Vector Space Model, Probabilistic Model

국민대학교 컴퓨터공학부

강 승 식

Vector Space Model

- Motivation
  - Assign *non-binary* weights to index terms
  - A framework in which partial matching is possible
    - Instead of attempting to predict whether a document is relevant or not
    - Rank the documents according to their degree of similarity to the query

Similarity: Query and Document

$$\vec{q} = (w_{1q}, w_{2q}, \ldots, w_{tq}) \qquad w_{iq} \geq 0$$

$$\vec{d}_j = (w_{1j}, w_{2j}, \ldots, w_{tj}) \qquad w_{ij} \geq 0$$

$$sim(d_j, q) = \frac{\vec{d}_j \cdot \vec{q}}{|\vec{d}_j| \times |\vec{q}|} = \frac{\sum_{i=1}^{t} w_{ij} \times w_{iq}}{\sqrt{\sum_{i=1}^{t} w_{ij}^2} \times \sqrt{\sum_{i=1}^{t} w_{iq}^2}}$$

$$0 \leq sim(d_j, q) \leq 1 \qquad \text{(cosine similarity)}$$

$|\vec{q}|$: Does not affect the ranking

$|\vec{d}_j|$: Normalization in the space of the documents

Tf-idf

- Clustering Problem
  - Intra-cluster similarity
    - What are the features which better *describe* the objects
  - Inter-cluster similarity
    - What are the features which better *distinguish* the objects

- IR Problem
  - Intra-cluster similarity (*tf* factor)
    - Raw frequency of a term k_i inside a document d_j
  - Inter-cluster similarity (*idf* factor)
    - Inverse of the frequency of a term k_i among the documents

Weighting Scheme

- Term Frequency (*tf*)
  - Measure of *how well that term describes the document* contents

$$f_{ij} = \frac{freq_{ij}}{\max_l \ freq_{lj}} \quad (\, freq_{ij} : \text{Raw frequency of term } k_i \text{ in the document } d_j \,)$$

- Inverse Document Frequency (*idf*)
  - *Terms which appear in many documents are not very useful for distinguishing* a relevant document from a non-relevant one

$$idf_i = \log \frac{N}{n_i}$$

n_i : Number of documents in which the index term k_i appears

N : Total number of documents

- Best known index term weighting scheme
  - Balance *tf* and *idf* (*tf-idf* scheme)

$$w_{ij} = f_{ij} \times idf_i$$

- Query term weighting scheme

$$w_{iq} = \left(0.5 + 0.5\, f_{iq}\right) \times idf_i$$

Example

Q : "gold silver truck"

D_1 : "Shipment of gold damaged in a fire"

D_2 : "Delivery of silver arrived in a silver truck"

D_3 : "Shipment of gold arrived in a truck"

$$idf_i = \log \frac{N}{n_i} \qquad w_{ij} = f_{ij} \times idf_i \qquad w_{iq} = f_{iq} \times idf_i$$

Term	a	arrived	damaged	delivery	fire	gold	in	of	silver	shipment	truck
idf	0	.176	.477	.477	.477	.176	0	0	.477	.176	.176

	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}	t_{11}
D_1	0	0	.477	0	.477	.176	0	0	0	.176	0
D_2	0	.176	0	.477	0	0	0	0	.954	0	.176
D_3	0	.176	0	0	0	.176	0	0	0	.176	.176
Q	0	0	0	0	0	.176	0	0	.477	0	.176

	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}	t_{11}
D_1	0	0	.477	0	.477	.176	0	0	0	.176	0
D_2	0	.176	0	.477	0	0	0	0	.954	0	.176
D_3	0	.176	0	0	0	.176	0	0	0	.176	.176
Q	0	0	0	0	0	.176	0	0	.477	0	.176

$$Sim(Q, D_j) = \sum_{i=1}^{t} w_{iq} \times w_{ij}$$

$$Sim(Q, D_1) = (0)(0) + (0)(0) + (0)(0.477) + (0)(0) + (0)(0.477)$$
$$+ (0.176)(0.176) + (0)(0) + (0)(0) + (0.477)(0) + (0)(0.176) + (0.176)(0)$$
$$= (0.176)^2 \approx 0.031$$

$$Sim(Q, D_2) = (0.954)(0.477) + (0.176)^2 \approx 0.486$$

$$Sim(Q, D_3) = (0.176)^2 + (0.176)^2 \approx 0.062$$

Hence, the ranking would be D_2, D_3, D_1

Probabilistic Model

Probabilistic Model

- Motivation
  - Attempt to capture the IR problem within a probabilistic framework
- Assumption (Probabilistic Principle)
  - Probability of relevance depends on the query and the document representations only
  - There is a subset of all documents which the user prefers as the answer set for the query *q*
  - Such an *ideal* answer set is labeled *R* and should maximize the overall probability of relevance to the user
  - Documents in the set *R* are predicted to be *relevant* to the query
  - Documents not in this set are predicted to be *non-relevant*

$(\overline{R})$

Probabilistic Model

- Definition

Using Bayes' rule

$P(R), P(\bar{R})$ are the same all the documents

$w_{ij} \in \{0,1\}, w_{iq} \in \{0,1\}$: index term weight variables are all binary

$$sim(d_j, q) = \frac{P(R \mid \vec{d}_j)}{P(\bar{R} \mid \vec{d}_j)} = \frac{P(\vec{d}_j \mid R) \times P(R)}{P(\vec{d}_j \mid \bar{R}) \times P(\bar{R})} \sim \frac{P(\vec{d}_j \mid R)}{P(\vec{d}_j \mid \bar{R})}$$

R: Set of documents known to be relevant

$\bar{R}$: Set of documents known to be non-relevant

$P(R \mid \vec{d}_j)$: Probability that the document d_j is relevant to the query q

$$\boxed{Bayes' rule: P(a \mid b) = \frac{P(a \cap b)}{P(b)} = \frac{P(b \mid a) P(a)}{P(b)}}$$

$P(\vec{d}_j \mid R)$: Probability of randomly selecting the document d_j from the set R

$P(R)$: Probability that a document randomly selected from the entire collection is relevant.

Probabilistic Model

- Definition (Cont.)

$$sim(d_j,q) \sim \frac{\left(\prod_{g_i(\vec{d}_j)=1} P(k_i \mid R)\right) \times \left(\prod_{g_i(\vec{d}_j)=0} P(\bar{k}_i \mid R)\right)}{\left(\prod_{g_i(\vec{d}_j)=1} P(k_i \mid \bar{R})\right) \times \left(\prod_{g_i(\vec{d}_j)=0} P(\bar{k}_i \mid \bar{R})\right)}$$

Assuming independence of index terms

$$sim(d_j,q) \sim \sum_{i=1}^{t} w_{iq} \times w_{ij} \times \left(\log \frac{P(k_i \mid R)}{1-P(k_i \mid R)} + \log \frac{1-P(k_i \mid \bar{R})}{P(k_i \mid \bar{R})} \right)$$

Taking logarithms, ignoring constants, $P(k_i \mid R) + P(\bar{k}_i \mid R) = 1$

$P(k_i \mid R)$: the probability that the index term k_i is present in a document randomly selected from the set R

$P(\bar{k}_i \mid R)$: the probability that the index term k_i is not present in a document randomly selected from the set R

Probabilistic Model

- Initial Probability

$$P(k_i \mid R) = 0.5$$

$$P(k_i \mid \overline{R}) = \frac{n_i}{N}$$

n_i : number of documents which contain the index term k_i

- Improving Probability

$$P(k_i \mid R) = \frac{V_i}{V} \quad = \frac{V_i + 0.5}{V + 1} \quad = \frac{V_i + \frac{n_i}{N}}{V + 1}$$

$$P(k_i \mid \overline{R}) = \frac{n_i - V_i}{N - V} = \frac{n_i - V_i + 0.5}{N - V + 1} = \frac{n_i - V_i + \frac{n_i}{N}}{N - V + 1}$$

V : subset of documents initially retrieved

V_i : subset of V which contain the index term k_i

Adjustment the problems for small values of V and V_i

Probabilistic Model

- Advantage
  - Documents are ranked in decreasing order of their *probability* of being relevant
- Disadvantage
  - *Guess the initial separation* of documents into relevant and non-relevant sets
  - Binary weights
    - Does not take into account the frequency with which an index term occurs inside a document
  - Independence assumption for index terms

Brief Comparison of Classic Models

- Boolean Model
  - Weakest classic method
  - Inability to recognize partial matches
- Vector Model
  - Popular retrieval model
- Vector Model vs. Probabilistic Model
  - Croft
    - Probabilistic model provides a better retrieval performance
  - Salton, Buckley
    - Vector model is expected to outperform the probabilistic model with general collections

Example of Probabilistic Model

- Query : $q(k_1, k_2)$
- 5 documents

	d_1	d_2	d_3	d_4	d_5
k_1	1	0	1	1	0
k_2	0	1	0	1	1

- Assume d_2 and d_4 are relevant

$$P(k_1 \mid R) = \frac{1}{2} \quad P(k_1 \mid \overline{R}) = \frac{2}{3} \quad P(k_2 \mid R) = 1 \quad P(k_2 \mid \overline{R}) = \frac{1}{3}$$

Example of Probabilistic Model

Q: “gold silver truck”

D1: “Shipment of gold damaged in a fire.”

D2: “Delivery of silver arrived in a silver truck.”

D3: “Shipment of gold arrived in a truck.”

- Assume that documents D2 and D3 are relevant to the query

N= number of documents in the collection

R= number of relevant documents for a given query Q

n= number of documents that contain term t

r= number of relevant documents that contain term t

variable	Gold	Silver	Truck
N	3	3	3
R	2	2	2
n	2	1	2
r	1	1	2

Example of Probabilistic Model (Cont.)

- Computing Term Relevance Weight

$$tr_j = \log\left(\frac{r+0.5}{R-r+0.5} \div \frac{n-r+0.5}{(N-n)-(R-r)+0.5}\right)$$

$$gold: \log\left(\frac{1+0.5}{2-1+0.5} \div \frac{2-1+0.5}{3-2-2+1+0.5}\right) = \log\frac{1}{3} = -0.477$$

$$silver: \log\left(\frac{1+0.5}{2-1+0.5} \div \frac{1-1+0.5}{3-1-2+1+0.5}\right) = \log\frac{1}{0.333} = 0.477$$

$$truck: \log\left(\frac{2+0.5}{2-2+0.5} \div \frac{2-2+0.5}{3-2-2+2+0.5}\right) = \log\frac{5}{0.333} = 1.176$$

- Similarity Coefficient for each document
  - D1: -0.477, D2: 1.653, D3: 0.699

Review of Probability Theory

Review of Probability Theory

- Try to predict whether or not a baseball team (e.g. LA Dodgers) will win one of its games
  - $P(win) = 0.5,\ P(win \mid sunny) = 0.75,\ P(win \mid chanho) = 0.6$
  - $P(win \mid sunny, chanho) = ?$
- Let's assume the independence of evidences
  - $P(w \mid s,c) = \alpha,\ \ P(w \mid s) = \beta,\ \ P(w \mid c) = \gamma$
- By Bayes' Theorem

$$\alpha = P(w \mid s,c) = \frac{P(w,s,c)}{P(s,c)} = \frac{P(s,c \mid w)P(w)}{P(s,c)}$$

We can't calculate it, so…

Probability to lose

$$\frac{\alpha}{1-\alpha} = \frac{P(s,c \mid w)P(w)}{P(s,c)} \div \frac{P(s,c \mid l)P(l)}{P(s,c)} = \frac{P(s,c \mid w)P(w)}{P(s,c \mid l)P(l)}$$

Review of Probability Theory (Cont.)

- Independence assumption

$$\begin{pmatrix} P(s,c\mid w) = P(s\mid w)P(c\mid w) \\ P(s,c\mid l) = P(s\mid l)P(c\mid l) \end{pmatrix}$$

- Making substitutions

$$\frac{\alpha}{1-\alpha} = \frac{P(s,c\mid w)P(w)}{P(s,c\mid l)P(l)} = \frac{P(s\mid w)P(c\mid w)P(w)}{P(s\mid l)P(c\mid l)P(l)}$$

$$\beta = P(w\mid s) = \frac{P(s\mid w)P(w)}{P(s)}, \quad \frac{\beta}{1-\beta} = \frac{P(s\mid w)P(w)}{P(s\mid l)P(l)}$$

$$\gamma = P(w\mid c) = \frac{P(c\mid w)P(w)}{P(c)}, \quad \frac{\gamma}{1-\gamma} = \frac{P(c\mid w)P(w)}{P(c\mid l)P(l)}$$

Review of Probability Theory (Cont.)

- Making substitutions (Cont.)

$$\frac{\alpha}{1-\alpha} = \frac{P(s|w)P(c|w)P(w)}{P(s|l)P(c|l)P(l)}$$

$$= \left(\frac{\beta}{1-\beta}\frac{P(l)}{P(w)}\right)\left(\frac{\gamma}{1-\gamma}\frac{P(l)}{P(w)}\right)\frac{P(w)}{P(l)}$$

$$= \frac{\beta}{1-\beta} \times \frac{\gamma}{1-\gamma} \times \frac{P(l)}{P(w)}$$

$$= \frac{0.75}{0.25} \times \frac{0.6}{0.4} \times \frac{0.5}{0.5} = 4.5$$

- $\alpha = \frac{9}{11} = 0.818$

The probability of LA Dodgers will win its game on sunny days when Chanho plays