Latent Dirichlet allocation

Jérôme DOCKÈS (jerome@dockes.org)
Pascal LU (pascal.lu@centraliens.net)
École Normale Supérieure de Cachan

We consider the problem of modeling text corpora. The goal is to find short descriptions of the members of a collection that enable efficient processing of large collections while preserving the essential statistical relationships. Our work is mainly based on [BNJ03].

1 Latent Dirichlet allocation

1.1 Presentation of the model

Notations

- $\mathcal{D} = \{d_1, d_2, \ldots, d_M\}$ is a corpus (collection of $M = |\mathcal{D}|$ documents). We denote $|\mathcal{D}|$ (`num_docs`) the number of documents.
- $\mathcal{V}$ is the vocabulary. Its size is denoted V (`voc_size`).
- The number of topics is denoted k (`num_topics`).

For a document $d \in \mathcal{D}$,

- $d = (w_1^{(d)}, \ldots, w_{N_d}^{(d)})$ represents the document d. N_d is the number of words in the document d.
- $w^{(d)}$ is a matrix whose element $w_{ni}^{(d)} = 1$ if the word n in the document is the word i in the vocabulary. The size of $w^{(d)}$ is $N_d \times V$.
- $\theta^{(d)}$ is an array of size k, representing a probability density.
- $z^{(d)}$ is the set of topics : $z_{ni}^{(d)} = 1$ if the word n is linked with the topic i. Hence, it is a matrix of size $N_d \times k$.

Latent Dirichlet allocation (LDA) is a generative probabilistic model of a corpus. The main idea is that documents are represented as random mixtures over latent topics, where each topic is characterized by a distribution over words.

Algorithm 1: Generative process

Result: corpus $\mathcal{D}$
begin
  for each document $d \in \mathcal{D}$ **do**
    Choose $N_d \sim \text{Poisson}(\xi)$;
    Choose $\theta^{(d)} \sim \text{Dir}(\alpha)$;
    for each of the N_d words $w_n^{(d)}$ **do**
      Choose a topic $z_n^{(d)} \sim \text{Multinomial}(\theta^{(d)})$;
      Choose a word $w_n^{(d)}$ from $p(w_n^{(d)} | z_n^{(d)}, \beta)$, a multinomial probability conditioned on the topic $z_n^{(d)}$.

The following parameters are introduced:

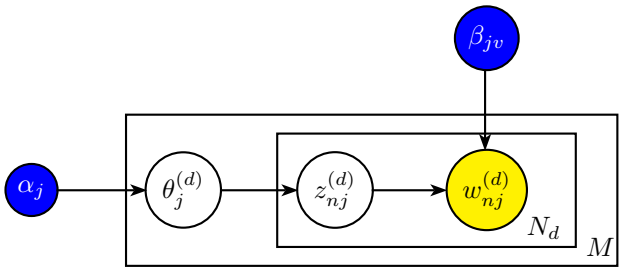

Figure 1: Generative model

- α (`dirich_param`) is an estimate of the parameter of the dirichlet distribution which generates the parameter for the (multinomial) probability distribution over topics in the document. The size of α is the number of topics, k. **We suppose that $\alpha = \alpha \mathbf{1}_k$ (exchangeable Dirichlet distribution)** [1] **.**
- β (`word_prob_given_topic`) is a matrix of size (number of topics $\times$ vocabulary size $= k \times V$) which gives the (estimated) probability that a given topic will generate a certain word:
$$\beta_{ij} = p(w^j = 1 | z^i = 1)$$
where w^j the j^{th} word of the vocabulary and z^i the i^{th} topic.

LDA is based on the computation of the parameters (α, β), for instance by maximizing the log-likelihood. For a document d, the probability $p(d|\alpha, \beta)$ is given by:

$$\begin{aligned} p(d|\alpha,\beta) &= \int p(\theta^{(d)}|\alpha) \left(\prod_{n=1}^{N_d} p(w_n^{(d)}|\theta^{(d)},\beta) \right) \mathrm{d}\theta \\ &= \int p(\theta^{(d)}|\alpha) \left(\prod_{n=1}^{N_d} \sum_{z_n^{(d)}} p(z_n|\theta^{(d)}) p(w_n^{(d)}|z_n^{(d)},\beta) \right) \mathrm{d}\theta \\ &= \frac{\Gamma\left(\sum_i \alpha_i\right)}{\prod_i \Gamma(\alpha_i)} \int \left(\prod_{i=1}^{k} (\theta_i^{(d)})^{\alpha_i - 1} \right) \left(\prod_{n=1}^{N_d} \sum_{i=1}^{k} \prod_{j=1}^{V} (\theta_i^{(d)} \beta_{ij})^{w_{nj}^{(d)}} \right) \mathrm{d}\theta \end{aligned}$$

1.2 Inference and parameter estimation

In the inference part, we need to compute the posterior distribution of the hidden variables given a document d:

$$p(\theta^{(d)}, z^{(d)} | d, \alpha, \beta) = \frac{p(\theta^{(d)}, z^{(d)}, d | \alpha, \beta)}{p(d|\alpha,\beta)}$$

Unfortunately, the distribution $p(d|\alpha, \beta)$ is not computable in general.

The idea is to use Jensen's inequality to obtain an adjustable lower bound on the log likelihood and to introduce new latent variables.

For each document $d \in \mathcal{D}$, the following latent variables are introduced:

- $\gamma^{(d)}$ (`var_dirich`) the variational parameter for the dirichlet distribution. The size of $\gamma^{(d)}$ is the number of topics, k.
- $\phi^{(d)}$ (`var_multinom`) the variational parameter for the multinomial distribution The size of $\phi^{(d)}$ is (number of distinct words in document d $\times$ number of topics), $N_d \times k$.

[1] This assumption is suggested by the authors of [BNJ03].

$\phi_{ni}^{(d)}$ depends on the relation between the word in position n of the document and the topic i of the list of topics.

The conditional probability is $q(\theta^{(d)}, z^{(d)}|\gamma^{(d)}, \delta^{(d)}) = q(\theta^{(d)}|\gamma^{(d)}) \prod_{n=1}^{N_d} q(z_n^{(d)}|\phi_n^{(d)})$

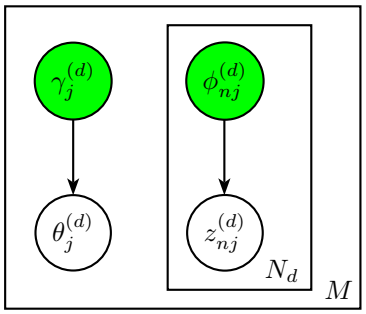

Figure 2: Variational model

We will estimate them instead of $\theta^{(d)}$ and $z_n^{(d)}$:

$$(\gamma^{(d)}, \phi^{(d)}) = \underset{(\gamma,\phi)}{\operatorname{argmin}} \mathrm{D}\left(q(\theta^{(d)}, z^{(d)}|\gamma, \phi) \middle\| p(\theta^{(d)}, z^{(d)}|d, \alpha, \beta)\right)$$

where $\mathrm{D}(\cdot\|\cdot)$ is the Kullback-Leibler (KL).

Algorithm 2: Variational Inference Procedure for a document d

Data: `word_incidences` $(w^{(d)})$, `dirich_param` (α), `word_prob_given_topic` (β)
Result: `var_dirich` $(\gamma^{(d)})$, `var_multinom` $(\phi^{(d)})$
begin
  Initialize $\phi_{ni}^{(d)} = \frac{1}{k}$ for all i and n;
  Initialize $\gamma_i^{(d)} = \alpha + \frac{1}{k}\sum_{n=1}^{N_d} w_n^{(d)}$ for all i ;
  while the expected log-likelihood for the document d has not converged **do**
    for $n = 1 \dots N_d$ **do**
      for $i = 1 \dots k$ **do**
        $\phi_{ni}^{(d)} = \beta_{iw_n^{(d)}} \exp(\Psi(\gamma_i^{(d)}))$
      normalize $\phi_n^{(d)}$ to sum to 1.
    $\gamma^{(d)} = \alpha + \sum_{n=1}^{N_d} \phi_n^{(d)}$

The article [BNJ03] tells us that a possible solution is:

$$\phi_{ni}^{(d)} \propto \beta_{iw_n^{(d)}} \exp(\mathbb{E}_q[\log(\theta_i)|\gamma])$$
$$\gamma^{(d)} = \alpha + \sum_{n=1}^{N_d} w_n^{(d)} \phi_n^{(d)}$$

and that $\mathbb{E}_q[\log(\theta_i)|\gamma] = \Psi(\gamma_i^{(d)}) - \Psi\left(\sum_{j=1}^{k} \gamma_j^{(d)}\right)$ where Ψ is the derivative of the $\log\Gamma$ function.

Algorithm 3: EM algorithm

Data: Corpus $\mathcal{D}$ of documents, number of topics k
Result: `dirich_param` (α), `word_prob_given_topic` (β)
begin
  for each $d \in \mathcal{D}$ **do**
    Compute `word_incidences` ($w^{(d)}$).
  Initialize `dirich_param` (α) with a uniform vector of size k;
  Initialize `word_prob_given_topic` (β) ;

  while the expected log-likelihood has not converged **do**
    for each $d \in \mathcal{D}$ **do**
      `var_dirich` ($\gamma^{(d)}$), `var_multinom` ($\phi^{(d)}$) = apply **variational-inference** to each document d given `word_incidences` ($w^{(d)}$), `dirich_param` (α), `word_prob_given_topic` (β)
    `dirich_param` (α), `word_prob_given_topic` (β) = Apply **M-step** given {`word_incidences` ($w^{(d)}$), `var_dirich` ($\gamma^{(d)}$), `var_multinom` ($\phi^{(d)}$), $d \in \mathcal{D}$}.

EM algorithm

The expected log-likelihood for a document d, is:

$$
\begin{aligned}
L(\gamma^{(d)}, \phi^{(d)}, \alpha, \beta) = {} & \log \Gamma\left(k\alpha\right) - k \log \Gamma(\alpha) + (\alpha - 1) \sum_{i=1}^{k} \left(\Psi(\gamma_i^{(d)}) - \Psi\left(\sum_{j=1}^{k} \gamma_j^{(d)} \right) \right) \\
& + \sum_{n=1}^{N_d} \sum_{i=1}^{k} \phi_{ni}^{(d)} \left(\Psi(\gamma_i^{(d)}) - \Psi\left(\sum_{j=1}^{k} \gamma_j^{(d)} \right) \right) \\
& + \sum_{n=1}^{N_d} \sum_{i=1}^{k} \sum_{j=1}^{V} \phi_{ni}^{(d)} w_{nj}^{(d)} \log \beta_{ij} \\
& - \log \Gamma\left(\sum_{j=1}^{k} \gamma_j^{(d)} \right) + \sum_{i=1}^{k} \log \Gamma(\gamma_i^{(d)}) - \sum_{i=1}^{k} (\gamma_i^{(d)} - 1) \left(\Psi(\gamma_i^{(d)}) - \Psi\left(\sum_{j=1}^{k} \gamma_j^{(d)} \right) \right) \\
& - \sum_{n=1}^{N_d} \sum_{i=1}^{k} \phi_{ni}^{(d)} \log \phi_{ni}^{(d)}
\end{aligned}
$$

Algorithm 4: M-step

Data: {`word_incidences` ($w^{(d)}$), `var_dirich` ($\gamma^{(d)}$), `var_multinom` ($\phi^{(d)}$), $d \in \mathcal{D}$}
Result: `dirich_param` (α), `word_prob_given_topic` (β)
begin
  $\beta \propto \sum_{d \in \mathcal{D}} (\phi^{(d)})^{\top} w^{(d)}$ (which corresponds to $\beta_{ij} \propto \sum_{d \in \mathcal{D}} \sum_{n=1}^{N_d} \phi_{ni}^{(d)} w_{nj}^{(d)}$)
  while α has not converged **do**
    $\frac{\partial L}{\partial \alpha}(\alpha) = |\mathcal{D}| k \left[\Psi\left(k\alpha\right) - \Psi(\alpha) \right] + \sum_{d \in \mathcal{D}} \left[\sum_{i=1}^{k} \Psi(\gamma_i^{(d)}) - \Psi\left(\sum_{j=1}^{k} \gamma_j^{(d)} \right) \right]$;
    $\frac{\partial^2 L}{\partial \alpha^2}(\alpha) = |\mathcal{D}| k [k \Psi'(k\alpha) - \Psi'\left(\alpha\right)]$;
    $\alpha \leftarrow \alpha - \frac{\frac{\partial L}{\partial \alpha}(\alpha)}{\frac{\partial^2 L}{\partial \alpha^2}(\alpha)}$

2 Implementation and results

2.1 Implementation

For implementation issues, we have chosen another representation for the documents.

- $d = (w_1^{(d)}, \dots, w_{N_d}^{(d)})$ represents the document d, where the $w_i^{(d)}$ are all **distinct**. We denote N_d (`doc_size`) the number of **distinct** words in the document d.
- $w^{(d)}$ (`word_incidences`) is a dictionary (of N_d entries) containing for each distinct word (indexed by its position in the vocabulary) the number of times it appears in the document.

These new notations change the algorithm 2. Its last line becomes:

$$\gamma^{(d)} = \alpha + \sum_{n=1}^{N_d} w_n^{(d)} \phi_n^{(d)}$$

The following algorithm was implemented :

Algorithm 5: Latent Dirichlet Allocation

Data: Corpus $\mathcal{D}$ of documents, number of topics k
Result: `dirich_param` (α), `word_prob_given_topic` (β)
begin
  for each $d \in \mathcal{D}$ **do**
    Compute `word_incidences` ($w^{(d)}$).
  Initialize `dirich_param` (α);
  Initialize `old_word_prob_given_topic` (β) ;

  while the expected log-likelihood $L(\mathcal{D}, \alpha, \beta)$ has not converged **do**
    Initialize `sum_psi_var_dirich` (Σ_γ) $= 0$;
    Initialize expected log-likelihood $L(\mathcal{D}, \alpha, \beta) = 0$;
    Initialize `word_prob_given_topic` (β_{new}) $= 0$;
    for each $d \in \mathcal{D}$ **do**
      `var_dirich` ($\gamma^{(d)}$), `var_multinom` ($\phi^{(d)}$) = apply **variational-inference** to each document d given `word_incidences` ($w^{(d)}$), `dirich_param` (α), `old_word_prob_given_topic` (β)
      Update $\beta_{\text{new}} \leftarrow \beta_{\text{new}} + (\phi^{(d)})^\top w^{(d)}$;
      Update $\Sigma_\gamma \leftarrow \Sigma_\gamma + \sum_{i=1}^k \Psi(\gamma_i^{(d)}) - \Psi\left(\sum_{j=1}^k \gamma_j^{(d)}\right)$;
      Update $L(\mathcal{D}, \alpha, \beta) \leftarrow L(\mathcal{D}, \alpha, \beta) + L(\gamma^{(d)}, \phi^{(d)}, \alpha, \beta)$;
    Normalize β_{new} and set $\beta = \beta_{\text{new}}$;
    while α has not converged **do**
      $\frac{\partial L}{\partial \alpha}(\alpha) = |\mathcal{D}| k \left[\Psi\left(k\alpha\right) - \Psi(\alpha)\right] + \Sigma_\gamma$;
      $\frac{\partial^2 L}{\partial \alpha^2}(\alpha) = |\mathcal{D}| k [k \Psi'(k\alpha) - \Psi'(\alpha)]$;
      $\alpha \leftarrow \alpha - \frac{\frac{\partial L}{\partial \alpha}(\alpha)}{\frac{\partial^2 L}{\partial \alpha^2}(\alpha)}$

2.2 Implementation tricks

Preprocessing: There is a document and vocabulary preprocessing before launching the LDA. The goal is to remove redundant words.

Log-likelihood: $\Gamma(x)$ becomes exponentially big when $x \to \infty$. We will prefer working with $\ln \Gamma(x)$ (`gammaln` in scipy) instead of $\Gamma(x)$ (or `numpy.log(gamma)`).

Initialization:

- $\alpha > 0$ is fixed. We have chosen $\alpha = 0.5$.
- β is chosen randomly, with $\sum_j \beta_{ij} = 1 \ \forall i \in \{1, \ldots, V\}$.

Exchangeable Dirichlet distribution assumption: Under this assumption, the computation of the derivates of the log-likelihood wrt α becomes much simpler.

2.3 Results on real data

The database we used may be found at the address http://www.daviddlewis.com/resources/testcollections/reuters21578/. We mainly worked with the corpus `reut2-000.sgm`, which contains approximatively 2000 documents. Table 1 shows the first words for five selected topics.

Topic 1	Topic 2	Topic 3	Topic 4	Topic 5
devices	prolonged	zestril	seasons	withdrawn
disk	council	annesthetic	hotels	expiration
megabyte	forum	hypertension	VMS	clearances
expandable	dissident	oth	Biltmore	expire
megabytes	flying	statil	Marriott	Willemijn
equipped	sparks	diabetic	rename	BV
monochrome	talks	complications	hotel	Rotterdam
peripheral	outweighed	Barbara	228	licensed
color	accomplishments	definitive	DH	NCR

Table 1: Results for 5 topics on the corpus `reut2-000.sgm` ($k = 20$)

Figure 3 represents the evolution of the expected log-likelihood computed on the whole corpus, whereas figure 4 represents the evolution of the expected log-likelihood computed one document. Different tries shows that we need at least 20 iterations so that the expected log-likelihood of the corpus converges.

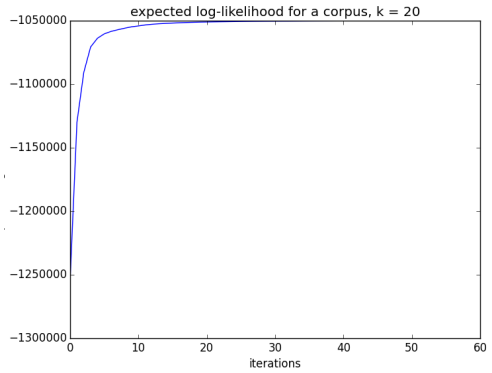

Figure 3: Expected log-likelihood for the corpus `reut2-000.sgm` ($k = 20$)

Multiple tries of our LDA inference program give different vectors of words for different tries. This is due to the fact that LDA inference is a non-convex optimization problem for α and β. The initialization of α and β may infer on the final stationary point that we get. We assumed that α follows the exchangeable Dirichlet distribution, which gives to the same weight to each topic.

For β, a random initialization may give us some coherent topics (see table 1). But sometimes, we obtain the same vector of words for each topic. A possible strategy to avoid that is to initialize β differently.

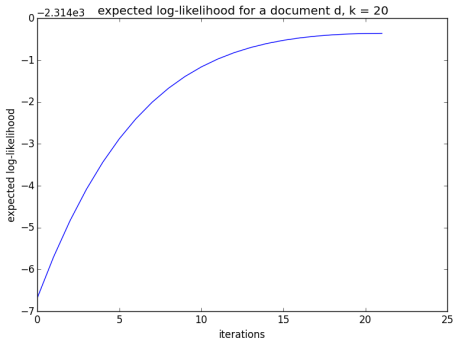

Figure 4: Expected log-likelihood for a document of the corpus `reut2-000.sgm` ($k = 20$)

3 Document classification

In this section, we are wondering whether the latent variables $(\gamma^{(d)}, \phi_n^{(d)})$ might be a good representation of a document d.

We consider the binary classification problem on a subset of the Reuters-21578 dataset (`reut2-000.sgm`, ..., `reut2-009.sgm`). The classification labels are "*earn*" and "*not earn*".

We trained two support vector machines (SVM) with a Gaussian RBF kernel, using the package `scikit-learn` for Python:

- one on the low-dimensional representation $\gamma^{(d)}$ provided by LDA,
- one on all the word features.

The accuracy rate for this binary classification problem with both representations are plotted in figure 5.

Figure 5: Accuracy rate for document classification problem

Figure 5 shows that the LDA low-dimensional representation has a better accuracy than the "all the word features" representation.

Topic-based representation provided by LDA may be useful as a fast filtering algorithm for feature selection in text classification.

4 Conclusion

LDA is an interesting way to apply graphical models to Natural Language Processing, and in our case, on information retrieval. Given a corpus and a set of vocabulary, LDA is able to group words in the same categories. The key idea in LDA is variational inference.

There are other interesting applications of LDA, such that biology (DNA sequence), content-based image retrieval...

References

[BNJ03] David M. Blei, Andrew Y. Ng and Michael I. Jordan. Latent Dirichlet Allocation. *The Journal of Machine Learning Research*, 3:993–1022, 2003.

[MH01] *An Experimental Comparison of Several Clustering and Initialization Methods*, Marina Meila, David Heckerman, 2001.

[YIS] *Latent Support Measure Machines for Bag-of-Words Data Classification*, Y. Yoshikawa, T. Iwata and H. Sawada