Neural Models for Documents with Metadata

Dallas Card, Chenhao Tan, Noah A. Smith
July 18, 2018
Main points of this talk:

1. Introducing *Scholar*¹: a neural model for documents with metadata
   - Background (LDA, SAGE, SLDA, etc.)
   - Model and related work
   - Experiments and Results

2. Power of neural variational inference for interactive modeling

¹Sparse Contextual Hidden and Observed Language Autoencoder
Latent Dirichlet Allocation

Types of metadata

- Date or time
- Author(s)
- Rating
- Sentiment
- Ideology
- etc.
Variations and extensions

- Author topic model (Rosen-Zvi et al, 2004)
- Supervised LDA (SLDA; McAuliffe and Blei, 2008)
- Dirichlet multinomial regression (Mimno and McCallum, 2008)
- Sparse additive generative models (SAGE; Eisenstein et al, 2011)
- Structural topic model (Roberts et al, 2014)
- ...
Desired features of model

- Fast, scalable inference.
- Easy modification by end-users.

Incorporation of metadata:
- Covariates: features which influence text (as in SAGE).
- Labels: features to be predicted along with text (as in SLDA).
- Possibility of sparse topics.
  → Use variational autoencoder (VAE) style of inference (Kingma and Welling, 2014).
Desired features of model

- Fast, scalable inference.
- Easy modification by end-users.
- Incorporation of metadata:
  - Covariates: features which influence text (as in SAGE).
  - Labels: features to be predicted along with text (as in SLDA).

→ Use variational autoencoder (VAE) style of inference (Kingma and Welling, 2014)
Desired features of model

- Fast, scalable inference.
- Easy modification by end-users.
- Incorporation of metadata:
  - Covariates: features which influence text (as in SAGE).
  - Labels: features to be predicted along with text (as in SLDA).
- Possibility of sparse topics.
Desired features of model

- Fast, scalable inference.
- Easy modification by end-users.
- Incorporation of metadata:
  - Covariates: features which influences text (as in SAGE).
  - Labels: features to be predicted along with text (as in SLDA).
- Possibility of sparse topics.
- Incorporate additional prior knowledge.
Desired features of model

- Fast, scalable inference.
- Easy modification by end-users.
- Incorporation of metadata:
  - Covariates: features which influence text (as in SAGE).
  - Labels: features to be predicted along with text (as in SLDA).
- Possibility of sparse topics.
- Incorporate additional prior knowledge.
  - Use variational autoencoder (VAE) style of inference (Kingma and Welling, 2014)
Desired outcome

- Coherent groupings of words (something like topics), with offsets for observed metadata
Desired outcome

- Coherent groupings of words (something like topics), with offsets for observed metadata
- Encoder to map from documents to latent representations
Desired outcome

- Coherent groupings of words (something like topics), with offsets for observed metadata
- Encoder to map from documents to latent representations
- Classifier to predict labels from latent representation
Model

\[ p(w | \theta_i) = f_g(\cdot) \]

\[ \theta_i \in \Delta^k \]

words
Model

The model is described by the following equation:

\[ p(w | \theta_i) = f_g(\cdot) \]

where \( \theta_i \in \Delta^k \) and \( p(w | \theta_i) \) represents the generator network for generating words.
Model

\[ p(\theta_i \mid w) \uparrow \quad q(\theta_i \mid w) \downarrow \quad \text{generator network:} \quad p(w \mid \theta_i) = f_g(\cdot) \]

\[ \theta_i \in \Delta^k \]

\[ \text{words} \]
Model

$$\text{ELBO} = \mathbb{E}_q[\log p(\text{words} | \theta_i)] - D_{KL}[q(\theta_i | \text{words}) || p(\theta_i)]$$
**Model**

encoder network:  \( q(\theta_i \mid w) = f_e(\cdot) \)

\( \theta_i \in \Delta^k \)

generator network:  \( p(w \mid \theta_i) = f_g(\cdot) \)

\[ \text{ELBO} = \mathbb{E}_q[\log p(\text{words} \mid \theta_i)] - D_{KL}[q(\theta_i \mid \text{words}) \| p(\theta_i)] \]
Model

Encoder network:  \[ q(\theta_i \mid w) = f_e(\cdot) \]

\[ r_i \in \mathbb{R}^k \]

\[ \theta_i = \text{softmax}(r_i) \]

\[ \theta_i \in \Delta^k \]

Generator network:  \[ p(w \mid \theta_i) = f_g(\cdot) \]

\[ \text{ELBO} = \mathbb{E}_q[\log p(\text{words} \mid r_i)] - D_{KL}[q(r_i \mid \text{words}) \parallel p(r_i)] \]
Model

words

encoder network: \( q(\theta_i | w) = f_e(\cdot) \)

\( r_i \in \mathbb{R}^k \)

\( \theta_i = \text{softmax}(r_i) \)

\( \theta_i \in \Delta^k \)

generator network: \( p(w | \theta_i) = f_g(\cdot) \)

words

\[
\text{ELBO} \approx \frac{1}{S} \sum_{s=1}^{S} \left[ \log p(\text{words} | r_i^{(s)}) \right] - D_{KL} [q(r_i | \text{words}) || p(r_i)]
\]
Model

\( \varepsilon \sim \mathcal{N}(0, I) \)

encoder network: \( q(\theta_i \mid w) = f_e(\cdot) \)

\( r_i \in \mathbb{R}^k \)

\( \theta_i = \text{softmax}(r_i) \)

\( \theta_i \in \Delta^k \)

generator network: \( p(w \mid \theta_i) = f_g(\cdot) \)

words

\[ \text{ELBO} \approx \frac{1}{S} \sum_{s=1}^{S} \left[ \log p(\text{words} \mid r_i^{(s)}) \right] - D_{\text{KL}}[q(r_i \mid \text{words}) \parallel p(r_i)] \]
Model

\[ \varepsilon \sim \mathcal{N}(0, I) \]

encoder network: \( q(\theta_i \mid w) = f_e(\cdot) \)

\[ r_i \in \mathbb{R}^k = \mu_q + \varepsilon^{(s)} \odot \sigma_q \]

\[ \theta_i = \text{softmax}(r_i) \]

\[ \theta_i \in \Delta^k \]

generator network: \( p(w \mid \theta_i) = f_g(\cdot) \)

\[ \text{ELBO} \approx \frac{1}{S} \sum_{s=1}^{S} \left[ \log p(\text{words} \mid r_i^{(s)}) \right] - D_{\text{KL}}[q(r_i \mid \text{words}) \| p(r_i)] \]
Model

\[ \varepsilon \sim \mathcal{N}(0, I) \]

encoder network: \[ q(\theta_i \mid w) = f_e(\cdot) \]

\[ r_i \in \mathbb{R}^k = \mu_q + \varepsilon^{(s)} \odot \sigma_q \]

\[ \theta_i = \text{softmax}(r_i) \]

\[ \theta_i \in \Delta^k \]

generator network: \[ p(w \mid \theta_i) = f_g(\cdot) \]

words

Model

\[ \varepsilon \sim \mathcal{N}(0, I) \]
\[ \text{encoder network: } q(\theta_i | w) = f_e(\cdot) \]
\[ r_i \in \mathbb{R}^k = \mu_q + \varepsilon^{(s)} \odot \sigma_q \]
\[ \theta_i = \text{softmax}(r_i) \]
\[ \theta_i \in \Delta^k \]
\[ \text{generator network: } p(w | \theta_i) = f_g(\cdot) \]

\[ y_i \quad \text{words} \]
**Model**

- **Encoder Network**: $q(\theta_i | w) = f_e(\cdot)$
- **Generator Network**: $p(w | \theta_i) = f_g(\cdot)$

Parameters:
- $\epsilon \sim \mathcal{N}(0, I)$
- $r_i \in \mathbb{R}^k = \mu_q + \epsilon^{(s)} \odot \sigma_q$
- $\theta_i = \text{softmax}(r_i)$
- $c_i$ and $y_i$
words, $c_i$, $y_i$

$\varepsilon \sim \mathcal{N}(0, I)$

encoder network: $q(\theta_i \mid w) = f_e(\cdot)$

$r_i \in \mathbb{R}^k = \mu_q + \varepsilon^{(s)} \circ \sigma_q$

$\theta_i = \text{softmax}(r_i)$

$\theta_i \in \Delta^k$

generator network: $p(w \mid \theta_i) = f_g(\cdot)$

words, $y_i$

ci, ci, yi
Generator network:

- \[ p(\text{word} \mid \theta_i, c_i) = \text{softmax}(d + \theta_i^T B^{\text{topic}} + c_i^T B^{\text{cov}}) \]
Generator network:

- \( p(\text{word} \mid \theta_i, c_i) = \text{softmax}(d + \theta_i^T B^{\text{topic}} + c_i^T B^{\text{cov}}) \)
- Optionally include interactions between topics and covariates
Generator network:

- \( p(\text{word} \mid \theta_i, c_i) = \text{softmax}(d + \theta_i^T B^{\text{topic}} + c_i^T B^{\text{cov}}) \)
- Optionally include interactions between topics and covariates
- \( p(y_i \mid \theta_i, c_i) = f_y(\theta_i, c_i) \)
Generator network:

- \( p(\text{word} \mid \theta_i, c_i) = \text{softmax}(d + \theta_i^T B^{\text{topic}} + c_i^T B^{\text{cov}}) \)
- Optionally include interactions between topics and covariates
- \( p(y_i \mid \theta_i, c_i) = f_y(\theta_i, c_i) \)

Encoder:

- \( \mu_i = f_\mu(\text{words}, c_i, y_i) \)
- \( \log \sigma_i = f_\sigma(\text{words}, c_i, y_i) \)
- Optional incorporation of word vectors to embed input
Optimization

- Stochastic optimization using mini-batches of documents
- Tricks from Srivastava and Sutton, 2017:
  - Adam optimizer with high-learning rate to bypass mode collapse
  - Batch-norm layers to avoid divergence
- Annealing away from batch-norm output to keep results interpretable
\textbf{Output of Scholar}

- $B^{(\text{topic})}$, $B^{(\text{cov})}$: Coherent groupings of positive and negative deviations from background ($\sim$ topics)
B^{topic}, B^{cov}: Coherent groupings of positive and negative deviations from background (∼ topics)

f_μ, f_σ: Encoder network: mapping from words to topics:
\[ \hat{\theta}_i = \text{softmax}(f_e(\text{words}, c_i, y_i, \epsilon)) \]
Output of Scholar

- $B^{(\text{topic})}, B^{(\text{cov})}$: Coherent groupings of positive and negative deviations from background ($\sim$ topics)
- $f_\mu, f_\sigma$: Encoder network: mapping from words to topics: 
  $\hat{\theta}_i = \text{softmax}(f_e(\text{words}, c_i, y_i, \epsilon))$
- $f_y$: Classifier mapping from $\hat{\theta}_i$ to labels: $\hat{y} = f_y(\theta_i, c_i)$
1. Performance as a topic model, without metadata (perplexity, coherence)
2. Performance as a classifier, compared to SLDA
3. Exploratory data analysis
Quantitative results: basic model

IMDB dataset (Maas, 2011)
Quantitative results: basic model

IMDB dataset (Maas, 2011)
Quantitative results: basic model

IMDB dataset (Maas, 2011)
Quantitative results: basic model

IMDB dataset (Maas, 2011)
Quantitative results: basic model

IMDB dataset (Maas, 2011)
Quantitative results: basic model

IMDB dataset (Maas, 2011)
Classification results

IMDB dataset (Maas, 2011)
Data: Media Frames Corpus (Card et al, 2015)

- Collection of thousands of news articles annotated in terms of tone and framing
- Relevant metadata: year of publication, newspaper, etc.
Tone as a label

$p(\text{pro-immigration} \mid \text{topic})$

- english language
- city
- spanish community
- boat
- desert
- died
- men
- miles
- coast
- haitian
- visas
- visa applications
- students
- citizenship
- asylum
- judge
- appeals
- deportation
- court
- labor
- jobs
- workers
- percent
- study
- wages
- bush
- border
- president
- bill
- republicans
- state
- gov
- benefits
- arizona law
- bill
- bills
- arrested
- charged
- charges
- agents
- operation
### Tone as a covariate, with interactions

<table>
<thead>
<tr>
<th>Base topics</th>
<th>Anti-immigration</th>
<th>Pro-immigration</th>
</tr>
</thead>
<tbody>
<tr>
<td>ice customs agency</td>
<td><strong>criminal</strong> customs</td>
<td><strong>detainees</strong> detention</td>
</tr>
<tr>
<td>population born percent</td>
<td>jobs million <strong>illegals</strong></td>
<td>english <strong>newcomers</strong></td>
</tr>
<tr>
<td>judge case court guilty</td>
<td><strong>guilty</strong> charges man</td>
<td>asylum <strong>court judge</strong></td>
</tr>
<tr>
<td>patrol border miles</td>
<td>patrol border</td>
<td>died authorities desert</td>
</tr>
<tr>
<td>licenses drivers card</td>
<td>foreign sept visas</td>
<td>green citizenship card</td>
</tr>
<tr>
<td>island story chinese</td>
<td>smuggling federal</td>
<td>island school ellis</td>
</tr>
<tr>
<td>guest worker workers</td>
<td>bill border house</td>
<td>workers tech skilled</td>
</tr>
<tr>
<td>benefits bill welfare</td>
<td>republican california</td>
<td>law welfare students</td>
</tr>
</tbody>
</table>
Conclusions

- Variational autoencoders (VAEs) provide a powerful framework for latent variable modeling.
- We use the VAE framework to create a customizable model for documents with metadata.
- We obtain comparable performance with enhanced flexibility and scalability.
- Code is available: www.github.com/dallascard/scholar