Redes Bayesianas Gerais para Classificação: Métodos de Estimação e Comparação

Camila Ozelame [1], **Anderson Ara** [2], **Francisco Louzada Neto** [3]

1 Introdução

As Redes Bayesianas são grafos acíclicos direcionados, os DAGs (do inglês em tradução livre, "*Directed Acyclic Graphs*"). Elas assumem que todas as variáveis são aleatórias, essas variáveis são consideradas os nós na rede e a dependência condicional entre elas é representada por arcos Cheng e Greiner (2001). Pela sua versatilidade e flexibilidade de modelos, as redes são amplamente utilizadas com diversos objetivos e um deles é a tarefa de classificação. Para tanto, elas fornecem a distribuição a posteriori de uma variável resposta dado um conjunto de variáveis explicativas Bobbio et al. (2001).

2 Métodos

2.1 Redes Bayesianas

Um modelo de Rede Bayesiana para um conjunto de d variáveis $\mathbf{X} = \{X_1, X_2, ..., X_d\}$ é composto por um parâmetros que especificam as distribuições de probabilidade condicional que quantificam os arcos, um grafo acíclico direcionado que é composto por vértices (ou nós), que correspondem às variáveis e por arcos, que codificam as (in)dependências probabilísticas entre trios de variáveis Bielza e Larrañaga (2014), essa é a representação gráfica da distribuição conjunta de probabilidade de $\mathbf{X}$.

Tais representações satisfazem a *Condição de Markov* que diz que se para todo $X_i \in V$, X_i é condicionalmente independente do conjunto de todos os seus não descendentes dado o conjunto de seus pais. Além disso, satisfazer essa condição garante a propriedade do DAG chamada d-separação - a qual dois nós são d-separados se existe um outro nó "bloqueando"o caminho entre eles - ou seja, toda d-separação em $\mathbb{G}$ garante independência condicional assim como toda independência condicional é identificada por d-separação Neapolitan et al. (2004).

2.2 Classificadores Bayesianos

Os classificadores são tipos especiais de estruturas das Redes Bayesianas que tem o propósito de classificação o que requer que a variável resposta seja categórica, ou numérica discreta. A tarefa de predição se dá a partir da escolha da classe que maximiza a probabilidade a posteriori (MAP) nas d variáveis preditivas, Y a variável resposta com c classes, obtida pela regra de Bayes Ruz e Araya-Díaz (2018), como segue:

$$k_{pred} = \arg\max_k P(Y = k | X_1 = x_1, ..., X_n = x_n) =$$

$$\arg\max_k \beta P(Y = k) P(X_1 = x_1, ..., X_n = x_n | Y = k)$$

[1]PIPGEs - UFSCar/USP. e-mail: *ozelamecs@usp.br*

[2]Universidade Federal da Bahia. e-mail: *anderson.ara@ufba.br*

[3]Universidade de São Paulo. e-mail: *louzada@icmc.usp.br*

Sendo $\beta = 1/\sum_{y=1}^{c} P(X_1 = x_1, ..., X_n = x_n|Y = y)$ a constante normalizadora, a *priori* $P(Y = k)$ e a *verossimilhança* $P(X_1 = x_1, ..., X_n = x_n|Y = k)$ que é a distribuição conjunta condicionada ao valor da classe.

Na literatura, existem inúmeros classificadores publicados, os mais utilizados são os de estrutura fixa ou semi-fixa que tem por característica fazer suposições a respeito da relação entre variáveis. Nesse estudo, a estrutura do classificador tem por característica considerar a variável resposta como uma variável ordinária - sem colocar restrição alguma a quantidade de pais dos nós -, aprender sua estrutura e utilizar apenas o *Markov Blanket* - conjunto de variáveis que faz com que a classe seja condicionalmente independente dos outros atributos - para predição Cheng e Greiner (1999), esse classificador é conhecido como *GBN (General Bayesian Network)*, ou Rede Bayesiana Geral.

2.3 Estimação de Estrutura do Classificador

Para a estimação da estrutura da rede mais plausíveis, os algoritmos de aprendizado podem ser classificados heurísticas baseadas em testes, conhecidas como *constraint learning*, baseadas em medidas ou *score-based learning* ou híbrida que é uma combinação entre as anteriores Scutari e Denis (2014).

2.3.1 Constraint Learning

Os algoritmos dessa categoria se baseiam em testes de independência condicional para encontrar a estrutura de relação mais plausível entre as variáveis. Os seguintes métodos foram utilizados no trabalho:

- O PC assume fidelidade aos testes de independência entre as variáveis, fazendo com que o grafo traduza exatamente as relações de (in)dependência entre elas Abellán et al. (2006).
- O Grow-Shrink é baseado no algoritmo de busca do Markov Blanket induzindo a estrutura de RB Margaritis (2003).
- No Incremental Association Markov Blanket (IAMB) para cada variável continua a procura de um conjunto hipotético de possíveis pais, onde o objetivo é obter, no final do algoritmo, o Markov Blanket. Possui duas fases, na primeira acontece a maximização de uma função da heurística e na segunda, os testes de independência são realizados Beretta et al. (2018). Existem algumas variações do algoritmo que visam otimizar a detecção do Markov Blanket ou testes de independência desnecessários.

Os testes de independência condicional que foram considerados no estudo foram: *Informação Mútua* que mede a quantidade de informação compartilhada por duas variáveis; *Estimador de encolhimento da Informação Mútua* que é um teste χ^2 assintótico melhorado baseado no estimador de informação mútua; e o clássico teste χ^2 de Pearson para tabelas de contingência Scutari (2010).

2.3.2 Score and Search Learning

Os métodos score-based consistem em dois algoritmos principais:

- Hill-Climbing que a cada passo seleciona o melhor modelo dada uma métrica pré estabelecida;

- Tabu Searching que consistem em um HC modeficado o qual procura evitar o máximo local selecionando uma rede que diminui minimamente a função score, no intuito de melhorar a eficiência da busca.

Como o nome sugere, score-based são baseados em funções score é definida como uma métrica de ajuste entre a estrutura gráfica e os dados Behjati e Beigy (2018), utilizadas em parte do processo de otimização. As métricas do estudo foram: o Critério de Informação de Akaike (AIC); o Critério de Informação Bayesiana (BIC), que é equivalente ao Minimum Description Length (MDL); o logaritmo da métrica equivalente Dirichlet Bayesiana (BDE); o logaritmo da média local Dirichlet Bayesiana (BDLA); e o logaritmo da métrica K2 Scutari (2010).

2.4 Métricas de Comparação

Parte das métricas de avaliação utilizadas nos modelos de predição propostos, amplamente são baseadas na matriz de confusão que compara a quantidade de indivíduos classificados em cada uma das categorias e qual era sua categoria original. Foram utilizadas: Acurácia (ACC) que é a fração de predições corretas do modelo; F1 Score que é um balanço entre a precisão - proporção de verdadeiros valores positivos dado valores preditos positivos - e o *recall* - proporção de verdadeiros positivos dentre os que realmente são positivos; Coeficiente de Correlação de Matthew (MCC) uma medida utilizada para verificar a classificação geral do modelo e é interpretada similarmente ao coeficiente de correlação de Pearson Louzada e Ara (2012) e o Spherical Payoff (SP) que dimensiona a qualidade de preditiva da classificação, varia de 0-1 onde 1 é a melhor performance do modelo Marcot (2012).

Para avaliar as técnicas estudadas, utilizou-se o método de *cross-validation* sugerido por Witten et al. (2016) para suavizar o viés de escolha dos dados de treinamento e teste. O procedimento é chamado de *10-times 10-fold cross-validation* que segundo ele, a divisão do banco em dez *folds* é ideal para uma boa estimação do erro, para diminuir a variância da aleatoriedade dos conjuntos, esse procedimento deve ser repetido 10 vezes e a média é a métrica utilizada.

2.5 Simulação

A simulação foi feita a partir das estruturas apresentadas na Figura 1, onde as variáveis são os nós do grafo e a (in)dependência entre elas são dadas pela direção das setas, como definido anteriormente.

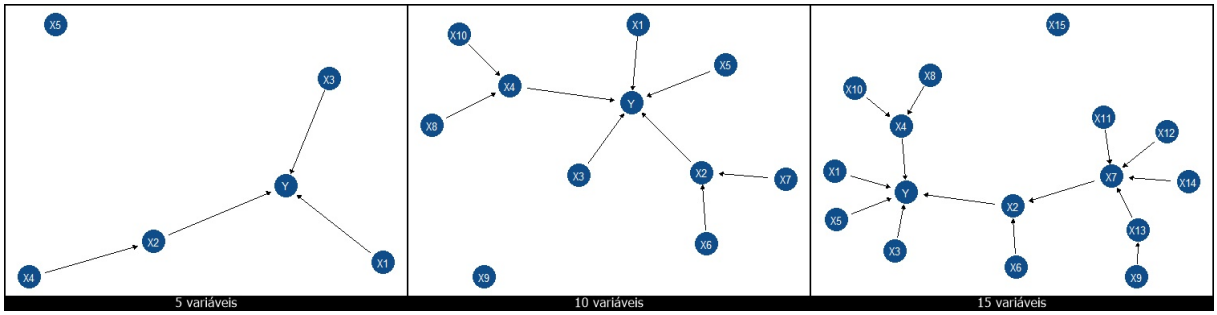

Figura 1: Estruturas Simuladas

A quantidade de valores para a comparação foi de 5000 com discretização em 2 categorias para a variável resposta e 3 categorias para as demais variáveis - alguns trabalhos anteriores sugeriram a utilização desse número de classes na discretização, para uma melhor performance dos algoritmos estudados.

3 Resultados

As análises foram realizadas por meio do *software* `R` R Core Team (2018) foi utilizado com auxílio da interface `RStudio` e do pacote `bnlearn` Scutari (2010). Depois de realizado o procedimento de *cross-validation* para as medidas descritas na subseção 2.4 que estão apresentadas na Figura 2, bem como a variação de cada um dos métodos.

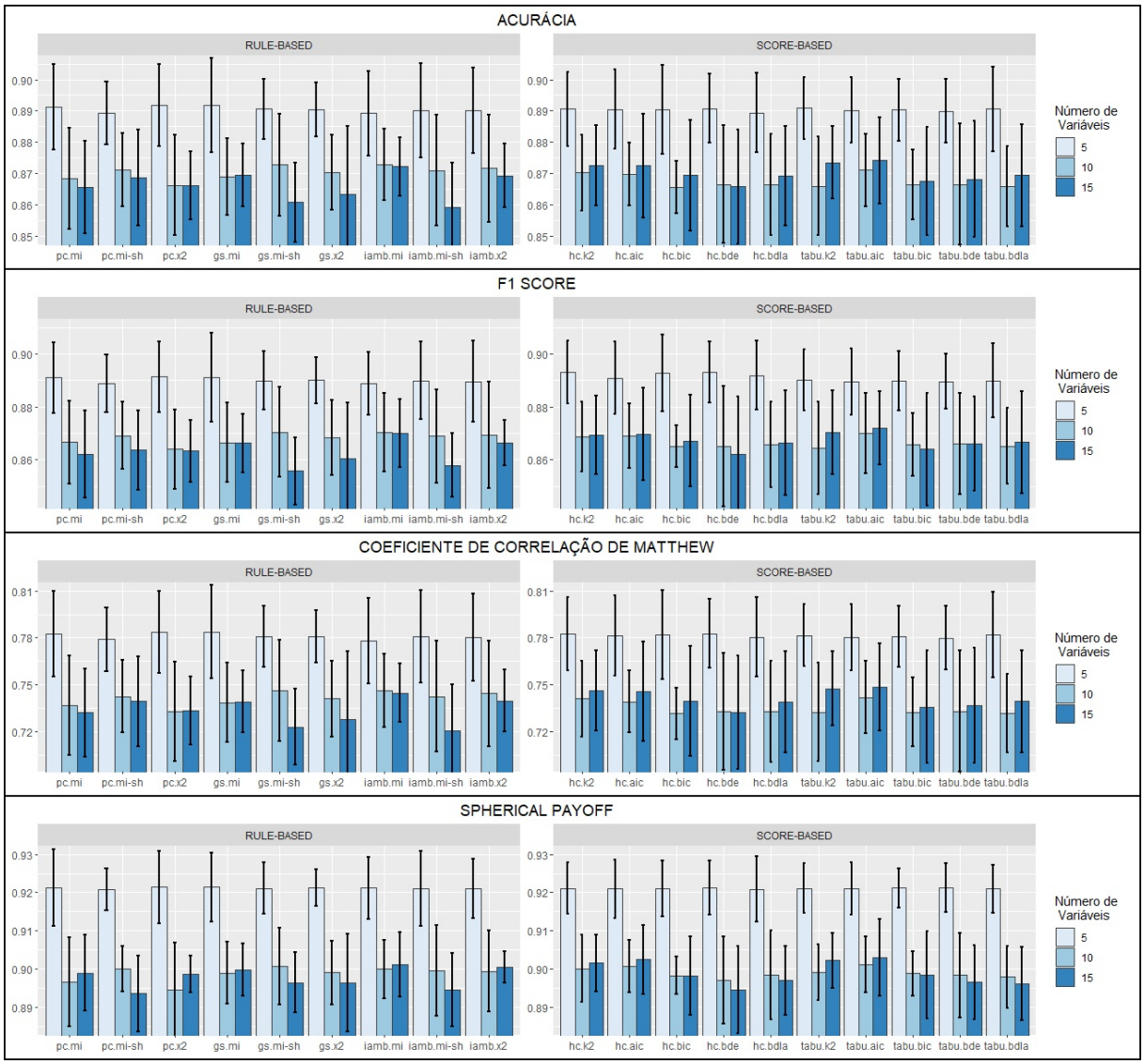

Figura 2: Medidas de Desempenho para cada uma das simulações.

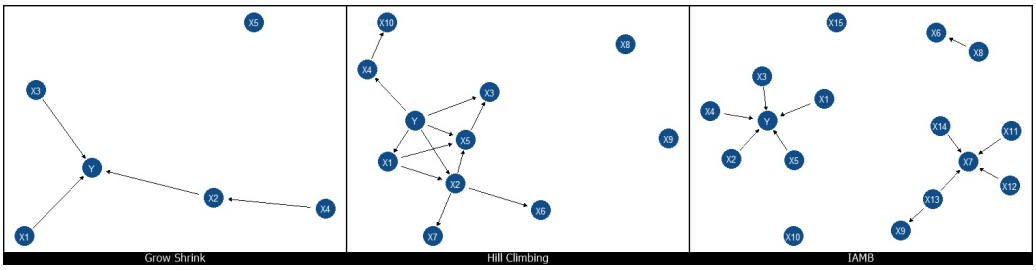

Figura 3: Estruturas com maiores medidas de desempenho para cada um dos conjuntos de variáveis.

4 Discussão

De acordo com os resultados apresentados na seção 3, os algoritmos baseados em testes (*rule-based*) apresentaram melhores valores para as medidas de desempenho tanto para a simulação com 5 quanto para com 15 variáveis explicativas, os métodos baseados em métricas (*score-based*) apresentaram melhores valores nas medidas bem para a simulação com 10 variáveis. As estruturas geradas para cada um dos melhores colocados está apresentada na Figura 3. No primeiro grafo a estrutura é igual a simulada e foi estimada pelo método *Grow Shrink*, com o teste de correlação de Pearson. Já no segundo grafo a rede estimada contém 10 além da resposta e foi o método *Hill Climbing* que teve os melhores valores, mesmo com mais conexões que a rede simulada. E no terceiro grafo o método de IAMB (*Incremented Association Markov Blanket*) que teve melhor colocação tanto para teste de Informação Mútua quanto para correlação de Pearson.

5 Conclusão

Conclui-se que os métodos tiveram desempenho semelhante entre os conjuntos simulados com pequena variação. Os métodos baseados em testes (PC, GS, IAMB) obetiveram medidas levemente maiores na maioria dos cenários, principalmente para o conjunto com menor número de variáveis para os testes de informação mútua e correlação de Pearson. Nos métodos baseados em métricas (TABU e HC) obtiveram melhores resultados as medidas de AIC, BIC e K2.

Agradecimentos

À Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) pela concessão da bolsa de mestrado.

Referências

ABELLÁN, J. et al. Some variations on the pc algorithm. In: *Probabilistic Graphical Models*. [S.l.: s.n.], 2006. p. 1–8.

BEHJATI, S.; BEIGY, H. An order-based algorithm for learning structure of bayesian networks. In: *International Conference on Probabilistic Graphical Models*. [S.l.: s.n.], 2018. p. 25–36.

BERETTA, S. et al. Learning the structure of bayesian networks: A quantitative assessment of the effect of different algorithmic schemes. *Complexity*, Hindawi, v. 2018, 2018.

BIELZA, C.; LARRAÑAGA, P. Discrete bayesian network classifiers: a survey. *ACM Computing Surveys (CSUR)*, ACM, v. 47, n. 1, p. 5, 2014.

BOBBIO, A. et al. Improving the analysis of dependable systems by mapping fault trees into bayesian networks. *Reliability Engineering & System Safety*, Elsevier, v. 71, n. 3, p. 249–260, 2001.

CHENG, J.; GREINER, R. Comparing bayesian network classifiers. In: MORGAN KAUFMANN PUBLISHERS INC. *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*. [S.l.], 1999. p. 101–108.

CHENG, J.; GREINER, R. Learning bayesian belief network classifiers: Algorithms and system. In: SPRINGER. *Conference of the Canadian Society for Computational Studies of Intelligence*. [S.l.], 2001. p. 141–151.

LOUZADA, F.; ARA, A. Bagging k-dependence probabilistic networks: An alternative powerful fraud detection tool. *Expert Systems with Applications*, Elsevier, v. 39, n. 14, p. 11583–11592, 2012.

MARCOT, B. G. Metrics for evaluating performance and uncertainty of bayesian network models. *Ecological modelling*, Elsevier, v. 230, p. 50–62, 2012.

MARGARITIS, D. *Learning Bayesian network model structure from data*. [S.l.], 2003.

NEAPOLITAN, R. E. et al. *Learning bayesian networks*. [S.l.]: Pearson Prentice Hall Upper Saddle River, NJ, 2004. v. 38.

R Core Team. *R: A Language and Environment for Statistical Computing*. Vienna, Austria, 2018. Disponível em: ⟨https://www.R-project.org/⟩.

RUZ, G. A.; ARAYA-DÍAZ, P. Predicting facial biotypes using continuous bayesian network classifiers. *Complexity*, Hindawi, v. 2018, 2018.

SCUTARI, M. Learning bayesian networks with the bnlearn R package. *Journal of Statistical Software*, v. 35, n. 3, p. 1–22, 2010.

SCUTARI, M.; DENIS, J.-B. *Bayesian networks: with examples in R*. [S.l.]: Chapman and Hall/CRC, 2014.

WITTEN, I. H. et al. *Data Mining: Practical machine learning tools and techniques*. [S.l.]: Morgan Kaufmann, 2016.