Mixed Membership Markov Models for Unsupervised Conversation Modeling

MICHAEL J. PAUL
JOHNS HOPKINS UNIVERSITY

The Center For Language and Speech Processing at the Johns Hopkins University

JOHNS HOPKINS UNIVERSITY
WHITING SCHOOL OF ENGINEERING
Conversation Modeling: High Level Idea

- We’ll be modeling sequences of documents
  - e.g. a sequence of email messages from a conversation

- We’ll use $M^4 = \text{Mixed Membership Markov Models}$

- $M^4$ is a combination of
  - Topic models (LDA, PLSA, etc.)
    - Documents are mixtures of latent classes/topics
  - Hidden Markov models
    - Documents in a sequence depend on the previous document
Generative Models of Text

- Some distinctions to consider...

<table>
<thead>
<tr>
<th>Intra-document structure</th>
<th>Inter-document structure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single-Class</td>
<td>Independent</td>
</tr>
<tr>
<td>Naïve Bayes</td>
<td>HMM</td>
</tr>
<tr>
<td>Mixed-Membership</td>
<td>LDA</td>
</tr>
<tr>
<td></td>
<td>This talk! 😊</td>
</tr>
</tbody>
</table>

Overview

- Unsupervised Content Models
  - Naïve Bayes
  - Topic Models

- Unsupervised Conversation Modeling
  - Hidden Markov Models

- Mixed Membership Markov Models (M^4)
  - Overview
  - Inference

- Experiments with Conversation Data
  - Thread reconstruction
  - Speech act induction

Motivation: Unsupervised Models

- Huge amounts of unstructured and unannotated data on the Web
- Unsupervised models can help manage this data and are robust to variations in language and genre
- Tools like topic models can uncover interesting patterns in large corpora

(Unsupervised) Naïve Bayes

• Each document belongs to some category/class $z$

• Each class $z$ is associated with its own distribution over words
(Unsupervised) Naïve Bayes

Probability distributions over words:
- football 0.03
- team 0.01
- hockey 0.01
- baseball 0.005
- ... ... 

- charge 0.02
- court 0.02
- police 0.015
- robbery 0.01
- ... ... 

- congress 0.02
- president 0.02
- election 0.015
- senate 0.01
- ... ... 

Imaginary class labels:
- “SPORTS”
- “CRIME”
- “POLITICS”


(Unsupervised) Naïve Bayes

- football 0.03
- team 0.01
- hockey 0.01
- baseball 0.005

- charge 0.02
- court 0.02
- police 0.015
- robbery 0.01

- congress 0.02
- president 0.02
- election 0.015
- senate 0.01

---

What if an article belongs to more than one category?
Jury Finds Baseball Star Roger Clemens Not Guilty On All Counts

A jury found baseball star Roger Clemens not guilty on six charges against. Clemens was accused of lying to Congress in 2008 about his use of performance enhancing drugs.
Topic Models

football 0.03
team 0.01
hockey 0.01
baseball 0.005
...
...

charge 0.02
court 0.02
police 0.015
robbery 0.01
...
...

congress 0.02
president 0.02
election 0.015
senate 0.01
...
...

Doc 1

Doc 2

Doc 3

Jury Finds Baseball Star Roger Clemens Not Guilty On All Counts

A jury found baseball star Roger Clemens not guilty on six charges against. Clemens was accused of lying to Congress in 2008 about his use of performance enhancing drugs.


Topic Models

- One class distribution $\theta_d$ per document
- One class value per token
  - (rather than per document)

Latent Dirichlet Allocation (LDA)

- One class distribution $\theta_d$ per document
- One class value per token (rather than per document)


Overview

- Unsupervised Content Models
- **Unsupervised Conversation Modeling**
- Mixed Membership Markov Models
- Experiments with Conversation Data
- Conclusion
Documents on the web are more complicated than news articles
Documents on the web are more complicated than news articles
What’s missing from Naïve Bayes and LDA?
- They assume documents are generated independently of each other

Messages in conversations aren’t at all independent
- Doesn’t make sense to pretend that they are
- But we’d like to represent this dependence in a reasonably simple way

Solution: Hidden Markov Models
Block HMM

- Message emitted at each time step of Markov chain

- Each message in thread depends on the message to which it is a response

Message 1
Message 2
Message 3

Bayesian Block HMM

- Each message in thread depends on the message to which it is a response

## Block HMM

<table>
<thead>
<tr>
<th>GREETING</th>
<th>SPORTS</th>
<th>CRIME</th>
<th>POLITICS</th>
</tr>
</thead>
<tbody>
<tr>
<td>hey 0.1</td>
<td>football 0.03</td>
<td>charge 0.02</td>
<td>congress 0.02</td>
</tr>
<tr>
<td>sup 0.06</td>
<td>team 0.01</td>
<td>court 0.02</td>
<td>president 0.02</td>
</tr>
<tr>
<td>hi 0.04</td>
<td>hockey 0.01</td>
<td>police 0.015</td>
<td>election 0.015</td>
</tr>
<tr>
<td>hello 0.01</td>
<td>baseball 0.005</td>
<td>robbery 0.01</td>
<td>senator 0.01</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>QUESTION</th>
<th>LAUGHTER</th>
<th>POLITICS</th>
</tr>
</thead>
<tbody>
<tr>
<td>what 0.03</td>
<td>lol 0.04</td>
<td>congress 0.02</td>
</tr>
<tr>
<td>what’s 0.025</td>
<td>haha 0.04</td>
<td>president 0.02</td>
</tr>
<tr>
<td>how 0.02</td>
<td>:) 0.03</td>
<td>election 0.015</td>
</tr>
<tr>
<td>is 0.02</td>
<td>lmao 0.01</td>
<td>senator 0.01</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

- hey, sup, hi, hello, what, what’s, how, is, lol, haha, :) lmao
- football, team, hockey, baseball, charge, court, police, robbery
- congress, president, election, senator


Block HMM

- Nice and simple way to model dependencies between messages

- This is similar to Naïve Bayes
  - One class per document!

- Let’s make it more like LDA
  - Documents are *mixtures* of classes
Generative Models of Text

Inter-document structure

<table>
<thead>
<tr>
<th>Single-Class</th>
<th>Independent</th>
<th>Markov</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><img src="image1" alt="Diagram" /></td>
<td><img src="image2" alt="Diagram" /></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Mixed-Membership</th>
<th>Independent</th>
<th>Markov</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image3" alt="Diagram" /></td>
<td><img src="image4" alt="Diagram" /></td>
<td><img src="image5" alt="Diagram" /></td>
</tr>
</tbody>
</table>

This talk! 😊
Overview

- Unsupervised Content Models
- Unsupervised Conversation Modeling
- Mixed Membership Markov Models
- Experiments with Conversation Data
- Conclusion
Mixed Membership Markov Models (M⁴)

- Like LDA
  - One distribution $\pi_d$ per doc
  - One class $z$ per token

- But now each message’s distribution depends on the class assignments of previous message

$\Lambda$ transition parameters

$\pi$ class distribution (function of $z$ and $\lambda$)

$z$ $w$

Message 1  Message 2  Message 3


Mixed Membership Markov Models (M⁴)

- **Core of M⁴:**
  Probability of class $j$ in message $d$
  \[ \pi_{dj} \propto \exp(\lambda_j^T z_{d-1}) \]

- Transition parameters
- Class distribution (function of $z$ and $\lambda$)

Message 1  Message 2  Message 3

Mixed Membership Markov Models (M⁴)

- Why not transition directly from $\pi$ to $\pi$?
- Makes more sense for next message to depend on actual classes of previous message (not the distribution over all possible classes)
Example

Suppose documents are mixtures of 4 classes: Y G R B

Then $\Lambda$ is a 4x4 matrix with values such as:

$\lambda_{G \rightarrow R} = -0.2 \quad \text{"The presence of G in doc 1 slightly decreases the likelihood of having R in doc 2"}$

$\lambda_{B \rightarrow B} = 5.0 \quad \text{"The presence of B in doc 1 greatly increases the likelihood of having B in doc 2"}$
- Multinomial parameters $\pi$
- Repeatedly sample $z$ from $\pi$
  - i.e. sample class histogram for doc 1

Counts of $z$:

- $Y$: 0
- $G$: 2
- $R$: 5
- $B$: 2
Example

\[
\pi_1 \propto \exp\left(0 \times \lambda_{Y \rightarrow Y} + 2 \times \lambda_{G \rightarrow Y} + 5 \times \lambda_{R \rightarrow Y} + 2 \times \lambda_{B \rightarrow Y}\right)
\]

Counts of \(z\):

- \(Y\): 0
- \(G\): 2
- \(R\): 5
- \(B\): 2

\[
\pi_{2Y} \propto \exp\left(0 \times \lambda_{Y \rightarrow Y} + 2 \times \lambda_{G \rightarrow Y} + 5 \times \lambda_{R \rightarrow Y} + 2 \times \lambda_{B \rightarrow Y}\right)
\]

\[
\pi_{2G} \propto \exp\left(0 \times \lambda_{Y \rightarrow G} + 2 \times \lambda_{G \rightarrow G} + 5 \times \lambda_{R \rightarrow G} + 2 \times \lambda_{B \rightarrow G}\right)
\]

\[
\pi_{2R} \propto \exp\left(0 \times \lambda_{Y \rightarrow R} + 2 \times \lambda_{G \rightarrow R} + 5 \times \lambda_{R \rightarrow R} + 2 \times \lambda_{B \rightarrow R}\right)
\]

\[
\pi_{2B} \propto \exp\left(0 \times \lambda_{Y \rightarrow B} + 2 \times \lambda_{G \rightarrow B} + 5 \times \lambda_{R \rightarrow B} + 2 \times \lambda_{B \rightarrow B}\right)
\]
Example

Counts of $z$:

- Doc 1:
  - $Y$: 0
  - $G$: 2
  - $R$: 5
  - $B$: 2

- Doc 2:
  - $Y$: 0
  - $G$: 2
  - $R$: 5
  - $B$: 2

$$
\pi_2Y \propto \exp(0 \times \lambda_{Y \rightarrow Y} + 2 \times \lambda_{G \rightarrow Y} + 5 \times \lambda_{R \rightarrow Y} + 2 \times \lambda_{B \rightarrow Y})
$$

$$
\pi_2G \propto \exp(0 \times \lambda_{Y \rightarrow G} + 2 \times \lambda_{G \rightarrow G} + 5 \times \lambda_{R \rightarrow G} + 2 \times \lambda_{B \rightarrow G})
$$

$$
\pi_2R \propto \exp(0 \times \lambda_{Y \rightarrow R} + 2 \times \lambda_{G \rightarrow R} + 5 \times \lambda_{R \rightarrow R} + 2 \times \lambda_{B \rightarrow R})
$$

$$
\pi_2B \propto \exp(0 \times \lambda_{Y \rightarrow B} + 2 \times \lambda_{G \rightarrow B} + 5 \times \lambda_{R \rightarrow B} + 2 \times \lambda_{B \rightarrow B})
$$
Example

Doc 1

\[ \pi \]

\[ z_1, z_2, z_3, z_4, z_5, z_6, z_7, z_8, z_9 \]

\[ Y: 0 \]
\[ G: 2 \]
\[ R: 5 \]
\[ B: 2 \]

Doc 2

\[ \pi \]

\[ z_1, z_2, z_3, z_4, z_5, z_6, z_7, z_8, z_9, z_{10} \]

\[ Y: 3 \]
\[ G: 1 \]
\[ R: 1 \]
\[ B: 5 \]

Doc 3

\[ \pi \]

\[ z_1, z_2, z_3, z_4, z_5, z_6, z_7, z_8 \]

\[ Y: 1 \]
\[ G: 2 \]
\[ R: 2 \]
\[ B: 3 \]


M^4 is a Markov chain where the state space is the set of all possible class histograms

- If no bound on document length, then the size of this space is countably infinite!
- But the transition matrix is given in terms of the same number parameters as in a standard HMM
(Approximate) Inference

- Monte Carlo EM
  - E-step: Sample from posterior over class assignments ($z$)
  - M-step: Direct optimization of transition parameters ($\lambda$)

- Inference algorithm alternates between:
  - 1 iteration of collapsed Gibbs sampling
  - 1 iteration (step) of gradient ascent

- Sampler is similar to LDA Gibbs sampler
  - Slower because the computing the relative probability of each class involves summing over all classes to compute $\exp(\lambda_j^T z_{d-1})$
Overview

- Unsupervised Content Models
- Unsupervised Conversation Modeling
- Mixed Membership Markov Models
- Experiments with Conversation Data
- Conclusion
Data

- Two sets of asynchronous web conversations
  - CNET forums
    - Technical help and discussion
    - Labeled with speech acts
  - Twitter
    - More personal communication
    - Short messages

<table>
<thead>
<tr>
<th># threads</th>
<th># messages</th>
<th># tokens per message</th>
</tr>
</thead>
<tbody>
<tr>
<td>321</td>
<td>1309</td>
<td>78</td>
</tr>
<tr>
<td>36K</td>
<td>100K</td>
<td>13</td>
</tr>
</tbody>
</table>


Experimental Details

- **Baselines:**
  - Bayesian Block HMM (BHMM)
  - Latent Dirichlet Allocation (LDA)

- **Symmetric Dirichlet prior on word distributions**
  - Fancy way of describing smoothing
  - Concentration parameter sampled via Metropolis-Hastings

- **$\alpha$-mean Gaussian prior on transition parameters $\lambda$**
  - Independent weights (diagonal covariance)
  - Acts as $L_2$ regularizer on weights

- **All Dirichlet hyperparameters are optimized**
  - Applies to LDA and BHMM
Thread Reconstruction

- Pretend we don’t know the thread structure of a conversation. Can we figure out which messages are in response to which?

> user1: hey
> user2: what’s up?
> user1: not much

> user1: hey
> user2: what’s up?
> user1: not much
> user2: what’s up?
> user1: hey

- Treat “parent” of each message as a hidden variable
  - Sample using simulated annealing
- Evaluate on held-out test data
  - Metric: accuracy (% of messages correctly aligned to parent)
  - Results pooled over many trials
Thread Reconstruction

- $M^4$ is a lot better than Block HMM on CNET corpus
  - Twitter messages are short, so single-class assumption is probably reasonable


Speech Act Induction

- Messages in CNET corpus are annotated with speech act labels

- 12 labels
  - **Question** (broken into subclasses)
  - **Answer** (broken into subclasses)
  - **Resolution, Reproduction, Other**

- We measured how well the latent classes induced by $M^4$ matched the human labels
  - Metric: variation of information (VI)
Speech Act Induction

M₄ is significantly better
What Does $M^4$ Learn?

- Top words from a subset of classes
- Arrows show sign of $\lambda$ from going from one class to another
Overview

- Unsupervised Content Models
- Unsupervised Conversation Modeling
- Mixed Membership Markov Models
- Experiments with Conversation Data
- Conclusion
Conclusion

- $M^4$
  - Combines properties of topic models and Markov models
  - Outperforms LDA and HMM individually

- Room for extensions
  - Richer model of *intra*-message structure
  - Bayesian formulations

- Code is available
  - [http://cs.jhu.edu/~mpaul](http://cs.jhu.edu/~mpaul)
Acknowledgements

• Advice:
  o Mark Dredze
  o Jason Eisner
  o Nick Andrews
  o Matt Gormley
  o Frank Ferraro, Wes Filardo, Adam Teichert, Tim Viera

• $$\$$:

Thank You 감사합니다
### Perplexity

<table>
<thead>
<tr>
<th># classes:</th>
<th>5</th>
<th>10</th>
<th>15</th>
<th>20</th>
<th>25</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CNET</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unigram</td>
<td>63.1</td>
<td>63.1</td>
<td>63.1</td>
<td>63.1</td>
<td>63.1</td>
</tr>
<tr>
<td>LDA</td>
<td>57.2</td>
<td>54.4</td>
<td>52.9</td>
<td>51.6</td>
<td>50.5</td>
</tr>
<tr>
<td>BHMM</td>
<td>61.3</td>
<td>61.1</td>
<td>60.9</td>
<td>60.9</td>
<td>60.9</td>
</tr>
<tr>
<td>M⁴</td>
<td>60.4</td>
<td>59.6</td>
<td>59.3</td>
<td>59.2</td>
<td>59.3</td>
</tr>
<tr>
<td><strong>Twitter</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unigram</td>
<td>93.0</td>
<td>93.0</td>
<td>93.0</td>
<td>93.0</td>
<td>93.0</td>
</tr>
<tr>
<td>LDA</td>
<td>83.7</td>
<td>78.4</td>
<td>74.0</td>
<td>70.9</td>
<td>70.2</td>
</tr>
<tr>
<td>BHMM</td>
<td>90.5</td>
<td>89.9</td>
<td>89.7</td>
<td>89.6</td>
<td>89.4</td>
</tr>
<tr>
<td>M⁴</td>
<td>88.4</td>
<td>86.2</td>
<td>85.5</td>
<td>85.6</td>
<td>86.31</td>
</tr>
</tbody>
</table>

- **M⁴ more predictive than the block HMM**