Learning Bayesian network classifiers with applications

Gonzalo A. Ruz, PhD

Faculty of Engineering and Sciences, University Adolfo Ibáñez, Santiago, Chile

Center of Applied Ecology and Sustainability (CAPES), Santiago, Chile

Data Observatory Foundation, Chile

Workshop on Machine and Statistical Learning with Applications

ANILLO ACT210096

June 14, 2023

Outline

- Background
    - Bayesian networks
    - Probabilistic classification
- Learning TAN classifiers using a Bayes factor approach
    - Application 1: Twitter sentiment analysis
- Learning TAN classifiers using an evolutionary computation approach
    - Application 2: Facial biotype classification for orthodontic treatment planning

What is a Bayesian network?

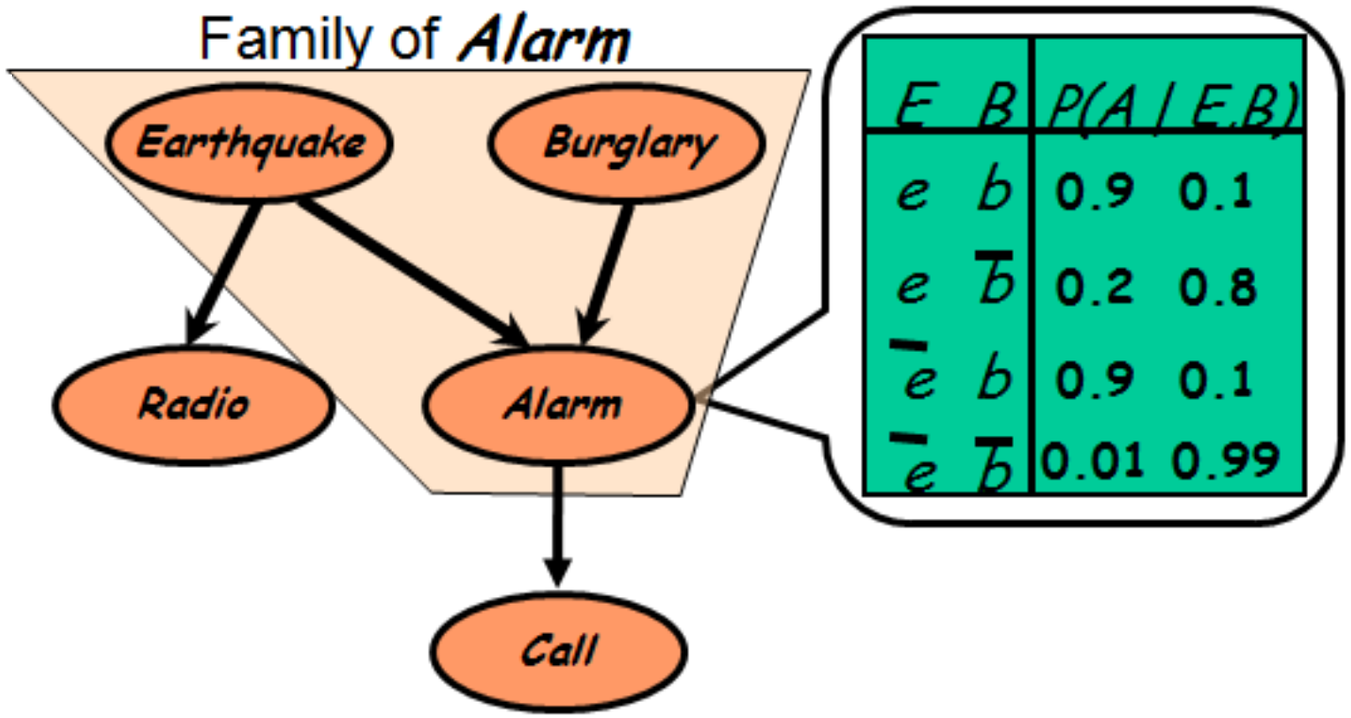


Figure from N. Friedman

What is a Bayesian network?

Qualitative part:

Directed acyclic graph (DAG)

- Nodes: random vars.
- Edges: direct influence

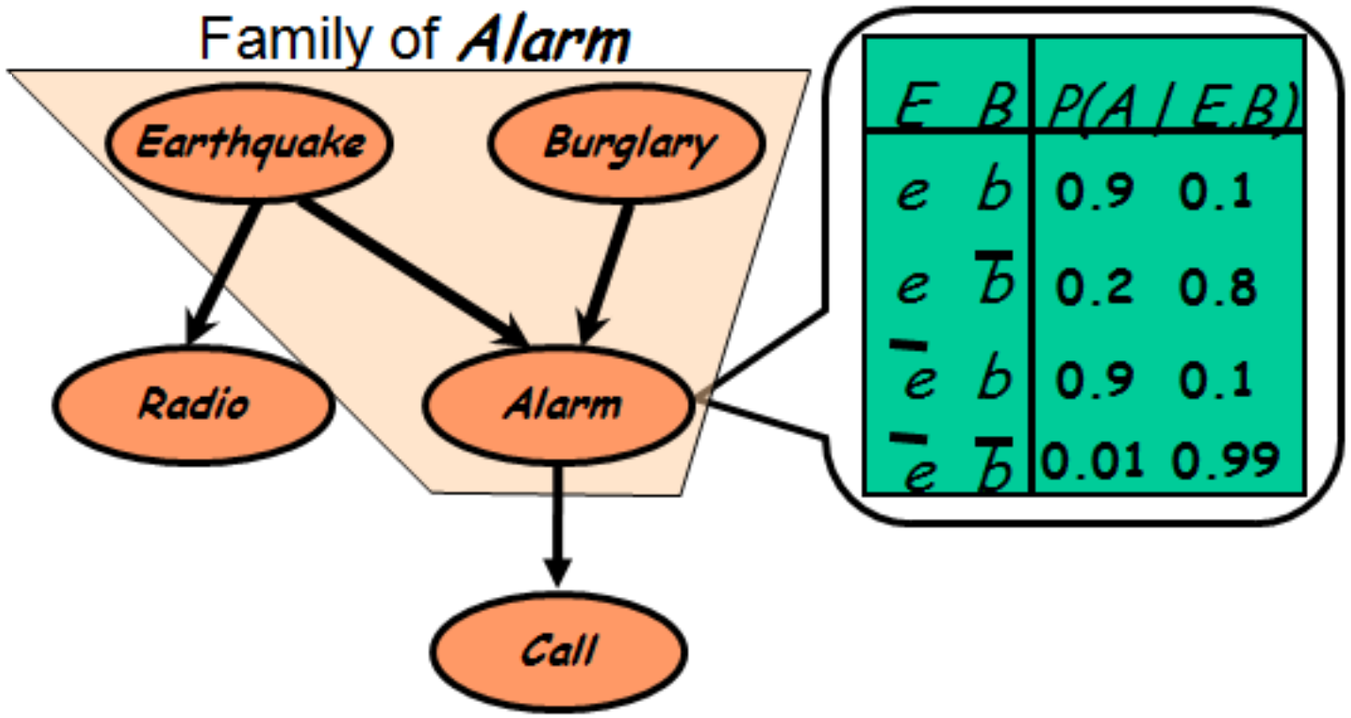


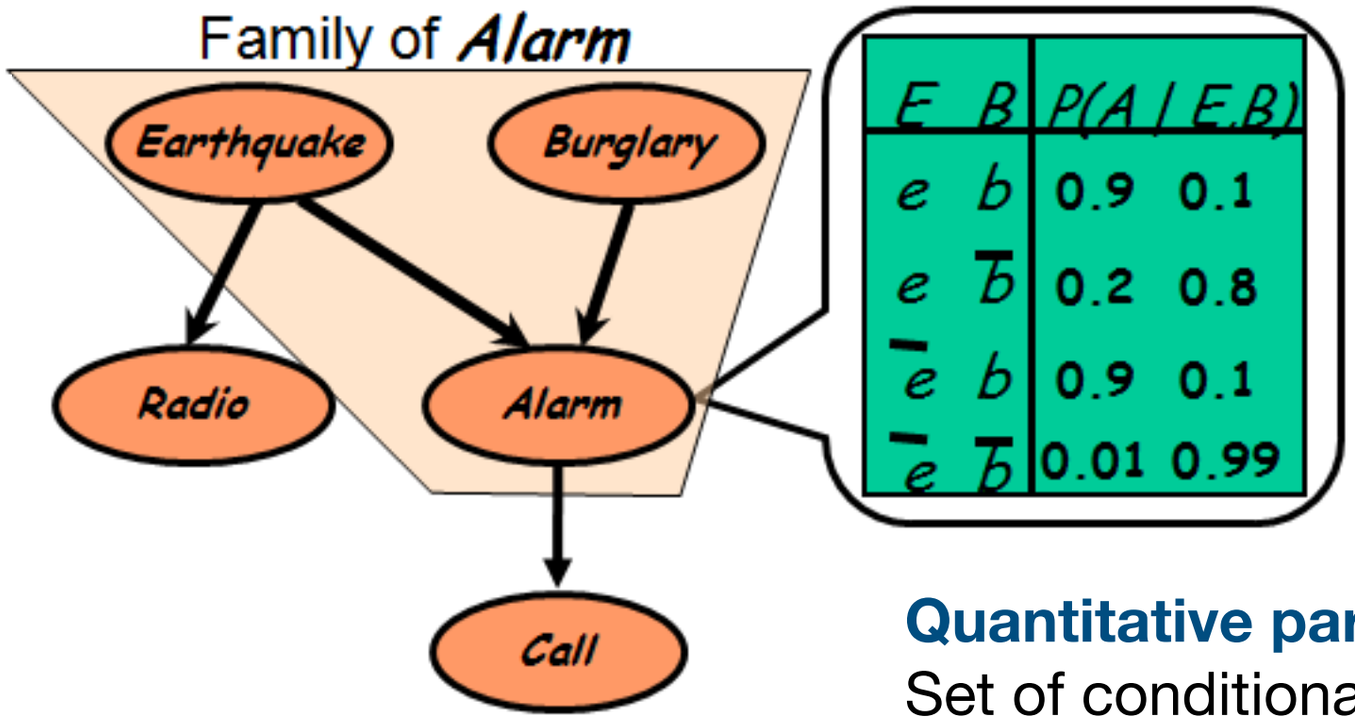
Figure from N. Friedman

What is a Bayesian network?

Qualitative part:

Directed acyclic graph (DAG)

- Nodes: random vars.
- Edges: direct influence



Quantitative part:

Set of conditional probability distributions

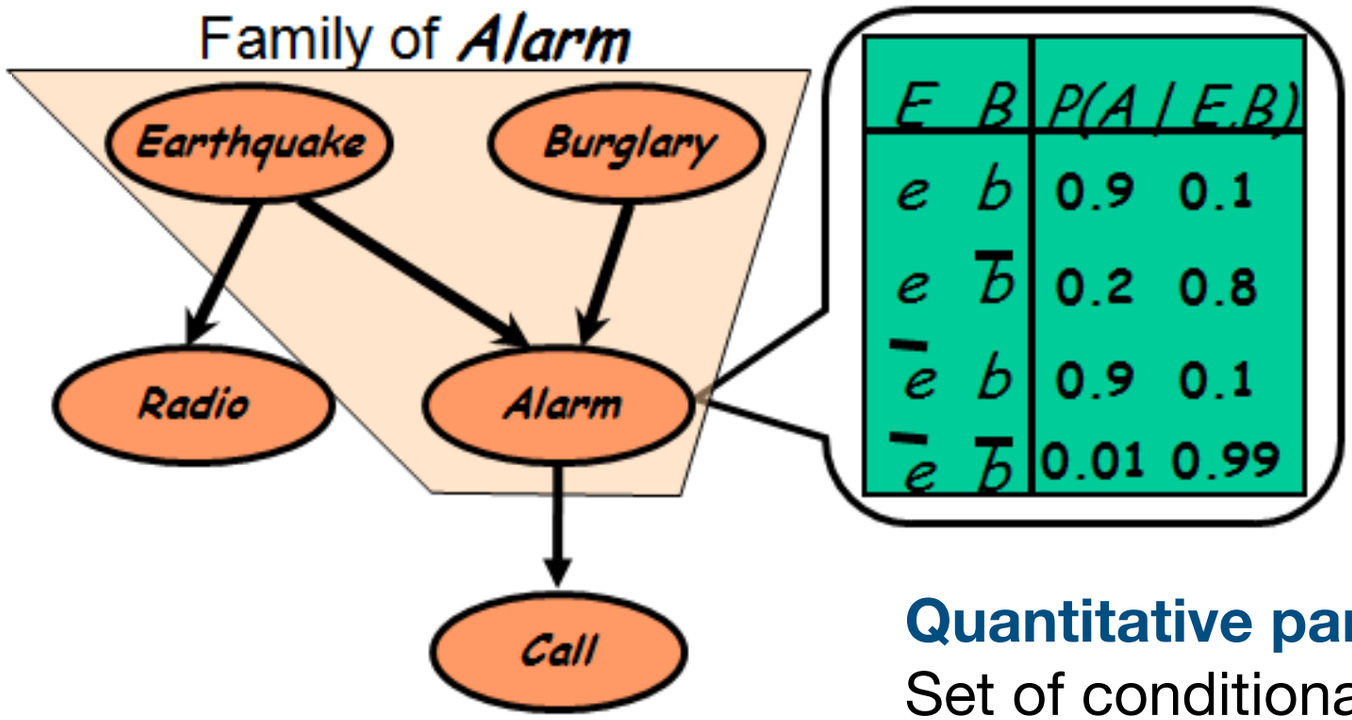
Figure from N. Friedman

What is a Bayesian network?

Qualitative part:

Directed acyclic graph (DAG)

- Nodes: random vars.
- Edges: direct influence



Quantitative part:

Set of conditional probability distributions

Together:

Define a unique distribution in a factored form

$$P(B, E, A, C, R) = P(B)P(E)P(A|B, E)P(R|E)P(C|A)$$

$$P(\boldsymbol{X}) = \prod_{i=1}^{n} P(X_i|\pi_i)$$

Figure from N. Friedman

The Markov Condition

- Let P be a joint probability distribution of the random variables in some set **X** and G = (**X**, **E**) a DAG.

- It is said that (G, P) satisfies the *Markov condition* if for each variable $X_i \in \mathbf{X}$, X_i is conditionally independent of the set of all its nondescendents given the set of all its parents.

Formal definition of a Bayesian network

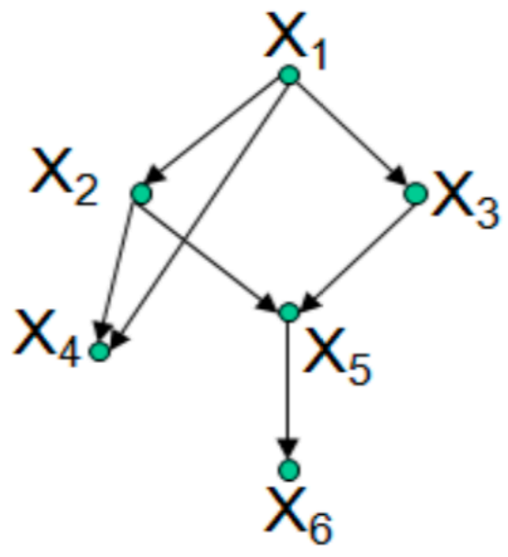
- Let P be a joint probability distribution of the random variables in some set **X** and G = (**X**, **E**) a DAG.

- Then, (G, P) is called a *Bayesian network* if (G, P) satisfies the Markov condition.

- Owing to the properties of the Markov condition, P is the product of its conditional distributions in G, and this is the way P is always represented in a Bayesian network.

Computing the joint probability distribution in a BN

$$P(X_1, X_2, \ldots, X_n) = \prod_i^n P(X_i | \Pi_{X_i})$$



$$P(X_1, X_2, X_3, X_4, X_5, X_6) =$$
$$P(X_6|X_5)P(X_5|X_2, X_3)P(X_4|X_1, X_2)P(X_3|X_1)P(X_2|X_1)P(X_1)$$

Some history

- Initial applications of Bayesian networks were for the representation of knowledge under conditions of uncertainty.

- In the beginning, most Bayesian networks were built by hand by a human expert.

- Machine Learning started developing algorithms to learn Bayesian networks from data (David Heckerman, 1995).

- Learning Bayesian networks from data is a difficult problem (NP-hard).

BN construction

- Constructing a BN involves two activities:
  - (1) structure learning - learning the structure of the network by discovering the underlying DAG of the domain.
  - (2) parameter learning - computing the conditional probabilities associated with each node of the given network structure.

Complexity in learning the structure

- Robinson [1977] showed that the number of DAGs containing n nodes is given by the following recursive expression:

$$f(n) = \sum_{i=1}^{n} (-1)^{i+1} \binom{n}{i} 2^{i(n-i)} f(n-i) \quad n > 2$$

$$f(0) = 1$$

$$f(1) = 1.$$

Complexity in learning the structure

- Robinson [1977] showed that the number of DAGs containing n nodes is given by the following recursive expression:

$$f(n) = \sum_{i=1}^{n} (-1)^{i+1} \binom{n}{i} 2^{i(n-i)} f(n-i) \quad n > 2$$

$$f(0) = 1$$

$$f(1) = 1.$$

Homework: show that f(2)=3, f(3)=25, f(5)=29000, and f(10)=4.2 x 10^{18}

Learning the structure of BNs automatically from data

1. Constraint-based method
2. Score-based method
3. Hybrid

Probabilistic classification

$$Y^{\text{predict}} = \underset{k}{\operatorname{argmax}} \quad P(Y = k | X_1 = x_1, \ldots, X_n = x_n)$$

Bayes T.

$$P(Y|X_1, \ldots, X_n) \propto P(Y)P(X_1, \ldots, X_n|Y)$$

Probabilistic classification

$$Y^{\text{predict}} = \underset{k}{\text{argmax}} \quad P(Y = k | X_1 = x_1, \ldots, X_n = x_n)$$

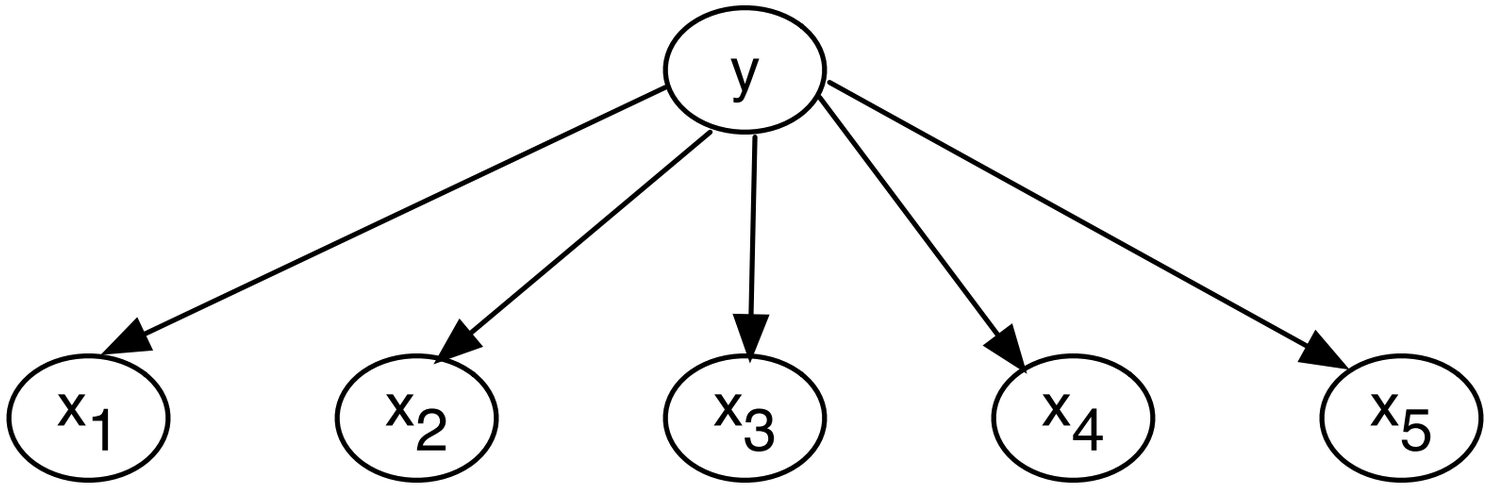
Bayes T.

$$P(Y|X_1, \ldots, X_n) \propto P(Y) \underbrace{P(X_1, \ldots, X_n|Y)}$$

Compute using BN

Bayesian network classifiers (1)

Naive Bayes network classifier

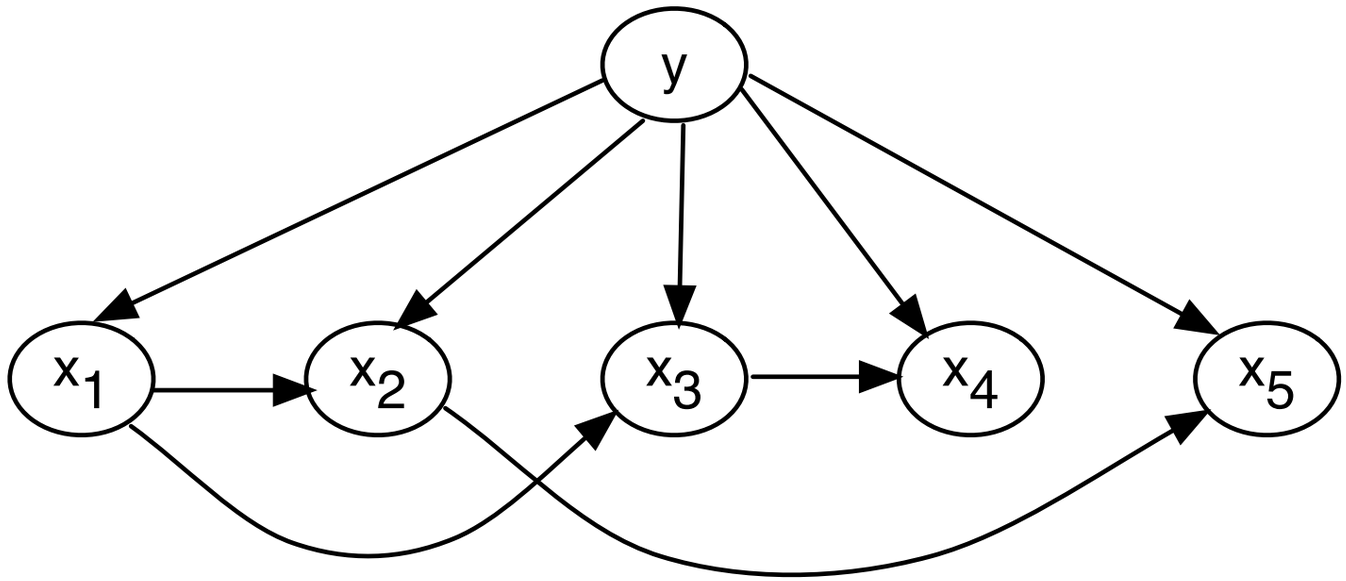


$$P(Y|X_1,\ldots,X_5) \propto P(Y)\prod_{i=1}^{5} P(X_i|\pi_i)$$

with $\pi_i = \{Y\}$

Bayesian network classifiers (2)

Tree augmented naive Bayes classifier (TAN)



$$P(Y|X_1,\ldots,X_5) \propto P(Y)\prod_{i=1}^{5} P(X_i|\pi_i)$$

with $\pi_1 = \{Y\}$

$\pi_i = \{X_j, Y\} \quad i = 2,\ldots,5$

$j \neq i$

TAN construction

- Uses conditional mutual information

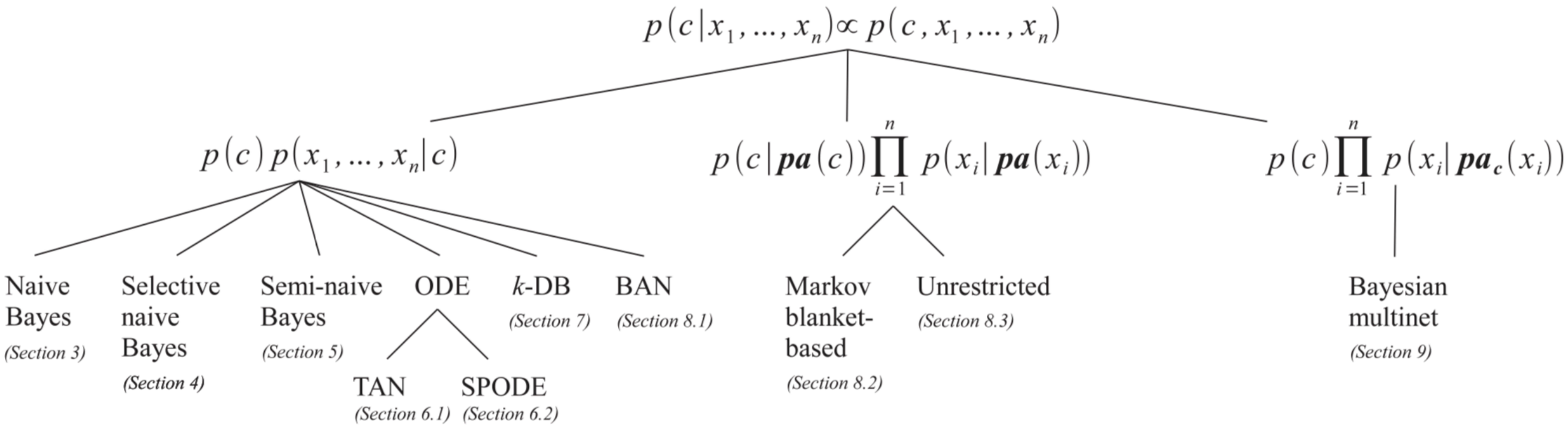
$$I(X;Y|Z) = \sum_{x \in X} \sum_{y \in Y} \sum_{z \in Z} P(x,y,z) \log \frac{P(x,y|z)}{P(x|z)P(y|z)}$$

TAN learning process

TAN procedure

1. Compute the conditional mutual information $I(X_i;X_j|C)$ *between each pair of attributes* $i \neq j$.
2. Build a complete undirected graph using the attributes as nodes and assign the weight of the edge that connects X_i to X_j by $I(X_i;X_j|C)$.
3. Apply the MWST algorithm.
4. Choose an attribute to be root and set the directions of all the edges to be outward from it.
5. Add a vertex node C and add an edge from C to every other attribute X_i.

Complete review of BN classifiers



Discrete Bayesian Network Classifiers: A Survey

Full Text: PDF Get this Article

Authors: Concha Bielza Universidad Politécnica de Madrid, Spain
Pedro Larrañaga Universidad Politécnica de Madrid, Spain

Published in:

· Journal
ACM Computing Surveys (CSUR) Surveys Homepage archive
Volume 47 Issue 1, July 2014
Article No. 5
ACM New York, NY, USA
table of contents doi>10.1145/2576868

2014 Article
Research
Refereed

Bibliometrics

· Citation Count: 17
· Downloads (cumulative): 1,721
· Downloads (12 Months): 243
· Downloads (6 Weeks): 21

Observation

Observation

- It is interesting to notice that while TAN was developed as a solution to the strong independence assumption in the naive Bayes classifier, in the tests presented in the TAN paper (Friedman et al. 1997), there are cases where the naive Bayes outperformed TAN.

Observation

- It is interesting to notice that while TAN was developed as a solution to the strong independence assumption in the naive Bayes classifier, in the tests presented in the TAN paper (Friedman et al. 1997), there are cases where the naive Bayes outperformed TAN.

- Can it be that given that TAN forces a tree structure amongst the attributes, there may be edges in the network which should not exist but are there in order to satisfy the tree structure?

Observation

- It is interesting to notice that while TAN was developed as a solution to the strong independence assumption in the naive Bayes classifier, in the tests presented in the TAN paper (Friedman et al. 1997), there are cases where the naive Bayes outperformed TAN.

- Can it be that given that TAN forces a tree structure amongst the attributes, there may be edges in the network which should not exist but are there in order to satisfy the tree structure?

- How can we relax the tree structure restriction for TAN?

A Bayes factor approach for learning TAN classifiers

Bayes factor for model selection

Given a dataset ***D***, the Bayes factor says that for a simple model B_s to be replaced by a more complex one B_c, the Bayes factor λ^* needs to satisfy the following:

$$\lambda^* = \frac{P(\boldsymbol{D}|B_\mathrm{s})P(B_\mathrm{s})}{P(\boldsymbol{D}|B_\mathrm{c})P(B_\mathrm{c})} < 1$$

$$\log_2 \lambda^* = \lambda = \log_2 P(\boldsymbol{D}|B_\mathrm{s}) - \log_2 P(\boldsymbol{D}|B_\mathrm{c}) + \log_2 P(B_\mathrm{s}) - \log_2 P(B_\mathrm{c})$$

with $\lambda<0$ —> replace B_s by B_c

Pham, D. T. and Ruz, G. A., *Proceedings of the Royal Society A*, (2009).

Computing the prior of a BN (1)

- The prior of a Bayesian network can be expressed using an encoding system of the network

- For a Bayesian network with n attribute nodes, let the representation of the network be constructed by using $n + 1$ symbols of length $m = \log_2(n + 1)$ bits.

E.g.

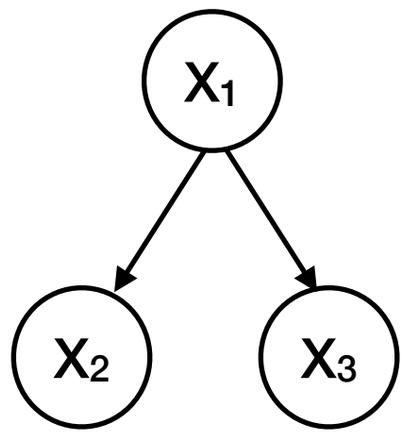
Each node can be represented by codes of length $m = \log_2(3 + 1) = 2$ bits

$C(X_1) = 00$, $C(X_2) = 01$, $C(X_3) = 10$, and Stop = 11

The encoding of the Bayesian network B:

$(X_1,X_2)(X_1,X_3)$ Stop = 0001001011

with code length $l(B) = 10$ bits



Computing the prior of a BN (2)

- It is known from coding theory that probabilities can be mapped to optimal codelengths, a uniquely decodeable code that minimizes the expected codelength.

- The expected length is minimized only if the codelengths $l(M)$ for a given model M are equal to the *Shannon information* contents, $l(M) = \log_2(1/\mathrm{P}(M))$.

- So, for a Bayesian network B with description length $l(B)$, the prior can be estimated by

$$P(B) = 2^{-l(B)}$$

Back to our Bayes factor

We consider that ***D*** is i.i.d.,

$$\lambda = l(B_c) - l(B_s) + \sum_{r=1}^{N} \sum_{i=1}^{n} \log_2 P(X_i = x_{i,r} | B_s) - \log_2 P(X_i = x_{i,r} | B_c)$$

Then, if we consider B_c to be the same as B_s, but with one additional edge.
This means that the description length of B_c uses two more symbols than B_s,

$$l(B_c) - l(B_s) = 2m$$

Thus, if the extra edge in B_c is due to attribute X_ω being the parent of attribute X_υ, we can express λ as

$$\lambda = 2m + \sum_{r=1}^{N} \log_2 P(X_\upsilon = x_{\upsilon,r} | Y = y_r) - \log_2 P(X_\upsilon = x_{\upsilon,r} | X_\omega = x_{\omega,r}, Y = y_r)$$

$$\Lambda_e = \sum_{i=1}^{e} \lambda_i$$

where λ_i represents the λ value for the *i*th edge being considered. Then, the adding of edges will continue while $\Lambda_e < 0$

BF TAN learning process

BF TAN procedure

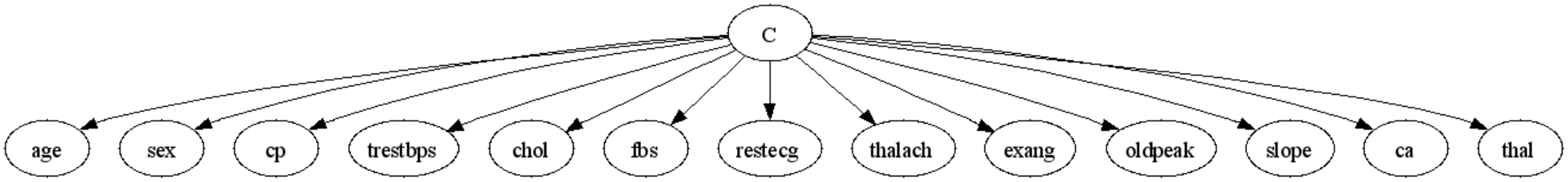
1. Compute the conditional mutual information $I(X_i;X_j|C)$ *between each pair of attributes* $i \neq j$.
2. Build a complete undirected graph using the attributes as nodes and assign the weight of the edge that connects X_i to X_j by $I(X_i;X_j|C)$.
3. Apply the MWST algorithm. For each iteration of the MWST algorithm:
   3.1 Compute the Bayesian measure λ from Eq.(4.8).
   3.2 Check that Λ satisfies condition in Eq.(4.9).
4. Choose an attribute to be root and set the directions of all the edges to be outward from it.
5. Add a vertex node C and add an edge from C to every other attribute X_i.

Example using UCI benchmark dataset (1)

- Data: Cleveland Heart Disease (UCI Repository)
- 13 attributes
- 2 clases
- 296 observations

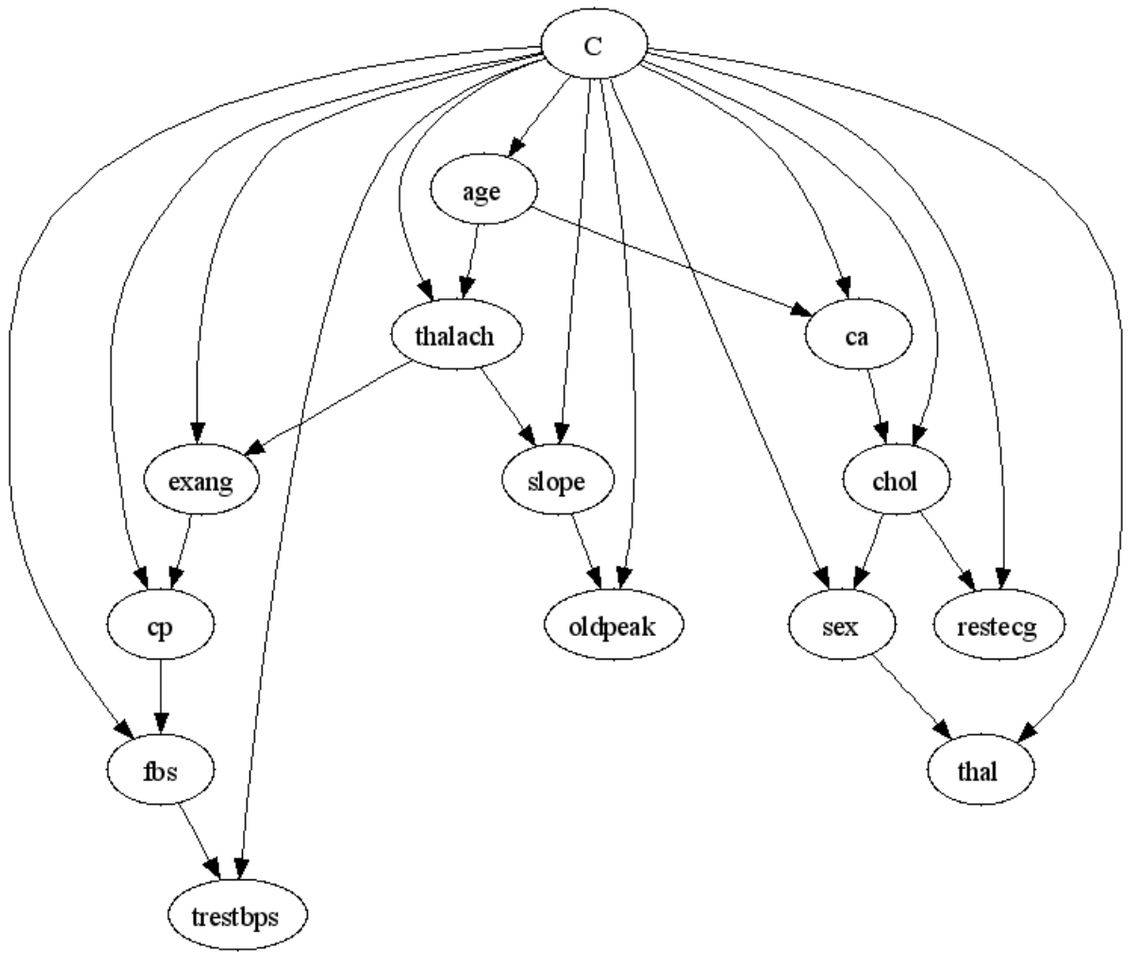
Example using UCI benchmark dataset (2)

- When 10% of the data is used to train the BF TAN model, the resulting structure is the naive Bayes



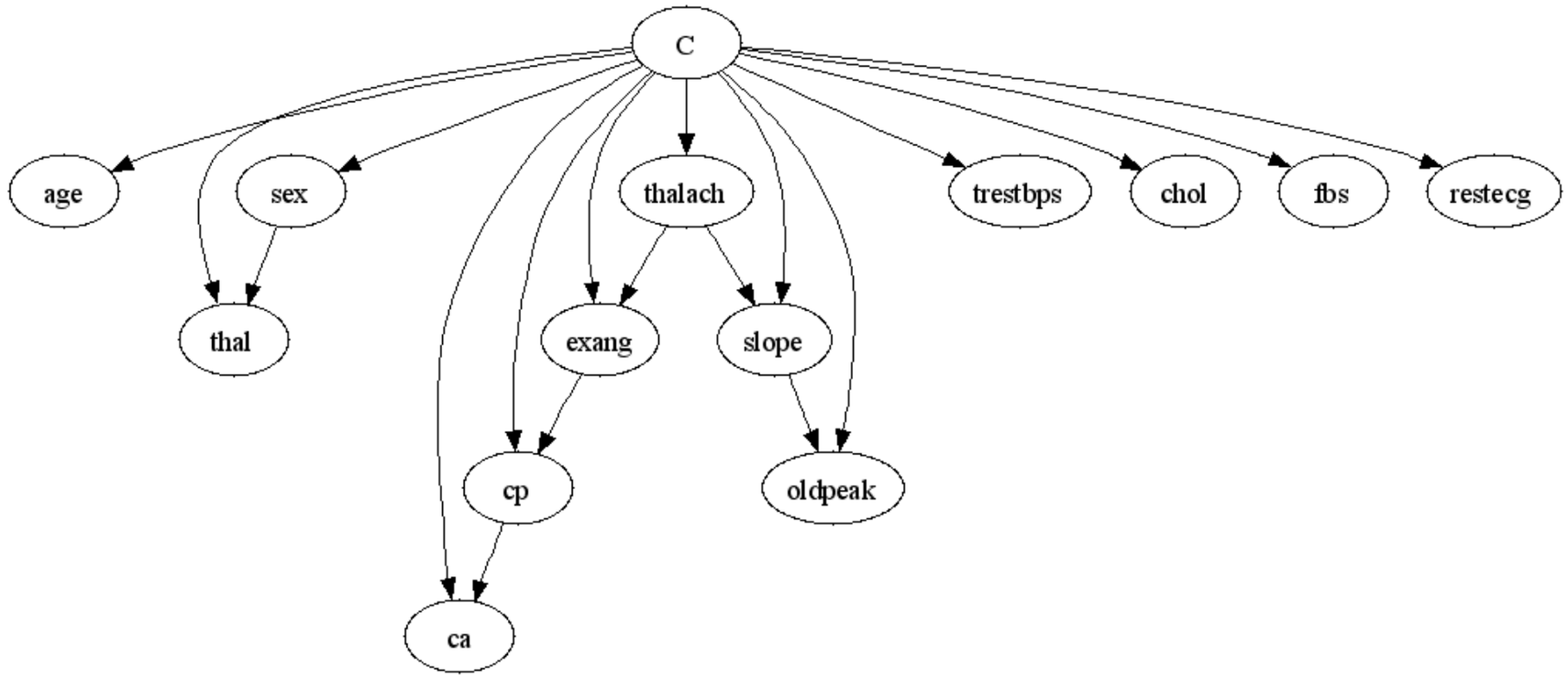
Example using UCI benchmark dataset (3)

- When 80% of the data is used to train the BF TAN model, the resulting structure is the TAN



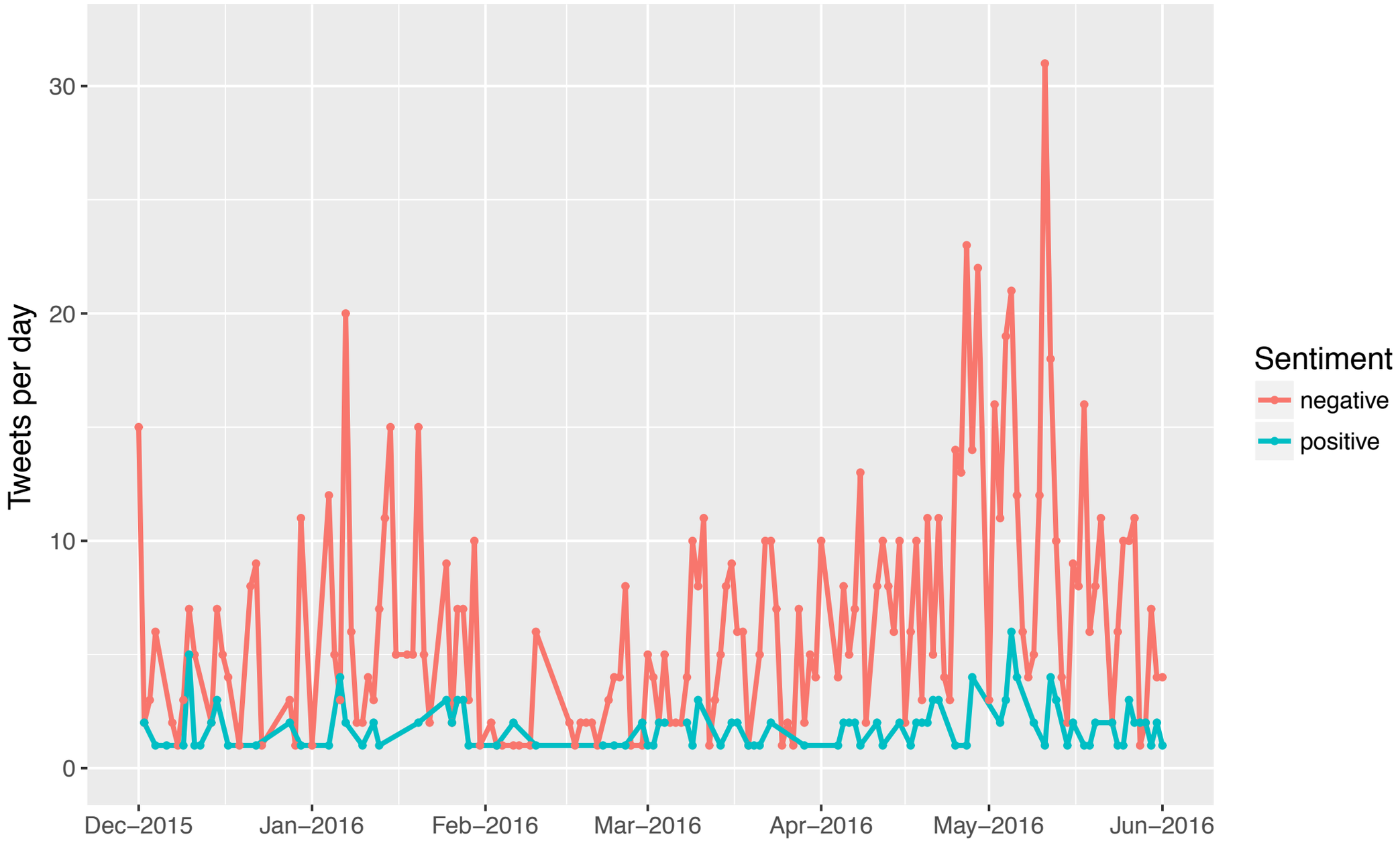
Example using UCI benchmark dataset (4)

- When 50% of the data is used to train the BF TAN model, the resulting structure is a forest with 6 edges



and if we test with the remaining 50%
NB=79.73%, TAN=81.76%, BF TAN = **83.12%**

Application: sentiment classification



Collaborators

Pablo Henríquez, PhD
Universidad Diego Portales, Chile

Aldo Mascareño, PhD
Centro de Estudios Públicos, Chile

Ruz, G.A., Henríquez, P.A., Mascareño, A., *Future Generation Computer Systems*, Vol. 106, 2020, 92-104.

Comparison of sentiment analysis approaches

Paper	Machine learning techniques	Languages	Reported accuracy (%)
Singh et al. [6]	NB, SVM	English	81.14
Zhu et al. [7]	SVM	Chinese	62.90
Tan and Zhang [8]	SVM, NB, k-NN	Chinese	82
Henríquez and Ruz [9]	RVFL	Spanish	82.90
Al-Ayyoub et al. [10]	SVM	Arabic	86.89
Ankit and Saleena [11]	NB, RF, SVM	English	75.81
Boiy and Moens [12]	SVM, NB	English	86.35
Ghorbel and Jacot [13]	SVM	French	93.25
Melville et al. [14]	NB	English	81.42
Wang et al. [15]	SVM	English	84.13
Gamon [16]	SVM	English	77.5
Pang and Lee [17]	SVM, regression	English	66.3
Pang et al. [4]	NB, SVM, maximum entropy	English	82.9
Prabowo and Thelwall [18]	SVM	English	87.30
Annett and Kondrak [19]	SVM, NB	English	77.5
Mullen and Collier [20]	Hybrid SVM	English	89

Data (1)

Data collection

- Dataset 1: contains a collection of 2187 tweets from the Chilean earthquake of 2010. This dataset was obtained from Cobo et al. 2015.

- Dataset 2: contains a collection of 60,000 tweets from the Catalan independence referendum of 2017. For this, we used the corresponding keywords for the event: #cataluña, #IndependenciaCatalunya, #2Oct, #CatalanReferedendum, #L6Nenlaencrucijada, and others.

Data (2)

Pre-processing

- Remove all URLs (e.g. www.xyz.com), hash tags (e.g. #topic), targets (@username)
- Remove all punctuations, symbols, numbers.
- Correct the spellings; sequence of repeated characters is to be handled.
- Remove Stop Words.
- Remove Non-Spanish Tweets

Feature representation

- The bag-of-words (BOW) technique is used to convert training tweets into a numeric representation resulting in a term document matrix (TDM).
- After learning the vocabulary, BOW describes the presence of known words within a tweet.
- This method creates an indicator vector signaling whether words in key-words-dictionary of a text are in the text.
- For example, consider the following two tweets:
  - tweet1: yesterday is past
  - tweet2: today is present.
- The vocabulary is {yesterday, is, past, today, present}. Now, the above tweets are represented as:
  - tweet1vector = [1, 1, 1, 0, 0]
  - tweet2vector = [0, 1, 0, 1, 1]

Sentiment analysis

$$\text{sentiment score} = \frac{positive - negative}{positive + negative + 2}$$

The sentiment score is in the range [-1,1]

Dictionary:
2246 positive terms
5247 negative terms

Datasets structures

Dataset	Positive	Negative	Total	Words
Dataset 1: Chilean earthquake	298	1889	2187	145
Dataset 2: Catalan independence referendum	10 816	49 184	60 000	207

Imbalanced class distribution-> SMOTE

Simulation setup

- We train the classifiers on the same training set and then compute, for each classifier, the confusion matrix (using the test set) where,
- True Positives (TP): The number of correctly classified positive tweets
- True Negatives (TN): The number of correctly classified negative tweets
- False Positives (FP): The number of incorrectly classified positive tweets
- False Negatives (FN): The number of incorrectly classified negative tweets

Then, the following performance measures are computed:

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN},$$

$$Precision = \frac{TP}{TP + FP},$$

$$Recall = \frac{TP}{TP + FN},$$

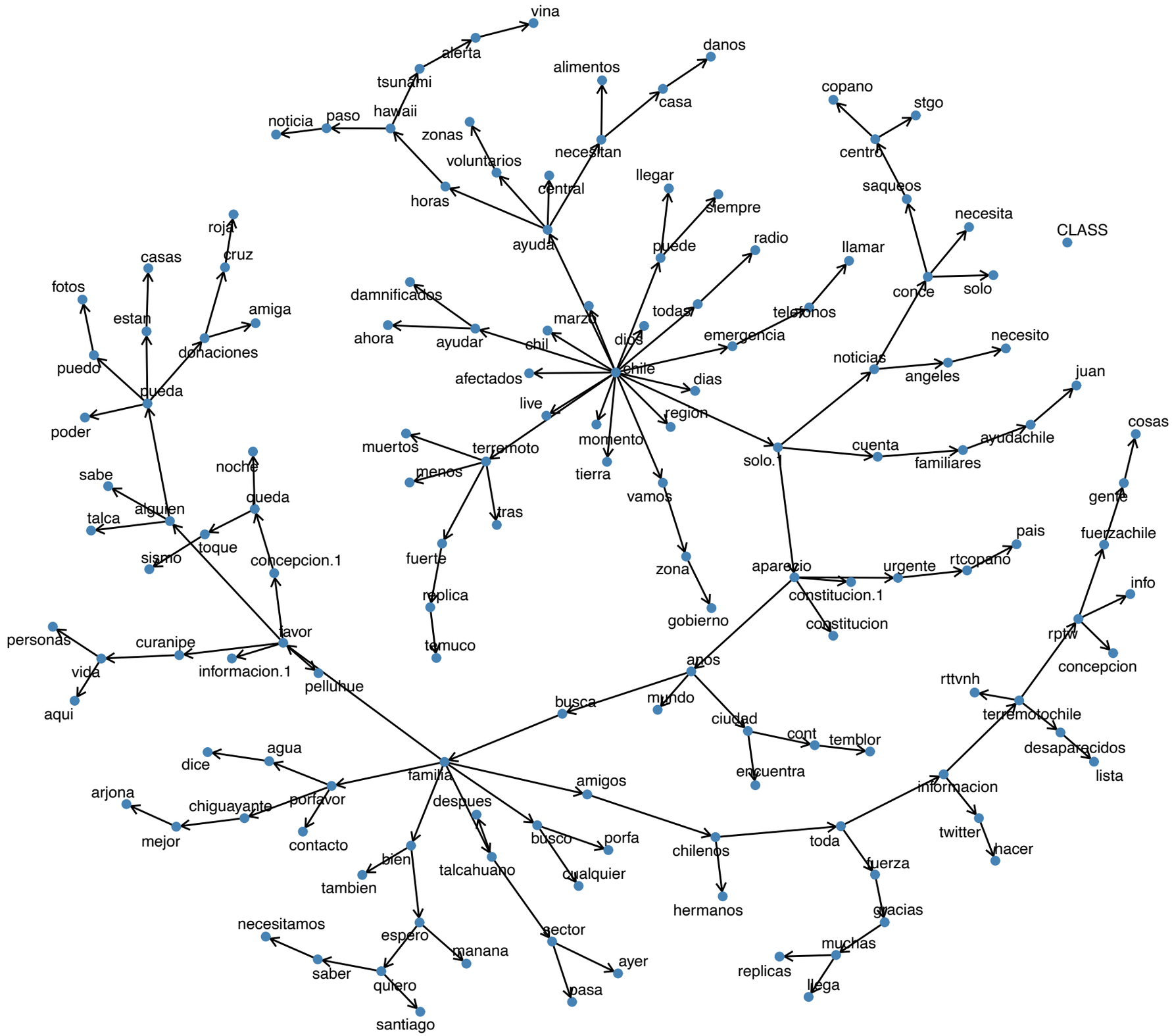
$$F_1 - score = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}.$$

For each dataset, we run 30 times the data splitting procedure, 70% for training and 30% for testing (the splitting was carried out randomly). For each run, we computed the classification performance measures on the test set, then the average and the standard deviation of each measure was reported.

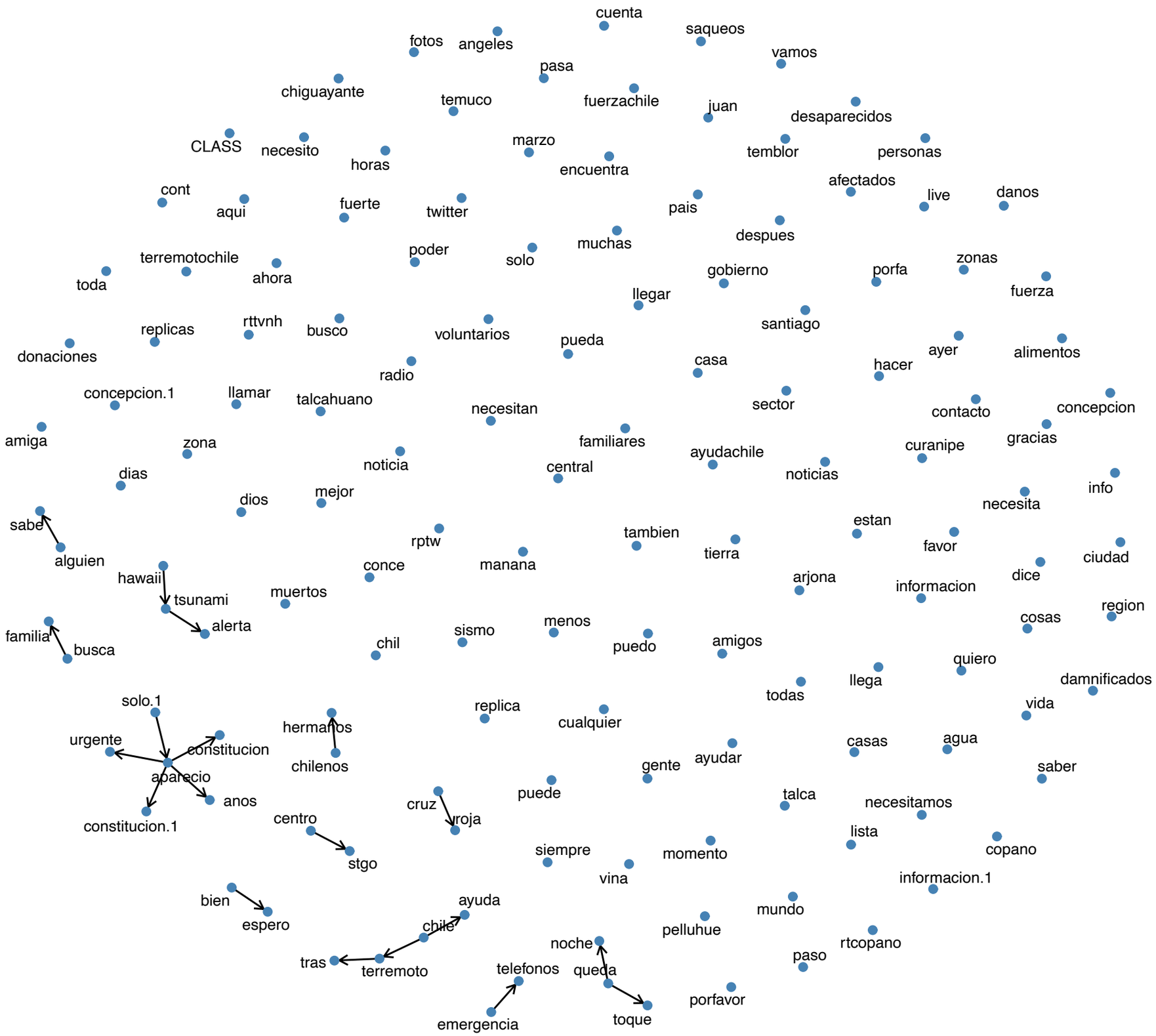
Results

Dataset 1 (Chilean earthquake)

Algorithm	Accuracy	Precision	Recall	F_1-score	N° Edges
NB	0.742 ± 0.027	0.895 ± 0.004	0.790 ± 0.009	0.841 ± 0.020	0
TAN	0.721 ± 0.029	0.896 ± 0.002	0.765 ± 0.032	0.825 ± 0.019	144
BF TAN	0.764 ± 0.007	0.898 ± 0.003	0.809 ± 0.034	0.849 ± 0.021	19
SVM	0.812 ± 0.067	0.867 ± 0.009	0.936 ± 0.081	0.899 ± 0.042	–
RF	0.725 ± 0.061	0.892 ± 0.011	0.776 ± 0.079	0.828 ± 0.054	–



TAN classifier for the Chilean earthquake dataset



Bayes factor TAN classifier for the Chilean earthquake dataset

Results

Dataset 2 (Catalan ind. ref.)

Algorithm	Accuracy	Precision	Recall	F_1-score	Nº Edges
NB	0.781 ± 0.013	0.885 ± 0.005	0.852 ± 0.004	0.868 ± 0.000	0
TAN	0.808 ± 0.004	0.906 ± 0.005	0.854 ± 0.009	0.879 ± 0.007	206
BF TAN	0.808 ± 0.004	0.906 ± 0.005	0.854 ± 0.009	0.879 ± 0.007	206
SVM	0.829 ± 0.005	0.841 ± 0.011	0.985 ± 0.008	0.907 ± 0.007	–
RF	0.858 ± 0.008	0.922 ± 0.002	0.895 ± 0.010	0.908 ± 0.005	–

TAN classifier for the Catalan independence referendum dataset

Summary 1

- We have presented a method that can automatically identify the edges of the TAN model that are supported by the training data.

- TAN and BF TAN offer interesting qualitative information to historically and socially comprehend the main features of the event dynamics, even if there are no sufficient training examples. Moreover, the resulting networks allow for the construction of a narrative or storytelling of the critical event been analyzed.

- Future research will consider other structures (e.g. more than one attribute parent)

An evolutionary computation approach for learning TAN classifiers

Motivation for this approach

- The TAN model corrects the strong assumption of conditional independence established by the naive version.

- In theory, it should produce better results (accuracy) than the naive Bayes classifier.

- However, TAN suffers from some problems, one in particular, the ability to make a correct estimate of the conditional mutual information.

- This is critical, since TAN obtains its tree structure using conditional mutual information as weights in the fully connected graph.

- A natural question arises, **can we learn the weights of the network from the data to obtain good classification results without having to estimate conditional mutual information?**

Proposed approach

- We propose a different approach for learning TAN classifiers without estimating conditional mutual information.

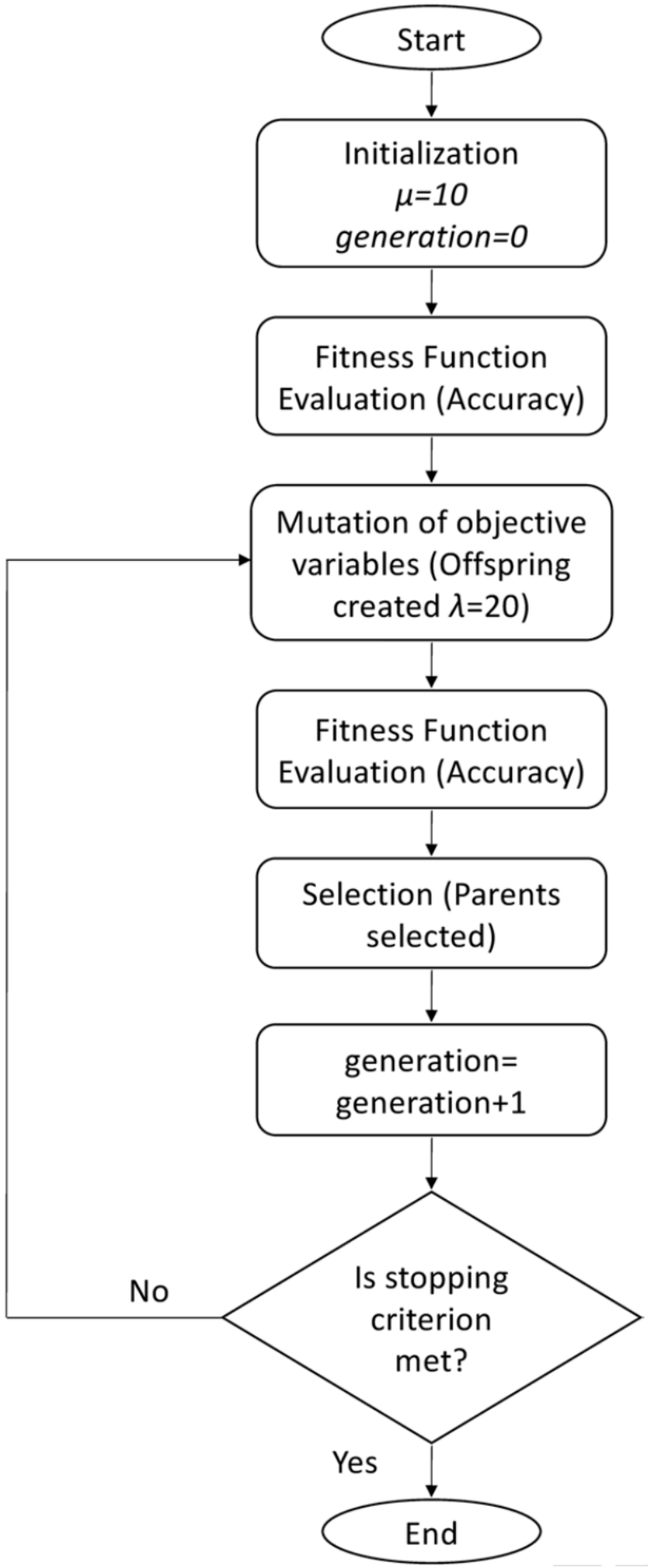
- Instead, we use an evolution strategy to learn the weights of the networks from the data.

TAN procedure

1. Compute the conditional mutual information $I(X_i;X_j|C)$ *between each pair of attributes* $i \neq j$.
2. Build a complete undirected graph using the attributes as nodes and assign the weight of the edge that connects X_i to X_j by $I(X_i;X_j|C)$.
3. Apply the MWST algorithm.
4. Choose an attribute to be root and set the directions of all the edges to be outward from it.
5. Add a vertex node C and add an edge from C to every other attribute X_i.

We will modify this part of the TAN procedure

An Evolution Strategy for Learning TAN Classifiers



We propose to use the deterministic survivor selection (μ, λ) method to obtain weights for the TAN model, which yields good classification results without estimating the conditional mutual information.

Since the TAN learning procedure starts with a fully connected weighted graph, for a network with n nodes we need to define m = n(n – 1)/2 weights.

An individual is a candidate solution coded as an m-dimensional vector containing the m weight values of a network.

Application: Facial biotype classification for orthodontic treatment planning

Application: Facial biotype classification for orthodontic treatment planning

Application: Facial biotype classification for orthodontic treatment planning

Application: Facial biotype classification for orthodontic treatment planning

Facial biotypes

(a) (b) (c)

FIGURE 1: Examples of the three facial biotypes. (a) Dolichofacial, (b) Brachyfacial, and (c) Mesofacial.

- (a) Dolichofacial: long and narrow face
- (b) Brachyfacial: short and wide face
- (c) Mesofacial: intermediate type between (a) and (b)

Vert Index

$$\text{Vert} = \frac{(FA - nv/3) + (FD - nv/3) + (nv - MP/4) + (nv - AIFH/4) + (MA - nv/4)}{5}$$

where:
nv = normal value for the age;
FA = facial axis;
FD = facial depth;
MP = mandibular plane;
AIFH = anteroinferior facial height;
MA = mandibular arch.

Vert < or = –2
Severe dolichofacial

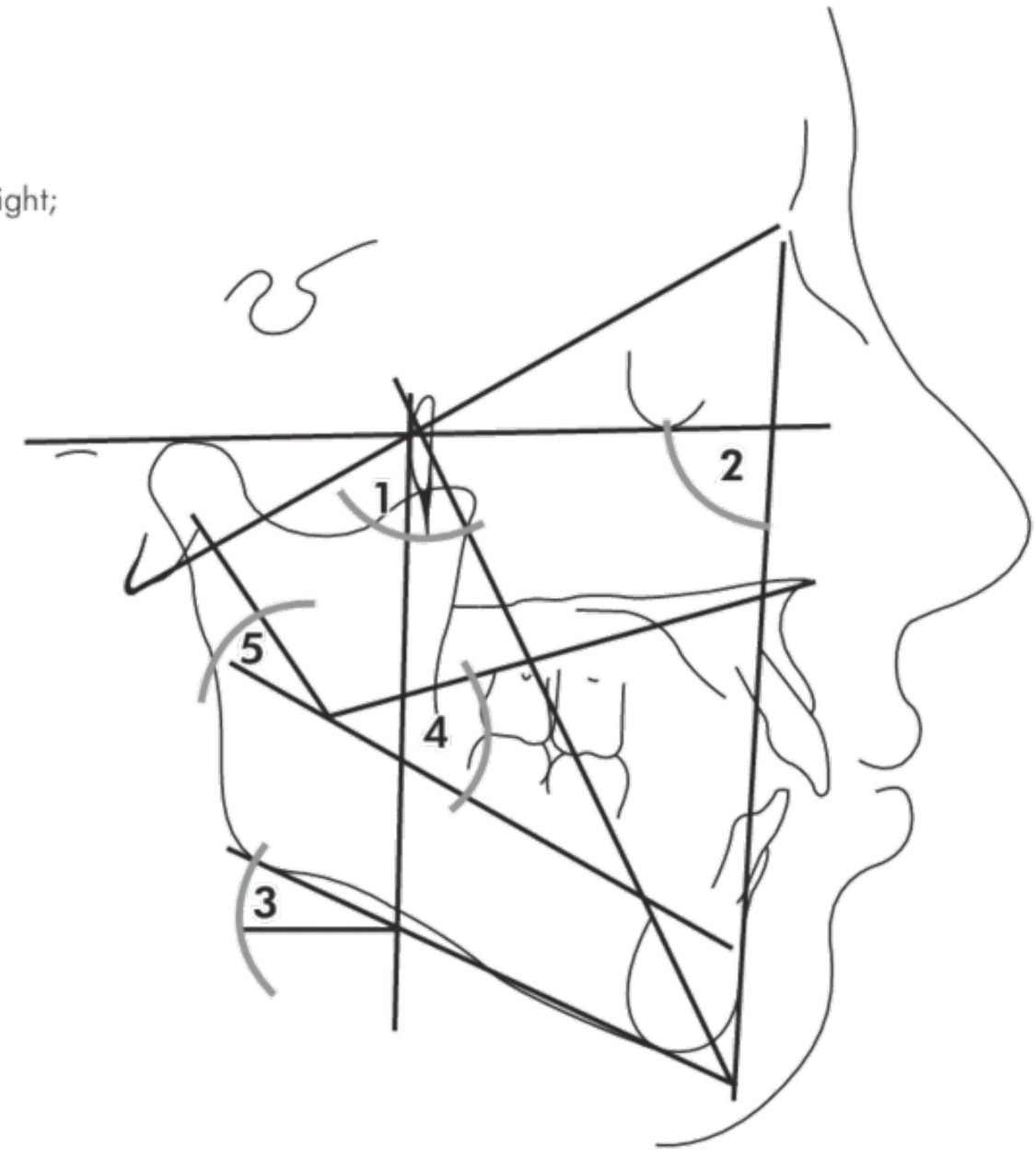
Vert = –1.9 to –1.0
Dolichofacial

Vert = –0.9 to –0.5
Light dolichofacial

Vert = –0.4 to +0.4
Mesofacial

Vert = +0.5 to +0.9
Brachyfacial

Vert > or = +1
Severe brachyfacial



Problem with Vert?

- It has been described that some attributes used in the VERT index can alter the index in patients in whom the sagittal relationship between the jaws is altered, leading to possible diagnostic errors.

- That is why, the possibility of automatically determining the facial biotype using attributes that are not altered by the sagittal position of the jaws would eliminate the errors observed with the use of the VERT index.

- Thus, in this work, we propose a machine learning approach to automatically classify a patient’s biotype using alternative attributes.

Collaborators

Pamela Araya-Díaz, DDS
Universidad Andrés Bello, Chile

Pablo Henríquez, PhD
Universidad Diego Portales, Chile

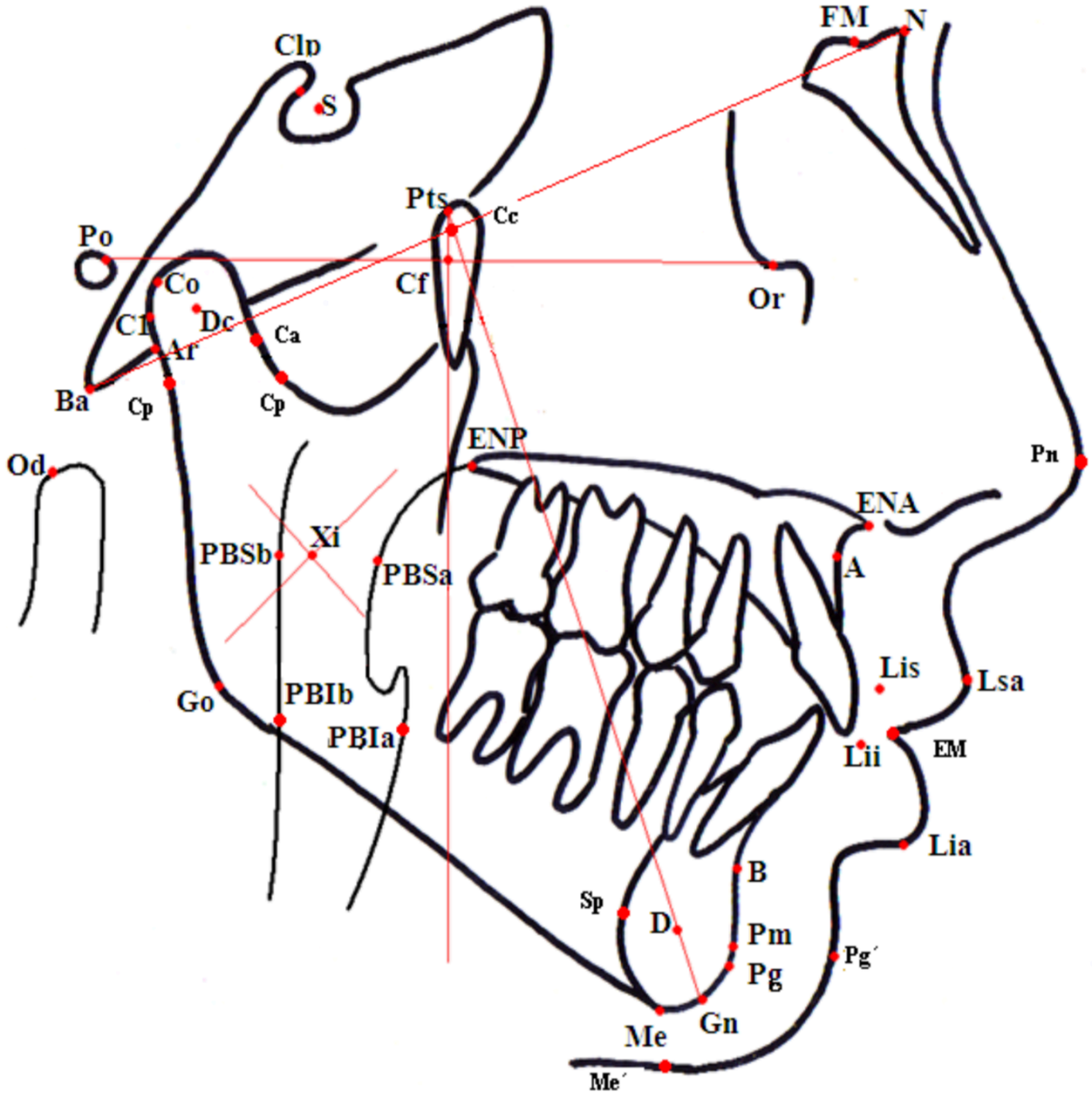
Ruz, G.A., Araya-Díaz, P., Henríquez, P.A.., *BMC Medical Informatics and Decision Making*, Vol. 22, 2022, 316.

The dataset

- The dataset consists of 182 lateral teleradiographies from Chilean patients.

- For each one, cephalometric analysis was performed to compute 31 continuous attributes that characterize the craniofacial morphology.

- Each lateral teleradiograph has been manually classified and validated by orthodontists into one of the three classes (Brachyfacial (70), Dolichofacial (45), and Mesofacial (67))



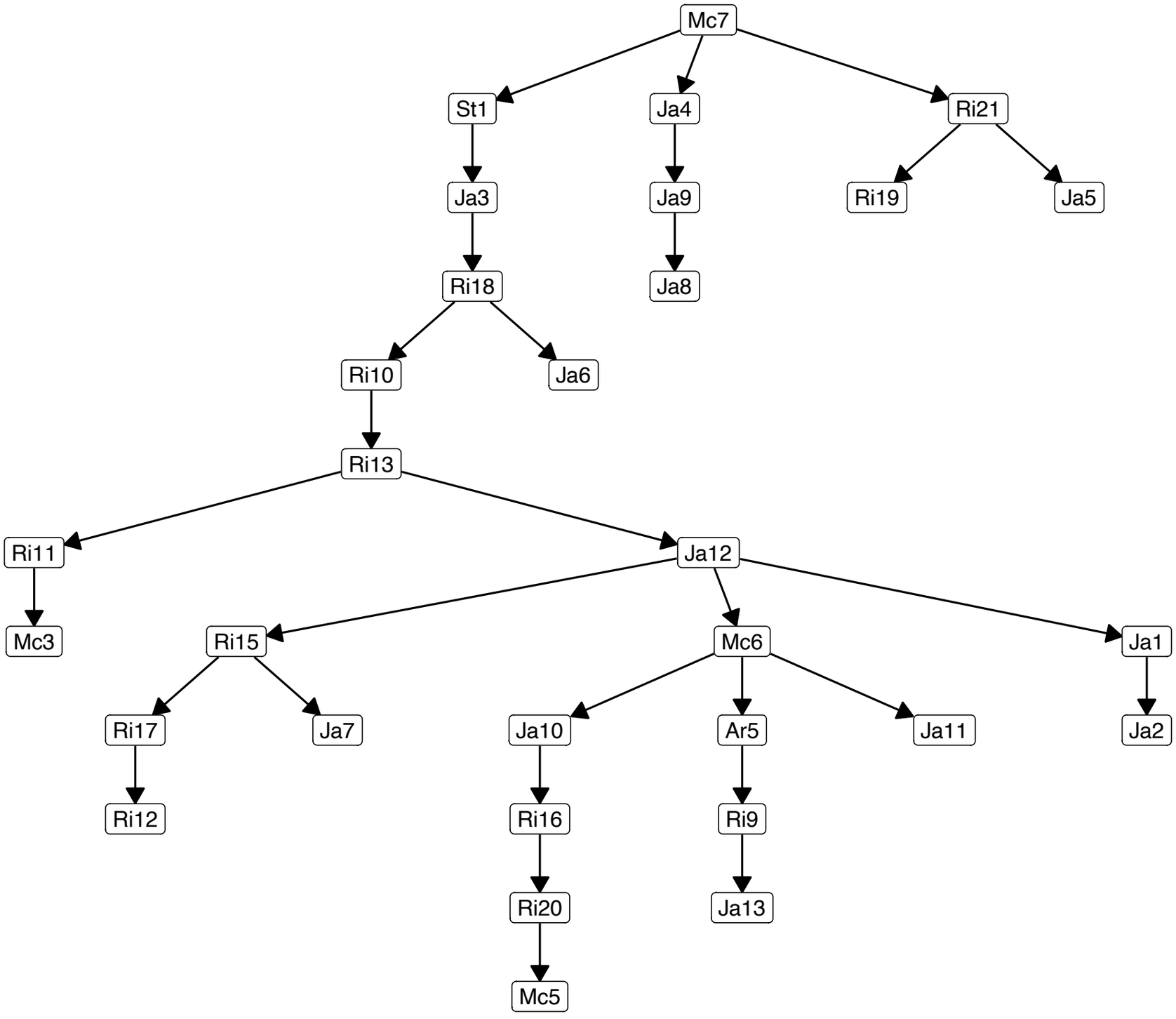
Results

Table 1: Performance measures for each model.

Algorithm	Accuracy Avg.±SD.	Precision Avg.±SD.	Recall Avg.±SD.	F_1-score Avg.±SD.	Kappa Avg.±SD.
NB	70.27±5.21	70.30±5.92	74.09±6.66	70.72±5.55	0.54±0.11
TAN	71.01±4.19	70.29±5.84	74.21±4.52	70.81±5.39	0.56±0.11
SVM	70.63±4.43	70.31±5.22	73.68±5.26	70.55±5.31	0.55±0.08
DT	69.27±7.19	69.93±5.02	73.16±4.14	70.72±4.82	0.52±0.10
RF	69.07±4.93	67.70±4.78	71.08±7.41	67.30±5.78	0.51±0.07
RVFL	70.11±5.34	70.44±4.31	74.16±4.32	71.19±4.35	0.54±0.11
ATAN	71.10±5.77	70.22±3.56	73.89±4.29	71.36±4.36	0.56±0.09
HC-TAN	70.41±7.44	69.67±6.39	73.95±6.04	70.22±6.26	0.58±0.11
HC-SP-TAN	70.81±6.48	71.63±4.81	74.98±5.98	71.98±5.36	0.56±0.11
BSEJ	71.09±4.24	72.09±5.95	74.28±5.22	71.09±5.11	0.55±0.12
FSSJ	71.69±3.92	72.03±3.34	73.88±5.02	72.27±4.56	0.58±0.09
(μ, λ)-TAN	74.09±3.62	73.89±2.54	76.88±2.34	75.14±3.24	0.59±0.08

Table 2: Statistical significance test for different simulations in terms of Accuracy. The ✓symbol denotes that these two methods are statistically significantly different with $p < 0.05$.

Algorithm	NB	TAN	SVM	DT	RF	RVFL	ATAN	HC-TAN	HC-SP-TAN	BSEJ	FSSJ
(μ, λ)-TAN	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓



The best (μ, λ)-TAN model obtained throughout the 20 runs. The (μ, λ)-TAN classifier for the facial biotype dataset.

Summary 2

- These results confirmed our view that the tree structures obtained using conditional mutual information do not necessarily yield the best classifiers.

- Moreover, based on the number of features and training samples, the performance of the TAN classifier can be quite affected.

- Another drawback when using conditional mutual information is that the resulting tree might not be reliable for interpretability purposes.

Thank you for your attention!

gonzalo.ruz@uai.cl

This presentation was supported by ANID PIA/BASAL FB0002 and ANID/PIA/ANILLOS ACT210096, and the Data Observatory Foundation.