Dirichlet Smoothing & TF-IDF

Dirichlet Smoothing:

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

TF-IDF Weighting:

\[ \text{sim}(q, d_i) = \sum_{k=1}^{K} tf_q(w_k) tf_i(w_k)idf(w_k)\text{norm}(d_i) \]
Dirichlet Smoothing & TF-IDF

**Dirichlet Smoothing:**

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

\[
\log p(q | \vec{d}_i) = \sum_{k=1}^{K} tf_q(w_k) \left\{ \log (1 + \frac{tf_i(w_k)}{\mu p_c(w_k)}) - \log(d_i + \mu) + \log \mu p_c(w_k) \right\}
\]

**TF-IDF Weighting:**

\[
sim(q, \vec{d}_i) = \sum_{k=1}^{K} tf_q(w_k) tf_i(w_k) idf(w_k) \cdot \text{norm}(\vec{d}_i)
\]
**Dirichlet Smoothing & TF-IDF**

**Dirichlet Smoothing:**

\[
\log p(q | \tilde{d}_i) = \sum_{k=1}^{K} t_f(q)(w_k) \left\{ \log \left(1 + \frac{t_f_i(w_k)}{\mu p_c(w_k)}\right) - \log(|\tilde{d}_i| + \mu) + \log \mu p_c(w_k) \right\}
\]

**Irrelevant part**

\[
\log p(q | \tilde{d}_i) \approx \sum_{k=1}^{K} t_f(q)(w_k) \left\{ \log \left(1 + \frac{t_f_i(w_k)}{\mu p_c(w_k)}\right) - \log(|\tilde{d}_i| + \mu) \right\}
\]

**TF-IDF Weighting:**

\[
sim(q, \tilde{d}_i) = \sum_{k=1}^{K} t_f(q)(w_k)t_f_i(w_k) idf(w_k) \text{norm}(\tilde{d}_i)
\]

---

**Dirichlet Smoothing & TF-IDF**

**Dirichlet Smoothing:**

Look at the tf.idf part

\[
\log \left(1 + \frac{t_f_i(w_k)}{\mu p_c(w_k)}\right)
\]

\[
\log \left(1 + \frac{t_f_i(w_k)}{\mu p_c(w_k)}\right)
\]

\[
p_c(w_k)
\]

\[
l\log \left(1 + \frac{t_f_i(w_k)}{\mu p_c(w_k)}\right)
\]
Dirichlet Smoothing Hyper-Parameter

Dirichlet Smoothing:

\[ p_i(w_k) = \frac{tf_i(w_k) + \mu p_c(w_k)}{|d_i| + \mu} \]

- When \( \mu \) is very small, approach MLE estimator
- When \( \mu \) is very large, approach probability on whole collection
- How to set appropriate \( \mu \) ?

Leave One out Validation:

\[ p_i(w_k) = \frac{tf_i(w_k) + \mu p_c(w_k)}{|d_i| + \mu} \]

\[ p_i(w_1 | d_i / w_1) = \frac{tf_i(w_1) - 1 + \mu p_c(w_1)}{|d_i| - 1 + \mu} \]

\[ p_i(w_j | d_i / w_j) = \frac{tf_i(w_j) - 1 + \mu p_c(w_j)}{|d_i| - 1 + \mu} \]
Dirichlet Smoothing Hyper-Parameter

Leave One out Validation:

Leave all words out one by one for a document

\[ L_{-1}(\mu, \tilde{d}_i) = \sum_{j=1}^{|\tilde{d}_i|} \log \left( \frac{tf_i(w_j) - 1 + \mu \nu_c(w_j)}{|\tilde{d}_i| - 1 + \mu} \right) \]

Do the procedure for all documents in a collection

\[ L_{-1}(\mu, C) = \sum_{i=1}^{|C|} \sum_{j=1}^{|\tilde{d}_i|} \log \left( \frac{tf_i(w_j) - 1 + \mu \nu_c(w_j)}{|\tilde{d}_i| - 1 + \mu} \right) \]

Find appropriate \( \mu \)

\[ \mu^* = \arg \max_{\mu} L_{-1}(\mu, C) \]

Dirichlet Smoothing Hyper-Parameter

- What type of document/collection would get large \( \mu \)?
  - Most documents use similar vocabulary and wording pattern as the whole collection
- What type of document/collection would get small \( \mu \)?
  - Most documents use different vocabulary and wording pattern than the whole collection
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 $P(\text{Lung Cancer}|\text{Smoke})$

Shrinkage

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

**JM Smoothing:**

$$p_i(w_k) = \lambda \frac{tf_i(w_k)}{d_i} + (1-\lambda)p_c(w_k)$$
Shrinkage

- Relationship between JM Smoothing and Dirichlet Smoothing

\[ p_i(w_k) = \frac{tf_i(w_k) + \mu p_c(w_k)}{|d_i| + \mu} \]

\[ = \frac{1}{|d_i| + \mu} (tf_i(w_k) + \mu p_c(w_k)) \]

\[ = \frac{1}{|d_i| + \mu} \left( \frac{|d_i| tf_i(w_k)}{|d_i|} + \mu p_c(w_k) \right) = \frac{|d_i|}{|d_i| + \mu} tf_i(w_k) + \frac{\mu}{|d_i| + \mu} p_c(w_k) \]

**JM Smoothing:**

\[ p_i(w_k) = \lambda \frac{tf_i(w_k)}{|d_i|} + (1 - \lambda) p_c(w_k) \]

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

\[ KL(p \parallel q) = \sum_x p(x) \log \left( \frac{p(x)}{q(x)} \right) \]

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

How to prove it?
Model Based Feedback

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

\[
Sim(\tilde{q}, \tilde{d}_i) = -KL(\tilde{q} \parallel \tilde{d}_i) \\
= -\sum_w q(w) \log \left( \frac{q(w)}{p_i(w)} \right) \\
= \sum_w q(w) \log (p_i(w)) - \sum_w q(w) \log (q(w))
\]

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

Model Based Feedback

- Retrieval results
- Estimate the generation probability of Pr(\(q \mid \tilde{d}_i\))
- Language Model for \(\tilde{d}_i\)
- Estimating language model
- Estimating query language model
- Language Model for \(q\)
- Language Model for \(\tilde{d}_i\)
- Calculating KL Divergence
  \(KL(\tilde{q} \parallel \tilde{d}_i)\)

44

45
Model Based Feedback

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

For each word in feedback document(s), given $\lambda$

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

$$= \arg \max_{q_F} \sum_{j=1}^{n} \left( \log(\lambda q_F(w_j) + (1-\lambda) p_C(w_j)) \right)$$
Model Based Feedback: Estimate $\widehat{q}_F$

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

**Background Model**

$\pi_C(w | C)$

- sport 0.0001
- basketball 0.00005
- ...

**Unknown query topic**

$p(w|\theta_F) =$?

- sport =?
- basketball =?
- game =?
- player =?
- ...

"Basketball"

1- $\lambda = 0.8$

**Feedback Documents**

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

**MLE Estimator**

Step 1: estimate hidden variable based current on model parameter (Expectation)

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

$$= \frac{\lambda q_F^{(i)}(w_i)}{\lambda q_F^{(i)}(w_i) + (1-\lambda) \pi_C(w_i | C)}$$

E-step

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

Step 2: Update model parameters based on the guess in step 1 (Maximization)

$$q_F^{(i+1)}(w_i | \theta_F) = \frac{c(w_i,F) p(z_i = 1 | w_i)}{\sum_j c(w_j,F) p(z_j = 1 | w_j)}$$

M-Step

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

- For each word, the hidden variable $Z_i = \{1 \text{ (feedback)}, 0 \text{ (background)}\}$

  - Step 1: estimate hidden variable based current on model parameter (Expectation)
  - Step 2: Update model parameters based on the guess in step 1 (Maximization)
Model Based Feedback: Estimate $\vec{q}_F$

- Expectation-Maximization (EM) algorithm
  
  Step 0: Initialize values of $q_F^0$

  Step 1: (Expectation) $p(z_i = 1 \mid w_i) = \frac{\lambda q_F^{(i)}(w_i)}{\lambda q_F^{(i)}(w_i) + (1 - \lambda) p_c(w_i \mid C)}$

  Step 2: (Maximization) $q_F^{(i+1)}(w_i \mid \theta_F) = \frac{c(w_i, F) p(z_i = 1 \mid w_i)}{\sum_j c(w_j, F) p(z_j = 1 \mid w_j)}$

  Give $\lambda = 0.5$

| Word   | #C(F,w) | pC(w) | Initial $q_F(w)$ | Iteration 1 p(e=1|w) | Iteration 1 $q_F(w)$ | Iteration 2 p(e=1|w) | Iteration 2 $q_F(w)$ |
|--------|---------|-------|------------------|------------------------|----------------------|----------------------|----------------------|
| the    | 4       | 0.5   | 0.25             | 0.33                   | 0.21                 | 0.30                 | 0.19                 |
| good   | 2       | 0.4   | 0.25             | 0.58                   | 0.12                 | 0.25                 | 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 $\lambda$
  
  - If $\lambda$ 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 $\lambda$ 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