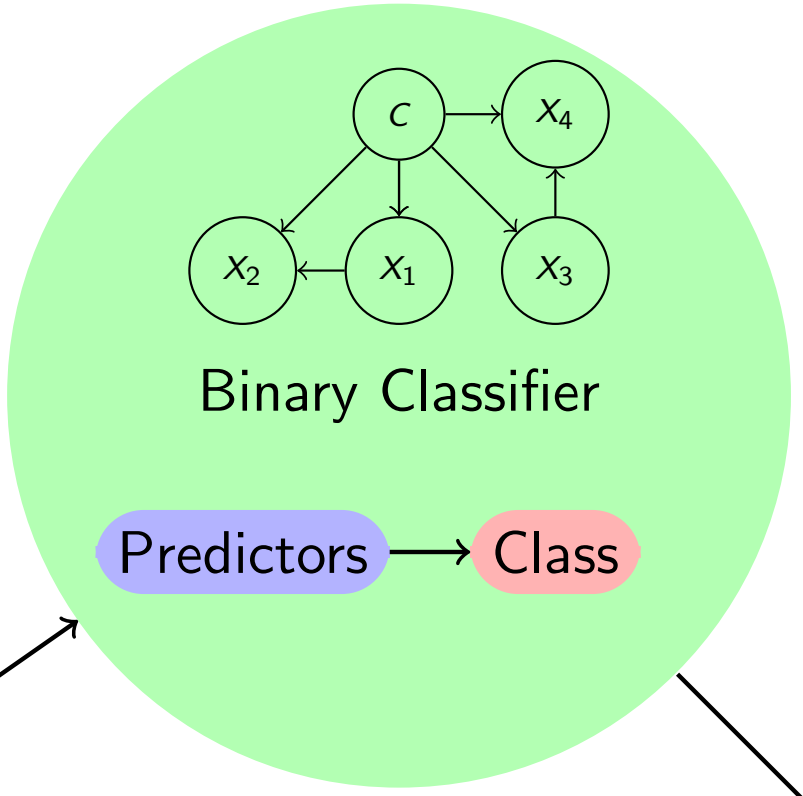
Polyonomial representation of Bayesian network classifiers

Gherardo Varando

Computational Intelligence Group
Universidad Politécnica de Madrid

23 January 2017

Classification problem



Training Data

X_1	X_2	X_3	X_4	C
1	1	4	0	+1
−1	1	5	0	−1
1	2	2	1	−1

New predictions

CIG

Some notations

- predictors, $\mathbf{X} = (X_1, \ldots, X_n)$, $X_i \in \Omega_i$, $|\Omega_i| = m_i$
  $\mathbf{\Omega} = \Omega_1 \times \Omega_2 \times \cdots \times \Omega_n$

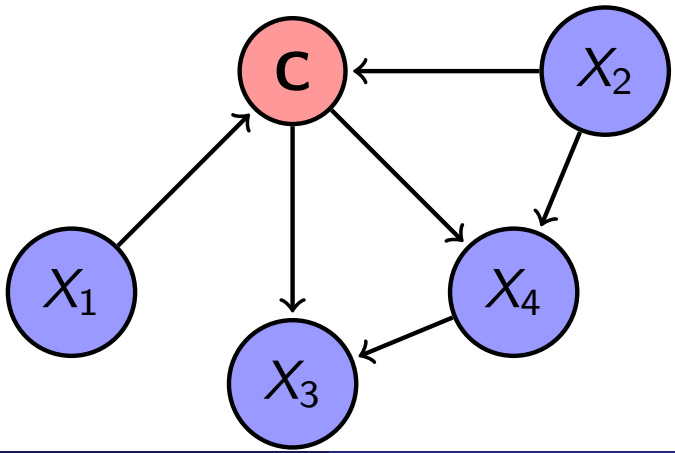
Some notations

- predictors, $\mathbf{X} = (X_1, \ldots, X_n)$, $X_i \in \Omega_i$, $|\Omega_i| = m_i$
  $\mathbf{\Omega} = \Omega_1 \times \Omega_2 \times \cdots \times \Omega_n$
- class, $C \in \{-1, +1\}$

Some notations

- predictors, $\mathbf{X} = (X_1, \ldots, X_n)$, $X_i \in \Omega_i$, $|\Omega_i| = m_i$
  $\mathbf{\Omega} = \Omega_1 \times \Omega_2 \times \cdots \times \Omega_n$
- class, $C \in \{-1, +1\}$

$$P(C, X_1, X_2, X_3, X_4) = P(X_1)P(X_2)P(C|X_1, X_2)P(X_4|C, X_2)P(X_3|X_4, C)$$

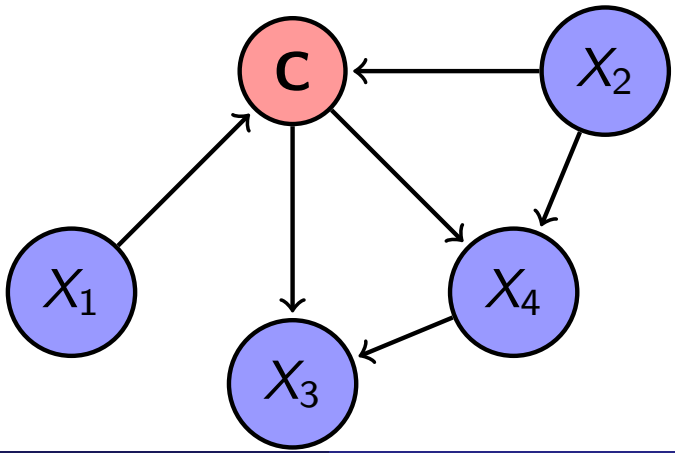


- conditional probabilities are estimated by data or given by experts
- we assume conditional probability tables different from zero

Some notations

- predictors, $\mathbf{X} = (X_1, \ldots, X_n)$, $X_i \in \Omega_i$, $|\Omega_i| = m_i$
  $\mathbf{\Omega} = \Omega_1 \times \Omega_2 \times \cdots \times \Omega_n$
- class, $C \in \{-1, +1\}$
- $P(C, X_1, \ldots) = P(C|\mathbf{pa}(C)) \prod_i P(X_i|\mathbf{pa}(X_i))$

$P(C, X_1, X_2, X_3, X_4) = P(X_1)P(X_2)P(C|X_1, X_2)P(X_4|C, X_2)P(X_3|X_4, C)$



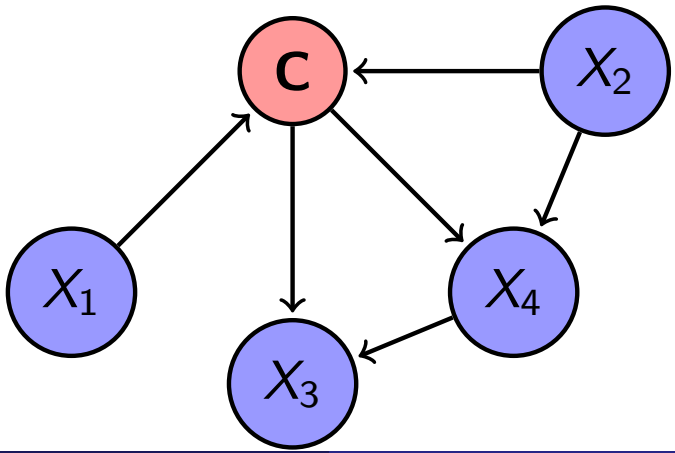
- conditional probabilities are estimated by data or given by experts
- we assume conditional probability tables different from zero

Some notations

- predictors, $\mathbf{X} = (X_1, \ldots, X_n)$, $X_i \in \Omega_i$, $|\Omega_i| = m_i$
  $\mathbf{\Omega} = \Omega_1 \times \Omega_2 \times \cdots \times \Omega_n$
- class, $C \in \{-1, +1\}$
- $P(C, X_1, \ldots) = P(C|\mathbf{pa}(C)) \prod_i P(X_i|\mathbf{pa}(X_i))$

$$\hat{c}(x_1, \ldots, x_n) = arg \max_{c \in \{-1,+1\}} P(C = c | X_1 = x_1, \ldots, X_n = x_n)$$
$$= arg \max_{c \in \{-1,+1\}} P(C = c, X_1 = x_1, \ldots, X_n = x_n)$$

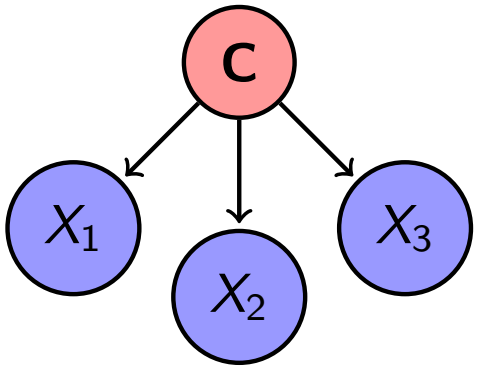
$P(C, X_1, X_2, X_3, X_4) = P(X_1)P(X_2)P(C|X_1, X_2)P(X_4|C, X_2)P(X_3|X_4, C)$



- conditional probabilities are estimated by data or given by experts
- we assume conditional probability tables different from zero

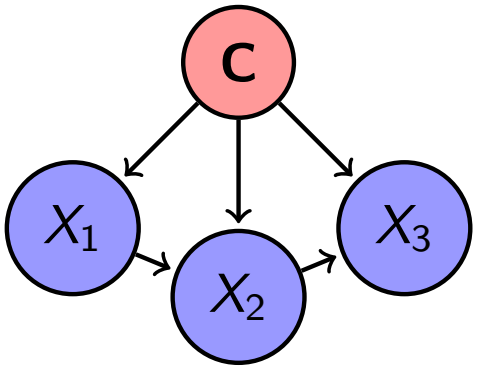
Some taxonomy

Naive Bayes

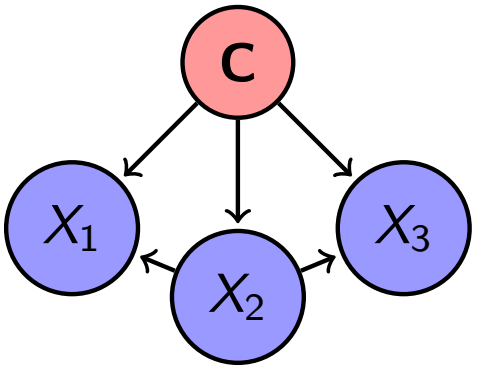


$P(C)\prod_i P(X_i|C)$

Tree Augmented NB



SPODE

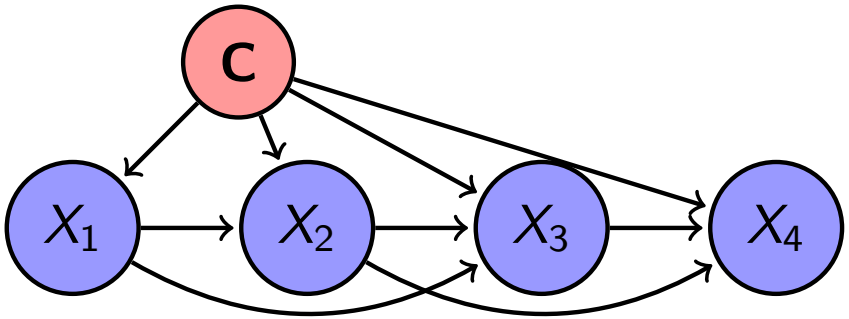


$P(C)\prod_i P(X_i|C, pa(X_i))$

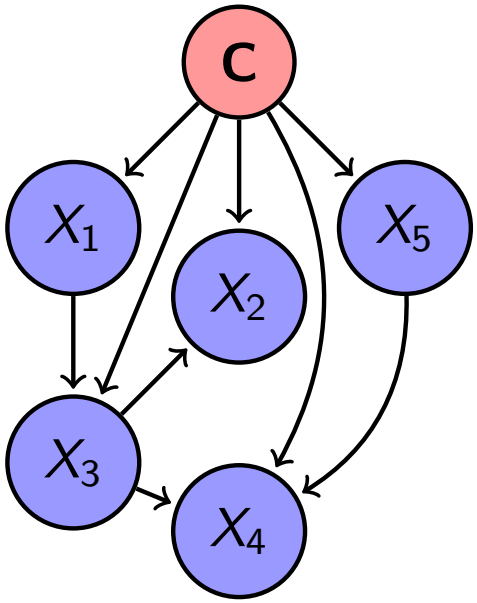
C. Bielza, P. Larrañaga (2014). **Discrete Bayesian network classifiers: A survey**. *ACM Computing Surveys*, 47 (1).

Some taxonomy

k-dependence



Bayesian Augmented NB

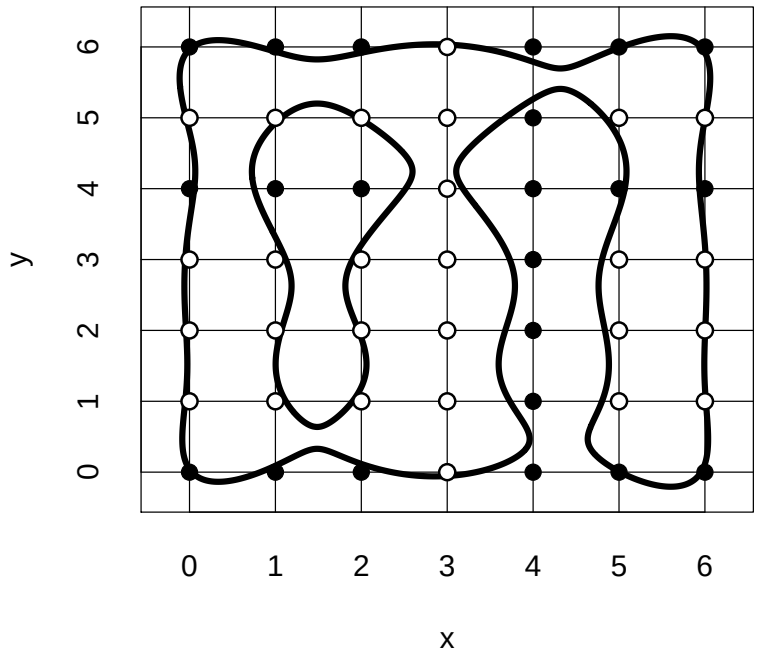


C. Bielza, P. Larrañaga (2014). **Discrete Bayesian network classifiers: A survey**. *ACM Computing Surveys*, 47 (1).

Expressive power

- a classifier is "equivalent" to a decision function

$$f : \mathbf{\Omega} \mapsto \{-1, +1\}$$



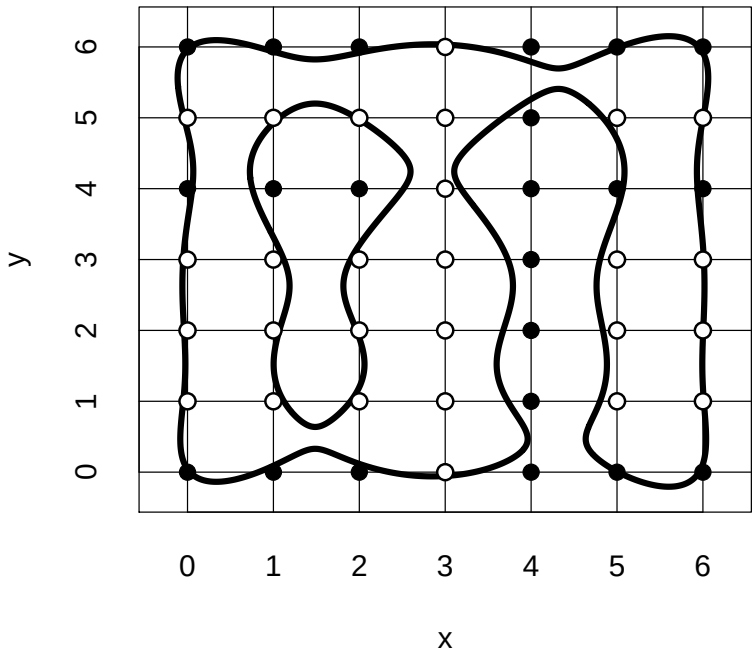
Expressive power

- a classifier is "equivalent" to a decision function

$$f : \mathbf{\Omega} \mapsto \{-1, +1\}$$

- how many decision functions do exist over $\mathbf{\Omega}$?

$$\left|\{-1, +1\}^{\mathbf{\Omega}}\right| = 2^{|\mathbf{\Omega}|} = 2^{\prod_i m_i}$$



Expressive power

- a classifier is "equivalent" to a decision function

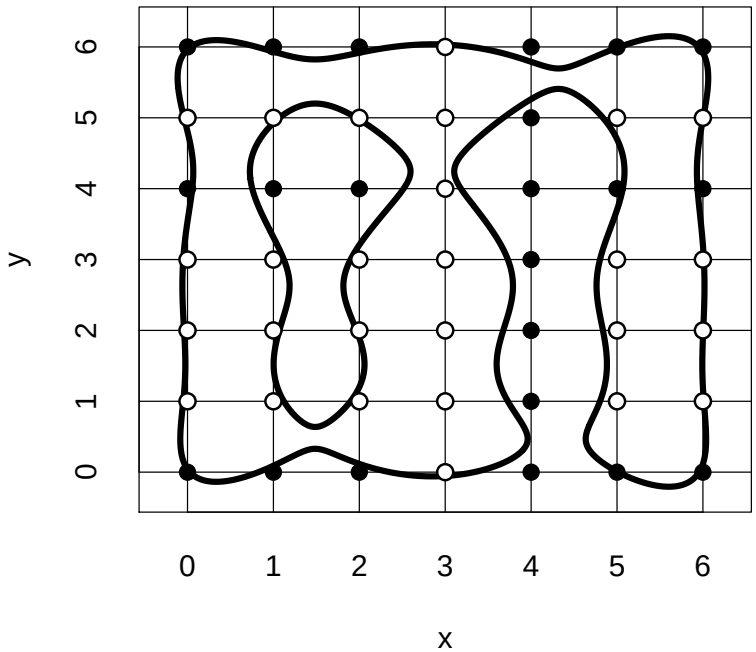
$$f : \mathbf{\Omega} \mapsto \{-1, +1\}$$

- how many decision functions do exist over $\mathbf{\Omega}$?

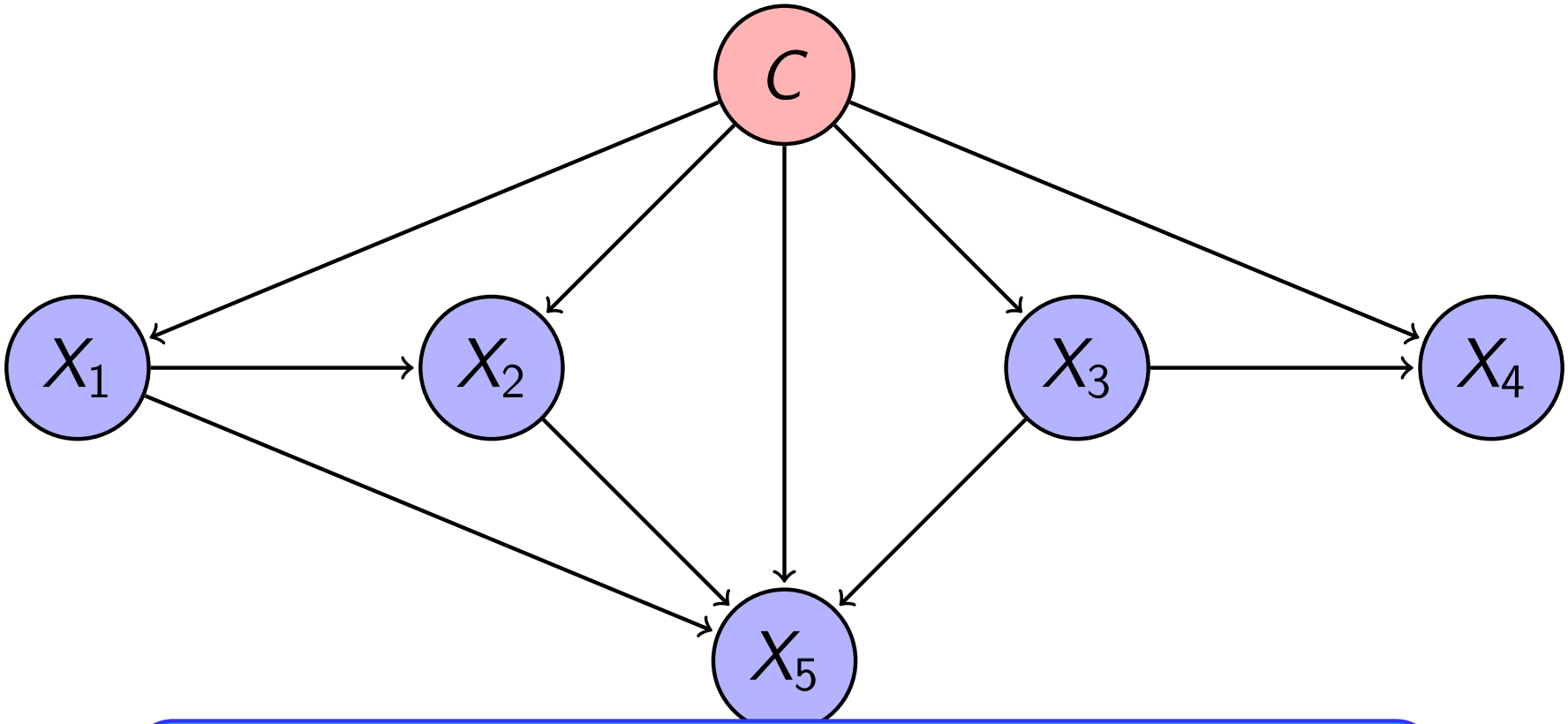
$$\left|\{-1, +1\}^{\mathbf{\Omega}}\right| = 2^{|\mathbf{\Omega}|} = 2^{\prod_i m_i}$$

- we can represent decision functions as sign of polynomial over $\mathbf{\Omega}$

$$f(\mathbf{x}) = sgn(r(\mathbf{x}))$$



BAN classifier



$$f(x_1, \ldots, x_n) = arg\,\max_c P(C = c | X_1 = x_1, \ldots, X_n = x_n)$$

Polynomial representations

- $\mathcal{G}$ a directed acyclic graph with nodes X_i without V-structures
- f, a decision function over predictor variables $X_i \in \Omega_i$

Then

$$f(\mathbf{x}) = sgn\big(r(\mathbf{x})\big)$$

$$r(\mathbf{x}) = \sum_{i=1}^{n}\sum_{j=1}^{m_i} \ell_j^{\Omega_i}(x_i) \sum_{\mathbf{k}\in\mathbb{M}_i} \beta_i(j|\mathbf{k}) \prod_{s\in\mathbf{pa}(i)} \ell_{k_s}^{\Omega_s}(x_s)$$

$$\Updownarrow$$

f is induced by a BAN classifier whose predictor sub-graph is $\mathcal{G}$

Varando, Bielza, Larrañaga (2015). **Decision boundary for discrete Bayesian network classifiers**. *Journal of Machine Learning Research, vol. 16, pp. 2725-2749, 2015.*

proof $\Rightarrow$

Lagrange interpolation of discrete probability

- $X_i \in \Omega_i = \{\xi_i^1, \xi_i^2, \ldots, \xi_i^{m_i}\} \subset \mathbb{R}$

proof $\Rightarrow$

Lagrange interpolation of discrete probability

- $X_i \in \Omega_i = \{\xi_i^1, \xi_i^2, \ldots, \xi_i^{m_i}\} \subset \mathbb{R}$

- Lagrange polynomials $\ell_j^{\Omega_i}(x) = \prod_{k \neq j} \frac{(x - \xi_i^k)}{(\xi_i^j - \xi_i^k)}$
  - $\prod_{i \in I} \ell_{j_i}^{\Omega_i}(x_i) = [x_i = \xi_i^{j_i} \ \forall i \in I]$
  - $\sum_{j_{i_1}=1}^{m_{i_1}} \sum_{j_{i_2}=1}^{m_{i_2}} \cdots \sum_{j_{i_p}=1}^{m_{i_p}} \prod_{s \in I} \ell_{j_s}^{\Omega_s}(x_s) = \prod_{i \in I \setminus J} \ell_{j_i}^{\Omega_i}(x_i)$

proof $\Rightarrow$

Lagrange interpolation of discrete probability

- $X_i \in \Omega_i = \{\xi_i^1, \xi_i^2, \ldots, \xi_i^{m_i}\} \subset \mathbb{R}$
- Lagrange polynomials $\ell_j^{\Omega_i}(x) = \prod_{k \neq j} \frac{(x - \xi_i^k)}{(\xi_i^j - \xi_i^k)}$
  - $\prod_{i \in I} \ell_{j_i}^{\Omega_i}(x_i) = [x_i = \xi_i^{j_i} \ \forall i \in I]$
  - $\sum_{j_{i_1}=1}^{m_{i_1}} \sum_{j_{i_2}=1}^{m_{i_2}} \cdots \sum_{j_{i_p}=1}^{m_{i_p}} \prod_{s \in I} \ell_{j_s}^{\Omega_s}(x_s) = \prod_{i \in I \setminus J} \ell_{j_i}^{\Omega_i}(x_i)$
- $P(X_1 = x_1 | X_2 = x_2, \ldots, X_n = x_n) =$
  $\prod_{(j_1,\ldots,j_n)} p(j_1 | j_2, \ldots, j_n)^{[x_i = \xi_i^{j_i} \ \forall i = 1,\ldots,n]} =$
  $\prod_{(j_1,\ldots,j_n)} p(j_1 | j_2, \ldots, j_n)^{\prod_{i=1}^{m} \ell_{j_i}^{\Omega_i}(x_i)}$

proof $\Rightarrow$

Given a BAN classifier we define now

$$\beta_i(j|\mathbf{k}) = \ln\left(\frac{P\left(X_i = \xi_i^j | C = +1, X_s = \xi_s^{k_s}, \forall s \in \mathbf{pa}(i)\right)}{P\left(X_i = \xi_i^j | C = -1, X_s = \xi_s^{k_s}, \forall s \in \mathbf{pa}(i)\right)}\right)$$

proof $\Rightarrow$

Given a BAN classifier we define now

$$\beta_i(j|\mathbf{k}) = \ln\left(\frac{P\left(X_i = \xi_i^j | C = +1, X_s = \xi_s^{k_s}, \forall s \in \mathbf{pa}(i)\right)}{P\left(X_i = \xi_i^j | C = -1, X_s = \xi_s^{k_s}, \forall s \in \mathbf{pa}(i)\right)}\right)$$

then from

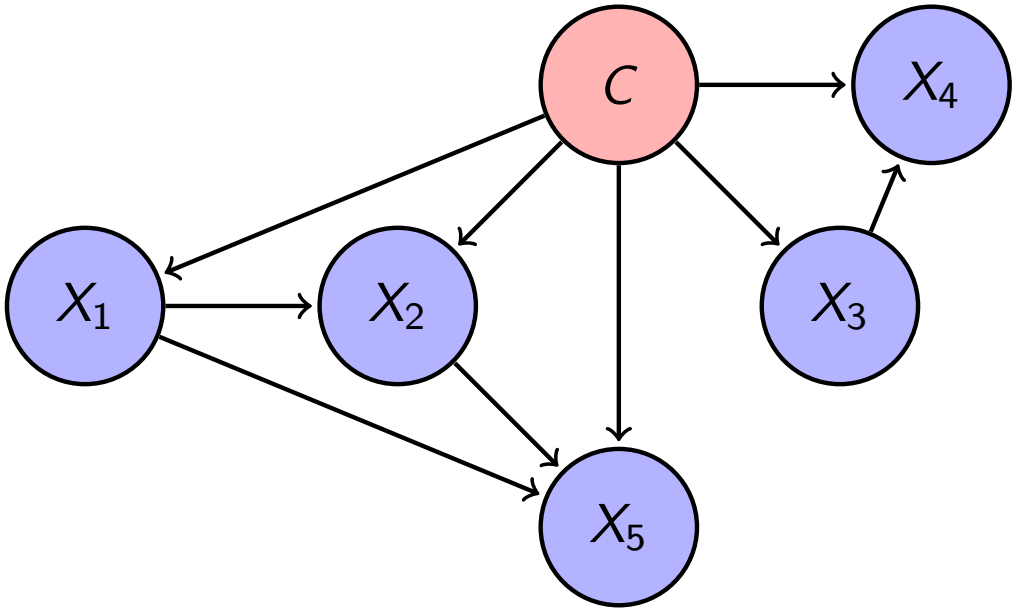
$$P(C = +1, X_1 = x_1, \ldots, X_n = x_n) > P(C = -1, X_1 = x_1, \ldots, X_n = x_n)$$

we derive

$$\sum_{i=1}^{n}\sum_{j=1}^{m_i} \ell_j^{\Omega_i}(x_i) \sum_{\mathbf{k}\in\mathbb{M}_i} \beta_i(j|\mathbf{k}) \prod_{s\in\mathbf{pa}(i)} \ell_{k_s}^{\Omega_s}(x_s) > 0$$

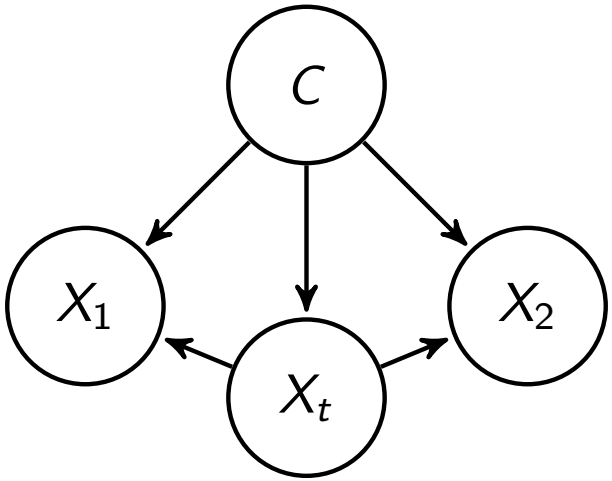
proof $\Leftarrow$

For the other implication we assume there are no V-structures in the predictor sub-graph.



for every polynomial $\sum_{i=1}^{n} \sum_{j=1}^{m_i} \ell_j^{\Omega_i}(x_i) \sum_{\mathbf{k} \in \mathbb{M}_i} \beta_i(j|\mathbf{k}) \prod_{s \in \mathbf{pa}(i)} \ell_{k_s}^{\Omega_s}(x_s)$ we prove there exist probability tables compatible to the given structure that induce the polynomial

SPODE example



$$r(\mathbf{x}) = \sum_{i=1}^{n} \sum_{j=1}^{m_i} \ell_j^{\Omega_i}(x_i) \sum_{\mathbf{k} \in \mathbb{M}_i} \beta_i(j|\mathbf{k}) \prod_{s \in \mathbf{pa}(i)} \ell_{k_s}^{\Omega_s}(x_s)$$

$$r(x_t, x_1, x_2) = (1 - x_t) \ln\left(\frac{0.4}{0.8}\right) + x_t \ln\left(\frac{0.6}{0.2}\right)$$

$$+ \left(\frac{(1-x_1)(2-x_1)}{2} \ln\left(\frac{0.2}{0.1}\right) + x_1(2 - x_1) \ln\left(\frac{0.7}{0.1}\right) + \frac{x_1(x_1-1)}{2} \ln\left(\frac{0.1}{0.8}\right) \right) (1 - x_t)$$

$$+ \left(\frac{(1-x_1)(2-x_1)}{2} \ln\left(\frac{0.7}{0.3}\right) + x_1(2 - x_1) \ln\left(\frac{0.1}{0.2}\right) + \frac{x_1(x_1-1)}{2} \ln\left(\frac{0.2}{0.5}\right) \right) x_t$$

CIG

Geometric considerations

Consider the space $\mathcal{P}^{FBN}$ as the space of interpolating polynomials over $\mathbf{\Omega}$

- $\mathcal{P}^{FBN}$ is generated by $\delta_{\mathbf{k}}(\mathbf{x}) = \prod_{i=1}^{n} \ell_{k_i}^{\Omega_i}(x_i)$ (canonical basis)

Geometric considerations

Consider the space $\mathcal{P}^{FBN}$ as the space of interpolating polynomials over $\mathbf{\Omega}$

- $\mathcal{P}^{FBN}$ is generated by $\delta_{\mathbf{k}}(\mathbf{x}) = \prod_{i=1}^{n} \ell_{k_i}^{\Omega_i}(x_i)$ (canonical basis)
- $\mathcal{P}^{\mathcal{G}}$ is a vectorial subspace of $\mathcal{P}^{FBN}$

Geometric considerations

Consider the space $\mathcal{P}^{FBN}$ as the space of interpolating polynomials over $\mathbf{\Omega}$

- $\mathcal{P}^{FBN}$ is generated by $\delta_{\mathbf{k}}(\mathbf{x}) = \prod_{i=1}^{n} \ell_{k_i}^{\Omega_i}(x_i)$ (canonical basis)
- $\mathcal{P}^{\mathcal{G}}$ is a vectorial subspace of $\mathcal{P}^{FBN}$
- $dim\left(\mathcal{P}^{\mathcal{G}}\right) = \sum_{i=1}^{n} \left((m_i - 1) \prod_{s \in \mathbf{pa}(i)} m_s \right) + 1$

Geometric considerations

Consider the space $\mathcal{P}^{FBN}$ as the space of interpolating polynomials over $\mathbf{\Omega}$

- $\mathcal{P}^{FBN}$ is generated by $\delta_{\mathbf{k}}(\mathbf{x}) = \prod_{i=1}^{n} \ell_{k_i}^{\Omega_i}(x_i)$ (canonical basis)
- $\mathcal{P}^{\mathcal{G}}$ is a vectorial subspace of $\mathcal{P}^{FBN}$
- $dim\left(\mathcal{P}^{\mathcal{G}}\right) = \sum_{i=1}^{n} \left((m_i - 1) \prod_{s \in \mathbf{pa}(i)} m_s \right) + 1$
- every decision function f is represented in $\mathcal{P}^{FBN}$ by an orthant

Geometric considerations

Consider the space $\mathcal{P}^{FBN}$ as the space of interpolating polynomials over $\mathbf{\Omega}$

- $\mathcal{P}^{FBN}$ is generated by $\delta_{\mathbf{k}}(\mathbf{x}) = \prod_{i=1}^{n} \ell_{k_i}^{\Omega_i}(x_i)$ (canonical basis)
- $\mathcal{P}^{\mathcal{G}}$ is a vectorial subspace of $\mathcal{P}^{FBN}$
- $dim\left(\mathcal{P}^{\mathcal{G}}\right) = \sum_{i=1}^{n} \left((m_i - 1) \prod_{s \in \mathbf{pa}(i)} m_s \right) + 1$
- every decision function f is represented in $\mathcal{P}^{FBN}$ by an orthant
- $\mathcal{P}^{\mathcal{G}}$ can sign-represent a given decision function if it intersects the corresponding orthant in $\mathcal{P}^{FBN}$

Bounding the expressive power

Corollary

BAN classifier without V-structures. Then

$$|sgn(\mathcal{P}_{\mathcal{G}})| \leq C(M,d) = 2\sum_{k=0}^{d-1}\binom{M-1}{k}$$

where $d = \sum_{i=1}^{n}\left((m_i-1)\prod_{s\in\mathbf{pa}(i)} m_s\right)+1$ *and* $M = \prod_{i=1}^{n} m_i$

Fraction of Decision Functions Representable

$$\frac{C(M,d)}{2^M} \to 0 \quad \text{as } n \to \infty$$

Varando, Bielza, Larrañaga (2015). **Decision boundary for discrete Bayesian network classifiers**. *Journal of Machine Learning research, vol. 16, pp. 2725-2749, 2015.*

Multi-Label Classification Problem

Multi-dimensional binary classification

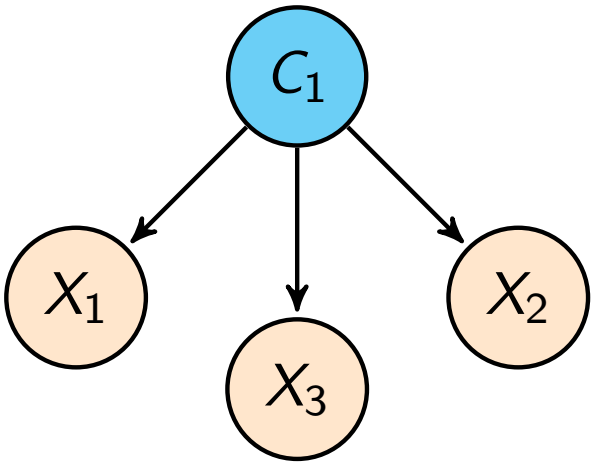
$$\begin{array}{rccc} \mathbf{f}: & \boldsymbol{\Omega} = \Omega_1 \times \cdots \times \Omega_n & \rightarrow & \{-1,+1\}^h \\ & (x_1,\ldots,x_n) & \mapsto & (c_1,\ldots,c_h) \end{array}$$

Equivalent to multi-label problem with h labels

$$\Omega_1 \times \cdots \times \Omega_n \rightarrow Y \subset \mathcal{Y} = \{y_1,\ldots,y_h\},$$

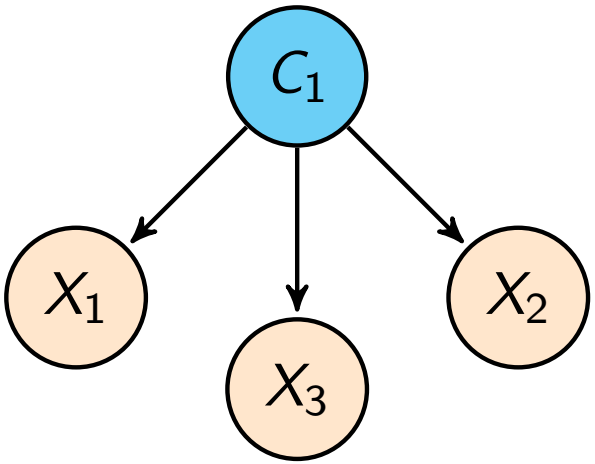
Binary Relevance Method

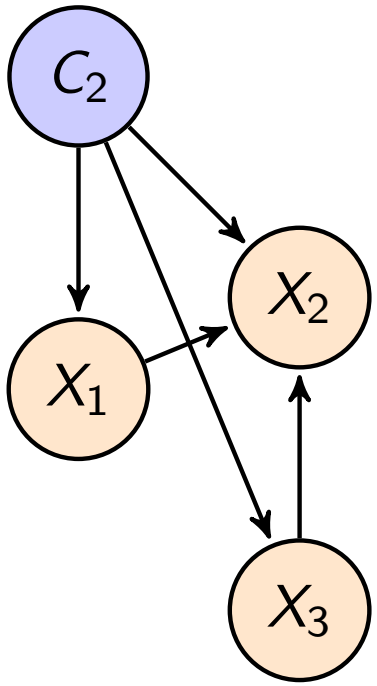
Binary Relevance (BR) method consists of building h independent one-dimensional classifier, one for each class variables $C_1, \ldots, C_h$



Binary Relevance Method

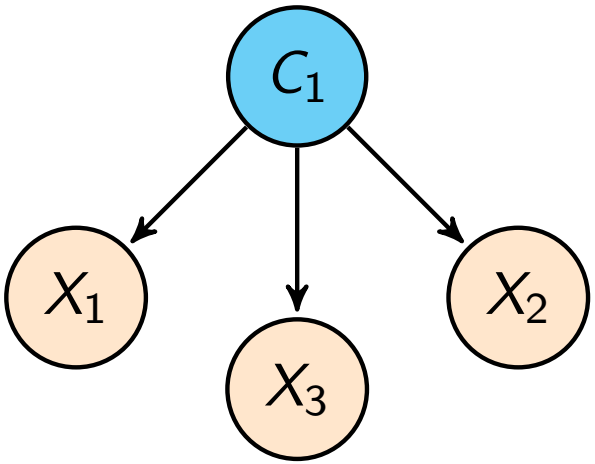
Binary Relevance (BR) method consists of building h independent one-dimensional classifier, one for each class variables $C_1, \ldots, C_h$

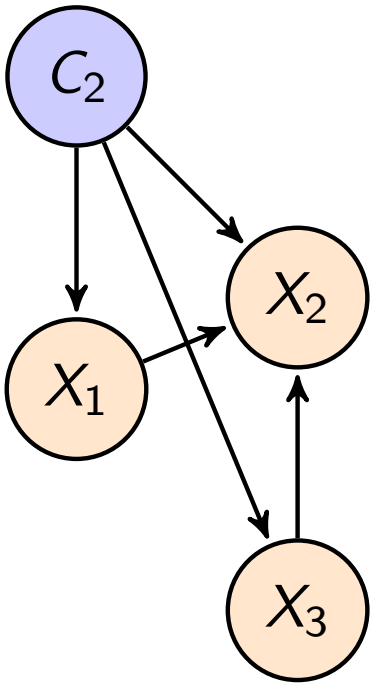


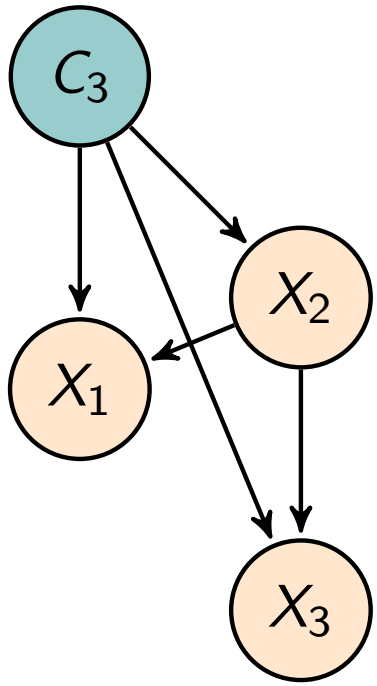


Binary Relevance Method

Binary Relevance (BR) method consists of building h independent one-dimensional classifier, one for each class variables $C_1, \ldots, C_h$







Binary Relevance Method

$\mathbf{f} = (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_h(\mathbf{x}))$ is the multi-valued decision function induced by the h BAN classifiers

$$\Downarrow$$

there exist $p_1(\mathbf{x}), \ldots, p_h(\mathbf{x}) \in \mathcal{P}_{\mathcal{G}}$ such that

$$f_i(\mathbf{x}) = sgn(p_i(\mathbf{x})) \text{ for every } i \in \{1, \ldots, h\}$$

$N(\mathcal{G}, h)$ = number of h-valued decision functions representable by Binary Relevance method using BAN classifier with predictor subgraph $\mathcal{G}$

$$N(\mathcal{G}, h) \leq C(M, d)^h$$

Binary Relevance Method

$\mathbf{f} = (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_h(\mathbf{x}))$ is the multi-valued decision function induced by the h BAN classifiers

$$\Downarrow$$

there exist $p_1(\mathbf{x}), \ldots, p_h(\mathbf{x}) \in \mathcal{P}_{\mathcal{G}}$ such that

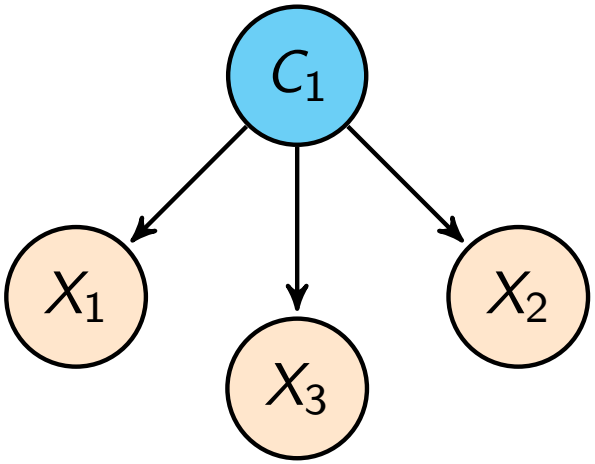
$$f_i(\mathbf{x}) = sgn(p_i(\mathbf{x})) \text{ for every } i \in \{1, \ldots, h\}$$

$N(\mathcal{G}, h)$ = number of h-valued decision functions representable by Binary Relevance method using BAN classifier with predictor subgraph $\mathcal{G}$

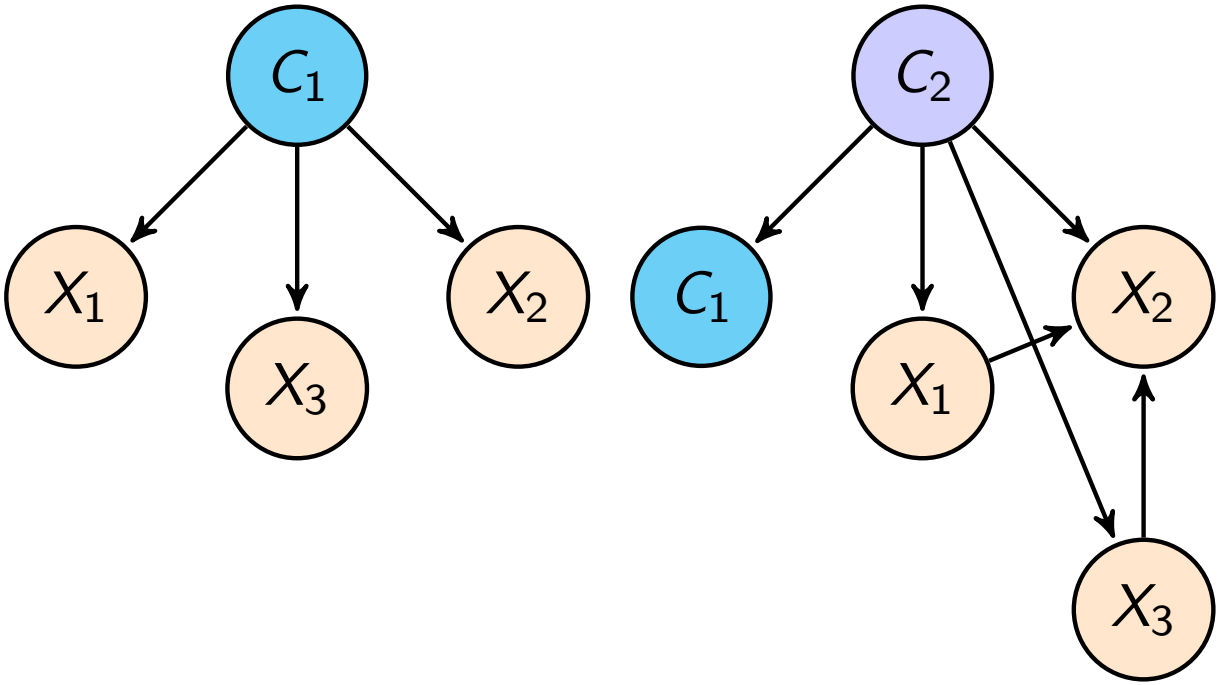
$$N(\mathcal{G}, h) \leq C(M, d)^h$$

$$\frac{N(\mathcal{G}, h)}{2^{hM}} \leq \left(\frac{C(M, d)}{2^M} \right)^h \to 0$$

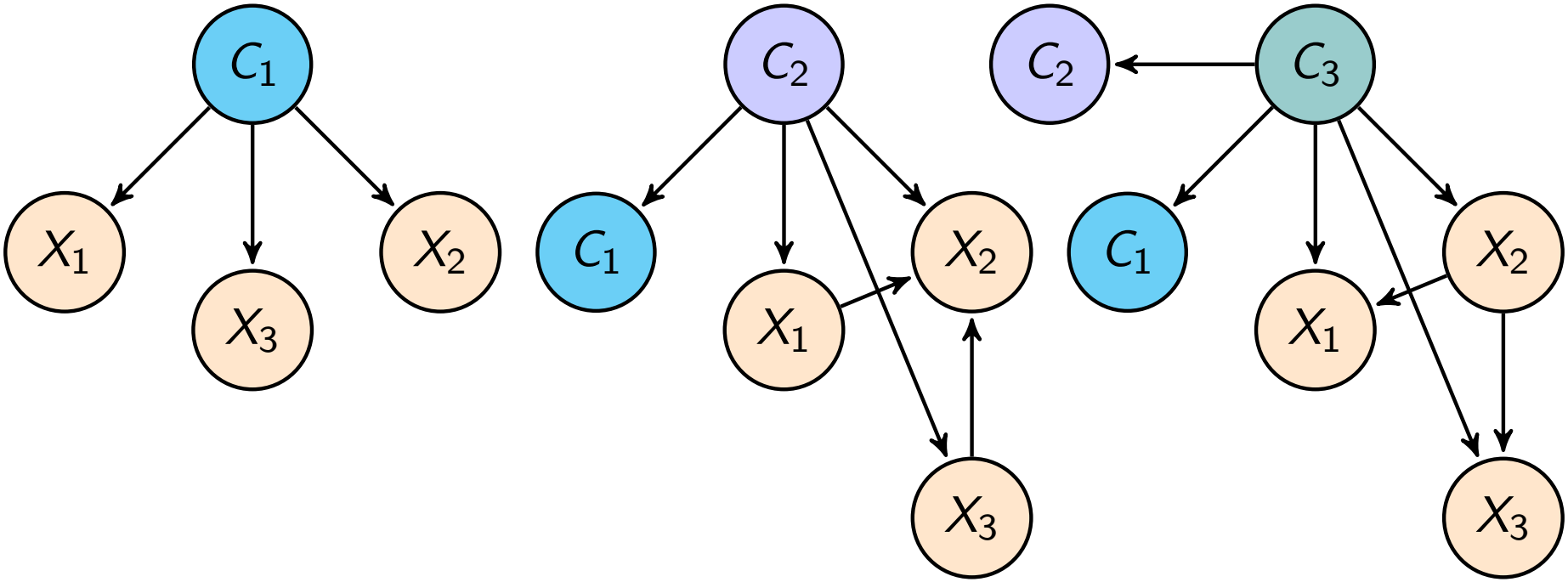
Chain classifier

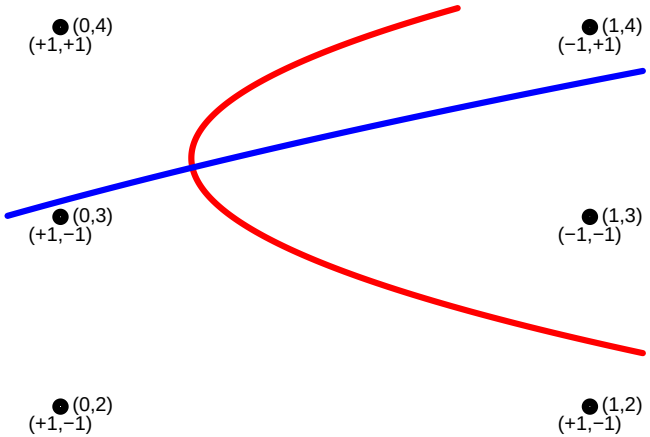


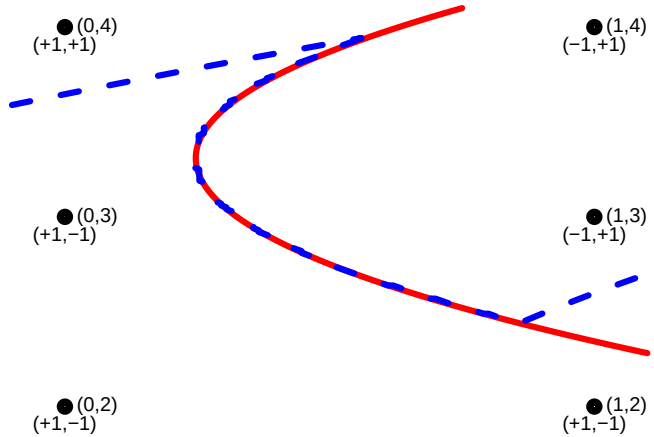
Chain classifier



Chain classifier







$$sgn\left(\hat{q}_k(\mathbf{x}) + \sum_{j=1}^{k-1}\left[\beta_j(-1)\ell_{-1}^{\{-1,+1\}}(\hat{c}_j) + \beta_j(+1)\ell_{+1}^{\{-1,+1\}}(\hat{c}_j)\right]\right)$$

$$|\mathcal{F}\setminus\mathcal{D}| > |\mathbf{\Omega}|\sum_{k=1}^{h-1}\binom{(h-1)}{k}2^k = |\mathbf{\Omega}|\left(3^{h-1}-1\right)$$

Varando, Bielza, Larrañaga (2015). **Decision functions for chain classifiers based on BN for multi-label classification**. *International Journal of Approximate Reasoning*

Some more geometrical observations...

Return to the simple naive Bayes, denote $\ell_j^{\Omega_i} = \ell_{i,j}$

$$\mathcal{P}^{NB} = \left\{ \sum_{i=1}^{n} \sum_{j=1}^{m_i} \alpha_i(j) \ell_{i,j}(x_i) \quad \alpha_i(j) \in \mathbb{R} \right\},$$

- $$1 = \ell_0 = \sum_{j=1}^{m_i} \ell_{i,j} \quad \forall i = 1, \ldots, n$$

Some more geometrical observations...

Return to the simple naive Bayes, denote $\ell_j^{\Omega_i} = \ell_{i,j}$

$$\mathcal{P}^{NB} = \left\{ \sum_{i=1}^{n} \sum_{j=1}^{m_i} \alpha_i(j) \ell_{i,j}(x_i) \quad \alpha_i(j) \in \mathbb{R} \right\},$$

- $$1 = \ell_0 = \sum_{j=1}^{m_i} \ell_{i,j} \quad \forall i = 1, \ldots, n$$
- we can use the following basis:

$$B = \left\{ \ell_{i,j}, \begin{array}{c} i = 1, \ldots, n \\ j = 2, \ldots, m_i \end{array} \right\} \cup \{\ell_0\}$$

Some more geometrical observations...

Return to the simple naive Bayes, denote $\ell_j^{\Omega_i} = \ell_{i,j}$

$$\mathcal{P}^{NB} = \left\{ \sum_{i=1}^{n} \sum_{j=1}^{m_i} \alpha_i(j) \ell_{i,j}(x_i) \quad \alpha_i(j) \in \mathbb{R} \right\},$$

- $$1 = \ell_0 = \sum_{j=1}^{m_i} \ell_{i,j} \quad \forall i = 1, \ldots, n$$
- we can use the following basis:

$$B = \left\{ \ell_{i,j}, \begin{array}{c} i = 1, \ldots, n \\ j = 2, \ldots, m_i \end{array} \right\} \cup \{\ell_0\}$$

- scalar product over $\mathcal{P}^{FBN}$

$$\langle p, q \rangle = \frac{1}{|\Omega|} \sum_{\mathbf{x} \in \Omega} p(\mathbf{x}) q(\mathbf{x})$$

Dual basis

With respect to the scalar product defined the dual basis is

$$B^* = \left\{ \ell^{i,j}, \begin{array}{c} i = 1, \dots, n \\ j = 2, \dots, m_i \end{array} \right\} \cup \{\ell^0\}$$

where

$$\ell^{i,j} = m_i(\ell_{i,j} - \ell_{i,1})$$

$$\ell^0 = \ell_0 - \sum_{i'=1}^{n} \sum_{j'=2}^{m_i} \frac{1}{m_{i'}} \ell^{i',j'}$$

Dual basis

With respect to the scalar product defined the dual basis is

$$B^* = \left\{ \ell^{i,j}, \begin{array}{c} i = 1, \ldots, n \\ j = 2, \ldots, m_i \end{array} \right\} \cup \{\ell^0\}$$

where

$$\ell^{i,j} = m_i(\ell_{i,j} - \ell_{i,1})$$

$$\ell^0 = \ell_0 - \sum_{i'=1}^{n} \sum_{j'=2}^{m_i} \frac{1}{m_{i'}} \ell^{i',j'}$$

so we can define a projection over $\mathcal{P}^{NB}$

$$Pr_{NB} : \mathcal{P} \mapsto \mathcal{P}^{NB}$$

$$Pr_{NB}(p) = \sum_{i=1}^{n} \sum_{j=2}^{m_i} \langle p, \ell^{i,j} \rangle \ell_{i,j} + \langle p, \ell^0 \rangle \ell_0$$

Scalar product and error

- f is the "true" decision function

Scalar product and error

- f is the "true" decision function
- $\hat{f}$ the estimated one

Scalar product and error

- f is the "true" decision function
- $\hat{f}$ the estimated one

Scalar product and error

- f is the "true" decision function
- $\hat{f}$ the estimated one
- $arg\,\min_{\hat{f}} err(f, \hat{f}) = arg\,\max_{\hat{f}} E[f(\mathbf{x})\hat{f}(\mathbf{x})] = arg\,\max_{\hat{f}} \langle Pf, \hat{f} \rangle$

Scalar product and error

- f is the "true" decision function
- $\hat{f}$ the estimated one
- $arg\min_{\hat{f}} err(f, \hat{f}) = arg\max_{\hat{f}} E[f(\mathbf{x})\hat{f}(\mathbf{x})] = arg\max_{\hat{f}} \langle Pf, \hat{f} \rangle$
- but $sign(\cdot)$ is non-linear

Scalar product and error

- f is the "true" decision function
- $\hat{f}$ the estimated one
- $arg\min_{\hat{f}} err(f, \hat{f}) = arg\max_{\hat{f}} E[f(\mathbf{x})\hat{f}(\mathbf{x})] = arg\max_{\hat{f}} \langle Pf, \hat{f} \rangle$
- but $sign(\cdot)$ is non-linear
- no direct link between $arg\max_{r \in \mathcal{P}^{NB}} \langle Pf, sign(r) \rangle$ and $arg\max_{r \in \mathcal{P}^{NB}} \langle Pf, r \rangle$

- $f = sign\left(\ln\left(\frac{P(\mathbf{x},+1)}{P(\mathbf{x},-1)}\right)\right)$ the Bayes decision function

- $f = sign\left(\ln\left(\frac{P(\mathbf{x},+1)}{P(\mathbf{x},-1)}\right)\right)$ the Bayes decision function
- $\mathcal{R} = arg\max_{r \in \mathcal{P}^{NB}} \langle P(\mathbf{x}) f, sign(r) \rangle$

- $f = sign\left(\ln\left(\frac{P(\mathbf{x},+1)}{P(\mathbf{x},-1)}\right)\right)$ the Bayes decision function
- $\mathcal{R} = arg\max_{r \in \mathcal{P}^{NB}} \langle P(\mathbf{x}) f, sign(r) \rangle$
- $Pr_{NB}(Pf) \notin \mathcal{R}$ in general

- $f = sign\left(\ln\left(\frac{P(\mathbf{x},+1)}{P(\mathbf{x},-1)}\right)\right)$ the Bayes decision function
- $\mathcal{R} = arg\max_{r \in \mathcal{P}^{NB}} \langle P(\mathbf{x}) f, sign(r) \rangle$
- $Pr_{NB}(Pf) \notin \mathcal{R}$ in general
- $\exists?\ g : \mathbf{\Omega} \mapsto \mathbb{R}$ s.t. $f = sign(g)$ and $Pr_{NB}(g) \in \mathcal{R}$

- $f = sign\left(\ln\left(\frac{P(\mathbf{x},+1)}{P(\mathbf{x},-1)}\right)\right)$ the Bayes decision function
- $\mathcal{R} = arg\max_{r \in \mathcal{P}^{NB}} \langle P(\mathbf{x}) f, sign(r) \rangle$
- $Pr_{NB}(Pf) \notin \mathcal{R}$ in general
- $\exists?\ g : \mathbf{\Omega} \mapsto \mathbb{R}$ s.t. $f = sign(g)$ and $Pr_{NB}(g) \in \mathcal{R}$
- $\exists?\phi$ s.t. $\forall \mathcal{D}$ training set, $g_{\mathcal{D}} = \phi(P_{\mathcal{D}})$ $\hat{f}_{\mathcal{D}} = sign(g_{\mathcal{D}})$ and $Pr_{NB}(g_{\mathcal{D}}) \in \mathcal{R}_{\mathcal{D}}$

Polyonomial representation of Bayesian network classifiers

Gherardo Varando

Computational Intelligence Group
Universidad Politécnica de Madrid

23 January 2017

Grazie per l'attenzione