Notes on Language Models

Liangjie Hong

May 20, 2010

1 Query Likelihood Language Models

The basic idea behind Query Likelihood Language Models (QLLM) is that a query is a sample drawn from a language model. In other words, we want to compute the likelihood, that given a document language model θ_D, how likely the posed query Q would be used. Formally, this can be expressed as $P(Q|\theta_D)$. Two questions will immediately arise following this formulation. (1) How to choose the model to represent θ_D? (2) How to estimate θ_D?

1.1 Multinomial Language Model

One popular choice for θ_D is multinomial distribution. The original multinomial distribution is

$$P(X_1 = x_1, X_2 = x_2, X_3 = x_3, ..., X_n = x_n) = \frac{n!}{x_1!x_2!...x_n!} p_1^{x_1} p_2^{x_2} ... p_n^{x_n}$$

In Language Modeling, we usually ignore the coefficient and therefore we obtain *unigram language model* so that the order of text sequence is not important. Use multinomial distribution to model θ_D, we obtain

$$\begin{aligned} P(Q|\theta_D) &= \prod_{w \in Q} P(w|\theta_D) & (1) \\ &= \prod_{w \in V} P(w|\theta_D)^{c(w,Q)} & (2) \end{aligned}$$

where $c(w, Q)$ is the number of times that term w appearing in query Q. Now, the problem becomes to estimate $P(w|\theta_D)$. In theory, we need to estimate it for all the terms in our vocabulary. However, since $c(w, Q)$ could be 0 (meaning that term w does not show up in the query), we only care about the terms in the query.

The key point here is that **we do not know** θ_D! How can we calculate $P(w|\theta_D)$ for the terms in the query when we really do not have a model in hand? One way is to use document D as a sample to estimate θ_D. Therefore, essentially, we choose the model θ_D such that $\hat{\theta_D} = \arg\max_\theta P(D|\theta)$.

One simple way to estimate $P(D|\theta)$ is through Maximum Likelihood Estimator (MLE). Now, let us to derive ML for Multinomial Language Model. First, we usually work on log-likelihood rather than the product of probabilities just to avoid very small numbers:

$$\log P(D|\theta) = \sum_{w \in V} c(w, D) \log P(w|\theta)$$

Then, use Lagrange Multipliers, we obtain:

$$L = \log P(D|\theta) + \lambda(1 - \sum_{w \in V} P(w|\theta))$$

Take all derivatives respect to $P(w|\theta)$ and λ:

$$\frac{\partial L}{\partial P(w|\theta)} = \frac{c(w,D)}{P(w|\theta)} - \lambda = 0$$
$$\frac{\partial L}{\partial \lambda} = 1 - \sum_{w \in V} P(w|\theta) = 0$$

From the first equation, we can get $P(w|\theta) = \frac{c(w,D)}{\lambda}$ and also $\sum_{w \in V} c(w,D) = \lambda \sum_{w \in V} P(w|\theta) = \lambda$. Therefore,

$$P(w|\theta) = \frac{c(w,D)}{\sum_{w \in V} c(w,D)} = \frac{c(w,D)}{|D|} \tag{3}$$

Note, here, we have two important assumptions:

- A query Q is sampled from a underlying language model θ_D. This point is usually stated explicitly in literature.
- The document D is used to estimate θ_D and therefore, we implicitly assume D is also sampled from θ_D. This is not clear in most related work.

Also, although we are using ML estimator to get the best guess of θ_D, we actually do not need θ_D due to Equation 1. All we need is *conditional probability* $P(w|\theta_D)$. So, the multinomial distribution is *never* explicitly calculated.

Using Equation 3, we can easily calcuate Equation 1 by:

$$\log P(Q|\theta_D) = \sum_{w \in Q} c(w,Q) \log P(w|\theta_D)$$

where $c(w,Q)$ is usually ignored in the literature.

1.1.1 Multinomial Language Model with Dirichlet Prior

Since $P(D|\theta)$ is a multinomial distribution, we can impose a prior distribution on it and obtain a Maximum A Prior (MAP) estimator rather than a MLE estimator. Usually, we use a Dirichlet distribution.

$$P(D|\theta) = \frac{\Gamma(|D| + \sum_i^V \alpha_i)}{\prod_i^V \Gamma(c(w_i,D) + \alpha_i)} \prod_i^V \theta_i^{c(w_i,D)+\alpha_i-1}$$

If we denote $\Lambda = \frac{\Gamma(|D|+\sum_i^V \alpha_i)}{\prod_i^V \Gamma(c(w_i,D)+\alpha_i)}$ and apply Lagrange Multipliers, we can have:

$$\log P(D|\theta) = \log \Lambda + \sum_i^V (c(w_i,D) + \alpha_i - 1) \log \theta_i + \lambda(1 - \sum_i^V \theta_i)$$

Taking derivatives respect to θ_i, we have:

$$\frac{\partial \log P(D|\theta)}{\partial \theta_i} = \frac{c(w_i,D) + \alpha_i - 1}{\theta_i} - \lambda = 0$$

Therefore, we can derive that $\sum_i^V \lambda\theta_i = \sum_i^V c(w_i,D) + \sum_i^V \alpha_i - |V|$ and conclude that:

$$\hat{\theta}_i = \frac{c(w_i,D) + \alpha_i - 1}{|D| + \sum_i^V \alpha_i - |V|}$$

When $\alpha_i = 1$, MAP estimator goes back to MLE estimator. If $\alpha_i = 2$, the equation gives Laplace smoothing and if $\alpha_i = \mu\frac{c(w_i,C)}{|C|} + 1$ gives the popular Dirichlet Prior smoothing.

1.1.2 Bayesian Multinomial Language Model

1.2 Multiple Bernoulli Language Model

When Language Model was firstly introduced, Multiple Bernoulli Language Model was used. Although it is less explored later in the literature mainly due to its weaker assumptions (e.g. the occurrences of different words are independent, compared to Multinomials, every occurrences of a word, including the multiple occurrences of the same word, is independent). We define a binary random variable $X_i \in \{0, 1\}$ for each term w_i to indicate whether word w_i is present or absent in the query.

According to this definition, we can express $P(Q|\theta_D)$ as :

$$P(Q|\theta_D) = \prod_{w_i \in Q} P(X_i = 1|\theta_D) \prod_{w_i \notin Q} P(X_i = 0|\theta_D)$$

Therefore, the problem is reduced to estimate $P(X_i = 1|\theta_D)$ and $P(X_i = 0|\theta_D)$.

Similar as Multinomial Language Model, we do not know θ_D and what we can do is to make the best guess through ML estimator. For a document D, the log-likelihood in terms of parameter $P(X_i = 1|\theta_D)$ and $P(X_i = 0|\theta_D)$ is:

$$\begin{aligned}
L = \log \prod_{i}^{N} P(X_i|\theta_D) &= \sum_{i}^{N} \log P(X_i|\theta_D) \\
&= N_k \log P(X_i = 1|\theta_D) + \bar{N_k} \log P(X_i = 0|\theta_D) \\
&= N_k \log P(X_i = 1|\theta_D) + \bar{N_k} \log(1 - P(X_i = 1|\theta_D))
\end{aligned}$$

where N is the number of terms in D, N_k is the number of times the word w_i showing up in D and $\bar{N_k}$ is the number of times of absence. Take derivatives respect to $P(X_i = 1|\theta_D)$:

$$\begin{aligned}
\frac{\partial L}{\partial P(X_i = 1|\theta_D)} &= \frac{N_k}{P(X_i = 1|\theta_D)} - \frac{\bar{N_k}}{1 - P(X_i = 1|\theta_D)} = 0 \\
\rightarrow P(X_i = 1|\theta_D) &= \frac{N_k}{N_k + \bar{N_k}} = \frac{N_k}{N}
\end{aligned}$$

Therefore, the ML estimator of Multiple Bernoulli Language Model is exactly the same as the ML estimator Multinomial Language Model.

1.2.1 Multiple Bernoulli Language Model with Beta Prior

Similar to Multinomial Language Model, we can impose a prior distribution on the Multiple Bernoulli Language Model. Here, we use Beta distribution and have the Language Model with the following form:

$$P(D|\theta) = \prod_{i}^{V} \frac{1}{B_i} \theta_i^{(N_i + \alpha_i - 1)} (1 - \theta_i)^{(N - N_i + \beta_i - 1)}$$

where B_i is the beta function. Again, we can obtain the solution in a closed form:

$$\hat{\theta}_i = \frac{N_i + \alpha_i - 1}{|D| + \alpha_i + \beta_i - 2}$$

A popular choice for hyper-parameters α_i and β_i is as follows:

$$\alpha_i = \mu \frac{N_i}{|C|} + 1 \quad \text{and} \quad \beta_i = \frac{|C|}{N_i} + \mu(1 - \frac{N_i}{|C|}) - 1$$

1.3 Poisson Language Model

A less explored formalism is to use Poisson distribution to model the queries and documents. One obvious advantage of Poisson distribution is to model the frequency of events. The probability density function of Poisson distribution is as follows:

$$f(x; \lambda t) = \frac{(\lambda t)^n e^{-\lambda t}}{n!} \tag{4}$$

where n is the number of occurrences of an event, λ is the rate of the occurrence and t is the time period of observation. To use Poisson distribution to model the documents, we attach **each** term with an independent Poisson process. In other words, we consider the frequency counts of the n unique terms in the corpus as n different types of events, sampled from n independent homogeneous Poisson processes.

Suppose t is the time period during which the author composed the text. With a homogeneous Poisson process, the frequency count of each event, i.e., the number of occurrences of w_i, follows a Poisson distribution with associated parameter $\lambda_i t$, where λ_i is a rate parameter characterizing the expected number of w_i in a unit time. In the literature, we usually set t to the length of the text either document or the query, i.e., $t = |\mathbf{w}|$. Therefore, for term w_i, the frequency of w_i in text $\mathbf{w}$ is:

$$P(c(w_i, \mathbf{w})|\Lambda_{\mathbf{w}}) = \frac{(\lambda_i|\mathbf{w}|)^{c(w_i,\mathbf{w})} e^{-\lambda_i|\mathbf{w}|}}{c(w_i, \mathbf{w})!}$$

Notice, even if a term does not appear in the document, we still can assign probability to it. Therefore, we explicitly model the absence of terms. According to the above equation, the likelihood of $\mathbf{w}$ to be generated from such Poisson Processes can be written as:

$$P(\mathbf{w}|\Lambda) = \prod_{i=1}^{n} \frac{(\lambda_i|\mathbf{w}|)^{c(w_i,\mathbf{w})} e^{-\lambda_i|\mathbf{w}|}}{c(w_i, \mathbf{w})!}$$

where $|\mathbf{w}| = \sum_{i=1}^{n} c(w_i, \mathbf{w})$. Note, the parameter λ_i is for each term w_i and therefore it is across documents. In order to estimate these parameters, we need the likelihood for the whole corpus as follows:

$$L(\Lambda) = \prod_{k=1}^{m} \prod_{i=1}^{n} \frac{(\lambda_i|\mathbf{w_k}|)^{c(w_i,\mathbf{w_k})} e^{-\lambda_i|\mathbf{w_k}|}}{c(w_i, \mathbf{w_k})!}$$

The MLE estimator can be obtained as follows:

$$\hat{\lambda_i} = \frac{\sum_{k=1}^{m} c(w_i, \mathbf{w_k})}{\sum_{k=1}^{m} |\mathbf{w_k}|} \tag{5}$$

Similarly, once we have the estimated model, the likelihood that a query q is generated from Λ_d can be written as:

$$p(q|d) = \prod_{w \in V} p(c(w, q)|\Lambda_d)$$

1.4 Discussion

For Query Likelihood Language Models, two major steps are usualy performed: First, we have to use the observation d to construct our estimate of the underlying document language model θ_D. Second, we can compute the probability $P(Q|\theta_D)$ of observing Q as a random sample from θ_D.

2 Smoothing in Language Models

3 Probabilistic Distance Language Models