

# Explorando Redes Bayesianas, ProbLog e ProbFoil

Cleiton Moya de Almeida

CPS840 – Tópicos Especiais em Inteligência Artificial  
Professor: Gerson Zaverucha

Rio de Janeiro, 23 de setembro de 2020

# Agenda

- **Redes Bayesianas**
  - ✓ Conceitos Básicos
  - ✓ Netica®
- **Rede *Chest Clinic***
  - ✓ Explorando o Modelo no Netica
  - ✓ Modelagem com ProbLog
  - ✓ Inferência com ProbLog
  - ✓ Aprendizado de Parâmetros com Netica
  - ✓ Aprendizado de Parâmetros com ProbLog
  - ✓ Aprendizado de Estrutura com Netica
- **ProbFoil**
  - ✓ Visão Geral
  - ✓ Exp. #1: Aprendizado Relacional
  - ✓ Exp. #2: Aprendizado Probabilístico
- **Conclusões**

# Redes Bayesianas

## ■ Conceitos Básicos

### • Rede Bayesiana:

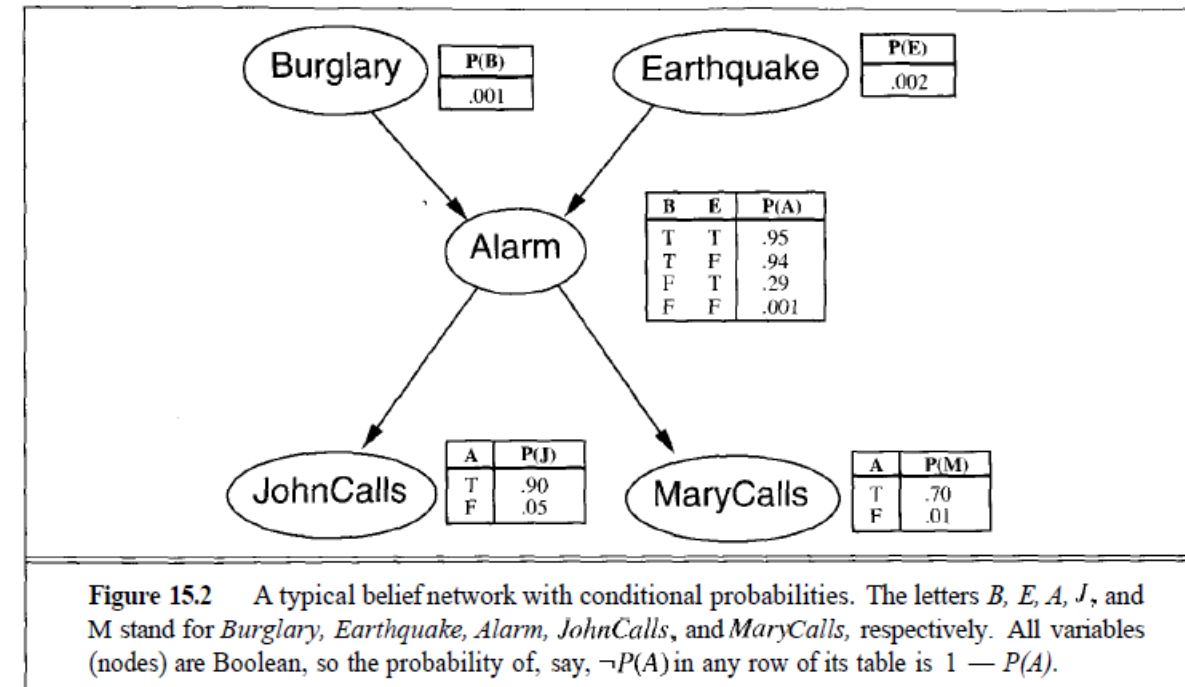
- ✓ Grafo direcional acíclico;
- ✓ Representação compacta da *Full Joint Probability Distribution* (independência condicional)
- ✓ Cada nodo possui uma tabela de probabilidade condicional;
- ✓ Modelagem de sistemas probabilísticos.

### • Vantagens:

- ✓ Representação visual;
- ✓ Espaço amostral reduzido.

### • Tarefas:

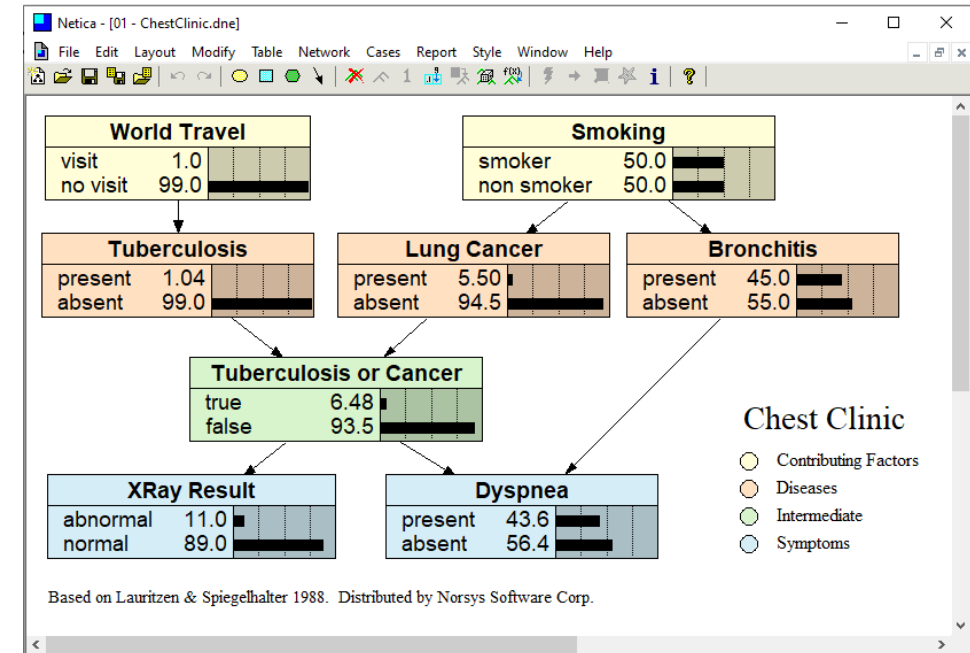
- ✓ Inferência Bayesiana (cálculo das *posteriors*);
- ✓ Aprendizado das tabelas à partir de exemplos (aprendizado de parâmetros);
- ✓ Aprendizado da estrutura.
- ✓ Suporte a decisões.



Fonte: Russel, Artificial Intelligence, p. 459.

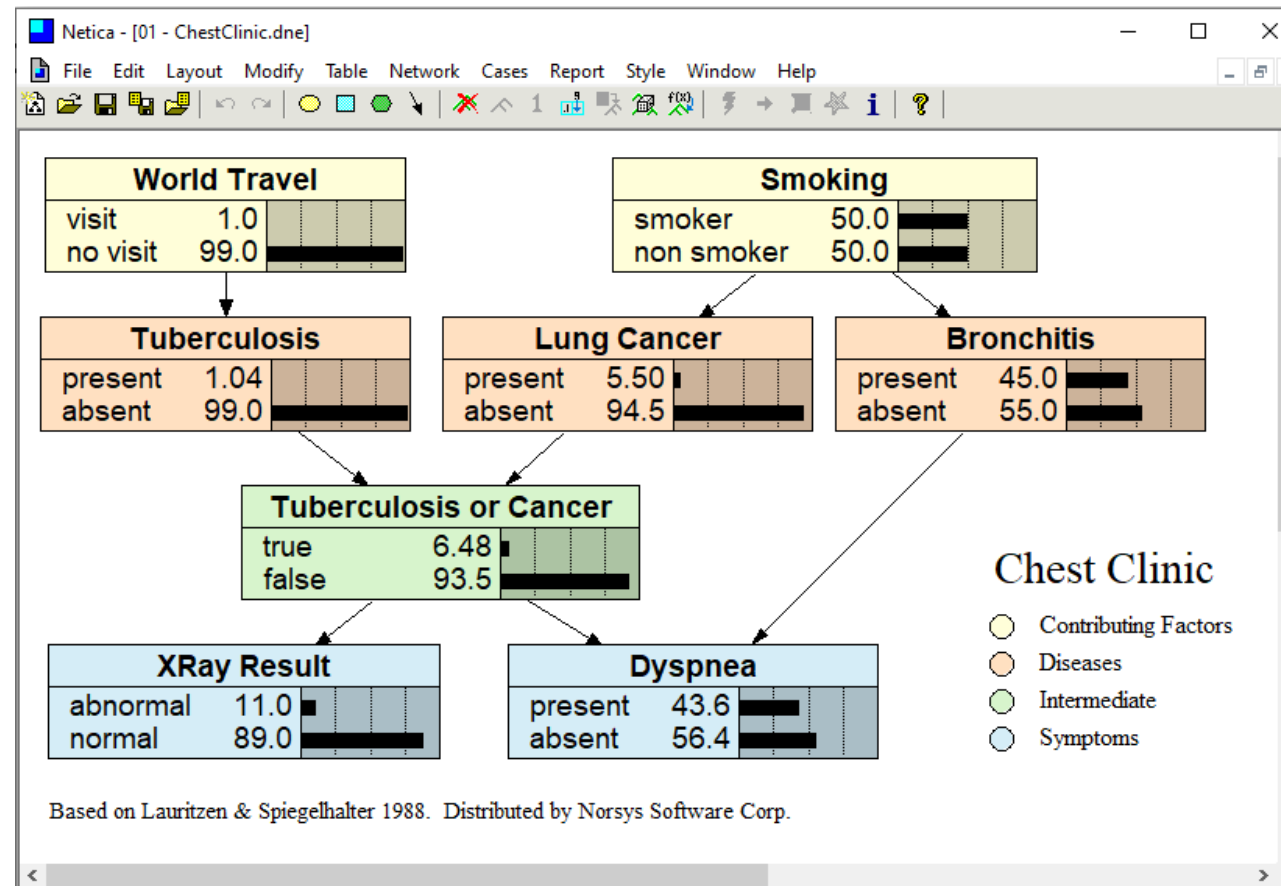
# Redes Bayesianas

- Netica<sup>©</sup>
  - Sistema desenvolvido pela **Norsys**<sup>1</sup> para o desenvolvimento de Redes Bayesianas;
  - Segundo a Norsys, é o mais utilizado no mundo;
  - Recursos:
    - ✓ Redes de Crenças Bayesianas;
    - ✓ Redes de Decisão;
    - ✓ Redes Bayesianas Dinâmicas;
    - ✓ Aprendizado de parâmetros;
    - ✓ Aprendizado de estrutura;
    - ✓ Simulação de casos;
    - ✓ Aplicação para Windows;
    - ✓ API para: C, C++, C#, Java, Visual Basic, Matlab, Clisp;
    - ✓ Versão gratuita com limitação no tamanho da rede;



# Rede *Chest Clinic*

- Explorando o Modelo no Netica



# Rede *Chest Clinic*

## ■ Modelagem com ProbLog

✓ Implementação no editor online: [Link do modelo salvo](#)

```
1 % The Chest Clinic Problem (Lauritzen & Spiegelhalter 1988)|
2 % Implemented by: Cleiton Almeida
3
4 % Contribution Facts
5 0.01::travel.
6 0.50::smoker.
7
8 % Diseasis
9 0.05::tuberculosis :- travel.
10 0.01::tuberculosis :- \+ travel.
11 0.10::cancer :- smoker.
12 0.01::cancer :- \+ smoker.
13 0.60::bronchitis :- smoker.
14 0.30::bronchitis :- \+ smoker.
15
16 % Intermediate
17 tb_or_ca :- tuberculosis.
18 tb_or_ca :- cancer.
19
20 % Symptoms
21 0.98::xray_abnormal :- tb_or_ca.
22 0.05::xray_abnormal :- \+ tb_or_ca.
23
24 0.90::dyspnea :- tb_or_ca, bronchitis.
25 0.70::dyspnea :- tb_or_ca, \+ bronchitis.
26 0.80::dyspnea :- \+ tb_or_ca, bronchitis.
27 0.10::dyspnea :- \+ tb_or_ca, \+ bronchitis.
```

# Rede *Chest Clinic*

## ■ Inferência com ProbLog

- ✓ Inferência no editor online:

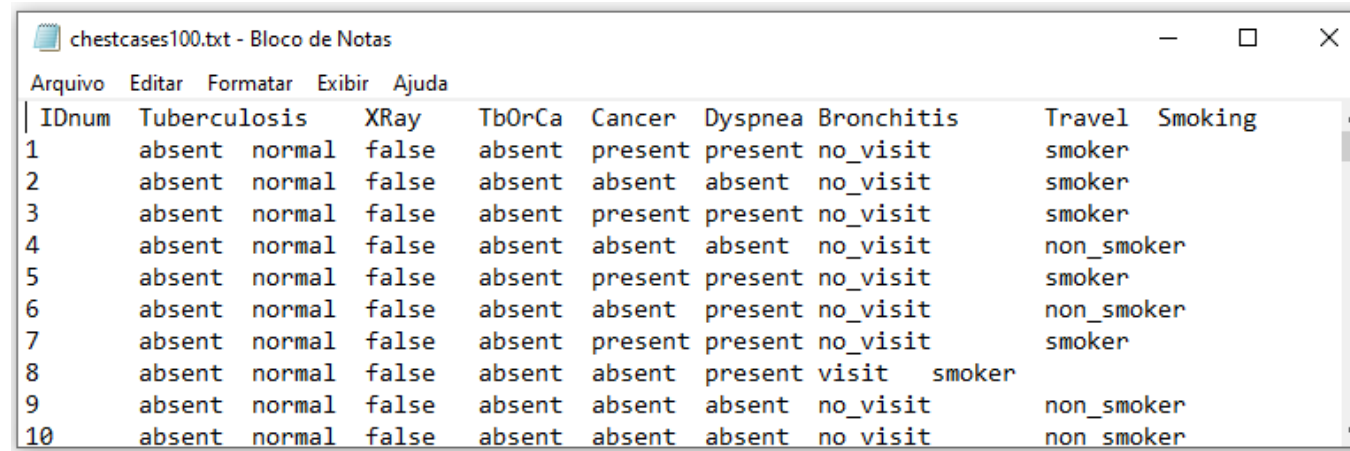
```

29 % Findings
30 evidence(dyspnea).
31 evidence(smoker).
32 evidence(\+ travel).
33
34 % Inferences
35 query(smoker).
36 query(travel).
37 query(tuberculosis).
38 query(cancer).
39 query(tb_or_ca).
40 query(bronchitis).
41 query(xray_abnormal).
42 query(dyspnea).
  
```

Query ▼	Probability
bronchitis	0.88029673
cancer	0.14836258
dyspnea	1
smoker	1
tb_or_ca	0.16171522
travel	0
tuberculosis	0.014836258
xray_abnormal	0.20039515

# Rede *Chest Clinic*

- Aprendizado de Parâmetros com Netica
  - ✓ Base de dados com **100 exemplares**;
  - ✓ Base gerada através do Netica à partir da tabela de probabilidades do modelo anterior;

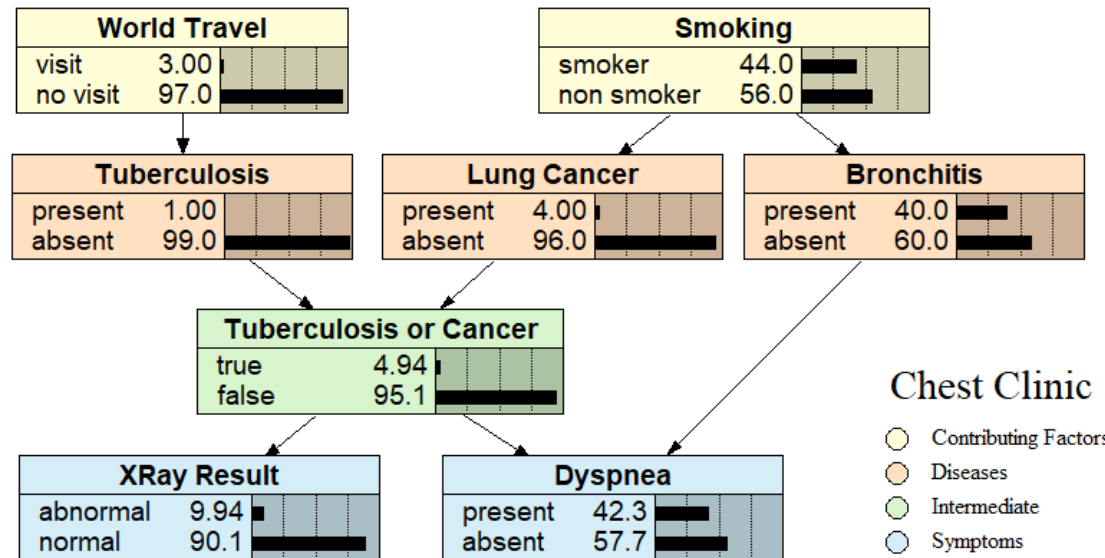


IDnum	Tuberculosis	XRay	TbOrCa	Cancer	Dyspnea	Bronchitis	Travel	Smoking
1	absent	normal	false	absent	present	present	no_visit	smoker
2	absent	normal	false	absent	absent	absent	no_visit	smoker
3	absent	normal	false	absent	present	present	no_visit	smoker
4	absent	normal	false	absent	absent	absent	no_visit	non_smoker
5	absent	normal	false	absent	present	present	no_visit	smoker
6	absent	normal	false	absent	absent	present	no_visit	non_smoker
7	absent	normal	false	absent	present	present	no_visit	smoker
8	absent	normal	false	absent	absent	present	visit	smoker
9	absent	normal	false	absent	absent	absent	no_visit	non_smoker
10	absent	normal	false	absent	absent	absent	no_visit	non_smoker



# Rede *Chest Clinic*

- Aprendizado de Parâmetros com Netica



Based on Lauritzen & Spiegelhalter 1988. Distributed by Norsys Software Corp.

- Netica informa o numero de iterações e o valor do *-Log Likelihood*:

```
Doing EM Learning, hold down <ctrl> + <mouse button> to stop ...
Iteration  -Log Likelihood  Change %
0          5.82917
1          2.23218      61.7068
2          2.23218      0.0000
```

# Rede *Chest Clinic*

- Aprendizado de Parâmetros com ProbLog
  - ✓ **Passo 1:** Identificação das probabilidades a serem aprendidas:

```
1 % Chest Clinic
2
3 % Contribution Facts
4 t(_)::travel.
5 t(_)::smoker.
6
7 % Diseasis
8 t(_)::tuberculosis :- travel.
9 t(_)::tuberculosis :- \+ travel.
10 t(_)::cancer :- smoker.
11 t(_)::cancer :- \+ smoker.
12 t(_)::bronchitis :- smoker.
13 t(_)::bronchitis :- \+ smoker.
14
15 % Intermediate
16 tb_or_ca :- tuberculosis.
17 tb_or_ca :- cancer.
18
19 % Symptoms
20 t(_)::xray_abnormal :- tb_or_ca.
21 t(_)::xray_abnormal :- \+ tb_or_ca.
22
23 t(_)::dyspnea :- tb_or_ca, bronchitis.
24 t(_)::dyspnea :- tb_or_ca, \+ bronchitis.
25 t(_)::dyspnea :- \+ tb_or_ca, bronchitis.
26 t(_)::dyspnea :- \+ tb_or_ca, \+ bronchitis.
```

# Rede *Chest Clinic*

## ■ Aprendizado de Parâmetros com ProbLog

✓ **Passo 2:** Conversão da base de dados para o formato ProbLog:

	A	B	C	D	E	F	G	H
1	tuberculosis	xray abnormal	tb or ca	cancer	dyspnea	bronchitis	travel	smoker
2	false	false	false	false	true	true	false	true
3	false	false	false	false	false	false	false	true
4	false	false	false	false	true	true	false	true
5	false	false	false	false	false	false	false	false
6	false	false	false	false	true	true	false	true
7	false	false	false	false	false	true	false	false
8	false	false	false	false	true	true	false	true
9	false	false	false	false	false	true	true	true
10	false	false	false	false	false	false	false	false

chest\_clinic\_100instances.pl - Bloco de Notas

Arquivo Editar Formatar Exibir Ajuda

```
evidence(tuberculosis,false).
evidence(xray_abnormal,false).
evidence(tb_or_ca,false).
evidence(cancer,false).
evidence(dyspnea,true).
evidence(bronchitis,true).
evidence(travel,false).
evidence(smoker,true).
```

```
---
evidence(tuberculosis,false).
evidence(xray_abnormal,false).
evidence(tb_or_ca,false).
evidence(cancer,false).
evidence(dyspnea,false).
evidence(bronchitis,false).
evidence(travel,false).
evidence(smoker,true).
---
```

# Rede *Chest Clinic*

## ■ Aprendizado de Parâmetros com ProbLog

✓ **Passo 3:** Utilização do ProbLog versão *standalone*<sup>1</sup>:

### ■ Instalação:

```
pip install problog
```

### ■ Execução do algoritmo:

Used as: `problog lfi <model> <evidence> [optional]` where:

- `<model>` is the ProbLog model file;
- `<evidence>` is the a file containing a set of examples to learn from.
- `[optional]` are optional arguments

```
(problog) cleiton@DEBIAN9-VM:~/Documentos/ProbLog$ problog lfi chest_clinic_learning.pl  
chest_clinic_100instances.pl -o chest_clinic_learned.pl
```

1- <https://problog.readthedocs.io/en/latest/cli.html>

# Rede *Chest Clinic*

## ■ Aprendizado de Parâmetros com ProbLog

### ✓ Resultados:

```

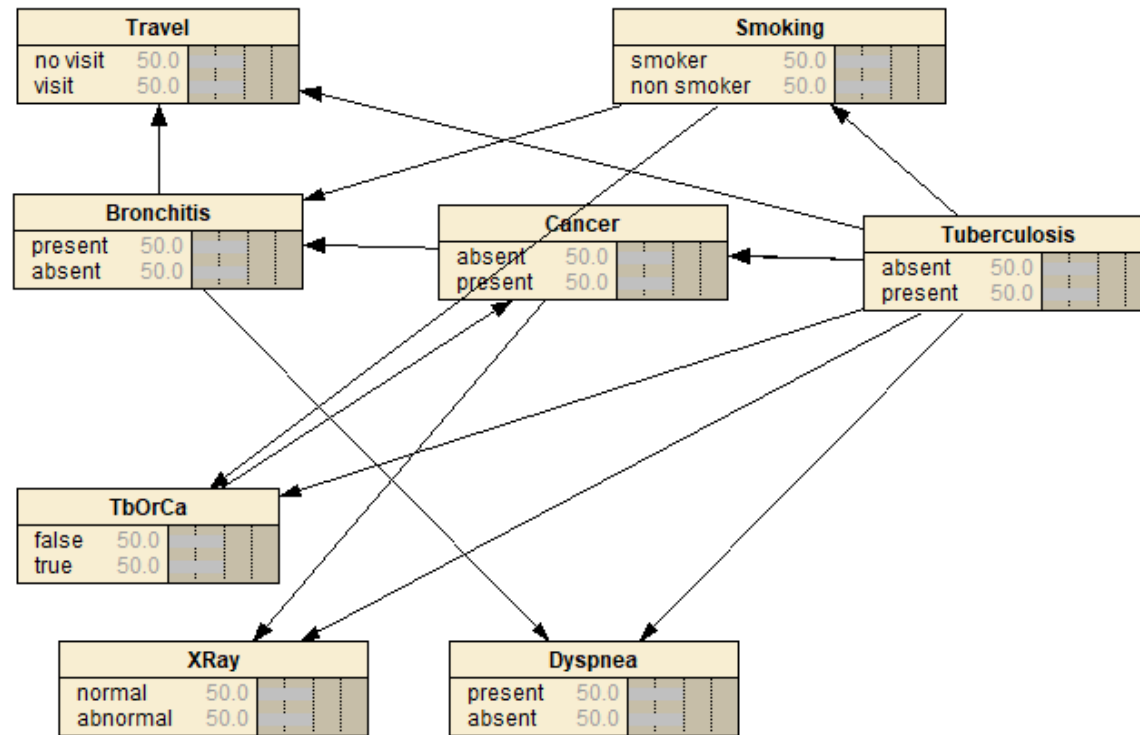
chest_clinic_100learned.txt - Bloco de Notas
Arquivo  Editar  Formatar  Exibir  Ajuda
-72.01054365429283 [0.03, 0.44, 0.0, 0.01030928, 0.09090909, 0.0, 0.56818182,
0.26785714, 1.0, 0.05263158, 0.75, 1.0, 0.83333333, 0.11864407] [t(_)::travel, t
(_)::smoker, t(_)::tuberculosis, t(_)::tuberculosis, t(_)::cancer, t(_)::cancer, t
(_)::bronchitis, t(_)::bronchitis, t(_)::xray_abnormal, t(_)::xray_abnormal, t
(_)::dyspnea, t(_)::dyspnea, t(_)::dyspnea, t(_)::dyspnea] 3
  
```

- ✓ Netica e ProbLog: mesmos resultados.
- ✓ Resultados satisfatórios;

Netica Messages			
Travel:			
visit	no visit		
0.0300001	0.97		
Smoking:			
smoker	non smoker		
0.44	0.56		
Tuberculosis:			
present	absent	Travel	
3.33331e-6	0.999997	visit	
0.0103094	0.989691	no visit	
Cancer:			
present	absent	Smoking	
0.0909093	0.909091	smoker	
1.78571e-7	1	non smoker	
Bronchitis:			
present	absent	Smoking	
0.568182	0.431818	smoker	
0.267857	0.732143	non smoker	
TbOrCa:			
true	false	Tuberculosis	Cancer
0.5	0.5	present	present
0.99999	9.9998e-6	present	absent
0.999997	2.49999e-6	absent	present
1.05263e-7	1	absent	absent
XRay:			
abnormal	normal	TbOrCa	
0.999998	1.99999e-6	true	
0.0526317	0.947368	false	
Dyspnea:			
present	absent	TbOrCa	Bronchitis
0.749999	0.250001	true	present
0.99999	9.9998e-6	true	absent
0.833333	0.166667	false	present
0.118644	0.881356	false	absent

# Rede *Chest Clinic*

## ■ Aprendizado de Estrutura com Netica



- ✓ Netica utiliza algoritmo TAN<sup>1</sup>;
- ✓ Tanto usando 100 exemplares quanto 1000, estrutura aprendida não foi satisfatória para este problema.

1 - Friedman, Nir, Dan Geiger, and Moises Goldszmidt (1997) "Bayesian network classifiers" in Machine Learning, Vol 29, 131-163.

# ProbFoil

## ■ Visão Geral

- Aprendizado de regras relacionais e probabilísticas à partir de exemplos;
- Construído à partir do ProbLog 2.1;
- Versão atual: ProbFoil 2.1<sup>1</sup>;
- Instalação: `pip install probfoil`.
- Referências:
  - ✓ L. De Raedt, A. Dries, I. Thon, G. Van den Broeck, M. Verbeke. Inducing Probabilistic Relational Rules from Probabilistic Examples (IJCAI, 2015)  
(<https://lirias.kuleuven.be/handle/123456789/499989>)
  - ✓ L. De Raedt, I. Thon. Probabilistic rule learning (Proceedings of ILP, 2010)  
(<https://lirias.kuleuven.be/handle/123456789/296011>)

1 - <https://bitbucket.org/problog/prob2foil/src/master/>

# ProbFoil

## ■ Visão Geral

### ■ Modos<sup>1</sup>:

The modes should be specified by adding facts of the form `mode(predicate(mode1, mode2, ...))`, where `modeX` is the mode specifier for argument X. Possible mode specifiers are:

- `+`: the variable at this position must already exist when the literal is added
- `-`: the variable at this position does not exist yet in the rule (note that this is stricter than usual)
- `c`: a constant should be introduced here; possible value are derived automatically from the data

### ■ Tipos:

For each relevant predicate (target and modes) there should be a type specifier. This specifier is of the form `base(predicate(type1, type2, ...))`, where `typeX` is a type identifier. Type can be identified by arbitrary Prolog atoms (e.g. `person`, `a`, etc.)

### ■ Target:

The target should be specified by adding a fact `learn(predicate/arity)`.

1 - <https://pypi.org/project/probfoil/>



# ProbFoil

## ■ Exp. #1: Aprendizado Relacional

### ■ Predicados e Target:

```
% Modes
mode(male(+)).
mode(parent(+,+)).
mode(parent(+,-)).
mode(parent(-,+)).

% Type definitions
base(parent(person,person)).
base(male(person)).
base(female(person)).
base(mother(person,person)).
base(grandmother(person,person)).
base(father(person,person)).
base(male_ancestor(person,person)).
base(female_ancestor(person,person)).

% Target
learn(grandmother/2).

% How to generate negative examples
example_mode(auto).
```

### ■ Dados:

```
father(bart, stijn).
father(bart, pieter).
father(luc, soetkin).

mother(katleen, stijn).
mother(katleen, pieter).
mother(lieve, soetkin).

parent(bart,stijn).
parent(bart,pieter).
parent(luc,soetkin).

female(alice).
female(an).
female(esther).

male(bart).
male(etienne).
male(leon).

grandmother(esther,soetkin).
grandmother(esther,stijn).
grandmother(esther,pieter).
```

### ■ Resultado:

```
===== SETTINGS =====
files:      ['family_settings.pl', 'family_data.pl']
probfoill:  False
beam_size:  5
           p:  None
           l:  None
verbose:    None
symmetry_breaking: False
target:     None
seed:      None
log:       None
===== FINAL THEORY =====
grandmother(A,B) :- parent(C,B), \+male(A), \+parent(C,A), \+parent(A,B)
===== SCORES =====
Accuracy: 0.875
Precision: 0.14285714285714285
Recall: 1.0
===== STATISTICS =====
Rule evaluations: 316
Total time: 9.5428s
```

# ProbFoil

## ■ Exp. #1: Aprendizado Relacional

- ✓ Aumentando a base de treinamento com 129 exemplares:

```
===== SETTINGS =====
      files:  ['family_settings.pl', 'family_data2.pl']
    probfoill: False
    beam_size: 5
         p:  None
         l:  None
    verbose:  None
symmetry_breaking: False
        target: None
        seed:  None
         log:  log.txt
===== FINAL THEORY =====
grandmother(A,B) :- parent(C,B), parent(A,C), \+male(A)
===== SCORES =====
    Accuracy: 1.0
    Precision: 1.0
     Recall:  1.0
===== STATISTICS =====
Rule evaluations: 280
    Total time:  38.6114s
```

# ProbFoil

## ■ Exp. #2: Aprendizado Probabilístico

- ✓ Inserindo probabilidade em father e parente:

33	<code>0.7::father(bart, stijn).</code>	57	<code>0.7::parent(bart, stijn).</code>
34	<code>0.9::father(bart, pieter).</code>	58	<code>0.9::parent(bart, pieter).</code>
35	<code>0.6::father(luc, soetkin).</code>	59	<code>0.6::parent(luc, soetkin).</code>
36	<code>0.7::father(willem, lieve).</code>	60	<code>0.7::parent(willem, lieve).</code>
37	<code>0.8::father(willem, katleen).</code>	61	<code>0.8::parent(willem, katleen).</code>
38	<code>0.9::father(rene, willem).</code>	62	<code>0.9::parent(rene, willem).</code>
39	<code>0.7::father(rene, lucy).</code>	63	<code>0.7::parent(rene, lucy).</code>
40	<code>0.8::father(leon, rose).</code>	64	<code>0.8::parent(leon, rose).</code>
41	<code>0.8::father(etienne, luc).</code>	65	<code>0.8::parent(etienne, luc).</code>
42	<code>0.9::father(etienne, an).</code>	66	<code>0.9::parent(etienne, an).</code>
43	<code>0.7::father(prudent, esther).</code>	67	<code>0.7::parent(prudent, esther).</code>

# ProbFoil

## ■ Exp. #2: Aprendizado Probabilístico

- ✓ Utilizando os mesmos modos do exemplo anterior:

```
cleiton@DEBIAN9-VM: ~/Documentos/ProbLog
Arquivo Editar Ver Pesquisar Terminal Ajuda
File "/home/cleiton/anaconda3/envs/problog/lib/python3.8/site-packages/problog/util.py", line 191, in subprocess_check_call
    raise subprocess.CalledProcessError(retcode, cmd)
subprocess.CalledProcessError: Command '['dsharp', '-Fnnf', '/tmp/tmpo3ubwaqg.nnf', '-smoothNNF', '-disableAllLits', '/tmp/tmpyxqr57z1.cnf']' died with <Signals.SIGABRT: 6>.

During handling of the above exception, another exception occurred:

Traceback (most recent call last):
  File "/home/cleiton/anaconda3/envs/problog/bin/probfoil", line 8, in <module>
    sys.exit(main())
  File "/home/cleiton/anaconda3/envs/problog/lib/python3.8/site-packages/problog/probfoil.py", line 452, in main
    hypothesis = learn.learn()
  File "/home/cleiton/anaconda3/envs/problog/lib/python3.8/site-packages/problog/probfoil.py", line 169, in learn
    next_hypothesis = self.best_rule(hypothesis)
  File "/home/cleiton/anaconda3/envs/problog/lib/python3.8/site-packages/problog/probfoil.py", line 97, in best_rule
    rule.scores = self._compute_scores_predict(rule)
  File "/home/cleiton/anaconda3/envs/problog/lib/python3.8/site-packages/problog/probfoil/learn.py", line 109, in _compute_scores_predict
    return self._compute_scores_predict_ground(rule)
  File "/home/cleiton/anaconda3/envs/problog/lib/python3.8/site-packages/problog/probfoil/learn.py", line 132, in _compute_scores_predict_ground
    result = self._data.evaluate(rule, functor=functor, arguments=examples)
  File "/home/cleiton/anaconda3/envs/problog/lib/python3.8/site-packages/problog/data.py", line 86, in evaluate
    knowledge = get_evaluable().create_from(ground_program)
  File "/home/cleiton/anaconda3/envs/problog/lib/python3.8/site-packages/problog/core.py", line 158, in create_from
    return ProbLog.convert(obj, cls, **kwargs)
  File "/home/cleiton/anaconda3/envs/problog/lib/python3.8/site-packages/problog/core.py", line 123, in convert
    next_obj = path[0](current_obj, path[1](**kwargs), **kwargs)
  File "/home/cleiton/anaconda3/envs/problog/lib/python3.8/site-packages/problog/ddnnf_formula.py", line 301, in _compile_with_dsharp
    raise DSharpError()
problog.ddnnf_formula.DSharpError: DSharp has encountered an error.
(problog) cleiton@DEBIAN9-VM:~/Documentos/ProbLog$
```

# ProbFoil

## ■ Exp. #2: Aprendizado Probabilístico

- ✓ Utilizando modos *female(+)*, *parent(+,+)*:

```
===== SETTINGS =====
      files:  ['family_settings3.pl', 'family_data3.pl']
      probfoill:  False
      beam_size:  5
      p:  None
      l:  None
      verbose:  None
      symmetry_breaking:  False
      target:  None
      seed:  None
      log:  None
===== FINAL THEORY =====
grandmother(A,B) :- female(A), female(B), \+parent(B,A), \+parent(A,B)
===== SCORES =====
      Accuracy:  0.7728531855955678
      Precision:  0.08139534883720931
      Recall:  0.7
===== STATISTICS =====
      Rule evaluations:  254
      Total time:  38.5498s
```

# Conclusões

- **Redes Bayesianas:**
  - ✓ Modelo bastante interessantes para a construção de sistemas de apoio à decisão;
- **Netica:**
  - ✓ Interface simples;
  - ✓ Diversas funcionalidades;
  - ✓ Não tem API para *Python* (mas tem C, C++, C#, Java...)
- **ProbLog:**
  - ✓ Sintaxe fácil;
  - ✓ Modelagem trabalhosa para redes maiores;
  - ✓ Versão *online*, *standalone* e API Python;
  - ✓ Inferência e aprendizagem de parâmetros.
- **ProbFoil:**
  - ✓ Confusão de versões no BitBucket<sup>1</sup> PyPI (YAP, Master, ProbLog, Paper..);
  - ✓ Página do ProbLog disponibiliza apenas exemplo relacional<sup>2</sup>;
  - ✓ Difícil depuração / documentação escassa;
  - ✓ Infelizmente não conseguimos obter sucesso com aprendizado probabilístico.

1 - <https://bitbucket.org/problog/prob2foil/src/master/>

2 - [https://dtai.cs.kuleuven.be/problog/tutorial/learning/05\\_structure\\_learning.html](https://dtai.cs.kuleuven.be/problog/tutorial/learning/05_structure_learning.html)

Obrigado  
Dúvidas?