Text Categorization with KL-Divergence

Harrison Kaiser, Matt Asnes

Using Kullback-Leibler Distance for Text Categorization

“This paper introduces a new effective model for text categorization with great corpus (more or less 1 million documents). Text categorization is performed using the Kullback-Leibler distance between the probability distribution of the document to classify and the probability distribution of each category. Using the same representation of categories, experiments show a significant improvement when the above mentioned method is used. KLD method achieve substantial improvements over the tf-idf performing method.”

Text Categorization

- What is it?
  - Given text documents and predetermined categories
  - Classify documents into categories
- Various Methods
  - tf-idf
  - SVM
  - Naive Bayes
  - K-NN

Term Frequency Inverse Document Frequency (tf-idf)

- Term Frequency
  - Measure of the frequency of terms appearing in a document; various definitions
    - Raw count
    - Raw count / Total words
    - Booleans
    - log(1 + raw count)
    - …
- Inverse Document Frequency
  - Measure of the significance of a word (if the word “the” appears in all docs, not very useful when categorizing
    - log([total # of documents] / [# of documents term appears in])
    - Many different weighting schemes…

tf-idf Example

Consider 2 documents:

Terms	Counts
Hello	1
world	1
whats	2
up	1

Terms	Counts
Hello	1
world	1
howbout	2
you	1

$$\text{tf}(\text{“Hello”}, d_1) = 1$$
$$\text{tf}(\text{“Hello”}, d_2) = 1$$
$$\text{tf}(\text{“whats”}, d_1) = 2$$
$$\text{tf}(\text{“whats”}, d_2) = 0$$
$$\text{idf}(\text{“}Hello''\text{)} = \log\left(\frac{2}{2}\right) = 0$$
$$\text{idf}(\text{“}whats''\text{)} = \log\left(\frac{1}{2}\right)$$

Justification for tf-idf

¯_(ツ)_/¯ … It works…

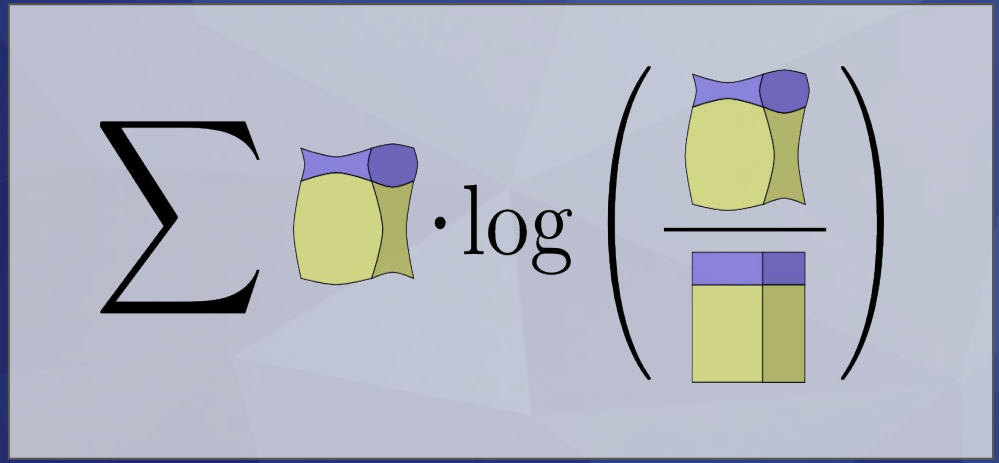
“idf was introduced, as ‘term specificity’ by Karen Spärck Jones in a 1972 paper. Although it has worked well as a heuristic, its theoretical foundations have been troublesome for at least three decades afterward, with many researchers trying to find information theoretic justifications for it.”

tf-idf In Detail

- Training
  - For each document, calculate tf vector (word counts)
  - For each word (“term”), calculate idf (log of [# docs] / [# docs term is in])
  - Multiply them to get tf-idf vector for each document
  - Get category vectors by summing tf-idf vectors over documents in that category
- Testing
  - For each document, calculate tf vector
  - Take a measure of similarity between tf vector and each category vector
    - Typically the cosine between the vectors (sum of product of weights divided by sqrt of sum of magnitudes of the vectors)
  - Report category with maximum similarity to this document

KL Divergence

- KL divergence measures difference between two similar probability distributions



What About Using KL Divergence?

- Goal: construct probability distributions for document vector and category vector, and take their KL divergence
- Problem 1: KL divergence is not symmetric
- Fix: Use a symmetric equivalent instead

$$D(P \,||\, Q) = -\sum_i P(i) \log\left(\frac{Q(i)}{P(i)}\right) \Rightarrow$$

$$D(P \,||\, Q) = -\sum_i (P(i) - Q(i)) \log\left(\frac{Q(i)}{P(i)}\right)$$

The Probability Distributions

- Have some list of words, either a document or a category
- Call this list **W**, and some given word (‘term’) **t**
- Then our distribution is:

$$P(t,W) = \begin{cases} \beta \dfrac{\mathrm{tf}(t,W))}{\sum_{x \in W} \mathrm{tf}(x,W)} & t \in W \\ \epsilon & t \notin W \end{cases}$$

- We have one distribution for W of the current document, and one for W of the each category
- Why ϵ and not 0? KL divergence explodes if one distribution is ever 0
- Then β is a normalization term based on # words
- ϵ is the same for documents and categories; sufficiently small

Normalization

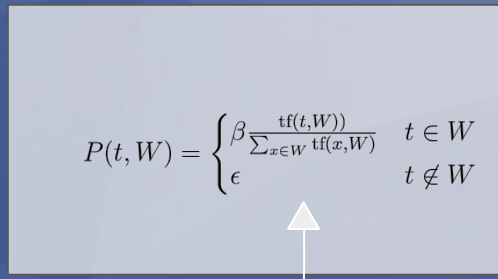
- Last step: normalize this KL distance across documents
- Given document and category, normalize as follows:

$$KL^*(c_i, d_j) = \frac{KL(c_i, d_j)}{KL(c_i, \vec{\epsilon})}$$

where the ϵ vector is a vector of the same length as c_i with each term being ϵ

Minimize (i.e. The Algorithm)

- KL divergence is 0 under identical distributions; then:
- Training
  - Compute tf vectors as before
  - Calculate probability distribution for each category based on these tf vectors
- Testing
  - For each document, calculate the document probability distribution
  - For each category, calculate KL* with this document
  - Find minimum KL* over all categories
  - Report category which had lowest KL* with this document



Implementation

- Used Reuters rcv1-v2 text categorization corpus
- 23,000 articles training, 800,000 testing
- 103 categories, articles member of ~5 categories (max 14)

Dataset Example

.I 26152

.W

world world world world qualif qualif sunday minut minut won hold athlet time time time time time lucky lucky komen komen komen komen break break record record record daniel keny

keny keny keny keny keny made shat noureddin morcel morcel morcel morcel morcel morcel morcel met met met met met met second second second second intern meet year year fail fail

atlant olymp olymp olymp olymp clock clock clock clock set set ago mont carl blist form grand grand prix prix circuit mark mark monac month brussel august fast fast hist hist

finish finish back back sixth plac david kisang led field lead carry ahead ahead near rival shem koror italian gennar di napol champ champ young good deserv today result told report

mean thing ve mile comfort burund venust niyongab born wilson kipket run denmark

Results

- KL Divergence slower, contradicting paper (Python)
- Both implementations ~75% accurate in identifying a document as one of its ~5 categories
- Paper looked into recall and precision
- Main benefit: theoretically founded; no corner-cases

END