, Fengxia Pan, Andrey Rzhetsky, and W John Wilbur. Multi-dimensional classification of biomedical text: Toward automated, practical provision of high-utility text to diverse users. *Bioinformatics*, 24(18):2086–2093, 2008.

Solomon Eyal Shimony. Finding maps for belief networks is np-hard. *Artificial intelligence*, 68(2):399–410, 1994.

Aydın Ulaş, Olcay Taner Yıldız, and Ethem Alpaydın. Cost-conscious comparison of supervised learning algorithms over multiple data sets. *Pattern Recognition*, 45(4):1772–1781, 2012.

Ilkay Ulusoy and Christopher M Bishop. Comparison of generative and discriminative techniques for object detection and classification. In *Toward Category-Level Object Recognition*, pages 173–195. Springer, 2006.

Willem Waegeman, Krzysztof Dembczyński, Arkadiusz Jachnik, Weiwei Cheng, and Eyke Hüllermeier. On the bayes-optimality of f-measure maximizers. *Journal of Machine Learning Research*, 15:3333–3388, 2014.

Guoqiang Wu and Jun Zhu. Multi-label classification: do hamming loss and subset accuracy really conflict with each other? In *Proceedings of the 34th International Conference on Neural Information Processing Systems (NeurIPS)*, pages 3130–3140, 2020.

Zhen Xu, David R So, and Andrew M Dai. Mufasa: Multimodal fusion architecture search for electronic health records. In *Proceedings of the Thirty-Fifth AAAI Conference on Artificial Intelligence (AAAI)*, volume 35, pages 10532–10540, 2021.

Harry Zhang. The optimality of naive bayes. In *Proceedings of the 17th International FLAIRS conference (FLAIRS)*, pages 562–567, 2004.

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *Proceedings of the 5th International Conference on Learning Representations (ICLR)*, 2017.