Abstract

Existing topic models do not capture the fundamental tendency of words to appear in bursts; once a word has appeared in a document, it is more likely to be used again. We introduce the DCMLDA model, a topic model that uses Dirichlet compound multinomial distributions to capture this burstiness. On both text and financial datasets, the new model achieves better held-out log-likelihoods with fewer topics than standard latent Dirichlet allocation (LDA).

Burlessness

Additional appearances of a word are less surprising than the first:

- Burstiness endemic in text and some non-text data
- Heavy-tailed distribution than multinomials

Generative Process

For each topic $k$:
1. Draw a topic multinomial $\phi_k \sim \text{Dirichlet}(\beta)$
2. For each document $d$:
   a. Draw topic distribution $\theta_d \sim \text{Dirichlet}(\alpha)$
   b. Draw a word $w_i \sim \text{Multinomial}(\phi_{\theta_d})$
Using LDA Training finds maximum-likelihood values for $\theta$ and $\phi$. $\phi$ vectors define topics as probabilities of each word in the topic. $\theta$ vectors are useful for classifying documents and measuring similarities between documents. [1]

Dirichlet Compound Multinomial

Generative Process

For each document $d$:
1. Draw multinomial $\phi_d \sim \text{Dirichlet}(\beta)$
2. Draw $N_d$ words $w_i \sim \text{Multinomial}(\phi_d)$

Document-specific topics Each document has its own document-specific multinomial $\phi_d$ drawn from a single shared high-level topic $\beta$.

Capturing burstiness DCM $\beta$ vector has one more degree of freedom than LDA $\phi$ vectors, allowing DCM to adjust for burstiness. [5]

The DCMLDA Model

Goal: Allow multiple topics in a single document like LDA while keeping topics document-specific to account for burstiness like DCM.

Generative Process

For each document $d$:
1. Draw topic distribution $\theta_d \sim \text{Dirichlet}(\alpha)$
2. For each topic $k$:
   a. Draw a topic multinomial $\phi_{\theta_d} \sim \text{Dirichlet}(\beta_k)$
3. For each of the $N_d$ words:
   a. Draw a word $w_i \sim \text{Multinomial}(\phi_{\theta_d})$

DCMLDA Graphical Model

Note that $\phi$ is re-drawn for each document in DCMLDA.

DCMLDA Training

Two things to learn
1. What words are from what topics?
2. What are the values of $\alpha$ and $\beta$?

Training process

Start with initial values of $\alpha$, $\beta$.
Repeat until convergence of $\beta$:
- Gibbs sample topics to steady-state
- Choose a single topic assignment vector $\vec{z}$
- Choose $\alpha$, $\beta$ to maximize corpus likelihood

Gibbs Sampling

Like LDA, DCMLDA topics learned through collapsed Gibbs sampling of $p(z_i|z_{-i}, w) = (n_{z_i, t} + \alpha_z - 1) (n_{w, z_i, t} + \beta_{z_i, w} - 1) / (\sum_k n_{z_i,k} + \alpha_z - 1) / (\sum_{z_i} n_{w,z_i} + \beta_{z_i,w} - 1)$

Gibbs sampling yields vectors of probable topic assignments for each word, given $\alpha$ and $\beta$.

Maximizing $\alpha$ and $\beta$

Given $\vec{z}$ from Gibbs sampling, choose $\alpha$, $\beta$ to maximize $p(w, \vec{z}|\alpha, \beta)$.

$$
\alpha' = \arg\max \sum_k \log \Gamma(n_{k,d}+\alpha_k) - \log \Gamma(\alpha_k)
+ \sum_d \log \Gamma(\sum_k \alpha_k) - \log \Gamma(\sum_k n_{k,d}+\alpha_k)
$$

$$
\beta' = \arg\max \sum_k \log \Gamma(n_{t,k}+\beta_k) - \log \Gamma(\beta_k)
+ \sum_t \log \Gamma(\sum_k \beta_k) - \log \Gamma(\sum_k n_{t,k}+\beta_k)
$$

Current Matlab implementation, using L-BFGS takes ~100 seconds on sample datasets.

Experimental Design

Goal: Check if DCMLDA’s handling of burstiness makes a better model than LDA

Why compare to LDA?
- LDA and DCMLDA are of comparable conceptual complexity
- DCMLDA does not compete with more complex models, as they can be modified to use DCMLDA topics

Conclusions

- A DCMLDA model with few topics is comparable to a LDA model with many topics.
- Optimal learned $\alpha$, $\beta$ values significantly different from common heuristic values.
- DCMLDA $\beta$ topics are as interpretable as LDA $\phi$ topics on these datasets.
- Burstiness is an important phenomenon to capture in topic modeling text and some non-text data.