NLP Programming Tutorial 7 - Topic Models

Graham Neubig
Nara Institute of Science and Technology (NAIST)

Topics in Documents

- In general, documents can be grouped into topics

Cuomo to Push for Broader Ban on Assault Weapons	2012 Was Hottest Year in U.S. History
…	…
…	…
…	…
…	…

Topics in Documents

- In general, documents can be grouped into topics

Cuomo to Push for Broader Ban on Assault Weapons … … … …	2012 Was Hottest Year in U.S. History … … … …
New York Politics Weapons Crime	Weather Climate Statistics U.S.

Topic Modeling

- Topic modeling finds topics Y given documents X

X

Cuomo to Push for Broader Ban on Assault Weapons

…
…
…
…

2012 Was Hottest Year in U.S. History

…
…
…
…

Topic Modeling

Y

New York
Politics
Weapons
Crime

Weather
Climate
Statistics
U.S.

- A type of "structured" prediction

Probabilistic Generative Model

- We assume some probabilistic model generated the topics Y and documents X jointly

$$P(Y, X)$$

- The topics Y with highest joint probability given X also has the highest conditional probability

$$\underset{Y}{\operatorname{argmax}}\, P(Y|X) = \underset{Y}{\operatorname{argmax}}\, P(Y, X)$$

Generative Topic Model

- Assume we have words X and topics Y:

X = Cuomo to Push for Broader Ban on Assault Weapons

Y = NY Func Pol Func Pol Pol Func Crime Crime

NY=New York, Func=Function Word, Pol=Politics, Crime=Crime

- First decide topics (independently)

$$P(Y)=\prod_{i=1}^{I} P(y_i)$$

- Then decide words given topics (independently)

$$P(X|Y)=\prod_{i=1}^{I} P(x_i|y_i)$$

Unsupervised Topic Modeling

- Given only the documents X, find topic-like clusters Y

X

Cuomo to Push for Broader Ban on Assault Weapons	2012 Was Hottest Year in U.S. History
…	…
…	…
…	…
…	…

Y

32	5
24	18
10	49
19	37

Unsupervised Topic Modeling

- A type of "structured" prediction
- But unlike before, we have no labeled training data!

Latent Dirichlet Allocation

- Most popular generative model for topic modeling
- First generate model parameters θ: $P(\theta)$
- For every document in X:
  - Generate document topic distribution T_i: $P(T_i|\theta)$
  - For each word $x_{i,j}$ in X_i:
    - Generate word topic $y_{i,j}$: $P(y_{i,j}|T_i,\theta)$
    - Generate the word $x_{i,j}$: $P(x_{i,j}|y_{i,j},\theta)$

$$P(X,Y)=\int_\theta P(\theta)\prod_i P(T_i|\theta)\prod_j P(y_{i,j}|T_i,\theta)P(x_{i,j}|y_{i,j},\theta)$$

Maximum Likelihood Estimation

- Assume we have words X and topics Y:

X_1 =	Cuomo	to	Push	for	Broader	Ban	on	Assault	Weapons
Y_1 =	32	7	24	7	24	24	7	10	10

- Can decide the topic distribution for each document:

$$P(y|Y_i) = c(y, Y_i)/|Y_i|$$ e.g.: $$P(y=24|Y_1) = 2/9$$

- Can decide word distribution for each topic:

$$P(x|y) = c(x, y)/c(y)$$ e.g.: $$P(x=\text{assault}|y=10) = 1/2$$

Problem: Unobserved Variables

- Problem: We do not know the values of $y_{i,j}$
- Solution: Use a method for unsupervised learning
  - EM Algorithm
  - Variational Bayes
  - **Sampling**

Sampling Basics

- Generate a sample from probability distribution:

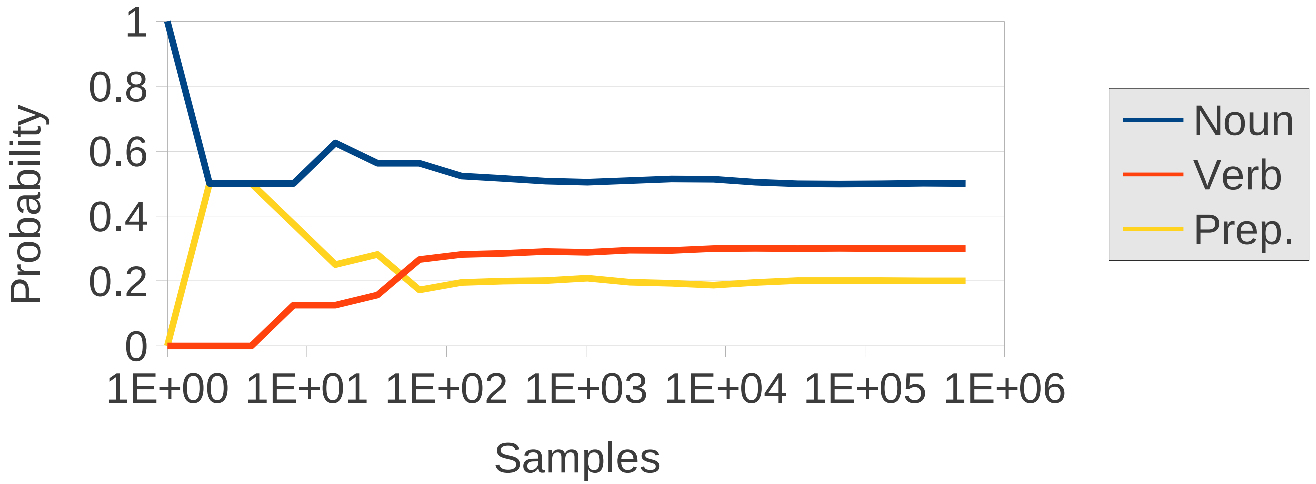
Distribution: P(Noun)=0.5 P(Verb)=0.3 P(Preposition)=0.2

Sample: Verb Verb Prep. Noun Noun Prep. Noun Verb Verb Noun …

- Count the samples and calculate probabilities

P(Noun)= 4/10 = 0.4, P(Verb)= 4/10 = 0.4, P(Preposition) = 2/10 = 0.2

- More samples = better approximation



Actual Algorithm

```
SampleOne(probs[])
    z = Sum(probs)
    remaining = Rand(z)
    for each i in 0 .. probs.size-1
        remaining -= probs[i]
        if remaining <= 0
            return i
```

Bug check, beware of overflow!

- Calculate sum of probs
- Generate number from uniform distribution over [0,z)
- Iterate over all probabilities
- Subtract current prob. value
- If smaller than zero, return current index as answer

Gibbs Sampling

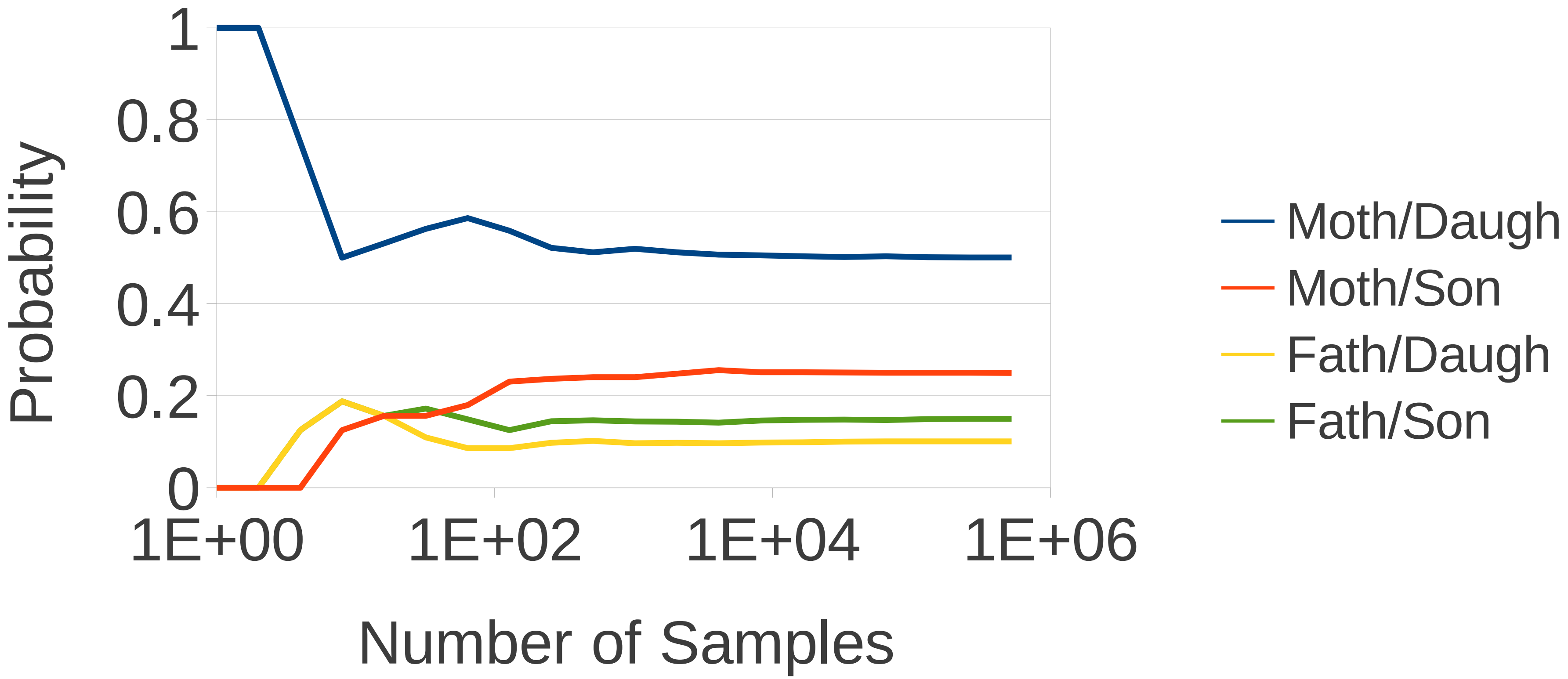
- Want to sample a 2-variable distribution P(A,B)
  - … but cannot sample directly from P(A,B)
  - … but can sample from P(A|B) and P(B|A)
- Gibbs sampling samples variables one-by-one to recover true distribution
- Each iteration:
  Leave A fixed, sample B from P(B|A)
  Leave B fixed, sample A from P(A|B)

Example of Gibbs Sampling

- Parent A and child B are shopping, what sex?
  P(Mother|Daughter) = 5/6 = 0.833
  P(Mother|Son) = 5/8 = 0.625
  P(Daughter|Mother) = 2/3 = 0.667
  P(Daughter|Father) = 2/5 = 0.4
- Original state: Mother/Daughter
  Sample P(Mother|Daughter)=0.833, chose Mother
  Sample P(Daughter|Mother)=0.667, chose Son
  c(Mother, Son)++
  Sample P(Mother|Son)=0.625, chose Mother
  Sample P(Daughter|Mother)=0.667, chose Daughter
  c(Mother, Daughter)++

…

Try it Out:



- In this case, we can confirm this result by hand

Sampling in Topic Models (1)

- Sample one $y_{i,j}$ at a time:

X_1 =	Cuomo	to	Push	for	Broader	Ban	on	Assault	Weapons
Y_1 =	5	7	4	$\boxed{7}$	3	4	7	6	6

- Subtract of $y_{i,j}$ and re-calculate topics and parameters

{0, 0, 1/9, 2/9, 1/9, 2/9, 3/9, 0}

{0, 0, 1/8, 2/8, 1/8, 2/8, 2/8, 0}

Sampling in Topic Models (2)

- Sample one $y_{i,j}$ at a time:

X_1 = Cuomo to Push for Broader Ban on Assault Weapons
Y_1 = 5 7 4 $\boxed{???}$ 3 4 7 6 6

- Multiply topic prob., by word given topic prob.:

Calculated from whole corpus

$$P(y_{i,j} \mid T_i) = \{0, 0, 0.125, 0.25, 0.125, 0.25, 0.25, 0\}$$

$$*$$

$$P(x_{i,j} \mid y_{i,j}, \theta) = \{0.01, 0.02, 0.01, 0.10, 0.08, 0.07, 0.70, 0.01\}$$

$$=$$

$$P(x_{i,j}, y_{i,j} \mid T_i, \theta) = \{0, 0, 0.00125, 0.01, 0.01, 0.00875, 0.175, 0\}/Z$$

Normalization constant

Sampling in Topic Models (3)

- Sample one value from this distribution:

$P(x_{i,j}, y_{i,j} \mid T_i, \theta)=\{\ 0,\ 0, 0.00125, 0.01, 0.01, \boxed{0.00875}, 0.175, 0\}/Z$

- Add the word with the new topic:

X_1 =	Cuomo	to	Push	for	Broader	Ban	on	Assault	Weapons
Y_1 =	5	7	4	**6**	3	4	7	6	6

- Update the counts and the probabilities:

{0, 0, 1/8, 2/8, 1/8, 2/8, 2/8, 0}

→

{0, 0, 1/9, 2/9, 1/9, 3/9, 2/9, 0}

Dirichlet Smoothing

- **Problem:** Many probabilities are zero!
  → Cannot escape from local minima
- **Solution:** Smooth the probabilities

Unsmoothed		Smoothed
$P(x_{i,j} \mid x_{i,j}) = \frac{c(x_{i,j}, y_{i,j})}{c(y_{i,j})}$	→	$P(x_{i,j} \mid y_{i,j}) = \frac{c(x_{i,j}, y_{i,j}) + \alpha}{c(y_{i,j}) + \alpha * N_x}$
$P(y_{i,j} \mid Y_i) = \frac{c(y_{i,j}, Y_i)}{c(Y_i)}$	→	$P(y_{i,j} \mid Y_i) = \frac{c(y_{i,j} \mid Y_i) + \beta}{c(Y_i) + \beta * N_y}$

- N_x and N_y are number of unique words and topics
- Equal to using a Dirichlet prior over the probabilities
  (More details in my Bayes tutorial)

Implementation: Initialization

```
make vectors xcorpus, ycorpus    # to store each value of x, y
make map xcounts, ycounts        # to store counts for probs
for line in file
    docid = size of xcorpus      # get a numerical ID for this doc
    split line into words
    make vector topics           # create random topic ids
    for word in words
        topic = RAND(NUM_TOPICS)    # random in [0,NUM_TOP)
        append topic to topics
        ADDCOUNTS(word, topic, docid, 1) # add counts
    append words (vector) to xcorpus
    append topics (vector) to ycorpus
```

Implementation: Adding Counts

ADDCOUNTS(*word, topic, docid, amount*)

xcounts[“topic”] += amount
xcounts[“word|topic”] += amount

for
$$P(x_{i,j}|y_{i,j}) = \frac{c(x_{i,j}, y_{i,j}) + \alpha}{c(y_{i,j}) + \alpha * N_x}$$

ycounts[“docid”] += amount
ycounts[“topic|docid”] += amount

for
$$P(y_{i,j}|Y_i) = \frac{c(y_{i,j}, Y_i) + \beta}{c(Y_i) + \beta * N_y}$$

bug check!
if any of these values < 0, throw error

Implementation: Sampling

```
for many iterations:
  for i in 0:SIZE(xcorpus):
    for j in 0:SIZE(xcorpus[i]):
      x = xcorpus[i][j]
      y = ycorpus[i][j]
      ADDCOUNTS(x, y, i, -1)   # subtract the counts (hence -1)
      make vector probs
      for k in 0 .. NUM_TOPICS-1:
        append P(x|k) * P(k|Y) to probs    # prob of topic k
      new_y = SAMPLEONE(probs)
      ll += log(probs[new_y])           # Calculate the log likelihood
      ADDCOUNTS(x, new_y, i, 1)     # add the counts
      ycorpus[i][j] = new_y
  print ll
print out wcounts and tcounts
```

Exercise

Exercise

- Write learn-lda
- Test the program, setting NUM_TOPICS to 2
  - Input: `test/07-train.txt`
  - Answer:
    - No correct answer! (Because sampling is random)
    - However, “a b c d” and “e f g h” should probably be different topics
- Train a topic model on `data/wiki-en-documents.word` with 20 topics
- Find some topics that match with your intuition
- Challenge: Change the model so you don't have to choose the number of topics in advance
(Read about <u>non-parametric</u> Bayesian techniques)

Thank You!