PURDUE UNIVERSITY

CS54701: Information Retrieval

Language Models

4 February 2016

Prof. Chris Clifton

Indiana Center for Database Systems ™

1

Retrieval Model: Language Models

- Introduction to language models
- Unigram language model
- Document language model estimation
  - Maximum Likelihood estimation
  - Maximum a posterior estimation
  - Jelinek Mercer Smoothing
- Model-based feedback

Language Models: Motivation

- Vector space model for information retrieval
  - Documents and queries are vectors in the term space
  - Relevance is measure by the similarity between document vectors and query vector
- Problems for vector space model
  - Ad-hoc term weighting schemes
  - Ad-hoc similarity measurement
  - No justification of relationship between relevance and similarity
- We need more principled retrieval models…

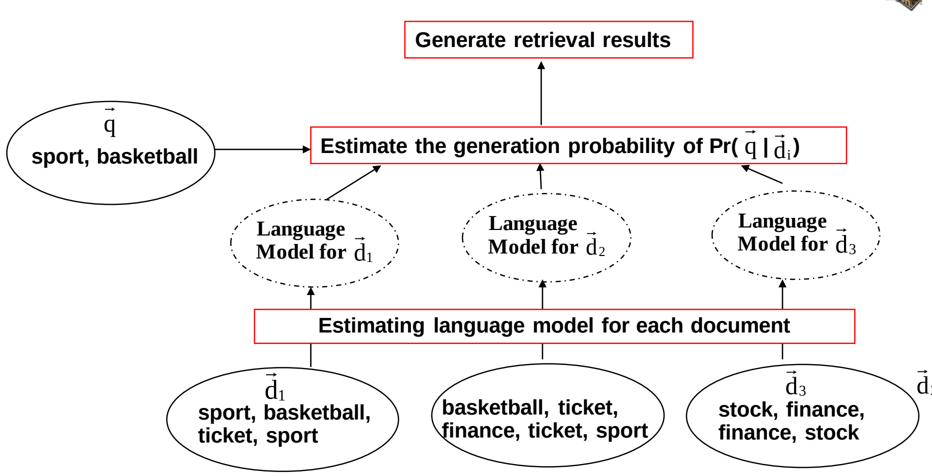
Introduction to Language Models:

- A Language model can be created for any language sample
  - A document
  - A collection of documents
  - Sentence, paragraph, chapter, query…
- The size of the language sample affects the quality of the language model
  - Long documents have a more accurate model
  - Short documents have a less accurate model
  - Model for sentence, paragraph or query may not be reliable

Introduction to Language Models:

- A document language model defines a probability distribution over indexed terms
  - E.g., the probability of generating a term
  - Sum of the probabilities is 1
- A query can be seen as observed data from unknown models
  - Query also defines a language model (more on this later)
- How might the models be used for IR?
  - Rank documents by $\Pr(\vec{q} \mid \vec{d}_i)$
  - Rank documents by language models of $\vec{q}$ and $\vec{d}_i$ based on kullback-Leibler (KL) divergence between the models (come later)

Language Model for IR: Example



Language Models

Three basic problems for language models

- What type of probabilistic distribution can be used to construct language models?
- How to estimate the parameters of the distribution of the language models?
- How to compute the likelihood of generating queries given the language modes of documents?

Multinomial/Unigram Language Models

- Language model built by multinomial distribution on single terms (i.e., unigram) in the vocabulary

Examples:

Five words in vocabulary (sport, basketball, ticket, finance, stock)

For a document $\vec{d}_i$, its language mode is:

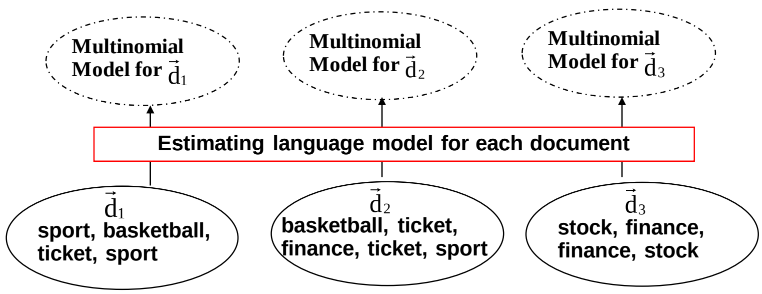
{P_i("sport"), P_i("basketball"), P_i("ticket"), P_i("finance"), P_i("stock")}

Formally:

The language model is: {$P_i(w)$ for any word w in vocabulary V}

$$\sum_k P_i(w_k) = 1 \qquad 0 \le P_i(w_k) \le 1$$

Multinomial/Unigram Language Models



Maximum Likelihood Estimation (MLE)

Maximum Likelihood Estimation:

- Find model parameters that make generation likelihood reach maximum:

$$M^*=\operatorname{argmax}_M \Pr(D|M)$$

There are K words in vocabulary, $w_1 \ldots w_K$ (e.g., 5)
Data: one document $\vec{d}_i$ with counts $tf_i(w_1), \ldots, tf_i(w_K)$, and length $|\vec{d}_i|$
Model: multinomial M with parameters $\{p_i(w_k)\}$
Likelihood: $\Pr(\vec{d}_i| M)$

$$M^*=\operatorname{argmax}_M \Pr(\vec{d}_i|M)$$

Maximum Likelihood Estimation (MLE)

$$p(\vec{d}_i \mid M) = \binom{|\vec{d}_i|}{tf_i(w_1)\ldots tf_i(w_K)} \prod_{k=1}^{K} p_i(w_k)^{tf_i(w_k)} \propto \prod_{k=1}^{K} p_i(w_k)^{tf_i(w_k)}$$

$$l(\vec{d}_i \mid M) = \log\ p(\vec{d}_i \mid M) = \sum_k tf_i(w_k) \log\ p_i(w_k)$$

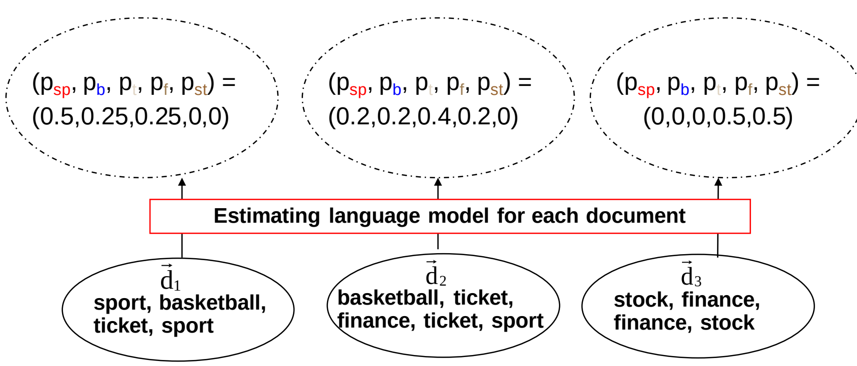
$$l'(\vec{d}_i \mid M) = \sum_k tf_i(w_k) \log\ p_i(w_k) + \lambda\left(\sum_k p_i(w_k) - 1\right)$$

$$\frac{\partial l'}{\partial p_i(w_k)} = \frac{tf_i(w_k)}{p_i(w_k)} + \lambda = 0 \quad \Rightarrow \quad p_i(w_k) = -\frac{tf_i(w_k)}{\lambda}$$

$$Since \sum_k p_i(w_k) = 1,\ \lambda = -\sum_k tf_i(w_k) = |\vec{d}_i| \qquad So,\ p_i(w_k) = \frac{c_i(w_k)}{|\vec{d}_i|}$$

Use Lagrange multiplier approach
Set partial derivatives to zero
Get maximum likelihood estimate

Maximum Likelihood Estimation (MLE)



Maximum Likelihood Estimation (MLE)

Maximum Likelihood Estimation:

- Assign zero probabilities to unseen words in small sample

A specific example:

Only two words in vocabulary (w_1=sport, w_2=business) like (head, tail) for a coin; A document $\vec{d}_i$ generates sequence of two words or draw a coin for many times

$$\Pr(\vec{d}_i \mid M) = \binom{\vec{d}_i}{tf_i(w_1)\ tf_i(w_2)} p_i(w_1)^{tf_i(w_1)} (1 - p_i(w_1))^{tf_i(w_2)}$$

Only observe two words (flip the coin twice) and MLE estimators are:

"business sport" $P_i(w_1)=0.5$

"sport sport" $P_i(w_1)=1$?

"business business" $P_i(w_1)=0$?

Maximum Likelihood Estimation (MLE)

A specific example:

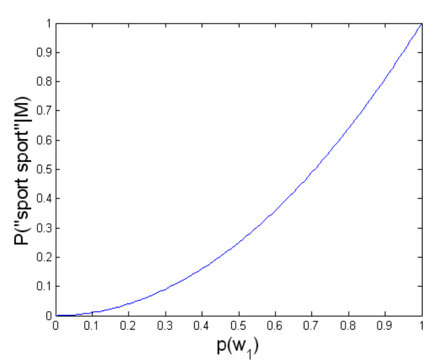
Only observe two words (flip the coin twice) and MLE estimators are:

"business sport" $P_i(w_1)^*=0.5$

"sport sport" $P_i(w_1)^*=1$?

"business business" $P_i(w_1)^*=0$?

Data sparseness problem



Solution to Sparse Data Problems

- Maximum a posterior (MAP) estimation
- Shrinkage
- Bayesian ensemble approach

Maximum A Posterior (MAP) Estimation

Maximum A Posterior Estimation:

- Select a model that maximizes the probability of model given observed data

  $M^*=\text{argmax}_M \Pr(M|D)=\text{argmax}_M \Pr(D|M)\Pr(M)$

  - Pr(M): Prior belief/knowledge
  - Use prior Pr(M) to avoid zero probabilities

A specific examples:

Only two words in vocabulary (sport, business)

For a document $\vec{d}_i$:

Prior Distribution

$$\Pr(M \mid \vec{d}_i) = \begin{pmatrix} \vec{d}_i \\ tf_i(w_1)\, tf_i(w_2) \end{pmatrix} p_i(w_1)^{tf_i(w_1)}\, p_i(w_2)^{tf_i(w_2)} \Pr(M)$$

Maximum A Posterior (MAP) Estimation

Maximum A Posterior Estimation:

- Introduce prior on the multinomial distribution
  - Use prior Pr(M) to avoid zero probabilities, most of coins are more or less unbiased
  - Use Dirichlet prior on p(w)

$$Dir(\vec{p}_i \mid \alpha_1, \ldots, \alpha_K) = \frac{\Gamma(\alpha_1 + \cdots + \alpha_K)}{\Gamma(\alpha_1)\cdots\Gamma(\alpha_K)} \prod_k p_i(w_k)^{\alpha_k - 1}, \quad \sum_k p_i(w_k) = 1, \ 0 \le p_i(w_k) \le 1$$

Hyper-parameters   Constant for p_K

$\Gamma(x)$ is gamma function

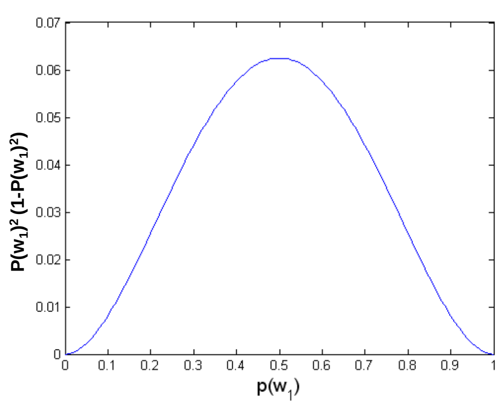
$$\Gamma(x) \equiv \int_0^\infty e^{-t} t^{x-1} dx$$

$$\Gamma(n+1) = n! \text{ if } n \in \mathbf{Z}$$

Maximum A Posterior (MAP) Estimation

For the two word example:

a Dirichlet prior $\Pr(M) \propto p(w_1)^2 (1 - p(w_1))^2$



Maximum A Posterior (MAP) Estimation

- **Maximum A Posterior:**

M*=argmax$_M$Pr(M|D)=argmax$_M$Pr(D|M)Pr(M)

$$\Pr(\vec{d}_i \mid M)\Pr(M) \propto p_i(w_1)^{tf_i(w_1)}(1-p_i(w_1))^{tf_i(w_2)}\, p_i(w_1)^{\alpha_1-1}\, p_i(w_1)^{\alpha_2-1}$$

$$= p_i(w_1)^{tf_i(w_1)+\boxed{\alpha_1-1}}(1-p_i(w_1))^{tf_i(w_2)+\boxed{\alpha_2-1}}$$

Pseudo Counts

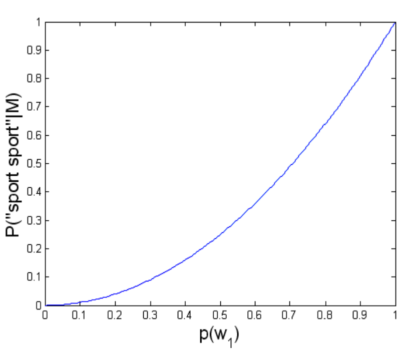
$$M^* = \arg\max_{p_i(w_1)} p_i(w_1)^{tf_i(w_1)+\alpha_1-1}(1-p_i(w_1))^{tf_i(w_2)+\alpha_2-1}$$

Maximum A Posterior (MAP) Estimation

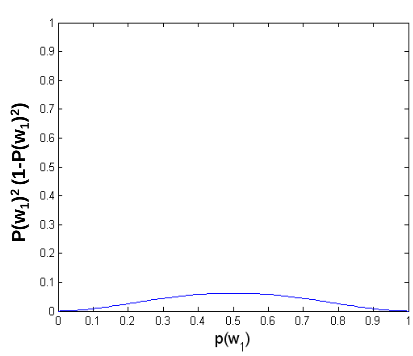
A specific example:

Only observe two words (flip a coin twice):

"sport sport" P$_i$(w$_1$)*=1 ?



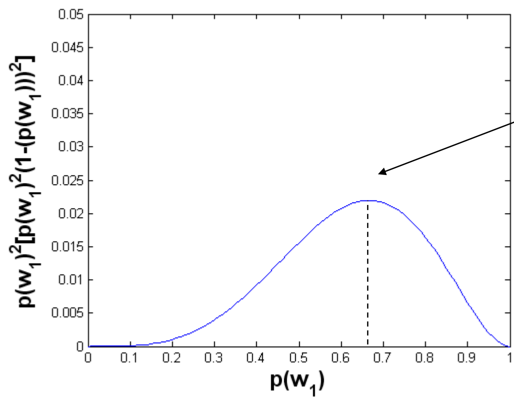
times



Maximum A Posterior (MAP) Estimation

A specific example:

Only observe two words (flip a coin twice):
"sport sport" $P_i(w_1)^*=1$?



$$p(w_1)^* = \frac{tf_i(w_1)+\alpha_1-1}{tf_i(w_1)+\alpha_1-1+tf_i(w_2)+\alpha_2-1}$$

$$= \frac{2+3-1}{2+3-1+0+3-1} = \frac{4}{6} = \frac{2}{3}$$

MAP Estimation Unigram Language Model

Maximum A Posterior Estimation:

- Use Dirichlet prior for multinomial distribution
- How to set the parameters for Dirichlet prior?

MAP Estimation
Unigram Language Model

Maximum A Posterior Estimation:

- Use Dirichlet prior for multinomial distribution

There are K terms in the vocabulary:

$$Multinomial: \vec{p}_i = \{p_i(w_1), ..., p_K(w_i)\}, \sum_k p_i(w_k) = 1,\ 0 \le p_i(w_k) \le 1$$

$$Dir(\vec{p}_i \mid \alpha_1, \ldots, \alpha_K) = \frac{\Gamma(\alpha_1 + \cdots + \alpha_K)}{\Gamma(\alpha_1)\cdots\Gamma(\alpha_K)} \prod_k p_i(w_k)^{\alpha_k - 1}, \quad \sum_k p_i(w_k) = 1,\ 0 \le p_i(w_k) \le 1$$

Hyper-parameters Constant for p_K

MAP Estimation
Unigram Language Model

MAP Estimation for unigram language model:

$$\vec{p}^{\,*} = \arg\max_{\vec{p}} \frac{\Gamma(\alpha_1 + \cdots + \alpha_K)}{\Gamma(\alpha_1)\cdots\Gamma(\alpha_K)} \prod_k p_i(w_k)^{tf_i(w_k)} \prod_k p_i(w_k)^{\alpha_k - 1}$$

$$st. \quad \sum_k p_i(w_k) = 1,\ 0 \le p_i(w_k) \le 1$$

$$= \arg\max_{\vec{p}} \prod_k p_i(w_k)^{tf_i(w_k) + \alpha_k - 1}$$

$$st. \quad \sum_k p_i(w_k) = 1,\ 0 \le p_i(w_k) \le 1$$

Use Lagrange Multiplier; Set derivative to 0

$$\vec{p}_i^{\,*}(w_k) = \frac{tf_i(w_k) + \boxed{\alpha_k - 1}}{\sum_k (tf_i(w_k) + \alpha_k - 1)}$$

Pseudo counts set by hyper-parameters

MAP Estimation
Unigram Language Model

MAP Estimation for unigram language model:

Use Lagrange Multiplier; Set derivative to 0

$$\vec{p}_i^{\,*}(w_k) = \frac{tf_i(w_k) + \boxed{\alpha_k - 1}}{\sum_k (tf_i(w_k) + \alpha_k - 1)}$$

How to determine the appropriate value for hyper-parameters?

- When nothing observed from a document

$$\vec{p}_i^{\,*}(w_k) = \frac{\alpha_k - 1}{\sum_k (\alpha_k - 1)}$$

- What is most likely $p_i(w_k)$ without looking at the content of the document?

MAP Estimation
Unigram Language Model

MAP Estimation for unigram language model:

- What is most likely $p_i(w_k)$ without looking at the content of the document?
- The most likely $p_i(w_k)$ without looking into the content of the document d is the unigram probability of the collection:
  - {$p(w_1|c)$, $p(w_2|c)$,…, $p(w_K|c)$}

Without any information, guess the behavior of one member on the behavior of whole population

Constant

$$\vec{p}_i^{\,*}(w_k) = \frac{\alpha_k - 1}{\sum_k (\alpha_k - 1)} = p_c(w_k) \quad \Rightarrow \quad \alpha_k - 1 = \mu p_c(w_k)$$

MAP Estimation Unigram Language Model

MAP Estimation for unigram language model:

$$\vec{p}^{\,*} = \arg\max_{\vec{p}} \frac{\Gamma(\alpha_1+\cdots+\alpha_K)}{\Gamma(\alpha_1)\cdots\Gamma(\alpha_K)} \prod_k p_i(w_k)^{tf_i(w_k)} \prod_k p_i(w_k)^{\mu p_c(w_k)}$$

$$st. \quad \sum_k p_i(w_k) = 1, \; 0 \le p_i(w_k) \le 1$$

$$= \arg\max_{\vec{p}} \prod_k p_i(w_k)^{tf_i(w_k)+\mu p_c(w_k)}$$

$$st. \quad \sum_k p_i(w_k) = 1, \; 0 \le p_i(w_k) \le 1$$

Use Lagrange Multiplier; Set derivative to 0

$$\vec{p}_i^{\,*}(w_k) = \frac{tf_i(w_k) + \boxed{\mu p_c(w_k)}}{\sum_k tf_i(w_k) + \mu}$$

Pseudo counts

Pseudo document length

Maximum A Posterior (MAP) Estimation

Dirichlet MAP Estimation for unigram language model:

Step 0: compute the probability on whole collection based collection unigram language model

$$p_c(w_i) = \frac{\sum_i tf_i(w_k)}{\sum_i |\vec{d}_i|}$$

Step 1: for each document $\vec{d}_i$, compute its smoothed unigram language model (Dirichlet smoothing) as

$$p_i(w_k) = \frac{tf_i(w_k) + \mu p_c(w_k)}{|\vec{d}_i| + \mu}$$

Maximum A Posterior (MAP) Estimation

Dirichlet MAP Estimation for unigram language model:

Step 2: For a given query $\vec{q}$ ={$tf_q(w_1)$,…, $tf_q(w_k)$}

- For each document $\vec{d}_i$, compute likelihood

$$p(q \mid \vec{d}_i) = \prod_{k=1}^{K} \left[p(w_i \mid \vec{d}_i) \right]^{tf_q(w_k)} = \prod_{k=1}^{K} \left[\frac{tf_i(w_k) + \mu p_c(w_k)}{\left|\vec{d}_i\right| + \mu} \right]^{tf_q(w_k)}$$

- The larger the likelihood, the more relevant the document is to the query