RITAL

Information retrieval and natural language processing
Recherche d'information et traitement automatique de la langue

Master 1 DAC, semestre 2

Nicolas Thome

Bag of Words (BOW) for document classification (2)

Outline

1. Bag of Words (BOW) for document classification (2)
2. Semantic modeling
3. Unsupervised approaches

Semantic modeling

Outline

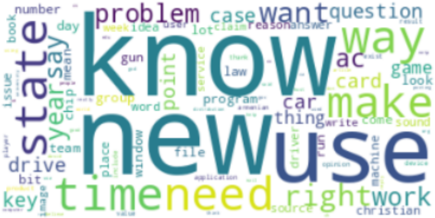
Unsupervised approaches

Outline

Word Clouds

- Use word clouds to visualize the most relevant words from a corpus
- From raw data (using stop words)
- From BoW (using vectorizer - and stop words)
- Ex: fetch20newsgroups (practical)

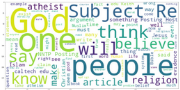
Raw data



BoW (frequencies)

Word Clouds: classes

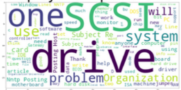
alt.atheism



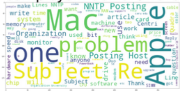
comp.graphics

comp.os.ms-windows.misc

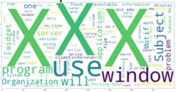
comp.sys.ibm.pc.hardware



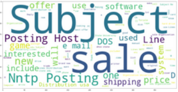
comp.sys.mac.hardware



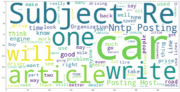
comp.windows.x



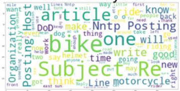
misc.forsale



rec.autos



rec.motorcycles



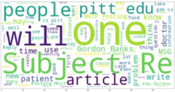
rec.sport.baseball

rec.sport.hockey

sci.crypt

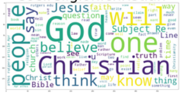
sci.electronics

sci.med



sci.space

soc.religion.christian



- Classes: semantic informtion
  - Odd ratios can improve discriminability
- How to extract semantic info without labels?
  $\Rightarrow$ **Unsupervised learning**

- Modeling: Word count (and BoW storage)

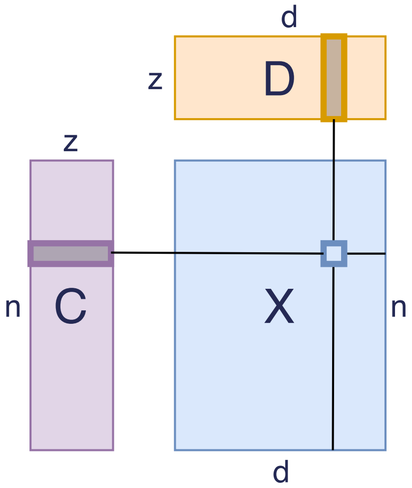
$$X = \begin{array}{cc} & \mathbf{t}_j \\ & \downarrow \\ \mathbf{d}_i \rightarrow & \begin{pmatrix} x_{1,1} & \cdots & x_{1,d} \\ \vdots & \ddots & \vdots \\ x_{N,1} & \cdots & x_{N,d} \end{pmatrix} \end{array}$$

- Basic proposal: semantics = metrics = similarity between columns in BoW

$$s(j,k) = \langle t_j, t_k \rangle, \qquad \text{Normalized: } s_n(j,k) = \cos(\theta) = \frac{\mathbf{t}_j \cdot \mathbf{t}_q}{\|\mathbf{t}_j\| \, \|\mathbf{t}_q\|}$$

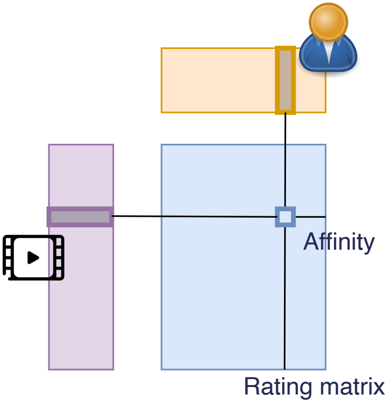
  - If two terms appear in the same document, they are similar

Matrix factorization = basic tool to understand the data



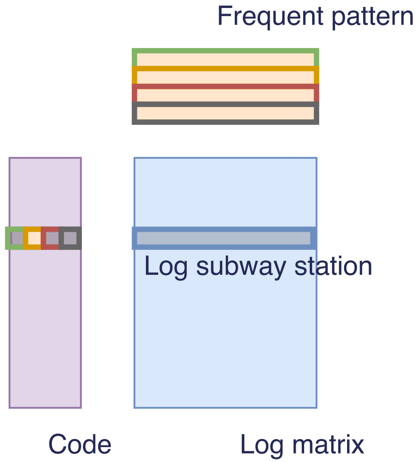
- Extract a compact representation
  - for words
  - for documents
- = focus on high-energy phenomenon
  - Eliminate noise in the data
- Optimal data compression [Mean Square criterion]

Matrix factorization = basic tool to understand the data



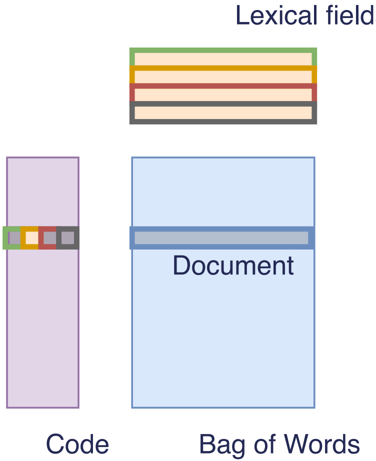
- Extract a compact representation
  - for words
  - for documents
- = focus on high-energy phenomenon
  - Eliminate noise in the data
- Optimal data compression [Mean Square criterion]

Matrix factorization = basic tool to understand the data



- Extract a compact representation
  - for words
  - for documents
- = focus on high-energy phenomenon
  - Eliminate noise in the data
- Optimal data compression [Mean Square criterion]

Matrix factorization = basic tool to understand the data



- Extract a compact representation
  - for words
  - for documents
- = focus on high-energy phenomenon
  - Eliminate noise in the data
- Optimal data compression [Mean Square criterion]

SVD : Singular Value decomposition

- In NLP : SVD = LSA: Latent Semantic Analysis
- Idea : grouping similar documents / learning a representation of documents

$$
\begin{array}{ccccccc}
 & X & = & U & \Sigma & V^T \\
 & \mathbf{t}_j & & \hat{\mathbf{d}}_i & & \\
 & \downarrow & & \downarrow & & \\
\mathbf{d}_i \rightarrow & \begin{pmatrix} x_{1,1} & \dots & x_{1,d} \\ \vdots & \ddots & \vdots \\ x_{N,1} & \dots & x_{N,d} \end{pmatrix} & = & \begin{pmatrix} (& \mathbf{u}_1 &) \\ & \vdots & \\ (& \mathbf{u}_k &) \end{pmatrix} & \begin{pmatrix} \sigma_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \sigma_k \end{pmatrix} & \left(\begin{pmatrix} \\ \mathbf{v}_1 \\ \\ \end{pmatrix} \dots \begin{pmatrix} \\ \mathbf{v}_k \\ \\ \end{pmatrix} \right)
\end{array}
$$

- Good news: functions well on sparse matrices
  - See `TruncatedSVD` in `sklearn.decomposition`

Factorization = robustness & clustering ability

S. Deerwester, et al., JSIS 1990
Indexing by latent semantic analysis

Discussion : SVD, LSA

Selecting the k greatest singular values: rank-k approximation of the occurence matrix X

$$
\begin{array}{cccccc}
 & X & = & U & \Sigma & V^T \\
 & \mathbf{t}_j & & \hat{\mathbf{d}}_i & & \\
 & \downarrow & & \downarrow & & \\
\mathbf{d}_i \rightarrow & \begin{pmatrix} x_{1,1} & \cdots & x_{1,d} \\ \vdots & \ddots & \vdots \\ x_{N,1} & \cdots & x_{N,d} \end{pmatrix} & = & \begin{pmatrix} (& \mathbf{u}_1 &) \\ & \vdots & \\ (& \mathbf{u}_k &) \end{pmatrix} & \begin{pmatrix} \sigma_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_k \end{pmatrix} & \begin{pmatrix} \begin{pmatrix} \\ \mathbf{v}_1 \\ \\ \end{pmatrix} \cdots \begin{pmatrix} \\ \mathbf{v}_k \\ \\ \end{pmatrix} \end{pmatrix}
\end{array}
$$

- Each $\mathbf{v}_i \in \mathbb{R}^d$: a weight vector associated to the vocabulary
- The base $\{\mathbf{v}_1, \ldots, \mathbf{v}_k\}$ is orthogonal
  - Each $\mathbf{v}_i$ corresponds to a different **lexical field**
- The new $\hat{\mathbf{d}}_i$ representation $\boldsymbol{u_i}$: weight vector associated to the lexical fields
  - **Clustering:** the strongest weight gives the document class

Thomas K. Landauer, Peter W. Foltz et Darrell Laham, Discourse Processes, vol. 25, 1998
Introduction to Latent Semantic Analysis

Many applications

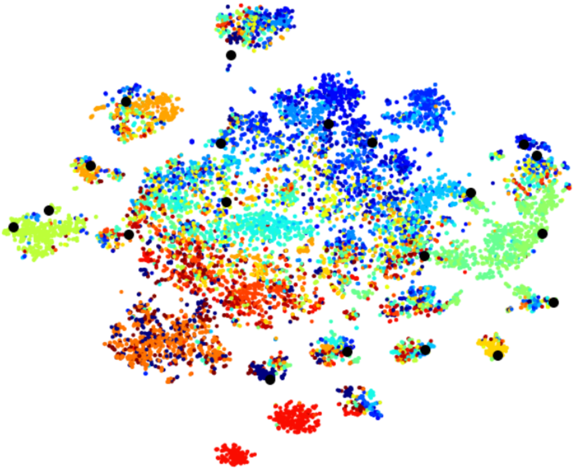
Usages:

- Clustering (each eigen vector describes a *topic)*
- Semantics: words have a representation over the topics
- IR Improvement:
  - Query expansion based on the topic definition
  - Detection of polysemic terms
- new representation $\Rightarrow$ new metrics
  - opportunities in question answering
    - Finding the part of a document relating to a specific topic
  - Automated summarization
    - Document segmentation + sentence extraction
  - TDT : Topic detection & Tracking

$$
\begin{array}{cccccc}
 & X & = & U & \Sigma & V^T \\
 & \mathbf{t}_j & & \hat{\mathbf{d}}_i & & \\
 & \downarrow & & \downarrow & & \\
\mathbf{d}_i \rightarrow & \begin{pmatrix} x_{1,1} & \dots & x_{1,d} \\ \vdots & \ddots & \vdots \\ x_{N,1} & \dots & x_{N,d} \end{pmatrix} & = & \begin{pmatrix} (\quad \mathbf{u}_1 \quad) \\ \vdots \\ (\quad \mathbf{u}_k \quad) \end{pmatrix} & \begin{pmatrix} \sigma_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \sigma_k \end{pmatrix} & \left(\begin{pmatrix} \\ \mathbf{v}_1 \\ \\ \end{pmatrix} \dots \begin{pmatrix} \\ \mathbf{v}_k \\ \\ \end{pmatrix} \right)
\end{array}
$$

- On fetch20newsgroups, test with $k = 20$
- **Qualitative assessment:**
  - $\mathbf{v}_i \in \mathbb{R}^d$: look at most important words
  - $\hat{\mathbf{d}}_i := \mathbf{u}_i$: cluster each document / topics
  - Word clouds for each cluster
  - t-SNE after LSA projection

- Each dot: cluster center
- color code: GT classes



$$
\begin{array}{cccccc}
 & X & = & U & \Sigma & V^T \\
 & \underset{\downarrow}{\mathbf{t}_j} & & \underset{\downarrow}{\hat{\mathbf{d}}_i} & & \\
\mathbf{d}_i \rightarrow & \begin{pmatrix} x_{1,1} & \dots & x_{1,d} \\ \vdots & \ddots & \vdots \\ x_{N,1} & \dots & x_{N,d} \end{pmatrix} & = & \begin{pmatrix} (& \mathbf{u}_1 &) \\ & \vdots & \\ (& \mathbf{u}_k &) \end{pmatrix} & \begin{pmatrix} \sigma_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \sigma_k \end{pmatrix} & \left(\begin{pmatrix} \\ \mathbf{v}_1 \\ \\ \end{pmatrix} \dots \begin{pmatrix} \\ \mathbf{v}_k \\ \\ \end{pmatrix} \right)
\end{array}
$$

- On fetch20newsgroups, test with $k = 20$
- **Quantitative assessment:** with 3 metrics
  - Purity: $p = \frac{|y^*|}{|C|}$, where y^* is the most frequent (GT) label in cluster C
  - Rand score: `https://en.wikipedia.org/wiki/Rand_index`
  - Adjusted Rand score (ARS)

LSA Limits

- Fully based on BOW: no word dependency modeling
  - issues regarding negative formulation
  - depends on document sizes
  - Not robust to stop words
    - associated to high singular values
    - + appear in many topics
- Topic modeling is link to a corpus
  - problem with rare words in small corpus
  - bias of the corpus

Outline

K-means

- Still a BOW modeling

$$X = \begin{array}{cc} & \begin{array}{ccc} \phantom{x_{1,1}} & \mathbf{t}_j & \phantom{x_{1,d}} \\ & \downarrow & \end{array} \\ \mathbf{d}_i \rightarrow & \begin{pmatrix} x_{1,1} & \cdots & x_{1,d} \\ \vdots & \ddots & \vdots \\ x_{N,1} & \cdots & x_{N,d} \end{pmatrix} \end{array}$$

- Algorithm that scale up well
  - Possible **on-line** version of the algorithm
  - Can be linked to chinese restaurant / indian buffet process
    - $\Rightarrow$ Discover k in an online process
- Orthogonality is not longer enforced

New vision of k-means :

- k clusters
- A priori probabilities : $\pi_k = p(\theta_k)$
- Probability of a word in a cluster : $p(w_j|\theta_k) = \mathbb{E}_{d \in \mathcal{D}_k}[w_j]$
- Document hard assignment in a cluster: $p(\theta_k|d_i) = 1/0$

$$y_i = \arg\max_k p(\theta_k)p(d_i|\theta_k) = \arg\max_k \log(\pi_k) + \sum_{w_j \in d_i} \log p(w_j|\theta_k)$$

$$y_i = \arg\max_k \sum_j t_{ij}\theta_{jk}, \text{ with } :\theta_{jk} = \log p(w_j|\theta_k) \text{ and uniform prior}$$

Algorithm:

Init. Random or expert knowledge
C/E Cluster assignment
M Parameter update (mean re-computation)

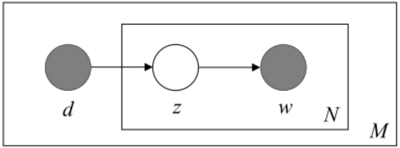
- K-means on top of LSA
- LSA acts as pre-processing (denoising, finding relevant words)
  - $\Rightarrow$ improved clustering over K-Means on raw data
  - Especially for large vocabulary

Outline

SCIENCES SORBONNE UNIVERSITÉ

Probabilistic Latent Semantic Analysis

- Idea: CEM $\Rightarrow$ EM (more complex / finer)
- All documents belongs to all clusters... With a weight $p(z|d)$
- Graphical model: conditional independence $\Rightarrow p(w, d|z) = p(w|z)p(z|d)$



- Doc d is drawn from $P(d)$
- Topic z is drawn from $P(z|d)$
- Word w is drawn from $P(w|z)$

We estimate the following parameters:

- $p(d)$
- $p(\alpha|d)$
- $p(w|\alpha)$

Maximizing the log-likelyhood:

$$\mathcal{L} = \sum_{d=1}^{D} \sum_{w=1}^{W} n(d, w) \log P(d, w)$$

- Expectation (probability of the missing variables)
- Maximization

Maximizing the log-likelyhood:

$$\mathcal{L} = \sum_{d=1}^{D} \sum_{w=1}^{W} n(d, w) \log P(d, w)$$

- **E**xpectation (probability of the missing variables)

$$P(\alpha|d,w) = \frac{P(d)P(\alpha|d)P(w|\alpha)}{\sum_{\alpha' \in \mathcal{A}} P(d)P(\alpha'|d)P(w|\alpha')}$$

- **M**aximization

Maximizing the log-likelyhood:

$$\mathcal{L} = \sum_{d=1}^{D} \sum_{w=1}^{W} n(d, w) \log P(d, w)$$

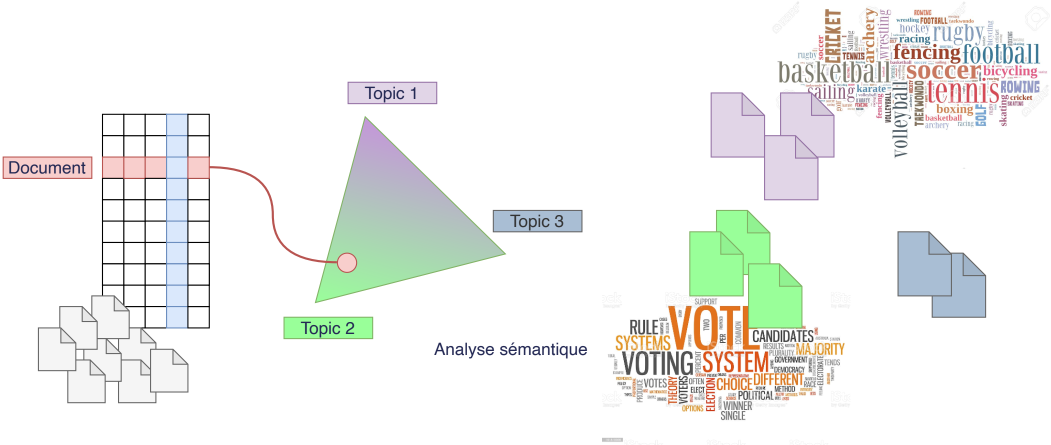
- Expectation (probability of the missing variables)
- Maximization

$$P(d) = \frac{\sum_{w \in \mathcal{W}} n(d, w)}{\sum_{d' \in \mathcal{D}} \sum_{w \in \mathcal{W}} n(d', w)}$$

$$P(\alpha|d) = \frac{\sum_{w \in \mathcal{W}} n(d, w) P(\alpha|d, w)}{\sum_{\alpha' \in \mathcal{A}} \sum_{w \in \mathcal{W}} n(d, w) P(\alpha'|d, w)}$$

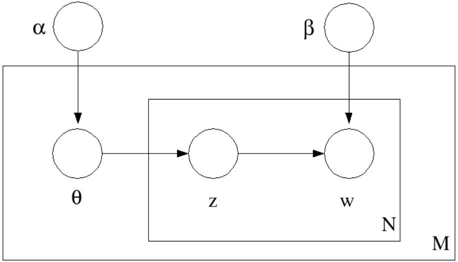
$$P(w|\alpha) = \frac{\sum_{d \in \mathcal{D}} n(d, w) P(\alpha|d, w)}{\sum_{w' \in \mathcal{W}} \sum_{d \in \mathcal{D}} n(d, w') P(\alpha|d, w')}$$

PLSA: results



Outline

Latent Dirichlet Allocation:

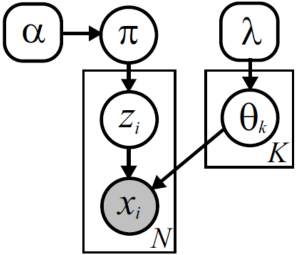


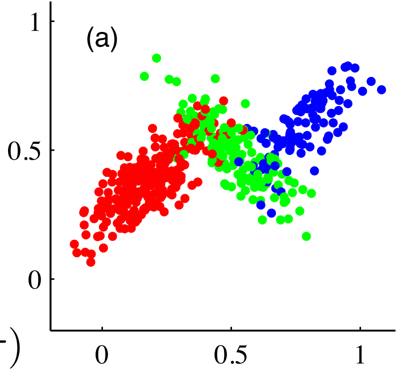
- Idea: adding a prior on the topic distribution
  - A document is supposed to belong to a topic **strongly or not**
- Learning through Gibbs sampling ($\sim$ MCMC)

not to be confused: LDA: Latent Dirichlet Allocation *vs* Linear Discriminant Analysis

Gibbs Sampling

On an example:





$$\theta_k = \{\mu_k, \Sigma_k\}$$

$$p(z_i \mid \pi) = \mathrm{Cat}(z_i \mid \pi)$$

$$p(x_i \mid z_i, \mu, \Sigma) = \mathcal{N}(x_i \mid \mu_{z_i}, \Sigma_{z_i})$$

Given mixture weights $\pi^{(t-1)}$ and cluster parameters $\{\theta_k^{(t-1)}\}_{k=1}^K$ from the previous iteration, sample a new set of mixture parameters as follows:

1. Independently assign each of the N data points x_i to one of the K clusters by sampling the indicator variables $z = \{z_i\}_{i=1}^N$ from the following multinomial distributions:

$$z_i^{(t)} \sim \frac{1}{Z_i} \sum_{k=1}^{K} \pi_k^{(t-1)} f(x_i \mid \theta_k^{(t-1)})\, \delta(z_i, k) \qquad Z_i = \sum_{k=1}^{K} \pi_k^{(t-1)} f(x_i \mid \theta_k^{(t-1)})$$

2. Sample new mixture weights according to the following Dirichlet distribution:

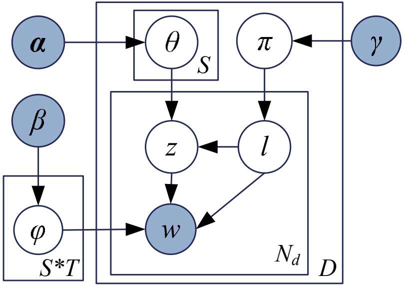
$$\pi^{(t)} \sim \mathrm{Dir}(N_1 + \alpha/K, \ldots, N_K + \alpha/K) \qquad N_k = \sum_{i=1}^{N} \delta(z_i^{(t)}, k)$$

3. For each of the K clusters, independently sample new parameters from the conditional distribution implied by those observations currently assigned to that cluster:

$$\theta_k^{(t)} \sim p(\theta_k \mid \{x_i \mid z_i^{(t)} = k\}, \lambda)$$

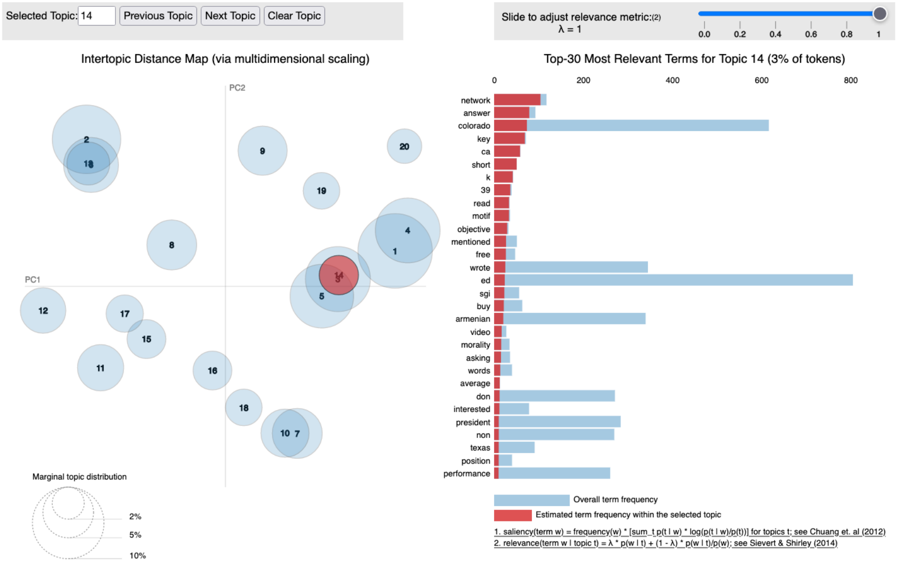
When λ defines a conjugate prior, this posterior distribution is given by Prop. 2.1.4.

- Graphical models = easy to adapt



- For each document d, choose a distribution $\pi_d \sim \text{Dir}(\gamma)$.
- For each sentiment label l under document d, choose a distribution $\theta_{d,l} \sim \text{Dir}(\alpha)$.
- For each word w_i in document d
  - choose a sentiment label $l_i \sim \text{Mult}(\pi_d)$,
  - choose a topic $z_i \sim \text{Mult}(\theta_{d,l_i})$,
  - choose a word w_i from $\varphi_{z_i}^{l_i}$, a Multinomial distribution over words conditioned on topic z_i and sentiment label l_i.

Practical: LDA viz



1. Quantitative results
   - Clustering
   - Major issue with frequent words
   - Human required in the loop (init., cluster selection, etc...)
   - Evaluation issue (purity, perplexity, ...)
2. Qualitative analysis
   - Word similarity
   - Lexical field extraction