Markov Logic Networks

Course project presentation: Sushma Akoju

Professor: Prof. James C. Bird

Contents

Learning Goal

Markov Logic Network

Learning from one example

Learning Goal and About the Project

Goal: To learn about Markov Logic Network

- Given a set of facts, relations and set of statements which conveys representation and reasoning about AI task
- find out how probabilistic w.r.t Ground atoms
- Write down a network of facts, relations and statements as an undirected graph
- Implement simple MLN
- Use Probabilistic framework pracmln with query-based inference

Problem definition, Task and example

Markov Logic Network

Learning from one example

Example description for this project

- Smoking causes cancer
- We need to stop people from smoking
- It's hard to do that since people are influenced by friends
- If friends keep smoking, they are likely to continue smoking

Peer influence doubles smoking risk for adolescents

Teens from collectivistic cultures also more swayed by peers than those in individualistic cultures

Date: August 21, 2017

Source: University of Pennsylvania

Summary: Having friends who smoke doubles the risk that youth ages 10 to 19 will pick up the habit, finds new meta-analysis of 75 longitudinal teen smoking studies. This influence is more powerful in collectivistic cultures than in individualistic ones.

https://www.sciencedaily.com/releases/2017/08/170821102718.htm

Example

- Smoking causes cancer
- Friends have similar habits

1. Define two predicates: Smokes(x), Cancer(x) and Friends(x,y)
2. Domain X: {people} , Y: {Friends for all x,y}
3. ∀x, smokes(x) => cancer(x)
4. smokes(a) = 1 and smokes(b) =1, cancer(a) = 1, cancer(b) =1, friends(a,b) = 1, friends(b,a) = 1
5. smokes(a) = 1 and smokes(b) =0, cancer(a) = 0, cancer(b) =0, friends(a,b) = 1, friends(b,a) = 1
6. Find most likely group of friends who smoke

First Order Logic

Markov Logic Network

Learning from one example

Example description for this project: contd.

$$\underbrace{i^2 + 3k}_{subject} \underbrace{\geq 10 + j}_{predicate}$$

Question:

$P(x): x + y \geq 6$

Possible Solutions:

Let $P(7,1)$ $\quad P(7,1): \quad (7)+(1) \geq 6$ $\quad$ True propositional statement

$8 \geq 6$

Let $P(3,2)$ $\quad P(3,2): \quad (3)+(2) \geq 6$ $\quad$ False propositional statement

$5 \ngeq 6$

First Order Logic

- Smoking causes cancer
- Friends have similar habits
- We use verbs : Smokes, hasCancer, Friends

Initial Weights	First Order Logic
1.5	**∀x, smokes(x) => cancer(x)**
1.1	**∀x,y friends(x,y) => (smokes(x) ⇔ smokes(y))**

Exceptions:

Not everyone who smokes, gets cancer.

Not all friends smoke

First Order Logic

Simple example:

There are two people in this world: **Alice (A) and Bob (B)**

Smokes(A), Smokes(B), Cancer(A), Cancer(B)

Friends(A,B) Friends(B,A) Friends(A,A) Friends(B,B)

Markov Property and Markov Random Field

Markov Logic Network

Learning from one example

Markov property and Markov Random Field

Probabilistic Graphical Models: Joint probability distributions and independence/dependence relations over a set of Random Variables.

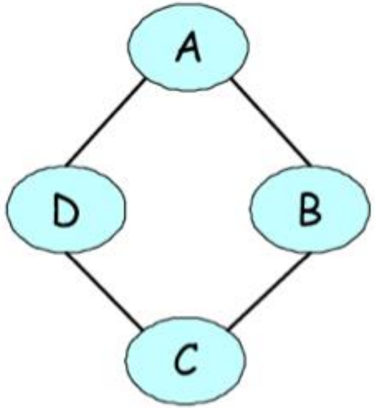
Bayesian networks: Directed Graphs

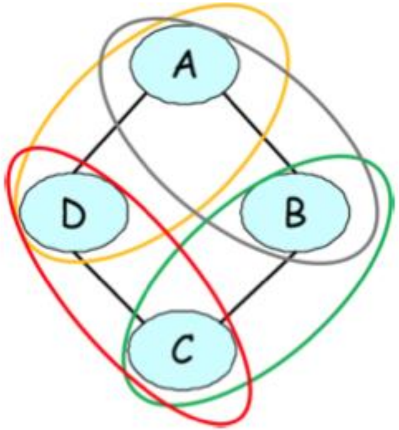
Markov Random Fields: Undirected Graphs

No edges indicate conditional Independence.

MRF: Friends and their similar Voting preferences

Goal: Learn Joint voting decision





$$\phi(X,Y) = \begin{cases} 10 & \text{if } X = Y = 1 \\ 5 & \text{if } X = Y = 0 \\ 1 & \text{otherwise.} \end{cases}$$

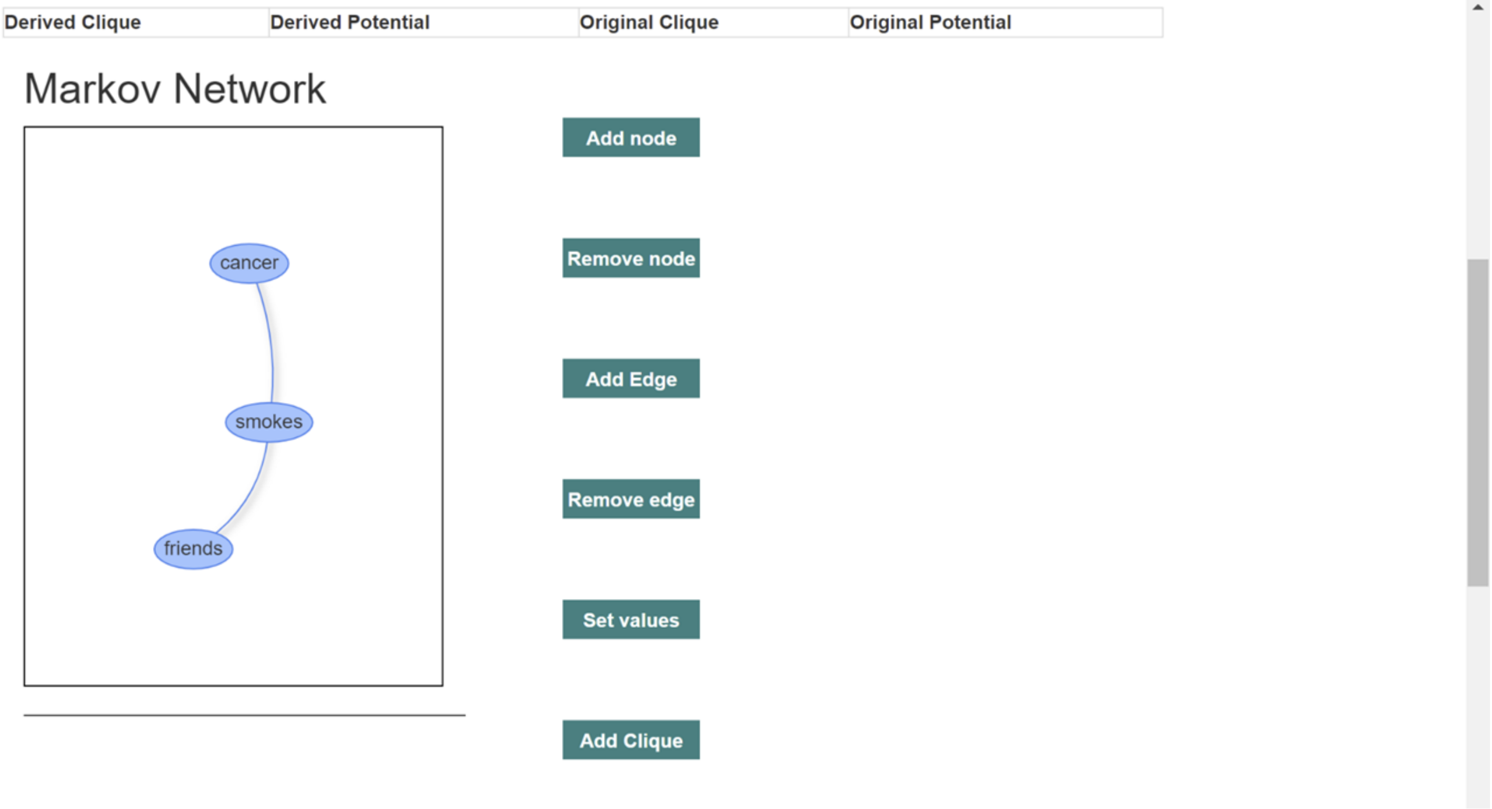
$$\tilde{p}(A,B,C,D) = \phi(A,B)\phi(B,C)\phi(C,D)\phi(D,A),$$

$$p(A,B,C,D) = \frac{1}{Z}\tilde{p}(A,B,C,D),$$

(A,B), (B,C), (C,D), (D,A) are friends

https://ermongroup.github.io/cs228-notes/representation/undirected/

MRF : Simulations



MRF : Simulations

Node values table

Node	Values
friends	0,1,2
smokes	0,1
cancer	0,1

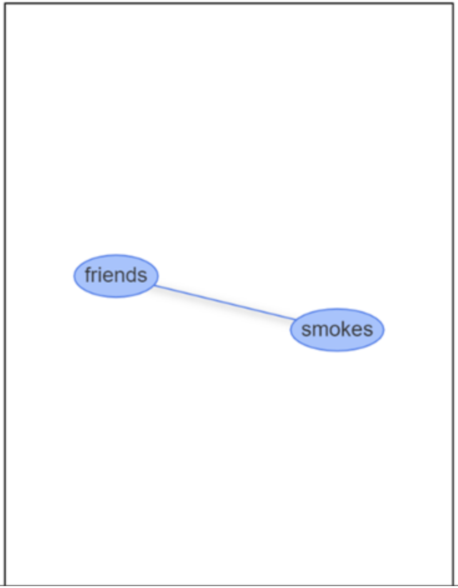
Potentials Table

Clique Variables	Variable Assignment	Potential
["friends","smokes"]	{"friends":"1","smokes":"1"}	0.800
["friends","smokes"]	{"friends":"2","smokes":"1"}	0.900
["smokes","cancer"]	{"smokes":"1","cancer":"1"}	0.900
["smokes","cancer"]	{"smokes":"0","cancer":"1"}	0.00
["smokes","cancer"]	{"smokes":"1","cancer":"0"}	0.900

Derived Potential Table

Derived Clique	Derived Potential	Original Clique	Original Potential
{"smokes":"1"}	1.800	{"smokes":"1","cancer":"1"}	0.900
		{"smokes":"1","cancer":"0"}	0.900
{"smokes":"0"}	0.000	{"smokes":"0","cancer":"1"}	0.000

Markov Network



Add node

Remove node

Add Edge

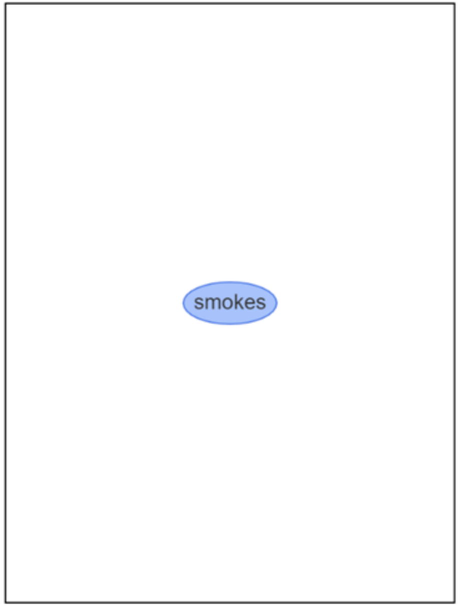
Remove edge

Set values

Derived Potential Table

Derived Clique	Derived Potential	Original Clique	Original Potential
{"smokes":"1"}	1.700	{"friends":"1","smokes":"1"}	0.800
		{"friends":"2","smokes":"1"}	0.900

Markov Network



Add node

Remove node

Add Edge

Remove edge

Set values

Markov Random Field

Markov Random Fields: Undirected Graphs

- X_v - set of Random variables
- X_v\N[v] - set of all other non-neighboring Random variables
- X_v is Independent of X_v\N[v] given X_N(v) where
- X_N(v) are neighbors of X_v in case of Markov Random Field

$$X_v \perp\!\!\!\perp X_{V \setminus \mathrm{N}[v]} \mid X_{\mathrm{N}(v)}$$

Markov Random Field

Φ(x) as the unnormalized probability distribution, the product of all potential functions i.e takes a value of 1 for all formulas that evaluate to True and 0 otherwise

$$P(X=x)=\frac{1}{Z}\prod_{C}\phi_C(x_c)=\frac{1}{Z}\Phi(x) \quad (1)$$

Exponential Family of Markov Random Field is Markov Network

$$P(X=x)=\frac{1}{Z}\exp\left(\sum_k w_k^\top f_k(x_{\{k\}})\right) \qquad Z=\sum_{x\in\mathcal{X}}\exp\left(\sum_k w_k^\top f_k(x_{\{k\}})\right).$$

MRF : Variable elimination algorithm

For each Variable F_i

- Multiply all factors Φ_i containing F_i
- Marginalize out F_i to obtain a new factor τ
- Replace the factors Φ_i with τ

Marginal inference: what is the probability of a given F_i in our model after we sum everything else out (e.g., probability of smoker vs. non-smoker)?

MRF : For very large number of variables ∀x

Choosing variable elimination orderings

Unfortunately, choosing the optimal VE ordering is an NP-hard problem. However, in practice, we may resort to the following heuristics:

- *Min-neighbors*: Choose a variable with the fewest dependent variables.
- *Min-weight*: Choose variables to minimize the product of the cardinalities of its dependent variables.
- *Min-fill*: Choose vertices to minimize the size of the factor that will be added to the graph.

Understanding why we need Markov Logic Networks

Markov Logic Network

Learning from one example

Markov Logic Network

- Logic handles complexity
- Probability handles uncertainty

http://swoh.web.engr.illinois.edu/courses/IE598/handout/fall2016_slide16.pdf

Markov Logic Network

- Logic handles complexity
- Probability handles uncertainty

http://swoh.web.engr.illinois.edu/courses/IE598/handout/fall2016_slide16.pdf

Understanding why we need Markov Logic Networks

Deep learning networks or present neural networks, deep learning approaches - Black box

We need to know how model “inferred” the result

We need concrete logical steps

But first order logic also provides set if predicates, which result in binary outputs.

With Markov Logic we can add probability to each predicate

Markov Logic Networks

Markov Logic Network

Learning from one example

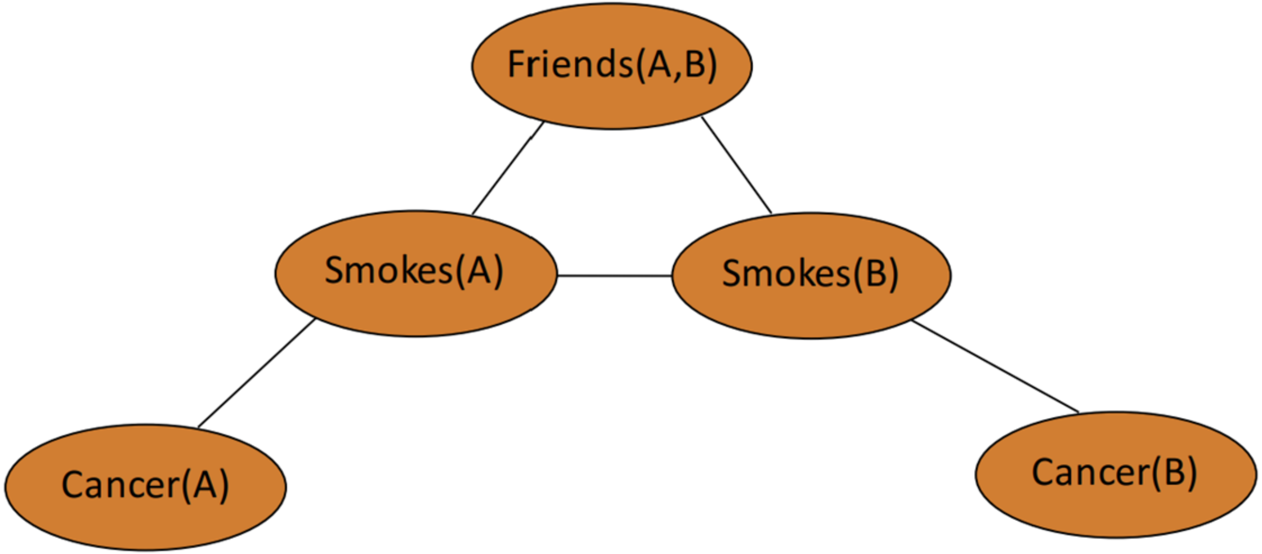
FOL vs Markov Random Field

FOL :

$$\forall x \; Smokes(x) \Rightarrow Cancer(x)$$

$$\forall x, y \; Friends(x, y) \Rightarrow \left(Smokes(x) \Leftrightarrow Smokes(y) \right)$$

MRF:



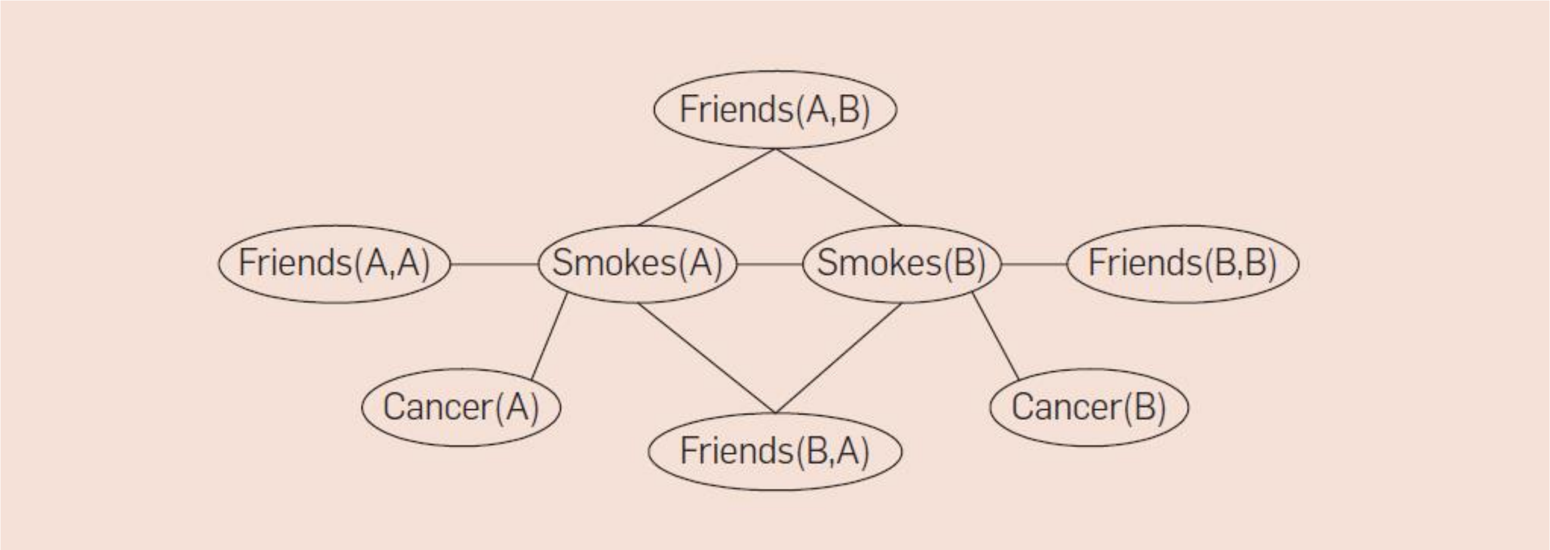
http://swoh.web.engr.illinois.edu/courses/IE598/handout/fall2016_slide16.pdf

FOL, MLN

English	First-order logic	Weight
"Friends of friends are friends."	$\forall x \forall y \forall z\ Fr(x, y) \wedge Fr(y, z) \Rightarrow Fr(x, z)$	0.7
"Friendless people smoke."	$\forall x\ (\neg(\exists y\ Fr(x, y)) \Rightarrow Sm(x))$	2.3
"Smoking causes cancer."	$\forall x\ Sm(x) \Rightarrow Ca(x)$	1.5
"If two people are friends, then either both smoke or neither does."	$\forall x \forall y\ Fr(x, y) \Rightarrow (Sm(x) \Leftrightarrow Sm(y))$	1.1

Fr() **is short for** Friends(), Sm() **for** Smokes(), **and** Ca() **for** Cancer().

Ground Markov network from Markov Logic Network



http://swoh.web.engr.illinois.edu/courses/IE598/handout/fall2016_slide16.pdf

MLN

Marginal Distribution of corresponding predicate.

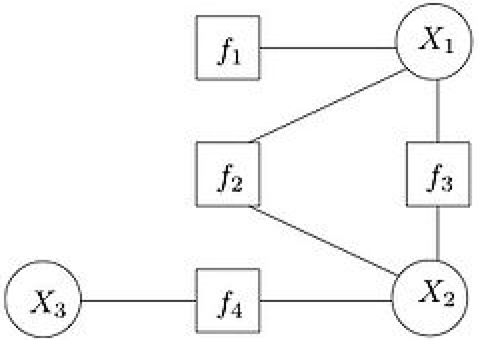
Log(odds) = logit(P) = ln(P/(1-P))

Probability of a world should increase as number of formulas that it violates decreases.

So higher the weight, less likely to occur.

FOL vs MLN

Verb/Predicate/Formula f_i	FOL (0 - violation)	MLN (likelihood of less violation)
smokes(x)	0 or 1	[0,1]
hasCancer(x)	0 or 1	[0,1]
Friends(x)	0 or 1	[0,1]



FOL : Some possible worlds: Domain(Alice, Bob)

Possible worlds	smokes(a)	Cancer(a)	smokes(b)	Cancer(b)	Friends **(a,b)**	Model
1	0	0	0	0	1	True
2	0	1	0	1	0	False
3	1	0	1	0	1	True
						SAT#: 2

Markov Logic Network Formulation

Our Markov Network scenario is a binary, first-order logic, knowledge base (KB) scenario.

Potential function state

Number of True groundings

Sum of true grounding weights

$$P(X = x) = \frac{1}{Z}\prod_i \phi_i\left(x_{\{i\}}\right)^{n_i(x)} = \frac{1}{Z}\exp\left(\sum_i w_i n_i(x)\right)$$

Weight

True groundings

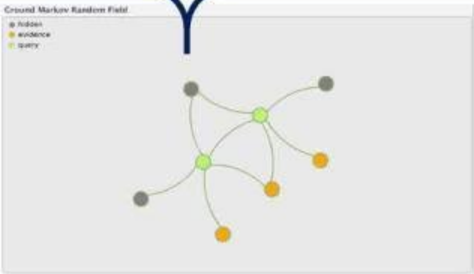
A pair of inputs
stopped(V1), stopped(V2)

Partition Function
(normalization)

Potential function

$x \in \{(0,0),(0,1),(1,0),(1,1)\}$

The probability of each state
$P(0,0), P(0,1)$
$P(1,0), P(1,1)$



To obtain a probability space, divide the weight of each world by Z = sum of weights of all worlds:

Markov Logic Network

$$Z = (w_1+\underline{w}_1)\ (w_2+\underline{w}_2)\ (w_3+\underline{w}_3)\ \ldots$$

Possible worlds	smokes(a)	smokes(b)	Friends **(a,b)**	Model	Weights	WFOMC
1	0	0	1	True	2*2*3 = 4	exp(12)
2	0	0	0	False	2*2*5 = 20	exp(20)
3	1	1	1	True	1*1*3 = 3	exp(3)
			SAT #: 2		WMC: 27	

Smokes: 1, Not smokes: 2, Friends: 3, Not Friends: 5

https://web.cs.ucla.edu/~guyvdb/talks/IJCAI16-tutorial/Part%203%20-%20Weighted%20Model%20Counting.pdf

Markov Logic Network : MC Satisfiability Algorithm

Algorithm 1 **MC-SAT**(*clauses, weights, num_samples*)

```
x^(0) ← Satisfy(hard clauses)
for i ← 1 to num_samples do
    M ← ∅
    for all c_k ∈ clauses satisfied by x^(i−1) do
        With probability 1 − e^(−w_k) add c_k to M
    end for
    Sample x^(i) ~ U_SAT(M)
end for
```

$x^{(0)} \leftarrow$ Satisfy(hard *clauses*)
for $i \leftarrow 1$ to *num_samples* **do**
  $M \leftarrow \emptyset$
  for all $c_k \in$ *clauses* satisfied by $x^{(i-1)}$ **do**
    With probability $1 - e^{-w_k}$ add c_k to M
  end for
  Sample $x^{(i)} \sim \mathcal{U}_{SAT(M)}$
end for

Markov Logic Network

https://colab.research.google.com/drive/1lLs-78bAcAvgs2ZlTvG__U5X27f2AQGa?usp=sharing

https://colab.research.google.com/drive/1UxY0bdG9_Oy05vWMP6XgpDrnenVgj3qi?usp=sharing

Markov Logic Network : All possible simulated worlds: 256

```
# Generate the all possible worlds
X = pd.DataFrame(columns=ground_atoms, data=list(product([1,0], repeat=len(ground_atoms))))
X.head()
```

	(Smokes, A)	(Smokes, B)	(Cancer, A)	(Cancer, B)	(Friends, A, A)	(Friends, A, B)	(Friends, B, A)	(Friends, B, B)
0	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	0
2	1	1	1	1	1	1	0	1
3	1	1	1	1	1	1	0	0
4	1	1	1	1	1	0	1	1

Markov Logic Network

An MLN can be viewed as a template for constructing Markov networks. From Definition 1 and Equations 1 and 2, the probability distribution over possible worlds x specified by the ground Markov network ML, C is given by

$$P(y|x) = \frac{1}{Z_x}\exp\left(\sum_{i\in F_Y} w_i n_i(x,y)\right) \quad (5)$$

```
[68] S = np.zeros(len(X))
     for f, neg, w in grounded_formulas:
       S += w * np.logical_xor(X[f], neg).any(1)
```

```
/usr/local/lib/python3.7/dist-packages/numpy/core/_asarray.py:83: VisibleDeprecationWarning: Creating an ndarray from ragged nested sequences (which is
  return array(a, dtype, copy=False, order=order)
```

```
[69] #partition function
     logZ = logsumexp(S)
```

```
[70] #joint probability
     joint_probability = X.copy()
     joint_probability['logP'] = S - logZ
     print("**********Joint Probability*********************")
     print(joint_probability)
```

Markov Logic Network

+ Code + Text

```
print("*********P(Friends(A,B)|Smokes(A))***************")
P_Friends_AB_SmA = np.exp(joint_probability.groupby([('Smokes','A'),('Friends', 'A', 'B')])['logP'].agg(logsumexp))
P_SmA = np.exp(joint_probability.groupby([('Smokes','A')])['logP'].agg(logsumexp))
print(P_Friends_AB_SmA/P_SmA)
```

```
*********P(Friends(A,B)|Smokes(A))***************
(Smokes, A)  (Friends, A, B)
0            0                  0.553456
             1                  0.446544
1            0                  0.605285
             1                  0.394715
Name: logP, dtype: float64
```

https://colab.research.google.com/drive/1lLs-78bAcAvgs2ZlTvG__U5X27f2AQGa?usp=sharing

Markov Logic Network

```
// domain declarations

// predicate declarations
Smokes(person)
Friends(person, person)
Cancer(person)

// formulas
1.126769    Smokes(x) => Cancer(x)
1.577776    Friends(x, y) => (Smokes(x) <=> Smokes(y))
```

https://colab.research.google.com/drive/1ILs-78bAcAvgs2ZlTvG__U5X27f2AQGa?usp=sharing

Markov Logic Network: Complex case

pracmln_base_case.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

```
import timeit
import sys
import os
import time
from pracmln import query
from pracmln import MLN, Database, query

pth = os.path.join("/content/drive/MyDrive/Colab Notebooks/mln (1)/pracmln/examples/smokers/mlns/",
                   'wts.pybpll.smoking-train-smoking.mln')
print(os.path.exists(pth))
mln = MLN(mlnfile=pth, grammar='StandardGrammar')
pth = os.path.join("/content/drive/MyDrive/Colab Notebooks/mln (1)/pracmln/examples/smokers/dbs/", 'smoking-test-smaller.db')
db = Database(mln, dbfile=pth)
with open(os.path.join("/content/drive/MyDrive/Colab Notebooks/mln (1)/pracmln/examples/smokers/", 'performance.txt'), 'a') as fl:

  start = time.time()
  query(queries='Cancer,Smokes,Friends', method='MC-SAT', mln=mln, db=db, verbose=False, multicore=True).run()
  t1 = time.time()-start
  start = time.time()
  query(queries='Cancer,Smokes,Friends', method='MC-SAT', mln=mln, db=db, verbose=False, multicore=False).run()
  t2 = time.time()-start
  print('Inference, MC-SAT, {}, {}'.format(t1, t2))
  fl.write(str(t1)+str(t2))
  fl.write('\t(Inference, MC-SAT)\n')
```

```
True

Inference, MC-SAT, 0.38447141647338867, 0.4017677307128906
```

https://colab.research.google.com/drive/1UxY0bdG9_Oy05vWMP6XgpDrnenVgj3qi?usp=sharing

Markov Logic Network: Complex case

https://github.com/sushmaakoju/markov-logic-networks

https://colab.research.google.com/drive/1UxY0bdG9_Oy05vWMP6XgpDrnenVgj3qi?usp=sharing

Markov Logic Network: Complex case

The most challenging part was to explain this with simple example by understanding the complex parts of the MLN.

References

1. Unifying Logic and Probability
2. An Introduction to Probabilistic Programming
3. Markov Logic : A step towards AI
4. Uncertainity Modeling in AI
5. Markov Logic
6. Markov logic networks
7. First-Order Probabilistic Reasoning

Q&A

Thank you!