# Bayesian Machine Learning

June 2024 - François HU
https://curiousml.github.io/

# Outline

1 **Bayesian statistics**

2 **Latent variable models**

3 **Variational Inference**
- Variational Inference for probabilistic models
- Introduction to NLP
- Application on textual data with LDA

4 **Causal Inference**

5 **Extensions and oral presentations**

# 1 Variational Inference for probabilistic models

# 1. Variational Inference for probabilistic models

## Reminder

**Posterior distribution**

Fixed by **model** — **Likelihood**

Fixed by **us** — **Prior**

$$P(Z|X) = \frac{P(X,Z)}{P(X)} = \frac{P(X|Z) \times P(Z)}{P(X)}$$

**Posterior**

**Evidence** — Fixed by **data**

# 1. Variational Inference for probabilistic models

## Reminder

### Posterior distribution

Fixed by **model** — **Likelihood**

Fixed by **us** — **Prior**

$$P(Z|X) = \frac{P(X,Z)}{P(X)} = \frac{P(X|Z) \times P(Z)}{P(X)}$$

**Posterior**

**Evidence** — Fixed by **data**

### Methods we have seen so far

- **Analytical inference.** Given $P(X|Z)$**,** we infer $P_X(Z) := P(Z|X)$ by
  1. **Conjugate priors** : *easy* with a good matching prior
  2. **Optimization** using EM algorithm : *tricky*,

     needs the computation of $\mathbb{E}_T\left[\log P(X,T|\theta)\right]$ with $Z = \{T,\theta\}$

# 1. Variational Inference for probabilistic models

## Reminder

### Posterior distribution

Fixed by **model** — Likelihood

Fixed by **us** — Prior

$$P(Z|X) = \frac{P(X,Z)}{P(X)} = \frac{P(X|Z) \times P(Z)}{P(X)}$$

Posterior

Evidence — Fixed by **data**

$$\hat{\theta}_{MLE} = \arg\max_{\theta} \log p(x|\theta)$$

$$\hat{\theta}_{EM} = \arg\max_{\theta} \left[ \max_{q} \mathcal{L}(\theta, q) \right]$$

$\mathcal{L}(\theta, q)$ → "evidence lower bound" (ELBO)

$$\mathcal{L}(\theta, q) = \sum_t q(t) \log \left( \frac{p(x,t|\theta)}{q(t)} \right)$$

Key Idea

- log likelihood is hard to optimize

$$\max_{\theta} \log p(x|\theta)$$

- typically introducing a latent variable is easier to optimize

$$\max_{\theta} \log p(x,T|\theta)$$

- IF we had a distribution $q(T)$ for the l.v T THEN

$$\max_{\theta} \underbrace{\sum_t q(t) \log p(x,t|\theta)}_{E_T[\log p(x,T|\theta)]}$$

EM assumes this maximization relatively easy

E step : let $q^*(t) = p(t|x, \theta^{old})$

$$J(\theta) = \mathcal{L}(q^*, \theta) = \sum_t q^*(t) \log \left( \frac{p(x,t|\theta)}{q^*(t)} \right)$$

M-step : $\theta^{new} = \arg\max_{\theta} J(\theta)$

# 1. Variational Inference for probabilistic models

## Approximate inference

### Posterior distribution

Fixed by **model** — **Likelihood**

Fixed by **us** — **Prior**

$$P(Z|X) = \frac{P(X,Z)}{P(X)} = \frac{P(X|Z) \times P(Z)}{P(X)}$$

**Posterior**

**Evidence** — Fixed by **data**

### Methods we have seen so far

- **Analytical inference.** Given $P(X|Z)$, we infer $P_X(Z) := P(Z|X)$ by
    1. **Conjugate priors** : *easy* with a good matching prior
    2. **Optimization** using EM algorithm : *tricky*,

       needs the computation of $\mathbb{E}_T \left[\log P(X, T|\theta)\right]$ with $Z = \{T, \theta\}$

### In lecture 4 and 5

- **Approximate inference.** Approximate $P_X(Z) \approx \hat{P}_X(Z)$
    1. **Deterministic approach** : Variational Inference
    2. **Stochastic approach** : Markov Chain Monte Carlo

# 1. Variational Inference for probabilistic models

## Variational Inference : Definition

### Posterior distribution

Fixed by model — Likelihood

Fixed by us — Prior

$$P(Z|X) = \frac{P(X,Z)}{P(X)} = \frac{P(X|Z) \times P(Z)}{P(X)}$$

Posterior

Evidence — Fixed by data

### Methods we have seen so far

- **Analytical inference.** Given $P(X|Z)$, we infer $P_X(Z) := P(Z|X)$ by
  1. **Conjugate priors** : *easy* with a good matching prior
  2. **Optimization** using EM algorithm : *tricky*,

     needs the computation of $\mathbb{E}_T \left[\log P(X, T|\theta)\right]$ with $Z = \{T, \theta\}$

### In lecture 4 and 5

- **Approximate inference.** Approximate $P_X(Z) \approx \hat{P}_X(Z)$
  1. **Deterministic approach** : Variational Inference
  2. **Stochastic approach** : Markov Chain Monte Carlo

### Variational Inference (VI)

(i) Select a family of distributions $\mathcal{Q}$

(ii) Find the « **best** » approximation $\hat{P}_X \in \mathcal{Q}$ : « $P_X(Z) \approx \hat{P}_X(Z)$ »

# 1. Variational Inference for probabilistic models

## Variational Inference : KL-divergence

### Posterior distribution

Fixed by **model** — Likelihood

Fixed by **us** — Prior

$$P(Z|X) = \frac{P(X,Z)}{P(X)} = \frac{P(X|Z) \times P(Z)}{P(X)}$$

Posterior

Evidence — Fixed by **data**

### Kullback-Leibler (KL) divergence

Consider $P$ and $Q$ two distributions

we want to compare their « differences » / divergence.

Ex. of measure : $D_{KL}(Q||P) = \int_{z \in Supp(Z)} Q(z) \cdot \log \left( \frac{Q(z)}{P(z)} \right) dz$

### Methods we have seen so far

- **Analytical inference.** Given $P(X|Z)$, we infer $P_X(Z) := P(Z|X)$ by
  1. **Conjugate priors** : *easy* with a good matching prior
  2. **Optimization** using EM algorithm : *tricky*,

     needs the computation of $\mathbb{E}_T \left[ \log P(X, T | \theta) \right]$ with $Z = \{T, \theta\}$

### In lecture 4 and 5

- **Approximate inference.** Approximate $P_X(Z) \approx \hat{P}_X(Z)$
  1. **Deterministic approach** : Variational Inference
  2. **Stochastic approach** : Markov Chain Monte Carlo

### Variational Inference (VI)

(i) Select a family of distributions $\mathcal{Q}$

(ii) Find the « **best** » approximation $\hat{P}_X \in \mathcal{Q}$ : « $P_X(Z) \approx \hat{P}_X(Z)$ »

# 1. Variational Inference for probabilistic models

## Variational Inference : KL-divergence

### Posterior distribution

Fixed by **model** — Likelihood

Fixed by **us** — Prior

$$P(Z|X) = \frac{P(X,Z)}{P(X)} = \frac{P(X|Z) \times P(Z)}{P(X)}$$

Posterior

Evidence — Fixed by **data**

### Kullback-Leibler (KL) divergence

Consider $P$ and $Q$ two distributions

we want to compare their « differences » / divergence.

Ex. of measure : $D_{KL}(Q||P) = \int_{z \in Supp(Z)} Q(z) \cdot \log\left(\frac{Q(z)}{P(z)}\right) dz$

### Methods we have seen so far

- **Analytical inference.** Given $P(X|Z)$, we infer $P_X(Z) := P(Z|X)$ by
  1. **Conjugate priors** : *easy* with a good matching prior
  2. **Optimization** using EM algorithm : *tricky*,

     needs the computation of $\mathbb{E}_T\left[\log P(X,T|\theta)\right]$ with $Z = \{T, \theta\}$

### In lecture 4 and 5

- **Approximate inference.** Approximate $P_X(Z) \approx \hat{P}_X(Z)$
  1. **Deterministic approach** : Variational Inference
  2. **Stochastic approach** : Markov Chain Monte Carlo

### Variational Inference (VI)

(i) Select a family of distributions $\mathcal{Q}$

(ii) Find the **best** approximation : $\hat{P}_X = \arg\min_{Q \in \mathcal{Q}} D_{KL}(Q||P_X)$

# 1. Variational Inference for probabilistic models

## Variational Inference : Mean Field Approximation

### Variational Inference (VI)

(i) Select a family of distributions $\mathbb{Q}$ on $Z = (Z_1, \ldots, Z_d)$

(ii) Find the **best** approximation : $\hat{P}_X = \arg\min_{Q \in \mathbb{Q}} D_{KL}(Q||P_X)$

# 1. Variational Inference for probabilistic models

## Variational Inference : Mean Field Approximation

### Variational Inference (VI)

(i) Select a family of distributions $\mathcal{Q}$ on $Z = (Z_1, \ldots, Z_d)$

(ii) Find the **best** approximation : $\hat{P}_X = \arg\min_{Q \in \mathcal{Q}} D_{KL}(Q || P_X)$

### Mean Field Approximation

(i) we choose $\mathcal{Q} = \left\{ Q = (Q_1, \ldots, Q_d) : Q(Z) = \prod_{i=1,\ldots,d} Q_i(Z_i) \right\}$

instead of $Q(Z_1, \ldots, Z_n) = \prod_{i=1,\ldots,n} Q(Z_i | pa(Z_i))$

# 1. Variational Inference for probabilistic models

## Variational Inference : Mean Field Approximation

### Variational Inference (VI)

(i) Select a family of distributions $\mathcal{Q}$ on $Z = (Z_1, \ldots, Z_d)$

(ii) Find the **best** approximation : $\hat{P}_X = \arg\min_{Q \in \mathcal{Q}} D_{KL}(Q || P_X)$

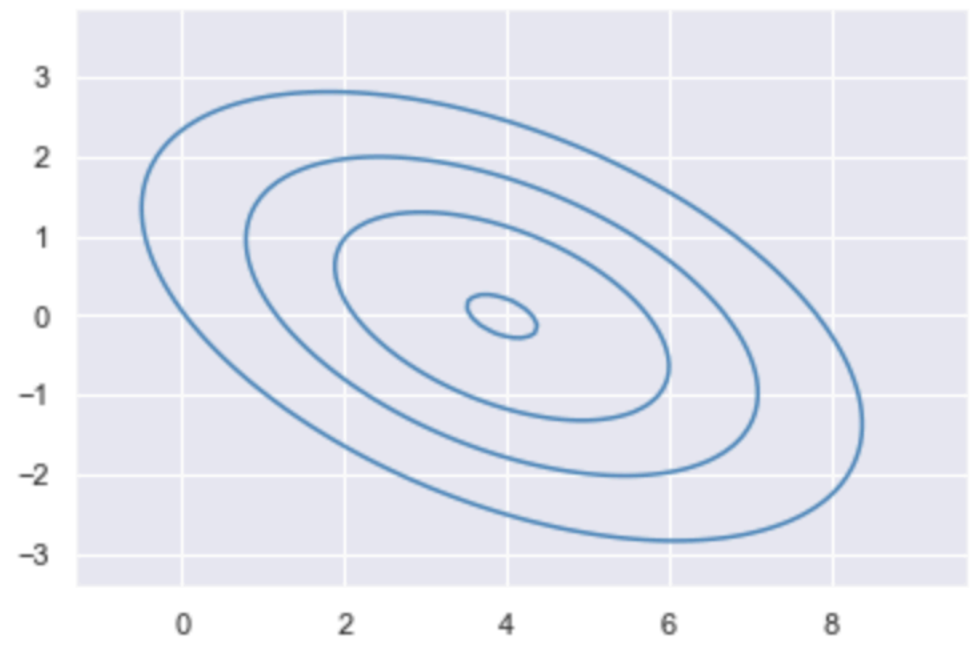
### Mean Field Approximation

(i) we choose $\mathcal{Q} = \left\{ Q = (Q_1, \ldots, Q_d) : Q(Z) = \prod_{i=1,\ldots,d} Q_i(Z_i) \right\}$

instead of $Q(Z_1, \ldots, Z_n) = \prod_{i=1,\ldots,n} Q(Z_i | pa(Z_i))$

### Example : Normal distribution

$P(z) = P(z_1, z_2) = \mathcal{N}_2(z | \mu, \Sigma)$



# 1. Variational Inference for probabilistic models

## Variational Inference : Mean Field Approximation

### Variational Inference (VI)

(i) Select a family of distributions $\mathcal{Q}$ on $Z = (Z_1, \ldots, Z_d)$

(ii) Find the **best** approximation : $\hat{P}_X = \arg\min_{Q \in \mathcal{Q}} D_{KL}(Q || P_X)$

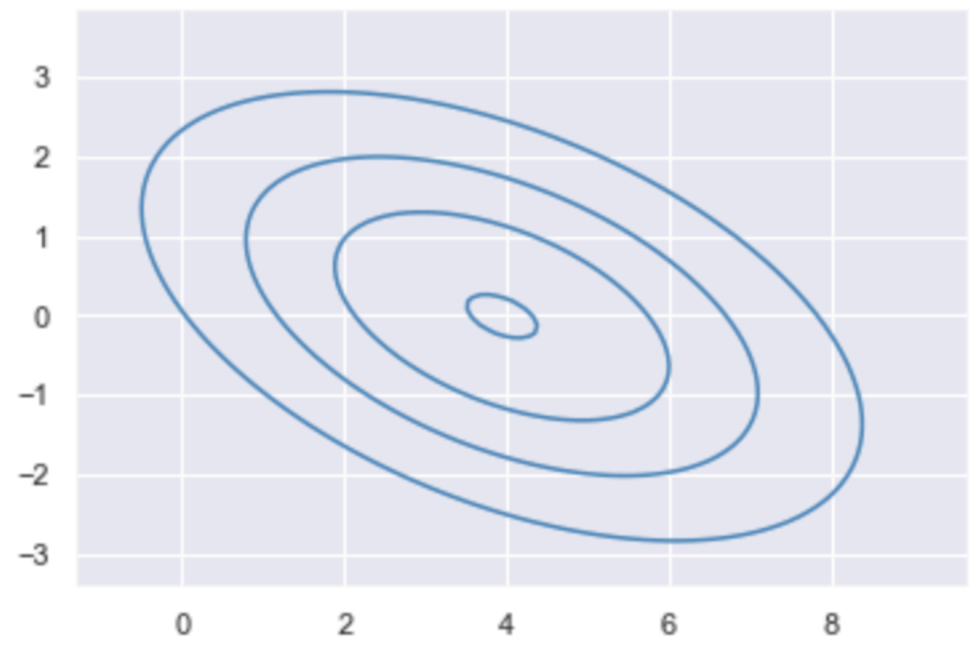
### Mean Field Approximation

(i) we choose $\mathcal{Q} = \left\{ Q = (Q_1, \ldots, Q_d) : Q(Z) = \prod_{i=1,\ldots,d} Q_i(Z_i) \right\}$

instead of $Q(Z_1, \ldots, Z_n) = \prod_{i=1,\ldots,n} Q(Z_i | pa(Z_i))$

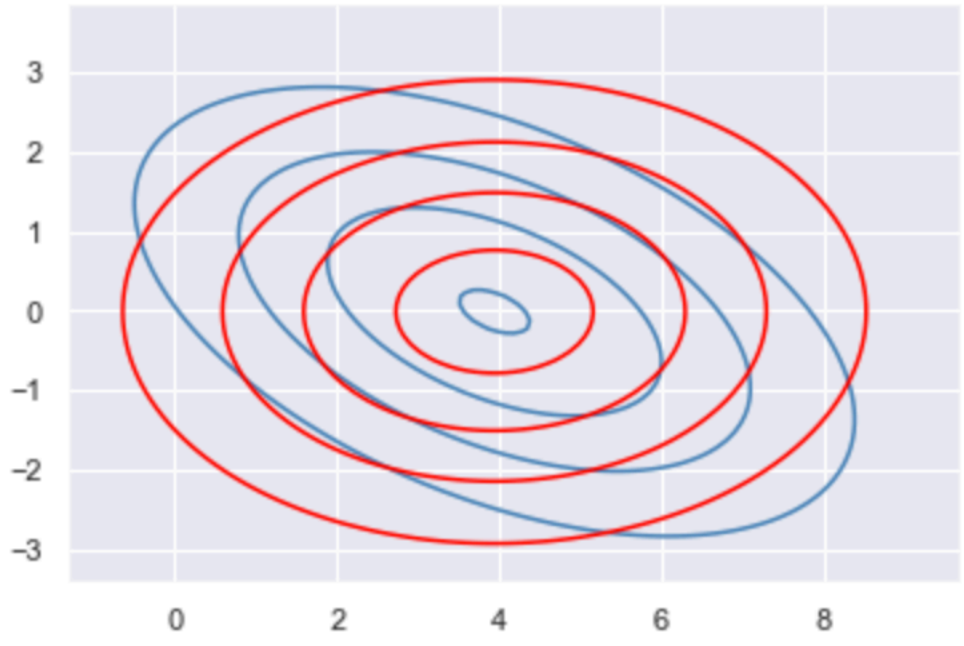
### Example : Normal distribution

$P(z) = P(z_1, z_2) = \mathcal{N}_2(z | \mu, \Sigma)$




**Mean Field** →

$P(z_1, z_2) \approx Q(z_1, z_2) = Q_1(z_1) \times Q_2(z_2)$ with $Q_i(z_i) = \mathcal{N}(z_i | \mu_i, \sigma_i^2)$



# 1. Variational Inference for probabilistic models

## Variational Inference : Mean Field Approximation

### Variational Inference (VI)

(i) Select a family of distributions $\mathcal{Q}$ on $Z = (Z_1, \ldots, Z_d)$

(ii) Find the **best** approximation : $\hat{P}_X = \underset{Q \in \mathcal{Q}}{\arg\min}\, D_{KL}(Q || P_X)$

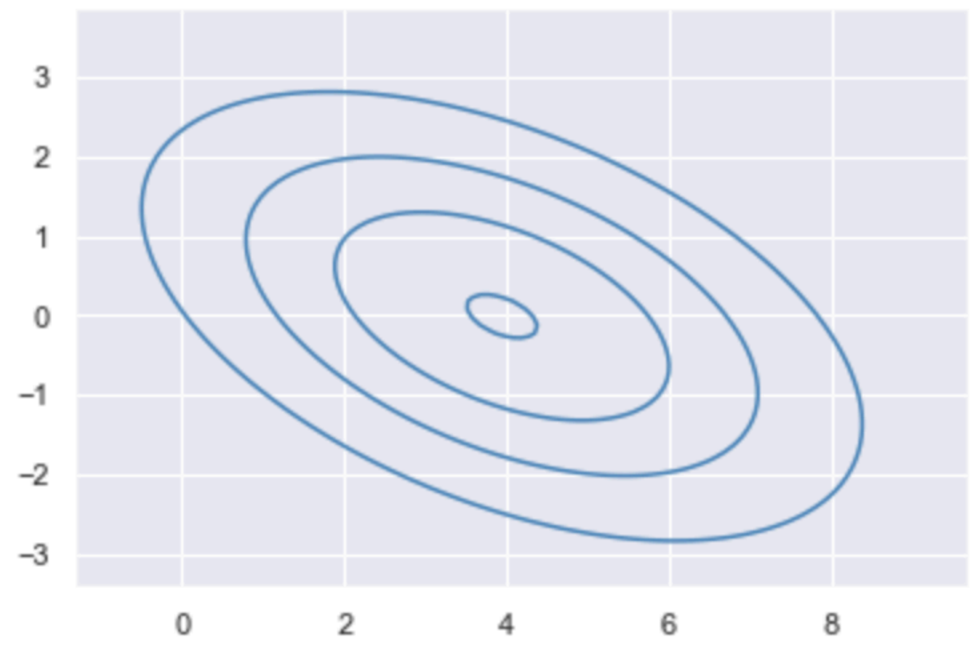
### Mean Field Approximation

(i) we choose $\mathcal{Q} = \left\{ Q = (Q_1, \ldots, Q_d) : Q(Z) = \prod_{i=1,\ldots,d} Q_i(Z_i) \right\}$

instead of $Q(Z_1, \ldots, Z_n) = \prod_{i=1,\ldots,n} Q(Z_i | pa(Z_i))$

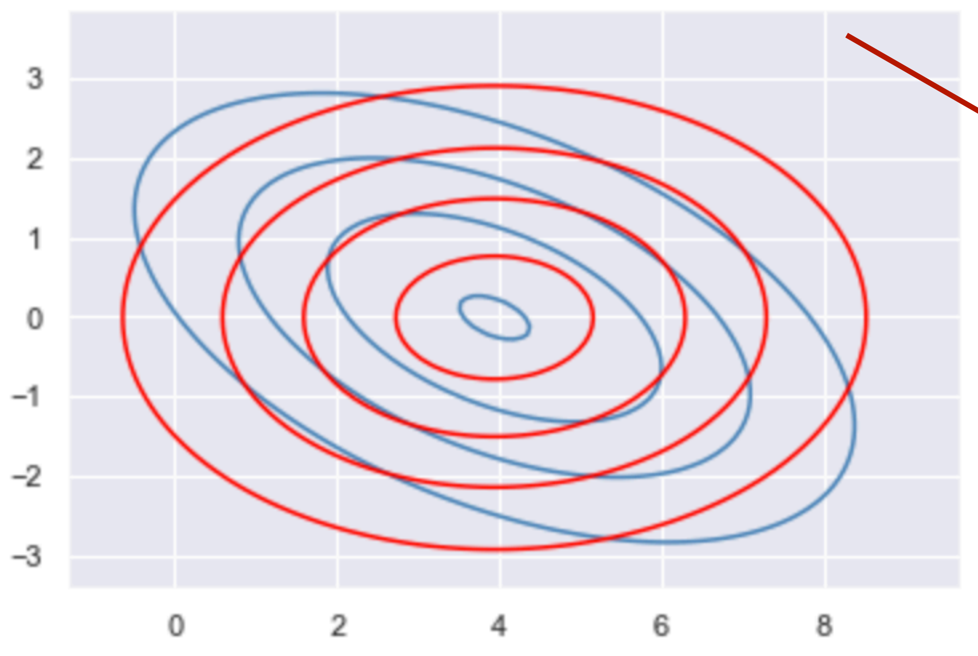
### Example : Normal distribution

$P(z) = P(z_1, z_2) = \mathcal{N}_2(z | \mu, \Sigma)$



**Mean Field** →

$P(z_1, z_2) \approx Q(z_1, z_2) = Q_1(z_1) \times Q_2(z_2)$ with $Q_i(z_i) = \mathcal{N}(z_i | \mu_i, \sigma_i^2)$



$\mathcal{N}_2\left(\begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}, \begin{pmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{pmatrix}\right)$

# 1. Variational Inference for probabilistic models

## Variational Inference : Mean Field Approximation

**Optimization algorithm : coordinate descent**

$$\hat{P} = \arg \min_{(Q_1,\ldots,Q_d)\in\mathbb{Q}} D_{KL}(Q_1 \times Q_2 \times Q_3 \times \cdots \times Q_d \,||\, P)$$

# 1. Variational Inference for probabilistic models

## Variational Inference : Mean Field Approximation

**Optimization algorithm : coordinate descent**

$$\hat{P} = \arg \min_{(Q_1,\ldots,Q_d)\in\mathcal{Q}} D_{KL}(Q_1 \times Q_2 \times Q_3 \times \cdots \times Q_d \,||\, P)$$

↓ Coordinate descent

$$\hat{P}_1 = \arg \min_{Q_1} D_{KL}(Q_1 \times Q_2 \times \ldots \times Q_d \,||\, P)$$

# 1. Variational Inference for probabilistic models

## Variational Inference : Mean Field Approximation

**Optimization algorithm : coordinate descent**

$$\hat{P} = \arg \min_{(Q_1,\ldots,Q_d)\in\mathcal{Q}} D_{KL}(Q_1 \times Q_2 \times Q_3 \times \cdots \times Q_d \,||\, P)$$

↓ Coordinate descent

$$\hat{P}_1 = \arg\min_{Q_1} D_{KL}(Q_1 \times Q_2 \times \ldots \times Q_d \,||\, P)$$

$$\hat{P}_2 = \arg\min_{Q_2} D_{KL}(\hat{P}_1 \times Q_2 \times \ldots \times Q_d \,||\, P)$$

# 1. Variational Inference for probabilistic models

## Variational Inference : Mean Field Approximation

**Optimization algorithm : coordinate descent**

$$\hat{P} = \arg \min_{(Q_1,\ldots,Q_d)\in\mathcal{Q}} D_{KL}(Q_1 \times Q_2 \times Q_3 \times \cdots \times Q_d \,||\, P)$$

↓ Coordinate descent

$$\hat{P}_1 = \arg\min_{Q_1} D_{KL}(Q_1 \times Q_2 \times \ldots \times Q_d \,||\, P)$$

$$\hat{P}_2 = \arg\min_{Q_2} D_{KL}(\hat{P}_1 \times Q_2 \times \ldots \times Q_d \,||\, P)$$

...

$$\hat{P}_d = \arg\min_{Q_d} D_{KL}(\hat{P}_1 \times \hat{P}_2 \times \ldots \times Q_d \,||\, P)$$

Repeat until convergence with $Q_1, \cdots, Q_d = \hat{P}_1, \cdots, \hat{P}_d$

# 1. Variational Inference for probabilistic models

## Variational Inference : Mean Field Approximation

**Optimization algorithm : coordinate descent**

$$\hat{P} = \arg \min_{(Q_1,\ldots,Q_d)\in\mathcal{Q}} D_{KL}(Q_1 \times Q_2 \times Q_3 \times \cdots \times Q_d \,||\, P)$$

↓ Coordinate descent

$$\hat{P}_1 = \arg\min_{Q_1} D_{KL}(Q_1 \times Q_2 \times \ldots \times Q_d \,||\, P)$$

$$\hat{P}_2 = \arg\min_{Q_2} D_{KL}(\hat{P}_1 \times Q_2 \times \ldots \times Q_d \,||\, P)$$

…

$$\hat{P}_d = \arg\min_{Q_d} D_{KL}(\hat{P}_1 \times \hat{P}_2 \times \ldots \times Q_d \,||\, P)$$

Repeat until convergence with

$$Q_1, \cdots, Q_d = \hat{P}_1, \cdots, \hat{P}_d$$

?

?

# 1. Variational Inference for probabilistic models

## Variational Inference : Mean Field Approximation

**Optimization algorithm : coordinate descent**

$$\hat{P} = \arg \min_{(Q_1,\ldots,Q_d)\in\mathcal{Q}} D_{KL}(Q_1 \times Q_2 \times Q_3 \times \cdots \times Q_d \,||\, P)$$

Coordinate descent

$$\hat{P}_1 = \arg\min_{Q_1} D_{KL}(Q_1 \times Q_2 \times \ldots \times Q_d \,||\, P)$$

$$\hat{P}_2 = \arg\min_{Q_2} D_{KL}(\hat{P}_1 \times Q_2 \times \ldots \times Q_d \,||\, P)$$

...

$$\hat{P}_d = \arg\min_{Q_d} D_{KL}(\hat{P}_1 \times \hat{P}_2 \times \ldots \times Q_d \,||\, P)$$

Repeat until convergence with

$Q_1, \cdots, Q_d$ = $\hat{P}_1, \cdots, \hat{P}_d$

?

**Optimal solution in Mean Field**

$$\log \hat{P}_i(Z_i) = \mathbb{E}_{Z_{-i}}\left[\log P(X, Z)\right] + \text{const}$$

# 1. Variational Inference for probabilistic models

## Variational Inference : Mean Field Approximation

**Optimization algorithm : coordinate descent**

$$\hat{P} = \arg\min_{(Q_1,\dots,Q_d)\in\mathcal{Q}} D_{KL}(Q_1 \times Q_2 \times Q_3 \times \cdots \times Q_d \,||\, P)$$

Coordinate descent

$$\hat{P}_1 = \arg\min_{Q_1} D_{KL}(Q_1 \times Q_2 \times \ldots \times Q_d \,||\, P)$$

$$\hat{P}_2 = \arg\min_{Q_2} D_{KL}(\hat{P}_1 \times Q_2 \times \ldots \times Q_d \,||\, P)$$

...

$$\hat{P}_d = \arg\min_{Q_d} D_{KL}(\hat{P}_1 \times \hat{P}_2 \times \ldots \times Q_d \,||\, P)$$

Repeat until convergence with $Q_1, \cdots, Q_d = \hat{P}_1, \cdots, \hat{P}_d$

**Optimal solution in Mean Field**

$$\log \hat{P}_i(Z_i) = \mathbb{E}_{Z_{-i}}\left[\log P(X,Z)\right] + \text{const}$$

$$\mathbb{E}\left[\log P(X,Z)\right] - \mathbb{E}\left[\log P(Z)\right] \approx 0$$

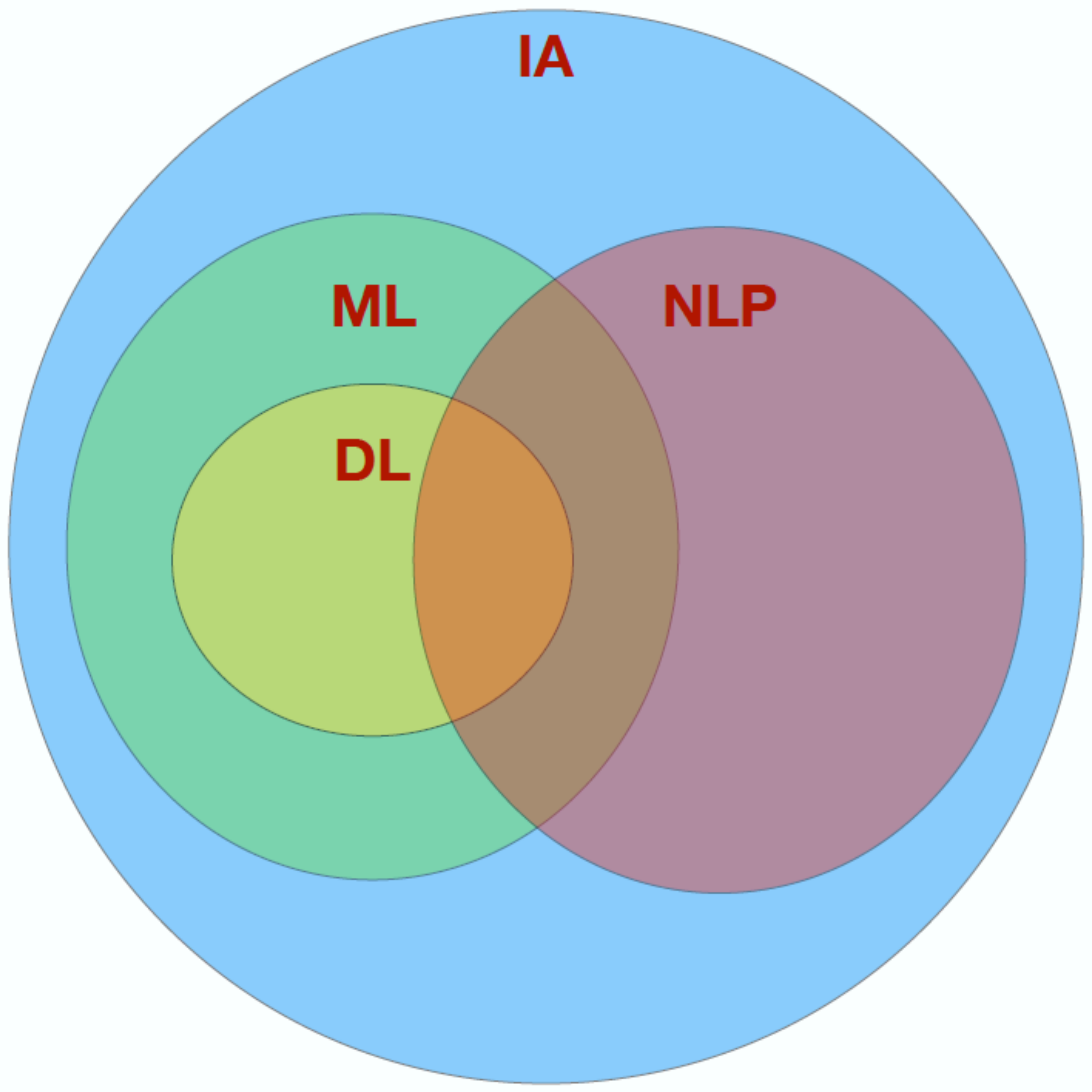
We will see in section 3 an example with the model LDA

# 2 Introduction to NLP

# 2. Introduction to NLP

## Preprocessing : Tokenization

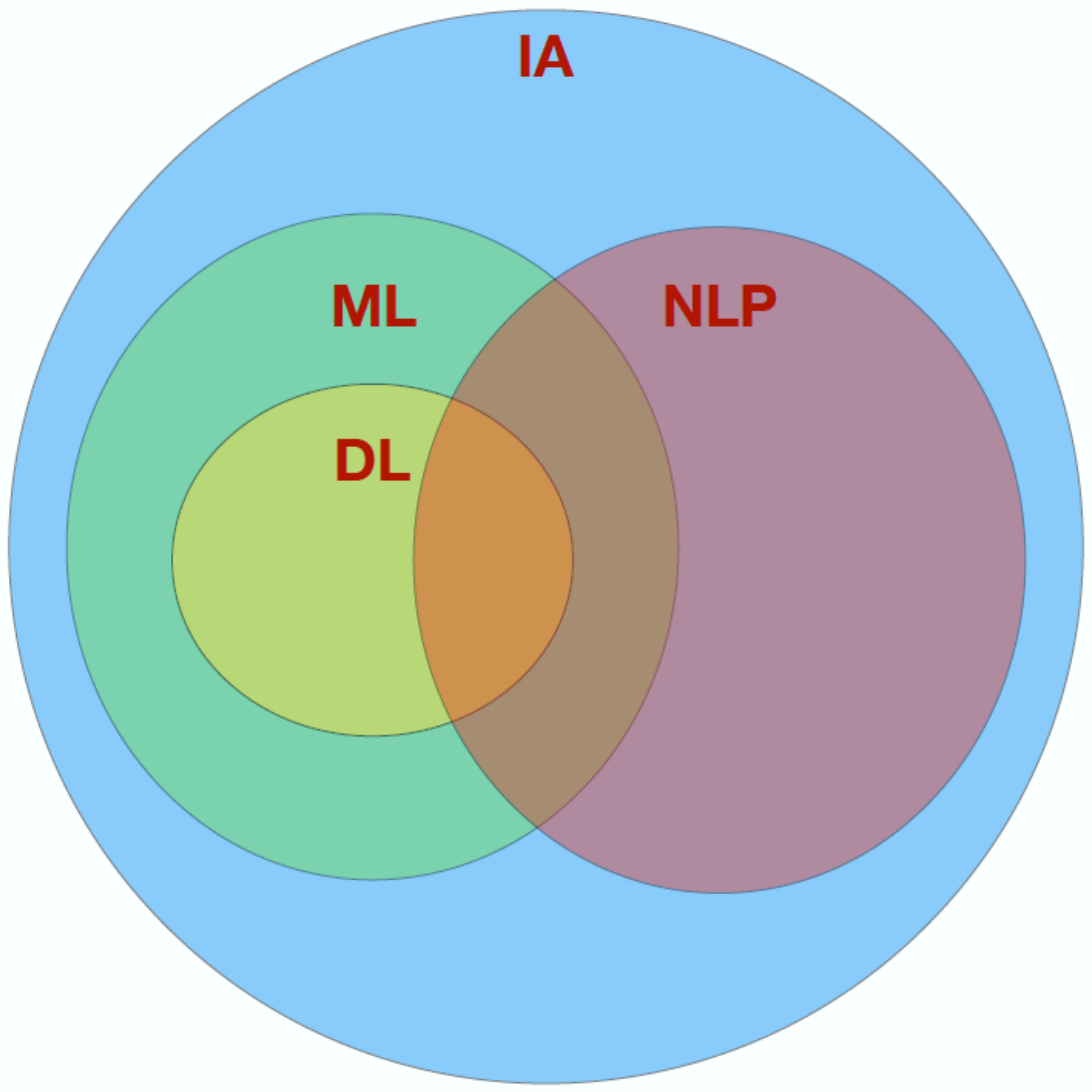
**Natural Language Processing** : The science of programming computers to understand human language



# 2. Introduction to NLP

## Preprocessing : Tokenization

**Natural Language Processing** : The science of programming computers to understand human language



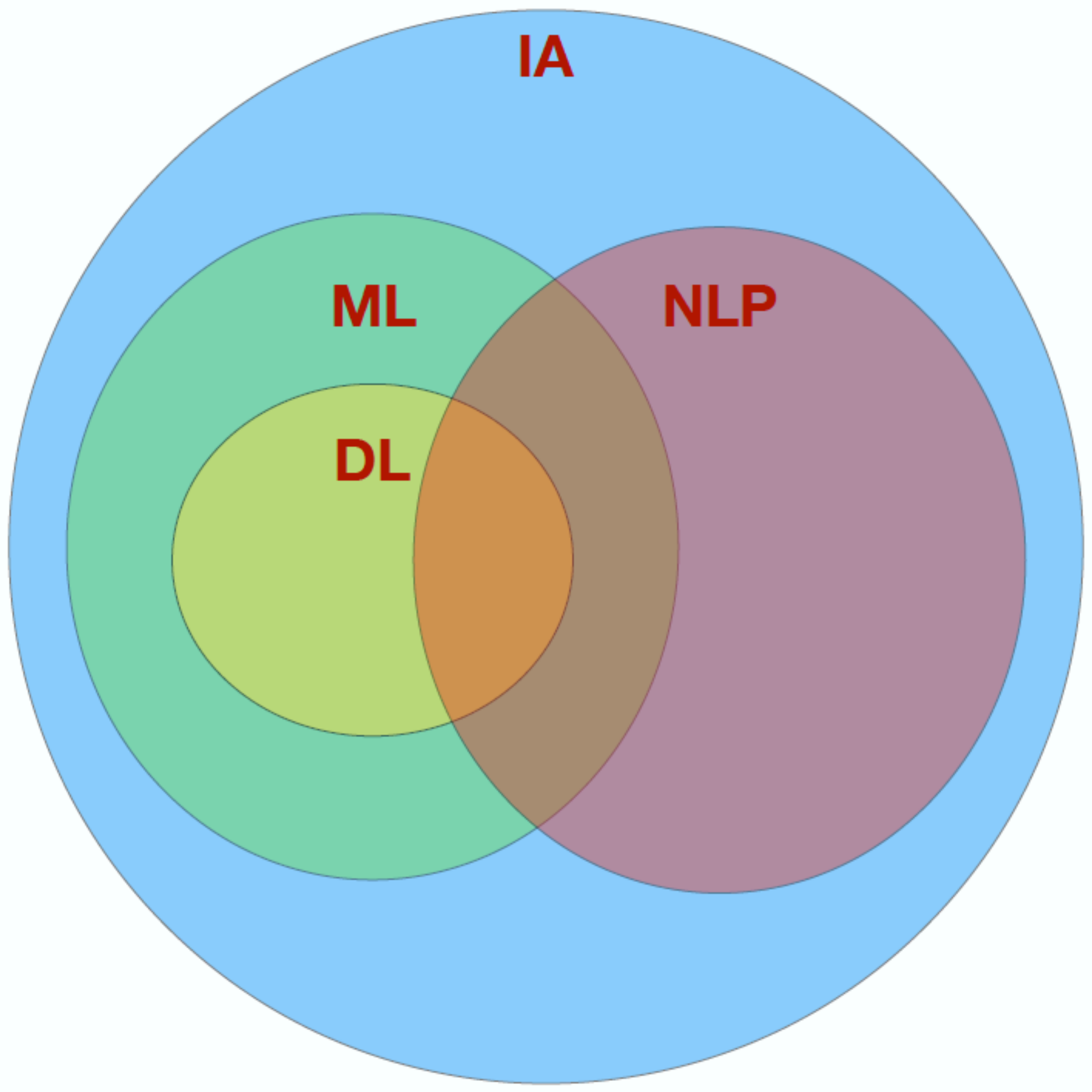

**Some intuitions :** we want to perform some learning tasks with textual data

- We know how to train a model with a tabular data. **How about textual data ?**
- Textual data can be highly sophisticated. **Can we simplify them ?**

# 2. Introduction to NLP

## Preprocessing : Tokenization

**Natural Language Processing** : The science of programming computers to understand human language




**Some intuitions :** we want to perform some learning tasks with textual data

- We know how to train a model with a tabular data. **How about textual data ?**
- Textual data can be highly sophisticated. **Can we simplify them ?**

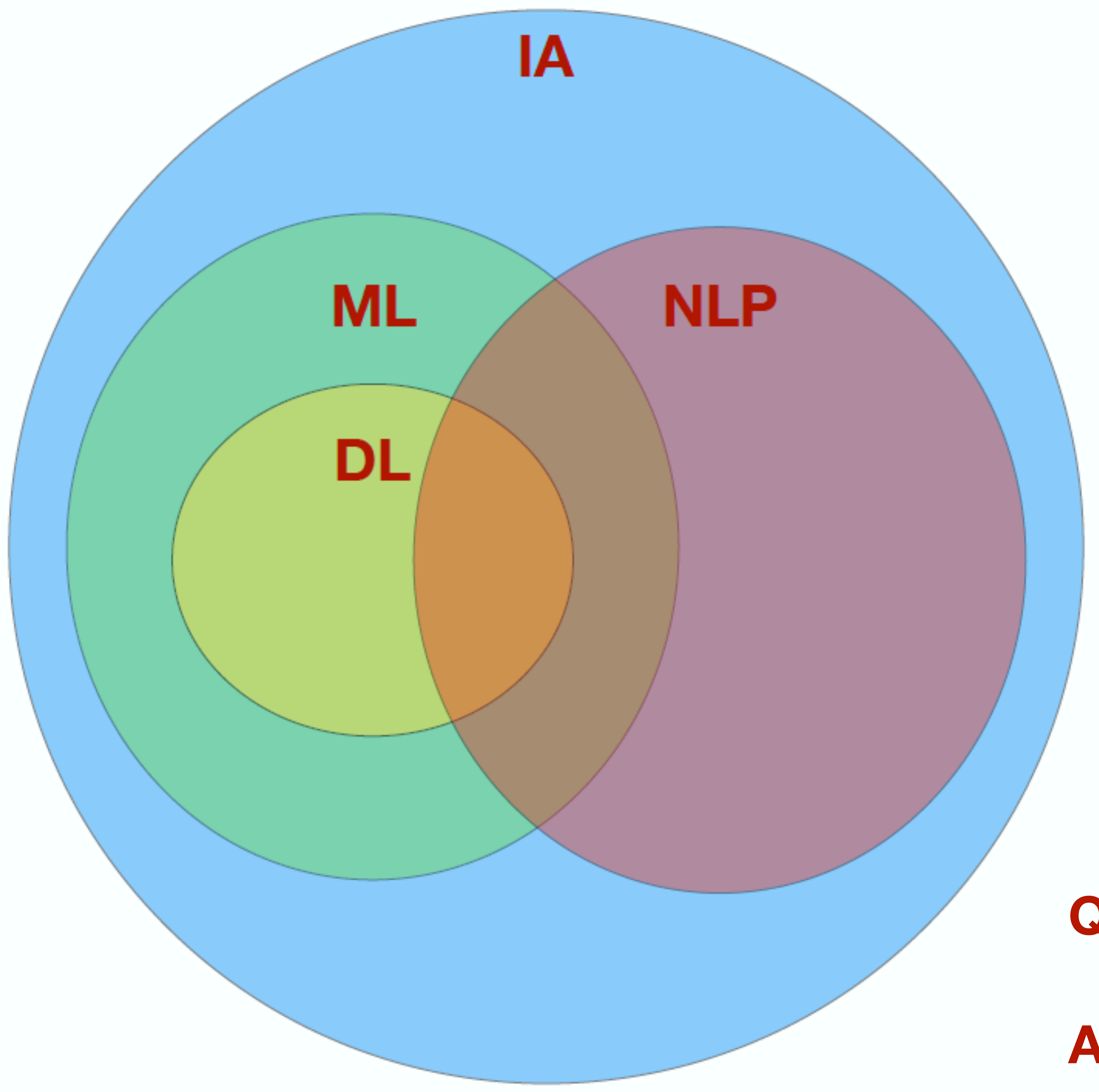
**Definitions**

- **Text** : sequence of words
- **Word** : sequence of logical characters
- **Tokenization** : process that separates a sequence (text) into a list of tokens (words

# 2. Introduction to NLP

## Preprocessing : Tokenization

**Natural Language Processing** : The science of programming computers to understand human language




**Some intuitions :** we want to perform some learning tasks with textual data

- We know how to train a model with a tabular data. **How about textual data ?**
- Textual data can be highly sophisticated. **Can we simplify them ?**

**Definitions**

- **Text** : sequence of words
- **Word** : sequence of logical characters
- **Tokenization** : process that separates a sequence (text) into a list of tokens (words

**Question :** how to find the limits of a word?

**Answer** : In French/English, we can separate words by spaces and punctuation

**Example** : **When should I start my job search ?** → **['When', 'should', 'I', 'start', 'my', 'job', 'search']**

# 2. Introduction to NLP

## Preprocessing : Normalization & stop-words

**Stemming** : keep the root of a term by cutting off the end or the beginning of the word

**Example :** wait, wait**ing**, wait**ed**, wait**s** ⟶ wait

**there exists many text-preprocessing packages in python :** nltk, spacy, …

# 2. Introduction to NLP

## Preprocessing : Normalization & stop-words

**Stemming** : keep the root of a term by cutting off the end or the beginning of the word

**Example :** wait, waiting, waited, waits → wait

**Lemmatization** : keep the root of a term by transforming the words into its root words

**Example :** study, studying, studies → study **(In stemming :** stud)

**there exists many text-preprocessing packages in python :** nltk, spacy, …

# 2. Introduction to NLP

## Preprocessing : Normalization & stop-words

**Stemming** : keep the root of a term by cutting off the end or the beginning of the word

**Example :** wait, wait**ing**, wait**ed**, wait**s** → wait

**Lemmatization** : keep the root of a term by transforming the words into its root words

**Example :** study, study**ing**, studi**es** → study **(In stemming :** stud)

**Stop-words** : set of words frequently used in a language and which do not bring any important meaning

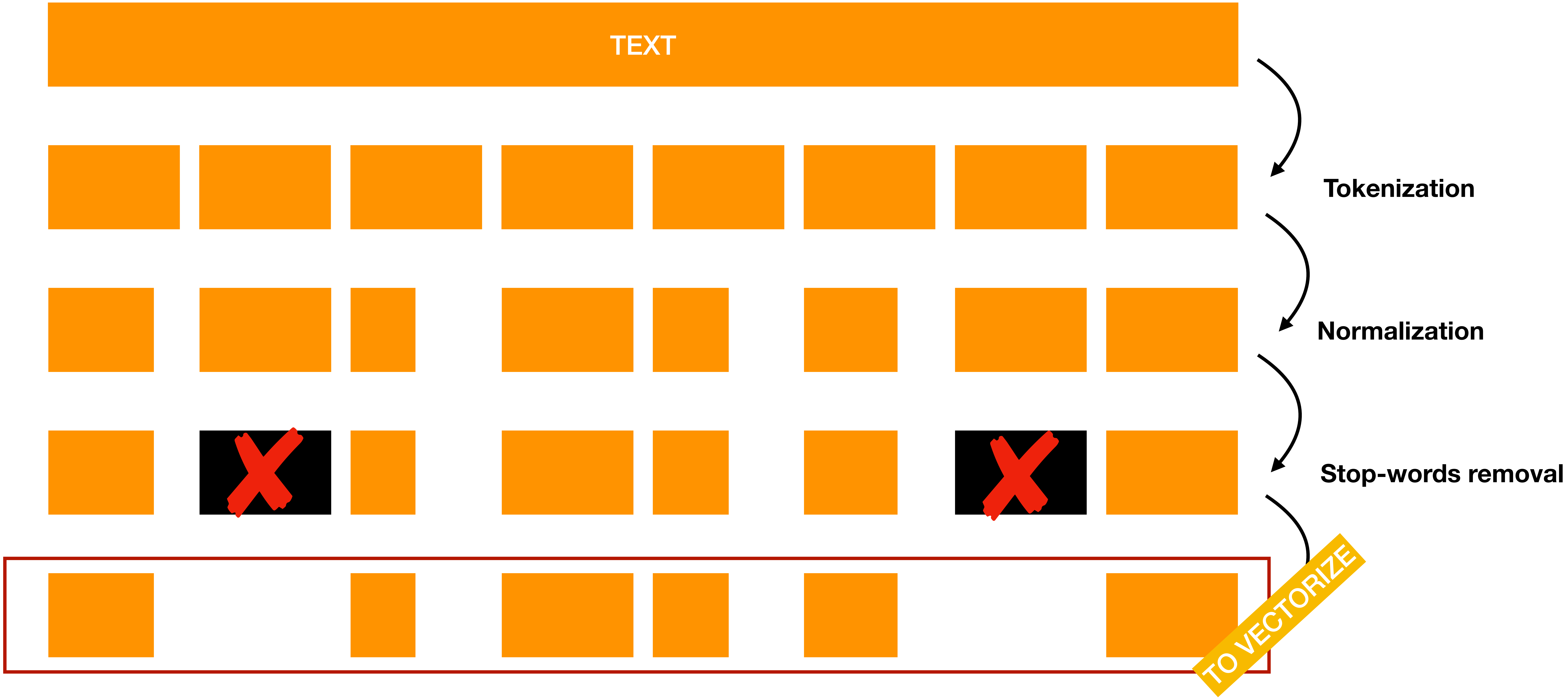
**Example :** the, a, of, is, at, which, …

**Aim :** Remove these stop-words

**there exists many text-preprocessing packages in python :** nltk, spacy, …

# 2. Introduction to NLP

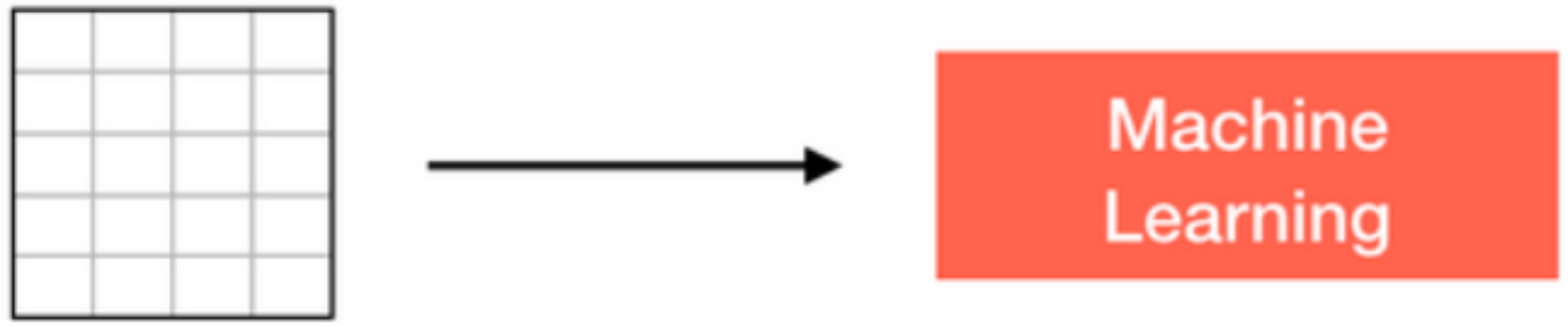
## Preprocessing : overview



# 2. Introduction to NLP

## Processing : Textual data into tabular data

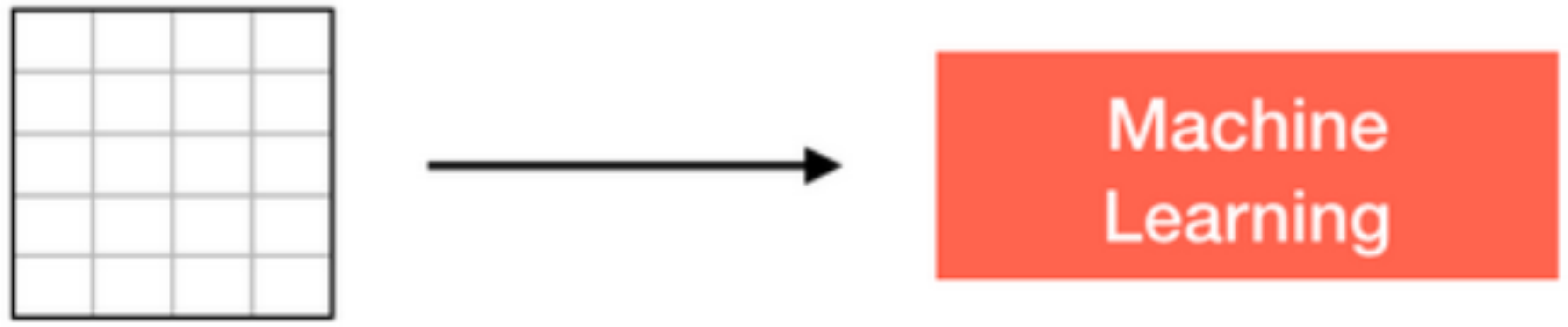
**For tabular data :**



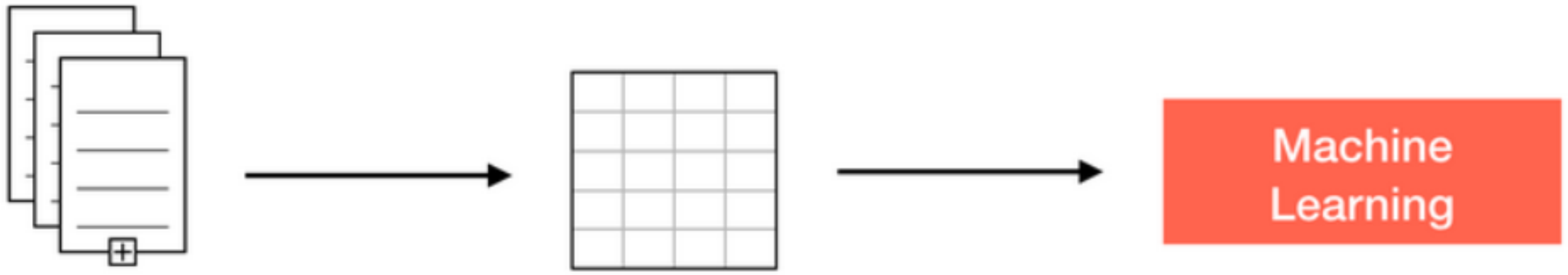
# 2. Introduction to NLP

## Processing : Textual data into tabular data

**For tabular data :**



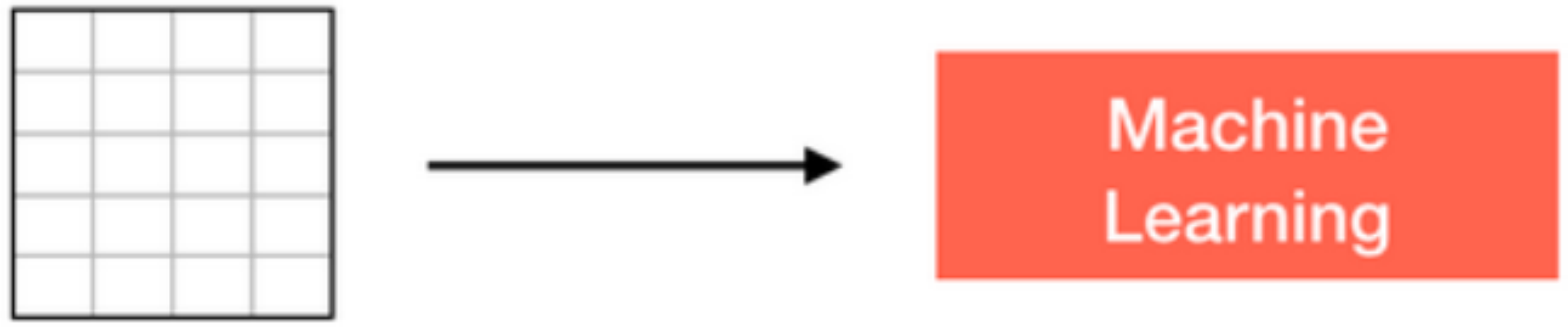

**For textual data :**



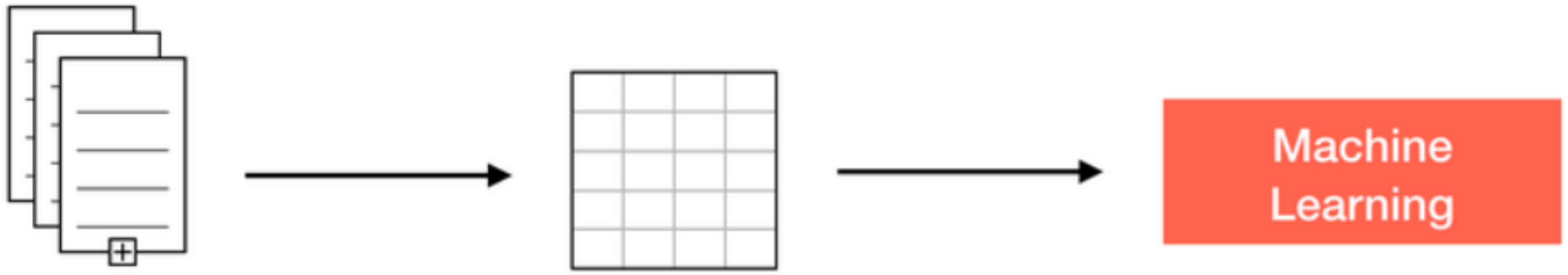
# 2. Introduction to NLP

## Processing : Textual data into tabular data

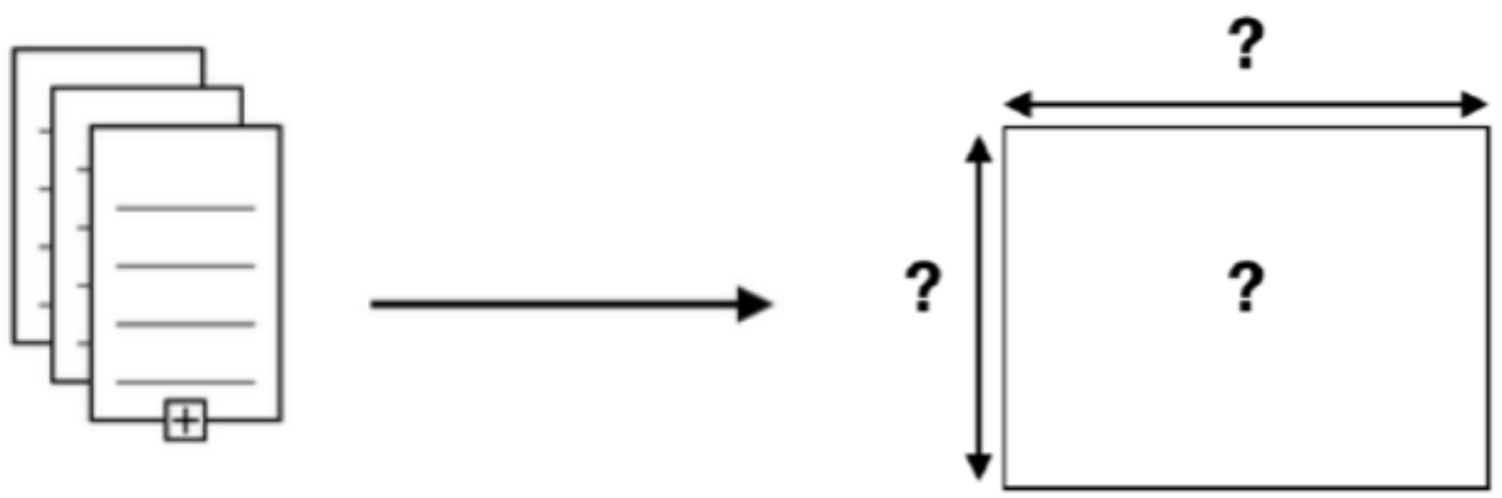
**For tabular data :**




**For textual data :**




**Problems :**

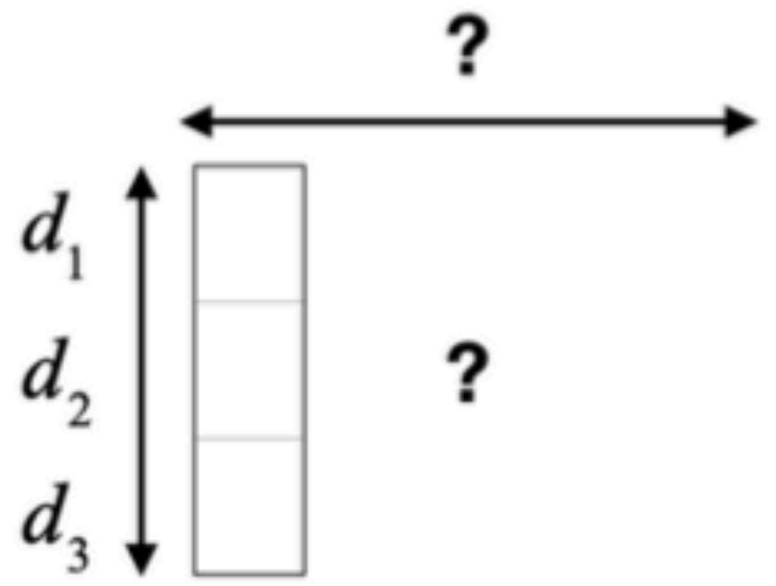


# 2. Introduction to NLP

## Processing : Textual data into tabular data

**Corpus**

| | |
|---|---|
| $d_1$ | trouver bonne assurance |
| $d_2$ | contrat satisfaisant |
| $d_3$ | changement contrat assurance |



# 2. Introduction to NLP

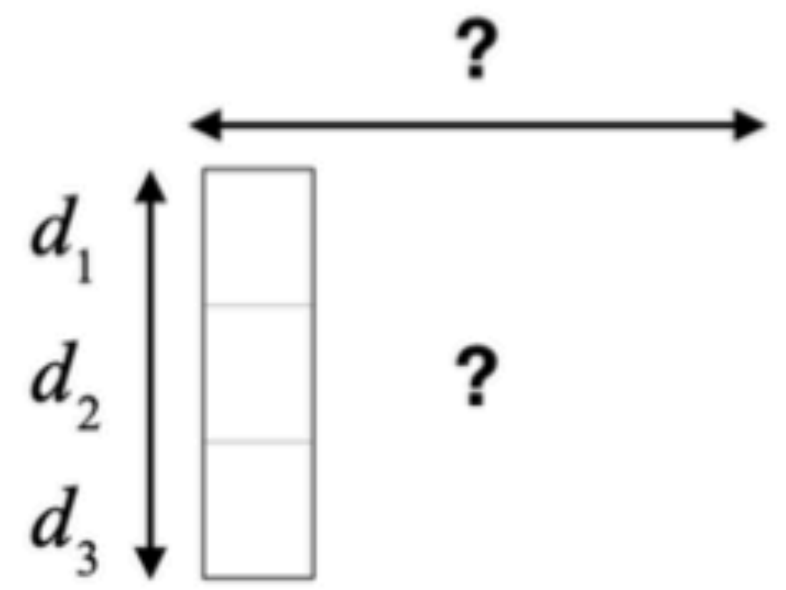
## Processing : Textual data into tabular data

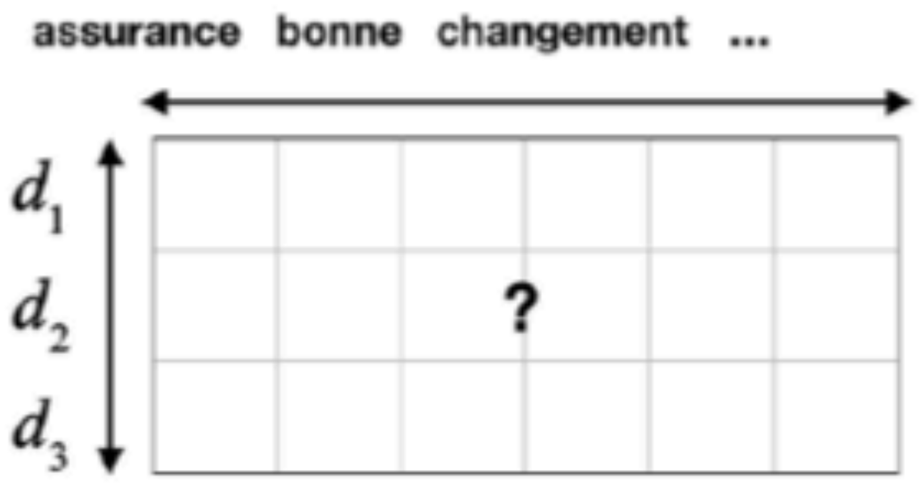
Corpus

Dictionary

| | |
|---|---|
| $d_1$ | trouver bonne assurance |
| $d_2$ | contrat satisfaisant |
| $d_3$ | changement contrat assurance |

V = {
'assurance' : 1,
'bonne' : 2,
'changement' : 3,
'contrat' : 4,
'satisfaisant' : 5,
'trouver' : 6
}



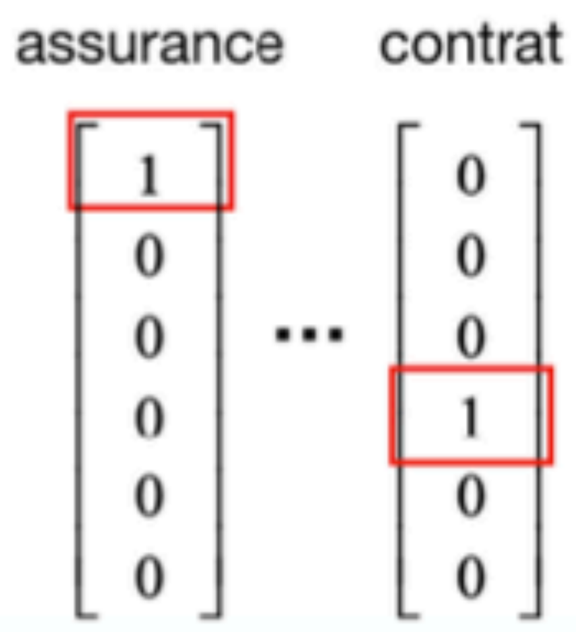
# 2. Introduction to NLP

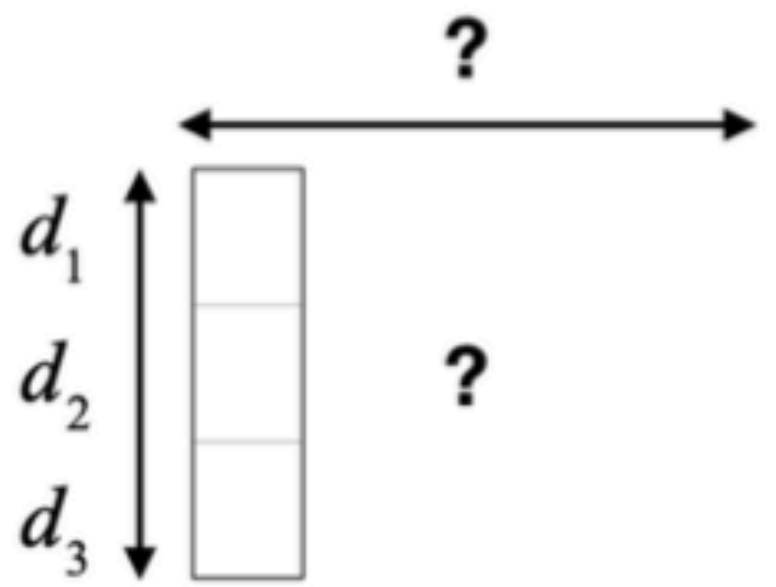
## Processing : Textual data into tabular data

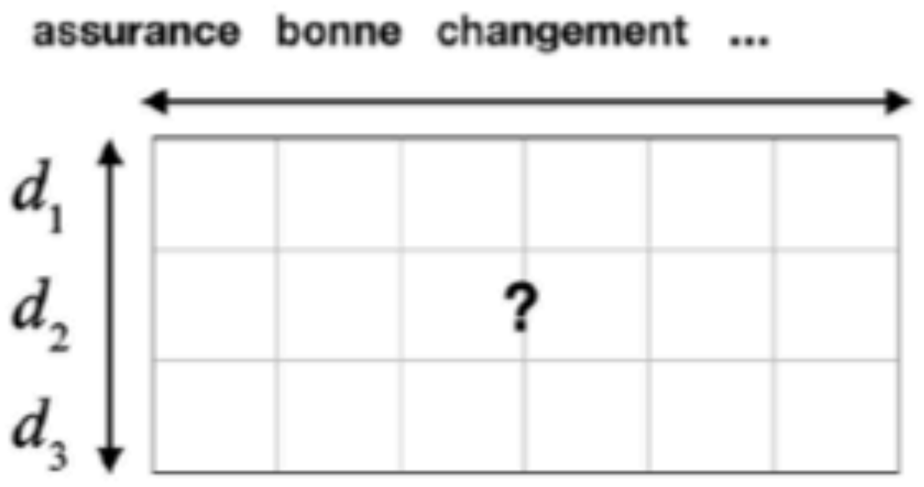
**Corpus** → **Dictionary** → **Encoding**

| | |
|---|---|
| $d_1$ | trouver bonne assurance |
| $d_2$ | contrat satisfaisant |
| $d_3$ | changement contrat assurance |

V = {
'assurance' : 1,
'bonne' : 2,
'changement' : 3,
'contrat' : 4,
'satisfaisant' : 5,
'trouver' : 6
}




| | assurance | bonne | changement | ... | | |
|---|---|---|---|---|---|---|
| $d_1$ | 1 | 1 | 0 | 0 | 0 | 1 |
| $d_2$ | 0 | 0 | 0 | 1 | 1 | 0 |
| $d_3$ | 1 | 0 | 1 | 1 | 0 | 0 |

**Bag-of-Words approach**

# 2. Introduction to NLP

## Some important considerations on vectorization

| trouver | contrat | assurance | ... |
|---|---|---|---|
| 1 | 0 | 1 | ... |
| 0 | 1 | 0 | ... |
| 0 | 1 | 1 | ... |

### Bag-of-Words (BoW) approach

- based on term frequency
- problem : don’t keep the word orders
- solution : n-grams approach

| trouver | assurance | contrat assurance | ... |
|---|---|---|---|
| 1 | 1 | 0 | ... |
| 0 | 0 | 0 | ... |
| 0 | 1 | 1 | ... |

### n-grams approach

- based on sequence of *n* words frequency
- problem : too many features / too sparse
- solution : stop-words and some n-grams removal (too **high** or too **low** frequencies)

| trouver | assurance | contrat assurance | ... |
|---|---|---|---|
| 0.10 | 0.41 | 0 | ... |
| 0 | 0 | 0 | ... |
| 0 | 0.41 | 0.10 | ... |

### TF-IDF approach

- Based on the product of two values :
  - **Term frequency (TF) :**

$$\mathrm{TF}(\mathrm{t}, \mathrm{d}) = \text{frequency of t in d}$$

  - **Inverse Document Frequency (IDF):**

$$IDF(t, D) = \log \frac{\text{\#documents}}{\text{\#documents with terme t}}$$

## 3 Application on textual data with LDA

# 3. Latent Dirichlet Allocation

## Topic modeling

**Topic modeling** : a statistical model for **finding out the hidden « topics »** that occur in a collection of documents

# 3. Latent Dirichlet Allocation

## Topic modeling

**Topic modeling** : a statistical model for **finding out the hidden « topics »** that occur in a collection of documents

**Motivations :** This method is also used in

- create **recommendation systems** (used by e-tailers, search engines, …)
- text **categorization**
- **data mining** processes
- in bioinformatics: **extracting hidden knowledge** from biological data (DNA molecules)

# 3. Latent Dirichlet Allocation

## Topic modeling

**Topic modeling** : a statistical model for **finding out the hidden « topics »** that occur in a collection of documents

**Motivations :** This method is also used in

- create **recommendation systems** (used by e-tailers, search engines, …)
- text **categorization**
- **data mining** processes
- in bioinformatics: **extracting hidden knowledge** from biological data (DNA molecules)

# 3. Latent Dirichlet Allocation
## Topic modeling

**Topic modeling** : a statistical model for **finding out the hidden « topics »** that occur in a collection of documents

**Motivations :** This method is also used in

- create **recommendation systems** (used by e-tailers, search engines, …)
- text **categorization**
- **data mining** processes
- in bioinformatics: **extracting hidden knowledge** from biological data (DNA molecules)

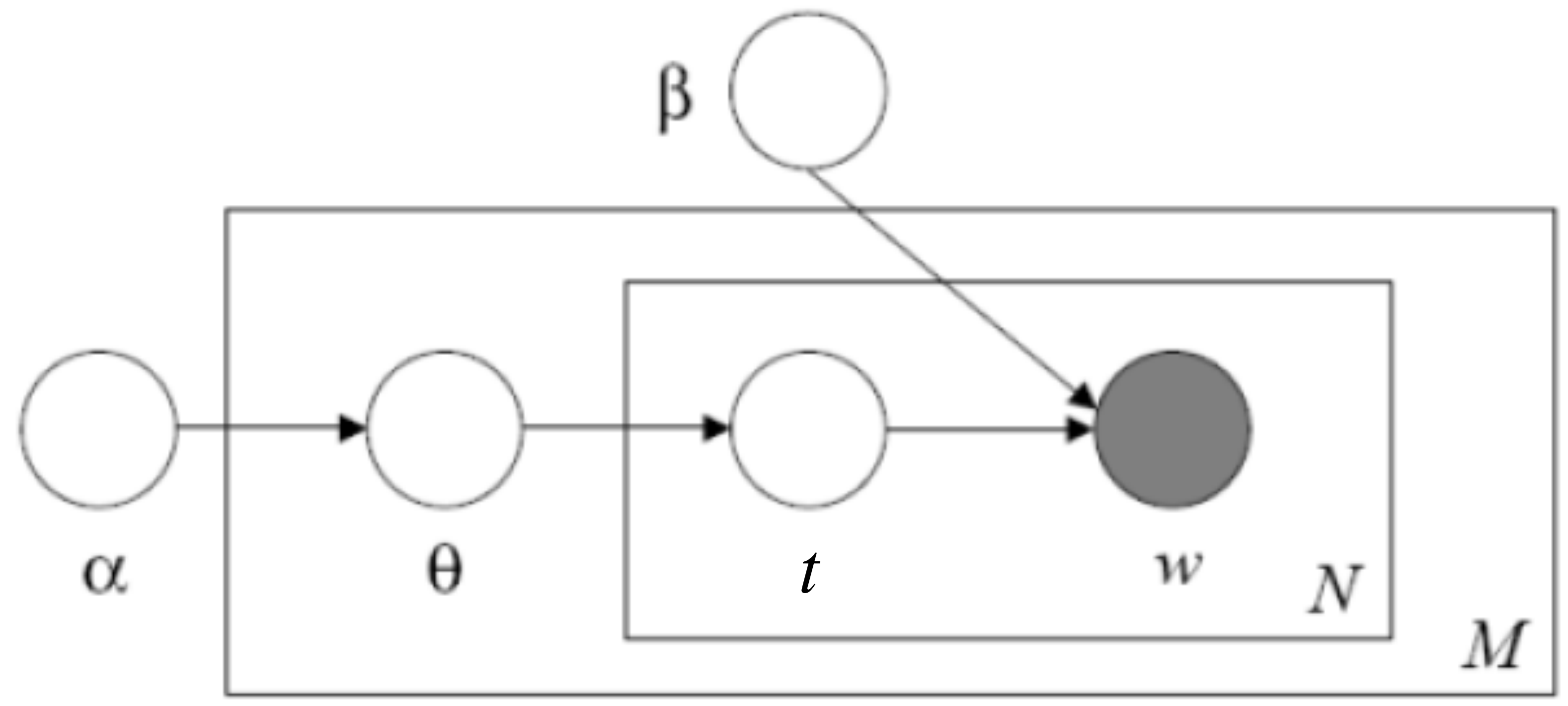

**Idea :**

- Every **document** consists of a mix of **topics**
- Every **topics** consists of a mix of **words**

# 3. Latent Dirichlet Allocation

## LDA : high-level view

**Latent Dirichlet Allocation (LDA)** : (popular) topic modeling based on Bayesian inference with the following PGM



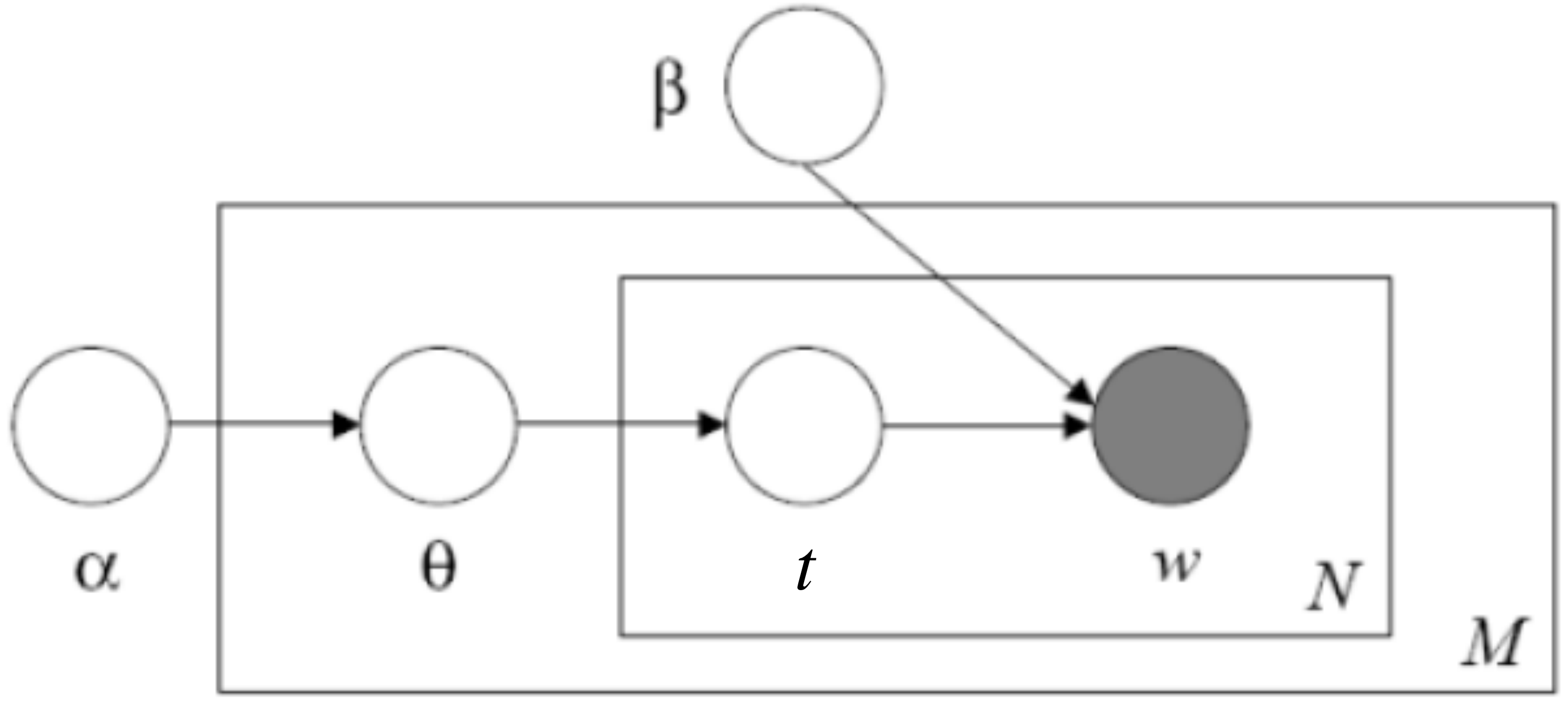

$$P(\theta, t, w \mid \alpha, \beta) = P(\theta \mid \alpha) \cdot P(t \mid \theta) \cdot P(w \mid t, \beta)$$

$$= \prod_{d \in [M]} P(\theta_d \mid \alpha) \cdot \prod_{n \in [N]} P(t_{d,n} \mid \theta_d) \cdot P(w_{d,n} \mid t_{d,n}, \beta)$$

# 3. Latent Dirichlet Allocation

## LDA : high-level view

**Latent Dirichlet Allocation (LDA)** : (popular) topic modeling based on Bayesian inference with the following PGM




~ Dirichlet ~ Multinomial ~ Multinomial

$$P(\theta, t, w \mid \alpha, \beta) = P(\theta \mid \alpha) \cdot P(t \mid \theta) \cdot P(w \mid t, \beta)$$

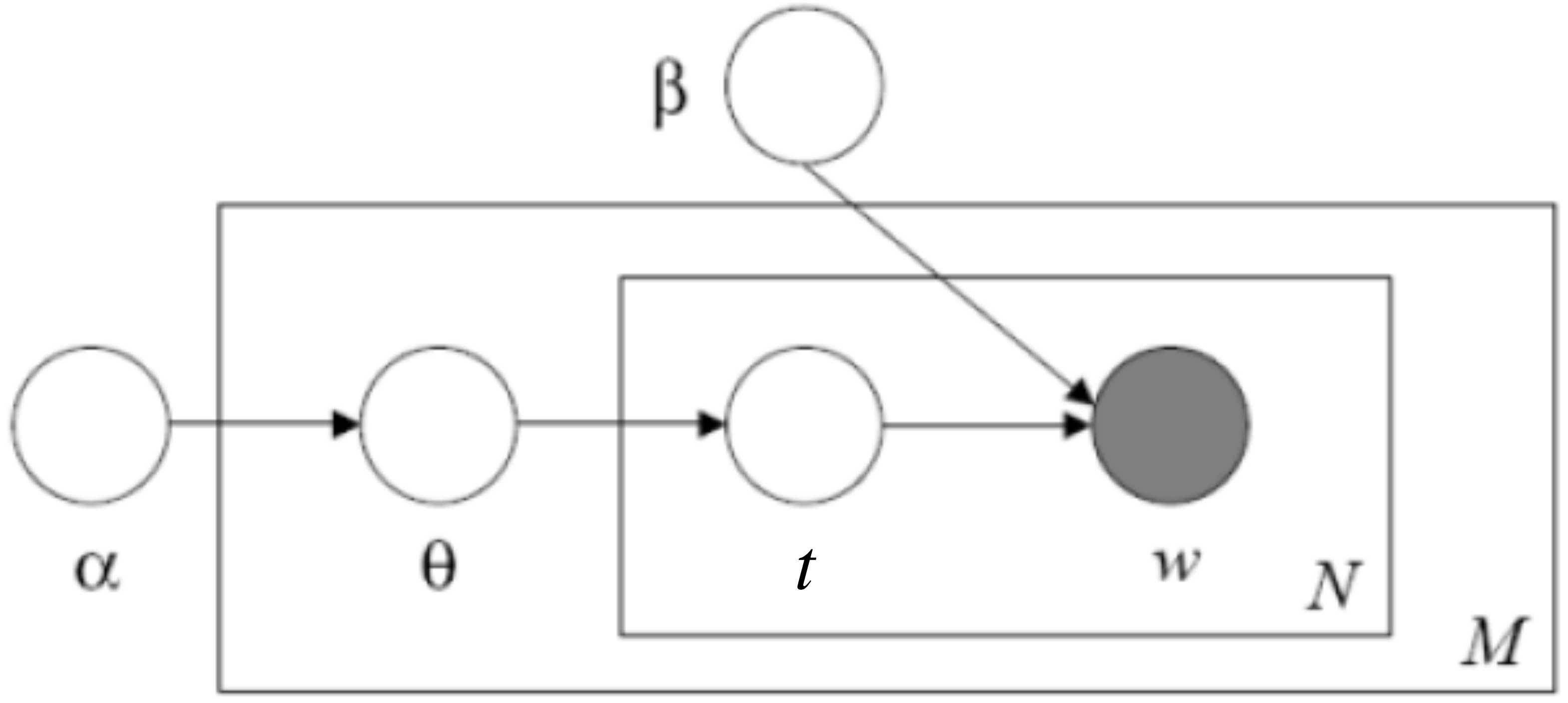
$$= \prod_{d \in [M]} P(\theta_d \mid \alpha) \cdot \prod_{n \in [N]} P(t_{d,n} \mid \theta_d) \cdot P(w_{d,n} \mid t_{d,n}, \beta)$$

~ Dirichlet ~ Multinomial ~ Multinomial

# 3. Latent Dirichlet Allocation

## LDA : high-level view

**Latent Dirichlet Allocation (LDA)** : (popular) topic modeling based on Bayesian inference with the following PGM



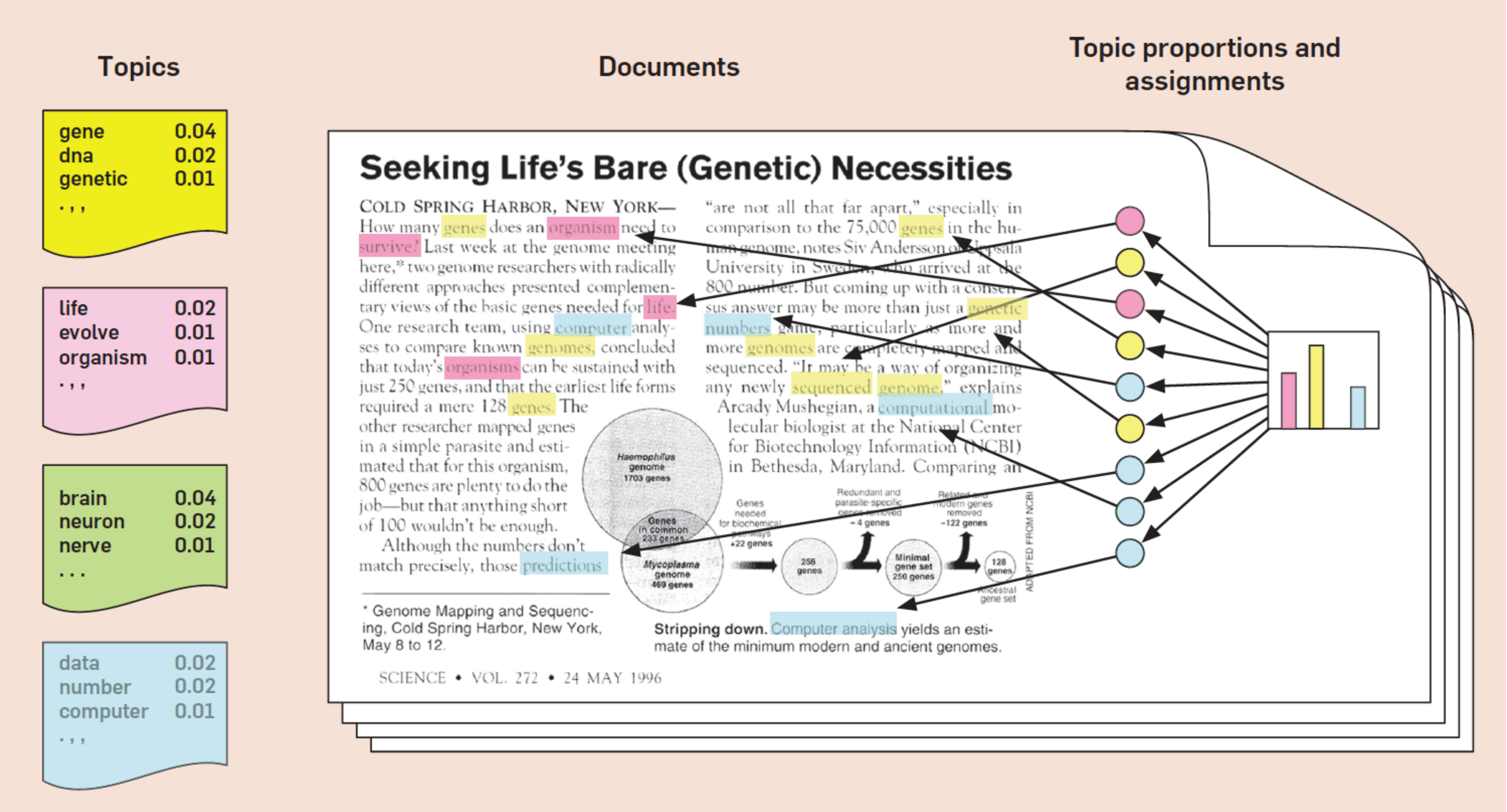

~ Dirichlet ~ Multinomial ~ Multinomial

$$P(\theta, t, w \mid \alpha, \beta) = P(\theta \mid \alpha) \cdot P(t \mid \theta) \cdot P(w \mid t, \beta)$$

$$= \prod_{d \in [M]} P(\theta_d \mid \alpha) \cdot \prod_{n \in [N]} P(t_{d,n} \mid \theta_d) \cdot P(w_{d,n} \mid t_{d,n}, \beta)$$

~ Dirichlet ~ Multinomial ~ Multinomial

**Example :**




**source** : Blei, D.M. (2012). Probabilistic topic models. Communications of the ACM, 55(4), 77-84.

**Assumption on the generation of texts :**

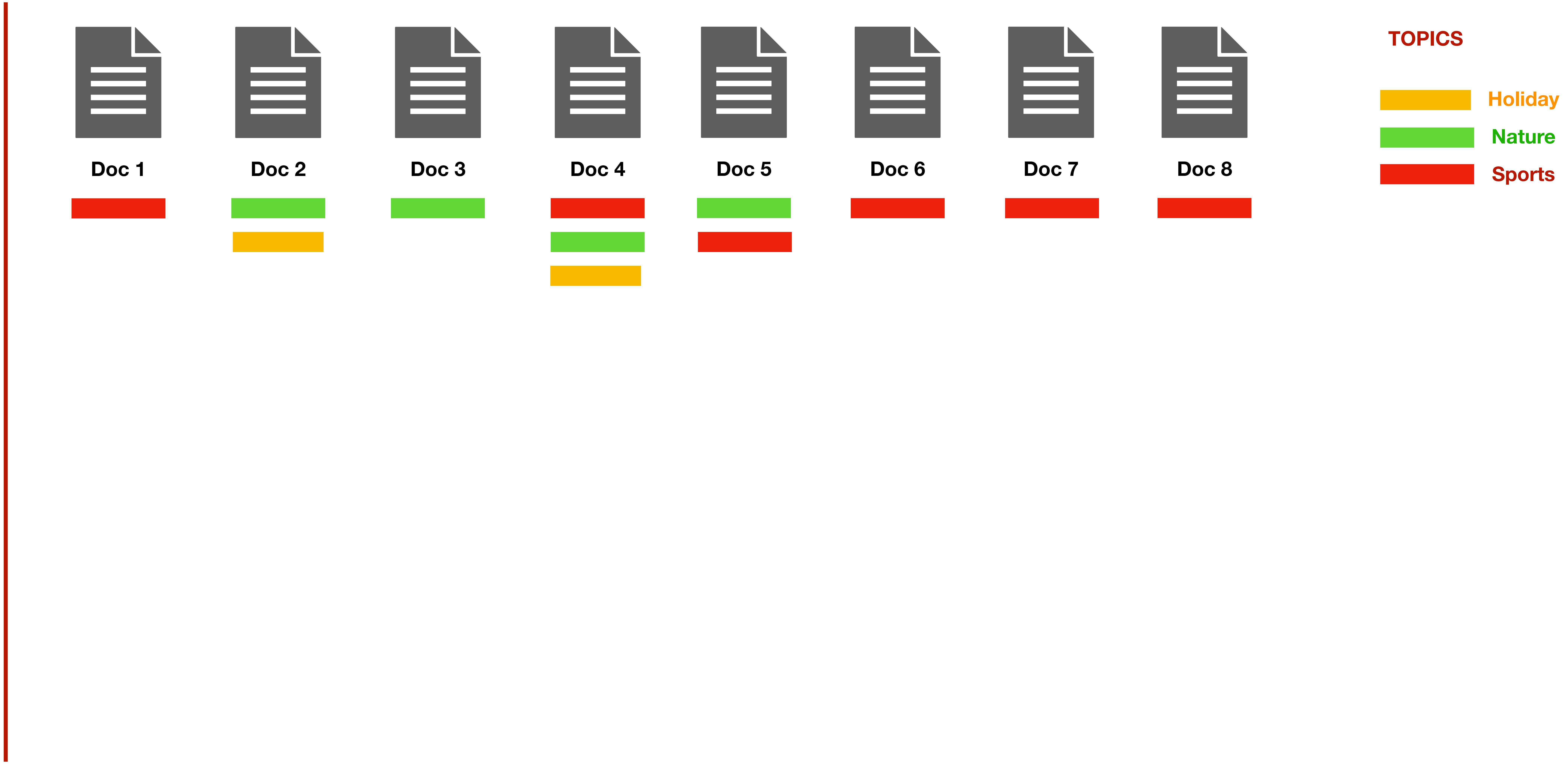
For each of M documents $d$,

- Choose the **topic distribution** $\theta_d \sim \text{Dirichlet}(\alpha)$
- For each of N words $w$,
  - choose a **topic** $t \sim \text{Multinomial}(\theta_d)$
  - choose a **word** $w \sim \text{Multinomial}(\beta)$

# 3. Latent Dirichlet Allocation

## LDA : Dirichlet distribution

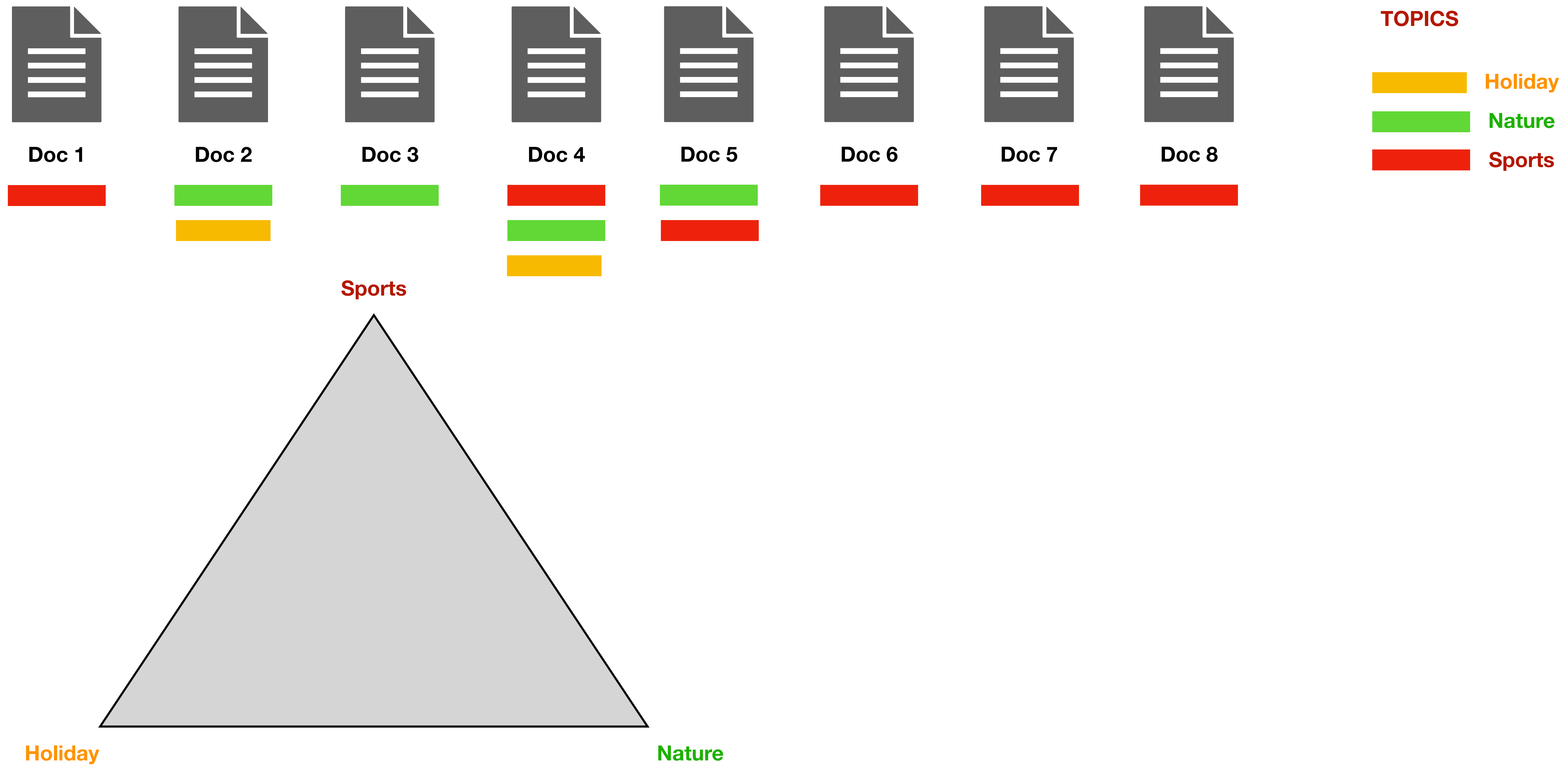
### Dirichlet distribution (dimension 3) :



# 3. Latent Dirichlet Allocation

## LDA : Dirichlet distribution

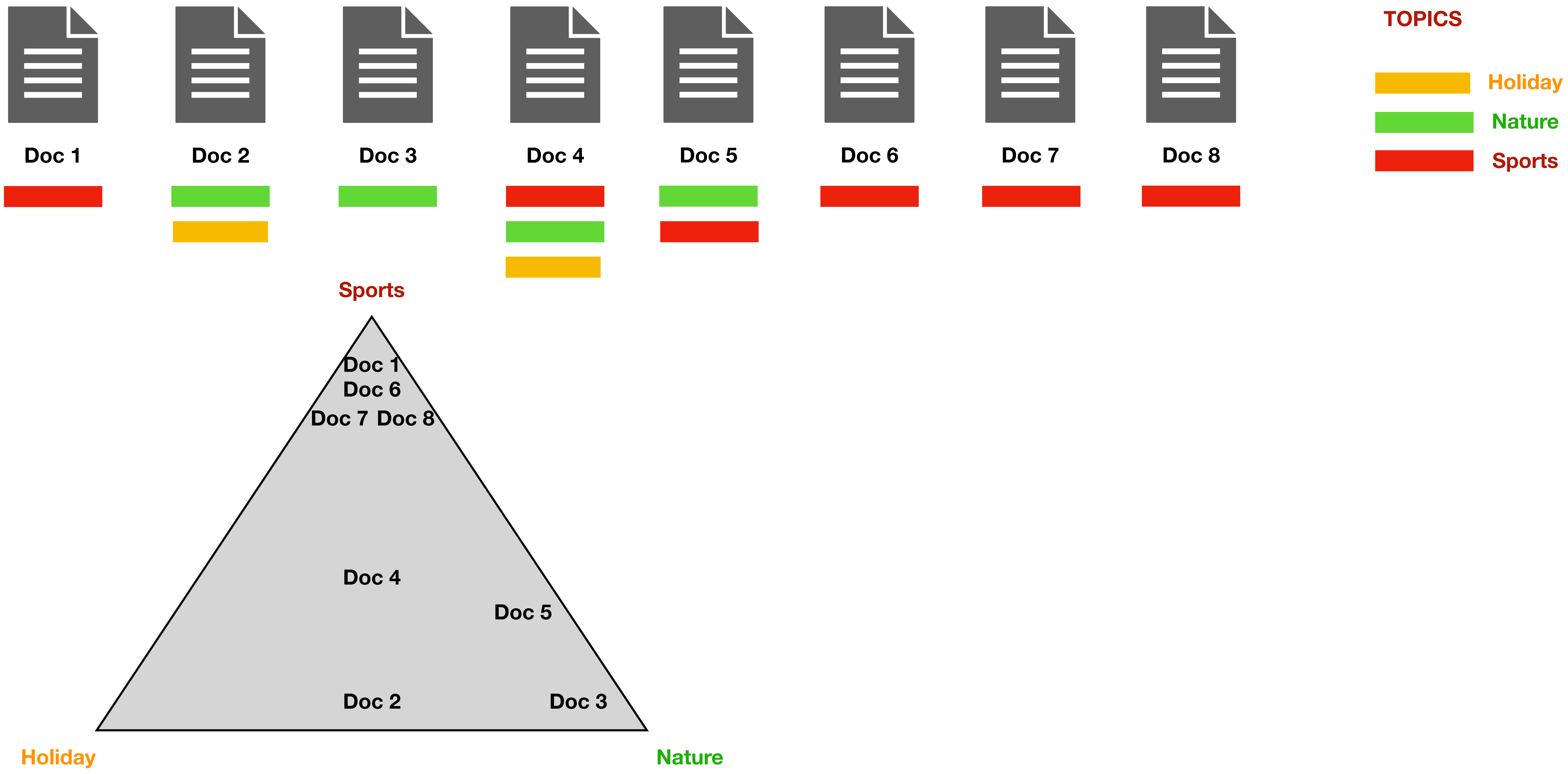
### Dirichlet distribution (dimension 3) :



# 3. Latent Dirichlet Allocation

## LDA : Dirichlet distribution

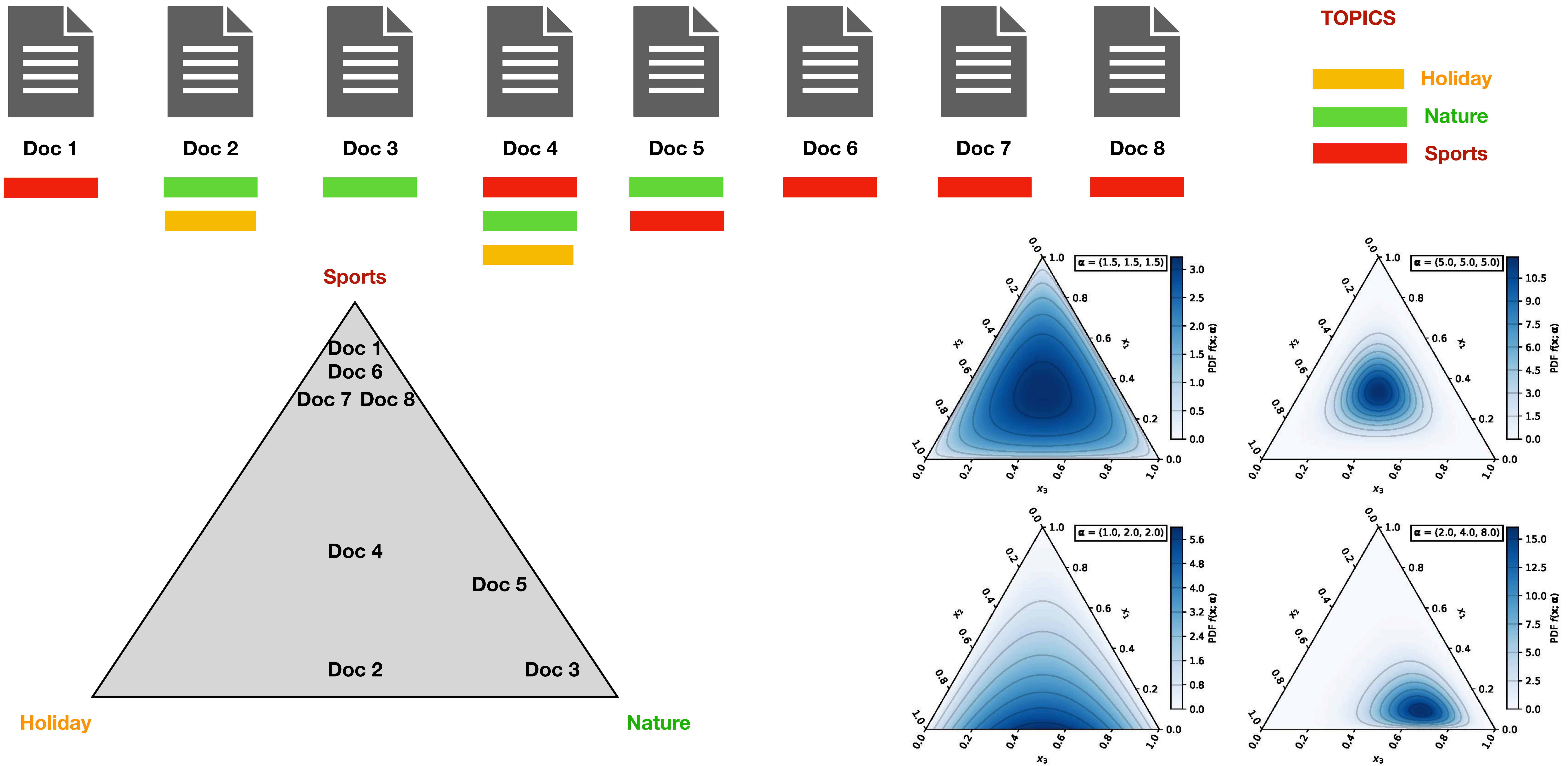
### Dirichlet distribution (dimension 3) :



# 3. Latent Dirichlet Allocation

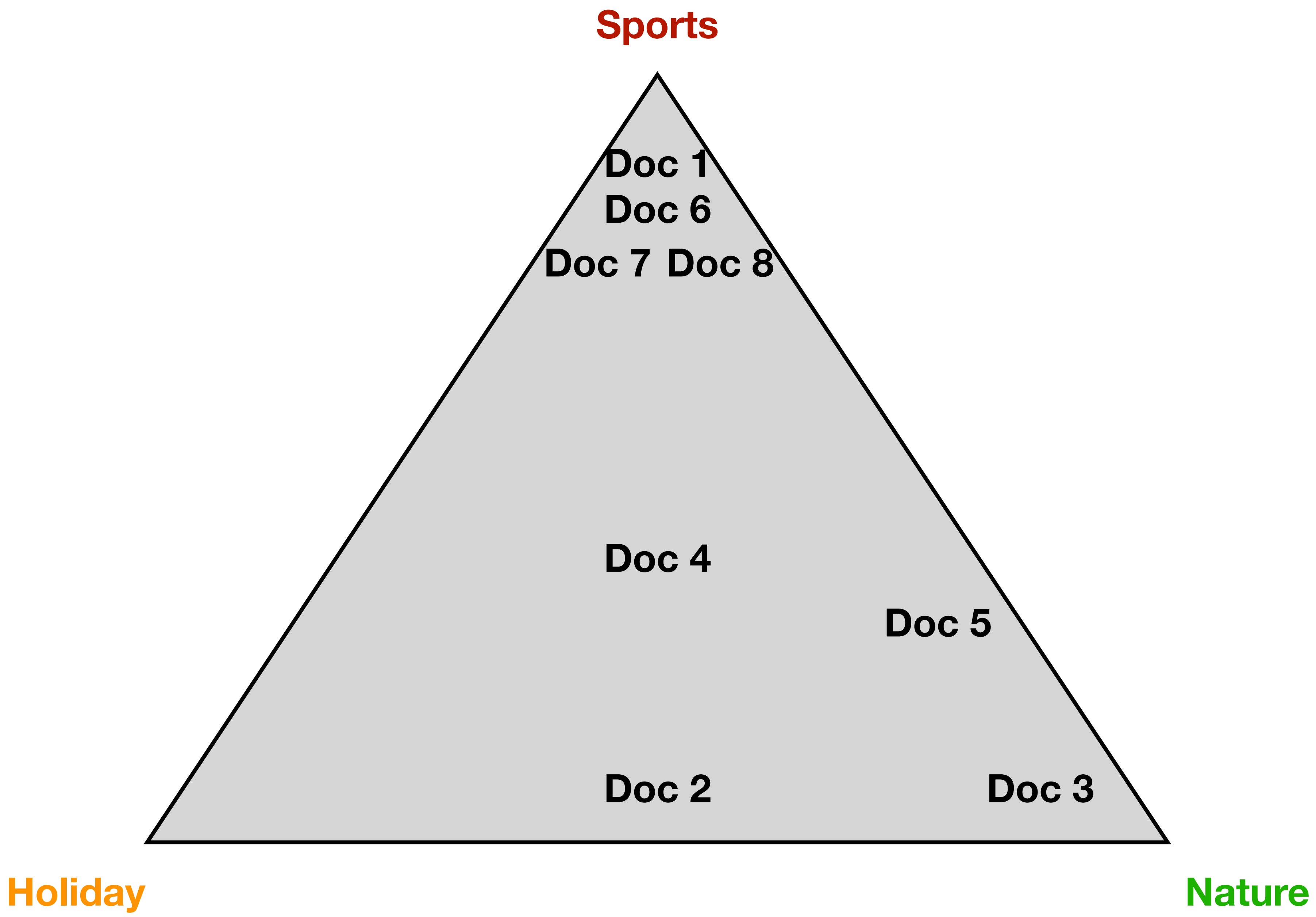
## LDA : Dirichlet distribution

### Dirichlet distribution (dimension 3) :



# 3. Latent Dirichlet Allocation

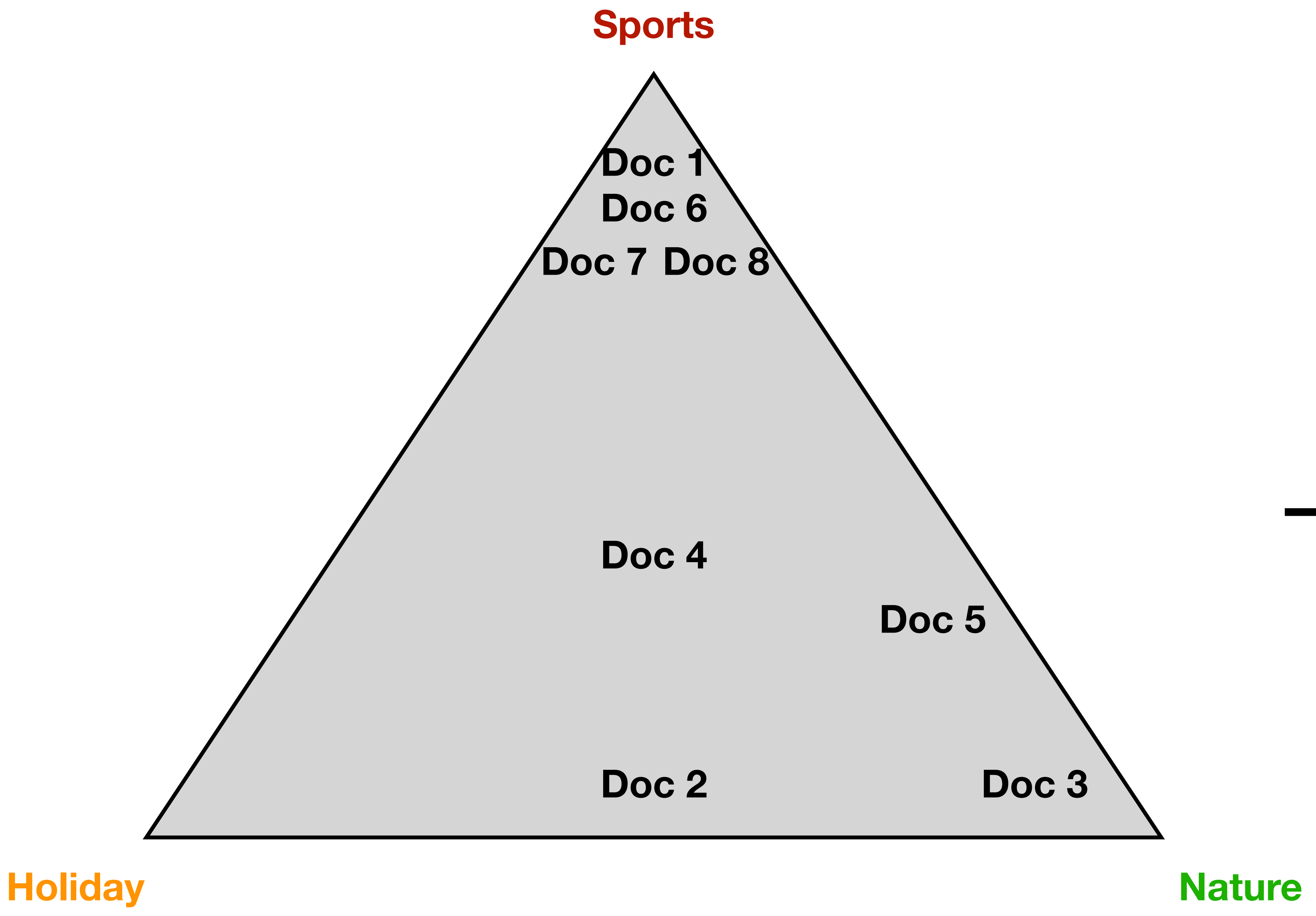
## LDA : Multinomial distribution




Dirichlet distribution
« distribution of distribution »

# 3. Latent Dirichlet Allocation

## LDA : Multinomial distribution

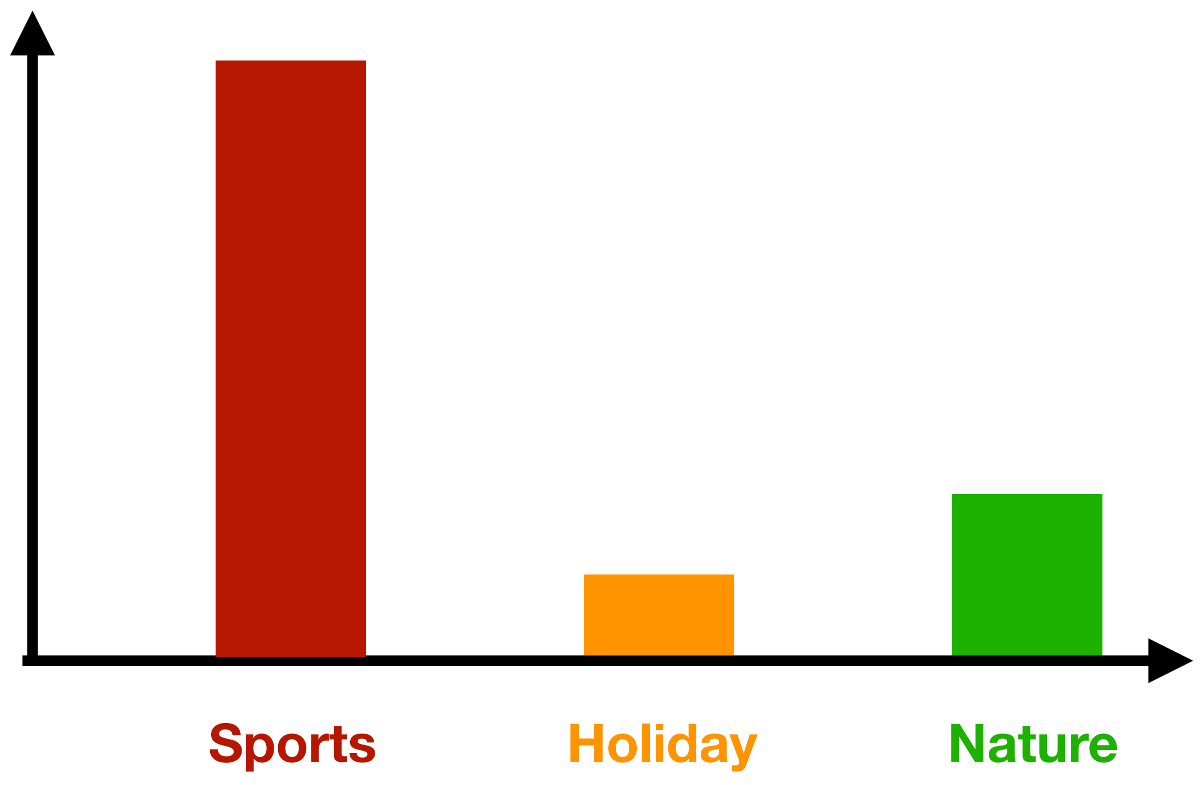



**Dirichlet distribution**
**« distribution of distribution »**

$$\theta_{\text{sports}} = P(\text{sports} \,|\, \alpha) = 0.7$$

$$\theta_{\text{holiday}} = P(\text{holiday} \,|\, \alpha) = 0.1$$

$$\theta_{\text{nature}} = P(\text{nature} \,|\, \alpha) = 0.2$$

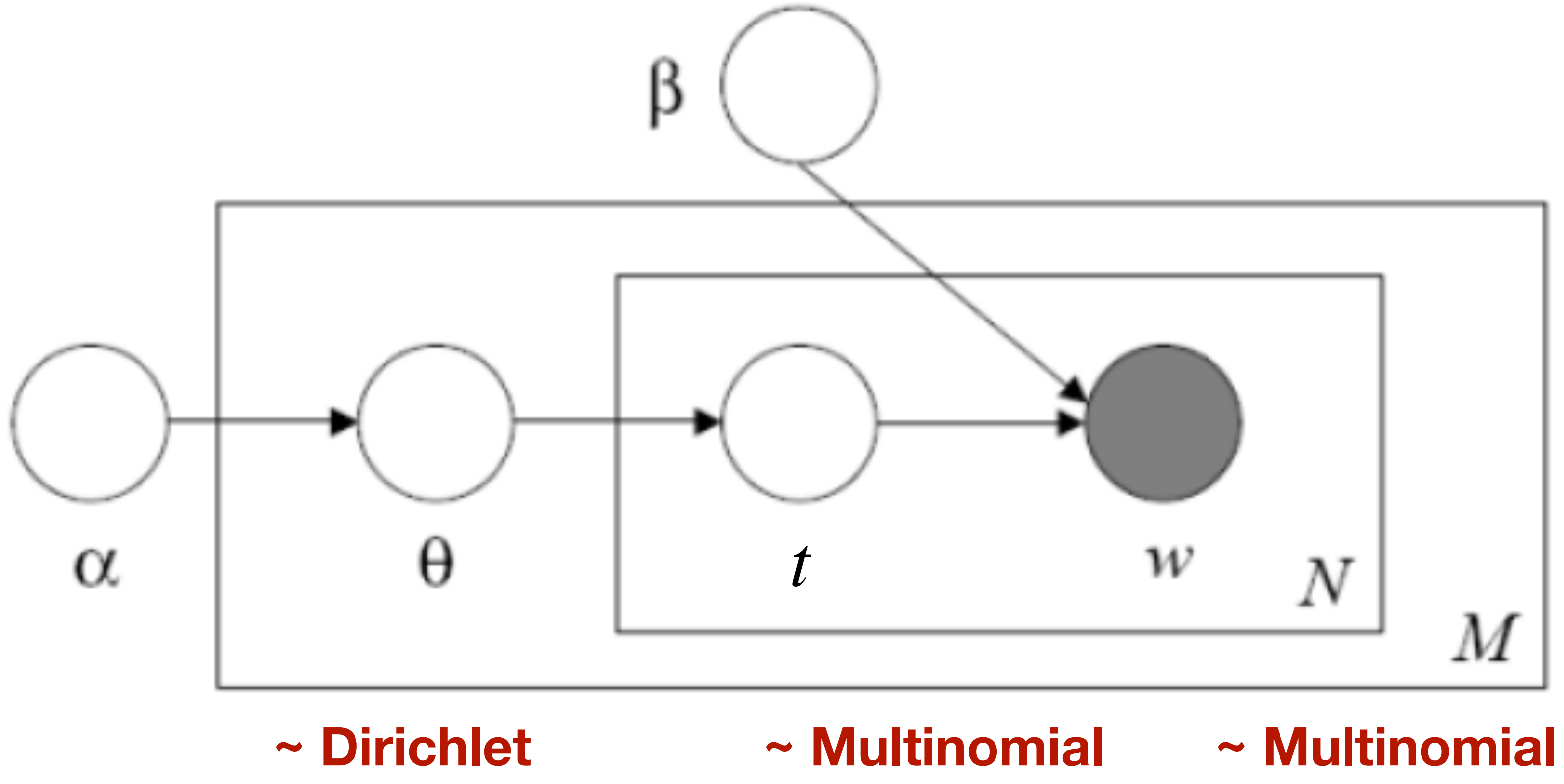



**Multinomial distribution**

# 3. Latent Dirichlet Allocation

## LDA : E-step ; calibration of theta and t

**Latent Dirichlet Allocation (LDA)** : (popular) topic modeling based on Bayesian inference with the following PGM




$$P(\theta, t, w \mid \alpha, \beta) = P(\theta \mid \alpha) \cdot P(t \mid \theta) \cdot P(w \mid t, \beta)$$

$$= \prod_{d \in [M]} \mathrm{Dir}(\theta_d \mid \alpha) \cdot \prod_{n \in [N]} \mathrm{Multi}(t_{d,n} \mid \theta_d) \cdot \mathrm{Multi}(w_{d,n} \mid t_{d,n})$$

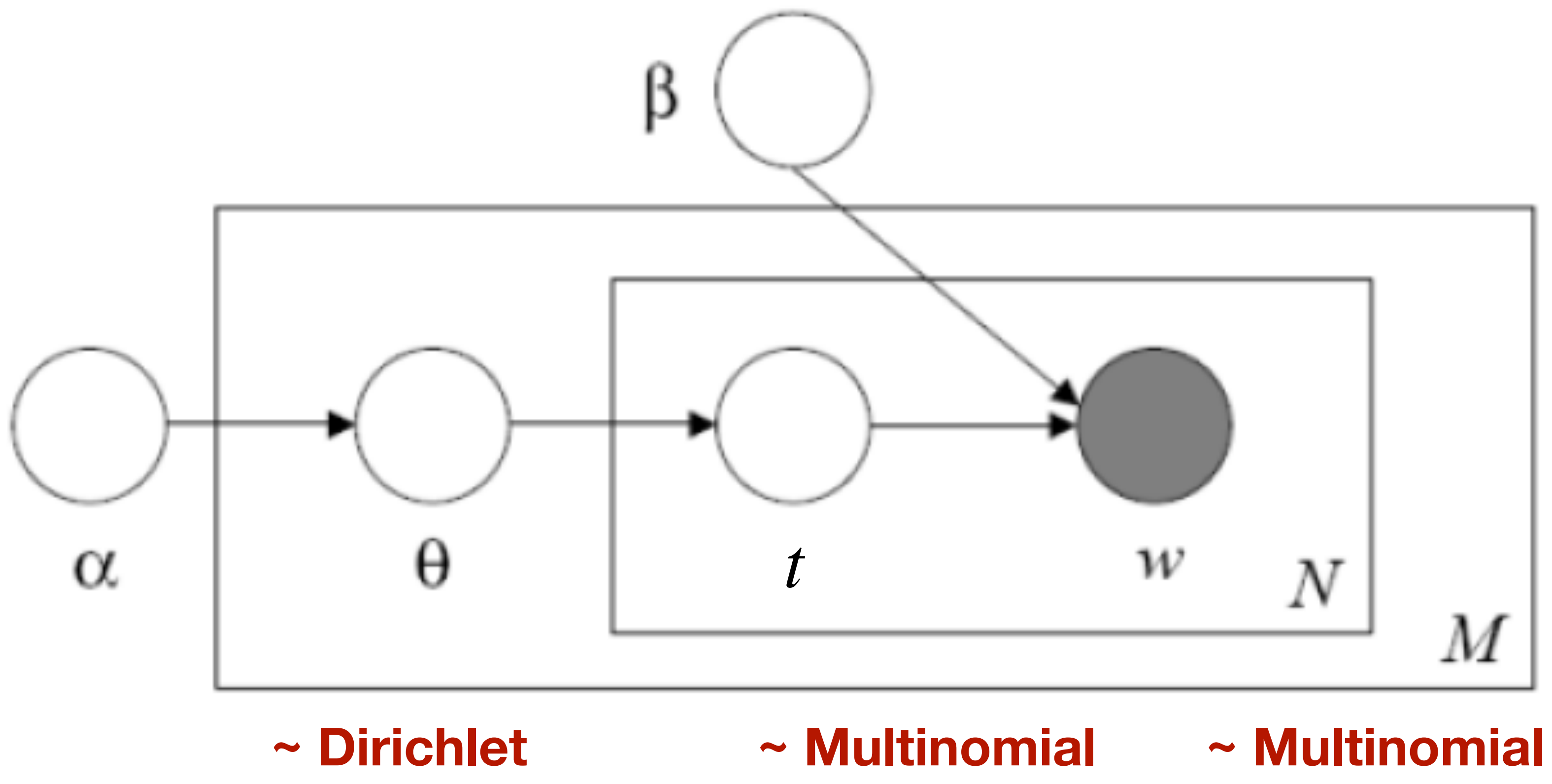
$$\propto \prod_{d \in [M]} \prod_{k \in [K]} \theta_{d,k}^{\alpha_t - 1} \cdot \prod_{n \in [N]} 1_{k = t_{d,n}} \cdot \theta_{d, t_{d,n}} \cdot \beta_{t_{d,n}, w_{d,n}}$$

**E step :**

# 3. Latent Dirichlet Allocation

## LDA : E-step ; calibration of theta and t

**Latent Dirichlet Allocation (LDA)** : (popular) topic modeling based on Bayesian inference with the following PGM




$$P(\theta, t, w \mid \alpha, \beta) = P(\theta \mid \alpha) \cdot P(t \mid \theta) \cdot P(w \mid t, \beta)$$

$$= \prod_{d \in [M]} \mathrm{Dir}(\theta_{\mathrm{d}} \mid \alpha) \cdot \prod_{n \in [N]} \mathrm{Multi}(\mathrm{t}_{\mathrm{d,n}} \mid \theta_{\mathrm{d}}) \cdot \mathrm{Multi}(\mathrm{w}_{\mathrm{d,n}} \mid \mathrm{t}_{\mathrm{d,n}})$$

$$\propto \prod_{d \in [M]} \prod_{k \in [K]} \theta_{d,k}^{\alpha_t - 1} \cdot \prod_{n \in [N]} 1_{k = t_{d,n}} \cdot \theta_{d, t_{d,n}} \cdot \beta_{t_{d,n}, w_{d,n}}$$

**E step :**

**Objective** :

$$\hat{P} = \arg \min_{Q(\theta),\, Q(t)} D_{KL}(\, Q(\theta) \times Q(t) \,||\, P(\theta, t \mid w) \,)$$

**Optimal solution** :

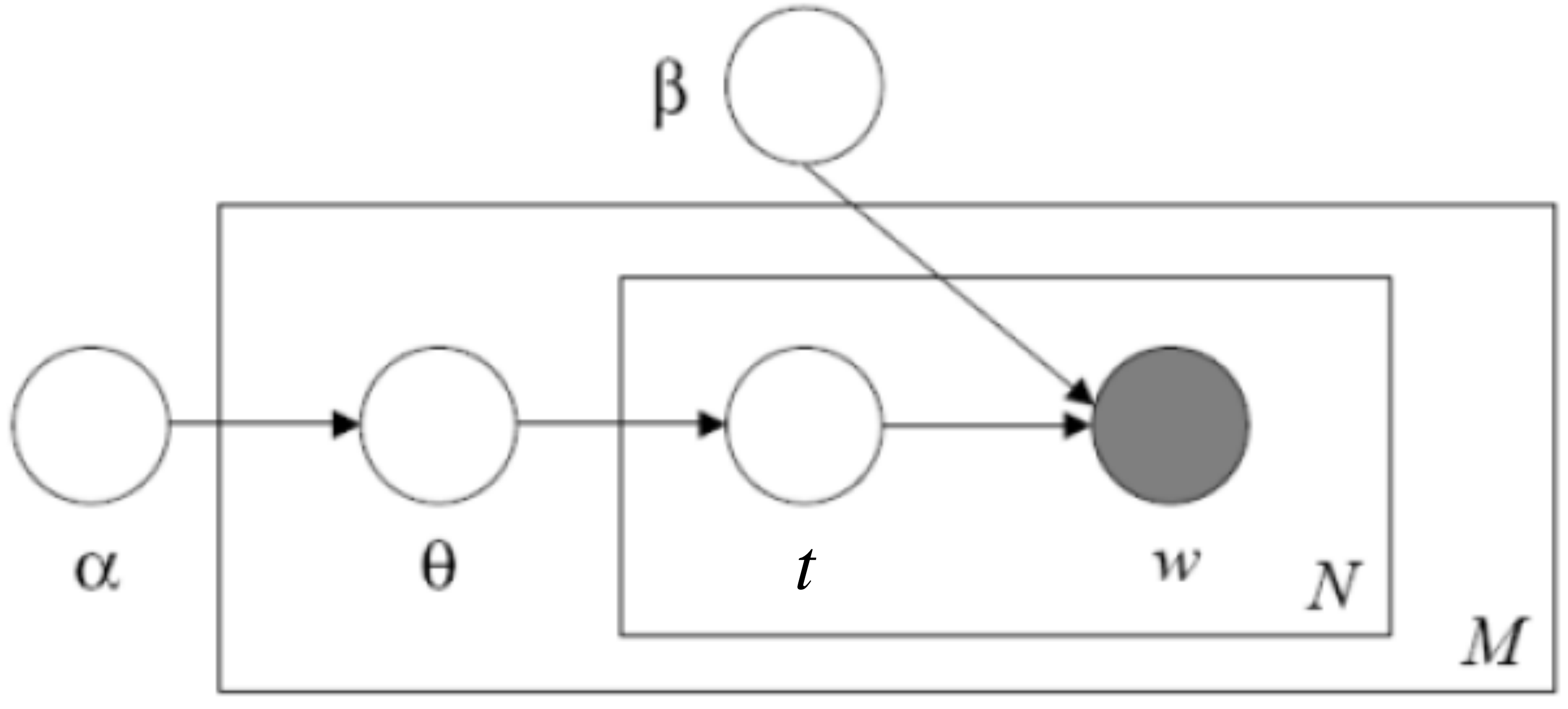
$$\log \hat{P}(\theta) = \mathbb{E}_{Q(t)} \left[ \log P(\theta, t, w) \right] + \mathrm{const}$$

$$\log \hat{P}(t) = \mathbb{E}_{Q(\theta)} \left[ \log P(\theta, t, w) \right] + \mathrm{const}$$

# 3. Latent Dirichlet Allocation

## LDA : E-step ; calibration of theta and t

**Latent Dirichlet Allocation (LDA)** : (popular) topic modeling based on Bayesian inference with the following PGM




~ Dirichlet ~ Multinomial ~ Multinomial

$$P(\theta, t, w \mid \alpha, \beta) = P(\theta \mid \alpha) \cdot P(t \mid \theta) \cdot P(w \mid t, \beta)$$

$$= \prod_{d\in[M]} \mathrm{Dir}(\theta_d \mid \alpha) \cdot \prod_{n\in[N]} \mathrm{Multi}(t_{d,n} \mid \theta_d) \cdot \mathrm{Multi}(w_{d,n} \mid t_{d,n})$$

$$\propto \prod_{d\in[M]} \prod_{k\in[K]} \theta_{d,k}^{\alpha_t - 1} \cdot \prod_{n\in[N]} 1_{k=t_{d,n}} \cdot \theta_{d,t_{d,n}} \cdot \beta_{t_{d,n}, w_{d,n}}$$

**E step :**

**Objective** :

$$\hat{P} = \arg\min_{Q(\theta),\, Q(t)} D_{KL}(\, Q(\theta) \times Q(t) \,||\, P(\theta, t \mid w)\,)$$

**Optimal solution** :

$$\log \hat{P}(\theta) = \mathbb{E}_{Q(t)} \left[ \log P(\theta, t, w) \right] + \text{const}$$

$$\log \hat{P}(t) = \mathbb{E}_{Q(\theta)} \left[ \log P(\theta, t, w) \right] + \text{const}$$

$$\log P(\theta, t, w \mid \alpha, \beta) = \sum_{d\in[M]} \left[ \sum_{k\in[K]} (\alpha_k - 1) \log \theta_{d,k} + \sum_{n\in[N]} \sum_{k\in[K]} 1_{\{k = t_{d,n}\}} \left( \log \theta_{d,t_{d,n}} + \log \beta_{t_{d,n}, w_{d,n}} \right) \right] + \text{const}$$

For $\theta$,

$$\log \hat{p}(\theta) = \mathbb{E}_{Q(t)} \left[ \log P(\theta, t, w) \right] + \text{const}$$

$$= \mathbb{E}_{Q(t)} \left[ \sum_{d\in[M]} \left( \sum_{k\in[K]} (\alpha_k - 1) \log \theta_{d,k} + \sum_{n\in[N]} \sum_{k\in[K]} 1_{k=t_{d,n}} \left( \log \theta_{d,k} \right) \right) \right] + \text{const} \quad \perp\!\!\!\perp \theta$$

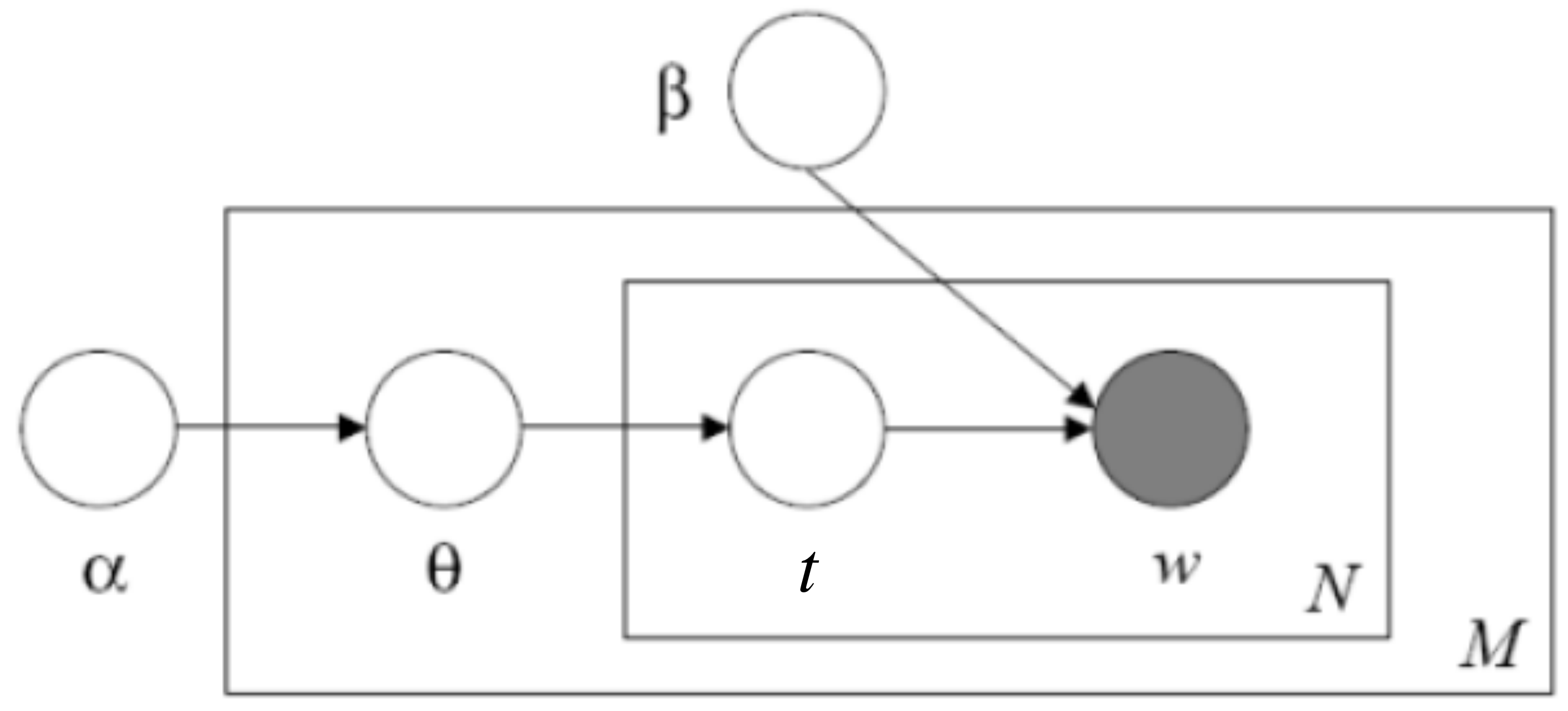
$$= \sum_{d\in[M]} \sum_{k\in[K]} \Big[ (\alpha_k - 1) + \sum_{n\in[N]} \underbrace{\mathbb{E}_{Q(t_{d,n})} \left[ 1_{t_{d,n} = k} \right]}_{\gamma_{d,n}(k)} \Big] \times \log \theta_{d,k} + \text{const}$$

$$\hat{p}(\theta) \propto \prod_{d\in[M]} \prod_{k\in[K]} \theta_{d,k}^{\alpha_k + \sum_n \gamma_{d,n}(k) - 1} \Rightarrow \hat{p}(\theta_d) \propto \mathrm{Dir}\left(\theta_d \mid \alpha + \sum_n \gamma_{d,n}(k)\right)$$

# 3. Latent Dirichlet Allocation

## LDA : E-step ; calibration of theta and t

**Latent Dirichlet Allocation (LDA)** : (popular) topic modeling based on Bayesian inference with the following PGM




~ Dirichlet ~ Multinomial ~ Multinomial

$$P(\theta, t, w \mid \alpha, \beta) = P(\theta \mid \alpha) \cdot P(t \mid \theta) \cdot P(w \mid t, \beta)$$

$$= \prod_{d \in [M]} \mathrm{Dir}(\theta_d \mid \alpha) \cdot \prod_{n \in [N]} \mathrm{Multi}(t_{d,n} \mid \theta_d) \cdot \mathrm{Multi}(w_{d,n} \mid t_{d,n})$$

$$\propto \prod_{d \in [M]} \prod_{k \in [K]} \theta_{d,k}^{\alpha_t - 1} \cdot \prod_{n \in [N]} 1_{k = t_{d,n}} \cdot \theta_{d, t_{d,n}} \cdot \beta_{t_{d,n}, w_{d,n}}$$

**E step :**

**Objective** :

$$\hat{P} = \arg \min_{Q(\theta),\, Q(t)} D_{KL}(\, Q(\theta) \times Q(t) \;||\; P(\theta, t \mid w)\,)$$

**Optimal solution** :

$$\log \hat{P}(\theta) = \mathbb{E}_{Q(t)} \left[ \log P(\theta, t, w) \right] + \text{const}$$

$$\log \hat{P}(t) = \mathbb{E}_{Q(\theta)} \left[ \log P(\theta, t, w) \right] + \text{const}$$

$$\log P(\theta, t, w \mid \alpha, \beta) = \sum_{d \in [M]} \left[ \sum_{k \in [K]} (\alpha_k - 1) \log \theta_{d,k} + \sum_{n \in [N]} \sum_{k \in [K]} 1_{\{k = t_{d,n}\}} \left( \log \theta_{d, t_{d,n}} + \log \beta_{t_{d,n}, w_{d,n}} \right) \right] + \text{const}$$

for t,

$$\log \hat{p}(t) = E_{Q(\theta)} \left[ \log P(\theta, t, w) \right] + \text{const}$$

$$= E_{Q(\theta)} \left[ \sum_d \sum_n \sum_k 1_{\{t_{d,n} = k\}} \left( \log \theta_{d,k} + \log \beta_{k, w_{d,n}} \right) \right] + \text{const} \quad (\perp t)$$

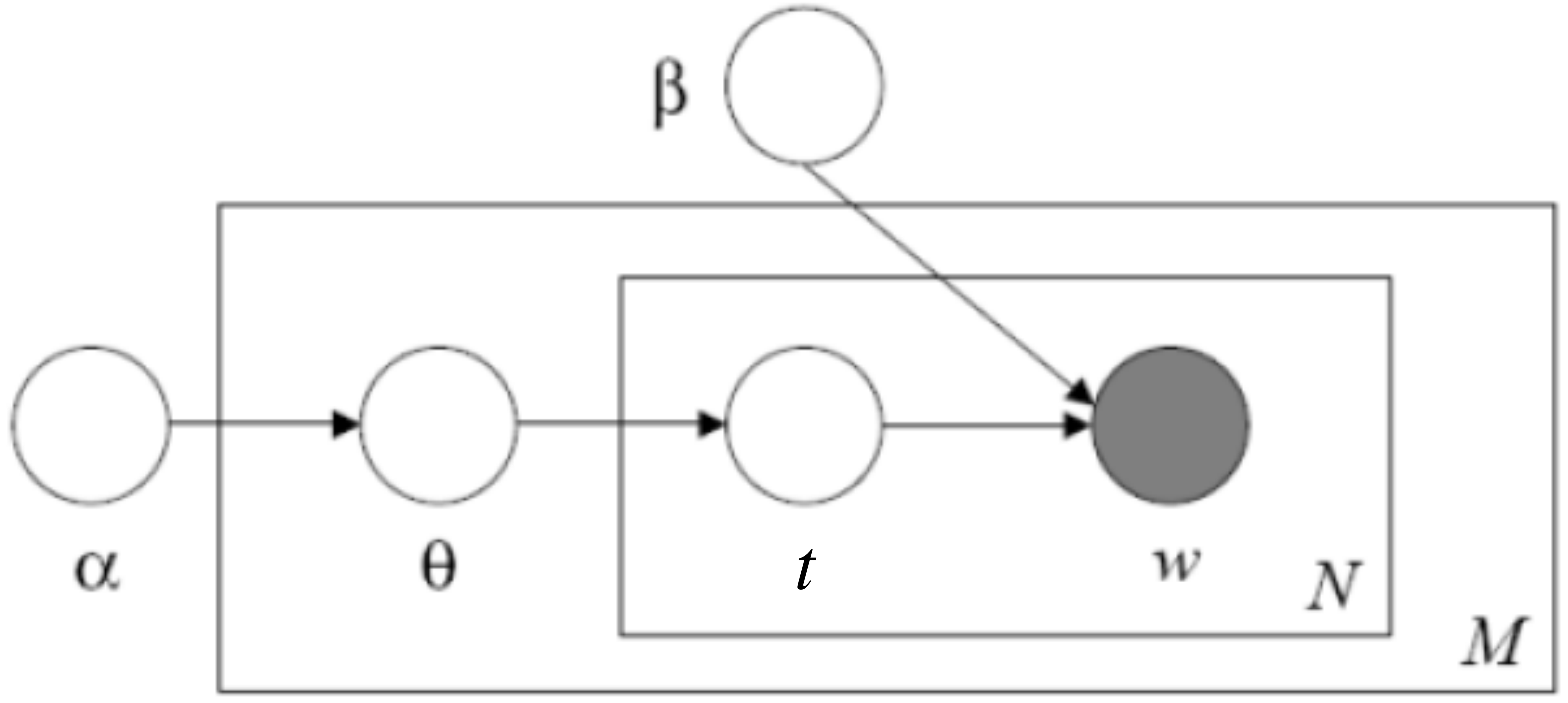
$$= \sum_d \sum_n \sum_k 1_{\{t_{d,n} = k\}} \left( E_{Q(\theta)} \left[ \log \theta_{d,k} \right] + \log \beta_{k, w_{d,n}} \right) + \text{const}$$

$$\hat{p}(t) = \prod_d \prod_n \hat{p}(t_{d,n}) \Rightarrow \hat{p}(t_{d,n} = k) = \frac{\beta_{k, w_{d,n}} e^{E_{Q(\theta)}[\log \theta_{d,k}]}}{\sum_{k'} \beta_{k', w_{d,n}} e^{E_{Q(\theta)}[\log \theta_{d,k'}]}}$$

# 3. Latent Dirichlet Allocation

## LDA : M-step

**Latent Dirichlet Allocation (LDA)** : (popular) topic modeling based on Bayesian inference with the following PGM




~ Dirichlet ~ Multinomial ~ Multinomial

$$P(\theta, t, w \mid \alpha, \beta) = P(\theta \mid \alpha) \cdot P(t \mid \theta) \cdot P(w \mid t, \beta)$$

$$= \prod_{d \in [M]} \mathrm{Dir}(\theta_d \mid \alpha) \cdot \prod_{n \in [N]} \mathrm{Multi}(t_{d,n} \mid \theta_d) \cdot \mathrm{Multi}(w_{d,n} \mid t_{d,n})$$

**M step :**

**Objective** :

$E_{Q(\theta), Q(t)} \log P(\theta, t, w)$ to maximize w.r.t. $\beta$

$$= E_{Q(\theta), Q(t)} \left[ \sum_d \sum_n \sum_k \mathbb{1}_{\{t_{d,n} = k\}} \left( \log \beta_{k, w_{d,n}} \right) \right] + const \quad (\perp\!\!\!\perp \theta, t)$$

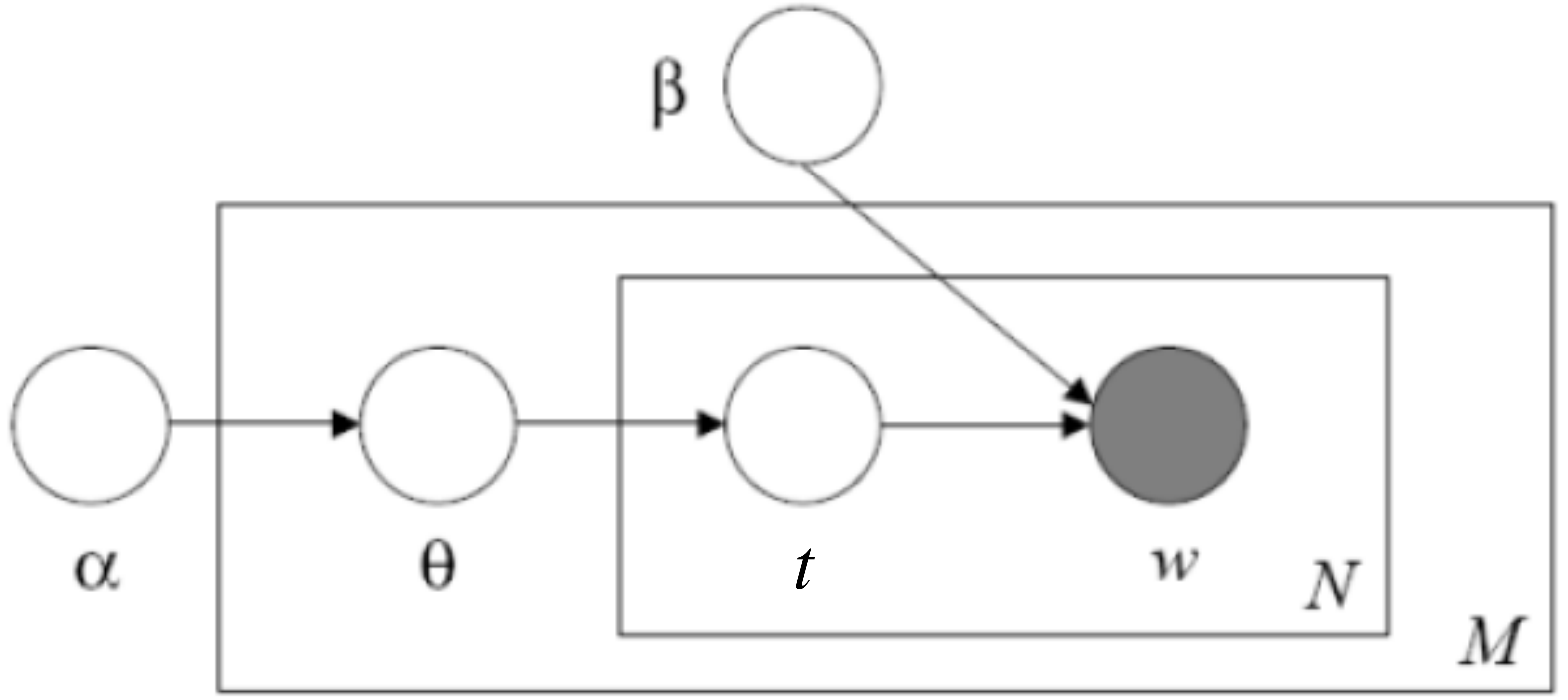
with the following constraint : $\begin{cases} \beta_{k,w} \geq 0 \\ \sum_w \beta_{k,w} = 1 \text{ for all } k \in [K] \end{cases}$

Let's compute the Lagrange

# 3. Latent Dirichlet Allocation

## LDA : M-step

**Latent Dirichlet Allocation (LDA)** : (popular) topic modeling based on Bayesian inference with the following PGM




$$P(\theta, t, w \mid \alpha, \beta) = P(\theta \mid \alpha) \cdot P(t \mid \theta) \cdot P(w \mid t, \beta)$$

$$= \prod_{d \in [M]} \mathrm{Dir}(\theta_d \mid \alpha) \cdot \prod_{n \in [N]} \mathrm{Multi}(t_{d,n} \mid \theta_d) \cdot \mathrm{Multi}(w_{d,n} \mid t_{d,n})$$

**M step :**

**Objective** :

$E_{Q(\theta), Q(t)} \log P(\theta, t, w)$ to maximize w.r.t. $\beta$

$= E_{Q(\theta), Q(t)} \left[ \sum_d \sum_n \sum_k 1_{\{t_{d,n} = k\}} (\log \beta_{k, w_{d,n}}) \right] + const$ ($\perp\!\!\!\perp \theta, t$)

with the following constraint: $\begin{cases} \beta_{k,w} \geq 0 \\ \sum_w \beta_{k,w} = 1 \text{ for all } k \in [K] \end{cases}$

Let's compute the Lagrange

Reminder: in order to max $f(x)$ with $g(x) = 0$ constraint:

denote the Lagrangian function : $L(x, \lambda) = f(x) - \lambda g(x)$

and find the stationary point.

$L(x, \lambda) = \sum_d \sum_n \sum_k \gamma_{d,n}(k) (\log \beta_{k, w_{d,n}}) - \sum_k \lambda_k (\sum_w \beta_{k,w} - 1)$

$\frac{\partial L}{\partial \beta_{k,w}}(x, \lambda) = 0 \Leftrightarrow$ (left as exercise)

$$\beta_{k,w} = \frac{\sum_{d,n,k} \gamma_{d,n}(k) 1_{\{w_{d,n} = w\}}}{\sum_{w', d, n, k} \gamma_{d,n}(k) 1_{\{w_{d,n} = w'\}}}$$

# 4 Application and examples : notebook

# Application and examples

website : https://curiousml.github.io/

- Master of Science in Artificial Intelligence Systems : **Bayesian Machine Learning** by François HU
  - **Lecture 1** : Bayesian statistics [Lecture]
  - **Lecture 2** : Latent Variable Models and EM-algorithm [Lecture]
  - **Lecture 3** : Variational Inference and intro to NLP [Soon available]
  - **Lecture 4** : Markov Chain Monte Carlo [Soon available]
  - **Lecture 5** : [Oral presentations]
  - **Training session / prerequisite :** Statistics with python [Notebook], [Data]
  - **Practical work 1** : Conjugate distributions [Notebook] [Correction]
  - **Practical work 2** : Probabilistic K-means and probabilistic PCA [Notebook]
  - **Practical work 3** : Topic Modeling with LDA [Notebook]
  - **Practical work 4** : MCMC samples [Soon available]

TODO

! Road map

## 1 Bayesian statistics

**Bayesian perspective :**

$$P(\theta|X) = \frac{P(X,\theta)}{P(X)} = \frac{P(X|\theta)\cdot P(\theta)}{P(X)}$$

Likelihood: $P(X|\theta)$

Prior distribution: $P(\theta)$

Posterior distribution: $P(\theta|X)$

Evidence: $P(X)$

Hard to compute !

$\theta$ parameters

$X$ observations

MAP : $\arg\max_{\theta} P(X|\theta)\cdot P(\theta)$

Conjugate distribution

**Exemple** :
Naive Bayes classifier,
Linear regression, ….

Pros :
- exact posterior

Cons :
- conjugate prior maybe inadequate

## 2 Latent variable models

**Hidden variable models :**

$$P(X|\theta) = \sum_{t\in T_{indexes}} P(X,T=t|\theta)$$

$$P(X,T|\theta) = P(X|T,\theta)P(T|\theta)$$

**Exemple** :
GMM, K-means, PCA/PPCA

Pros :
- fewer parameters / simpler models
- hidden variable sometimes meaningful
- clustering / dimensionality reduction

Cons :
- harder to work with
- requires math
- only local maximum or saddle point
- EM : the posterior of T could be intractable

## 3 Variational Inference

**Deterministic approximation of posterior :**

$$p(Z|X) = \frac{P(X|Z)\cdot P(Z)}{P(X)}$$

Mean Field Approximation !

**Exemple** :
Topic modelling, LDA trained by VI

Pros :
- Useful when the posterior is intractable
- Suited to large dataset

Cons :
- can never generate exact result

## 4 Causal Inference

## 5 Extensions