Text Mining

Emanuele Guidotti

2019/2020

Vector

A **vector** is an ordered finite list of numbers. Its entries are called **elements** of the vector. The **dimension** of the vector is the number of elements it contains.

Denoting an n-dimensional vector using the symbol $\boldsymbol{a}$, the i-th element of the vector $\boldsymbol{a}$ is denoted with a_i, where the subscript i is an integer index that runs from 1 to n.

A vector is said to be **sparse** if many of its elements are zero, i.e. if $a_i = 0$ for many i.

Vector Space

A **vector space** is a collection of vectors, which may be added together and multiplied by numbers, called scalars.

Two vectors of the **same size** can be added together by adding the corresponding elements:

$$\boldsymbol{a} + \boldsymbol{b} = (a_1 + b_1, ..., a_n + b_n)$$

A vector can be multiplied by a scalar, k, by multiplying every element of the vector by the scalar.

$$k \cdot \boldsymbol{a} = (ka_1, ..., ka_n)$$

The vector space can be extended with additional structures.

- **Inner product** of two n-vectors:

$$\boldsymbol{a} \cdot \boldsymbol{b} = a_1 b_1 + ... + a_n b_n$$

- **Euclidean norm** of a n-vector:

$$||\boldsymbol{a}|| = \sqrt{a_1^2 + ... + a_n^2}$$

- **Euclidean distance** between two n-vectors:

$$dist(\boldsymbol{a}, \boldsymbol{b}) = ||\boldsymbol{a} - \boldsymbol{b}||$$

- **Angle** between two n-vectors:

$$\theta = arccos\Big(\frac{\boldsymbol{a} \cdot \boldsymbol{b}}{||\boldsymbol{a}||||\boldsymbol{b}||}\Big)$$

Examples

Location: 3-vector is used to represent a location or position of some point in 3-dimensional (3-D) space. The elements of the vector give the coordinates (x, y, z).

Color: A 3-vector can represent a color, with its entries giving the Red, Green, and Blue (RGB) intensity.

Portfolio: An n-vector can represent a stock portfolio or investment in n different assets.

Time Series: An n-vector can represent a time series or signal, that is, the value of some quantity at different times.

Images: A black and white image can be represented by a vector of length $m \times n$, with the elements giving grayscale levels at the pixel locations, typically ordered column-wise (n) or row-wise (m).

Features: An n-vector can collect together n different quantities that pertain to a single object (e.g. age, height, weight, blood pressure, temperature, gender).

Text Data Vectorization

Processing natural language text and extract useful information requires the text to be converted into a set of numerical features.

Word Embeddings or Word Vectorization is a methodology in NLP to map text to a corresponding vector of real numbers which can be used to support later automated text mining algorithms.

The process of converting text into numbers is called **Vectorization**.

Vector Space Model

Definitions

Terms are generic features that can be extracted from text documents. Typically terms are single words, keywords, n-grams, or longer phrases.

Documents are represented as vectors of terms. Each dimension corresponds to a separate term. If a term occurs in the document, its value in the vector is non-zero. Several different ways of computing these values, also known as (term) **weights**, have been developed.

$$\boldsymbol{d} = (w_1, ..., w_n)$$

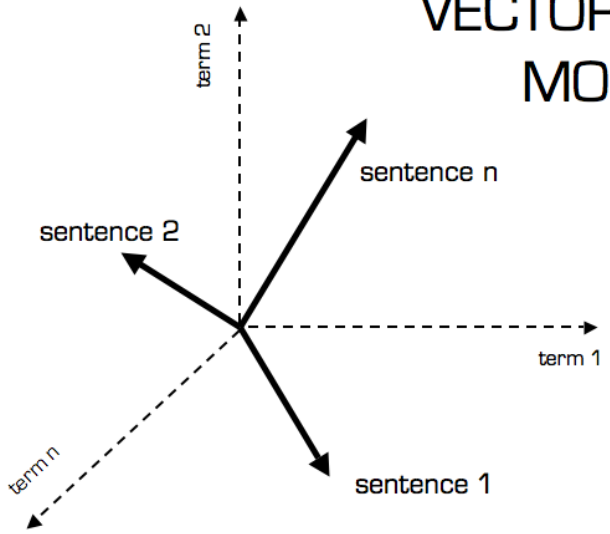
The **Corpus** represents a collection of documents (the dataset). It is represented as a vector of documents, i.e. a matrix of terms.

$$\mathbf{C} = \begin{pmatrix} \boldsymbol{d}_1 \\ \boldsymbol{d}_2 \\ \vdots \\ \boldsymbol{d}_m \end{pmatrix} = \begin{pmatrix} w_{1,1} & w_{1,2} & \cdots & w_{1,n} \\ w_{2,1} & w_{2,2} & \cdots & w_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ w_{m,1} & w_{m,2} & \cdots & w_{m,n} \end{pmatrix}$$

Each element $C_{d,t} = w_{d,t}$ represents the weight of the t-th term in the d-th document.

The **Vocabulary** is the set of all unique terms in the corpus.

VECTOR SPACE MODEL



Remarks

- The vocabulary corresponds to the canonical base of the vector space.
- The dimension of the space, n, is the number of the elements in the vocabulary.
- Each document vector has exactly n elements, one for each term in the vocabulary. If a term does not occur in the document, its value in the vector is zero.
- Vector operations can be used to compare documents.

Bag of Words (BOW)

With Bag of Words (BOW), we refer to a Vector Space Model where:

- Terms: words (more generally we may use n-grams, etc.)
- Weights: number of occurences of the terms in the document.

```
from sklearn.feature_extraction.text import CountVectorizer
vectorizer = CountVectorizer(analyzer = "word", ngram_range = (1,1))

# Learn the vocabulary dictionary and return term-document matrix.
vectorizer.fit_transform(corpus)
```

CountVectorizer Documentation

TF-IDF

With TF-IDF (Term Frequency-Inverse Document Frequency), we refer to a Vector Space Model where:

- Terms: words, n-grams, etc.
- Weights: higher weight to terms that are frequent in the document but not common in the corpus.

```
from sklearn.feature_extraction.text import TfidfVectorizer
vectorizer = TfidfVectorizer(analyzer = "word", ngram_range = (1,2))

# Learn the vocabulary dictionary and return term-document matrix.
vectorizer.fit_transform(corpus)
```

TfidfVectorizer Documentation

Let $n_{d,t}$ denote the nuber of times the t-th term appears in the d-th document.

$$TF_{d,t} = \frac{n_{d,t}}{\sum_i n_{d,i}}$$

Let N denote the total number of documents and N_t denote the nuber of documents containing the t-th term.

$$IDF_t = log\Big(\frac{N}{N_t}\Big)$$

TF-IDF weight:

$$w_{d,t} = TF_{d,t} \cdot IDF_t$$

Dimensionality Reduction

Feature Extraction

Feature extraction is intended to extract informative and non-redundant features, facilitating the subsequent learning.

- Stop-words removal
- Stemming/lemmatization
- Normalization
- Removing rare terms
- ...

It is especially important in text mining due to the high dimensionality of text features and the existence of irrelevant (noisy) features.

Feature Space Reduction

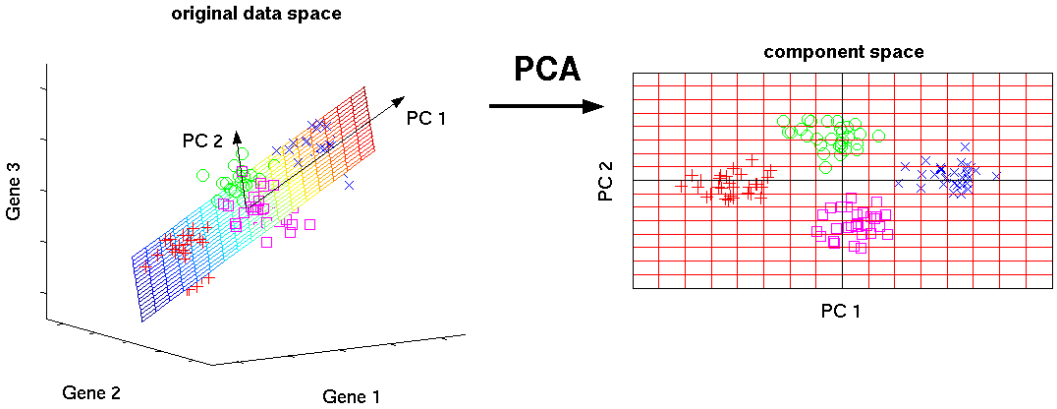
In Vector Space Models, a text will typically be a very sparsely populated vector living in a very high-dimensional space. It is often desirable to reduce the dimension of the feature space while retaining as much information as possible.

Given an $d \times t$ matrix $\mathbf{C}$ with t large, it is often desirable to project the rows onto a smaller-dimensional space, giving a matrix of shape $d \times k$ with $k \ll t$.

We would like this projection to keep the variance of the samples as large as possible, because this corresponds to losing as little information as possible.

Principal Component Analysis (PCA)

A standard method for feature space reduction is **Principal Component Analysis**, which projects a set of points onto a smaller dimensional affine subspace of "best fit".



Singular value decomposition

In general, if we want k features, it is optimal to take the k eigenvectors of $\mathbf{C}^{\mathsf{T}}\mathbf{C}$ with maximal eigenvalues.

Finding the singular vectors along with their eigenvalues is essentially the process known as singular value decomposition (SVD).

In PCA, we (usually) first perform some normalizations: we scale the columns to have variance 1 and translate them to have mean 0 before applying SVD. Geometrically, this amounts to normalizing the features to the same scale, i.e. giving them the same magnitude before applying SVD.

Take Home Concepts

Take Home Concepts

- Processing natural language text requires the text to be converted into a set of numerical features.
- Text data can be represented as vectors.
- BOW and TF-IDF are models to vectorize text data.
- Vectors representing text data are usually sparse and high-dimensional.
- PCA reduces the dimension of the vector space while retaining as much information as possible.

References

References

- https://towardsdatascience.com/the-magic-behind-embedding-models-part-1-974d539f21fd
  - read (5 mins)
- http://cs229.stanford.edu/proj2017/final-reports/5163902.pdf
  - (pag. 1-6)