University of West Bohemia
Faculty of Applied Sciences
Department of Computer Science and Engineering

Master's Thesis

Incremental News Clustering

Pilsen 2018

Martin Váňa

ZÁPADOČESKÁ UNIVERZITA V PLZNI
Fakulta aplikovaných věd
Akademický rok: **2017/2018**

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení:	**Bc. Martin VÁŇA**
Osobní číslo:	**A15N0083P**
Studijní program:	**N3902 Inženýrská informatika**
Studijní obor:	**Softwarové inženýrství**
Název tématu:	**Inkrementální shlukování zpravodajských textů**
Zadávající katedra:	**Katedra informatiky a výpočetní techniky**

Zásady pro vypracování:

1. Seznamte se s algoritmy shlukování textů, zejména s "distance-dependent Chinese restaurant process".
2. Navrhněte metodu, která bude schopná udržovat tématické shluky zpravodajských textů, které budou postupně přicházet z crawleru.
3. Metodu implementujte a vytvořte jednoduchý demonstrátor.
4. Otestujte kvalitu shlukování a výsledky zhodnoťte.

Rozsah grafických prací: **dle potřeby**

Rozsah kvalifikační práce: **doporuč. 50 s. původního textu**

Forma zpracování diplomové práce: **tištěná**

Seznam odborné literatury:

dodá vedoucí diplomové práce

Vedoucí diplomové práce: **Doc. Ing. Josef Steinberger, Ph.D.**
Katedra informatiky a výpočetní techniky

Datum zadání diplomové práce: **1. září 2017**

Termín odevzdání diplomové práce: **17. května 2018**

Radová

Doc. Dr. Ing. Vlasta Radová
děkanka

Západočeská univerzita v Plzni
L.S.
Fakulta aplikovaných věd

Doc. Ing. Přemysl Brada, MSc. Ph.D.
vedoucí katedry

V Plzni dne 14. září 2017

Declaration

I hereby declare that this master's thesis is completely my own work and that I used only the cited sources.

Pilsen, 11th May 2018

Martin Váňa

Acknowledgement

I would like to thank my thesis supervisor Doc. Ing. Josef Steinberger, PhD. for his beneficial advice and comments on this thesis.

Abstract

The goal was to research model-based clustering methods, notably the Distance Dependent Chinese Restaurant Process (ddCRP), and propose an incremental clustering system which would be capable of maintaining the growing number of topic clusters of news articles coming online from a crawler. LDA, LSA, and doc2vec methods were used to represent a document as a fixed-length numeric vector. Cluster assignments given by a proof-of-concept implementation of such a system were evaluated using various metrics, notably purity, F-measure and V-measure. A modification of V-measure – NV-measure – was introduced in order to penalize an excessive or insufficient number of clusters. The best results were achieved with doc2vec and ddCRP.

Abstrakt

Cílem práce bylo prozkoumat možnosti shlukovacích metod založených na statistických modelech, zejména metodu založenou na Distance Dependent Chinese Restaurant Process (ddCRP), a navrhnout shlukovací systém, který bude schopný udržovat tematické shluky zpravodajských textů, které budou postupně přicházet z crawleru. Metody LDA, LSA a doc2vec byly použity k reprezentaci dokumentu jako číselný vektor fixní délky. Výsledné shluky odhalené proof-of-concept implementací takového systému byly vyhodnoceny zejména pomocí purity, F-measure a V-measure. Dále byla představena evaluační metrika NV-measure vycházející z V-measure, které penalizuje nadměrné či naopak nedostatečné množství shluků. Nejlepších výsledků bylo dosaženo pomocí doc2vec a ddCRP.

Contents

1 Introduction

Every day millions of news articles and blog posts are written. It is not even remotely possible to read, understand and manually categorize such amounts of data. Because the internet is a rapidly growing medium, an incremental approach is needed. The number of topic clusters is continuously growing because it would be very difficult to argue that the number of topics eventually runs up against a finite bound and remains fixed.

The goal is to research model-based clustering methods, notably the Distance Dependent Chinese Restaurant Process (ddCRP), and propose an incremental clustering system which would be capable of maintaining growing number of topic clusters of news articles coming online from a crawler.

Initially, the necessary mathematics and statistics are defined. Latent Semantic Analysis (LSA), Latent Dirichlet Allocation (LDA), and doc2vec methods are used to represent a document as a fixed-length numeric vector. Both finite and infinite model-based clustering methods are covered. The focus is on the most common variant of Latent Variable Model (LVM) – the Gaussian mixture model (GMM). Various criteria evaluating the quality of a clustering are described.

The practical part focuses on the architecture and implementation of the incremental clustering system. Finally, the experiments with clustering system are performed on real data.

2 Functions useful in Multivariate Statistics

Later in the text few non-trivial functions which deserve detailed description appear.

2.1 Gamma Function

The gamma function [14, p. 42] is a generalization of the factorial function to real and complex numbers, and is defined via an improper integral:

$$\Gamma(x) = \int_0^\infty u^{x-1} e^{-u} du \tag{2.1}$$

Figure 2.1 shows that for $x \in \mathbb{Z}^+$, the gamma function is equivalent to the factorial function with the argument decremented by 1:

$$\Gamma(x) = (x-1)! \tag{2.2}$$

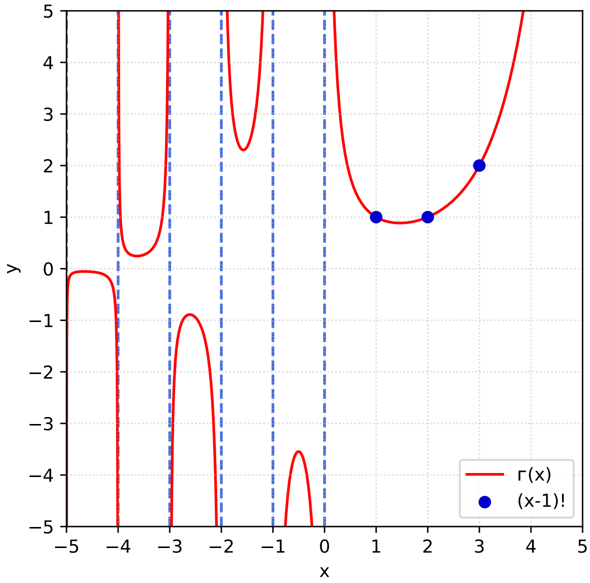

Figure 2.1: Gamma function

2.2 Multivariate Gamma Function

The multivariate gamma function [14, p. 126] is a generalization of the gamma function to higher dimensions. It is widely used in multivariate statistics and is defined as:

$$\Gamma_D(x) = \pi^{\frac{D(D-1)}{4}} \prod_{i=1}^{D} \Gamma\left(x + \frac{1-i}{2}\right) \tag{2.3}$$

For $D = 1$ it holds that $\Gamma_1(x) = \Gamma(x)$.

3 Selected Probability Distributions

The probability distributions described in this chapter are essential for understanding the statistical models and clustering methods described later.

3.1 Multinomial Distribution

The multinomial distribution [14, p. 35] is a discrete probability distribution which generalizes the binomial distribution for multiple outcomes of n independent experiments.

Let $\mathbf{x} = (x_1, \ldots, x_K)$ be a random vector, where x_i is a number of times we observe an outcome i. Then the probability mass function of the multinomial distribution is:

$$\mathrm{Mu}\left(\mathbf{x} \,|n, \boldsymbol{\theta}\right) = \binom{n}{x_1 \ldots x_K} \prod_{i=1}^{K} \theta_i^{x_i} \tag{3.1}$$

where θ_i is the probability that we observe an outcome i, and

$$\binom{n}{x_1 \ldots x_K} = \frac{n!}{x_1! x_2! \ldots x_K!} = \frac{\Gamma(n+1)}{\prod_{i=1}^{K} \Gamma(x_i + 1)} \tag{3.2}$$

is the multinomial coefficient (the number of ways to divide a set of size $n = \sum_{i=1}^{K} x_i$ into subsets with sizes $x_1, \ldots, x_K$).

3.2 Categorical Distribution

The categorical distribution [14, pp. 35–36] (a.k.a. discrete or multinoulli distribution) is a special case of the multinomial distribution, where $n = 1$. In this case, we can think of x as being a scalar categorical random variable with K states (values), and $\mathbf{x}$ is its dummy (one-hot) encoding, that is, $\mathbf{x} = (\mathbb{I}(x = 1), \ldots, \mathbb{I}(x = K))$.

$$\mathrm{Cat}\left(\mathbf{x} \,|\, \boldsymbol{\theta}\right) = \mathrm{Mu}\left(\mathbf{x} \,|1, \boldsymbol{\theta}\right) = \prod_{i=1}^{K} \theta_i^{\mathbb{I}(x_i = 1)} \tag{3.3}$$

In other words, if $x \sim \mathrm{Cat}\left(\theta\right)$, then $\mathrm{p}(x = j |\, \boldsymbol{\theta}) = \theta_j$.

3.3 Dirichlet Distribution

The Dirichlet distribution [14, p. 47] is a continuous multivariate probability distribution widely used as a prior distribution in Bayesian statistics. It is a multivariate generalization of the beta distribution, which has support over the probability simplex[1]:

$$S_K = \{\mathbf{x} : 0 \leq x_k \leq 1, \sum_{k=1}^{K} x_k = 1\} \tag{3.4}$$

The figure 3.1 shows the probability density function which is defined as follows:

$$\mathrm{Dir}\left(\mathbf{x} \,|\boldsymbol{\alpha}\right) = \frac{1}{\mathrm{B}\left(\boldsymbol{\alpha}\right)} \prod_{k=1}^{K} x_k^{\alpha_k - 1}\, \mathbb{I}(\mathbf{x} \in S_K) \tag{3.5}$$

where $\mathrm{B}\left(\boldsymbol{\alpha}\right)$ is a natural generalization of the beta function to K variables:

$$\mathrm{B}\left(\boldsymbol{\alpha}\right) = \frac{\prod_{k=1}^{K} \Gamma(\alpha_k)}{\Gamma(\alpha_0)} \tag{3.6}$$

where $\alpha_0 = \sum_{k=1}^{K} \alpha_k$. A special case of the Dirichlet distribution is a symmetric Dirichlet distribution, where $\alpha_k = \frac{\alpha}{K}$.

3.4 Multivariate Normal Distribution

The Multivariate normal distribution (MVN) [14, p. 46] or multivariate Gaussian is the most widely used multivariate continuous probability distribution which generalize normal distribution to higher dimensions.

The probability density function of the MVN in D dimensions is defined as follows:

$$\mathcal{N}\left(\mathbf{x} |\, \boldsymbol{\mu}, \boldsymbol{\Sigma}\right) = \frac{1}{(2\pi)^{\frac{D}{2}} |\,\boldsymbol{\Sigma}\,|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^T\, \boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu})\right) \tag{3.7}$$

where $\boldsymbol{\mu} = \mathbb{E}\left[\mathbf{x}\right] \in \mathbb{R}^D$ is the mean vector, and $\boldsymbol{\Sigma} = \mathrm{cov}\left[\mathbf{x}\right]$ is the $D \times D$ covariance matrix.

[1] Simplex is a generalized notion of a triangle to arbitrary dimension.

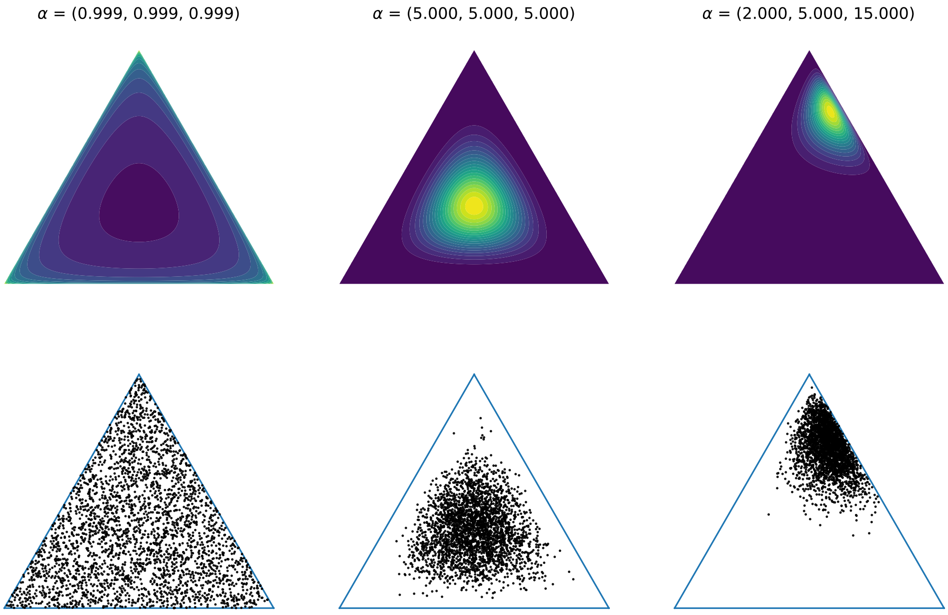

Figure 3.1: Dirichlet distribution for various parameters $\boldsymbol{\alpha}$ when $K = 3$

3.4.1 MLE for an MVN

We can estimate the parameters of an MVN using Maximum likelihood estimation (MLE) [14, p. 99] for N i.i.d. (independent and identically distributed) samples $\mathbf{x}_i \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. The empirical mean is defined as:

$$\begin{aligned}\hat{\boldsymbol{\mu}}_{mle} &= \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_i \\ &= \frac{1}{N}\mathbf{X}^T\,\mathbf{1}_N \\ &= \bar{\mathbf{x}}\end{aligned} \tag{3.8}$$

The scatter matrix, used to make estimates of the empirical covariance matrix, is the $D \times D$ positive semi-definite matrix ($\mathbf{S}_{\bar{\mathbf{x}}} \succ 0$):

$$\begin{aligned}\mathbf{S}_{\bar{\mathbf{x}}} &= \sum_{i=1}^{N}(\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T \\ &= \sum_{i=1}^{N}\mathbf{x}_i\,\mathbf{x}_i^T - N\,\bar{\mathbf{x}}\,\bar{\mathbf{x}}^T \\ &= \mathbf{X}^T\,\mathbf{X} - N\,\bar{\mathbf{x}}\,\bar{\mathbf{x}}^T\end{aligned} \tag{3.9}$$

The MLE estimate of the covariance is:

$$\begin{aligned}
\hat{\mathbf{\Sigma}}_{mle} &= \frac{1}{N}\,\mathbf{S}_{\bar{\mathbf{x}}} \\
&= \frac{1}{N}\sum_{i=1}^{N}(\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T \\
&= \frac{1}{N}\left(\sum_{i=1}^{N}\mathbf{x}_i\,\mathbf{x}_i^T\right) - \bar{\mathbf{x}}\,\bar{\mathbf{x}}^T \\
&= \frac{1}{N}\,\mathbf{X}^T\,\mathbf{X} - \bar{\mathbf{x}}\,\bar{\mathbf{x}}^T \qquad (3.10)
\end{aligned}$$

After application of Bessel's correction [8], an unbiased estimation of the empirical covariance is:

$$\begin{aligned}
\hat{\mathbf{\Sigma}} &= \frac{1}{N-1}\sum_{i=1}^{N}(\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T \\
&= \frac{N}{N-1}\hat{\mathbf{\Sigma}}_{mle} \\
&= \frac{\mathbf{X}^T\,\mathbf{X} - N\,\bar{\mathbf{x}}\,\bar{\mathbf{x}}^T}{N-1} \qquad (3.11)
\end{aligned}$$

3.4.2 Inferring the Parameters of an MVN

Assume we observe data $\mathbf{X} = \{\mathbf{x}_i : \mathbf{x}_i \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{\Sigma})\}_{i=1}^{N}$ drawn from an MVN and we want to infer the unknown parameters $\boldsymbol{\theta} = (\boldsymbol{\mu}, \mathbf{\Sigma})$.

Likelihood

The likelihood [14, p. 132] says how probable is it to see the observed data $\mathbf{X}$ given parameters $\boldsymbol{\theta}$, and is given by:

$$\begin{aligned}
\mathrm{p}(\mathbf{X}\,|\,\boldsymbol{\mu}, \mathbf{\Sigma}) &= \prod_{i=1}^{N}\mathcal{N}(\mathbf{x}_i\,|\,\boldsymbol{\mu}, \mathbf{\Sigma}) \qquad (3.12) \\
&= \frac{1}{(2\pi)^{\frac{ND}{2}}}|\,\mathbf{\Sigma}\,|^{-\frac{N}{2}} \exp\left(-\frac{1}{2}\sum_{i=1}^{N}(\mathbf{x}_i - \boldsymbol{\mu})^T\,\mathbf{\Sigma}^{-1}(\mathbf{x}_i - \boldsymbol{\mu})\right) \qquad (3.13) \\
&= \frac{1}{(2\pi)^{\frac{ND}{2}}}|\,\mathbf{\Sigma}\,|^{-\frac{N}{2}} \\
&\quad \times \exp\left(-\frac{N}{2}(\boldsymbol{\mu} - \bar{\mathbf{x}})^T\,\mathbf{\Sigma}^{-1}(\boldsymbol{\mu} - \bar{\mathbf{x}}) - \frac{1}{2}\,\mathrm{tr}\left(\mathbf{\Sigma}^{-1}\,\mathbf{S}_{\bar{\mathbf{x}}}\right)\right) \qquad (3.14)
\end{aligned}$$

Prior

A prior probability is our belief (e.g. in parameters $\boldsymbol{\theta}$) before we observe data $\mathbf{X}$. A fully conjugate prior [14, p. 132] for the MVN has a form of joint distribution of the form:

$$\mathrm{p}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \mathrm{p}(\boldsymbol{\mu} \,|\, \boldsymbol{\Sigma})\, \mathrm{p}(\boldsymbol{\Sigma}) \tag{3.15}$$

Specifically, it is defined [14, p. 133] as the Normal-inverse-Wishart distribution (NIW):

$$\begin{aligned}
&\mathrm{NIW}\left(\boldsymbol{\mu}, \boldsymbol{\Sigma} \,|\, \mathbf{m}_0, \kappa_0, \nu_0, \mathbf{S}_0\right) = \\
&\qquad \mathcal{N}\left(\boldsymbol{\mu} \,|\, \mathbf{m}_0, \frac{1}{\kappa_0}\boldsymbol{\Sigma}\right) \times \mathrm{IW}\left(\boldsymbol{\Sigma} \,|\, \mathbf{S}_0, \nu_0\right) && (3.16)\\
&\quad = \frac{1}{Z_{\mathrm{NIW}}} |\,\boldsymbol{\Sigma}\,|^{-\frac{1}{2}} \exp\left(-\frac{\kappa_0}{2}(\boldsymbol{\mu} - \mathbf{m}_0)^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu} - \mathbf{m}_0)\right)\\
&\qquad \times |\,\boldsymbol{\Sigma}\,|^{-\frac{\nu_0+D+1}{2}} \exp\left(-\frac{1}{2}\operatorname{tr}\left(\boldsymbol{\Sigma}^{-1}\mathbf{S}_0\right)\right) && (3.17)\\
&\quad = \frac{1}{Z_{\mathrm{NIW}}} |\,\boldsymbol{\Sigma}\,|^{-\frac{\nu_0+D+2}{2}}\\
&\qquad \times \exp\left(-\frac{\kappa_0}{2}(\boldsymbol{\mu} - \mathbf{m}_0)^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu} - \mathbf{m}_0) - \frac{1}{2}\operatorname{tr}\left(\boldsymbol{\Sigma}^{-1}\mathbf{S}_0\right)\right) && (3.18)
\end{aligned}$$

$$Z_{\mathrm{NIW}}\left(D, \kappa_0, \nu_0, \mathbf{S}_0\right) = 2^{\frac{\nu_0 D}{2}} \Gamma_D\left(\frac{\nu_0}{2}\right)\left(\frac{2\pi}{\kappa_0}\right)^{\frac{D}{2}} |\,\mathbf{S}_0\,|^{-\frac{\nu_0}{2}} \tag{3.19}$$

The parameters of the NIW can be interpreted as follows:

- $\mathbf{m}_0$ is our prior mean for $\boldsymbol{\mu}$.
- κ_0 exhibits how strongly we believe the prior $\mathbf{m}_0$.
- $\mathbf{S}_0$ is (proportional to) our prior mean for $\boldsymbol{\Sigma}$.
- ν_0 exhibits how strongly we believe the prior $\mathbf{S}_0$.

Posterior

The posterior probability states a probability conditional on the observed evidence $\mathbf{X}$. In our case, it can be shown [14, p. 134] that the posterior probability is NIW with updated parameters:

$$
\begin{aligned}
\mathrm{p}(\boldsymbol{\mu}, \boldsymbol{\Sigma} \,|\, \mathbf{X}) &= \mathrm{NIW}\left(\boldsymbol{\mu}, \boldsymbol{\Sigma} \,|\, \mathbf{m}_N, \kappa_N, \nu_N, \mathbf{S}_N\right) && (3.20)\\
\mathbf{m}_N &= \frac{\kappa_0\, \mathbf{m}_0 + N\, \bar{\mathbf{x}}}{\kappa_N} = \frac{\kappa_0}{\kappa_0 + N}\, \mathbf{m}_0 + \frac{N}{\kappa_0 + N}\, \bar{\mathbf{x}} && (3.21)\\
\kappa_N &= \kappa_0 + N && (3.22)\\
\nu_N &= \nu_0 + N && (3.23)\\
\mathbf{S}_N &= \mathbf{S}_0 + \mathbf{S}_{\bar{\mathbf{x}}} + \frac{\kappa_0 N}{\kappa_0 + N}(\bar{\mathbf{x}} - \mathbf{m}_0)(\bar{\mathbf{x}} - \mathbf{m}_0)^T && (3.24)\\
&= \mathbf{S}_0 + \mathbf{S} + \kappa_0\, \mathbf{m}_0\, \mathbf{m}_0^T - \kappa_N\, \mathbf{m}_N\, \mathbf{m}_N^T && (3.25)
\end{aligned}
$$

where $\mathbf{S} = \sum_{i=1}^{N} \mathbf{x}_i\, \mathbf{x}_i^T$ is the uncentered sum-of-squares matrix also known as the scatter matrix.

The posterior probability can be used to calculate Maximum a posteriori estimation (MAP):

$$
\begin{aligned}
\boldsymbol{\theta}_{MAP} &= \arg\max_{\boldsymbol{\theta}} \mathrm{p}(\boldsymbol{\theta} \,|\, \mathbf{X}) && (3.26)\\
&= \arg\max_{\boldsymbol{\theta}} \frac{\mathrm{p}(\mathbf{X} \,|\, \boldsymbol{\theta})\, \mathrm{p}(\boldsymbol{\theta})}{\mathrm{p}(\mathbf{X})} && (3.27)\\
&= \arg\max_{\boldsymbol{\theta}} \mathrm{p}(\mathbf{X} \,|\, \boldsymbol{\theta})\, \mathrm{p}(\boldsymbol{\theta}) && (3.28)
\end{aligned}
$$

MAP estimate for MVN [14, p. 134] is:

$$
\begin{aligned}
\hat{\boldsymbol{\mu}}_{MAP} &= \mathbf{m}_N && (3.29)\\
\hat{\boldsymbol{\Sigma}}_{MAP} &= \frac{\mathbf{S}_N}{\nu_N + D + 2} && (3.30)
\end{aligned}
$$

Posterior predictive

The posterior predictive probability gives information about a new value $\mathbf{x}^*$ before being observed:

$$
\mathrm{p}(\mathbf{x}^* \,|\, \mathbf{X}) = \frac{\mathrm{p}(\mathbf{x}^*, \mathbf{X})}{\mathrm{p}(\mathbf{X})} \qquad (3.31)
$$

In our case, it has the form of multivariate Student t distribution [14, p. 135]:

$$
\begin{aligned}
\mathrm{p}(\mathbf{x}^* \,|\, \mathbf{X}) &= \int\!\!\int \mathcal{N}\left(\mathbf{x}^* \,|\, \boldsymbol{\mu}, \boldsymbol{\Sigma}\right) \mathrm{NIW}\left(\boldsymbol{\mu}, \boldsymbol{\Sigma} \,|\, \mathbf{m}_N, \kappa_N, \nu_N, \mathbf{S}_N\right) d\,\boldsymbol{\mu}\, d\,\boldsymbol{\Sigma} && (3.32)\\
&= \mathcal{T}\left(\mathbf{x}^* \,|\, \mathbf{m}_N, \frac{\kappa_N + 1}{\kappa_N(\nu_N - D + 1)}\, \mathbf{S}_N, \nu_N - D + 1\right) && (3.33)
\end{aligned}
$$

Prior predictive

The prior predictive probability is a special case of the posterior predictive probability, when no data $\mathbf{X}$ are observed:

$$\mathrm{p}(\mathbf{x}^* \,|\emptyset) = \mathcal{T}\left(\mathbf{x}^* \,|\, \mathbf{m}_0, \frac{\kappa_0 + 1}{\kappa_0(\nu_0 - D + 1)}\mathbf{S}_0, \nu_0 - D + 1\right) \tag{3.34}$$

Marginal likelihood

The marginal likelihood $\mathrm{p}(\mathbf{X})$ [14, pp. 160–161] tells us how probable are the observed data $\mathbf{X}$ regardless of the parameters $\boldsymbol{\theta}$:

$$\mathrm{p}(\mathbf{X}) = (2\pi)^{-\frac{ND}{2}} \frac{Z_{\mathrm{NIW}}(D, \kappa_N, \nu_N, \mathbf{S}_N)}{Z_{\mathrm{NIW}}(D, \kappa_0, \nu_0, \mathbf{S}_0)} \tag{3.35}$$

$$= \pi^{-\frac{ND}{2}} \left(\frac{\kappa_0}{\kappa_N}\right)^{\frac{D}{2}} \frac{|\,\mathbf{S}_0\,|^{\frac{\nu_0}{2}}}{|\,\mathbf{S}_N\,|^{\frac{\nu_N}{2}}} \frac{\Gamma_D\left(\frac{\nu_N}{2}\right)}{\Gamma_D\left(\frac{\nu_0}{2}\right)} \tag{3.36}$$

$$= \pi^{-\frac{ND}{2}} \left(\frac{\kappa_0}{\kappa_N}\right)^{\frac{D}{2}} \frac{|\,\mathbf{S}_0\,|^{\frac{\nu_0}{2}}}{|\,\mathbf{S}_N\,|^{\frac{\nu_N}{2}}} \prod_{i=1}^{D} \frac{\Gamma\left(\frac{\nu_N+1-i}{2}\right)}{\Gamma\left(\frac{\nu_0+1-i}{2}\right)} \tag{3.37}$$

3.5 Multivariate Student *t* Distribution

The multivariate Student t distribution [14, p. 46] is a continuous probability distribution which generalizes the Student t distribution to higher dimensions. It is useful when we try to estimate the mean of a normally distributed population[2], the sample size is small, and the covariance matrix is unknown.

The probability density function is given by:

$$\mathcal{T}(\mathbf{x}|\,\boldsymbol{\mu}, \boldsymbol{\Sigma}, \nu) = \frac{\Gamma\left(\frac{\nu}{2} + \frac{D}{2}\right)}{\Gamma\left(\frac{\nu}{2}\right)} \frac{|\,\boldsymbol{\Sigma}\,|^{-\frac{1}{2}}}{\nu^{\frac{D}{2}}\pi^{\frac{D}{2}}} \left(1 + \frac{1}{\nu}(\mathbf{x} - \boldsymbol{\mu})^T\,\boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu})\right)^{-\frac{\nu+D}{2}} \tag{3.38}$$

where $\boldsymbol{\mu}$ is the mean vector, $\nu > 2$ is the degrees of freedom, and $\boldsymbol{\Sigma}$ is the scale matrix (since it is not exactly the covariance matrix).

This distribution has fatter tails than an MVN; and as $\nu \to \infty$, the distribution tends towards the MVN.

[2] A population has the (multivariate) normal distribution.

3.6 Wischart Distribution

The Wishart distribution [14, p. 125] is the distribution of the covariance matrix of independent samples drawn from MVN. It is a generalization of the $\mathcal{X}^2$ distribution (a special case of the gamma distribution) to multiple dimensions.

It is used to model our uncertainty in covariance matrices, $\mathbf{\Sigma}$, or their inverses, $\mathbf{\Lambda} = \mathbf{\Sigma}^{-1}$. The probability density function is defined as follows, for $\nu > D - 1$ and $\mathbf{S} \succ 0$:

$$\mathrm{Wi}\left(\mathbf{\Lambda} \,|\, \mathbf{S}, \nu\right) = \frac{1}{Z_{\mathrm{Wi}}} |\, \mathbf{\Lambda} \,|^{\frac{\nu-D-1}{2}} \exp\left(-\frac{1}{2}\operatorname{tr}(\mathbf{\Lambda}\,\mathbf{S}^{-1})\right) \tag{3.39}$$

$$Z_{\mathrm{Wi}} = 2^{\frac{\nu D}{2}} \Gamma_D\left(\frac{\nu}{2}\right) |\, \mathbf{S} \,|^{\frac{\nu}{2}} \tag{3.40}$$

where ν is degrees of freedom and $\mathbf{S}$ is the scatter matrix.

There is a connection between the Wishart distribution and the MVN. In particular, let $\mathbf{x}_i \sim \mathcal{N}\left(\mathbf{0}, \mathbf{\Sigma}\right)$. Then the scatter matrix of N independent draws $\mathbf{S} = \sum_{i=1}^{N} \mathbf{x}_i \, \mathbf{x}_i^T$ has a Wishart distribution $\mathbf{S} \sim \mathrm{Wi}\left(\mathbf{\Sigma}, N\right)$.

3.7 Inverse Wishart Distribution

The inverse Wishart distribution [14, p. 126] is a probability distribution defined on real-valued positive-definite matrices. It is a generalization of the inverse $\mathcal{X}^2$ distribution to multiple dimensions. It is used as the conjugate prior for the covariance matrix of an MVN in Bayesian statistics, and the probability density function is defined as follows, for $\nu > D - 1$ and $\mathbf{S} \succ 0$:

$$\mathrm{IW}\left(\mathbf{\Sigma} \,|\, \mathbf{S}, \nu\right) = \frac{1}{Z_{IW}} |\, \mathbf{\Sigma} \,|^{-\frac{\nu+D+1}{2}} \exp\left(-\frac{1}{2}\operatorname{tr}\left(\mathbf{S}^{-1}\,\mathbf{\Sigma}^{-1}\right)\right) \tag{3.41}$$

$$Z_{IW} = 2^{\frac{\nu D}{2}} \Gamma_D\left(\frac{\nu}{2}\right) |\, \mathbf{S} \,|^{-\frac{\nu}{2}} \tag{3.42}$$

where ν is degrees of freedom and $\mathbf{S}$ is the scatter matrix.

It is a fact that if $\mathbf{\Sigma}^{-1} \sim \mathrm{Wi}\left(\mathbf{S}, \nu\right)$ then $\mathbf{\Sigma} \sim \mathrm{IW}\left(\mathbf{S}^{-1}, \nu + D + 1\right)$.

4 Vector Representation for News Articles

Text unlike image (raw pixel-intensities) and audio (power spectral density coefficients) does not have a natural way of vector representation. For Natural language processing (NLP), methods for text encoding have been invented. Three semantic methods (LSA, LDA, and doc2vec) are described in this chapter, sorted by date of publication.

4.1 Latent Semantic Analysis

One of the oldest methods for term representation is one-hot representation. Initially, a dictionary (e.g. $100,000$ entries) is created, and each term is assigned a unique identifier (e.g. 123). This dummy variable can be thought of as a vector of length N equal to the size of the dictionary with exactly one 1 at position 123. This term representation, however, does not have any semantical information since we have no way of comparing any two terms with regard to similarity because the term representation is assigned arbitrarily.

For a given list of terms (a document), we can create a vector which combines vectors for each term into one (using weighting). This technique is known as Vector Space Model (VSM) (for more details see [11]) and it is based on bag-of-words (BoW)[1] hypothesis. Resulting term-by-document matrix $\boldsymbol{M}$ is huge and sparse. It is a $N \times D$ matrix where N is the number of terms and D is the number of documents. For that reason, the following method was invented.

Latent Semantic Analysis (LSA) [6], also known as Latent Semantic Indexing (LSI) is exactly equivalent to applying Principal Component Analysis (PCA) to the term-by-document matrix ([14, p. 947]). Singular Value Decomposition (SVD) [14, p. 392] is applied to VSM term-document matrix $\boldsymbol{M}$ in order to reduce its dimensionality:

[1] A text (a document) is represented as the bag (multiset) of its terms. In other words, it does not take a term order into account.

$$\underset{N \times D}{\overset{\boldsymbol{M}}{\begin{pmatrix} m_{1,1} & \cdots & m_{1,D} \\ \vdots & \ddots & \vdots \\ m_{N,1} & \cdots & m_{N,D} \end{pmatrix}}} = \underset{N \times D}{\overset{\boldsymbol{U}}{\begin{pmatrix} u_{1,1} & \cdots & u_{1,D} \\ \vdots & \ddots & \vdots \\ u_{N,1} & \cdots & u_{N,D} \end{pmatrix}}} \underset{D \times D}{\overset{\boldsymbol{\Sigma}}{\begin{pmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_D \end{pmatrix}}} \underset{D \times D}{\overset{\boldsymbol{V}^T}{\begin{pmatrix} v_{1,1} & \cdots & v_{1,D} \\ \vdots & \ddots & \vdots \\ v_{D,1} & \cdots & v_{D,D} \end{pmatrix}}} \tag{4.1}$$

$$\simeq \underset{N \times K}{\overset{\boldsymbol{U}_K}{\begin{pmatrix} u_{1,1} & \cdots & u_{1,K} \\ \vdots & \ddots & \vdots \\ u_{N,1} & \cdots & u_{N,K} \end{pmatrix}}} \underset{K \times K}{\overset{\boldsymbol{\Sigma}_K}{\begin{pmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_K \end{pmatrix}}} \underset{K \times D}{\overset{\boldsymbol{V}_K^T}{\begin{pmatrix} v_{1,1} & \cdots & v_{1,D} \\ \vdots & \ddots & \vdots \\ v_{K,1} & \cdots & v_{K,D} \end{pmatrix}}} \tag{4.2}$$

$$= \underset{N \times D}{\overset{\boldsymbol{M}_K}{\begin{pmatrix} \hat{m}_{1,1} & \cdots & \hat{m}_{1,D} \\ \vdots & \ddots & \vdots \\ \hat{m}_{N,1} & \cdots & \hat{m}_{N,D} \end{pmatrix}}} \tag{4.3}$$

If we select $K < D$ largest singular values and corresponding singular vectors from $\boldsymbol{U}$ and $\boldsymbol{V}$, we get an approximation to $\boldsymbol{M}$ with the smallest error measured by Frobenius norm:

$$\|M - M_K\|_F = \sqrt{\sum_{i=1}^{N} \sum_{j=1}^{D} |m_{ij} - \hat{m}_{ij}|^2} \tag{4.4}$$

After the decomposition, it has the following properties:

- Rows in $\boldsymbol{U}_k$ represent term semantics in a lower-dimensional space.
- Columns in $\boldsymbol{V}_k$ represent document semantics in a lower-dimensional space.
- New document can be mapped to lower-dimensional space using the formula $\hat{\boldsymbol{d}} = \boldsymbol{\Sigma}_K^{-1} \boldsymbol{U}_K^T \boldsymbol{d}$.

4.2 Latent Dirichlet Allocation

The Latent Dirichlet Allocation (LDA) [5] is a generative probabilistic model of a text corpus. Each document is represented as a random mixture of latent topics where each topic is characterized by a distribution over words.

The LDA assumes the following generative process for each document d in the corpus of N documents as visualized in figure 4.1:

1. Choose the number of words $M \sim \text{Poisson}(\xi)$.
2. Choose topic probabilities $\boldsymbol{\theta}_d \sim \text{Dir}(\boldsymbol{\alpha})$.
3. For each of M words $w_{d,i}$:
    (a) Choose a topic $z_{d,i} \sim \text{Cat}(\boldsymbol{\theta}_d)$.
    (b) Choose a word $w_{d,i} \sim \text{Cat}(\boldsymbol{\phi}_{z_{d,i}})$.

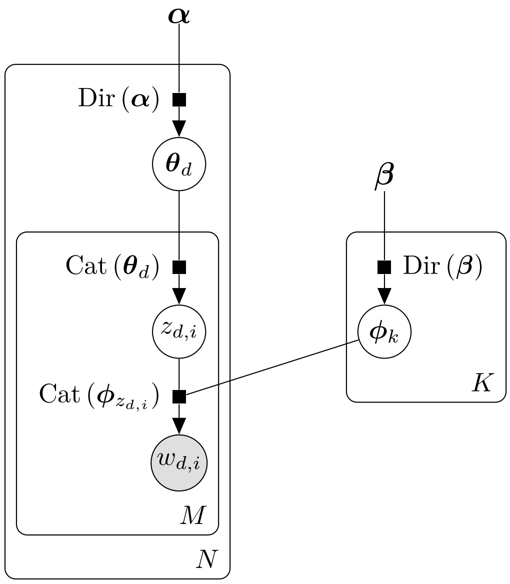
$\boldsymbol{\alpha}$
Dir $(\boldsymbol{\alpha})$
$\boldsymbol{\theta}_d$
$\boldsymbol{\beta}$
Cat $(\boldsymbol{\theta}_d)$
Dir $(\boldsymbol{\beta})$
$z_{d,i}$
$\boldsymbol{\phi}_k$
Cat $(\boldsymbol{\phi}_{z_{d,i}})$
K
$w_{d,i}$
M
N

Figure 4.1: LDA model

where:

- N is the number of documents in a text corpus.
- M is the number of words in document d.
- $\boldsymbol{\alpha}$ is hyperparameter of a random mixture of latent topics.
- $\boldsymbol{\beta}$ is hyperparameter of a distribution over words.
- $\boldsymbol{\theta}_d$ are topic probabilities for a given document d.
- $\boldsymbol{\phi}_k$ are word probabilities for a given topic k.
- $z_{d,i}$ is the topic of word i in document d.
- $w_{d,i}$ is an observed word i in document d.

4.3 doc2vec

A doc2vec model [10] is used to represent a variable length document as a fixed-length numeric vector in a way that related documents are close in a result vector space. This method is an extension of an existing technique word2vec [12] by adding a document unique feature vector (Paragraph id).

As you can see in figure 4.2, it is an extension of a Continuous Bag-of-Words (CBOW) model. In addition to word vector matrix $\boldsymbol{W}$ it also trains a document vector matrix $\boldsymbol{D}$ which contains a numeric representation of the document. In this model, the concatenation or average of paragraph vector with a context of three words is used to predict the fourth word.

The model is called Distributed Memory version of Paragraph Vector (PV-DM). Matrix $\boldsymbol{D}$ contains vectors representing a document context – the topic of the document, while the vectors in matrix $\boldsymbol{W}$ contain the concept of a word.

Alternatively, we could ignore word context and let the model predict words randomly sampled from the paragraph. This technique, which is similar to skip-gram model [12], is shown in figure 4.3 and it is called Distributed Bag of Words version of Paragraph Vector (PV-DBOW).

The authors recommend using a combination of both algorithms in order to achieve more consistent results.

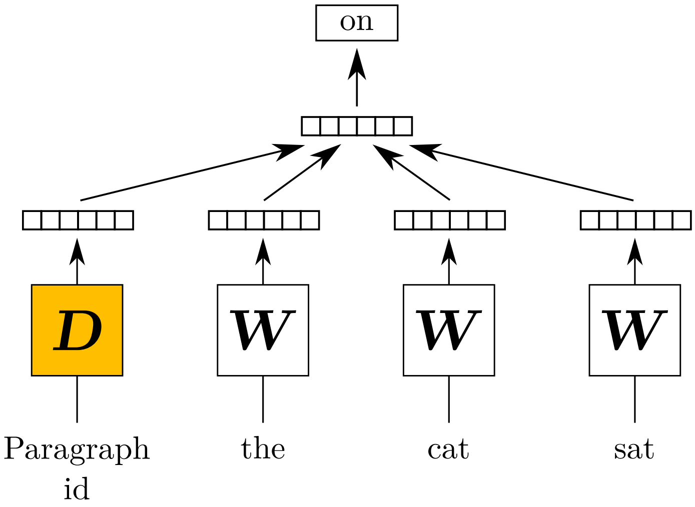

Figure 4.2: doc2vec PV-DM model

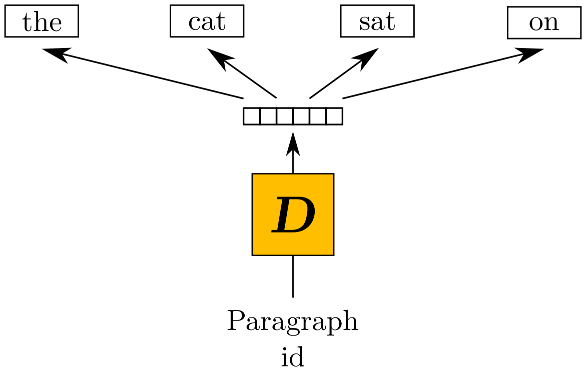

Figure 4.3: doc2vec PV-DBOW model

5 Model-based Clustering

Model-based clustering methods assume that the observed data $\mathbf{X}$ are a result of a generative process. These models which have hidden (latent) variables are called Latent Variable Models (LVMs) [14, p. 337].

The simplest form of LVM is a mixture model, which has discrete latent state $z_i \in \{1, \dots, K\}$, discrete prior $\mathrm{p}(z_i) = \mathrm{Cat}\,(\boldsymbol{\pi})$, and likelihood in form $\mathrm{p}(\mathbf{x}_i \,|z_i = k, \boldsymbol{\theta}_k) = \mathrm{p}_k(\mathbf{x}_i \,|\, \boldsymbol{\theta}_k)$ where p_k is the k-th base distribution for the observations.

The overall model, which mixes (convex combination[1] of p_k's) K base distributions, is defined as follows:

$$\mathrm{p}(\mathbf{x}_i \,|\, \boldsymbol{\theta}) = \sum_{k=1}^{K} \mathrm{p}(z_i = k|\boldsymbol{\pi})\,\mathrm{p}(\mathbf{x}_i \,|z_i = k, \boldsymbol{\theta}_k) \tag{5.1}$$

$$= \sum_{k=1}^{K} \pi_k\, \mathrm{p}_k(\mathbf{x}_i \,|\, \boldsymbol{\theta}_k) \tag{5.2}$$

The most widely used mixture model, despite the fact that it might be an oversimplification of reality, is the Gaussian mixture model (GMM) [14, p. 339]. In this model, each base distribution belonging to a certain cluster is the multivariate Gaussian (MVN) with mean $\boldsymbol{\mu}_k$ and covariance matrix $\boldsymbol{\Sigma}_k$:

$$\mathrm{p}(\mathbf{x}_i \,|\, \boldsymbol{\theta}) = \sum_{k=1}^{K} \pi_k\, \mathcal{N}\,(\mathbf{x}_i \,|\, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \tag{5.3}$$

This model is also known as Finite Gaussian Mixture Model (FGMM) whose graphical model is in figure 5.1.

The likelihood [3, p. 433] of GMM for N observations in dataset $\mathbf{X}$ expresses how well the model with given parameters matches the observed data. Often its logarithm is used. It is defined as follows:

$$\mathrm{p}(\mathbf{X} \,|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \prod_{i=1}^{N} \sum_{k=1}^{K} \pi_k\, \mathcal{N}\,(\mathbf{x}_i \,|\, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \tag{5.4}$$

$$\log \mathrm{p}(\mathbf{X} \,|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{i=1}^{N} \log \left(\sum_{k=1}^{K} \pi_k\, \mathcal{N}\,(\mathbf{x}_i \,|\, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right) \tag{5.5}$$

[1] Convex combination is a linear combination of mixing weights π_k where $\sum_{i=1}^{K} \pi_k = 1$ and $0 \le \pi_k \le 1$.

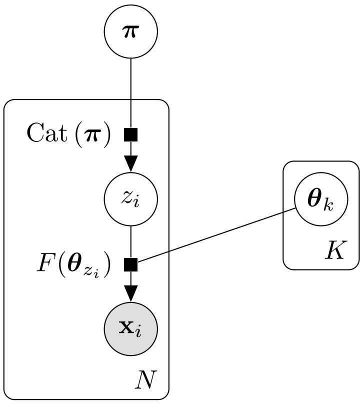

Figure 5.1: A graphical model of Finite Gaussian Mixture model

Gibbs sampling

The Gibbs sampling [14, p. 838] is a method often used to find an estimate of the model parameters. As opposed to using an expectation–maximization (EM) algorithm, which is a simple deterministic iterative algorithm, often with closed-form updates at each step, the Gibbs sampling is a stochastic iterative algorithm. It is one of the most popular Markov Chain Monte Carlo (MCMC) algorithms.

The basic idea behind MCMC is to construct a Markov chain on the state space $\mathcal{X}$ whose stationary distribution is the target density $\mathrm{p}^*(\mathbf{x})$. We perform a random walk in such a way making the fraction of time spend in each state $\mathbf{x}$ is proportional to p^*.

During the Gibbs sampling, we sample each variable in turn, conditional on the values of all other variables, and based on the most recent values of the other variables. For example, for $D = 3$ we use:

$$\begin{aligned} x_1^{s+1} &\sim \mathrm{p}(x_1|x_2^s, x_3^s) \\ x_2^{s+1} &\sim \mathrm{p}(x_2|x_1^{s+1}, x_3^s) \\ x_3^{s+1} &\sim \mathrm{p}(x_3|x_1^{s+1}, x_2^{s+1}) \end{aligned} \tag{5.6}$$

If x_i is an observed variable, we do not sample it, since its value is already known. The $\mathrm{p}(x_i|\,\mathbf{x}_{-i})$ is full conditional for variable i.

Since we can start the algorithm from an arbitrary initial state, the beginning of the chain should be thrown away [14, p. 856] because it is not sampled from a stationary distribution. It is, however, difficult to diagnose when the chain has burned in. It is one of the fundamental weaknesses of MCMC methods. As presented by Morris, Descombes, and Zerubia [13], the sampling before reaching convergence can lead to misleading conclusions.

5.1 Bayesian Gaussian Mixture Model

The Bayesian Gaussian Mixture Model (BGMM) use Bayesian inference to estimate the model parameters which are modelled as a random variable since a basic principle of Bayesian statistics is that all forms of uncertainty should be expressed as randomness.

The uncertainty of parameters is modelled by a prior. The difference between FGMM and BGMM can be seen in figures 5.1 and 5.2, respectively.

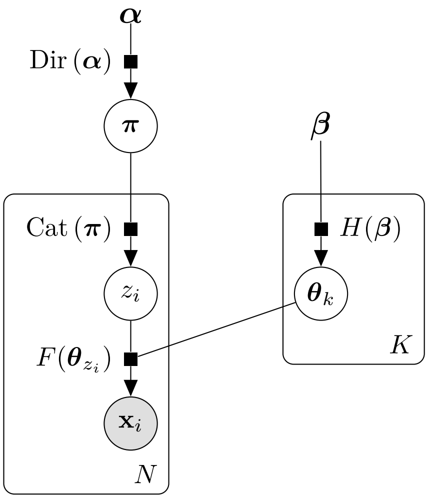

Figure 5.2: A graphical model of Bayesian Gaussian Mixture model

where:

- K is the number of clusters.
- N is the number of observations.
- $\boldsymbol{\pi} \sim \mathrm{Dir}\,(\boldsymbol{\alpha}) = \mathrm{Dir}\,(\frac{\alpha}{K}\mathbf{1}_K)$ are prior cluster probabilities.
- z_i is cluster assignment of observation i.
- $\boldsymbol{\theta}_k$ are latent cluster parameters which we are looking for.
- $\boldsymbol{\theta}_k \sim H(\boldsymbol{\beta}) = \mathrm{NIW}$ is parameters prior.
- $\mathbf{x}_i \sim F(\boldsymbol{\theta}_{z_i}) = \mathcal{N}$ is base cluster distribution.
- $\mathbf{x}_i$ is i-th observation.

5.1.1 Collapsed Gibbs Sampling for GMM

Consider a GMM with a fully conjugate prior (eq. 3.15). A collapsed Gibbs sampler [14, p. 842] has analytically integrated out the model parameters $\boldsymbol{\mu}_k$, $\boldsymbol{\Sigma}_k$, and $\boldsymbol{\pi}$. Then we sample just cluster assignments $\mathbf{z}$.

Then, the full conditional can be derived as follows:

$$
\begin{aligned}
\mathrm{p}(z_i = k | \, \mathbf{z}_{-i}, \mathbf{X}, \boldsymbol{\alpha}, \boldsymbol{\beta}) &\propto \mathrm{p}(z_i = k | \, \mathbf{z}_{-i}, \boldsymbol{\alpha}, \cancel{\boldsymbol{\beta}}) \, \mathrm{p}(\mathbf{X} \, | z_i = k, \mathbf{z}_{-i}, \cancel{\boldsymbol{\alpha}}, \boldsymbol{\beta}) && (5.7) \\
&\propto \mathrm{p}(z_i = k | \, \mathbf{z}_{-i}, \boldsymbol{\alpha}) \, \mathrm{p}(\mathbf{x}_i \, | \, \mathbf{X}_{-i}, z_i = k, \mathbf{z}_{-i}, \boldsymbol{\beta}) && (5.8) \\
&\quad \mathrm{p}(\mathbf{X}_{-i} \, | \cancel{z_i = k}, \mathbf{z}_{-i}, \boldsymbol{\beta}) && (5.9) \\
&\propto \mathrm{p}(z_i = k | \, \mathbf{z}_{-i}, \boldsymbol{\alpha}) \, \mathrm{p}(\mathbf{x}_i \, | \, \mathbf{X}_{-i}, z_i = k, \mathbf{z}_{-i}, \boldsymbol{\beta}) && (5.10)
\end{aligned}
$$

where $\boldsymbol{\beta} = (\mathbf{m}_0, \kappa_0, \nu_0, \mathbf{S}_0)$ is the hyper-parameter of cluster prior. Cluster assignment prior is symmetric $\boldsymbol{\pi} \sim \mathrm{Dir}\,(\boldsymbol{\alpha})$ where $\alpha_k = \frac{\alpha}{K}$.

Term $\mathrm{p}(z_i = k | \, \mathbf{z}_{-i}, \boldsymbol{\alpha})$

The first term in the equation 5.10 is a mixture probability. It can be shown that it is a marginal likelihood for Dirichlet-multinoulli model [14, p. 160]:

$$
\begin{aligned}
\mathrm{p}(\mathbf{z} \, | \boldsymbol{\alpha}) &= \mathrm{p}(z_1, \dots, z_N | \alpha) && (5.11) \\
&= \frac{\Gamma(\alpha)}{\Gamma(N + \alpha)} \prod_{k=1}^{K} \frac{\Gamma\left(N_k + \frac{\alpha}{K}\right)}{\Gamma\left(\frac{\alpha}{K}\right)} && (5.12)
\end{aligned}
$$

Thus from conditional probability, we can derive an expression for the desired term:

$$
\begin{aligned}
\mathrm{p}(z_i = k | \, \mathbf{z}_{-i}, \boldsymbol{\alpha}) &= \frac{\mathrm{p}(z_i = k, \mathbf{z}_{-i} \, | \boldsymbol{\alpha})}{\mathrm{p}(\mathbf{z}_{-i} \, | \boldsymbol{\alpha})} && (5.13) \\
&= \frac{\mathrm{p}(\mathbf{z} \, | \boldsymbol{\alpha})}{\mathrm{p}(\mathbf{z}_{-i} \, | \boldsymbol{\alpha})} && (5.14) \\
&= \frac{\Gamma(N + \alpha - 1)}{\Gamma(N + \alpha)} \frac{\Gamma\left(N_k + \frac{\alpha}{K}\right)}{\Gamma\left(N_{k,-i} + \frac{\alpha}{K}\right)} && (5.15) \\
&= \frac{\Gamma(N + \alpha - 1)}{\Gamma(N + \alpha)} \frac{\Gamma\left(N_{k,-i} + 1 + \frac{\alpha}{K}\right)}{\Gamma\left(N_{k,-i} + \frac{\alpha}{K}\right)} && (5.16) \\
&= \frac{N_{k,-i} + \frac{\alpha}{K}}{N + \alpha - 1} && (5.17)
\end{aligned}
$$

where we exploited the fact that $\Gamma(x+1) = x\Gamma(x)$ and $N_{k,-i} = \sum_{n \neq i} \mathbb{I}(z_n = k) = N_k - 1$.

Term $\mathrm{p}(\mathbf{x}_i \,|\, \mathbf{X}_{-i}, z_i = k, \mathbf{z}_{-i}, \boldsymbol{\beta})$

The second term in the equation 5.10 is a posterior predictive dictistribution [14, p. 843]:

$$\mathrm{p}(\mathbf{x}_i \,|\, \mathbf{X}_{-i}, z_i = k, \mathbf{z}_{-i}, \boldsymbol{\beta}) = \mathrm{p}(\mathbf{x}_i \,|\, \mathbf{X}_{-i,k}) \tag{5.18}$$

where $\mathbf{X}_{-i,k} = \{\mathbf{x}_j : z_j = k, j \neq i\}$ are the observations without taking a observation $\mathbf{x}_i$ in account.

For GMM we use equation 3.33.

Pseudocode

Pseudo code for the collapsed Gibbs sampler for BGMM is given in algorithm 1:

Algorithm 1 Collapsed Gibbs sampler for BGMM

```
Choose initial cluster assignments $\mathbf{z}$
for $T$ iterations do
    for each observation $\mathbf{x}_i$ $i = 1 : N$ in random order do
        Remove $\mathbf{x}_i$'s sufficient statistics from old cluster $z_i$
        If any cluster is empty, remove it and decrease $K$
        for each cluster $k = 1 : K$ do
            Calculate $\mathrm{p}(z_i = k | \mathbf{z}_{-i}, \mathbf{X}, \boldsymbol{\alpha}, \boldsymbol{\beta})$
        end for
        Normalize $\mathrm{p}(z_i | \mathbf{z}_{-i}, \mathbf{X}, \boldsymbol{\alpha}, \boldsymbol{\beta})$
        Sample $z_i \sim \mathrm{Cat}\left(\mathrm{p}(z_i | \mathbf{z}_{-i}, \mathbf{X}, \boldsymbol{\alpha}, \boldsymbol{\beta})\right)$
        Add $\mathbf{x}_i$'s sufficient statistics to new cluster $z_i$
    end for
end for
```

where sampled expression is a full conditional from equation 5.10:

$$\begin{aligned}
\mathrm{p}(z_i = k | \mathbf{z}_{-i}, \mathbf{X}, \boldsymbol{\alpha}, \boldsymbol{\beta}) &\propto \mathrm{p}(z_i = k | \mathbf{z}_{-i}, \boldsymbol{\alpha})\, \mathrm{p}(\mathbf{x}_i \,|\, \mathbf{X}_{-i}, z_i = k, \mathbf{z}_{-i}, \boldsymbol{\beta}) && (5.19) \\
&\propto \frac{N_{k,-i} + \frac{\alpha}{K}}{N + \alpha - 1}\, \mathrm{p}(\mathbf{x}_i \,|\, \mathbf{X}_{-i,k}) && (5.20) \\
&\propto \left(N_{k,-i} + \frac{\alpha}{K} \right) \mathrm{p}(\mathbf{x}_i \,|\, \mathbf{X}_{-i,k}) && (5.21)
\end{aligned}$$

5.2 Infinite Gaussian Mixture Model

The Infinite Gaussian Mixture Model (IGMM) is a nonparametric Bayesian model whose parameter space has infinite dimension.

5.2.1 The Chinese Restaurant Process

The Chinese Restaurant Process (CRP) [14, p. 884] is a stochastic process analogous to seating customers at tables in a Chinese restaurant, as shown in figure 5.3: The tables represent clusters, customers are observations. A customer enters the restaurant and has two options. Either he joins an existing table with probability proportional to the number of people already sitting at this table (N_k), or he sits at the new table with a probability which diminishes as more customers enter the restaurant.

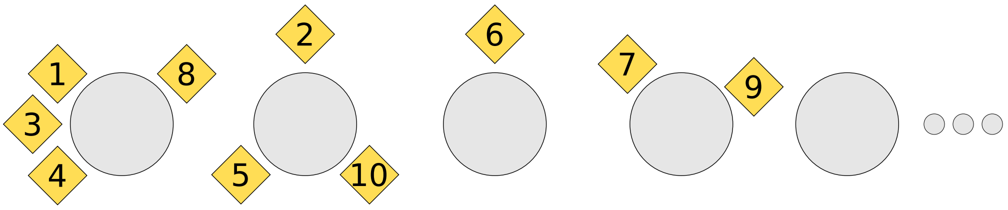

Figure 5.3: An illustration of the table assignments **z** in the CRP. The process operates at the level of table assignments, where each customer chooses either existing table or sits alone.

It can be shown [14, p. 886], that the CRP, as shown in figure 5.4, is a modification of the Gibbs sampling for BGMM from section 5.1.1.

Term $\mathrm{p}(z_i = k |\, \mathbf{z}_{-i}, \boldsymbol{\alpha})$

By exchangeability, we can assume that z_i is the last customer to enter the restaurant. The mixture probability is given by:

$$\mathrm{p}(z_i = k |\, \mathbf{z}_{-i}, \boldsymbol{\alpha}) = \begin{cases} \frac{N_{k,-i}}{N+\alpha-1} & \text{if } k \text{ has been seen before} \\ \frac{\alpha}{N+\alpha-1} & \text{if } k \text{ is a new cluster} \end{cases} \tag{5.22}$$

It is equal to equation 5.17 for $K \to \infty$.

Term $\mathrm{p}(\mathbf{x}_i \,|\, \mathbf{X}_{-i}, z_i = k, \mathbf{z}_{-i}, \boldsymbol{\beta})$

For an existing cluster, the posterior predictive is exactly the same as the equation 5.18:

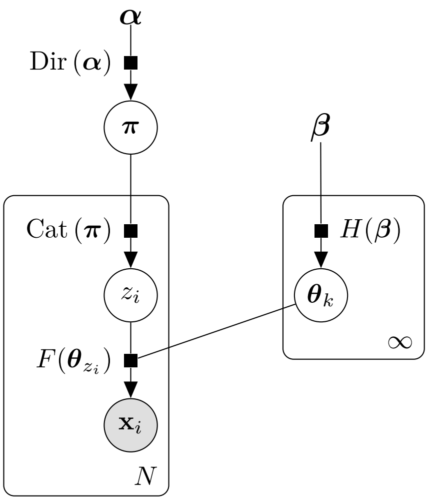

Figure 5.4: The Chinese restaurant process model

$$\mathrm{p}(\mathbf{x}_i \,|\, \mathbf{X}_{-i}, z_i = k, \mathbf{z}_{-i}, \boldsymbol{\beta}) = \mathrm{p}(\mathbf{x}_i \,|\, \mathbf{X}_{-i,k}) \tag{5.23}$$

For a new cluster, it is a prior predictive (eq. 3.34):

$$\mathrm{p}(\mathbf{x}_i \,|\, \mathbf{X}_{-i}, z_i = k^{(new)}, \mathbf{z}_{-i}, \boldsymbol{\beta}) = \mathrm{p}(\mathbf{x}_i \,|\emptyset) \tag{5.24}$$

Pseudocode

Pseudo code for the collapsed Gibbs sampler for CRP is given in algorithm 2 where full conditionals are:

$$\begin{aligned}
\mathrm{p}(z_i = k | \, \mathbf{z}_{-i}, \mathbf{X}, \boldsymbol{\alpha}, \boldsymbol{\beta}) &\propto \ \mathrm{p}(z_i = k | \, \mathbf{z}_{-i}, \boldsymbol{\alpha}) \, \mathrm{p}(\mathbf{x}_i \,|\, \mathbf{X}_{-i}, z_i = k, \mathbf{z}_{-i}, \boldsymbol{\beta}) && (5.25)\\
&\propto \frac{N_{k,-i}}{N + \alpha - 1} \, \mathrm{p}(\mathbf{x}_i \,|\, \mathbf{X}_{-i,k}) && (5.26)\\
&\propto N_{k,-i} \, \mathrm{p}(\mathbf{x}_i \,|\, \mathbf{X}_{-i,k}) && (5.27)
\end{aligned}$$

$$\begin{aligned}
\mathrm{p}(z_i = k^{(new)} | \, \mathbf{z}_{-i}, \mathbf{X}, \boldsymbol{\alpha}, \boldsymbol{\beta}) &\propto \ \mathrm{p}(z_i = k^{(new)} | \, \mathbf{z}_{-i}, \boldsymbol{\alpha}) && (5.28)\\
&\quad\ \mathrm{p}(\mathbf{x}_i \,|\, \mathbf{X}_{-i}, z_i = k, \mathbf{z}_{-i}, \boldsymbol{\beta}) && (5.29)\\
&\propto \frac{\alpha}{N + \alpha - 1} \, \mathrm{p}(\mathbf{x}_i \,|\emptyset) && (5.30)\\
&\propto \alpha \, \mathrm{p}(\mathbf{x}_i \,|\emptyset) && (5.31)
\end{aligned}$$

Algorithm 2 Collapsed Gibbs sampler for a CRP

```
Choose initial cluster assignments $\mathbf{z}$
for $T$ iterations do
    for each observation $\mathbf{x}_i$ $i = 1 : N$ in random order do
        Remove $\mathbf{x}_i$'s sufficient statistics from old cluster $z_i$
        If any cluster is empty, remove it and decrease $K$
        for each cluster $k = 1 : K$ do
            Calculate $\mathrm{p}(z_i = k|\, \mathbf{z}_{-i}, \mathbf{X}, \boldsymbol{\alpha}, \boldsymbol{\beta})$
        end for
        Calculate $\mathrm{p}(z_i = k^{(new)}|\, \mathbf{z}_{-i}, \mathbf{X}, \boldsymbol{\alpha}, \boldsymbol{\beta})$
        Normalize $\mathrm{p}(z_i|\, \mathbf{z}_{-i}, \mathbf{X}, \boldsymbol{\alpha}, \boldsymbol{\beta})$
        Sample $z_i \sim \mathrm{Cat}\left(\mathrm{p}(z_i|\, \mathbf{z}_{-i}, \mathbf{X}, \boldsymbol{\alpha}, \boldsymbol{\beta})\right)$
        Add $\mathbf{x}_i$'s sufficient statistics to new cluster $z_i$
    end for
end for
```

5.2.2 The Distance Dependent Chinese Restaurant Process

The Distance Dependent Chinese Restaurant Process (ddCRP) [4] is a stochastic process analogous to seating customers at tables in a restaurant, as shown in figure 5.5: The tables represent clusters, customers are observations. The main difference to the CRP is that a customer entering the restaurant does not join a table but sits next to one of the customers who is already in the restaurant or he sits by himself. The table allocation $z(\mathbf{c})$ is a by-product of this representation. If two customers are reachable by a sequence of interim customer assignments, then they sit at the same table.

The Gibbs sampler for the ddCRP is:

$$\mathrm{p}(c_i = j|\, \mathbf{c}_{-i}, \mathbf{X}, \eta) \propto \begin{cases} \alpha & \text{a)} \\ f(d_{ij}) & \text{b)} \\ f(d_{ij}) \frac{\mathrm{p}(\mathbf{X}_{z^k(\mathbf{c}_{-i}) \cup z^l(\mathbf{c}_{-i})} | G_0)}{\mathrm{p}(\mathbf{X}_{z^k(\mathbf{c}_{-i})} | G_0)\, \mathrm{p}(\mathbf{X}_{z^l(\mathbf{c}_{-i})} | G_0)} & \text{c)} \end{cases} \tag{5.32}$$

where

a) if c_i is equal to i

b) if $c_i = j$ does not join two tables

c) if $c_i = j$ joins tables k and l

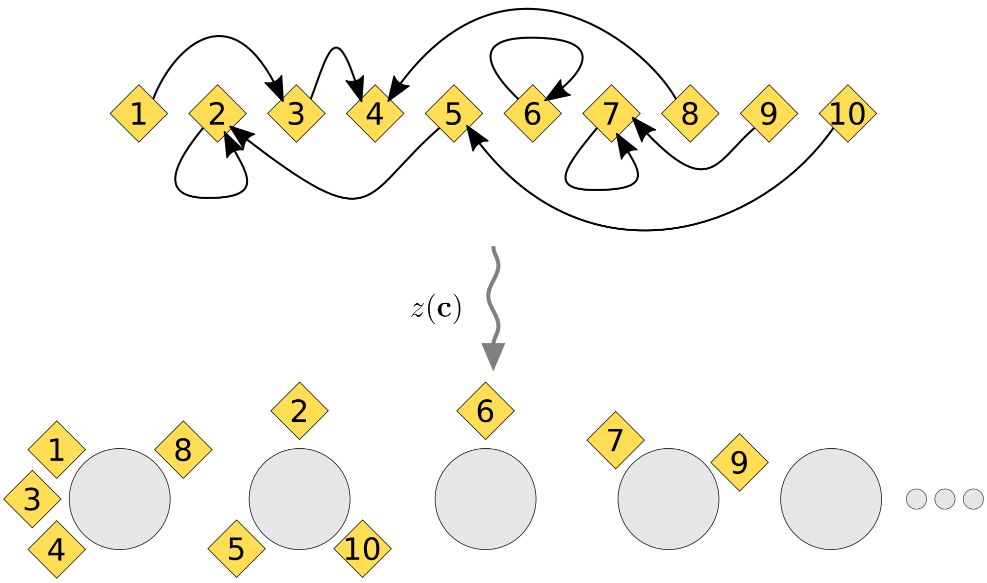

Figure 5.5: An illustration of the customer assignments $\mathbf{c}$ and table assignments $z(\mathbf{c})$ in the ddCRP. The process operates at the level of customer assignments, where each customer chooses either another customer or himself (self link). The table assignments are given by $z(\mathbf{c})$.

and where $\eta = \{D, \alpha, f, G_0\}$ is set of hyperparameters, G_0 is base measure, c_i denote the i-th customer assignment, d_{ij} denote the distance measurement between customers i and j, $f(d_{ij})$ is a decay function influencing the willingness of customer i to sit next to customer j, and $\mathrm{p}(\mathbf{X}_{z^k(\mathbf{c})} | G_0)$ is marginal likelihood:

$$\mathrm{p}(\mathbf{X}_{z^k(\mathbf{c})} | G_0) = \int \mathrm{p}(\mathbf{X}_{z^k(\mathbf{c})} | \theta)\, \mathrm{p}(\theta | G_0) d\theta \tag{5.33}$$

$$= \int \left(\prod_{i \in z^k(\mathbf{c})} \mathrm{p}(\mathbf{x}_i | \theta) \right) \mathrm{p}(\theta | G_0) d\theta \tag{5.34}$$

where we use equation 3.37 for GMM.

Pseudo code for the collapsed Gibbs sampler for ddCRP is given in algorithm 3:

Algorithm 3 Collapsed Gibbs sampler for a ddCRP

```
Choose initial customer assignments $\mathbf{c}$
for $T$ iterations do
    for each observation $\mathbf{x}_i$ $i = 1 : N$ in random order do
        Remove customer assignment $c_i$ (may split tables)
        for each observation $\mathbf{x}_j$ $j = 1 : N$ do
            Calculate $\mathrm{p}(c_i = j |\, \mathbf{c}_{-i}, \mathbf{X}, \eta)$
        end for
        Normalize $\mathrm{p}(c_i |\, \mathbf{c}_{-i}, \mathbf{X}, \eta)$
        Sample $c_i \sim \mathrm{Cat}\,(\mathrm{p}(c_i |\, \mathbf{c}_{-i}, \mathbf{X}, \eta))$
        Add new customer assignment $c_i$ (may join tables)
    end for
end for
```

6 Evaluation of Clustering

The evaluation of clustering is the most difficult part of cluster analysis. Clustering is an unsupervised learning technique, therefore it is hard to evaluate the quality of the output of given methods.

Intuitively, the goal of clustering is to assign similar observations to the same cluster, and to ensure that dissimilar observations are in different clusters.

There are several groups we can divide the evaluating techniques into:

- **Internal criteria** measure the validity of clustering without external information.
- **External criteria** measure the validity of clustering with external information (the ground truth), as depicted in figure 6.1.
- **Information criteria** measure the validity of clustering if a statistical model was used.
- **Manual criteria** measure the validity of clustering using an expert opinion.

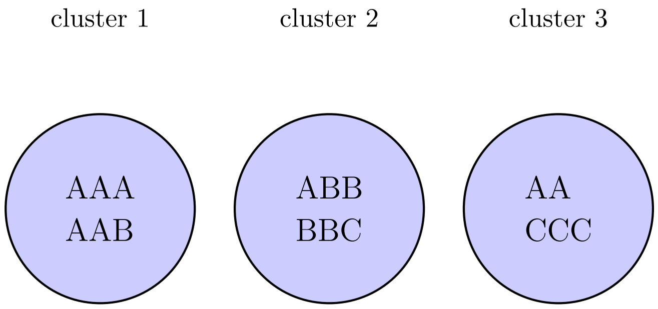

Figure 6.1: Labeled clustering example [11]

6.1 Internal Criteria

6.1.1 Likelihood

The likelihood of N observations in dataset $\mathbf{X}$ expresses how well the model with given parameters matches the observed data. Often its logarithm is used. It is defined as follows:

$$\mathrm{p}(\mathbf{X} \,|\, \boldsymbol{\theta}) = \prod_{i=1}^{N} \mathrm{p}(\mathbf{x}_i \,|\, \boldsymbol{\theta}) \tag{6.1}$$

$$\log \mathrm{p}(\mathbf{X} \,|\, \boldsymbol{\theta}) = \sum_{i=1}^{N} \log \mathrm{p}(\mathbf{x}_i \,|\, \boldsymbol{\theta}) \tag{6.2}$$

The likelihood of the GMM follows equations 5.4 and 5.5:

$$\mathrm{p}(\mathbf{X} \,|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \prod_{i=1}^{N} \sum_{k=1}^{K} \pi_k \mathcal{N}\left(\mathbf{x}_i \,|\, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right) \tag{6.3}$$

$$\log \mathrm{p}(\mathbf{X} \,|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{i=1}^{N} \log \left(\sum_{k=1}^{K} \pi_k \mathcal{N}\left(\mathbf{x}_i \,|\, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right) \right) \tag{6.4}$$

6.1.2 Entropy

The entropy [14, p. 56] of a random variable $\boldsymbol{\Omega}$ (cluster assignments) with distribution p is a measure of uncertainty, and is defined as follows using an MLE of the probabilities [11]:

$$\mathrm{H}\left(\boldsymbol{\Omega}\right) = -\sum_{i=1}^{K} \mathrm{p}(\omega_i) \log \mathrm{p}(\omega_i) \tag{6.5}$$

$$= -\sum_{i=1}^{K} \frac{|\omega_i|}{N} \log \frac{|\omega_i|}{N} \tag{6.6}$$

If logarithm base 2 is used the units are called bits, if logarithm base e is used the units are called nats.

6.2 External Criteria

6.2.1 Conditional Entropy

The conditional entropy on true class labels **C** and vice versa is sometimes needed [17]:

$$\mathrm{H}\left(\mathbf{\Omega} \,|\, \mathbf{C}\right) = -\sum_{j=1}^{C}\sum_{i=1}^{K} \mathrm{p}_{\mathbf{\Omega}\,\mathbf{C}}(i,j) \log \frac{\mathrm{p}_{\mathbf{\Omega}\,\mathbf{C}}(i,j)}{\mathrm{p}_{\mathbf{C}}(j)} \tag{6.7}$$

$$= -\sum_{j=1}^{C}\sum_{i=1}^{K} \frac{|\omega_i \cap c_j|}{N} \log \frac{|\omega_i \cap c_j|}{|c_j|} \tag{6.8}$$

$$\mathrm{H}\left(\mathbf{C} \,|\, \mathbf{\Omega}\right) = -\sum_{j=1}^{C}\sum_{i=1}^{K} \mathrm{p}_{\mathbf{\Omega}\,\mathbf{C}}(i,j) \log \frac{\mathrm{p}_{\mathbf{\Omega}\,\mathbf{C}}(i,j)}{\mathrm{p}_{\mathbf{\Omega}}(j)} \tag{6.9}$$

$$= -\sum_{j=1}^{C}\sum_{i=1}^{K} \frac{|\omega_i \cap c_j|}{N} \log \frac{|\omega_i \cap c_j|}{|\omega_i|} \tag{6.10}$$

where:

- $\mathrm{p}_{\mathbf{\Omega}\,\mathbf{C}}(i,j) = \frac{|\omega_i \cap c_j|}{N}$ is the probability that a randomly chosen object belongs to cluster ω_i in $\mathbf{\Omega}$ and class c_j in **C** (class labels).

- $\mathrm{p}_{\mathbf{\Omega}}(i) = \frac{|\omega_i|}{N}$ is the probability that a randomly chosen object belongs to cluster ω_i in $\mathbf{\Omega}$.

- $\mathrm{p}_{\mathbf{C}}(j) = \frac{|c_j|}{N}$ is the probability that a randomly chosen object belongs to class c_j in **C**.

6.2.2 Purity

One of the favourite cluster evaluating measures is purity [14, p. 877]. It says how "pure" the clusters are. In other words, it penalizes clusters which contains observations not belonging to this cluster.

$$\mathrm{purity}\left(\mathbf{\Omega}, \mathbf{C}\right) = \sum_{i=1}^{K} \frac{N_i}{N}\, \mathrm{p}_i \tag{6.11}$$

$$= \sum_{i=1}^{K} \frac{N_i}{N} \max_j \frac{N_{ij}}{N_i} \tag{6.12}$$

$$= \frac{1}{N} \sum_{i=1}^{K} \max_j N_{ij} \tag{6.13}$$

where:

- $\mathbf{\Omega} = \{\omega_1, \ldots, \omega_K\}$ is set of clusters.
- $\mathbf{C} = \{\omega_1, \ldots, \omega_C\}$ is set of classes.
- N is the total number of objects.
- N_{ij} is the number of objects in cluster i which belong to class j.
- $N_i = \sum_{j=1}^{C} N_{ij}$ is total number of objects in cluster i.
- $\mathrm{p}_{ij} = \frac{N_{ij}}{N_i}$ is empirical distribution over class labels for cluster i.
- $\mathrm{p}_i = \max_j \mathrm{p}_{ij}$ is the purity of a cluster.

The purity of example clustering (fig. 6.1) is $\mathrm{purity}(\mathbf{\Omega}, \mathbf{C}) = \frac{5+4+3}{17} \approx 0.706$.

6.2.3 Rand Index

Another technique for cluster evaluation is called the Rand index [14, p. 877]. It penalizes both false positive and false negative decisions during clustering [11].

$$\mathrm{RI}(\mathbf{\Omega}, \mathbf{C}) = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{FP} + \mathrm{FN} + \mathrm{TN}} \tag{6.14}$$

For an example clustering depicted in figure 6.1, the contingency table 6.1 is calculated using the following equations.

The number of pairs of objects put in the same cluster regardless of class label is:

$$\mathrm{TP} + \mathrm{FP} = \binom{6}{2} + \binom{6}{2} + \binom{5}{2} = 40 \tag{6.15}$$

The number of pairs of objects put in the same class regardless of cluster label is:

$$\mathrm{TP} + \mathrm{FN} = \binom{8}{2} + \binom{5}{2} + \binom{4}{2} = 44 \tag{6.16}$$

Table 6.1: Contingency table for example clustering (fig. 6.1)

		True class	
		Same class	Different classes
Predicted cluster	Same cluster	TP = 20	FP = 20
	Different clusters	FN = 24	TN = 72

The number of pairs of objects put in the same class with the same cluster label is:

$$\mathrm{TP} = \binom{5}{2} + \binom{4}{2} + \binom{3}{2} + \binom{2}{2} = 20 \tag{6.17}$$

And the total number of pairs is:

$$\mathrm{TP} + \mathrm{FP} + \mathrm{FN} + \mathrm{TN} = \binom{N}{2} = \binom{17}{2} = 136 \tag{6.18}$$

Then, the number of true negatives TN is:

$$\begin{aligned}\mathrm{TN} &= (\mathrm{TP} + \mathrm{FP} + \mathrm{FN} + \mathrm{TN}) - (\mathrm{TP} + \mathrm{FP}) - (\mathrm{TP} + \mathrm{FN}) + \mathrm{TP} && (6.19)\\ &= 136 - 40 - 44 + 20 = 72 && (6.20)\end{aligned}$$

Finally, $\mathrm{RI}\left(\mathbf{\Omega}, \mathbf{C}\right) = \frac{20+72}{136} \approx 0.676$ since there is no actual need for calculating FP and FN.

6.2.4 F-measure

Having a contingency table 6.1, an F-measure [11] metrics for clustering, which supports differential weighting of these two types of errors (false positive and false negative), can be calculated.

The precision is defined as follows:

$$\mathrm{P}\left(\mathbf{\Omega}, \mathbf{C}\right) = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{6.21}$$

The recall is defined as follows:

$$\mathrm{R}\left(\mathbf{\Omega}, \mathbf{C}\right) = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{6.22}$$

The F-measure is a weighted combination of precision and recall:

$$\begin{aligned}\mathrm{F}_\beta(\mathbf{\Omega}, \mathbf{C}) &= \frac{(1 + \beta^2)\,\mathrm{P}\,\cdot \mathrm{R}}{\beta^2\,\mathrm{P}\, + \mathrm{R}} && (6.23)\\ &= \frac{(1 + \beta^2)\,\mathrm{TP}}{(1 + \beta^2)\,\mathrm{TP}\, + \beta^2\,\mathrm{FN} + \mathrm{FP}} && (6.24)\end{aligned}$$

The F1-measure is a special, symmetrical, and widely used case of F-measure with $\beta = 1$:

$$\begin{aligned}\mathrm{F}_1(\mathbf{\Omega}, \mathbf{C}) &= 2\frac{\mathrm{P}\,\cdot \mathrm{R}}{\mathrm{P}\, + \mathrm{R}} && (6.25)\\ &= \frac{2\,\mathrm{TP}}{2\,\mathrm{TP} + \mathrm{FN} + \mathrm{FP}} && (6.26)\end{aligned}$$

For an example clustering in figure 6.1 we get $F_1(\mathbf{\Omega}, \mathbf{C}) = \frac{2\cdot 20}{2\cdot 20+24+20} \approx 0.476$.

6.2.5 Mutual Information

Another way to measure cluster quality is to compute the mutual information [14, p. 878] between $\mathbf{\Omega}$ and $\mathbf{C}$. It can be information-theoretically interpreted. The mutual information is defined as:

$$\mathrm{I}\left(\mathbf{\Omega}, \mathbf{C}\right) = \sum_{i=1}^{K} \sum_{j=1}^{C} \mathrm{p}_{\mathbf{\Omega}\,\mathbf{C}}(i,j) \log \frac{\mathrm{p}_{\mathbf{\Omega}\,\mathbf{C}}(i,j)}{\mathrm{p}_{\mathbf{\Omega}}(i)\,\mathrm{p}_{\mathbf{C}}(j)} \tag{6.27}$$

where we use MLE:

- $\mathrm{p}_{\mathbf{\Omega}\,\mathbf{C}}(i,j) = \frac{|\omega_i \cap c_j|}{N}$ is the probability that a randomly chosen object belongs to cluster ω_i in $\mathbf{\Omega}$ and class c_j in $\mathbf{C}$.
- $\mathrm{p}_{\mathbf{\Omega}}(i) = \frac{|\omega_i|}{N}$ is the probability that a randomly chosen object belongs to cluster ω_i in $\mathbf{\Omega}$.
- $\mathrm{p}_{\mathbf{C}}(j) = \frac{|c_j|}{N}$ is the probability that a randomly chosen object belongs to class c_j in $\mathbf{C}$.

The mutual information has the following properties:

$$\begin{aligned} \mathrm{I}\left(\mathbf{\Omega}, \mathbf{C}\right) &= \mathrm{H}\left(\mathbf{\Omega}\right) + \mathrm{H}\left(\mathbf{C}\right) - \mathrm{H}\left(\mathbf{C}, \mathbf{\Omega}\right) && (6.28) \\ &= \mathrm{H}\left(\mathbf{C}\right) - \mathrm{H}\left(\mathbf{C} \,|\, \mathbf{\Omega}\right) && (6.29) \\ &= \mathrm{H}\left(\mathbf{\Omega}\right) - \mathrm{H}\left(\mathbf{\Omega} \,|\, \mathbf{C}\right) && (6.30) \end{aligned}$$

6.2.6 Normalized Mutual Information

One disadvantage of the mutual information is that the maximum value of I (sec. 6.2.5) can be achieved by using lots of small clusters, which have low entropy. To compensate for this, the normalized mutual information [14, p. 879] is used:

$$\mathrm{NMI}\left(\mathbf{\Omega}, \mathbf{C}\right) = \frac{\mathrm{I}\left(\mathbf{\Omega}, \mathbf{C}\right)}{\frac{\mathrm{H}\left(\mathbf{\Omega}\right) + \mathrm{H}\left(\mathbf{C}\right)}{2}} \tag{6.31}$$

where $\mathrm{H}\left(\mathbf{\Omega}\right)$ is entropy described in section 6.1.2.

Alternatively, we could define the normalized mutual information [19] as:

$$\mathrm{NMI}\left(\mathbf{\Omega}, \mathbf{C}\right) = \frac{\mathrm{I}\left(\mathbf{\Omega}, \mathbf{C}\right)}{\sqrt{\mathrm{H}\left(\mathbf{\Omega}\right) \mathrm{H}\left(\mathbf{C}\right)}} \tag{6.32}$$

6.2.7 Homogeneity

A clustering result satisfies homogeneity if all of its clusters contain only data points which are members of a single class [17], and it is defined as follows:

$$\mathrm{h}(\mathbf{\Omega}, \mathbf{C}) = \begin{cases} 1 & \text{if } \mathrm{H}\left(\mathbf{C}\right) = 0 \\ 1 - \frac{\mathrm{H}\left(\mathbf{C} \,|\, \mathbf{\Omega}\right)}{\mathrm{H}\left(\mathbf{C}\right)} = \frac{\mathrm{I}\left(\mathbf{\Omega}, \mathbf{C}\right)}{\mathrm{H}\left(\mathbf{C}\right)} & \text{else} \end{cases} \tag{6.33}$$

6.2.8 Completeness

A clustering result satisfies completeness if all the data points which are members of a given class are elements of the same cluster [17], and it is defined as follows:

$$\mathrm{c}(\mathbf{\Omega}, \mathbf{C}) = \begin{cases} 1 & \text{if } \mathrm{H}\left(\mathbf{\Omega}\right) = 0 \\ 1 - \frac{\mathrm{H}\left(\mathbf{\Omega} \,|\, \mathbf{C}\right)}{\mathrm{H}\left(\mathbf{\Omega}\right)} = \frac{\mathrm{I}\left(\mathbf{\Omega}, \mathbf{C}\right)}{\mathrm{H}\left(\mathbf{\Omega}\right)} & \text{else} \end{cases} \tag{6.34}$$

6.2.9 V-measure

V-measure [17] is an entropy-based measure similar to F-measure (see section 6.2.4) which is computed as the (weighted) harmonic mean of distinct homogeneity and completeness score.

V-measure is defined as follows:

$$\mathrm{V}_{\beta}(\mathbf{\Omega}, \mathbf{C}) = \frac{(1+\beta)\,\mathrm{h}\cdot\mathrm{c}}{\beta\,\mathrm{h}+\mathrm{c}} \tag{6.35}$$

$$\tag{6.36}$$

Similarly to F1-measure (equation 6.25) we define V1-measure as a special case of V-measure with $\beta = 1$:

$$\mathrm{V}_{1}(\mathbf{\Omega}, \mathbf{C}) = 2\frac{\mathrm{h}\cdot\mathrm{c}}{\mathrm{h}+\mathrm{c}} \tag{6.37}$$

$$\tag{6.38}$$

6.2.10 Normalized V-measure

V-measure values can be, to a certain extent, cheated using too many small clusters when each observation is put in the separate cluster.

Let us imagine, ten clusters with six observations each. If we assign each observation to its own cluster, we get sixty clusters with V-measure $\mathrm{V}_1(\mathbf{\Omega}, \mathbf{C}) = 0.719903$. That is too high. This problem is the more visible the more clusters we have.

This is the reason why a Normalized V-measure (NV-measure) is introduced. It penalizes difference between the true class labels C and the number of clusters found K and is constructed in such a way giving the V-measure values exactly when $K = C$. NV-measure is defined as follows:

$$\mathrm{NV}_{p,\beta}(\mathbf{\Omega}, \mathbf{C}) = \left(1 - \left(1 - \left(\frac{\min(K, C)}{\max(K, C)}\right)^p\right)^{\frac{1}{p}}\right) \mathrm{V}_\beta(\mathbf{\Omega}, \mathbf{C}) \tag{6.39}$$

The normalization formula is derived from p-norm [9, p. 64]. Hence the parameter p whose influence is depicted in figure 6.2. For the singular example stated above, the NV-measure is $\mathrm{NV}_{1,1}(\mathbf{\Omega}, \mathbf{C}) = 0.119984$.

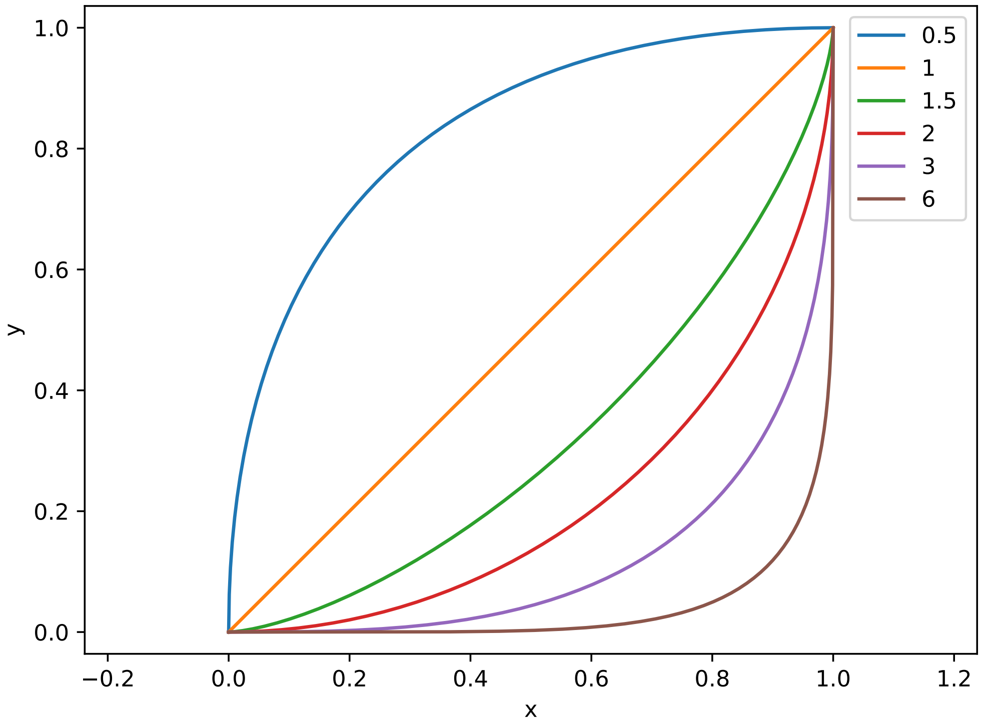

Figure 6.2: Function $y = \left(1 - (1 - x^p)^{\frac{1}{p}}\right)$ for various values p

6.3 Information Criteria

Based on information criteria we can select the best statistical model from a set of candidate models.

Since we are using a GMM with full covariance matrix, the number of degrees of freedom for covariance matrix is $D(D+1)/2$ [14, p. 46] and for mean it is D. Having a K Gaussians we get $k = KD + KD(D+1)/2$ parameters in total.

In this section, the following notation is used:

- M is a statistical model.
- θ is model parameters maximizing the likelihood function.
- $L = \mathrm{p}(\mathbf{X}\,|\theta, M)$ is likelihood.
- N is the number of observations.
- k is the number of parameters.

6.3.1 Akaike Information Criterion

The Akaike information criterion (AIC) [1] estimates the quality of the statistical model relative to other models penalizing the number of parameters needed by the model.

The AIC is given by:

$$\mathrm{AIC}\,(\theta) = 2k - 2\log(L) \tag{6.40}$$

6.3.2 Bayesian Information Criterion

The Bayesian information criterion (BIC) [21] also estimates the quality of the statistical model relative to other models penalizing the number of parameters needed by the model, and it is closely related to AIC.

The BIC assumes that data distribution is in an exponential family[1], and it is given by:

$$\mathrm{BIC}\,(\theta) = \log(N)k - 2\log(L) \tag{6.41}$$

[1] The normal distribution belongs in the exponential family.

7 Incremental News Clustering System

The goal of this thesis is to propose a system which would be capable of maintaining topic clusters of news coming online from a crawler and implement a demonstrator.

In this setting, the corpus is continuously growing; therefore, it can be safely assumed that new topics are going to emerge because it would be very difficult to argue that the number of topics eventually runs up against a finite bound and remains fixed.

7.1 Data Model

The data used for experiments come from a MediaGist[1] analyser which is an online system for cross-lingual analysis of aggregated news and commentaries based on summarization and sentiment analysis technologies. It is designed to assist journalists to detect and explore news topics which are controversially reported or discussed in different countries [18].

The MediaGist analyser contains a crawler[2] whose data can be exported in Extensible Markup Language (XML) format. A structure of the data file is illustrated in figure 7.1.

The root element of the document is element `rss` with nested element `channel` which contains information about a news article, such as language or publisher. The information about a certain news article is in an element `item` which contains a title, link to the article, description, publication date, Globally Unique Identifier (GUID), full text etc. It also contains information about recognized entities and sentiment which is irrelevant for my use case. From the data file, only article title, GUID, language, publication date and full text are used.

[1] MediaGist is available at: `http://mediagist.eu`.

[2] A program which visits websites and reads their pages and other information in order to create entries for a search engine index.

```
<rss xmlns:emm="http://emm.jrc.it" xmlns:iso="http://www.iso.org/3166" version="2.0">
  <channel>
    <title>
      The Guardian
    </title>
    <language>
      en
    </language>
    <guid>
      theguardian.com
    </guid>
    <pubDate>
      2017−03−12 00:57:35 CET
    </pubDate>
    <item emm:id="7222ac5a7799574b1e98622f8671450d">
      <title>
        Two convicted of conspiracy in armed standoff at Oregon wildlife refuge
      </title>
      <link>
        https://www.theguardian.com/us−news/...
      </link>
      <description>
        Verdict handed prosecutors some redemption after they failed to convict ...
      </description>
      <emm:contentType>
        text/html
      </emm:contentType>
      <pubDate>
        2017−03−11 00:01:57 CET
      </pubDate>
      <iso:language>
        en
      </iso:language>
      <guid>
        7222ac5a7799574b1e98622f8671450d
      </guid>
      <emm:entity id="11001303" type="p" count="1" pos="512" name="Ryan␣Bundy" sentiment="−32[failed]">
        Ryan Bundy
      </emm:entity>
      <emm:entity id="1923543" type="p" count="1" pos="2294" name="Ammon␣Bundy" sentiment="0">
        Ammon Bundy
      </emm:entity>
      <emm:entity id="10011843" type="p" count="1" pos="994" name="Ruby␣Ridge" sentiment="0">
        Ruby Ridge
      </emm:entity>
      <emm:entity id="197070" type="o" count="1" pos="3275" name="FBI" sentiment="−35[protection;feared]">
        FBI
      </emm:entity>
      <emm:tonality>
        −2
      </emm:tonality>
      <commentTonality pos="0" neut="0" neg="0">
        0
      </commentTonality>
      <emm:entityRef id="11001303" type="npr" count="1" pos="1732" name="Ryan␣Bundy" sentiment="0">
        Bundy
      </emm:entityRef>
      <emm:entityRef id="11001303" type="npr" count="1" pos="1335" name="Ryan␣Bundy"
          sentiment="33[guilty(not);conspiracy(not)]">
        Ryan
      </emm:entityRef>
      <emm:text wordCount="547">
        Verdict handed prosecutors some redemption after they failed to convict ...
      </emm:text>
    </item>
  </channel>
</rss>
```

Figure 7.1: The structure of an XML data file

7.2 Architecture

The architecture of the system is rather simple because it is a data processing program. In this case, a data flow diagram 7.2 illustrates the system best.

Initially, the raw historical XML data files are preprocessed and saved all in a single file – a corpus – for efficient data processing. Then a model, which is used to represent a variable length document as a fixed-length numeric vector, is trained. Furthermore, the news articles coming from a crawler are clustered. It is crutial to apply the same text preprocessing techniques which were used during model training. It is important to note that each news article can be seen only once. Finally, the clustering methods are evaluated using various metrics.

Object-oriented design described in this section is used to provide functionality encapsulation, code reusability, and interface unification. Fast low-level data structures are used instead of objects because the overhead of the broad object-oriented design would be unfeasible. The Python naming convention is used.

7.2.1 Preprocessor

There are thousands of articles on the input. The text of each article must be preprocessed and for performance reasons, stored in a corpus.

The following text preprocessing techniques are used:

- **Lower casing**
- **Converting to Unicode**
- **Deaccenting** – The accents are removed from the text. Also known as asciifolding.
- **Punctuation removal** – Punctuation characters are replaced with spaces.
- **Multiple whitespaces removal** – Repeating whitespace characters (spaces, tabs, line breaks) are converted to a single space character.
- **Short token removal** – Tokens shorter than 3 are removed.
- **Stop words removal** – The most common words (stop words) are removed.
- **Stemming** – It is a process of reducing inflected words to their word stem. For English language, the Porter stemmer [15] is used.

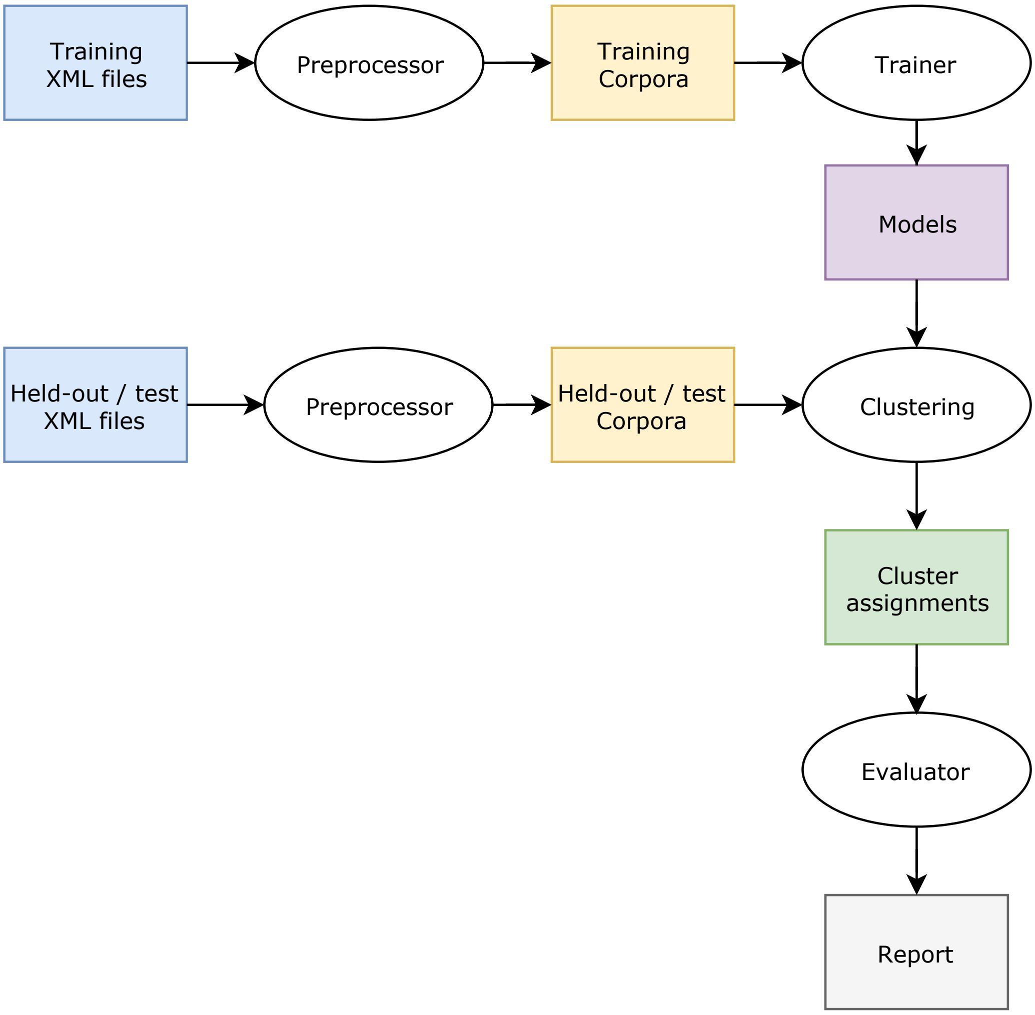

Figure 7.2: Data flow diagram

As a by-product, a dictionary of tokens is created. Infrequent tokens (with less than five occurrences) and frequent tokens (appearing in more than half of the documents) are removed.

In order to support text vectorization methods described in chapter 4, both BoW and list-of-words (LoW) corpora are needed as depicted in figure 7.3.

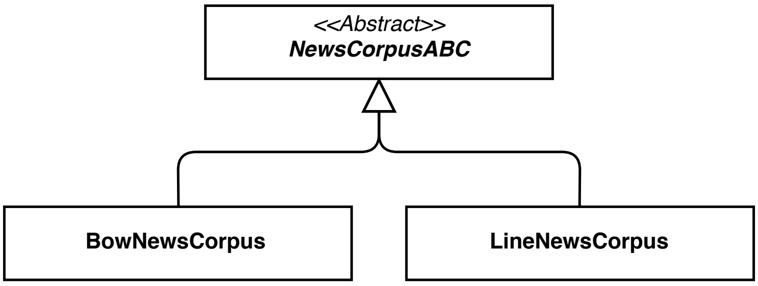

Figure 7.3: Class diagram for BoW/LoW corpora

7.2.2 Models

Representing a variable length document as a fixed-length numeric vector is quite tricky because the text does not have a natural way of doing so. Several methods were proposed in chapter 4.

Figure 7.4 illustrates a class structure which is necessary in order to achieve model interchangeability.

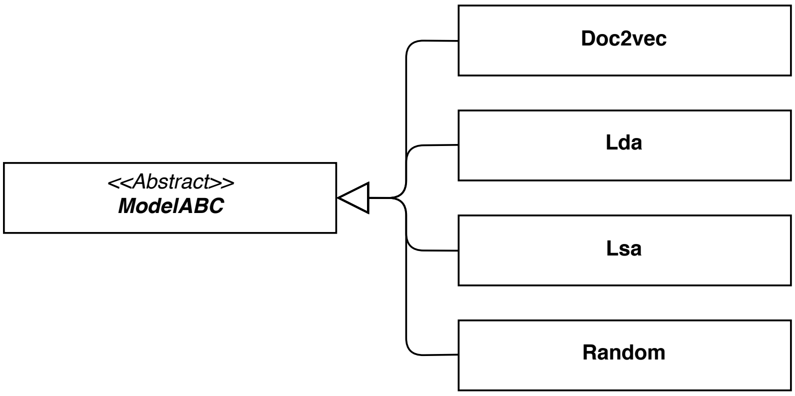

Figure 7.4: Class diagram for text vectorization models

- **doc2vec** is a neural network which produces document embeddings. It uses LoW corpus.

- **Latent Dirichlet Allocation (LDA)** is a generative probabilistic model based on BoW hypothesis.

- **Latent Semantic Analysis (LSA)** is model based on BoW hypothesis.

- **Random** model gives a random vector for arbitrary input and is used as a reference.

7.2.3 Clustering

Figure 7.5 shows a class diagram for clustering algorithms. In order to achieve unified interface, an abstract class `ClusteringABC` is in place. Because all clustering methods proposed in this thesis are going to be implemented using Gibbs sampling but since in the future there might be a different method, a special abstract class for Gibbs sampling based methods is introduced. The `FullGaussianMixture` with its `NormalInverseWishart` prior is used. This design is ready for future extensions to more constrained mixtures with different priors.

Pseudo codes of clustering algorithms and corresponding equations are defined in chapter 5.

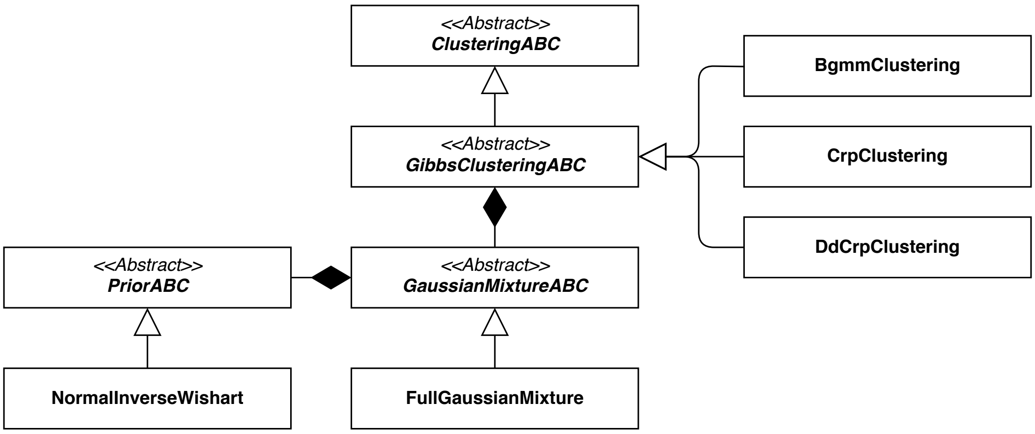

Figure 7.5: Class diagram for clustering algorithms

7.2.4 Evaluator

The clustering methods are evaluated using the ground truth data and evaluation metrics introduced in chapter 6. The ground truth was created from crawled news articles at the 10th week of the year 2017 using the following guidelines:

- Make a list of the news topics of the current week. Consider the number of articles and their internationality. The topic has to be mentioned in at least two languages.
- Use English labels even if the topic was found only for other languages, don't use determiners, rather a keyword style.
- The topic should not be too wide, e.g. Politics or Sports, actually it **should not** be a standard menu category found in news portals (Champions League, Football, Home politics)
- The use case is to display the most important topics of the current week (topics of the week).
- Tips:
  - Event-centred topics – If an important accident, attack, a scandal, an event happened, in general, make a separate topic for it. (e.g. Hurricane Catrina)
  - Entity-centred topics – If there was an entity discussed in a particular context, make a separate topic for it. (Trump and Putin)
- If it is the case that one topic is part of another one, use "other" to distinguish them. E.g. "North Korean nuclear programme" vs "other North Korea"
- The result will finally be hard clustering.

The ground truth used in this thesis is an English subset of multilingual ground truth created for multilingual clustering research purposes.

The data are being clustered and evaluated day after day. The class diagram for evaluator is displayed in figure 7.6. The abstract class `EvaluatorABC` contains common logic such as export to CSV and chart creation. Charts capture progress of evaluation metrics over time. Descendants of the abstract class are responsible for loading the ground truth and assigning the true class labels to documents.

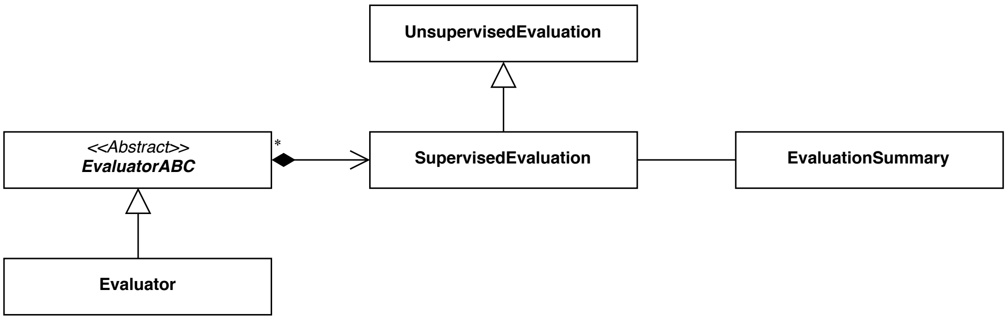

Figure 7.6: Class diagram for evaluator

The `UnsupervisedEvaluation` and `SupervisedEvaluation` classes contain unsupervised and supervised metrics, respectively.

The `EvaluationSummary` class takes evaluations from multiple clustering runs and creates summarization charts.

7.2.5 Visualization

In order to visualize the clustering using existing tool, first, the clustering results have to be saved in a machine-readable format. Graph Exchange XML Format (GEXF)[3] was chosen because it is a language for describing complex network structures, their associated data and dynamics.

Therefore, it can be used to describe cluster assignments changing in time, as shown in figure 7.7. The root element of the document is element `gexf` with nested element `graph` with attributes `defaultedgetype` and `mode` set to `directed` and `dynamic`, respectively, which indicates that it is a dynamic directed graph.

Then there are definitions of node attributes in elements `attributes`. A title is a static attribute. Its value does not change over time. A node's cluster assignment, on the other hand, is a dynamically changing attribute.

In element `nodes`, there are child elements `node` which are representations of documents. `node`'s attribute `id` is a document's GUID, the attributes `start` and `end` describe a time interval during which a document was clustered. Attribute `label` is useful during visualization and its value is equal to the first seven characters of GUID.

Each node contains element `viz:position` whose attributes `x`, `y` and `z` position the node during visualization. These coordinates are given by incremental PCA transformation of document's vector representation. In element `attvalues`, there are elements `attvalue` which define the values of

[3] `https://gephi.org/gexf/format/`

```
<?xml version=’1.0’ encoding=’utf−8’?>
<gexf xmlns:viz="http://www.gexf.net/1.2draft/viz" version="1.2" xmlns="http://www.gexf.net/1.2draft"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema−instance"
      xsi:schemaLocation="http://www.w3.org/2001/XMLSchema−instance">
  <graph defaultedgetype="directed" mode="dynamic" name="">
    <attributes class="node" mode="static">
      <attribute id="1" title="title" type="string" />
    </attributes>
    <attributes class="node" mode="dynamic">
      <attribute id="0" title="cluster" type="long" />
    </attributes>
    <nodes>
      <node end="4" id="67ccf777b7719e99eaf4cc9de521d250" label="67ccf77" start="2">
        <viz:position x="0.7456457461559365" y="0.5072068584528289" z="0" />
        <attvalues>
          <attvalue end="3" for="0" start="2" value="10" />
          <attvalue end="4" for="0" start="3" value="0" />
          <attvalue for="1" value="Lewis␣Hamilton␣questions␣F1’s␣new␣regulations␣before␣first␣test␣session" />
        </attvalues>
      </node>
      <node end="4" id="c81c03a67f011097b8276be5b571a352" label="c81c03a" start="2">
        <viz:position x="0.8512782228548975" y="0.18808625419599057" z="0" />
        <attvalues>
          <attvalue for="1" value="Lewis␣Hamilton␣not␣missing␣Rosberg␣as␣he␣looks␣ahead␣to␣new␣F1␣season" />
          <attvalue end="3" for="0" start="2" value="10" />
          <attvalue end="4" for="0" start="3" value="0" />
        </attvalues>
      </node>
      ...
    </nodes>
    <edges>
      <edge id="0" source="67ccf777b7719e99eaf4cc9de521d250" target="c81c03a67f011097b8276be5b571a352">
        <spells>
          <spell end="3" start="2" />
        </spells>
      </edge>
      ...
    </edges>
  </graph>
</gexf>
```

Figure 7.7: The structure of a GEXF data file

static or dynamic node attributes which were defined at the beginning of the file. The attributes `start` and `end` describe a time interval during the node attribute given by attribute `for` is equal to the value given by attribute `value`.

Nodes might be connected by an edge which represents the seating assignment in ddCRP. In element `edges`, there are nested elements `edge` with attributes `source` and `target` containing node IDs. Edges are also dynamic which is specified using `spell` attributes.

As a visualization tool, Gephi[4] was chosen [2]. It is a free and open-source tool for data analysis. It allows the user to explore and understand graphs by manipulating the structures, shapes, and colours. Document data, such as the title, can be viewed by clicking on a node.

7.3 Implementation

During the architectural design, it has been identified that a Command Line Interface (CLI) data processing program is going to be implemented because no Graphical User Interface (GUI) is needed. There are several languages or tools commonly used for scientific computations such as Matlab, Octave, R, Julia, and Python.

- Matlab[5] is a commercial numerical computing environment and programming language. There are many toolkits providing extra functionality which make it rather expensive.
- Octave[6] is an open-source and multi-platform alternative to the Matlab. Its language is largely compatible with Matlab; however, not fully.
- R[7] is an open-source and multi-platform software environment for statistical computing and graphics.
- Julia[8] is relatively new, high-level, high-performance dynamic programming language for numerical computing. It provides a sophisticated compiler, distributed parallel execution, numerical accuracy, and an extensive mathematical function library.

[4] `https://gephi.org/`

[5] `https://www.mathworks.com/products/matlab.html`

[6] `https://www.gnu.org/software/octave/`

[7] `https://www.r-project.org/`

[8] `https://julialang.org/`

- Python[9] is a high-level general purpose programming language. There are many mature and free packages for scientific computing.

Because the goal of this thesis is not to strive for a fast and heavily optimized code, other general purpose programming languages such as C or C++ were excluded. The speed and ease of the development was prefered during prototyping.

After a thorough research the Python language was chosen because it has surpassing packages for scientific computing such as gensim[10] [16], NumPy[11], SciPy[12], etc. Automated tests[13] were automatically run in Continuous integration (CI)[14] pipeline.

Having the data flow figure 7.2 in mind, the problem was split into several scripts because some parts of the data flow are one-off calculations.

7.3.1 Dataset Transformation

The data exported from MediaGist analyzer in XML format contains articles in all languages and file name for each article is not equal to its GUID. Hence a script was written. It reads the publication date, language, and GUID, and prepends it to the file name, because parsing an XML is a rather costly operation.

It simplifies filtering and sorting which can be done just by looking at articles' file name.

7.3.2 Corpora Creation

The second script was used for preprocessing the news articles and for the creation of BoW/LoW corpora and dictionary, as described in section 7.2.1. The implementation of preprocessing and corpora is heavily based on the gensim library.

Three files are created – BoW corpus, LoW corpus, and dictionary – which are later used by other scripts.

[9] https://www.python.org/

[10] https://radimrehurek.com/gensim/

[11] http://www.numpy.org/

[12] https://www.scipy.org/

[13] https://pytest.org/

[14] https://travis-ci.org/

7.3.3 Model Training

Another script is responsible for training text vectorization models implemented according to the analysis made in section 7.2.2. The script creates a pool of processes which are responsible for parallel model training with various vector dimensions.

It postulates BoW corpus, LoW corpus, and dictionary created by previous script. For each model and for each dimension it trains and saves the model. The models from the gensim library are used.

7.3.4 Clustering

In contrast to previous scripts, the clustering script is not a one-off calculation but it is run multiple times with various parameters in order to achieve the best results possible on held-out and test data. Figure 7.8 illustrates how the script works.

Initially, the raw held-out/test data are preprocessed and stored in a temporary corpus file. The documents grouped by day are then transformed to its vector representation by a pre-trained model and passed to a selected clustering algorithm. Several optimization techniques are in place in order not to perform the same calculation all over again during clustering. Resulting cluster assignments are evaluated and stored in files.

Even though the available data can easily fit in memory and the data could be processed multiple times, each document can be "seen" only once because the program acts as if the data were coming directly from a crawler. An integration with existing crawler was not implemented because it would bring unnecessary complexity and it would not change the results.

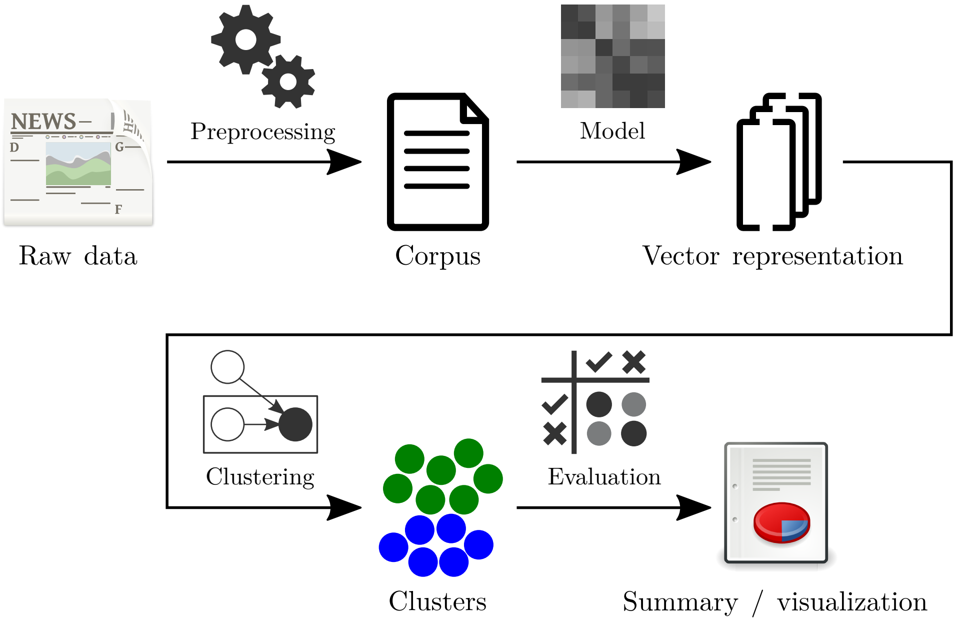

Figure 7.8: Illustration of the clustering script behaviour

8 Results and Discussion

The goal of experiments described in this chapter is to compare implemented model-based clustering methods. The vector representation for news articles was trained with dimension $D = 50$ and $D = 100$ using training dataset consisting of 186 931 English news articles. The held-out and test data consists of 197 news articles published in 4 days and 97 news articles published in the next 3 days, respectively.

For each IGMM clustering method, the best hyperparameters were chosen on held-out data. Then an experiment was run five times on held-out + test data (7 days of data in total) in day after day manner. Then a median of evaluation metrics was taken and a summary was generated.

Tables 8.1 and 8.2 show the most telling evaluation metrics for each experiment. For full result tables see appendix D.

The ddCRP performed much better than the CRP. It converged in fewer iterations and it was easier to find applicable hyperparameters. On the other hand, the ddCRP is computationally more costly.

Because the incremental clustering system was implemented, a weakly informative data-dependent prior presented in [14, p. 133] could not be used. In the beginning, we have no observations to make assumptions about. Therefore, a diagonal covariance prior was used. Further, the ddCRP used temporal information (a difference of publication dates) as a prior in form of distance matrix D.

Also, a choice of the vector representation of news articles had a huge impact. Experiments show that doc2vec outperformed LSA and LDA in most of the evaluation metrics.

Unfortunately, there are many criteria measuring the quality of clustering and none of them can be used universally. The purity, F-measure, V-measure, and NV-measure were selected because they evinced biggest changes depend-

Table 8.1: A brief result table for $D = 50$ on test data

		CRP			ddCRP		
	random	LSA	LDA	doc2vec	LSA	LDA	doc2vec
purity $(\mathbf{\Omega}, \mathbf{C})$	0.2415	0.3878	0.4864	**0.5816**	0.5340	0.6803	**0.8061**
$\mathrm{F}_1(\mathbf{\Omega}, \mathbf{C})$	0.0238	0.0695	**0.2535**	0.2327	0.2479	0.4937	**0.5002**
$\mathrm{V}_1(\mathbf{\Omega}, \mathbf{C})$	0.4929	0.5812	0.6484	**0.6964**	0.6757	0.7860	**0.8506**
$\mathrm{NV}_{1,1}(\mathbf{\Omega}, \mathbf{C})$	0.4378	0.4403	0.5187	**0.5394**	0.6740	0.7081	**0.7445**

Table 8.2: A brief result table for $D = 100$ on test data

		CRP			ddCRP		
	random	LSA	LDA	doc2vec	LSA	LDA	doc2vec
purity $(\mathbf{\Omega}, \mathbf{C})$	0.2415	0.3673	**0.5408**	0.5000	0.5272	0.6565	**0.7653**
$\mathrm{F}_1(\mathbf{\Omega}, \mathbf{C})$	0.0238	0.0752	0.2506	**0.2677**	0.2315	0.4549	**0.5316**
$\mathrm{V}_1(\mathbf{\Omega}, \mathbf{C})$	0.4929	0.5540	**0.6734**	0.6509	0.6764	0.7938	**0.8512**
$\mathrm{NV}_{1,1}(\mathbf{\Omega}, \mathbf{C})$	0.4378	0.4858	0.5576	**0.5678**	0.6579	0.6870	**0.8072**

ing on the character of clustering. A multicriteria optimization of hyperparameters was performed in order to achieve the best values as possible with a reasonable number of clusters. Many criteria, such as Rand index or V-measure, are sensitive to the number of clusters. For that reason, NV-measure was introduced.

The small number of annotated data is a major drawback of performed experiments. Also, the timespan of articles is too short for proper testing of perquisites of ddCRP. Data annotation is a tedious and ambiguous process and it is one of the biggest challenges of cluster analysis. Creation of larger corpus is an ongoing effort of the NLP research group.

9 Conclusion

The goal was to research model-based clustering methods, notably the Distance Dependent Chinese Restaurant Process (ddCRP), propose an incremental clustering system which would be capable of maintaining growing number of topic clusters of news articles coming online from a crawler, implement proof-of-concept application, and evaluate formed clusters.

Initially, the necessary mathematics and statistics were defined. Then, the problematics of the vector representation of text, namely LSA, LDA and doc2vec, was discussed. Both finite and infinite model-based clustering methods were covered. The focus was on the most common variant of Latent Variable Model (LVM) – the Gaussian mixture model (GMM). The most difficult part of cluster analysis is the evaluation of clustering. Various measures were discussed and a modification of V-measure – NV-measure – was introduced in order to penalize an excessive or insufficient number of clusters.

The practical part focused on the architecture and implementation of the incremental clustering system. The system is able to process news articles coming in batches and assign them to (newly created) clusters. The program acts as if the data were coming directly from a crawler. Therefore, each document can be "seen" only once. An integration with existing crawler was not implemented because it would bring unnecessary complexity and it would not change the results.

The ddCRP performed much better than the CRP. It converged in fewer iterations and it was easier to find applicable hyperparameters. On the other hand, the ddCRP is computationally more costly.

A choice of the vector representation of news articles had a huge impact. Experiments show that doc2vec outperformed LSA and LDA in most of the evaluation metrics.

9.1 Future Work

The proof-of-concept of monolingual incremental clustering system was implemented. A future research might consist of:

- carrying out experiments in other languages
- application of cross-lingual approach (documents describing the same topic written in multiple languages belong to the same cluster)

- experimenting with other distance priors in ddCRP
- constraining covariance matrices (e.q. diagonal, spherical, tied)
- removing obsolete data from clusters
- more advanced optimizations using Cholesky decomposition [20]
- compilation to C using Cython[1]
- integration inside the demo application (e.g. `http://mediagist.eu`)
- using deep auto-encoder for high-dimensional data reduction [7]
- using a mixture of auto-encoders for clustering [22]

[1] `http://cython.org/`

Bibliography

[1] H. Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6):716–723, Dec 1974. ISSN 0018-9286. doi: 10.1109/TAC.1974.1100705.

[2] Mathieu Bastian, Sebastien Heymann, and Mathieu Jacomy. Gephi: An open source software for exploring and manipulating networks, 2009. URL `http://www.aaai.org/ocs/index.php/ICWSM/09/paper/view/154`.

[3] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics).* Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006. ISBN 0387310738.

[4] David M. Blei and Peter I. Frazier. Distance dependent chinese restaurant processes. *Journal of Machine Learning Research*, 12:2461–2488, 2011. URL `http://dl.acm.org/citation.cfm?id=2078184`.

[5] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, March 2003. ISSN 1532-4435. URL `http://dl.acm.org/citation.cfm?id=944919.944937`.

[6] Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. Indexing by latent semantic analysis. *JOURNAL OF THE AMERICAN SOCIETY FOR INFORMATION SCIENCE*, 41(6):391–407, 1990.

[7] G E Hinton and R R Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, July 2006. doi: 10.1126/science.1127647. URL `https://www.cs.toronto.edu/~hinton/science.pdf`.

[8] John F. Kenney. *Mathematics of statistics.* Van Nostrand, 1966.

[9] E. Kreyszig. *Introductory Functional Analysis With Applications.* 1978. ISBN 0471507318.

[10] Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, ICML'14, pages II–1188–II–1196. JMLR.org, 2014. URL `http://dl.acm.org/citation.cfm?id=3044805.3045025`.

[11] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval.* Cambridge University Press, New York, NY, USA, 2008. ISBN 0521865719, 9780521865715.

[12] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, pages 3111–3119, USA, 2013. Curran Associates Inc. URL `http://dl.acm.org/citation.cfm?id=2999792.2999959`.

[13] R. D. Morris, X. Descombes, and J. Zerubia. The ising/potts model is not well suited to segmentation tasks. In *1996 IEEE Digital Signal Processing Workshop Proceedings*, pages 263–266, Sep 1996. doi: 10.1109/DSPWS.1996.555511.

[14] Kevin P. Murphy. *Machine Learning: A Probabilistic Perspective.* The MIT Press, 2012. ISBN 0262018020, 9780262018029.

[15] M.F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980. doi: 10.1108/eb046814.

[16] Radim Řehůřek and Petr Sojka. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, May 2010. ELRA. `http://is.muni.cz/publication/884893/en`.

[17] Andrew Rosenberg and Julia Hirschberg. V-measure: A conditional entropy-based external cluster evaluation measure. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning(EMNLP-CoNLL)*, pages 410–420, 2007. `http://www.aclweb.org/anthology/D07-1043`.

[18] Josef Steinberger. Mediagist: A cross-lingual analyser of aggregated news and commentaries. *Proceedings of ACL-2016 System Demonstrations*, 2016. doi: 10.18653/v1/p16-4025. URL `http://www.aclweb.org/anthology/P16-4025`.

[19] Alexander Strehl and Joydeep Ghosh. Cluster ensembles — a knowledge reuse framework for combining multiple partitions. *J. Mach. Learn. Res.*, 3: 583–617, March 2003. ISSN 1532-4435. doi: 10.1162/153244303321897735. URL `https://doi.org/10.1162/153244303321897735`.

[20] Y. W. Teh. Exponential families: Gaussian, gaussian-gamma, gaussian-wishart, multinomial. [online], Last visited 20.04.2018. URL

https://www.stats.ox.ac.uk/~teh/research/notes/GaussianInverseWishart.pdf.

[21] E Wit, Edwin Van den Heuvel, and Jan-Willem Romeijn. 'all models are wrong.': An introduction to model uncertainty. 66, 08 2012.

[22] Dejiao Zhang, Yifan Sun, Brian Eriksson, and Laura Balzano. Deep unsupervised clustering using mixture of autoencoders. *CoRR*, abs/1712.07788, 2017. URL http://arxiv.org/abs/1712.07788.

A Notation

General math notation

Symbol	Meaning
$\mathbb{I}(x)$	Indicator function, $\int(x) = 1$ if x is true, else $\int(x) = 0$
∞	Infinity
$\rightarrow$	Tends towards, e.g., $n \rightarrow \infty$
$\propto$	Proportional to, so $y = ax$ can be written as $y \propto x$
$\|x\|$	Absolute value
$n!$	Factorial function
$\mathbb{Z}$	Integer
$\mathbb{R}$	Real number
$1 : n$	Range (Matlab convention): $1 : n = \{1, 2, \ldots, n\}$
$\approx$	Approximately equal to
$\arg\max_x f(x)$	Argmax: the value x which maximizes f
$\binom{n}{k}$	n choose k
$\exp(x)$	Exponential function e^x

Linear algebra notation

We use boldface lowercase to denote vectors, such as $\mathbf{a}$, and boldface uppercase to denote matrices, such as $\mathbf{A}$.

Symbol	Meaning
$\mathbf{A} \succ 0$	$\mathbf{A}$ is a positive definite matrix
$\mathrm{tr}(\mathbf{A})$	Trace of a matrix
$\|\boldsymbol{A}\|$	Determinant of matrix $\mathbf{A}$
$\mathbf{A}^{-1}$	Inverse of a matrix
$\mathbf{A}^T$	Transpose of a matrix
$\mathbf{a}^T$	Transpose of a vector
$\mathbf{1}$ or $\mathbf{1}_d$	Vector of ones (of length d)

Probability notation

Symbol	Meaning
$X \sim p$	X is distributed according to distribution p
$\mathrm{cov}\left[\mathbf{x}\right]$	Covariance of $\mathbf{x}$
$\mathbb{E}\left[X\right]$	Expected value of X
$\mathrm{H}\left(X\right)$	Entropy of distribution $\mathrm{p}(X)$
$\mathrm{I}\left(X, Y\right)$	Mutual information between X and Y
$\boldsymbol{\Lambda}$	Precision matrix $\boldsymbol{\Lambda} = \boldsymbol{\Sigma}^{-1}$
$\boldsymbol{\mu}$	Mean of a multivariate distribution
$\mathrm{p}(x)$	Probability density or mass function
$\mathrm{p}(x\|y)$	Conditional probability density of x given y
$\boldsymbol{\Sigma}$	Covariance matrix
ν	Degrees of freedom parameter
Z	Normalization constant of a probability distribution

B Acronyms

AIC	Akaike information criterion
BGMM	Bayesian Gaussian Mixture Model
BIC	Bayesian information criterion
BoW	bag-of-words
CBOW	Continuous Bag-of-Words
CI	Continuous integration
CLI	Command Line Interface
CRP	Chinese Restaurant Process
CSV	Comma-separated values
ddCRP	Distance Dependent Chinese Restaurant Process
EM	expectation–maximization
FGMM	Finite Gaussian Mixture Model
GEXF	Graph Exchange XML Format
GMM	Gaussian mixture model
GUI	Graphical User Interface
GUID	Globally Unique Identifier
ID	Identifier
IGMM	Infinite Gaussian Mixture Model
LDA	Latent Dirichlet Allocation
LoW	list-of-words
LSA	Latent Semantic Analysis
LSI	Latent Semantic Indexing
LVM	Latent Variable Model

MAP	Maximum a posteriori estimation
MCMC	Markov Chain Monte Carlo
MLE	Maximum likelihood estimation
MVN	Multivariate normal distribution
NIW	Normal-inverse-Wishart distribution
NLP	Natural language processing
PCA	Principal Component Analysis
PV-DBOW	Distributed Bag of Words version of Paragraph Vector
PV-DM	Distributed Memory version of Paragraph Vector
SVD	Singular Value Decomposition
VSM	Vector Space Model
XML	Extensible Markup Language

C User Manual

The demonstrator was implemented using the Python programming language and its tools. Following text assumes GNU/Linux (specifically Debian derivatives such as Ubuntu) operating system. On other operating systems, it is analogous.

Prerequisites

In order to run the application, Python >=3.5 and the packaging system pipenv is needed. It can be installed using the following commands:

```
$ sudo apt-get install python3 python3-tk python3-pip
$ pip3 install pipenv
```

Installation

With Python and pipenv in place, dependencies are installed automatically using command:

```
$ pipenv install --dev
```

Then add the following line to `~/.bashrc` file:

```
export PYTHONPATH='.'
```

Usage

The application consists of multiple scripts[1]. Python scripts can be run using the following command from project root directory:

```
$ ./run.sh <path-to-python-script>
```

Notable scripts are:

- `data/genuine/training/create_training_corpora.py` – Create a dictionary, BoW and LoW corpora from training data files.

[1] The script's arguments are described in a script command help

- `data/genuine/training/train_models.py` – Train the LSA, LDA, doc2vec models for document vector representation.

- `clustering_system/main.py` – Cluster held-out/test documents and evaluate results. The clustering artefacts appear in a `temp` folder.

- `clustering_system/summary.py` – Summarize evaluations from multiple runs of clustering.

- `runner.sh` – Run and evaluate clustering experiments with ease.

- `test.sh` – Run automated tests.

D Experiments

The vector representation for news articles was trained with dimension $D = 50$ and $D = 100$ using training dataset consisting of 186 931 news articles. The held-out and test data consists of 197 news articles published in 4 days and 97 news articles published in next 3 days, respectively.

For each IGMM clustering method, the best hyperparameters were chosen on held-out data. Then an experiment was run five times on held-out + test data (7 days of data in total) in day after day manner. Then a median of evaluation metrics was taken and a summary was generated.

Table D.1: Result table for $D = 50$ on test data

		CRP			ddCRP		
	random	LSA	LDA	doc2vec	LSA	LDA	doc2vec
N	294	294	294	294	294	294	294
$K = \|\mathbf{\Omega}\|$	51	66	40	63	50	56	57
$C = \|\mathbf{C}\|$	50	50	50	50	50	50	50
purity $(\mathbf{\Omega}, \mathbf{C})$	0.2415	0.3878	0.4864	0.5816	0.5340	0.6803	0.8061
RI $(\mathbf{\Omega}, \mathbf{C})$	0.9427	0.9470	0.9454	0.9568	0.9443	0.9536	0.9691
H $(\mathbf{\Omega})$	3.8537	4.0358	3.4656	4.0041	3.5257	3.3781	3.8015
H $(\mathbf{C})$	3.5127	3.5127	3.5127	3.5127	3.5127	3.5127	3.5127
P $(\mathbf{\Omega}, \mathbf{C})$	0.0365	0.1125	0.2812	0.3832	0.2597	0.4336	0.6969
R $(\mathbf{\Omega}, \mathbf{C})$	0.0176	0.0487	0.2519	0.1703	0.2325	0.6301	0.3840
$\mathrm{F}_1(\mathbf{\Omega}, \mathbf{C})$	0.0238	0.0695	0.2535	0.2327	0.2479	0.4937	0.5002
$\mathrm{h}(\mathbf{\Omega}, \mathbf{C})$	0.5168	0.6245	0.6441	0.7504	0.6752	0.7703	0.8856
$\mathrm{c}(\mathbf{\Omega}, \mathbf{C})$	0.4689	0.5435	0.6519	0.6519	0.6789	0.8172	0.8183
$\mathrm{V}_1(\mathbf{\Omega}, \mathbf{C})$	0.4929	0.5812	0.6484	0.6964	0.6757	0.7860	0.8506
$\mathrm{NV}_{1,1}(\mathbf{\Omega}, \mathbf{C})$	0.4378	0.4403	0.5187	0.5394	0.6740	0.7081	0.7445
I $(\mathbf{\Omega}, \mathbf{C})$	1.8155	2.1936	2.2624	2.6360	2.3717	2.7060	3.1108
NMI $(\mathbf{\Omega}, \mathbf{C})$	0.4934	0.5826	0.6484	0.6982	0.6757	0.7861	0.8513
AIC (θ)	-	87426	-76123	139814	53112	15017	141155
BIC (θ)	-	409795	118057	447529	297330	288542	419563

Table D.2: Result table for $D = 100$ on test data

		CRP			ddCRP		
	random	LSA	LDA	doc2vec	LSA	LDA	doc2vec
N	294	294	294	294	294	294	294
$K = \lvert \boldsymbol{\Omega} \rvert$	41	54	60	44	50	57	53
$C = \lvert \mathbf{C} \rvert$	50	50	50	50	50	50	50
purity $(\boldsymbol{\Omega}, \mathbf{C})$	0.2415	0.3673	0.5408	0.5000	0.5272	0.6565	0.7653
RI $(\boldsymbol{\Omega}, \mathbf{C})$	0.9427	0.9381	0.9524	0.9511	0.9477	0.9357	0.9687
H $(\boldsymbol{\Omega})$	3.8537	3.7646	3.7833	3.5478	3.5981	3.1233	3.6558
H $(\mathbf{C})$	3.5127	3.5127	3.5127	3.5127	3.5127	3.5127	3.5127
P $(\boldsymbol{\Omega}, \mathbf{C})$	0.0365	0.0889	0.3340	0.3586	0.2755	0.3389	0.6488
R $(\boldsymbol{\Omega}, \mathbf{C})$	0.0176	0.0693	0.2049	0.2719	0.2043	0.7087	0.4392
$\mathrm{F}_1(\boldsymbol{\Omega}, \mathbf{C})$	0.0238	0.0752	0.2506	0.2677	0.2315	0.4549	0.5316
$\mathrm{h}(\boldsymbol{\Omega}, \mathbf{C})$	0.5168	0.5760	0.6994	0.6547	0.6831	0.7545	0.8626
$\mathrm{c}(\boldsymbol{\Omega}, \mathbf{C})$	0.4689	0.5337	0.6494	0.6441	0.6699	0.8407	0.8367
$\mathrm{V}_1(\boldsymbol{\Omega}, \mathbf{C})$	0.4929	0.5540	0.6734	0.6509	0.6764	0.7938	0.8512
$\mathrm{NV}_{1,1}(\boldsymbol{\Omega}, \mathbf{C})$	0.4378	0.4858	0.5576	0.5678	0.6579	0.6870	0.8072
I $(\boldsymbol{\Omega}, \mathbf{C})$	1.8155	2.0232	2.4567	2.2997	2.3995	2.6505	3.0300
NMI $(\boldsymbol{\Omega}, \mathbf{C})$	0.4934	0.5544	0.6739	0.6509	0.6764	0.7949	0.8513
AIC (θ)	-	377836	309205	349203	334371	292320	501732
BIC (θ)	-	1402435	1447649	1184061	1283073	1373841	1507356