Optimization of ARTM regularization coefficients via stochastic variational inference

Michael Khalman, Dmitry Vetrov, Michael Figurnov

November 13, 2015

Outline

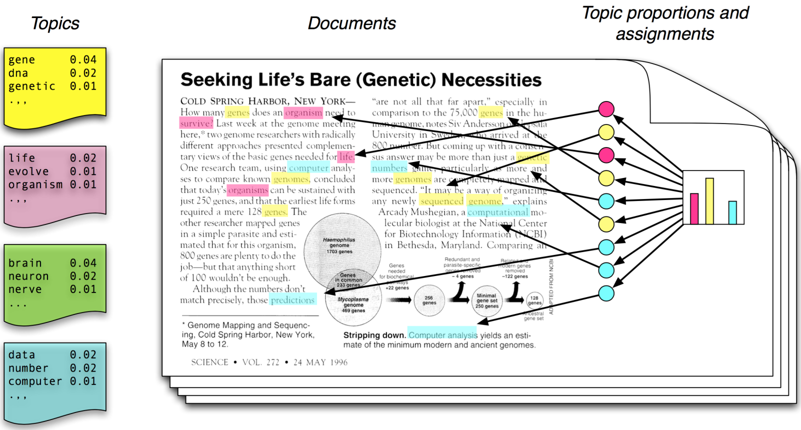
Topic modeling

Topic modeling an algorithmic tool that help us organize, search and understand vast amount of information.

Applications

- Exploratory search
- Clustering
- Creating annotations
- Recomendation systems
- Looking for similar documents in a huge collection
- ...

Probabilistic Topic Modeling



Probabilistic Topic Modeling

Consider a set of documents D, dictionary W. Each document d consists of some words $w \in W$.

- There's a set of latent topics T. Each $t \in T$ is a distribution over W: $\varphi_{wt} = p(w|t)$
- Each document $d \in D$ has specific distribution over T: $\theta_{td} = p(t|d)$.
- Bag of words assumption: order of words in d doesn't matter. Words in d are i.i.d. from $p(w|d)$
- Conditional independence assumption: $p(w|d, t) = p(w|t)$.
- Law of total probability: $p(w|d) = \sum_t \underbrace{p(w|t)}_{\varphi_{wt}} \underbrace{p(t|d)}_{\theta_{td}}$

ARTM

- $N = (n_{dw})_{W \times D}$ — matrix of counts of words in documents
- $F = (n_{dw}/n_d)_{W \times D}$ — normalized matrix of counts; $n_d \equiv \sum_w n_{dw}$.
- $(\varphi_{wt})_{W \times T} \equiv \Phi \in \mathbb{R}^{W \times T}$ — topic-specific distribution over words.
- $(\theta_{td})_{T \times D} \equiv \Theta \in \mathbb{R}^{T \times D}$ — document-specific distribution over topics.

$$p(w|d) = \sum_t \varphi_{wt} \theta_{td}$$

Maximization of log-likelihood:

$$p(N|\Phi, \Theta) = \prod_d \prod_w (\sum_t \varphi_{wt} \theta_{td})^{n_{dw}} \rightarrow \max_{\Phi, \Theta}$$

ARTM

- $N = (n_{dw})_{W \times D}$ — matrix of counts of words in documents
- $F = (n_{dw}/n_d)_{W \times D}$ — normalized matrix of counts; $n_d \equiv \sum_w n_{dw}$.
- $(\varphi_{wt})_{W \times T} \equiv \Phi \in \mathbb{R}^{W \times T}$ — topic-specific distribution over words.
- $(\theta_{td})_{T \times D} \equiv \Theta \in \mathbb{R}^{T \times D}$ — document-specific distribution over topics.

$$p(w|d) = \sum_t \varphi_{wt}\theta_{td}$$

Maximization of log-likelihood:

$$p(N|\Phi, \Theta) = \prod_d \prod_w (\sum_t \varphi_{wt}\theta_{td})^{n_{dw}} \to \max_{\Phi,\Theta}$$

$$\log p(N|\Phi, \Theta) = \sum_d \sum_w n_{dw} \log \sum_t \varphi_{wt}\theta_{td} \to \max_{\Phi,\Theta}$$

ARTM

- $N = (n_{dw})_{W \times D}$ — matrix of counts of words in documents
- $F = (n_{dw}/n_d)_{W \times D}$ — normalized matrix of counts; $n_d \equiv \sum_w n_{dw}$.
- $(\varphi_{wt})_{W \times T} \equiv \Phi \in \mathbb{R}^{W \times T}$ — topic-specific distribution over words.
- $(\theta_{td})_{T \times D} \equiv \Theta \in \mathbb{R}^{T \times D}$ — document-specific distribution over topics.

$$p(w|d) = \sum_t \varphi_{wt}\theta_{td}$$

Maximization of log-likelihood:

$$p(N|\Phi,\Theta) = \prod_d \prod_w (\sum_t \varphi_{wt}\theta_{td})^{n_{dw}} \to \max_{\Phi,\Theta}$$

$$\log p(N|\Phi,\Theta) = \sum_d \sum_w n_{dw} \log \sum_t \varphi_{wt}\theta_{td} \to \max_{\Phi,\Theta}$$

$$\log p(N|\Phi,\Theta) = -\sum_d n_d \,\mathrm{KL}\left(F_d \,||\, (\Phi\Theta)_d\right) \to \max_{\Phi,\Theta}$$

ARTM

Non-negative matrix factorization: $F \approx \Phi\Theta$:

$$\mathcal{L}(\Phi, \Theta) \equiv \log p(N|\Phi, \Theta) \to \max_{\Phi,\Theta}$$

$$\mathcal{L}(\Phi, \Theta) = -\sum_d n_d \, \mathrm{KL}\left(F_d \,||\, (\Phi\Theta)_d\right)$$

Ill-posed problem: $F = \Phi\Theta = (\Phi S)(S^{-1})\Theta)$
Prior on Φ, Θ:

$$\hat{p}(\Phi, \Theta|\boldsymbol{\tau}) \propto \exp(\sum_i \tau_i R_i(\Phi, \Theta))$$

$$p(N, \Phi, \Theta|\boldsymbol{\tau}) = p(N|\Phi, \Theta)\hat{p}(\Phi, \Theta|\boldsymbol{\tau}) \to \max_{\Phi,\Theta}$$

ARTM

Regularized non-negative matrix factorization: $F \approx \Phi\Theta$:

$$\underbrace{\mathcal{L}(\Phi,\Theta)}_{\text{log-likelihood}} + \underbrace{\boldsymbol{R}(\Phi,\Theta)}_{\text{regularization}} \to \max_{\Phi,\Theta}$$

$$\mathcal{L}(\Phi,\Theta) = -\sum_d n_d \,\mathrm{KL}\left(F_d \,||\, (\Phi\Theta)_d\right)$$

$$\boldsymbol{R}(\Phi,\Theta) = \sum_i \tau_i R_i(\Phi,\Theta)$$

$R_i(\Phi,\Theta)$ — any differentiable regularization function

ARTM — regularizers

We can use $R(\Phi, \Theta)$ to:

- Make model better (easy to interpret, coherent topics)
  - Smoothing/sparsing (LDA)
    $R = \sum_{t \in T} \sum_{w \in W} \beta_{wt} \log \varphi_{wt} + \sum_{d \in D} \sum_{t \in T} \alpha_{td} \log \theta_{td}$
  - Decorellation: $R = -\tau \sum_{t \in T} \sum_{s \in T \setminus t} \sum_{w \in W} \varphi_{ws} \varphi_{ws}$
- Add soft restrictions to the model
  - Semi-supervised regularizer
    $R = \sum_{t \in T_0} \sum_{w \in W_t} \beta_{wt} \log \varphi_{wt} + \sum_{d \in D_0} \sum_{t \in T_d} \alpha_{td} \log \theta_{td}$
- Add new information to the model
  - Author-topic models
- Make the model specified for solving specific problem
  - Topic models with time
  - Topic models for classification
- ...

ARTM — advantages

- **Really** easy to infer new models
- There is an iterative algorithm for any differentiable $R(\Phi, \Theta)$ based on fixed-point iteration method for stationarity conditions
- You can combine any amount of regularizers in one model

ARTM — problems

How to fit hyperparamters (regularization coefficients $\boldsymbol{\tau}$)?

Current approach: heuristic, manual fitting.
How to fit them **automatically** is an **open problem**

- Too many parameters for grid-search
- We don't know what to optimize

Maximum-evidence approach

$$p(N|\boldsymbol{\tau}) = \int p(N|\Phi, \Theta)p(\Phi, \Theta|\boldsymbol{\tau})d\Phi d\Theta \longrightarrow \max_{\tau}$$

Or, if we denote (Φ, Θ) as $\boldsymbol{z}$, and N as $\boldsymbol{x}$

$$p(\boldsymbol{x}|\boldsymbol{\tau}) = \int p(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{z}|\boldsymbol{\tau})d\boldsymbol{z} = \mathbb{E}_{\boldsymbol{z}|\boldsymbol{\tau}}p(\boldsymbol{x}|\boldsymbol{z}) \longrightarrow \max_{\tau}$$

We'll use variational inference to optimize it.

Variational inference

Recall our probabilistic model:

$$p(\boldsymbol{x}|\boldsymbol{z}) \equiv p(N|\Phi, \Theta) = \prod_d \prod_w (\sum_t \varphi_{wt}\theta_{td})^{n_{dw}}$$

$$p(\boldsymbol{z}|\boldsymbol{\tau}) \equiv p(\Phi, \Theta|\boldsymbol{\tau}) \propto \exp(\sum_i \tau_i R_i(\Phi, \Theta))$$

We introduce some family of distributions $q(\boldsymbol{z}|\boldsymbol{\lambda}) \approx p(\boldsymbol{z}|\boldsymbol{x})$

$$\log p(\boldsymbol{x}|\boldsymbol{\tau}) \geq \mathcal{F}(\boldsymbol{\lambda}, \boldsymbol{\tau}) \equiv \int q(\boldsymbol{z}|\boldsymbol{\lambda}) \log \frac{p(\boldsymbol{x}, \boldsymbol{z}|\boldsymbol{\tau})}{q(\boldsymbol{z}|\boldsymbol{\lambda})} d\boldsymbol{z} =$$

$$= \mathbb{E}_{q_\lambda} \log p(\boldsymbol{x}|\boldsymbol{z}) - \mathbb{E}_{q_\lambda} \log \frac{q(\boldsymbol{z}|\boldsymbol{\lambda})}{\hat{p}(\boldsymbol{z}|\boldsymbol{\tau})} - \log Z(\boldsymbol{\tau}) \to \max_{\boldsymbol{\tau}, \boldsymbol{\lambda}}$$

$\mathcal{F}(\boldsymbol{\lambda}, \boldsymbol{\tau})$ — ELBO (evidence lower-bound)
Variational inference can be used to optimize evidence over hyperparamters of the model.

Reparametrization trick

Stochastic optimization problem:

$$\begin{aligned} \boldsymbol{z} &\sim p(\boldsymbol{z}) \\ \mathbb{E}_z f(\boldsymbol{z}, \boldsymbol{\lambda}) &\to \max_{\boldsymbol{\lambda}} \\ \nabla_{\boldsymbol{\lambda}} \mathbb{E}_z f(\boldsymbol{z}, \boldsymbol{\lambda}) &= ? \end{aligned}$$

$$\nabla_{\boldsymbol{\lambda}} \mathbb{E}_z f(\boldsymbol{z}, \boldsymbol{\lambda}) = \mathbb{E}_z \nabla_{\boldsymbol{\lambda}} f(\boldsymbol{z}, \boldsymbol{\lambda}) \approx \frac{1}{m} \sum_{i=1}^{m} \nabla_{\boldsymbol{\lambda}} f(\boldsymbol{z_i}, \boldsymbol{\lambda})$$

We can use stochastic gradient descent

Reparametrization trick

What if probability $p(\boldsymbol{z})$ depends on $\boldsymbol{\lambda}$?

$$\boldsymbol{z} \sim p(\boldsymbol{z}|\boldsymbol{\lambda}) \equiv p_\lambda(\boldsymbol{z})$$
$$\mathbb{E}_{p_\lambda} f(\boldsymbol{z}) \to \max_{\boldsymbol{\lambda}}$$
$$\nabla_{\boldsymbol{\lambda}} \mathbb{E}_z f(\boldsymbol{z}) \text{ — ?}$$

Reparametrization trick

Let's assume that if $\boldsymbol{z} \sim q(\boldsymbol{z}|\boldsymbol{\lambda}) \equiv q_\lambda$,
$\boldsymbol{z} = g(\boldsymbol{\varepsilon}, \boldsymbol{\lambda})$, where $\varepsilon \sim q_0$

$$\nabla_\lambda \mathbb{E}_{q_\lambda} f(\boldsymbol{z}) = \mathbb{E}_{q_0} \nabla_\lambda f(g(\boldsymbol{\varepsilon}, \boldsymbol{\lambda})) \approx \frac{1}{m} \sum_{i=1}^{m} \nabla_\lambda f(g(\boldsymbol{\varepsilon}_i, \boldsymbol{\lambda}))$$

Example:

$$\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}); \quad \boldsymbol{\lambda} \equiv (\boldsymbol{\mu}, \boldsymbol{\Sigma})$$
$$g(\boldsymbol{\varepsilon}, \boldsymbol{\lambda}) = \boldsymbol{\mu} + \boldsymbol{\Sigma}^{1/2} \boldsymbol{\varepsilon}; \quad \varepsilon \sim \mathcal{N}(0, \mathrm{I})$$
$$\nabla_\lambda \mathbb{E}_{q_\lambda} f(\boldsymbol{z}) = \mathbb{E}_{q_0} f(\boldsymbol{\mu} + \boldsymbol{\Sigma}^{1/2} \boldsymbol{\varepsilon})$$

Estimation has low variance.

Gradients of ELBO

Reparametrization trick for $q(\boldsymbol{z}|\boldsymbol{\lambda})$:

$$\boldsymbol{z} = g(\varepsilon, \boldsymbol{\lambda}); \quad \varepsilon \sim q_0$$

$$\begin{aligned}
\mathcal{F}(\boldsymbol{\lambda}, \boldsymbol{\tau}) &= \mathbb{E}_{q_\lambda} \log p(\boldsymbol{x}|\boldsymbol{z}) - \mathbb{E}_{q_\lambda} \log \frac{q(\boldsymbol{z}|\boldsymbol{\lambda})}{\hat{p}(\boldsymbol{z}|\boldsymbol{\tau})} - \log Z(\boldsymbol{\tau}) \\
\frac{\partial \mathcal{F}}{\partial \boldsymbol{\lambda}} &= \mathbb{E}_{q_0} \nabla_{\boldsymbol{\lambda}} \log p(\boldsymbol{x}|g(\varepsilon, \boldsymbol{\lambda})) - \mathbb{E}_{q_0} \nabla_{\boldsymbol{\lambda}} \log q(g(\varepsilon, \boldsymbol{\lambda})|\boldsymbol{\lambda}) \\
&+ \mathbb{E}_{q_0} \nabla_{\boldsymbol{\lambda}} \log \hat{p}(g(\varepsilon, \boldsymbol{\lambda})|\boldsymbol{\tau}) \\
\frac{\partial \mathcal{F}}{\partial \boldsymbol{\tau}} &= \mathbb{E}_{q_0} \nabla_{\boldsymbol{\tau}} \log \hat{p}(g(\varepsilon, \boldsymbol{\lambda})|\boldsymbol{\tau}) - \mathbb{E}_{q_0} \nabla_{\boldsymbol{\tau}} \log Z(\boldsymbol{\tau})
\end{aligned} \tag{1}$$

Algorithm

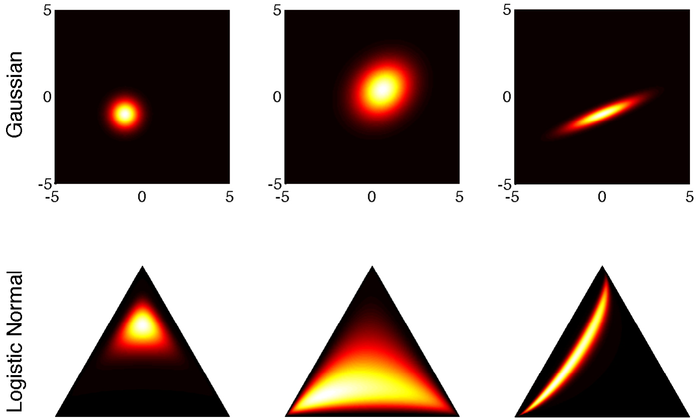
Algorithm 1 Sketch

Data: Flow of data $\boldsymbol{x}$; learning rate κ
Result: $\boldsymbol{\lambda}$, $\boldsymbol{\tau}$
  initialize $\boldsymbol{\lambda}$ and $\boldsymbol{\tau}$
  repeat
    $\boldsymbol{x} \leftarrow$ {Get mini-batch}
    Sample $\boldsymbol{\varepsilon} \leftarrow q_0$
    Compute stochastic estimations of $\frac{\partial \mathcal{F}}{\partial \boldsymbol{\tau}}$ and $\frac{\partial \mathcal{F}}{\partial \boldsymbol{\lambda}}$ according to (1)
    $\boldsymbol{\tau} = \boldsymbol{\tau} - \kappa \frac{\partial \mathcal{F}}{\partial \boldsymbol{\tau}}$ // *or* any other stochastic-gradient-kind
    $\boldsymbol{\lambda} = \boldsymbol{\lambda} - \kappa \frac{\partial \mathcal{F}}{\partial \boldsymbol{\lambda}}$ // *optimization* algorithm, e.g. AdaGrad
  until convergence

Logistic-normal distribution

$$\varepsilon \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}); \quad \boldsymbol{z} = \mathcal{P}(\varepsilon) \implies \boldsymbol{z} \sim \mathcal{P}(\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})),$$

where $\mathcal{P}(\boldsymbol{x}) = \boldsymbol{z} \implies \boldsymbol{z}_i = \frac{\exp(x_i)}{\sum_i \exp(x_i)}$ — softmax function



Logistic-normal distribution — Properties

- Distribution over simplex $\mathcal{S}_x = \{x : x_i \geq 0; \sum_i x_i = 1\}$;
- Similar to Dirichilet, but **never** exactly the same;
- Although can be approximated with Dirichilet with large α
- No analytical solution for mode/mean/variance.
- We can write down component-wise median: $\mathcal{P}(\boldsymbol{\mu})$
- Easy to sample from

Reparametrization of q

$$\boldsymbol{z} = g(\varepsilon, \boldsymbol{\lambda}) = \mathcal{P}(\mathrm{diag}(\boldsymbol{\sigma})\varepsilon + \boldsymbol{\mu}); \quad \varepsilon \sim \mathcal{N}(0, \mathrm{I})$$
$$\boldsymbol{z} \sim \mathcal{P}(\mathcal{N}(\boldsymbol{\mu}, \mathrm{diag}(\boldsymbol{\sigma})))$$

where $\boldsymbol{\lambda}$ denotes $(\boldsymbol{\mu}, \boldsymbol{\sigma})$.
Individual vector of parameters $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ for each column of Φ and Θ

$$\log q(g(\varepsilon, \boldsymbol{\lambda})) = \log q_0(\varepsilon) + \log \left| \frac{\partial g}{\partial \varepsilon} \right| = \log q_0(\varepsilon) + \sum_k (\log z_k + \log \sigma_k)$$

Recall gradients of ELBO

$$
\begin{aligned}
\boldsymbol{z} &= g(\varepsilon, \boldsymbol{\lambda}); \quad \varepsilon \sim \mathcal{N}(0, \mathrm{I}) \\
\frac{\partial \mathcal{F}}{\partial \boldsymbol{\lambda}} &= \mathbb{E}_{q_0} \nabla_{\boldsymbol{\lambda}} \log p(\boldsymbol{x}|\boldsymbol{z}) - \mathbb{E}_{q_0} \nabla_{\boldsymbol{\lambda}} \log q(\boldsymbol{z}|\boldsymbol{\lambda}) + \mathbb{E}_{q_0} \nabla_{\boldsymbol{\lambda}} \log \hat{p}(\boldsymbol{z}|\boldsymbol{\tau}) \\
\frac{\partial \mathcal{F}}{\partial \boldsymbol{\tau}} &= \mathbb{E}_{q_0} \nabla_{\boldsymbol{\tau}} \log \hat{p}(\boldsymbol{z}|\boldsymbol{\tau}) - \mathbb{E}_{q_0} \nabla_{\boldsymbol{\tau}} \log Z(\boldsymbol{\tau})
\end{aligned} \tag{2}
$$

- $\log p(\boldsymbol{x}|\boldsymbol{z})$ — log-likelihood of our model
- $\log \hat{p}(\boldsymbol{z}|\boldsymbol{\tau}) = \sum_i \tau_i R_i(\boldsymbol{z})$ — regularization term.
- $\mathbb{E}_{q_0} \nabla_{\boldsymbol{\lambda}} \log q(\boldsymbol{z}|\boldsymbol{\lambda}) = \mathbb{E}_{q_0} \nabla_{\boldsymbol{\lambda}} \sum_k (\log z_k + \log \sigma_k) + \mathrm{const}$
- $\nabla_{\boldsymbol{\lambda}} \log \hat{p}(\boldsymbol{z}|\boldsymbol{\tau}) = \sum_i \tau_i \nabla_\lambda R_i(\boldsymbol{z})$
- $\mathbb{E}_{q_0} \nabla_{\boldsymbol{\tau}} \log \hat{p}(\boldsymbol{z}|\boldsymbol{\tau}) = \mathbb{E}_{q_0} \boldsymbol{R}(\boldsymbol{z})$
- $\nabla_\tau \log Z(\boldsymbol{\tau}) = \mathbb{E}_{p(\boldsymbol{z}|\boldsymbol{\tau})} \boldsymbol{R}(\boldsymbol{z})$

How to optimize with $Z(\boldsymbol{\tau})$ unknown?

$$\nabla_\tau \log Z(\boldsymbol{\tau}) = \mathbb{E}_{p(\boldsymbol{z}|\boldsymbol{\tau})} R(\boldsymbol{z})$$

We need samples from $p(\boldsymbol{z}|\boldsymbol{\tau})$

- Contrastive divergence
  - After sampling $\boldsymbol{z}_i \sim q_\lambda$ do **a few** iterations of MCMC (starting from $\boldsymbol{z}_i$) to obtain samples from $p(\boldsymbol{z}|\boldsymbol{\tau})$.
- Importance sampling
  - $Z(\boldsymbol{\tau}) = \mathbb{E}_{q_0} \frac{\hat{p}(\boldsymbol{z}|\boldsymbol{\tau})}{q_0(\boldsymbol{z})}$
  - $\mathbb{E}_{p(\boldsymbol{z}|\boldsymbol{\tau})} R(\boldsymbol{z}) = \frac{1}{Z(\boldsymbol{\tau})} \mathbb{E}_{q_0} \frac{\boldsymbol{R}(\boldsymbol{z})\hat{p}(\boldsymbol{z}|\boldsymbol{\tau})}{q_0(\boldsymbol{z})}$
  - We use the same samples $\boldsymbol{z}_i \sim q_\lambda$ to estimate $Z(\boldsymbol{\tau})$ and $\mathbb{E}_{q_0} \frac{\boldsymbol{R}(\boldsymbol{z})\hat{p}(\boldsymbol{z}|\boldsymbol{\tau})}{q_0(\boldsymbol{z})}$

Review

Our algorithm

- Our algorithm is scalable on data size
- Time of each iteration on one batch of D docs with N total words is $O(N + TD + TW)$ (no DW term)
- ...or $O(N + T^2D + TW)$ with decorrelation
- Returns both fitted regularization coeffitients $\boldsymbol{\tau}$ and ~~matricies (Φ, Θ)~~ whole distribution over (Φ, Θ).
- Coefficients τ and distributions (Φ, Θ) are fitted **simultaneously** as long as algorithms iterates over data (over batches)

Questions still not answered

- What to do with that distribuition (what is the final answer)
  - Sample from it (probably multiple times)
  - Return median $\mathcal{P}(\mu)$
  - Compute mode $\text{argmax}_{\Phi,\Theta}\, q(\Phi, \Theta|\boldsymbol{\lambda})$
- Is maximum-evidence approach really going to work in this scenario?
- How to organize online-learning process carefully?

Thank you for your attention!