Fine Grained Weight Learning in Markov Logic Networks

Happy Mittal, Shubhankar Suman Singh — Dept. of Comp. Sci. & Engg., IIT Delhi

Vibhav Gogate — Dept. of Comp. Sci., UT Dallas

Parag Singla — Dept. of Comp. Sci. & Engg., IIT Delhi

INTRODUCTION

Markov Logic Networks (MLN) [Richardson & Domingos, 2006]

- A Markov Logic Network (MLN) is a set of pairs (F,w) where
  - F is a formula in first order Logic
  - w is a real number (weight of formula)
- When a world violates a formula, it becomes less probable, but not impossible.
- Together with a set of constants, it defines a Markov network.
  - Ground atoms become nodes.
  - Ground formulas become features.

$\Delta_x = \Delta_y = \{Ana, Bob, ...\}$

1.5	$\forall x\ Smokes(x) \Rightarrow Cancer(x)$
1.1	$\forall x, y\ Smokes(x) \wedge Friends(x, y) \Rightarrow Smokes(y)$

- **Joint Probability :**

y : query atoms
x : evidence atoms

$$P(y \mid x; w) = \frac{1}{Z_x} \exp\left(\sum_i w_i\, n_i(x, y)\right)$$

w_i: Weight of Formula i; $n_i(x, y)$: No. of true groundings of formula i

- **Weight Learning :**

Generative Learning

- No separate notion of evidence or query.
- Weights learned by maximizing log-likelihood of entire set of ground atoms.
- Pseudo log-likelihood :

$$\log P^*(X = x) = \sum_{l=1}^{n} \log P(X_l = x_l \mid MB(X_l))$$

- Maximize gradient of pseudo log-likelihood instead.

Discriminative Learning

- Weights learned by maximizing log-likelihood of the query atoms given evidence atoms.
- Gradient of conditional log-likelihood is :

$$\frac{\partial}{\partial w_i} \log P(y \mid x) = n_i(x, y) - E[n_i(x, y)]_{P(Y|X)}$$

- Preconditioned Scaled Conjugate Gradient (PSCG) : One of the popular algorithms [Lowd & Domingos 2007]

MOTIVATION

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

Per constant learning

$w_1 : Smokes(P_1) \Rightarrow Cancer(P_1)$
$w_2 : Smokes(P_2) \Rightarrow Cancer(P_2)$
$\vdots$
$w_n : Smokes(P_n) \Rightarrow Cancer(P_n)$

Fine Grained Learning

$w_1 : Smokes(s_1) \Rightarrow Cancer(s_1)$, $\Delta s_1 = \{P_1, P_2, ..., P_{l_1}\}$
$w_2 : Smokes(s_2) \Rightarrow Cancer(s_2)$, $\Delta s_2 = \{P_{l_1+1}, ..., P_{l_2}\}$
$\vdots$
$w_k : Smokes(s_k) \Rightarrow Cancer(s_k)$, $\Delta s_k = \{P_{l_{k-1}}, ..., P_{l_k}\}$

First order learning

$w : Smokes(x) \Rightarrow Cancer(x)$, $\Delta x = \{P_1, P_2, ..., P_n\}$

High Variance ← **Right tradeoff b/w Bias and Variance** → **High Bias**

- Objective of the paper :
  - Automatically partition the ground formulas.
  - Each partition may show statistical regularity with different weight.
  - Learn separate weight for each partition.

PARTITIONING THE GROUND FORMULAS

- Partition the underlying constants into subtypes.
- Partition ground formulas with same subtype signatures.

$Smokes(x) \wedge Friends(x, y) \Rightarrow Smokes(y)$
$\Delta x = \Delta y = \{A, B, C\}$

Partitioning of constants

$sub(A) = sub(B) = s_1$
$sub(C) = s_2$

- Granularity of first order formula
  - Number of Partitions
  - $g_f = \prod_{j=1}^{l} |SubTypeSet(t_j)|$

Type of j^{th} variable

$w_1 : Smokes(x) \wedge Friends(x, y) \Rightarrow Smokes(y)$, $\Delta x = \Delta y = \{A.B\}$

$w_1 : Smokes(x) \wedge Friends(x, y) \Rightarrow Smokes(y)$, $\Delta x = \{A, B\}, \Delta y = \{C\}$

$w_1 : Smokes(x) \wedge Friends(x, y) \Rightarrow Smokes(y)$, $\Delta x = \{C\}, \Delta y = \{A, B\}$

$w_1 : Smokes(x) \wedge Friends(x, y) \Rightarrow Smokes(y)$, $\Delta x = \Delta y = \{C\}$

SUBTYPING OF CONSTANTS

Incorporating subtypes of constants in MLN

- Subtype of constant added by creating predicate $HasSubType^t(subtype, const)$
  - A grounding $HasSubType^t(ST, c)$ is true if subtype of constant c of type t is ST.
  - Modify MLN according to algorithm 1
- Probability distribution over modified MLN

$$P(y, s \mid x) = \frac{1}{Z_x} \exp\left(\sum_{i=1}^{n} \sum_{j=1}^{g_{f_i}} w_i^j n_i^j(x, y, s)\right) - (1)$$

w_i^j: Truth assignment to all groundings of $HasSubType^t(st, c)$

$n_i^j(x, y, s)$: Weight of partition j of formula i

No. of true groundings of partition j of formula i

Algorithm 1

ModifyMLN(MLN M, **Type** t)
  M' = ϕ
  for each (f, w) in M **do**
    for each variable v of type t in f **do**
      $f \leftarrow f \wedge HasSubType^t(+st, v)$
    end for
    Add (f, w) to M'
  end for
  return M'

LEARNING WEIGHTS

Weight Learning

Subtypes are known → Learn Discriminatively by (1); Gradient of (1) : $\frac{\partial}{\partial w_i} \log P(y, s \mid x) = n_i(x, y, s) - E[n_i^j(x, y, s)]_{P(Y,S|X)}$

Subtypes are hidden → Model the subtypes as hidden predicates. Two methods to deal with this case : K-means clustering and Joint learning of Sub Types

Method 1 : K-means Clustering

$P(x) \Rightarrow Q(y)$	$\Delta x = \Delta z = \{c, d\}$
$P(z)$	$\Delta y = \{1, 2\}$

Learn untied formulas

$w_1 : P(c) \Rightarrow Q(1)$	$w_3 : P(d) \Rightarrow Q(1)$	$w_5 : P(c)$
$w_2 : P(c) \Rightarrow Q(2)$	$w_4 : P(d) \Rightarrow Q(2)$	$w_6 : P(d)$

Create feature vectors

c	$\frac{w_1 + w_2}{2}$	w_5
d	$\frac{w_3 + w_4}{2}$	w_6
1	$\frac{w_1 + w_3}{2}$	
2	$\frac{w_2 + w_4}{2}$	

Method 2 : Joint learning of SubTypes

- Treat subtypes as hidden variables.
- Maximize log-likelihood by summing over hidden variables :

$$P(y \mid x) = \sum_s \frac{1}{Z_x} \exp\left(\sum_{i=1}^{n} \sum_{j=1}^{g_{f_i}} w_i^j n_i^j(x, y, s)\right)$$

EM Algorithm → E Step : Fill in s values; M Step : Find ML parameters using filled in values

Generalize to new Domains

- Treat subtypes as hidden variables.
- Infer the subtypes jointly with other query variables.
- Generalization not possible in parameter tying approach of Chou et al (2016).

EXPERIMENTS & RESULTS

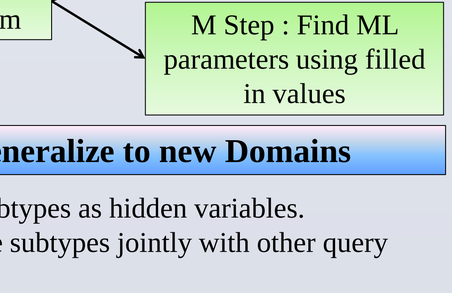
IMDB Dataset

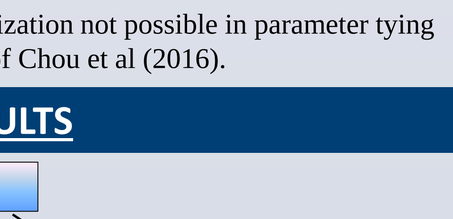
Rules
- WorkedUnder(p1,p2) => Actor(p1)
- WorkedUnder(p1,p2) => Director(p2)
- Actor(p1) ^ Director(p2) ^ Movie(m,p1) ^ Movie(m,p2) => WorkedUnder(p1,p2)
- Actor(p1) ^ Director(p2) ^ Movie(m,p1) ^ WorkedUnder(p1,p2) => Movie(m,p2)
- Actor(p1) ^ Director(p2) ^ Movie(m,p2) ^ WorkedUnder(p1,p2) => Movie(m,p1)

Predicates
- Actor(person), Director(person)
- Movie(movie, person)
- WorkedUnder(person, person)

Methodology
- Subtyping done on Persons (by K-means)
- Evidence : Random subset of ground atoms.
- Query : Rest of the ground atoms.

Vary Evidence / Vary Clusters





Future Work
- Experimenting with alternative clustering approaches
  - Structure Learning by Kok & Domingos (2009)
  - Evidence based clustering by Venugopal et. Al (2014)
  - Approximate Binary evidence by low-rank boolean matrix factorization (Broeck & darwiche 2013)
- Experimenting with joint learning algorithm.
- Compute PAC bounds for our learning framework.