Text Mining
Statistical Modeling of Textual Data
Lecture 2

Måns Magnusson, Mattias Villani

Division of Statistics and Machine Learning
Dept. of Computer and Information Science
Linköping University

Overview

Text clustering

Hiearchical document clustering

Flat document clustering

Latent variable inference

Evaluation methods

Co-training / Semi-supervision

Section 1

Text clustering

TEXT CLUSTERING

- Unsupervised learning
  - Seldom a lot of training data.
  - More and more data
  - Probabilistic unsupervised learning, modeling our corpus

- Similary/distance based vs. generative models (toolbox vs. model)
- What do we want to cluster?
  - Words (co-occuring)
  - Text segments/documents
- Hiearchical clustering vs. flat clustering

Section 2

HIEARCHICAL DOCUMENT CLUSTERING

HIEARCHICAL DOCUMENT CLUSTERING

- Top-down
  - Start with all documents in one cluster
- Bottom-up
  - Start with each document in its own cluster
- Common approaches
  - Distance based (define different distances such as cosine, KL)
  - See *Mining text data* Ch. 4.

Section 3

Flat document clustering

FLAT DOCUMENT CLUSTERING

- **Problem:** Partion documents into K different clusters
- **Why?** Browse, identify similar documents, partition a corpus, understand.

- **Basic idea:** K-means klustering (**Wikipedia**)
- Basic algorithm:
  - **Assignment:**
    Assign all observations to clusters c_i based on centroid θ_k
  - **Update:**
    Update centroids θ_k based on cluster assignments c_i

PROBABILISTIC K-MEANS CLUSTERING*

- We need a **generative model**
- This is called a **mixture model**

$$p(y|\Theta) = \sum_{k}^{K} \pi_k p(y|C = k, \theta_k)$$

where $p(y|C = k, \theta_k)$ is a probability distribution for cluster k.

- In the context of text? Same idea - but we need a generative model for text...

NAIVE BAYES RECAP

- We model both $\mathbf{x}$ and $\mathbf{s}$ as $p(\mathbf{s}, \mathbf{x})$
- **Multivariate Bernoulli**

$$p(\mathbf{x}|s) = \prod_{j=1}^{n} p(\mathbf{x}_j|s_j)$$
$$= \prod_{j=1}^{n} \prod_{v=1}^{V} p(x_{j,v}|s_j)$$

  where $p(x_{j,v}) \sim Bernoulli(p_{v,s})$
- **Multinomial model**

$$p(\mathbf{w}|s) = \prod_{j=1}^{n} p(\mathbf{w}_j|s_j)$$

  where $p(\mathbf{w}_j|s_j) \sim MN(\theta_{s_j}, n_j)$

The document clustering situation

- **Now:** We do not know **s**, it is a *latent* variable
- **But:** Need to set K
- Choose the generative model:
  - Multinomial
  - Multivariate Bernoulli
  - von Mises Distribution
  - other...

VON MISES DISTRIBUTION

- **von Mises (Fisher) distribution**: random variable on a *circle* $x \in \{-\pi, \pi\}$ or for x^R a *hypersphere*.
  - Think of a Normal distribution on a hypersphere
  - Parameters:
    - $\boldsymbol{\mu}$ (point in sphere) with $\|\boldsymbol{\mu}\| = 1$,
    - κ (variance around the point) where $\kappa > 0$
- **Probability mass function**:

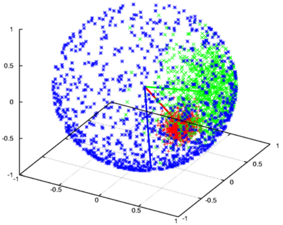
$$p(\mathbf{x}|\boldsymbol{\mu}, \kappa) = \frac{\kappa^{p/2-1}}{(2\pi)^{p/2} I_{p/2-1}(\kappa)} \exp\left(\kappa \boldsymbol{\mu}^T \mathbf{x}\right)$$

  where

$$I_{p/2-1}(\kappa)$$

  is the Bessel function of order $p/2 - 1$
- **Example:**
  - A shotgun shot is $vMF()$ with $p = 3$ and $\boldsymbol{\mu}$ is the aim and κ is the spread of the shot.

VON MISES DISTRIBUTION



FIGUR: Points sampled from three von Mises–Fisher distributions on the sphere (blue: $\kappa = 1$, green: $\kappa = 10$, red: $\kappa = 100$). The mean directions μ are shown with arrows. (Taken from Wikipedia)

VON MISES USAGE IN TEXT

- Cluster **normalized vectors**
- Using TF-IDF vectors (vector space) as data points $\mathbf{x}_d$
- Good clustering performance (?)
- `movMF` R package can fit these models (Python)?

- **Generative model**
  - Generate cluster probability as $\pi \sim \text{Dir}(\alpha)$
  - Generate K mixture components:
    - $\mu_k \sim vMF(\mu_0, \kappa_0)$
    - $\kappa_k \sim logNormal(\mu_\kappa, \sigma_\kappa)$
  - Generate D documents as
    - Generate cluster id $s_d \sim Categorical(\pi)$
    - Generate document vector $\mathbf{x}_d \sim vMF(\mu_{s_d}, \kappa_{s_d})$

GENERATIVE MODEL - MULTINOMIAL EXAMPLE

- **Generative model**
  - Generate cluster probability as $\pi \sim \text{Dir}(\alpha)$
  - Generate K mixture components:
    - $\phi_k \sim \text{Dir}(\beta)$
  - Generate D documents as
    - Generate cluster $s_d \sim \textit{Categorical}(\theta)$
    - Generate document $\mathbf{w}_d \sim \textit{Multinomial}(\phi_{s_d})$

Section 4

Latent variable inference

Expectation - Maximization

- Want to estimate $\phi_1, ..., \phi_k$ using MLE from our data

$$p(\mathbf{w}|\Theta) = \sum_{k}^{K} \pi_k p(\mathbf{w}|s = k, \phi_k)$$

where $\Theta = (\pi_1, ..., \pi_K, \phi_1, ..., \phi_K)$

- Our log likelihood is

$$L(\Theta) = \sum_{d=1}^{D} \log \left\{ \sum_{k}^{K} \pi_k p(\mathbf{w}|s = k, \phi_k) \right\}$$

- This is difficult to optimize!

EXPECTATION - MAXIMIZATION

- Instead... say we introduce $\mathbf{s}$ as a random variable. Then we get the *full observed likelihood*

$$\begin{aligned} L(\Theta, \mathbf{s}) &= \sum_{d=1}^{D} \log \left\{ \sum_{k}^{K} \mathbf{I}(s_d) \pi_k p(\mathbf{w}|s = k, \phi_k) \right\} \\ &= \sum_{d=1}^{D} \log \left\{ \sum_{k}^{K} \mathbf{I}(s_d) \pi_k p(\mathbf{w}|s = k, \phi_k) \right\} \\ &= \sum_{d=1}^{D} \sum_{k}^{K} \mathbf{I}(s_d) \log \left\{ \pi_k p(\mathbf{w}|s = k, \phi_k) \right\} \end{aligned}$$

since where $\mathbf{I}(s_d)$ is an indicator vector with 0 for all k but one that is 1.

EXPECTATION - MAXIMIZATION

- Now, $L(\Theta, \mathbf{s})$ is a random variable.
- We can now estimate Θ in two steps (but multiple iterations):
  1. **Expectation step**
     Compute $E_s(L(\Theta, \mathbf{s}))$ given $\hat{\Phi}$
  2. **Maximazion step**
     Compute $\hat{\Phi}$, a weighted estimate (we now "know" π_k)
- We converge to a local mode of the likelihood

GIBBS SAMPLING (TAKEN FROM BAYESIAN LEARNING)

- Easily implemented methods for sampling from multivariate distributions, $p(\theta_1, ..., \theta_k)$.
- **Requirements:** Easily sampled full conditional posteriors:
  - $p(\theta_1|\theta_2, \theta_3 ..., \theta_k)$
  - $p(\theta_2|\theta_1, \theta_3 ..., \theta_k)$
  - $p(\theta_3|\theta_1, \theta_2 ..., \theta_k)$
  - ...
- Started out in the early 80's in the image analysis literature.
- Gibbs sampling is a Markov Chain Monte Carlo (MCMC) algorithm.
- Generate samples from the posterior distribution $p(\theta|\mathbf{w})$.
- **Straight-forward** for latent variables.

GIBBS SAMPLING (TAKEN FROM BAYESIAN LEARNING)

1. Choose initial values $\theta_1^{(0)}, \theta_2^{(0)}, ..., \theta_k^{(0)}$.
2. Draw
   2.1 $\theta_1^{(1)}$ from $p(\theta_1|\theta_2^{(0)}, ..., \theta_k^{(0)})$
   2.2 $\theta_2^{(1)}$ from $p(\theta_2|\theta_1^{(1)}, \theta_3^{(0)}, ..., \theta_k^{(0)})$
   2.3 $\theta_3^{(1)}$ from $p(\theta_3|\theta_1^{(1)}, \theta_2^{(1)}, ..., \theta_4^{(0)}, ..., \theta_k^{(0)})$
3. Repeat Step 2 N times.

GIBBS SAMPLING (TAKEN FROM BAYESIAN LEARNING)

- The Gibbs draws $\theta^{(0)}, \theta^{(1)}, ..., \theta^{(N)}$ are dependent (autocorrelated), but arithmetic means converge to expected values

$$\frac{1}{N}\sum^{N}\theta_j \to E(\theta_j)$$

$$\frac{1}{N}\sum^{N}g(\theta_j) \to E[g(\theta_j)]$$

- $\theta^{(1)}, ..., \theta^{(N)}$ **converges in distribution** to the target $p(\theta)$
- $\theta^{(1)}, ..., \theta^{(N)}$ converge to the marginal distribution of θ_j , $p(\theta_j)$.
- Dependent draws $\to$ less efficient than iid sampling.

Mixture of Multinomial - Gibbs sampling

- We want the posterior

$$p(\mathbf{s}, \Phi, \theta | \mathbf{w}) \propto p(\mathbf{w} | \mathbf{s}, \Phi, \theta) p(\mathbf{s}, \Phi, \theta)$$

- Need to derive the conditional posteriors (see lab).
- Three step Gibbs sampling
  - Sample $\mathbf{s}_d$ given Φ and θ:

$$p(s_d = k) \propto \theta_k \prod^{V} \phi_k^{n_d^{(w)}}$$

  - Estimate $\Phi | s$:

$$\phi_k \sim Dir(\mathbf{n}_k^{(w)} + \beta)$$

    where $\mathbf{n}^{(w)}$ is the number of **tokens** in each cluster
  - Estimate $\theta | \mathbf{s}$:

$$\theta \sim Dir(\mathbf{n}^{(s)} + \alpha)$$

    where $\mathbf{n}^{(s)}$ is the number of **documents** in each cluster
  - Do this until convergance...

Mixture of Multinomial - Gibbs sampling

- Integrate out Φ and θ can be done so we only sample $\mathbf{s}$.
- Study what the cluster is about by looking at top terms in Φ
- The model has been proposed to clustered small texts such as tweets (KDD2014)

- **Text Mining project:** Compare this model and the von Mises mixture model to cluster tweets

Section 5

Evaluation methods

EVALUATING

- Evaluation is a hard problem
  - As $K \to \infty$ we can fit better and better models
  - No gold standard
- **External** measures :
  - Needs a gold standard (why would we cluster?)
  - **NOT** to be confused with classification
  - confusion matrix, classification accuracy, F1 measure, average purity
- **Internal** measures:
  - Similarity within and between clusters (can be define in various ways)
- **Probabilistic models**
  - Estimate the marginal likelihood on a **test set**

$$p(\mathbf{w}_{test}) = \int p(\mathbf{w}_{test}|\Theta)d\Theta$$

  - AIC, WAIC, BIC, DIC, Perplexity

Section 6

Co-training / Semi-supervision

CO-TRAINING / SEMI-SUPERVISION

- What if we know some **s** but not all?
- **Co-training**: Sample **s** for missing classes
- Can increase accuracy for small training sizes
- **Problem:**
  - Cluster $\neq$ classes (think sentiment in news wire - probably cluster is "content" if not content words are removed)
- **Solutions:**
  - Multiple mixtures/clusters per class

Text mining project idea: But what if?

- Remember the Part-of-Speech tagger with a Hidden Markov Model.
- What if we do not know all $\mathbf{t}$?
  - Treat unknown tags as a parameter!
  - Use a lexicon of tags to restrict the Dirichlet prior distributions (conditional Dirichlet).
- **Text mining project:**
  - Using a Gibbs sampler to combine supervised tags, types and unsupervised data.
  - Will more data improve accuracy? Type supervision, tag supervision?