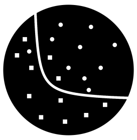
PROGETTO PER SISTEMI INTELLIGENTI

Prof. Nunzio Alberto Borghese
6 CFU

Luca Cappelletti

Project Documentation
Year 2017/2018



Magistrale Informatica
Università di Milano
Italy
18 settembre 2018

Indice

1 Introduction

The Zipf law is an empirical law that refers to the fact that in nature, many data can be approximated by a Zipfian distribution, for example texts, some images[1], even sounds in spoken languages[2]. It is therefore of interest to identify ways to exploit this relatively simple way to convert documents into representative vectors in problems such as classifications.

[1] https://www.dcs.warwick.ac.uk/bmvc2007/proceedings/CD-ROM/papers/paper-288.pdf

[2] https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0033993

2 The Zipf

The zipf function z converts any document $d \subseteq \mathbb{D}^m$ comprised of elements into a representative vector $\underline{\boldsymbol{v}} \subseteq [0,1]^n$, $n \leqslant m$ based on the frequency of said elements in the given document. Taken in consideration the set of elements $d \subseteq \mathbb{D}$ that comprise the document and $d_{\neq} \subseteq d$ the set of distinct elements of the set d, for any given element $d_{\neq_i} \in d_{\neq}$, the value assigned by the zipf function is:

$$z(d_{\neq_i}) : \mathbb{D} \to \mathbb{R} = \frac{\#\left\{\forall d_i \in d : d_{\neq_i} = d_i\right\}}{\#d}$$

3
The classifier

3.1 Training the classifier

Given a set of training class-labeled elements T, we proceed as follows:

1. Convert every document $d \in T$ into its representative vector $\underline{\boldsymbol{v}}$.
2. Execute, for each class C_j of vectors, a **PCA reduction** from the initial vector size (sometimes up to thousands) to a few decades.
3. Using the reduced classes, iterate for each class C_{r_j} the **KMeans** algorithm incrementing the number of the clusters Q_i k until the value of the mean density of points $\overline{\rho}$ increases, with $\overline{\rho}$ being defined as:

$$\overline{\rho}_{jk} = \frac{1}{k}\sum_{i=1}^{k}\rho_{jk_i} = \frac{1}{k}\sum_{i=1}^{k}\left(\frac{\#\left\{\underline{\boldsymbol{v}} \in C_{r_j} : \underline{\boldsymbol{v}} \in Q_i\right\}}{\#C_{r_j}}\right)^k \cdot \frac{1}{r_{Q_i}^2} \qquad r_{Q_i}^2 = \frac{1}{n}\sum_{h=1}^{n}(\underline{\boldsymbol{c}}_i - \underline{\boldsymbol{p}}_h)^2$$

   Where r_{Q_i} is the approximated radius of the cluster Q_i, using the farthest n frontier points p_f. This gives an approximate number k of centroids that describe the given class.
4. For every class C_j, given a percentage of points p, we choose a number $r = p \cdot \#C_j$ of representative vectors, distributed in weighted fashion thorough the class clusters determined at the point3. To this set of points, we add also the clusters centroids.
5. We move every point $\underline{\boldsymbol{p}}$ of every class C_j towards their centroid $\underline{\boldsymbol{c}}_i$ of a constant percentage α along the segment $\overline{PC_i}$.

3.2 Classifying a document

To classify a given a document d we proceed as follows:

1. Convert the document d to a zipf representative vector: $\underline{\boldsymbol{v}} = z(d)$.
2. The document is classified as the closest representative point in the classifier model.