

Applications

November 20, 2008

CS 486/686

University of Waterloo

Outline

- Alchemy applications
- Readings:
 - Marc Summer and Pedro Domingos (2007),
The Alchemy Tutorial, Department of
Computer Science and Engineering,
University of Washington

Multinomial Distribution

Example: Throwing die

Types: $\text{throw} = \{ 1, \dots, 20 \}$

$\text{face} = \{ 1, \dots, 6 \}$

Predicate: $\text{Outcome}(\text{throw}, \text{face})$

Formulas: $\text{Outcome}(t, f) \wedge f \neq f' \Rightarrow \neg \text{Outcome}(t, f').$
 $\text{Exist } f \text{ Outcome}(t, f).$

Too cumbersome!

Multinomial Distrib.: ! Notation

Example: Throwing die

Types: $\text{throw} = \{ 1, \dots, 20 \}$

$\text{face} = \{ 1, \dots, 6 \}$

Predicate: $\text{Outcome}(\text{throw}, \text{face}!)$

Formulas:

Semantics: Arguments without "!" determine args with "!".
Only one face possible for each throw.

Multinomial Distrib.: + Notation

Example: Throwing biased die

Types: `throw = { 1, ... , 20 }`

`face = { 1, ... , 6 }`

Predicate: `Outcome(throw, face!)`

Formulas: `Outcome(t, +f)`

Semantics: Learn weight for each grounding of args with "+".

Text Classification

page = { 1, ... , n }
word = { ... }
topic = { ... }

Topic(page, topic!)
HasWord(page, word)
Links(page, page)

HasWord(p, +w) => **Topic**(p, +t)
Topic(p, t) ^ **Links**(p, p') => **Topic**(p', t)

Information Retrieval

InQuery(word)

HasWord(page, word)

Relevant(page)

Links(page, page)

InQuery(+w) $\wedge$ HasWord(p, +w) $\Rightarrow$ Relevant(p)

Relevant(p) $\wedge$ Links(p, p') $\Rightarrow$ Relevant(p')

Cf. L. Page, S. Brin, R. Motwani & T. Winograd, “The PageRank Citation Ranking: Bringing Order to the Web,” Tech. Rept., Stanford University, 1998.

Record deduplication

Problem: Given database, find duplicate records

HasToken(token, field, record)

SameField(field, record, record)

SameRecord(record, record)

HasToken(+t, +f, r) $\wedge$ HasToken(+t, +f, r')
 $\Rightarrow$ SameField(f, r, r')

SameField(+f, r, r') $\Rightarrow$ SameRecord(r, r')

SameRecord(r, r') $\wedge$ SameRecord(r', r'')
 $\Rightarrow$ SameRecord(r, r'')

Cf. A. McCallum & B. Wellner, “Conditional Models of Identity Uncertainty with Application to Noun Coreference,” in *Adv. NIPS 17*, 2005.

Record resolution

Can also resolve fields:

HasToken(token, field, record)
SameField(field, record, record)
SameRecord(record, record)

HasToken(+t, +f, r) ^ HasToken(+t, +f, r')
 => SameField(f, r, r')
SameField(+f, r, r') <=> SameRecord(r, r')
SameRecord(r, r') ^ SameRecord(r', r'')
 => SameRecord(r, r'')
SameField(f, r, r') ^ SameField(f, r', r'')
 => SameField(f, r, r'')

More: P. Singla & P. Domingos, “Entity Resolution with Markov Logic”, in *Proc. ICDM-2006*.

Information Extraction

- **Problem:** Extract database from text or semi-structured sources
- **Example:** Extract database of publications from citation list(s) (the "CiteSeer problem")
- Two steps:
 - **Segmentation:**
Use HMM to assign tokens to fields
 - **Record resolution:**
Use logistic regression and transitivity

Information Extraction

Token(token, position, citation)
InField(position, field, citation)
SameField(field, citation, citation)
SameCit(citation, citation)

Token(+t,i,c) => InField(i,+f,c)
InField(i,+f,c) <=> InField(i+1,+f,c)
 $f \neq f' \Rightarrow (!\text{InField}(i,+f,c) \vee !\text{InField}(i,+f',c))$

Token(+t,i,c) ^ InField(i,+f,c) ^ Token(+t,i',c')
^ InField(i',+f,c') => SameField(+f,c,c')
SameField(+f,c,c') <=> SameCit(c,c')
SameField(f,c,c') ^ SameField(f,c',c'') => SameField(f,c,c'')
SameCit(c,c') ^ SameCit(c',c'') => SameCit(c,c'')

Information Extraction

Token(token, position, citation)
InField(position, field, citation)
SameField(field, citation, citation)
SameCit(citation, citation)

Token(+t,i,c) => InField(i,+f,c)
InField(i,+f,c) ^ !Token(".",i,c) <=> InField(i+1,+f,c)
f != f' => (!InField(i,+f,c) v !InField(i,+f',c))

Token(+t,i,c) ^ InField(i,+f,c) ^ Token(+t,i',c')
^ InField(i',+f,c') => SameField(+f,c,c')
SameField(+f,c,c') <=> SameCit(c,c')
SameField(f,c,c') ^ SameField(f,c',c'') => SameField(f,c,c'')
SameCit(c,c') ^ SameCit(c',c'') => SameCit(c,c'')

More: H. Poon & P. Domingos, "Joint Inference in Information Extraction", in *Proc. AAAI-2007*.

Next Class

- Lifted inference