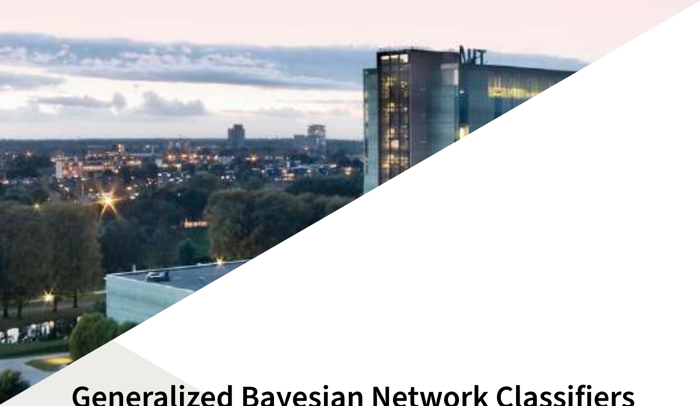


TU/e EINDHOVEN UNIVERSITY OF TECHNOLOGY

Generalized Bayesian Network Classifiers

Master’s Thesis Presentation

Yang Yang
Supervisors: Cassio de Campos, Vu-Linh Nguyen

November 5, 2022

Overview

1. Introduction
2. Background
3. Generalized Bayesian Network Classifiers (GBNCs)
4. Experiments
5. Conclusion

Introduction

Multi-Dimensional Classification (MDC)

In a nutshell

Multi-Dimensional Classification

- Each instance is characterized by **multiple** class variables $\mathbf{Y} = \{Y_1, \ldots, Y_d\}$ ($d \geq 2$).
- Each class variable can take **multiple** (≥ 2) values ($r_j = |\mathcal{Y}_j| \geq 2, \forall 1 \leq j \leq d$).

Multi-Dimensional Classification (MDC)

In a nutshell

Multi-Dimensional Classification

- Each instance is characterized by **multiple** class variables $\mathbf{Y} = \{Y_1, \ldots, Y_d\}$ $(d \geq 2)$.
- Each class variable can take **multiple** (≥ 2) values $(r_j = |\mathcal{Y}_j| \geq 2, \forall 1 \leq j \leq d)$.

A general MDC task

$$\underbrace{x_1, x_2, \ldots, x_m}_{\mathbf{x}} \xRightarrow{h(\mathbf{x})} \underbrace{y_1, y_2, \ldots, y_d}_{\mathbf{y}}$$

Multi-Dimensional Classification (MDC)

In a nutshell

Multi-Dimensional Classification

- Each instance is characterized by **multiple** class variables $\mathbf{Y} = \{Y_1, \ldots, Y_d\}$ $(d \geq 2)$.
- Each class variable can take **multiple** (≥ 2) values $(r_j = |\mathcal{Y}_j| \geq 2, \forall 1 \leq j \leq d)$.

A general MDC task

$$\underbrace{x_1, x_2, \ldots, x_m}_{\mathbf{x}} \xRightarrow{h(\mathbf{x})} \underbrace{y_1, y_2, \ldots, y_d}_{\mathbf{y}}$$

Multi-Dimensional Classification (MDC)

In a nutshell

Multi-Dimensional Classification

- Each instance is characterized by **multiple** class variables $\mathbf{Y} = \{Y_1, \ldots, Y_d\}$ $(d \geq 2)$.
- Each class variable can take **multiple** (≥ 2) values $(r_j = |\mathcal{Y}_j| \geq 2, \forall 1 \leq j \leq d)$.

A general MDC task

$$\underbrace{x_1, x_2, \ldots, x_m}_{\mathbf{x}} \xrightarrow{h(\mathbf{x})} \underbrace{y_1, y_2, \ldots, y_d}_{\mathbf{y}}$$

Feature Space	Class Space		
	Color	Brand	Type
	*White*	*BMW*	*Car*
	*White*	*Mercedes*	*Truck*
	*Yellow*	*Lamborghni*	*Supercar*
⋮	⋮	⋮	⋮
	*Red/White*	*Toyota*	*Offroad*
	?	?	?

Table 1: Multi-dimensional vehicle image classification. Adapted from [3].

Multi-Dimensional Classification (MDC)

While MDC is hard, it may become harder!

Multi-Dimensional Classification (MDC)

While MDC is hard, it may become harder!

Various types of features can coexist

- Numeric values.
  - $0.6, 2.36, 17.85, \ldots$
- Binary values.
  - $0, 1$
- Ordinal values.
  - Likert scale: Like (0), Like Somewhat (1), Neutral (2), Dislike Somewhat (3), Dislike (4)
- Non-ordinal discrete signals.
  - Color: Red, Blue, Yellow, Green, ...
  - Gender: Male, Female, ...

Multi-Dimensional Classification (MDC)

While MDC is hard, it may become harder!

Various types of features can coexist

- Numeric values.
  - $0.6, 2.36, 17.85, \ldots$
- Binary values.
  - $0, 1$
- Ordinal values.
  - Likert scale: Like (0), Like Somewhat (1), Neutral (2), Dislike Somewhat (3), Dislike (4)
- Non-ordinal discrete signals.
  - Color: Red, Blue, Yellow, Green, . . .
  - Gender: Male, Female, . . .

World is **multi-modal**!

Figure 1: "A cat is playing with a dog."

Probabilistic MDC

A General MDC Task

$$\underbrace{x_1, x_2, \ldots, x_m}_{\mathbf{x}} \xRightarrow{h(\mathbf{x})} \underbrace{y_1, y_2, \ldots, y_d}_{\mathbf{y}}$$

A Probabilistic MDC Task

$$\underbrace{x_1, x_2, \ldots, x_m}_{\mathbf{x}}$$

$$\xRightarrow{\text{Modeling}} \quad P(\mathbf{Y} \mid \mathbf{X})$$

$$\xRightarrow{\text{Inference}} \quad \underbrace{y_1, y_2, \ldots, y_d}_{\mathbf{y}}$$

Probabilistic MDC

A General MDC Task

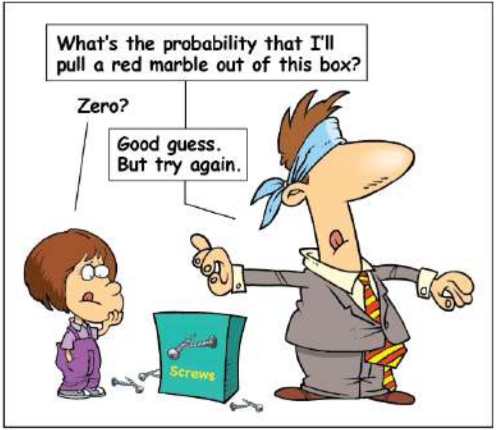
$$\underbrace{x_1, x_2, \ldots, x_m}_{\mathbf{x}} \xRightarrow{h(\mathbf{x})} \underbrace{y_1, y_2, \ldots, y_d}_{\mathbf{y}}$$

A Probabilistic MDC Task

$$\underbrace{x_1, x_2, \ldots, x_m}_{\mathbf{x}}$$

$$\xRightarrow{\text{Modeling}} P(\mathbf{Y} \mid \mathbf{X})$$

$$\xRightarrow{\text{Inference}} \underbrace{y_1, y_2, \ldots, y_d}_{\mathbf{y}}$$



Probabilistic MDC

A General MDC Task

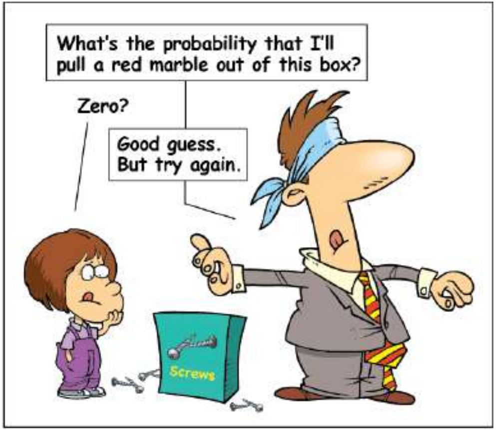
$$\underbrace{x_1, x_2, \ldots, x_m}_{\mathbf{x}} \overset{h(\mathbf{x})}{\Longrightarrow} \underbrace{y_1, y_2, \ldots, y_d}_{\mathbf{y}}$$

A Probabilistic MDC Task

$$\underbrace{x_1, x_2, \ldots, x_m}_{\mathbf{x}}$$

$$\xRightarrow{\text{Modeling}} P(\mathbf{Y} \mid \mathbf{X})$$

$$\xRightarrow{\text{Inference}} \underbrace{y_1, y_2, \ldots, y_d}_{\mathbf{y}}$$



- Probabilities associated to predictions.
- Optimal predictions under different loss functions.
- How to capture the probabilistic relationships between class variables?

Background

Bayesian Network Classifiers (BNCs)

What is a Bayesian network (BN)?

Bayesian Network Classifiers (BNCs)

What is a Bayesian network (BN)?

$\mathcal{B} = (\mathcal{G}, \mathbf{\Theta})$

- $\mathcal{G} = (\mathbf{V}, \mathbf{E})$ is a DAG with $\mathbf{V}$ a collection of nodes associated to random variables (RVs) $\mathbf{X}$.
- $\mathbf{\Theta} = \{\Theta_i \mid 1 \leq i \leq m\}$ is a collection of parameters
  - encodes local conditional probability distributions (CPDs) $\{P(X_i \mid \mathbf{Pa}(X_i)) \mid 1 \leq i \leq m\}$ of $\mathbf{X}$.

Bayesian Network Classifiers (BNCs)

What is a Bayesian network (BN)?

$\mathcal{B} = (\mathcal{G}, \boldsymbol{\Theta})$

- $\mathcal{G} = (\mathbf{V}, \mathbf{E})$ is a DAG with $\mathbf{V}$ a collection of nodes associated to random variables (RVs) $\mathbf{X}$.
- $\boldsymbol{\Theta} = \{\Theta_i \mid 1 \leq i \leq m\}$ is a collection of parameters
  - encodes local conditional probability distributions (CPDs) $\{P(X_i \mid \mathbf{Pa}(X_i)) \mid 1 \leq i \leq m\}$ of $\mathbf{X}$.
- $\mathcal{B}$ represents a joint probability distribution over $\mathbf{X}$.

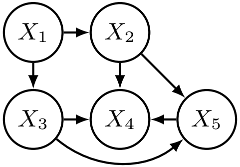


Figure 2: $P_{\mathcal{B}} = P(X_1)P(X_2 \mid X_1)P(X_3 \mid X_1)P(X_4 \mid X_2, X_3, X_5)P(X_5 \mid X_2, X_3)$

Bayesian Network Classifiers (BNCs)

Why using a Bayesian network (BN)?

Bayesian Network Classifiers (BNCs)

Why using a Bayesian network (BN)?

Advantages of Bayesian Networks (BNs)

- Intuitive graphical formalisms of probabilistic relationships.
- Interpretable representations of uncertainties supported by probability theory.

Bayesian Network Classifiers (BNCs)

Why using a Bayesian network (BN)?

Advantages of Bayesian Networks (BNs)

- Intuitive graphical formalisms of probabilistic relationships.
- Interpretable representations of uncertainties supported by probability theory.

(Multi-dimensional) BNCs are simply BNs applied to (multi-dimensional) classification problems.

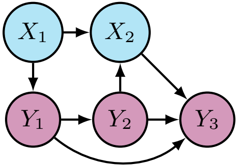


Figure 3: An example (M)BNC over $\mathbf{X} = \{X_1, X_2\}$ and $\mathbf{Y} = \{Y_1, Y_2, Y_3\}$.

Learning an (M)BNC

Learning an (M)BNC is essentially learning the underlying Bayesian network.

Learning an (M)BNC

Learning an (M)BNC is essentially learning the underlying Bayesian network.

Structure Learning: Learning $\mathcal{G}$ from data $\mathcal{D}$

- Constraint-based
  - Employ statistical tests to identify independencies between variables.
- Score-based
  - Define a structure score metric (such as BD scores and the BIC score).
  - Find a structure achieving the maximum score from the structure space.

Learning an (M)BNC

Learning an (M)BNC is essentially learning the underlying Bayesian network.

Structure Learning: Learning $\mathcal{G}$ from data $\mathcal{D}$

- Constraint-based
  - Employ statistical tests to identify independencies between variables.
- Score-based
  - Define a structure score metric (such as BD scores and the BIC score).
  - Find a structure achieving the maximum score from the structure space.

Parameter Learning: Learning $\mathbf{\Theta}$ from data $\mathcal{D}$

- Bayesian Learning
  - Treat parameters as random variables and update $\mathbf{\Theta}$ by using Bayes' rule: $p(\mathbf{\Theta} \mid \mathcal{D}) = \frac{p(\mathcal{D}|\mathbf{\Theta})p(\mathbf{\Theta})}{p(\mathcal{D})}$.
- Maximum Likelihood Estimation (MLE)
  - Pick parameters that maximize the model's probability of generating $\mathcal{D}$.
  - Given enough data, uncover the real data-generating distribution $P(\mathbf{X}, \mathbf{Y})$.

Learning an (M)BNC

Maximum Likelihood Estimation (MLE)

- Pick parameters to make the model to be close to the data-generating distribution $P(\mathbf{X}, \mathbf{Y})$.

Learning an (M)BNC

Maximum Likelihood Estimation (MLE)

- Pick parameters to make the model to be close to the data-generating distribution $P(\mathbf{X}, \mathbf{Y})$.

However, classification is a **discriminative** or **supervised** task.

In an (M)BNC, what we need is the conditional distribution $P(\mathbf{Y} \mid \mathbf{X})$ of class variables.

Learning an (M)BNC

Maximum Likelihood Estimation (MLE)

- Pick parameters to make the model to be close to the data-generating distribution $P(\mathbf{X}, \mathbf{Y})$.

However, classification is a **discriminative** or **supervised** task.

In an (M)BNC, what we need is the conditional distribution $P(\mathbf{Y} \mid \mathbf{X})$ of class variables.

Maximizing the Conditional Log-Likelihood (CLL)

Given data $\mathcal{D} = \{(\mathbf{x}^l, \mathbf{y}^l) \mid 1 \leq l \leq N\}$, the CLL is a discriminative objective function:

$$\mathrm{CLL}(\mathcal{G}, \boldsymbol{\Theta} \mid \mathcal{D}) := \log \prod_{l=1}^{N} p_{\mathcal{B}}(\mathbf{y}^l \mid \mathbf{x}^l) = \log \prod_{l=1}^{N} \frac{f_{\mathcal{B}}(\mathbf{x}^l, \mathbf{y}^l)}{\sum_{\mathbf{y} \in \mathcal{Y}} f_{\mathcal{B}}(\mathbf{x}^l, \mathbf{y}^l)} \tag{1}$$

Learning an (M)BNC

Maximum Likelihood Estimation (MLE)

- Pick parameters to make the model to be close to the data-generating distribution $P(\mathbf{X}, \mathbf{Y})$.

However, classification is a **discriminative** or **supervised** task.

In an (M)BNC, what we need is the conditional distribution $P(\mathbf{Y} \mid \mathbf{X})$ of class variables.

Maximizing the Conditional Log-Likelihood (CLL)

Given data $\mathcal{D} = \{(\mathbf{x}^l, \mathbf{y}^l) \mid 1 \leq l \leq N\}$, the CLL is a discriminative objective function:

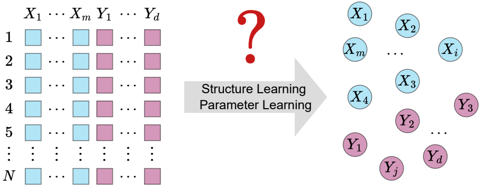
$$\mathrm{CLL}(\mathcal{G}, \boldsymbol{\Theta} \mid \mathcal{D}) := \log \prod_{l=1}^{N} p_{\mathcal{B}}(\mathbf{y}^l \mid \mathbf{x}^l) = \log \prod_{l=1}^{N} \frac{f_{\mathcal{B}}(\mathbf{x}^l, \mathbf{y}^l)}{\sum_{\mathbf{y} \in \mathcal{Y}} f_{\mathcal{B}}(\mathbf{x}^l, \mathbf{y}^l)} \tag{1}$$

- Unfortunately, CLL does not decompose over $\mathcal{G}$ into a separate term for each variable.
- There is no closed-form solution for optimizing parameters to maximize the CLL.

Generalized Bayesian Network Classifiers (GBNCs)

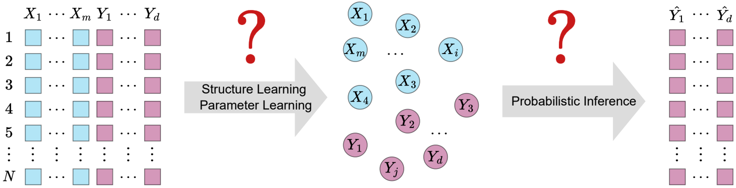
Research Questions

- How to discriminatively learn the parameters and the structure of a BNC?



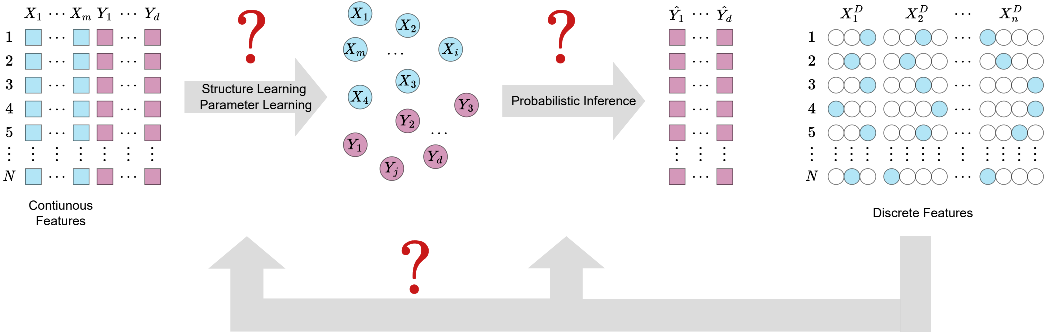
Research Questions

- How to discriminatively learn the parameters and the structure of a BNC?
- How to perform probabilistic inference to compute optimal predictions under different loss functions?



Research Questions

- How to discriminatively learn the parameters and the structure of a BNC?
- How to perform probabilistic inference to compute optimal predictions under different loss functions?
- How to handle mixed data, i.e., continuous and discrete feature variables coexist?



Discriminative Parameter Learning

Discriminative Parameter Learning

To enable the CLL decomposability

Structural constraints in GBNCs

- There is no directed edge from class variables to feature variables.
- There is a directed edge from any feature variable to any class variable.

Discriminative Parameter Learning

To enable the CLL decomposability

Structural constraints in GBNCs

- There is no directed edge from class variables to feature variables.
- There is a directed edge from any feature variable to any class variable.

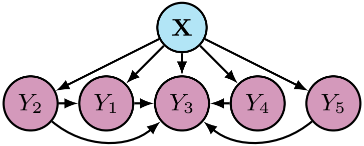


Figure 4: An example GBNC over $\mathbf{X}$ and $\mathbf{Y} = \{Y_1, Y_2, Y_3, Y_4, Y_5\}$.

Discriminative Parameter Learning

Proposition 1: CLL Decomposition

Under our structural constraints, the CLL can be simpilified as

$$\mathrm{CLL}(\mathcal{G}, \boldsymbol{\Theta} \mid \mathcal{D}) = \sum_{l=1}^{N} \sum_{j=1}^{d} \log P_{\mathcal{B}}(y_j^l \mid \mathbf{pa}(y_j)^l) \tag{2}$$

- $\boldsymbol{\Pi}_j = \mathbf{Pa}(Y_j) \cap \mathbf{Y}$, with the number of configurations of $\boldsymbol{\Pi}_j$ as q_j.
- $\boldsymbol{\Phi}_j = \mathbf{Pa}(Y_j) \cap \mathbf{X}$.

The CLL can be further simplified as

$$\mathrm{CLL}(\mathcal{G}, \boldsymbol{\Theta} \mid \mathcal{D}) = \sum_{l=1}^{N} \sum_{j=1}^{d} \sum_{\substack{k=1: \\ \mathbb{1}_{\boldsymbol{\pi}_j^l = \boldsymbol{\pi}_{jk}}}}^{q_j} \log P_j^{\boldsymbol{\pi}_{jk}}(y_j^l \mid \boldsymbol{\phi}_j^l) \tag{3}$$

Discriminative Parameter Learning

$$\mathrm{CLL}(\mathcal{G}, \boldsymbol{\Theta} \mid \mathcal{D}) = \sum_{l=1}^{N} \sum_{j=1}^{d} \sum_{\substack{k=1: \\ \mathbb{1}_{\boldsymbol{\pi}_j^l = \boldsymbol{\pi}_{jk}}}}^{q_j} \log P_j^{\boldsymbol{\pi}_{jk}}(y_j^l \mid \boldsymbol{\phi}_j^l) \tag{4}$$

- In words, we will need $q = q_1 + \cdots + q_d$ probabilistic models to represent the distribution $P_{\mathcal{B}}(\mathbf{Y} \mid \mathbf{X})$.
  - One for each $P_j^{\boldsymbol{\pi}_{jk}}(Y_j \mid \boldsymbol{\Phi}_j)$ $(1 \leq j \leq d, 1 \leq k \leq q_j)$.

Discriminative Parameter Learning

$$\mathrm{CLL}(\mathcal{G}, \mathbf{\Theta} \mid \mathcal{D}) = \sum_{l=1}^{N} \sum_{j=1}^{d} \sum_{\substack{k=1: \\ \mathbb{1}_{\boldsymbol{\pi}_j^l = \boldsymbol{\pi}_{jk}}}}^{q_j} \log P_j^{\boldsymbol{\pi}_{jk}}(y_j^l \mid \boldsymbol{\phi}_j^l) \tag{4}$$

- In words, we will need $q = q_1 + \cdots + q_d$ probabilistic models to represent the distribution $P_{\mathcal{B}}(\mathbf{Y} \mid \mathbf{X})$.
  - One for each $P_j^{\boldsymbol{\pi}_{jk}}(Y_j \mid \mathbf{\Phi}_j)$ $(1 \leq j \leq d, 1 \leq k \leq q_j)$.
- The probabilistic relationships between feature variables do not affect the CLL.

Discriminative Parameter Learning

$$\mathrm{CLL}(\mathcal{G}, \boldsymbol{\Theta} \mid \mathcal{D}) = \sum_{l=1}^{N} \sum_{j=1}^{d} \sum_{\substack{k=1: \\ \mathbb{1}_{\boldsymbol{\pi}_j^l = \boldsymbol{\pi}_{jk}}}}^{q_j} \log P_j^{\boldsymbol{\pi}_{jk}}(y_j^l \mid \boldsymbol{\phi}_j^l) \tag{4}$$

- In words, we will need $q = q_1 + \cdots + q_d$ probabilistic models to represent the distribution $P_{\mathcal{B}}(\mathbf{Y} \mid \mathbf{X})$.
  - One for each $P_j^{\boldsymbol{\pi}_{jk}}(Y_j \mid \boldsymbol{\Phi}_j)$ $(1 \leq j \leq d, 1 \leq k \leq q_j)$.
- The probabilistic relationships between feature variables do not affect the CLL.
- Although we assume any $Y \in \mathbf{Y}$ is connected to all $X \in \mathbf{X}$

Discriminative Parameter Learning

$$\mathrm{CLL}(\mathcal{G}, \boldsymbol{\Theta} \mid \mathcal{D}) = \sum_{l=1}^{N} \sum_{j=1}^{d} \sum_{\substack{k=1: \\ \mathbb{1}_{\boldsymbol{\pi}_j^l = \boldsymbol{\pi}_{jk}}}}^{q_j} \log P_j^{\boldsymbol{\pi}_{jk}}(y_j^l \mid \boldsymbol{\phi}_j^l) \tag{4}$$

- In words, we will need $q = q_1 + \cdots + q_d$ probabilistic models to represent the distribution $P_{\mathcal{B}}(\mathbf{Y} \mid \mathbf{X})$.
  - One for each $P_j^{\boldsymbol{\pi}_{jk}}(Y_j \mid \boldsymbol{\Phi}_j)$ $(1 \leq j \leq d, 1 \leq k \leq q_j)$.
- The probabilistic relationships between feature variables do not affect the CLL.
- Although we assume any $Y \in \mathbf{Y}$ is connected to all $X \in \mathbf{X}$
  - Learning a base probabilistic model $\mathsf{C}_j^{\boldsymbol{\pi}_{jk}}$ for each $P_j^{\boldsymbol{\pi}_{jk}}(y_j \mid \phi_j)$ is essentially a feature selection!

Discriminative Parameter Learning: Input Space Partitioning

How do we learn a base probabilistic model $\mathsf{C}_j^{\boldsymbol{\pi}_{jk}}$?

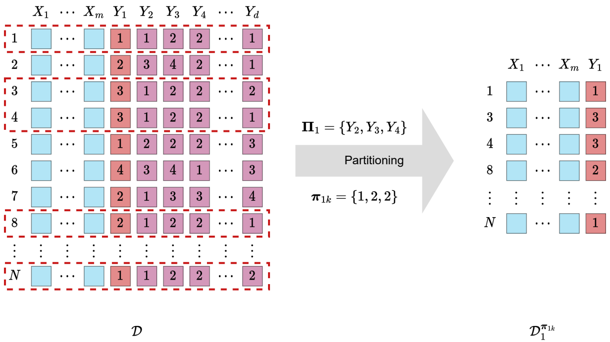
	X_1	$\cdots$	X_m	Y_1	Y_2	Y_3	Y_4	$\cdots$	Y_d
1		$\cdots$		1	1	2	2	$\cdots$	1
2		$\cdots$		2	3	4	2	$\cdots$	1
3		$\cdots$		3	1	2	2	$\cdots$	2
4		$\cdots$		3	1	2	2	$\cdots$	1
5		$\cdots$		1	2	2	2	$\cdots$	3
6		$\cdots$		4	3	4	1	$\cdots$	3
7		$\cdots$		2	1	3	3	$\cdots$	4
8		$\cdots$		2	1	2	2	$\cdots$	1
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
N		$\cdots$		1	1	2	2	$\cdots$	2

$\mathcal{D}$

Discriminative Parameter Learning: Input Space Partitioning

How do we learn a base probabilistic model $\mathsf{C}_j^{\boldsymbol{\pi}_{jk}}$?

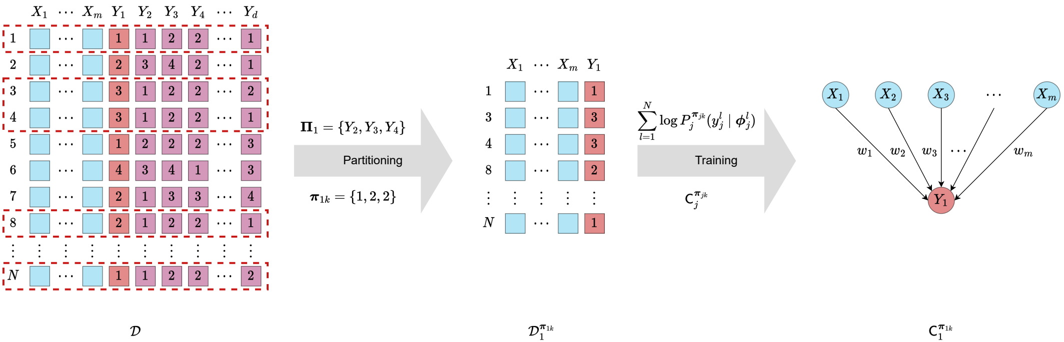
- Extract $\mathcal{D}_j^{\boldsymbol{\pi}_{jk}}$ from $\mathcal{D}$ according to $\boldsymbol{\pi}_{jk}$.



Discriminative Parameter Learning: Input Space Partitioning

How do we learn a base probabilistic model $\mathsf{C}_j^{\boldsymbol{\pi}_{jk}}$?

- Extract $\mathcal{D}_j^{\boldsymbol{\pi}_{jk}}$ from $\mathcal{D}$ according to $\boldsymbol{\pi}_{jk}$.
- Train $\mathsf{C}_j^{\boldsymbol{\pi}_{jk}}$ using $\mathcal{D}_j^{\boldsymbol{\pi}_{jk}}$ by iteratively optimizing $\sum_{l=1}^{N} \log P_j^{\boldsymbol{\pi}_{jk}}(y_j^l \mid \boldsymbol{\phi}_j^l)$.



Discriminative Structure Learning

Discriminative Structure Learning

The probabilistic relationships between feature variables do not affect the CLL.

- Learning $\mathcal{G}$ is essentially learning the class subgraph $\mathcal{G}_{\mathbf{Y}}$.

Discriminative Structure Learning

The probabilistic relationships between feature variables do not affect the CLL.

- Learning $\mathcal{G}$ is essentially learning the class subgraph $\mathcal{G}_{\mathbf{Y}}$.

A score-based structure learning approach

- Use a penalized CLL score as a (decomposable) structure score function:

$$
\begin{aligned}
S(\mathcal{G}, \boldsymbol{\Theta} \mid \mathcal{D}) &= \mathrm{CLL}(\mathcal{G}, \boldsymbol{\Theta} \mid \mathcal{D}) + \mathrm{PEN}(\mathcal{G}, \boldsymbol{\Theta} \mid \mathcal{D}) \\
&= \sum_{j=1}^{d} (\mathrm{CLL}(\mathcal{G}_{\mathbf{\Pi}_j}, \Theta_j \mid \mathcal{D}) + \mathrm{PEN}(\mathcal{G}_{\mathbf{\Pi}_j}, \Theta_j \mid \mathcal{D})) \\
&= \sum_{j=1}^{d} S(\mathcal{G}_{\mathbf{\Pi}_j}, \Theta_j \mid \mathcal{D})
\end{aligned} \tag{5}
$$

where $\mathrm{PEN}(\mathcal{G}_{\mathbf{\Pi}_j}, \Theta_j \mid \mathcal{D}) = -\frac{\log N}{2}(r_j - 1)q_j$.

Discriminative Structure Learning

The probabilistic relationships between feature variables do not affect the CLL.

- Learning $\mathcal{G}$ is essentially learning the class subgraph $\mathcal{G}_{\mathbf{Y}}$.

A score-based structure learning approach

- Use a penalized CLL score as a (decomposable) structure score function:

$$\begin{aligned} S(\mathcal{G}, \boldsymbol{\Theta} \mid \mathcal{D}) &= \mathrm{CLL}(\mathcal{G}, \boldsymbol{\Theta} \mid \mathcal{D}) + \mathrm{PEN}(\mathcal{G}, \boldsymbol{\Theta} \mid \mathcal{D}) \\ &= \sum_{j=1}^{d} (\mathrm{CLL}(\mathcal{G}_{\mathbf{\Pi}_j}, \Theta_j \mid \mathcal{D}) + \mathrm{PEN}(\mathcal{G}_{\mathbf{\Pi}_j}, \Theta_j \mid \mathcal{D})) \\ &= \sum_{j=1}^{d} S(\mathcal{G}_{\mathbf{\Pi}_j}, \Theta_j \mid \mathcal{D}) \end{aligned} \tag{5}$$

where $\mathrm{PEN}(\mathcal{G}_{\mathbf{\Pi}_j}, \Theta_j \mid \mathcal{D}) = -\frac{\log N}{2}(r_j - 1)q_j$.

- We can further prune the search space of potential $\mathbf{\Pi}_j$ for Y_j by using rules from [2].

Mixed Data

Continuous and discrete feature variables coexist.

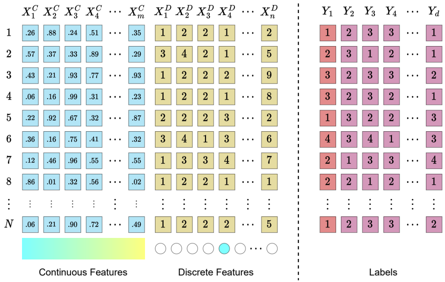
	X_1^C	X_2^C	X_3^C	X_4^C	$\cdots$	X_m^C	X_1^D	X_2^D	X_3^D	X_4^D	$\cdots$	X_n^D	Y_1	Y_2	Y_3	Y_4	$\cdots$	Y_d
1	.26	.88	.24	.51	$\cdots$	.35	1	2	2	1	$\cdots$	2	1	2	3	3	$\cdots$	1
2	.57	.37	.33	.89	$\cdots$	.29	3	4	2	1	$\cdots$	5	2	3	1	2	$\cdots$	1
3	.43	.21	.93	.77	$\cdots$	.93	1	2	2	2	$\cdots$	9	3	2	3	3	$\cdots$	2
4	.06	.16	.99	.31	$\cdots$	.23	1	2	2	1	$\cdots$	8	3	2	3	2	$\cdots$	1
5	.22	.92	.67	.32	$\cdots$	.87	2	2	2	3	$\cdots$	2	1	3	2	2	$\cdots$	3
6	.36	.16	.75	.41	$\cdots$	.32	3	4	1	3	$\cdots$	6	4	3	4	1	$\cdots$	3
7	.12	.46	.96	.55	$\cdots$	.55	1	3	3	4	$\cdots$	7	2	1	3	3	$\cdots$	4
8	.86	.01	.32	.56	$\cdots$	.02	1	2	2	1	$\cdots$	1	2	2	1	2	$\cdots$	1
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
N	.06	.21	.90	.72	$\cdots$	.49	1	2	2	2	$\cdots$	5	1	2	3	3	$\cdots$	2

Continuous Features — Discrete Features — Labels

Mixed Data

Continuous and discrete feature variables coexist.

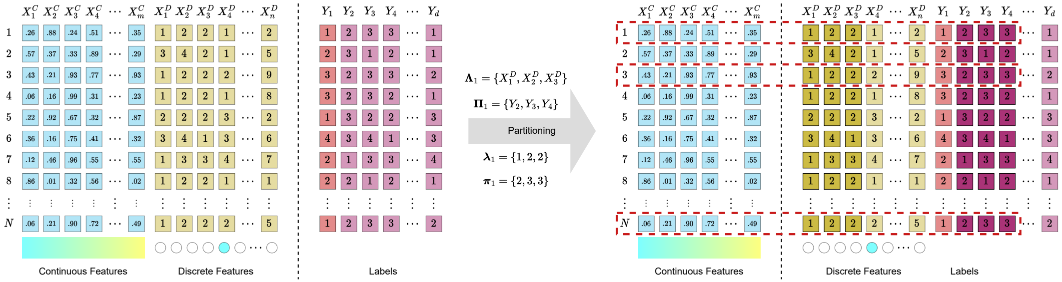
- How to handle discrete features?



Mixed Data

Continuous and discrete feature variables coexist.

- How to handle discrete features?
- $\mathbf{\Lambda}_j = \mathbf{Pa}(Y_j) \cap \mathbf{X}^D.$
- Treat discrete feature variables as "special" class variables.
- Use discrete features to do further input space partitioning!



Inference in GBNCs

Inference in GBNCs

Given a loss function $\ell : \boldsymbol{\mathcal{Y}} \times \boldsymbol{\mathcal{Y}} \to \mathbb{R}_+$ and an input $\mathbf{x} \in \boldsymbol{\mathcal{X}}$, the Bayes-Optimal Prediction (BOP) $\hat{\mathbf{y}}$ is

$$\hat{\mathbf{y}} = \arg\min_{\mathbf{y}' \in \boldsymbol{\mathcal{Y}}} \sum_{\mathbf{y} \in \boldsymbol{\mathcal{Y}}} \ell(\mathbf{y}, \mathbf{y}') P(\mathbf{y} \mid \mathbf{x}) \tag{6}$$

Inference in GBNCs

Given a loss function $\ell : \boldsymbol{\mathcal{Y}} \times \boldsymbol{\mathcal{Y}} \to \mathbb{R}_+$ and an input $\mathbf{x} \in \boldsymbol{\mathcal{X}}$, the Bayes-Optimal Prediction (BOP) $\hat{\mathbf{y}}$ is

$$\hat{\mathbf{y}} = \operatorname*{arg\,min}_{\mathbf{y}' \in \boldsymbol{\mathcal{Y}}} \sum_{\mathbf{y} \in \boldsymbol{\mathcal{Y}}} \ell(\mathbf{y}, \mathbf{y}') P(\mathbf{y} \mid \mathbf{x}) \tag{6}$$

Subset 0/1 Loss: $\ell_{0/1}(\mathbf{y}, \hat{\mathbf{y}}) = \mathbb{1}_{\mathbf{y} \neq \hat{\mathbf{y}}}$

Most Probable Explanation (MPE) Inference for $\mathbf{Y}$.

$$\hat{\mathbf{y}} = \operatorname*{arg\,min}_{\mathbf{y}' \in \boldsymbol{\mathcal{Y}}} \sum_{\mathbf{y} \in \boldsymbol{\mathcal{Y}}} \mathbb{1}_{\mathbf{y} \neq \mathbf{y}'} P(\mathbf{y} \mid \mathbf{x}) = \operatorname*{arg\,max}_{\mathbf{y}' \in \boldsymbol{\mathcal{Y}}} P(\mathbf{y}' \mid \mathbf{x}) \tag{7}$$

Hamming Loss: $\ell_{\mathrm{HL}}(\mathbf{y}, \hat{\mathbf{y}}) = \sum_{j=1}^{d} \mathbb{1}_{y_j \neq \hat{y}_j}$

Maximum A Posteriori (MAP) Inference for each $Y \in \mathbf{Y}$.

$$\hat{\mathbf{y}} = \{\hat{y}_j \mid 1 \leq j \leq d\}$$
$$\text{where, } \hat{y}_j = \operatorname*{arg\,min}_{y' \in \mathcal{Y}_j} \sum_{y \in \mathcal{Y}_j} \mathbb{1}_{y \neq y'} P(y \mid \mathbf{x}) = \operatorname*{arg\,max}_{y' \in \mathcal{Y}_j} P(y' \mid \mathbf{x}) \tag{8}$$

Inference in GBNCs

Algorithm 1 BOP under the subset 0/1 loss

Require: Test set $\mathcal{T} = \{\mathbf{x}^l \mid 1 \leq l \leq L\}$, local probabilistic classifiers **C**, class subgraph $\mathcal{G}_{\mathbf{Y}}$

Ensure: The set of predictions $\mathcal{T}_{\mathbf{Y}} = \{\hat{\mathbf{y}}^l \mid 1 \leq l \leq L\}$

Initialize $\mathcal{T}_{\mathbf{Y}} \longleftarrow \emptyset$

Initialize CPT of $\mathcal{G}_{\mathbf{Y}}$: $P(y_{jt} \mid \boldsymbol{\pi}_{jk}) \longleftarrow 0$, with all $j \in \{1, 2, \ldots, d\}$, $t \in \{1, 2, \ldots, r_j\}$ and $k \in \{1, 2, \ldots, q_j\}$

for $l = 1, 2, \ldots, L$ **do**

  for $j = 1, 2, \ldots, d$ **do**

    for $t = 1, 2, \ldots, r_j$ **do**

      for $k = 1, 2, \ldots, q_j$ **do**

        Update $P(y_{jt} \mid \boldsymbol{\pi}_{jk}) \longleftarrow \mathsf{C}_j^{\boldsymbol{\pi}_{jk}}(\mathbf{x}^l)$

      end for

    end for

  end for

  Perform MPE inference on $\mathcal{G}_{\mathbf{Y}}$ to compute $\hat{\mathbf{y}}^l \longleftarrow \arg\max_{\mathbf{y} \in \mathcal{Y}} P(\mathbf{y} \mid \mathbf{x}^l)$

  Update $\mathcal{T}_{\mathbf{Y}} \longleftarrow \mathcal{T}_{\mathbf{Y}} \cup \{\hat{\mathbf{y}}^l\}$

end for

Experiments

Tabular Data Sets

- 17 data sets containing only continuous feature variables.
- 3 mixed data sets containing both continuous and discrete feature variables.
- The number of class variables range from 2 to 16.
- 5, 6 and 22 discrete feature variables in the 3 mixed data sets, respectively.

Data Set	#CV	#Samples	#States/CV	#Features
Edm	2	154	3	$16n$
Jura	2	359	4,5	$9n$
Enb	2	768	2,4	$6n$
Voice	2	3136	4,2	$19n$
Song	3	785	3	$98n$
Flickr	5	12198	$3, 4, 3, 4, 4$	$1536n$
Fera	5	14052	6	$136n$
WQplants	7	1060	4	$16n$
WQanimals	7	1060	4	$16n$
Rf1	8	8987	$4, 4, 3, 4, 4, 3, 4, 3$	$64n$
Pain	10	9734	$2, 5, 4, 2, 2, 5, 2, 5, 2, 2$	$136n$
Disfa	12	13095	$5, 5, 6, 3, 4, 4, 5, 4, 4, 4, 6, 4$	$136n$
WaterQuality	14	1060	4	$16n$
Oes97	16	334	3	$263n$
Oes10	16	403	3	$298n$
Scm20d	16	8966	4	$61n$
Scm1d	16	9803	4	$280n$
Adult	4	18419	$7, 7, 5, 2$	$5n, 5x$
Default	4	28779	$2, 7, 4, 2$	$14n, 6x$
Thyroid	7	9172	$5, 5, 3, 2, 4, 4, 3$	$7n, 22x$

Table 2: Statistics of the tabular benchmark data sets.

Image Data Set: PASCAL VOC 2007

- 9963 natural images.
  - 50% for training and 50% for test.
- 4 class variables with different numbers of states.
- Remove images that take multiple values for a class variable
  - E.g., an image that contains both cats and dogs.
- Resize to 256×256.

CV	States
Person	no person, person
Animal	no animals, bird, cat, cow, dog, horse, sheep
Vehicle	no vehicles, aeroplane, bicycle, boat, bus, car, motorbike, train
Indoor	no indoor objects, bottle, chair, dining table, potted plant, sofa, tv/monitor

Table 3: Characteristics of the PASCAL VOC 2007 data set in an MDC setting.

Figure 5: Example images from the PASCAL VOC 2007 data set.

Evaluation Metrics

Given a test set $\mathcal{T} = \{(\mathbf{x}^l, \mathbf{y}^l) \mid 1 \leq l \leq L\}$ and an MDC classifier $h : \boldsymbol{\mathcal{X}} \rightarrow \boldsymbol{\mathcal{Y}}$:

Exact Match Score (EMS), which is equivalent to the subset 0/1 loss $\ell_{0/1}$.

$$\begin{aligned} \mathrm{EMS}(h) &= \frac{1}{L} \sum_{l=1}^{L} \mathbb{1}_{\mathbf{y}^l = \hat{\mathbf{y}}^l} \\ &= \frac{1}{L} \sum_{l=1}^{L} (1 - \ell_{0/1}(\mathbf{y}^l, \hat{\mathbf{y}}^l)) \end{aligned} \tag{9}$$

Hamming Score (HS), which is equivalent to the Hamming loss ℓ_{HS}.

$$\begin{aligned} \mathrm{HS}(h) &= \frac{1}{L} \sum_{l=1}^{L} \frac{1}{d} \sum_{j=1}^{d} \mathbb{1}_{y_j^l = \hat{y}_j^l} \\ &= \frac{1}{L} \sum_{l=1}^{L} (1 - \ell_{\mathrm{HL}}(\mathbf{y}^l, \hat{\mathbf{y}}^l)) \end{aligned} \tag{10}$$

Probabilistic MDC Baselines

Binary Relevance (BR)

- Assume all class variables are independent of each other.
- Train a base classifier for each class variable.

Class Powerset (CP)

- Transform each unique combination of class values in the training set into a new meta class.
- Train an overall MCC classifier on the transformed meta classes.

Classifier Chain (CC)

- Represent the class dependencies using a chain-like structure.
- Train a base classifier for each class variable by incorporating previous predictions in the chain.

Experimental Results: Tabular Data #1

Data Set	Exact Match Score (EMS)			
	GBNC ($\ell_{0/1}$)	BR	CC	CP
Edm	$\mathbf{0.570 \pm 0.126}$	0.475 ± 0.129	0.469 ± 0.126	$0.450 \pm 0.141 \uparrow$
Jura	$\mathbf{0.415 \pm 0.054}$	0.409 ± 0.066	0.368 ± 0.114	$0.014 \pm 0.042 \uparrow$
Enb	0.561 ± 0.061	$\mathbf{0.574 \pm 0.090}$	0.525 ± 0.041	$0.371 \pm 0.065 \uparrow$
Voice	$\mathbf{0.858 \pm 0.024}$	$0.836 \pm 0.031 \uparrow$	$0.837 \pm 0.033 \uparrow$	$0.187 \pm 0.024 \uparrow$
Song	$\mathbf{0.438 \pm 0.049}$	0.422 ± 0.041	$0.392 \pm 0.061 \uparrow$	$0.068 \pm 0.035 \uparrow$
Flickr	0.289 ± 0.013	$0.324 \pm 0.012 \downarrow$	$\mathbf{0.325 \pm 0.013} \downarrow$	$0.042 \pm 0.006 \uparrow$
Fera	$\mathbf{0.196 \pm 0.012}$	0.187 ± 0.014	0.193 ± 0.011	$0.164 \pm 0.015 \uparrow$
WQplants	0.093 ± 0.034	0.093 ± 0.026	$\mathbf{0.095 \pm 0.029}$	$0.075 \pm 0.033 \uparrow$
WQanimals	0.046 ± 0.008	0.047 ± 0.020	$\mathbf{0.057 \pm 0.015} \downarrow$	$0.023 \pm 0.016 \uparrow$
Rf1	$\mathbf{0.532 \pm 0.018}$	$0.283 \pm 0.020 \uparrow$	$0.291 \pm 0.018 \uparrow$	$0.062 \pm 0.009 \uparrow$
Pain	$\mathbf{0.758 \pm 0.018}$	$0.751 \pm 0.015 \uparrow$	$0.754 \pm 0.017 \uparrow$	$0.751 \pm 0.015 \uparrow$
Disfa	$\mathbf{0.401 \pm 0.011}$	$0.393 \pm 0.012 \uparrow$	$0.394 \pm 0.013 \uparrow$	$0.371 \pm 0.011 \uparrow$
WaterQuality	0.005 ± 0.005	$\mathbf{0.007 \pm 0.006}$	$\mathbf{0.007 \pm 0.006}$	0.006 ± 0.006
Oes97	$\mathbf{0.057 \pm 0.044}$	0.051 ± 0.046	$0.030 \pm 0.014 \uparrow$	$0.000 \uparrow$
Oes10	0.087 ± 0.041	0.089 ± 0.032	$\mathbf{0.092 \pm 0.034}$	$0.005 \pm 0.010 \uparrow$
Scm20d	$\mathbf{0.124 \pm 0.014}$	$0.045 \pm 0.008 \uparrow$	$0.062 \pm 0.011 \uparrow$	$0.076 \pm 0.012 \uparrow$
Scm1d	$\mathbf{0.189 \pm 0.010}$	$0.098 \pm 0.009 \uparrow$	$0.109 \pm 0.008 \uparrow$	$0.091 \pm 0.009 \uparrow$
No. of Wins	11	2	5	0

Table 4: Exact match scores (mean $\pm$ std.) of each MDC approach (**base classifier: *logisitic regression***). GBNC performs inference by **optimizing** $\ell_{0/1}$. The best performance is highlighted in bold, and $\uparrow$ / $\downarrow$ indicates whether GBNC is significantly superior/inferior to other approaches on each data set by using a *Wilcoxon signed-rank test*.

Experimental Results: Tabular Data #2

Data Set	Hamming Score (HS)			
	GBNC (ℓ_{HL})	BR	CC	CP
Edm	$\mathbf{0.736 \pm 0.091}$	0.696 ± 0.093	0.685 ± 0.099	0.725 ± 0.071
Jura	$\mathbf{0.634 \pm 0.029}$	0.625 ± 0.043	0.607 ± 0.075	$0.317 \pm 0.051 \uparrow$
Enb	0.781 ± 0.031	$\mathbf{0.787 \pm 0.045}$	0.762 ± 0.020	$0.685 \pm 0.033 \uparrow$
Voice	$\mathbf{0.927 \pm 0.013}$	$0.915 \pm 0.015 \uparrow$	$0.916 \pm 0.017 \uparrow$	$0.584 \pm 0.013 \uparrow$
Song	$\mathbf{0.766 \pm 0.028}$	0.752 ± 0.023	$0.738 \pm 0.039 \uparrow$	$0.507 \pm 0.040 \uparrow$
Flickr	0.784 ± 0.006	$\mathbf{0.797 \pm 0.006} \downarrow$	$0.796 \pm 0.006 \downarrow$	$0.506 \pm 0.004 \uparrow$
Fera	$\mathbf{0.624 \pm 0.010}$	$0.616 \pm 0.007 \uparrow$	$0.605 \pm 0.009 \uparrow$	$0.475 \pm 0.018 \uparrow$
WQplants	$\mathbf{0.658 \pm 0.013}$	0.655 ± 0.010	0.650 ± 0.014	$0.611 \pm 0.024 \uparrow$
WQanimals	$\mathbf{0.630 \pm 0.018}$	0.626 ± 0.019	$0.624 \pm 0.021 \uparrow$	$0.579 \pm 0.024 \uparrow$
Rf1	$\mathbf{0.902 \pm 0.008}$	$0.836 \pm 0.004 \uparrow$	$0.835 \pm 0.006 \uparrow$	$0.635 \pm 0.007 \uparrow$
Pain	$\mathbf{0.953 \pm 0.003}$	$\mathbf{0.953 \pm 0.003}$	$0.951 \pm 0.003 \uparrow$	$0.948 \pm 0.003 \uparrow$
Disfa	$\mathbf{0.897 \pm 0.002}$	$0.894 \pm 0.003 \uparrow$	$0.894 \pm 0.002 \uparrow$	$0.871 \pm 0.003 \uparrow$
WaterQuality	$\mathbf{0.642 \pm 0.018}$	0.637 ± 0.011	0.639 ± 0.017	$0.597 \pm 0.018 \uparrow$
Oes97	$\mathbf{0.731 \pm 0.023}$	$0.716 \pm 0.018 \uparrow$	$0.706 \pm 0.030 \uparrow$	$0.521 \pm 0.032 \uparrow$
Oes10	$\mathbf{0.809 \pm 0.014}$	0.801 ± 0.019	$0.791 \pm 0.021 \uparrow$	$0.605 \pm 0.046 \uparrow$
Scm20d	$\mathbf{0.685 \pm 0.007}$	$0.640 \pm 0.008 \uparrow$	$0.613 \pm 0.011 \uparrow$	$0.424 \pm 0.012 \uparrow$
Scm1d	$\mathbf{0.815 \pm 0.003}$	$0.763 \pm 0.006 \uparrow$	$0.748 \pm 0.005 \uparrow$	$0.444 \pm 0.011 \uparrow$
No. of Wins	**15**	**3**	**0**	**0**

Table 5: Hamming scores (mean $\pm$ std.) of each MDC approach (**base classifier: *logisitic regression***). GBNC performs inference by **optimizing** ℓ_{HL}. The best performance is highlighted in bold, and $\uparrow$ /$\downarrow$ indicates whether GBNC is significantly superior/inferior to other approaches on each data set by using a *Wilcoxon signed-rank test*.

Experimental Results: Mixed Data

Data Set	Exact Match Score (EMS)			
	GBNC ($\ell_{0/1}$)	BR	CC	CP
Adult	0.245 ± 0.006	$\mathbf{0.274 \pm 0.011} \downarrow$	$\mathbf{0.274 \pm 0.013} \downarrow$	$0.134 \pm 0.007 \uparrow$
Default	$\mathbf{0.187 \pm 0.005}$	$0.177 \pm 0.009 \uparrow$	$0.177 \pm 0.012 \uparrow$	$0.060 \pm 0.004 \uparrow$
Thyroid	$\mathbf{0.784 \pm 0.018}$	$0.774 \pm 0.014 \uparrow$	$0.768 \pm 0.016 \uparrow$	$0.751 \pm 0.016 \uparrow$

Table 6: Exact match scores (mean $\pm$ std.) of each MDC approach (**base classifier: *logisitic regression***) on **mixed data**. GBNC performs inference by **optimizing** $\ell_{0/1}$.

Data Set	Hamming Score (HS)			
	GBNC (ℓ_{HL})	BR	CC	CP
Adult	0.676 ± 0.004	$\mathbf{0.718 \pm 0.005} \downarrow$	$0.715 \pm 0.005 \downarrow$	$0.585 \pm 0.006 \uparrow$
Default	$\mathbf{0.666 \pm 0.004}$	0.664 ± 0.007	0.663 ± 0.009	$0.561 \pm 0.004 \uparrow$
Thyroid	$\mathbf{0.966 \pm 0.003}$	$0.965 \pm 0.002 \uparrow$	$0.964 \pm 0.003 \uparrow$	$0.962 \pm 0.003 \uparrow$

Table 7: Hamming scores (mean $\pm$ std.) of each MDC approach (**base classifier: *logistic regression***) on **mixed data**. GBNC performs inference by **optimizing** ℓ_{HL}.

- 5 continuous and 5 discrete feature variables in the `Adult` data set.

Experimental Results: Image Data

Metrics	Methods			
	GBNC ($\ell_{0/1}$)	GBNC (ℓ_{HL})	BR	CP
HS	**0.897**	**0.897**	0.891	0.887
EMS	**0.662**	**0.662**	0.604	0.655

Table 8: Results of each MDC approach (**base classifier: *ResNet-18***) for the PASCAL VOC 2007 data set.

- Sparse BN structure since there are only **4** class variables.
- Extreme probability estimates due to uncalibrated neural networks.

Analysis of Learned BN Structures

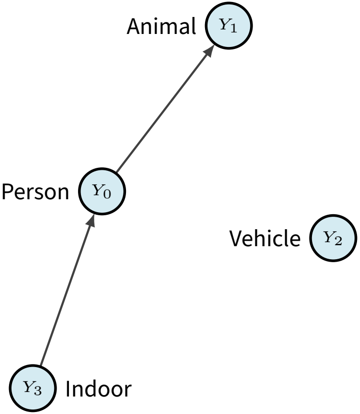


Figure 6: Learned BN structure of GBNC on the PASCAL VOC 2007 data set. The base classifier is *ResNet-18* with weights pre-trained on ImageNet.

Figure 7: Example images from the PASCAL VOC 2007 data set.

- The `Vehicle` dimension appear to be independent.
- GBNC tends to consider the mutual relationships between `Indoor`, `Person` and `Animal`.

Conclusion

Contributions

Proposed GBNC, a generalized framework for solving probabilistic MDC tasks with complex types of input.

- The first approach that exactly optimizes the CLL function in learning MBNCs.
- Introduced a new structural constraint for learning (multi-dimensional) BNCs, which enables the decomposability of the CLL function over a BN structure.
- Proposed a input space partitioning algorithm to discriminatively learn BNC structures and parameters simultaneously, in which the CLL function is optimized.
- GBNC converts the prediction problem into an inference problem in the learned class BN structure, which allows computing the Bayes-Optimal Prediction under different loss functions.

Contributions

Proposed GBNC, a generalized framework for solving probabilistic MDC tasks with complex types of input.

- The first approach that exactly optimizes the CLL function in learning MBNCs.
- Introduced a new structural constraint for learning (multi-dimensional) BNCs, which enables the decomposability of the CLL function over a BN structure.
- Proposed a input space partitioning algorithm to discriminatively learn BNC structures and parameters simultaneously, in which the CLL function is optimized.
- GBNC converts the prediction problem into an inference problem in the learned class BN structure, which allows computing the Bayes-Optimal Prediction under different loss functions.
- By employing the pruning rules from [2] and using GOBNILP [1], GBNC is able to exactly and efficiently learn class BN structures with the penalized CLL as a structure score function.
- By using the same partitioning idea, GBNC is able to handle mixed data.
- GBNC achieves leading performance among other discriminative MDC approaches.

Future Work

- Structure complexity from feature variables.
- Compute Bayes-optimal prediction under more loss functions, such as the Brier score and the AUC score.
- Further utilize the discrete features other than input space partitioning.
- More efficient inference algorithms.

Thank You!

Questions?

Parameter Learning Algorithm

Algorithm 2 Extract training data

Require: Training data $\mathcal{D} = \{(\mathbf{x}^l, \mathbf{y}^l) \mid 1 \leq l \leq N\}$, class variable Y_j $(1 \leq j \leq d)$, parent set $\mathbf{\Pi}_j$, parent configuration $\boldsymbol{\pi}_{jk}$, parent set $\mathbf{\Phi}_j$

Ensure: $\mathcal{D}_j^{\boldsymbol{\pi}_{jk}}$

```
Initialize 𝒟_j^{π_jk} ⟵ ∅
for l = 1, 2, ..., N do
    if π_j^l = π_jk then
        Update 𝒟_j^{π_jk} ⟵ 𝒟_j^{π_jk} ∪ {(φ_j^l, y_j)}
    end if
end for
```

Initialize $\mathcal{D}_j^{\boldsymbol{\pi}_{jk}} \longleftarrow \emptyset$

for $l = 1, 2, \ldots, N$ **do**

  if $\boldsymbol{\pi}_j^l = \boldsymbol{\pi}_{jk}$ **then**

    Update $\mathcal{D}_j^{\boldsymbol{\pi}_{jk}} \longleftarrow \mathcal{D}_j^{\boldsymbol{\pi}_{jk}} \cup \{(\phi_j^l, y_j)\}$

  end if

end for

Parameter Learning Algorithm

Algorithm 3 Train local probabilistic model

Require: Training data $\mathcal{D} = \{(\mathbf{x}^l, \mathbf{y}^l) \mid 1 \leq l \leq N\}$, class variable Y_j $(1 \leq j \leq d)$, parent set $\mathbf{\Pi}$, parent configuration $\boldsymbol{\pi}$, parent set $\mathbf{\Phi}$ ($\mathbf{\Phi} = \mathbf{X}$ by default if not specified)

Ensure: Local classifier $\mathrm{C}_j^{\boldsymbol{\pi}}$

1: Extract $\mathcal{D}_j^{\boldsymbol{\pi}}$ using 2 with input $\mathcal{D}$, Y_j, $\mathbf{\Pi}$, $\boldsymbol{\pi}$ and $\mathbf{\Phi}$

2: Train $\mathrm{C}_j^{\boldsymbol{\pi}}$ using $\mathcal{D}_j^{\boldsymbol{\pi}}$ by maximizing the CLL

Parameter Learning Algorithm

TU/e

Algorithm 4 Parameter learning in $\mathcal{G}$

Require: Training data $\mathcal{D} = \{(\mathbf{x}^l, \mathbf{y}^l) \mid 1 \leq l \leq N\}$, BN structure $\mathcal{G}$

Ensure: Local classifiers **C**

```
Initialize C ⟵ ∅
for j = 1, 2, ..., d do
    Extract Π_j and Φ_j from G
    for k = 1, 2, ..., q_j do
        Train local classifier C_j^{π_jk} using 3 with input D, Y_j, Π_j, π_jk and Φ_j
        Update C ⟵ C ∪ C_j^{π_jk}
    end for
end for
```

Initialize $\mathbf{C} \longleftarrow \emptyset$

for $j = 1, 2, \ldots, d$ **do**

  Extract $\mathbf{\Pi}_j$ and $\mathbf{\Phi}_j$ from $\mathcal{G}$

  for $k = 1, 2, \ldots, q_j$ **do**

    Train local classifier $\mathrm{C}_j^{\boldsymbol{\pi}_{jk}}$ using 3 with input $\mathcal{D}$, Y_j, $\mathbf{\Pi}_j$, $\boldsymbol{\pi}_{jk}$ and $\mathbf{\Phi}_j$

    Update $\mathbf{C} \longleftarrow \mathbf{C} \cup \mathrm{C}_j^{\boldsymbol{\pi}_{jk}}$

  end for

end for

Structure Learning Algorithm

Algorithm 5 Structure learning of $\mathcal{G}_{\mathbf{Y}}$

Require: Training data $\mathcal{D} = \{(\mathbf{x}^l, \mathbf{y}^l) \mid 1 \leq l \leq N\}$
Ensure: Class subgraph $\mathcal{G}_{\mathbf{Y}}$

Initialize a score dict **S** ▷ where $\mathbf{S}[j][\mathbf{\Pi}]$ stores the local score for Y_j given the parent set $\mathbf{\Pi}$
for $j = 1, 2, \ldots, d$ **do**
  Initialize candidate parent sets $\mathbf{K}_j$ of Y_j as the powerset of $\mathbf{Y} \setminus Y_j$
  for $u = 1, 2, \ldots, |\mathbf{K}_j|$ **do**
    Compute $\mathrm{PEN}(\mathcal{G}_{\mathbf{K}_{ju}}) \longleftarrow -\frac{\log N}{2}(r_j - 1)q_j$ ▷ here $q_j = \prod_{c=1:Y_c \in \mathbf{K}_{ju}}^{d} r_c$
    if $\mathbf{K}_{ju}$ can be pruned **then** ▷ using Lemma 2 in the thesis
      Remove all supersets of $\mathbf{K}_{ju}$ from $\mathbf{K}$
      continue
    end if
    Initialize $\mathsf{S} \longleftarrow 0$
    for $k = 1, 2, \ldots, q_j$ **do**
      Extract $\mathcal{D}_j^{\mathbf{k}_{juk}}$ using 2 with input $\mathcal{D}, Y_j, \mathbf{\Pi}_j = \mathbf{K}_{ju}, \boldsymbol{\pi}_j = \mathbf{k}_{juk}$ and $\mathbf{\Phi} = \mathbf{X}$
      Train local classifier $\mathsf{C}_j^{\mathbf{k}_{juk}}$ using 3 with input $\mathcal{D}, Y_j, \mathbf{\Pi}_j = \mathbf{K}_{ju}, \boldsymbol{\pi}_j = \mathbf{k}_{juk}$ and $\mathbf{\Phi}_j = \mathbf{X}$
      Update $\mathsf{S} \longleftarrow \mathsf{S} + \mathsf{C}_j^{\mathbf{k}_{juk}}(\mathcal{D}_j^{\mathbf{k}_{juk}})$
    end for
    if $\mathbf{K}_{ju}$ can be pruned **then** ▷ using Lemma 1 in the thesis
      continue
    end if
    $\mathbf{S}[j][\mathbf{K}_{ju}] \longleftarrow \mathsf{S}$
  end for
end for
Learn $\mathcal{G}_{\mathbf{Y}}$ using GOBNILP with input **S**

References I

J. Cussens.
Gobnilp: Learning bayesian network structure with integer programming.
In *International Conference on Probabilistic Graphical Models*, pages 605–608. PMLR, 2020.

C. P. de Campos, M. Scanagatta, G. Corani, and M. Zaffalon.
Entropy-based pruning for learning bayesian networks using BIC.
Artif. Intell., 260:42–50, 2018.

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