PURDUE
UNIVERSITY

CS54701: Information Retrieval

Language Models

9 February 2016

Prof. Chris Clifton

Indiana Center for Database Systems

Dirichlet Smoothing & TF-IDF

Dirichlet Smoothing:

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

?

TF-IDF Weighting:

$$sim(q, \vec{d}_i) = \sum_{k=1}^{K} tf_q(w_k) tf_i(w_k) idf(w_k) norm(\vec{d}_i)$$

31

Dirichlet Smoothing & TF-IDF

Dirichlet Smoothing:

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

$$\Downarrow$$

$$\log p(q \mid \vec{d}_i) = \sum_{k=1} tf_q(w_k) \left\{ \log\left(1 + \frac{tf_i(w_k)}{\mu p_c(w_k)}\right) - \log(\left|\vec{d}_i\right| + \mu) + \log \mu p_c(w_k) \right\}$$

$$\Downarrow$$

TF-IDF Weighting:

$$sim(q, \vec{d}_i) = \sum_{k=1}^{K} tf_q(w_k) tf_i(w_k) idf(w_k) norm(\vec{d}_i)$$

32

Dirichlet Smoothing & TF-IDF

Dirichlet Smoothing:

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

$$\log p(q \mid \vec{d}_i) = \sum_{k=1} tf_q(w_k) \left\{ \log\left(tf_i(w_k) + \mu p_c(w_k)\right) - \log(\left|\vec{d}_i\right| + \mu) \right\}$$

$$= \sum_{k=1} tf_q(w_k) \left\{ \log\left(\frac{\mu p_c(w_k) + tf_i(w_k)}{\mu p_c(w_k)} \mu p_c(w_k) \right) - \log(\left|\vec{d}_i\right| + \mu) \right\}$$

$$= \sum_{k=1} tf_q(w_k) \left\{ \log\left(1 + \frac{tf_i(w_k)}{\mu p_c(w_k)}\right) + \log \mu p_c(w_k) - \log(\left|\vec{d}_i\right| + \mu) \right\}$$

33

Dirichlet Smoothing & TF-IDF

Dirichlet Smoothing:

Irrelevant part

$$\log p(q \mid \vec{d}_i) = \sum_{k=1} tf_q(w_k)\left\{\log\left(1+\frac{tf_i(w_k)}{\mu p_c(w_k)}\right) - \log(|\vec{d}_i| + \mu) + \boxed{\log \mu p_c(w_k)}\right\}$$

$$\Downarrow$$

$$\log p(q \mid \vec{d}_i) \cong \sum_{k=1} tf_q(w_k)\left\{\log\left(1+\frac{tf_i(w_k)}{\mu p_c(w_k)}\right) - \log(|\vec{d}_i| + \mu)\right\}$$

TF-IDF Weighting:

$$sim(q, \vec{d}_i) = \sum_{k=1}^{K} tf_q(w_k) tf_i(w_k) idf(w_k) norm(\vec{d}_i)$$

Dirichlet Smoothing & TF-IDF

Dirichlet Smoothing:

Look at the tf.idf part

$$\log\left(1+\frac{tf_i(w_k)}{\mu p_c(w_k)}\right) \begin{cases} tf_i(w_k)\uparrow & \log\left(1+\frac{tf_i(w_k)}{\mu p_c(w_k)}\right)\uparrow \\ p_c(w_k)\uparrow & \log\left(1+\frac{tf_i(w_k)}{\mu p_c(w_k)}\right)\downarrow \end{cases}$$

Dirichlet Smoothing Hyper-Parameter

Dirichlet Smoothing:

Hyper-parameter

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

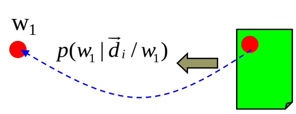
- When μ is very small, approach MLE estimator
- When μ is very large, approach probability on whole collection
- How to set appropriate μ ?

36

Dirichlet Smoothing Hyper-Parameter

Leave One out Validation:

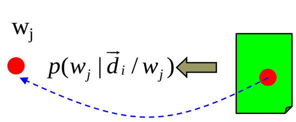
$$p_i(w_k) = \frac{tf_i(w_k) + \mu p_c(w_k)}{|\vec{d}_i| + \mu}$$



Leave w_1 out

$$p_i(w_1 | \vec{d}_i / w_1) = \frac{tf_i(w_1) - 1 + \mu p_c(w_1)}{|\vec{d}_i| - 1 + \mu}$$

⋮ ⋮



Leave w_j out

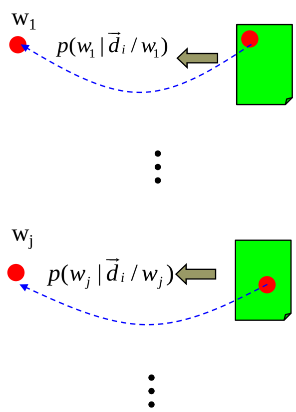
$$p_i(w_j | \vec{d}_i / w_j) = \frac{tf_i(w_j) - 1 + \mu p_c(w_j)}{|\vec{d}_i| - 1 + \mu}$$

⋮ ⋮

37

Dirichlet Smoothing Hyper-Parameter

Leave One out Validation:



Leave all words out one by one for a document

$$l_{-1}(\mu, \vec{d}_i) = \sum_{j=1}^{|\vec{d}_i|} \log\left(\frac{tf_i(w_j) - 1 + \mu p_c(w_j)}{|\vec{d}_i| - 1 + \mu}\right)$$

Do the procedure for all documents in a collection

$$l_{-1}(\mu, C) = \sum_{i=1}^{|C|} \sum_{j=1}^{|\vec{d}_i|} \log\left(\frac{tf_i(w_j) - 1 + \mu p_c(w_j)}{|\vec{d}_i| - 1 + \mu}\right)$$

Find appropriate μ

$$\mu^* = \arg\max_{\mu} l_{-1}(\mu, C)$$

38

Dirichlet Smoothing Hyper-Parameter

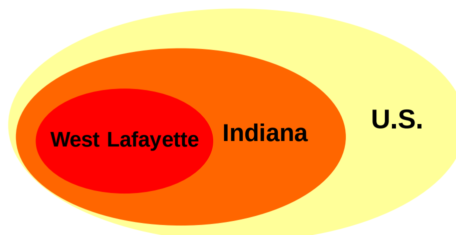
- What type of document/collection would get large μ ?
  - Most documents use similar vocabulary and wording pattern as the whole collection
- What type of document/collection would get small μ ?
  - Most documents use different vocabulary and wording pattern than the whole collection

39

Shrinkage

- Maximum Likelihood (MLE) builds model purely on document data and generates query word
  - Model may not be accurate when document is short (many unseen words)
- Shrinkage estimator builds more reliable model by consulting more general models (e.g., collection language model)

Example: Estimate P(Lung_Cancer|Smoke)



Shrinkage

- Jelinek Mercer Smoothing
  - Assume for each word, with probability λ , it is generated from document language model (MLE), with probability 1-λ , it is generated from collection language model (MLE)
  - Linear interpolation between document language model and collection language model

JM Smoothing:

$$p_i(w_k) = \lambda \frac{tf_i(w_k)}{|\vec{d}_i|} + (1-\lambda) p_c(w_k)$$

Shrinkage

- Relationship between JM Smoothing and Dirichlet Smoothing

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

$$= \frac{1}{|\vec{d}_i| + \mu}\left(tf_i(w_k) + \mu p_c(w_k)\right)$$

$$= \frac{1}{|\vec{d}_i| + \mu}\left(\frac{|\vec{d}_i| tf_i(w_k)}{|\vec{d}_i|} + \mu p_c(w_k)\right) = \boxed{\frac{|\vec{d}_i|}{|\vec{d}_i| + \mu}} \frac{tf_i(w_k)}{|\vec{d}_i|} + \boxed{\frac{\mu}{|\vec{d}_i| + \mu}} p_c(w_k)$$

JM Smoothing:

$$p_i(w_k) = \lambda \frac{tf_i(w_k)}{|\vec{d}_i|} + (1-\lambda) p_c(w_k)$$

42

Model Based Feedback

- Equivalence of retrieval based on query generation likelihood and Kullback-Leibler (KL) Divergence between query and document language models

Kullback-Leibler (KL) Divergence between two probabilistic distributions

$$KL(\vec{p} \| \vec{q}) = \sum_x p(x) \log\left(\frac{p(x)}{q(x)}\right)$$

- It is the distance between two probabilistic distributions
- It is always larger than zero

How to prove it ?

43

Model Based Feedback

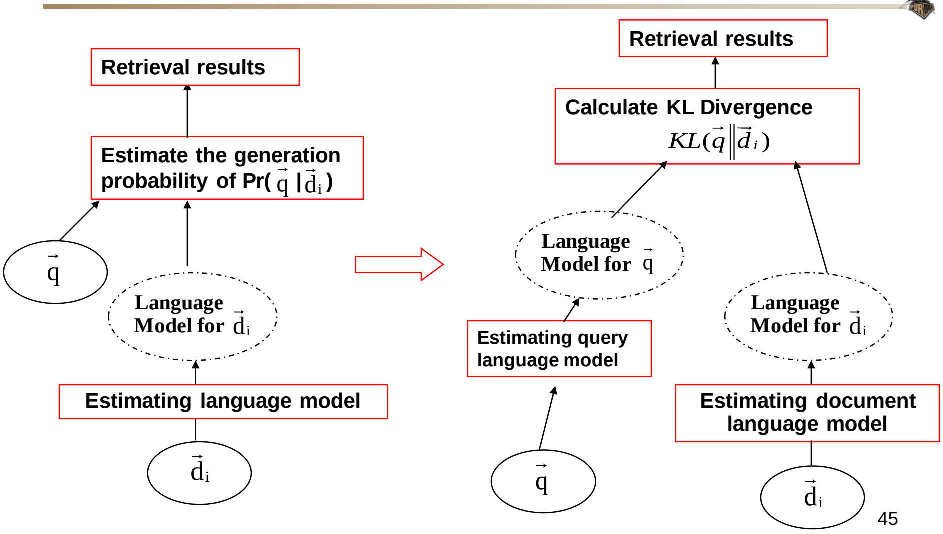
- Equivalence of retrieval based on query generation likelihood and Kullback-Leibler (KL) Divergence between query and document language models

$$
\begin{aligned}
Sim(\vec{q},\vec{d}_i) &= -KL(\vec{q}\,\|\,\vec{d}_i) \\
&= -\sum_w q(w)\log\left(\frac{q(w)}{p_i(w)}\right) \\
&= \underbrace{\sum_w q(w)\log\left(p_i(w)\right)}_{\text{Loglikelihood of query generation probability}} - \underbrace{\sum_w q(w)\log\left(q(w)\right)}_{\text{Document independent constant}}
\end{aligned}
$$

  - Generalize query representation to be a distribution (fractional term weighting)

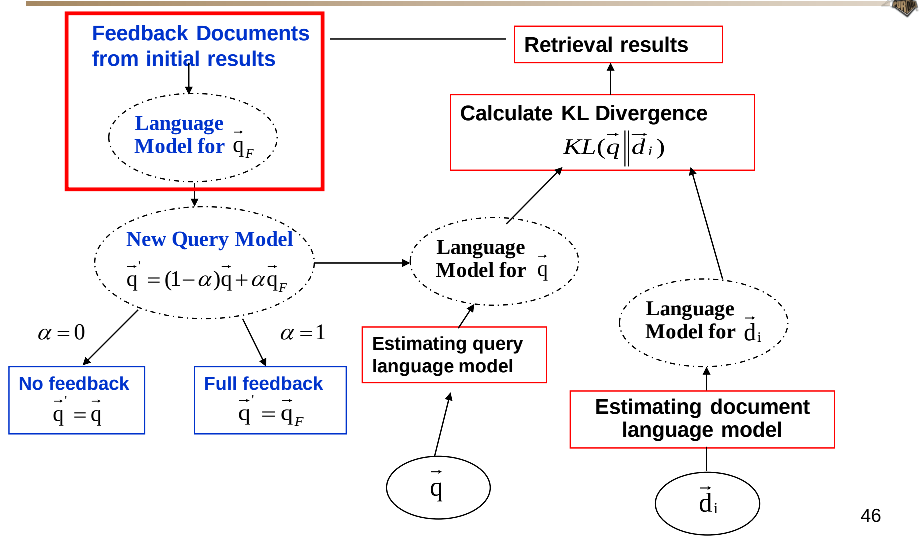
44

Model Based Feedback



45

Model Based Feedback

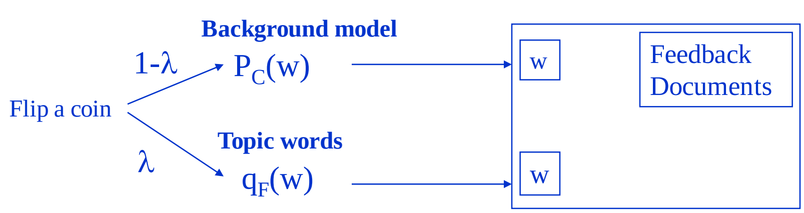


46

Model Based Feedback: Estimate $\vec{q}_F$

- Assume there is a generative model to produce each word within feedback document(s)

For each word in feedback document(s), given λ



$$\vec{q}_F^{\,*} = \arg\max_{\vec{q}_F} l(X,\lambda)$$

$$= \arg\max_{\vec{q}_F} \sum_{i=1}^{n} \left(\log\left(\lambda q_F(w_i) + (1-\lambda) p_C(w_i) \right) \right)$$

47

Model Based Feedback: Estimate $\vec{q}_F$

- For each word, there is a hidden variable telling which language model it comes from

Background Model $p_C(w|C)$

the 0.12
to 0.05
it 0.04
a 0.02
…
sport 0.0001
basketball 0.00005
…

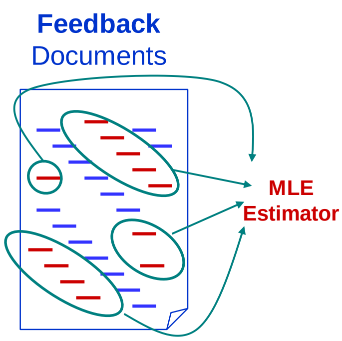
$1-\lambda=0.8$

Unknown query topic $p(w|\theta_F)$=?

"Basketball"

…
sport =?
basketball =?
game =?
player =?
…

$\lambda=0.2$



If we know the value of hidden variable of each word ...

Model Based Feedback: Estimate $\vec{q}_F$

- For each word, the hidden variable Z_i = {1 (feedback), 0 (background}
  - Step1: estimate hidden variable based current on model parameter **(Expectation)**

$$p(z_i=1|w_i)=\frac{p(z_i=1)p(w_i|z_i=1)}{p(z_i=1)p(w_i|z_i=1)+p(z_i=0)p(w_i|z_i=0)}$$

$$=\frac{\lambda q_F^{(t)}(w_i)}{\lambda q_F^{(t)}(w_i)+(1-\lambda)p_C(w_i|C)} \quad \text{E-step}$$

the (0.1) basketball (0.7) game (0.6) is (0.2) ….

  - Step2: Update model parameters based on the guess in step1 **(Maximization)**

$$q_F^{(t+1)}(w_i|\theta_F)=\frac{c(w_i,F)p(z_i=1|w_i)}{\sum_j c(w_j,F)p(z_j=1|w_j)} \quad \text{M-Step}$$

Model Based Feedback: Estimate $\vec{q}_F$

- Expectation-Maximization (EM) algorithm
  - Step 0: Initialize values of $\vec{q}_F^{\,0}$
  - Step1: **(Expectation)** $p(z_i=1|w_i)=\dfrac{\lambda q_F^{(t)}(w_i)}{\lambda q_F^{(t)}(w_i)+(1-\lambda)p_C(w_i|C)}$
  - Step2: **(Maximization)** $q_F^{(t+1)}(w_i|\theta_F)=\dfrac{c(w_i,F)p(z_i=1|w_i)}{\sum_j c(w_j,F)p(z_j=1|w_j)}$

Give λ=0.5

Word	#C(F,w)	p_C(w)	Initial	Iteration 1		Iteration 2	
			q_F(w)	p(z=1|w)	q_F(w)	p(z=1|w)	q_F(w)
the	4	0.5	0.25	0.33	0.21	0.30	0.19
good	2	0.4	0.25	0.38	0.12	0.23	0.07
basketball	4	0.1	0.25	0.71	0.45	0.82	0.52
game	2	0.1	0.25	0.71	0.22	0.69	0.22
Loglikelihood			-16.6	-15.7		-15.5	

Model Based Feedback: Estimate $\vec{q}_F$

- Properties of parameter λ
  - If λ is close to 0, most common words can be generated from the collection language model, so more topic words in the query language model
  - If λ is close to 1, the query language model has to generate most common words, so fewer topic words in the query language model

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

52