

Text Preprocessing II

From textual information to numerical vector

Bag-of-Words: counting is everything

Vector Representation for Documents

- **Without any deep analysis of the linguistic content of the documents**, we can describe each document by features that represent the most frequent tokens.
- Each row is a document, and each column represents a feature.
- Thus, a cell in the csv/excel file is a measurement of a feature (corresponding to the column) for a document (corresponding to a row).

Bag-of-Words

- Steps
 - Build vocab i.e., set of all the words in the corpus
 - Count the occurrence of words in each document

The cat and the dog play
The cat is on the mat

corpus

and, the, cat, dog, play, on, mat, is
--

vocab.

1	2	1	1	1	0	0	0
1	2	0	0	1	1	1	1

countVec

Document Features

- How to define document features (i.e., entry value in the matrix)
 - Presence (0 or 1)
 - Frequencies (0,1,2,3)
 - Thresholding frequencies - three values
 - 0 (do not exist), 1 (occurred once), and 2 (occurred 2 or more times)

Term Frequency-Inverse Document Frequency

- Tf-idf(w): $tf(w) * idf(w)$, where $idf(w) = \log(1 + \frac{N}{df(w)})$
 - The tf-idf weight assigned to word w is the **term frequency** (i.e., the word count) modified by a scale factor for the importance of the word.
 - The scale factor is called the **inverse document frequency**, which checks the number of documents containing word w (i.e., $df(w)$) and reverses the scaling.
 - The N is the number of documents.

Term Frequency-Inverse Document Frequency

- Intuitive logic:
 - Capture the importances of a word to document in a corpus
 - Importance of words is proportionally to the number of times a word appears
 - Importance of words is inversely proportionally to the document containing the word
 - Thus, when a word appears in many documents, it is considered unimportant and the scale is lowered, perhaps near zero, e.g., “the”, “I”, “on”, “document”, etc.

Multiword Features

- A variety of measures can be used for this purpose.
 - E.g., frequent n-grams, such as “text mining”, “hip hop”
- As another method, an Association Measure AM for the multiword T, is used for evaluation multiword features, where $size(T)$ is the number of words in phrase T and $freq(T)$ is the number of times phrases T occurs in the document collection.

$$AM(T) = \frac{size(T) \log_{10}(freq(T)) freq(T)}{\sum_{word_i \in T} freq(word_i)}$$

- Generally, multiword features are not found too frequently in a document collection, but when they do occur they are often high predictive.

Compressed Sparse Row (CSR) Representation

- When prepare the feature matrix, most of the entries will be zero.
- Most documents contain a small subset of the vocab's words
- Rather than storing all the zeros, it may be better to represent the matrix as a set of sparse vectors, where a row is represented by a list of pairs, one element of the pair being a column number and the other element being the corresponding nonzero feature value.

0	15	0	3
12	0	0	0
8	0	5	2

(2,15) (4,3)
(1,12)
(1,8) (3,5) (4,2)

Bag-of-Words

- Pros

- Simple
- Surprisingly effective
- Fast

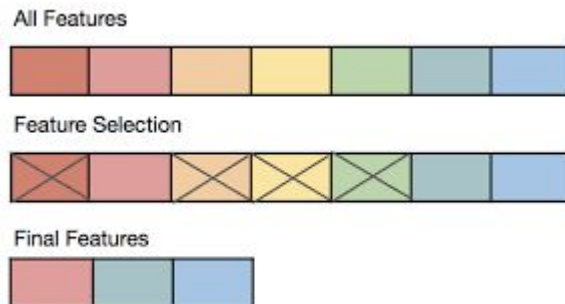
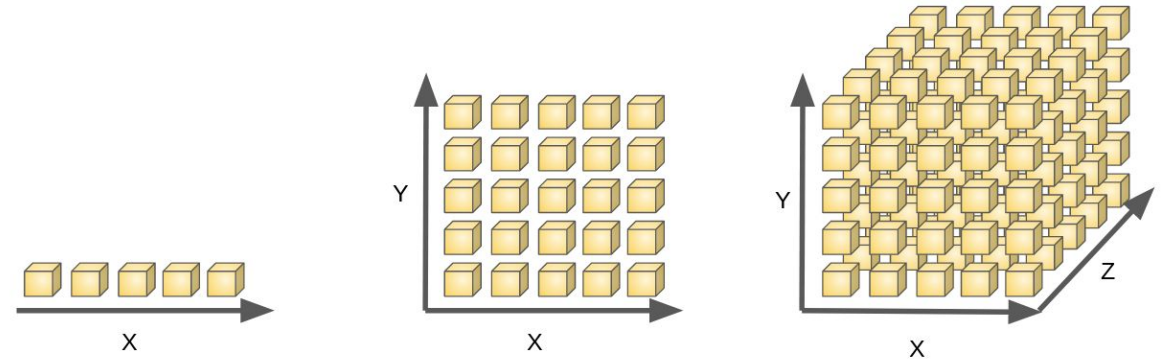
- Cons

- Order of words does not matter
- Cannot capture syntactic/semantic information
- **High dimensionality**

Dictionary Reduction

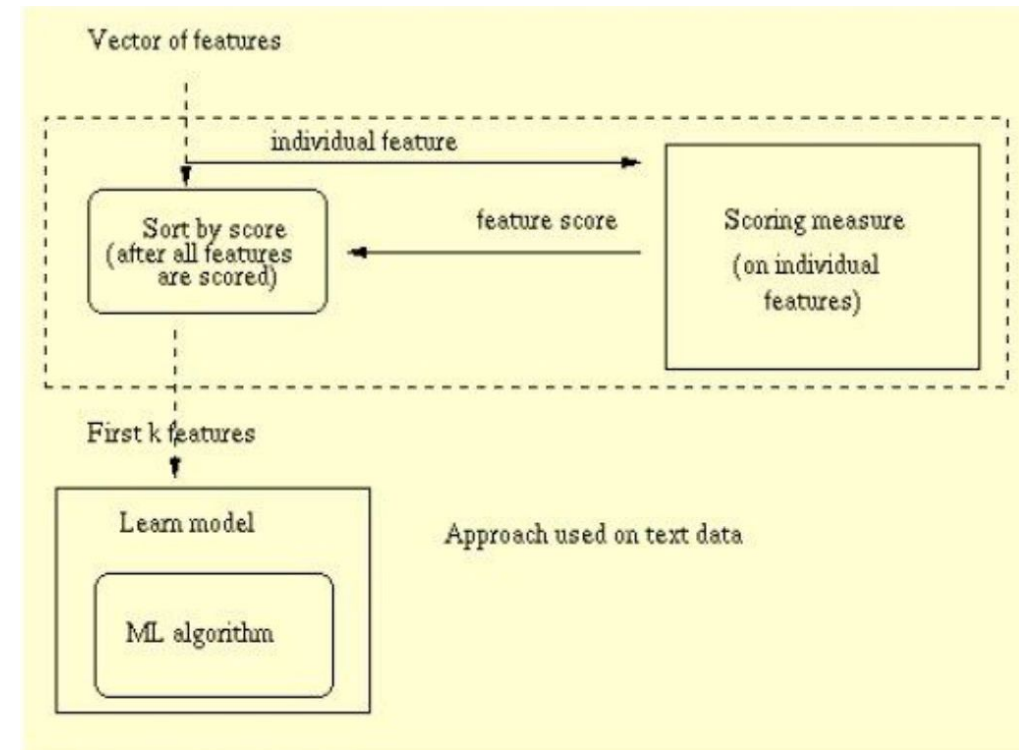
Dictionary Reduction

- Also called feature reduction techniques
- Due to **curse of high dimensionality**
- For BoW models:
 - Local dictionary
 - Removing Stopwords
 - Frequent Words
 - **Feature Selection**
 - Token reduction (stemming and synonyms)
 - Feature transformation (PCA, or Topic models)



Feature Selection by Attribute Ranking

- Can select a set of features (e.g., a set of words) to form a local dictionary.
- Rank feature attributes according to their predictive abilities for the category under consideration.
 - **Sports:** *soccer, football, etc.* **Travel:** *airport, cruise, etc*
- In this approach, simply select the top-ranking features.
- Feature Selection approaches:
 - Document Frequency
 - Information Gain
 - Mutual Information
 - CHI
 - [A survey](#)



Feature Selection based on Information Gain

On Widely Used Example

Outlook	Temperature	Humidity	Windy	Play
Sunny	Hot	High	False	<i>No</i>
Sunny	Hot	High	True	<i>No</i>
Overcast	Hot	High	False	<i>Yes</i>
Rainy	Mild	High	False	<i>Yes</i>
Rainy	Cool	Normal	False	<i>Yes</i>
Rainy	Cool	Normal	True	<i>No</i>
Overcast	Cool	Normal	True	<i>Yes</i>
Sunny	Mild	High	False	<i>No</i>
Sunny	Cool	Normal	False	<i>Yes</i>
Rainy	Mild	Normal	False	<i>Yes</i>
Sunny	Mild	Normal	True	<i>Yes</i>
Overcast	Mild	High	True	<i>Yes</i>
Overcast	Hot	Normal	False	<i>Yes</i>
Rainy	Mild	High	True	<i>No</i>

Features/
Attributes

Label

You may think the most important feature is the one that can be most related to the label.

Impurity of Splits

- S contain 20 occurrences of P and 20 of N.
- Assume each data has three binary features f_1, f_2, f_3 . Then, based on each feature, we are going to have three possible splits on the data.
- S1 means the feature is 0 and S2 means the feature is 1.
- For feature 1: $S_1 = 20P$ and $S_2 = 20N$
- For feature 2: $S_1 = 10P, 10N$ and $S_2 = 10P, 10N$
- For feature 3: $S_1 = 17P, 1N$ and $S_2 = 3P, 19N$

Entropy

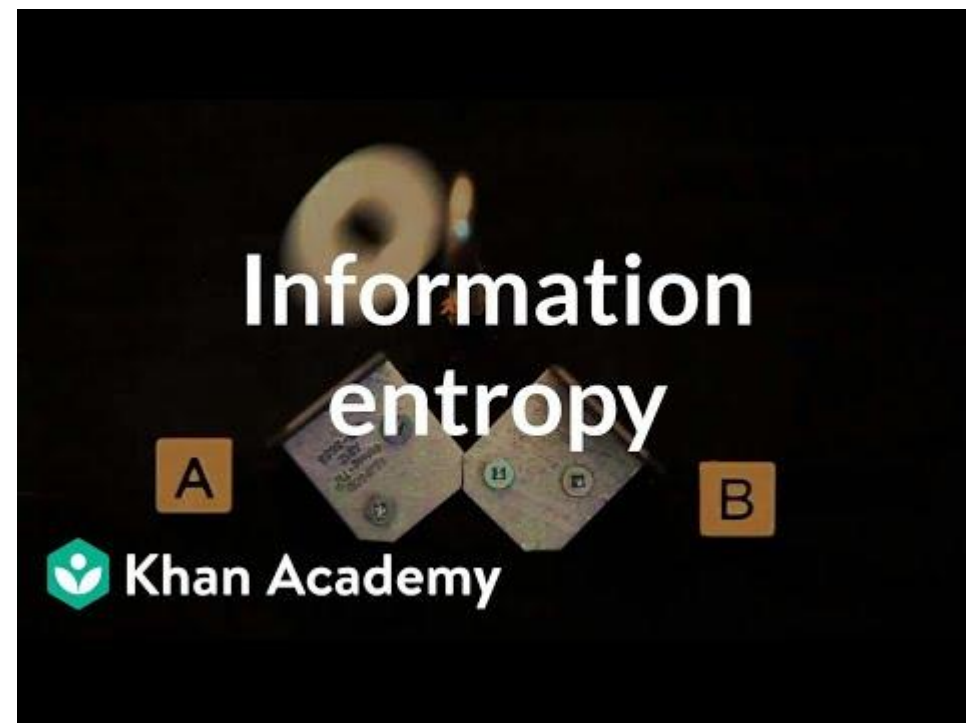
- Entropy is the measure of the information in a set of examples.

$$Entropy = - \sum_{i=1}^K p_i \log_2 p_i$$

- Where $i=\{1,\dots,K\}$, K is the number of possible actions, p_i is the proportion of each action i in the example set
- For example:

$$Entropy([9*, 5+, 6-]) = -\frac{9}{20} \log_2 \frac{9}{20} \\ - \frac{5}{20} \log_2 \frac{5}{20} - \frac{6}{20} \log_2 \frac{6}{20}$$

- High Entropy: more information
- Low Entropy: less information



Properties of Entropy

- Maximized when events are heterogeneous (impure):
 - A set of many mixed classes (say, rgb ○○○) is unpredictable. High Entropy

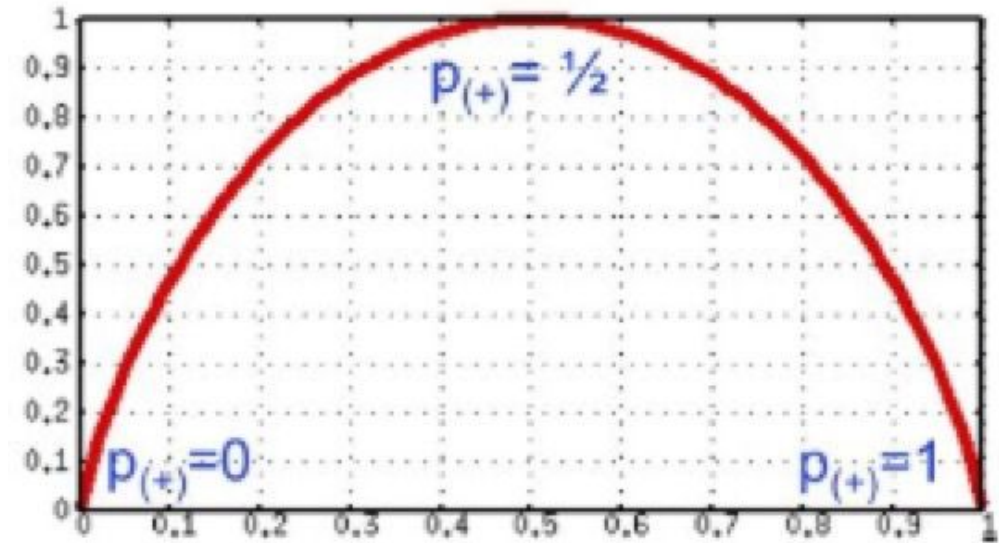
$$\textit{Entropy} = \log_2 K \quad \text{if all } p_i = \frac{1}{K}$$

- Minimized when events are homogenous (pure):
 - A set of only one class (say, blue ○○○) is extremely predictable. Low entropy

$$\textit{Entropy} = 0 \quad \text{if one } p_i = 1 \text{ the rest are zeros}$$

Entropy for binary case

- S is a sample of training examples
 - P_+ is the proportion of positive examples in S
 - P_- is the proportion of negative examples in S
- Entropy measures the impurity of S



$$\text{Entropy}(S) = -p_+ \log_2 p_+ - p_- \log_2 p_-$$

$$\text{Entropy}([9+, 5-]) = -\frac{9}{14} \log_2 \left(\frac{9}{14} \right) - \frac{5}{14} \log_2 \left(\frac{5}{14} \right) = 0.94$$

Information Gain

- Entropy:

$$E(X) = - \sum_{i=1}^K p(X = X_i) \log_2 p(X = X_i)$$

- **Intuition:** uncertainty of X, information contained in X, expected information bits required to represent X.

- Conditional Entropy

$$E(X|Y) = \sum_{i=1} p(Y = Y_i) E(X|Y = Y_i)$$

- **Intuition:** given y, how much uncertainty remains in X

- **Mutual Information (Information Gain)**

$$I(X, Y) = E(X) - E(X|Y) = E(Y) - E(Y|X)$$

High IG, More Entropy Removed

- **Intuition:** how much knowing Y reduces uncertainty about X, and vice versa.

IG: Example

Outlook	Temperature	Humidity	Windy	Play
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Overcast	Hot	High	False	Yes
Rainy	Mild	High	False	Yes
Rainy	Cool	Normal	False	Yes
Rainy	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Sunny	Mild	High	False	No
Sunny	Cool	Normal	False	Yes
Rainy	Mild	Normal	False	Yes
Sunny	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Rainy	Mild	High	True	No

$$\begin{aligned} E &= - \sum_{k=1}^K p_k \log_2 p_k \\ &= - \frac{5}{14} \log_2 \frac{5}{14} - \frac{9}{14} \log_2 \frac{9}{14} \\ &= 0.94 \end{aligned}$$

IG: Example

Outlook	Temperature	Humidity	Windy	Play
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Overcast	Hot	High	False	Yes
Rainy	Mild	High	False	Yes
Rainy	Cool	Normal	False	Yes
Rainy	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Sunny	Mild	High	False	No
Sunny	Cool	Normal	False	Yes
Rainy	Mild	Normal	False	Yes
Sunny	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Rainy	Mild	High	True	No

$$\begin{aligned}\Delta E(\text{Humidity}) &= E - \frac{m_{i=H}}{m} E(i = N) - \frac{m_{i=N}}{m} E(i = H) \\ &= 0.94 - \frac{7}{14} H_L - \frac{7}{14} H_R\end{aligned}$$

IG: Example

Outlook	Temperature	Humidity	Windy	Play
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Overcast	Hot	High	False	Yes
Rainy	Mild	High	False	Yes
Rainy	Cool	Normal	False	Yes
Rainy	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Sunny	Mild	High	False	No
Sunny	Cool	Normal	False	Yes
Rainy	Mild	Normal	False	Yes
Sunny	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Rainy	Mild	High	True	No

$$\Delta E(\text{Humidity}) = E - \frac{m_{i=H}}{m} E(i = N) - \frac{m_{i=N}}{m} E(i = H)$$

$$= 0.94 - \frac{7}{14} H_L - \frac{7}{14} H_R$$

$$H_L = -\frac{6}{7} \log_2 \frac{6}{7} - \frac{1}{7} \log_2 \frac{1}{7}$$

IG: Example

Outlook	Temperature	Humidity	Windy	Play
Sunny	Hot	High	False	<i>No</i>
Sunny	Hot	High	True	<i>No</i>
Overcast	Hot	High	False	<i>Yes</i>
Rainy	Mild	High	False	<i>Yes</i>
Rainy	Cool	Normal	False	<i>Yes</i>
Rainy	Cool	Normal	True	<i>No</i>
Overcast	Cool	Normal	True	<i>Yes</i>
Sunny	Mild	High	False	<i>No</i>
Sunny	Cool	Normal	False	<i>Yes</i>
Rainy	Mild	Normal	False	<i>Yes</i>
Sunny	Mild	Normal	True	<i>Yes</i>
Overcast	Mild	High	True	<i>Yes</i>
Overcast	Hot	Normal	False	<i>Yes</i>
Rainy	Mild	High	True	<i>No</i>

$$\Delta E(\text{Humidity}) = E - \frac{m_{i=H}}{m} E(i = N) - \frac{m_{i=N}}{m} E(i = H)$$

$$= 0.94 - \frac{7}{14} H_L - \frac{7}{14} H_R$$

$$H_L = -\frac{6}{7} \log_2 \frac{6}{7} - \frac{1}{7} \log_2 \frac{1}{7}$$
$$= 0.592$$

$$H_R = -\frac{3}{7} \log_2 \frac{3}{7} - \frac{4}{7} \log_2 \frac{4}{7}$$
$$= 0.985$$

IG: Example

Outlook	Temperature	Humidity	Windy	Play
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Overcast	Hot	High	False	Yes
Rainy	Mild	High	False	Yes
Rainy	Cool	Normal	False	Yes
Rainy	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Sunny	Mild	High	False	No
Sunny	Cool	Normal	False	Yes
Rainy	Mild	Normal	False	Yes
Sunny	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Rainy	Mild	High	True	No

$$\Delta E(\text{Humidity}) = E - \frac{m_{i=H}}{m} E(i = N) - \frac{m_{i=N}}{m} E(i = H)$$

$$= 0.94 - \frac{7}{14} H_L - \frac{7}{14} H_R$$

$$\begin{aligned}
 & 0.94 - \frac{7}{14} 0.592 - \frac{7}{14} 0.985 \\
 & = 0.94 - 0.296 - 0.4925 \\
 & = 0.1515
 \end{aligned}$$

IG: Example

Outlook	Temperature	Humidity	Windy	Play
Sunny	Hot	High	False	<i>No</i>
Sunny	Hot	High	True	<i>No</i>
Overcast	Hot	High	False	<i>Yes</i>
Rainy	Mild	High	False	<i>Yes</i>
Rainy	Cool	Normal	False	<i>Yes</i>
Rainy	Cool	Normal	True	<i>No</i>
Overcast	Cool	Normal	True	<i>Yes</i>
Sunny	Mild	High	False	<i>No</i>
Sunny	Cool	Normal	False	<i>Yes</i>
Rainy	Mild	Normal	False	<i>Yes</i>
Sunny	Mild	Normal	True	<i>Yes</i>
Overcast	Mild	High	True	<i>Yes</i>
Overcast	Hot	Normal	False	<i>Yes</i>
Rainy	Mild	High	True	<i>No</i>

1. Compute the information gain for the rest three features:
 - outlook
 - temperature
 - windy
2. Should we select features with high IG or low IG?

When it comes to text mining

- The previous features/attributes will be “words” or “terms”
- The information gain of a term measures:
 - The expected reduction in entropy caused by partitioning the sample documents according to the term:

$$IG(t) = - \sum_{i=1}^m p(c_i) \log p(c_i) + p(t) \sum_{i=1}^m p(c_i|t) \log p(c_i|t) + p(\bar{t}) \sum_{i=1}^m p(c_i|\bar{t}) \log p(c_i|\bar{t})$$

where

t is a term,

m is the total number of classes

$p(c_i)$ is the percentage of documents in category c_i from total sample documents

$p(t)$ is the percentage of documents in which term t is present

$p(\bar{t})$ is the percentage of documents in which term t is absent

$p(c_i|t)$ is the conditional probability of category given term t

$p(c_i|\bar{t})$ is conditional probability of category given term t is absent