Authorship Attribution with Author-aware Topic Models

Disjoint Author-Document Topic Model
Seroussi, Zukerman, Bohnert 2012

Thomas Rometsch

September 25, 2015
Sommerakademie 2015 La Colle-sur-Loup

Outline

1. The Disjoint Author-Document Topic model
2. Experiments in paper
3. Results of the paper
4. Reproducibility and implementation

DADT

- **D**isjoint **A**uthor-**D**ocument **T**opic Model
- Statistical model $\rightarrow$ distributions over topic and words.
- Generative topic model. Model is defined by the way how a document is created.
- Topics distributions are used for authorship attribution.

How to write a document

Write a text with possibly multiple authors by choosing words from distributions.

- prior believes $\rightarrow$ topic and word distributions

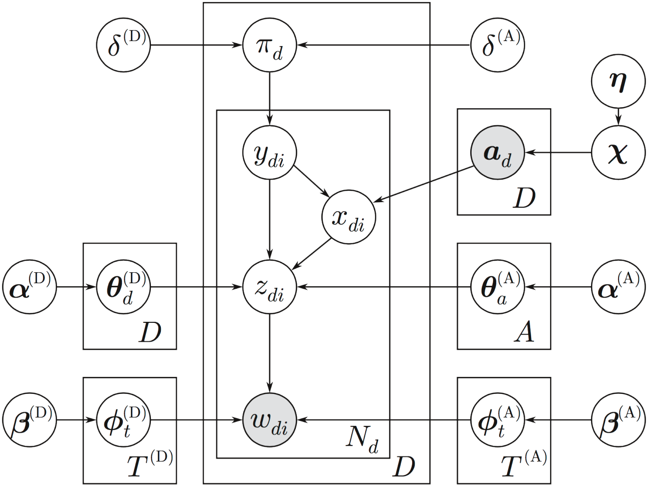
Now repeat

- draw author
- draw indicator: author or document topic
- draw $\begin{cases} \text{author} \\ \text{document} \end{cases}$ topic
- draw word from $\begin{cases} \text{topic word distribution of author} \\ \text{topic word distribution of document} \end{cases}$

Model variables

- Distributions over
  - authors χ
  - topics for every author $\theta^{(A)}$
  - topics for every document $\theta^{(D)}$
  - words for every topic and document $\phi_t^{(D)}$
  - words for every topic and author $\phi_t^{(A)}$
- and prior believes (draw distributions from Dirichlet $\rightarrow$ smoothing). Different prior values for **stopwords.**
- Observables:
  - words
  - authors (in training phase)

DADT in plate notation



Model Inference

Task: Given a text corpus, find the distributions.

- Gibbs sampler (Metropolis-Hasting)
- Markov Chain Monte Carlo (MCMC) algorithm
- markov chain with transition probabilities according to the statistical model
  - distributions integrated out
  - all document variables
  - word / topic counters
- iterate until convergence (often enough)
- calculate distributions with counters

Attribute authorship

For every text with unknown authorship.

- Assume text was written by previously unknown author and infer topic distributions.
- DADT-SVM: concatenate author and document topic distributions and use as input for SVM
- DADT-P: Probabilistic measure according to model. Calculate most probable author using

$$\underset{a \in \{1,\ldots,A\}}{\arg\max}\; p\left(\tilde{a} = a \middle| \tilde{\boldsymbol{w}}, \tilde{\pi}, \tilde{\boldsymbol{\theta}}^{(D)}, \boldsymbol{\theta}_a^{(A)}, \boldsymbol{\Phi}^{(D)}, \boldsymbol{\Phi}^{(A)}, \chi_a\right) \propto$$

$$\underset{a \in \{1,\ldots,A\}}{\arg\max}\; \chi_a \prod_{i=1}^{\tilde{N}} \left(\tilde{\pi} \sum_{t=1}^{T^{(A)}} \theta_{at}^{(A)} \phi_{t\tilde{w}_i}^{(A)} + (1 - \tilde{\pi}) \sum_{t=1}^{T^{(D)}} \tilde{\theta}_t^{(D)} \phi_{t\tilde{w}_i}^{(D)} \right)$$

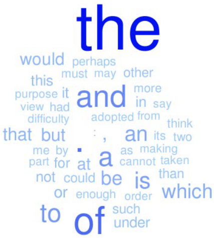
Experimental setup

- closed-class author attribution
- datasets:
  - **PAN11** with 72 authors: emails
  - **Blog** with 19 320 authors
- symmetric topic priors
- asymmetric word priors
  stopwords more likely to be in author topics $\leftrightarrow$ indicator of style
- compare to LDA and AT methods

Word clouds

Example of topics created with DADT.

document topic



author topic

Word clouds 2

document topic

author topic

Results of the paper

Comparison against AT and LDA (token frequency SVM - one vs. all, fa = fictitious author)

Method	PAN'11 Validation	PAN'11 Testing	Blog Prolific	Blog Full
SVM	48.61%	53.31%	33.31%	24.13%
LDA-H	34.95%	42.62%	21.61%	7.94%
AT	46.68%	53.08%	37.56%	23.03%
AT-FA	20.68%	24.23%	—	—
DADT	**54.24%**	**59.08%**	**42.51%**	**27.63%**

- DADT is better than more naive methods on short and noisy texts. Attributed to different types of topics.

Reproducibility

- Assigned paper not enough for non-expert.
- More detailed paper by Seroussi about DADT available.
- Overall detailed description.
- Datasets easily available and sources cited, but two documents excluded from PAN11 without giving IDs.
- Some minor details missing
  - tokenizer, preprocessing
  - handling of extra word in test texts
  - averaging procedure in testing phase

Implementation

Parts already implemented:

- Database import in Python, index, store to HDF5
- Gibbs sampler for model inference in C++

Challenges:

- PAN11 dataset only pseudo XML (& character) $\rightarrow$ own parser
- naive MCMC implementation with Python
  $\rightarrow \sim$ 100 days expected runtime for PAN11
- use C++ $\rightarrow$ no significant improvement
- be smarter
  $\rightarrow \sim$ 20 hours expected runtime for PAN11