Probabilistic Programming

Marius Popescu

popescunmarius@gmail.com

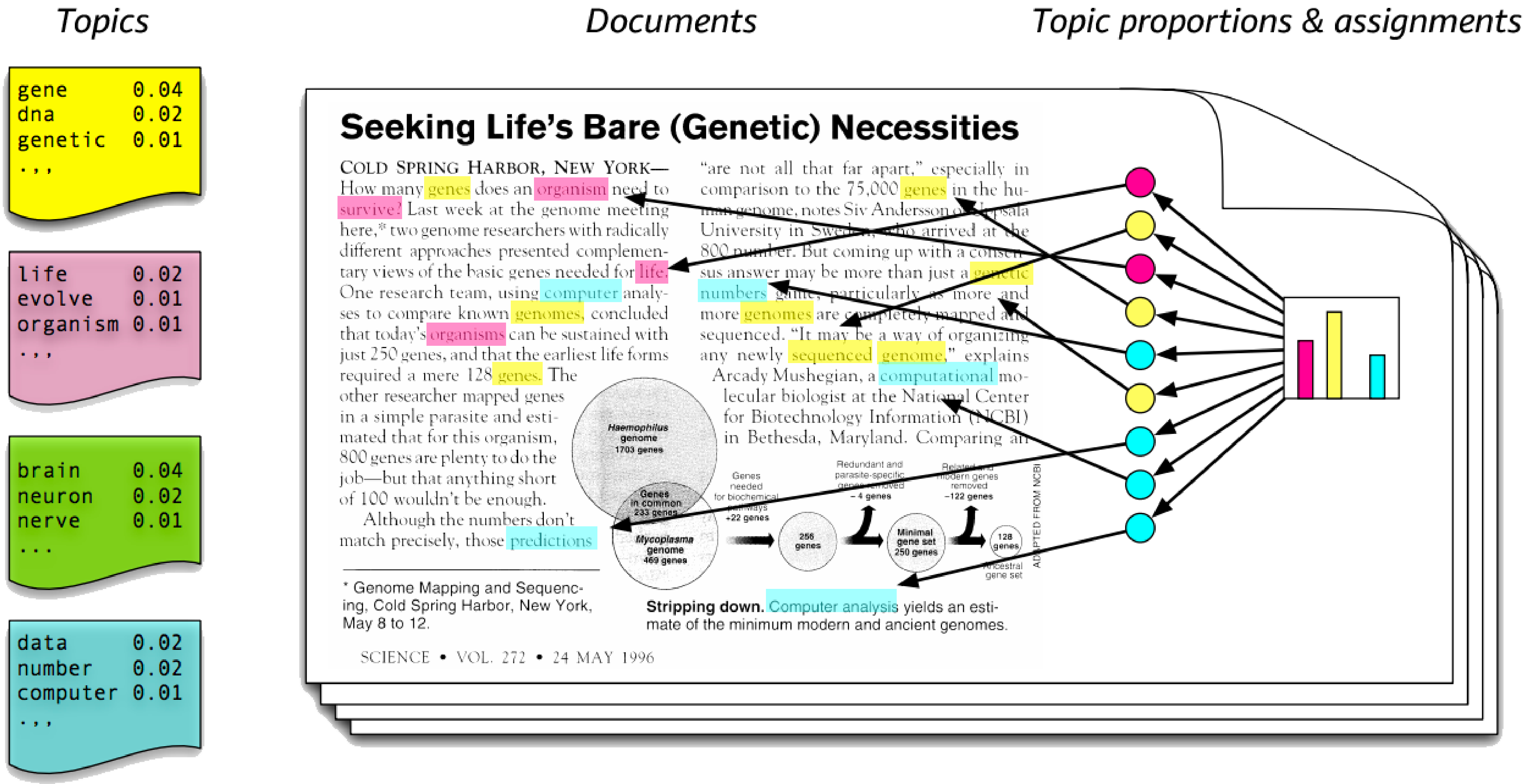
2019 - 2020

Project 1

Topic Modeling in PyMC

Topic Modeling

- Probabilistic models for uncovering the underlying semantic structure of a document collection based on a hierarchical Bayesian analysis of the original texts
- Topic models are algorithms for discovering the main themes that pervade a large and otherwise unstructured collection of documents. Topic models can organize the collection according to the discovered themes.



Latent Dirichlet Allocation (LDA)

- In the LDA model, each document is viewed as a mixture of topics that are present in the corpus. The model proposes that each word in the document is attributable to one of the document's topics
- The idea behind LDA is to model documents as arising from multiple topics, where a topic is defined to be a distribution over a fixed vocabulary of terms
- Given a dataset of documents, LDA backtracks and tries to figure out what topics would create those documents in the first place

http://www.cs.columbia.edu/~blei/papers/Blei2012.pdf

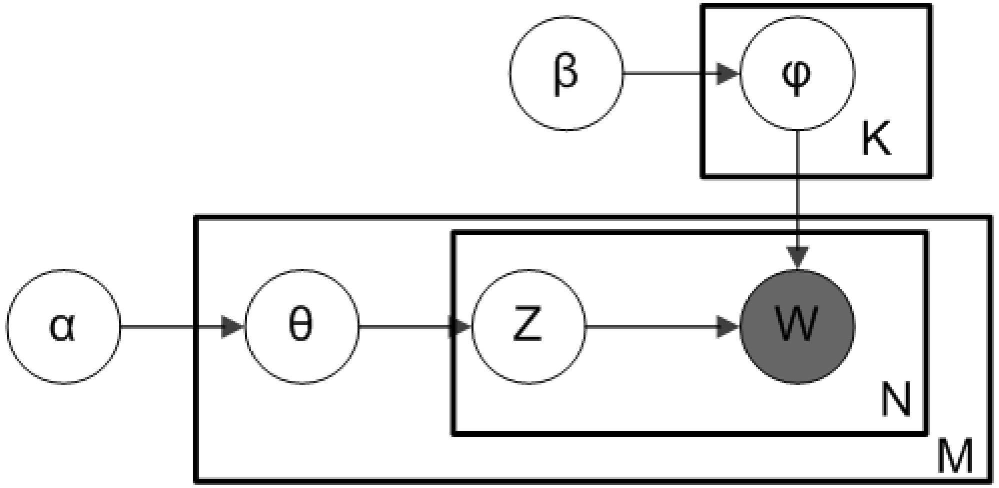
http://www.cs.columbia.edu/~blei/papers/BleiLafferty2009.pdf (sections 1 and 2)

https://en.wikipedia.org/wiki/Latent_Dirichlet_allocation

LDA: Generative Process

- Suppose, M is the number of documents in our collection. N_m is the number of words in the m-th document and V the size of the vocabulary. K is the number of predefined-topics. In this case, the number of topics is not automatically inferred. Instead, we will manually set the value of K based on our intuition.
- Documents are represented as random mixtures over latent topics, where each topic is characterized by a distribution over all the words

LDA: Generative Process



For each topic k, we draw its word distribution, which is denoted as φ_k. As prior for the per-topic word distribution we will use a V-dimensional symmetric Dirichlet distribution:

$$\varphi_k \sim \mathrm{Dir}(\beta), 1 \leq k \leq K$$

For each document m, we draw a topic distribution, which is denoted as θ_m. As prior for the per-document topic distribution we will use a K-dimensional symmetric Dirichlet distribution:

$$\theta_m \sim \mathrm{Dir}(\alpha), 1 \leq m \leq M$$

The hyperparameters α and β are assumed to be fixed typically sparse $\alpha, \beta \leq 1$ (set them to 1)

For each word n in document d (for each of the word positions), we draw a topic for that word according to a Multinomial (Categorical) distribution:

$$z_{m,n} \sim \mathrm{Multinomial}(\theta_m), 1 \leq m \leq M, 1 \leq n \leq N_m$$

Draw the physical word itself from the word distribution associated with its selected topic. Each word is denoted as $w_{m,n}$:

$$w_{m,n} \sim \mathrm{Multinomial}\left(\varphi_{z_{m,n}}\right), 1 \leq m \leq M, 1 \leq n \leq N_m$$

The Task

Document 1: I had a peanuts butter sandwich for breakfast.
Document 2: I like to eat almonds, peanuts and walnuts.
Document 3: My neighbor got a little dog yesterday.
Document 4: Cats and dogs are mortal enemies.
Document 5: You mustn't feed peanuts to your dog.

- Start with a corpus (document collection)
- Build the observed variable:

$$w_{m,n}\ 1 \le m \le M, 1 \le n \le N_m$$

- Infer the hidden topic structure:

$$\theta_m\ 1 \le m \le M$$

$$\varphi_k\ 1 \le k \le K$$

```
[[0, 1, 2, 3, 4, 5, 6, 7]
 [0, 8, 9, 10, 11, 3, 12, 13]
 [14, 15, 16, 2, 17, 18, 19]
 [20, 12, 21, 22, 23, 24]
 [25, 26, 27, 3, 9, 28, 18]]
```

```
[[ 0.19633795  0.80366205]]
[[ 0.18370696  0.81629304]]
[[ 0.2242626   0.7757374]]
[[ 0.72682194  0.27317806]]
[[ 0.19720619  0.80279381]]
```

```
[[ 0.0153611   0.04258534  0.02096933  0.03742481  0.01003486  0.00337692
   0.03687246  0.01759574  0.0308261   0.01290443  0.01975973  0.01656558
   0.08359959  0.00988798  0.00815009  0.02573078  0.00934478  0.02543332
   0.0267678   0.01700781  0.07986162  0.15302288  0.09572016  0.06718658
   0.03541293  0.01084848  0.00399011  0.02285365  0.06090502]]
[[ 0.0475376   0.03045149  0.03110808  0.07166036  0.02985312  0.03845919
   0.03310906  0.03910945  0.05759958  0.0438841   0.0596612   0.02342716
   0.03582313  0.02156961  0.03489024  0.01348096  0.00848053  0.04888085
   0.03665115  0.02932311  0.02411743  0.01016715  0.02278155  0.04463838
   0.00023452  0.05109875  0.03079135  0.04895514  0.03225578]]
```

Sanity Check

- Start with a corpus (document collection)
- Build the observed variable:

$$w_{m,n} \; 1 \le m \le M, 1 \le n \le N_m$$

- Infer the hidden topic structure:

$$\theta_m \; 1 \le m \le M$$

$$\varphi_k \; 1 \le k \le K$$

- Trace also:

$$z_{m,n} 1 \le m \le M, 1 \le n \le N_m$$

```
docs = [["aaa", "bbb", "aaa"],
        ["bbb", "aaa", "bbb"],
        ["aaa", "bbb", "bbb", "aaa"],
        ["uuu", "vvv"],
        ["uuu", "vvv", "vvv"],
        ["uuu", "vvv", "vvv", "uuu"]]

[[0, 1, 0] [1, 0, 1] [0, 1, 1, 0] [2, 3] [2, 3, 3] [2, 3, 3, 2]]
```

```
[[ 0.56363964  0.43636036]]
[[ 0.3713786   0.6286214]]
[[ 0.96678627  0.03321373]]
[[ 0.04743652  0.95256348]]
[[ 0.26409289  0.73590711]]
[[ 0.24063087  0.75936913]]
```

```
[[ 0.49718579  0.33963714  0.08627236  0.07690471]]
[[ 0.02396443  0.10446209  0.5212554   0.35031808]]
```

```
[0 0 0]
[0 0 0]
[0 0 0 0]
[1 1]
[1 1 1]
[1 1 1 1]
```

Extras

- Can the topic model be used to define a topic-based similarity measure between documents? (0.5)
- What about a new document? How can topics be assigned to it? (0.75)
- Extensions:
  - The correlated topic model (1.5)
  - The dynamic topic model (1.5)