CS 674/INFO 630: Advanced Language Technologies **Fall 2007**

Lecture 7 — September 18

Lecturer: Lillian Lee *Scribes: Nikos Karampatziakis & Ainur Yessenalina*

1 Recap of the Previous Lecture

In Appendix A we show the correct derivation of the scoring function for the binary-attribute variable case using the Croft-Harper assumptions. Last time we discussed two other derivations that also lead to the appearance of "IDF" in the scoring function (for binary attributes). So what was the point of this discussion?

(a) A metapoint is to ask the questions: Why do people try to come up with new derivations for IDF? Isn't one derivation sufficient? First of all, because IDF is important. Secondly, because it is possible to object to all three derivations.

(b) To show that the estimation of parameters is non-obvious even for simple models.

(c) Finally, to remind us to challenge old ideas.

2 Incorporating Term Frequencies

Apart from IDF, term frequencies are also important and we would like to incorporate them into our scoring function. From now on, we will treat A_j as a random variable that denotes the number of occurrences of term j in a document. So, what should $P(A_j = a)$ and $P(A_j = a|R_q = y)$ be? In other words, how do we model the distributions of these random variables? Here we have two options: continuous and discrete distributions. Picking a discrete distribution seems more natural, because we are dealing with word counts. Now we have to pick the kind of discrete distribution. Some natural options include:

(a) Uniform distribution. We have to rule out this distribution because we do not know the upper-limit of the word counts.

(b) Multinomial distribution. As Casella and Berger note [2], the multinomial distribution is a model for an experiment which consists of n independent trials and each trial results in one of k distinct possible outcomes. A simple special case is the binomial, discussed next, in which we consider each term separately of the others. (Recall that the scoring function has already been decomposed into individual-term-based components). The multinomial considers the counts for all k types "jointly".

(c) Binomial distribution. The underlying experiment consists of n independent Bernoulli trials. Each trial has two possible outcomes: "success" (the word appears in the i-th position of the text) and "failure" (otherwise). The binomial distribution is a distribution over the number of successes in n trials and it has two parameters: n (number of trials) and p (probability of success of each trial).

(d) Poisson distribution. As Casella and Berger note [2], suppose we are modeling a phenomenon in which we are waiting for an occurrence of an event. Then the number of occurrences in a given time interval can sometimes be modeled by the Poisson distribution. The Poisson distribution has one parameter λ, the arrival rate, and a meta-parameter, the time interval or the length of the document in our case. We can model the arrival rate of a term by stating that we expect this term to appear, say, three times within every document (of the "unit-length" type).

Now we have to choose between the binomial and the Poisson distribution. We want to have as few parameters as possible. Both distributions have one parameter that needs to be estimated (p for the binomial and λ for the Poisson) so we cannot decide based on this information. However, the binomial is seemingly analytically harder to work with since it involves a sum and the calculation of binomial coefficients. It is also a well-known fact that the Poisson(λ) distribution approximates the binomial(n, p) distribution as $n \to \infty$ and $p = \frac{\lambda}{n}$. Since we do have quite big documents and the occurrence of each word is a small probability event (except for words such as "the", "of" etc.), the Poisson approximation to the binomial is valid. Thus, we will choose the Poisson distribution for our models.

3 A Straightforward Approach

For now we assume that we have documents of the same length, an assumption that we will try to correct later. Suppose that for each term j we define ρ_j to be the expected number of occurrences of term j in relevant documents and γ_j to be the expected number of occurrences of term j in general documents. For a term j that appears in the query ($q[j] > 0$) we expect that $\rho_j > \gamma_j$ because a term that appears in the query is more likely to appear in relevant documents than in general documents. Then assuming that $p(A_j|R_q = y) \sim$ Poisson(ρ_j) and $p(A_j) \sim$ Poisson(γ_j) we can write

$$p(A_j = d[j]|R_q = y) = \frac{\rho_j^{d[j]}}{d[j]!}e^{-\rho_j}, \quad p(A_j = 0|R_q = y) = e^{-\rho_j},$$

$$p(A_j = d[j]) = \frac{\gamma_j^{d[j]}}{d[j]!}e^{-\gamma_j}, \quad p(A_j = 0) = e^{-\gamma_j},$$

and the scoring function becomes

$$\prod_{j:q[j],d[j]>0} \frac{\frac{\rho_j^{d[j]}}{d[j]!}e^{-\rho_j}}{\frac{\gamma_j^{d[j]}}{d[j]!}e^{-\gamma_j}} \times \frac{e^{-\gamma_j}}{e^{-\rho_j}} = \prod_{j:q[j],d[j]>0} \left(\frac{\rho_j}{\gamma_j}\right)^{d[j]} \overset{\text{rank}}{=} \sum_{j:q[j],d[j]>0} d[j] \log\left(\frac{\rho_j}{\gamma_j}\right).$$

We see the term frequency $d[j]$ appearing in the sum. How about the $\log\left(\frac{\rho_j}{\gamma_j}\right)$ quantity? Does it look like an IDF factor? We don't have any data to estimate ρ_j and even if we assume it is constant, $\frac{1}{\gamma_j}$ still does not look like an IDF factor, unless we decide to estimate γ_j by the number of documents that contain term j which is a very rough estimate. We will leave the discussion of this simple model here and consider a different perspective.

4 A Generative Perspective

So far we have only considered the question "given that some documents are relevant, what kind of characteristics do these documents have?". This is called a discriminative approach. We can have a generative approach if we think about the process that generates the documents in the corpus. Reasoning about this process may help us do inference better. When we are looking at the term frequencies we are assuming that there is a correlation between term counts and relevance. However, term counts are only proxies for relevance. For example, when we do length normalization it is really because term frequencies are not the perfect representation for how relevant the document is. So instead of looking at term frequencies we can think what relevance really is. *Relevance is about topics.* A document is relevant if it is about the same topic as the query, not necessarily if its term frequencies match the query. Moreover, the way an author writes a document hinges on the topic the document is about. A document about politics will generally use different words from a document about pets. So, the authors decide what words will appear in their documents based on what they are actually talking about.

We will follow the work of [1] and [3] and introduce a Poisson based "topic model" (which is a very simplified version of the types of topic models used nowadays in machine learning). We introduce the topic of the document as a primary object we will reason about and more specifically we introduce binary random variables T_j indicating whether or not the document is on the topic of term j. For now we assume again that the documents are of the same length, an assumption that we will try to correct later. We can write the probability distribution of A_j by marginalizing the joint distribution of A_j and T_j.

$$P(A_j = a) = \sum_{t \in \{y,n\}} P(A_j = a | T_j = t) P(T_j = t)$$

where we assume

$$P(A_j = a | T_j = y) \sim \text{Poisson}(\tau_j)$$
$$P(A_j = a | T_j = n) \sim \text{Poisson}(\mu_j)$$

and we expect $\tau_j > \mu_j$ because term j is more likely to appear in documents about its topic than in documents which are not on its topic. Using this mixture of Poisson distributions we can rewrite $P(A_j = d[j] | R_q = y)$ as

$$\sum_{t \in \{y,n\}} P(A_j = d[j], T_j = t | R_q = y) =$$

$$\sum_{t\in\{y,n\}} P(A_j = d[j]|T_j = t, R_q = y)P(T_j = t|R_q = y).$$

The A_j can be assumed conditionally independent of R_q given T_j because the event that term j appears $d[j]$ times in the document "happened" before the query was issued and when it happened it was based on whether the document was on the topic of term j or not. Hence we can write

$$P(A_j = d[j]|R_q = y) = \sum_{t\in\{y,n\}} P(A_j = d[j]|T_j = t)P(T_j = t|R_q = y)$$

Defining the probability that the document is on the topic of term j given that it is relevant as $tr_j = P(T_j = y|R_q = y)$ and the probability that the document is on the topic of term j in general as $tg_j = P(T_j = y)$ we can get the following scoring function

$$\prod_{j:q[j],d[j]>0} \frac{tr_j + (1-tr_j)\left(\frac{\mu_j}{\tau_j}\right)^{d[j]} e^{\tau_j-\mu_j}}{tg_j + (1-tg_j)\left(\frac{\mu_j}{\tau_j}\right)^{d[j]} e^{\tau_j-\mu_j}} \times \frac{tg_j e^{\mu_j-\tau_j} + (1-tg_j)}{tr_j e^{\mu_j-\tau_j} + (1-tr_j)}$$

by dividing the numerator and the denominator of the original scoring function by $\tau_j^{d[j]} e^{-\tau_j-\mu_j}$. This expression has many unknowns but we can still study it as a function of the term frequency $d[j]$. Stay tuned ...

5 Finger Exercises

1. In this exercise we will investigate the scoring functions we get by plugging in different distributions in the place of Poisson using the approach in section 3. Suppose we can use continuous distributions to model the term frequencies and we also don't have to worry about continuity corrections. Let us assume that the frequency of term j is distributed with some distribution f whose unknown mean is μ_j for relevant documents and ν_j for general documents.[1] What is the scoring function in the following cases?

   (a) f is the normal distribution with variance σ_j^2.

   (b) f is the double-exponential[2] (Laplace) distribution with variance $2\sigma_j^2$.

   (c) f is the exponential distribution. Assume $P(A_j = 0) = \lim_{x\to 0^+} f(x)$

2. Let's view the document as a vector of term frequencies and try an approach motivated by the fact that we didn't use the binomial distribution for our models. What happens

[1] Caveat: Some sort of "extrinsic" length normalization is needed, similar to the Poisson case, because a "mean of five term occurrences" should presumably be relative to the length of the document.

[2] Note that the normal and double exponential distributions may allocate a significant amount of probability mass to *negative* counts and thus may not provide a realistic generative model for term occurrences.

if we model the number of occurrences of all the terms in a document as a multinomial distribution and we rank documents according to

$$\frac{P(\vec{A} = \vec{d} | R_q = y)}{P(\vec{A} = \vec{d})} \tag{1}$$

where $\vec{d}$ is the vector of all term frequencies? Assume both the numerator and the denominator are multinomially distributed but the probabilities of the outcomes are different. For each term j we will have a probability θ_{jg} of occurrence in a general document and a probability θ_{jr} of occurrence in a relevant document. Assume that:

- For the terms that don't appear in the query, $\theta_{jr} = \beta\theta_{jg}$
- For the terms in the query, $\theta_{jr} = \theta_{jg} + \alpha$
- The sum of the term frequencies in a document for the terms that don't appear in the query is δ. This is similar to assuming that the documents have equal lengths.

and α, β and δ are constants independent of the document and the term.[3] What estimate for the probability of a term in a general document can we use in order to get an IDF in the final scoring function? Is it a good estimate?

3. Let's assume that there are k topics $t_1, t_2 \ldots, t_k$ in the corpus and each term j appears in topic t_i with probability θ_{ji}. Define an appropriate embedding of terms into $\mathbb{R}^n$ and estimate the θ_{ji} using, for example, the EM algorithm to learn a mixture of gaussians (or your favorite fuzzy clustering algorithm). In other words, terms are points in $\mathbb{R}^n$ and topics are clusters of terms. What would be an appropriate generative model for documents in this setting?

6 Solutions

1. Before we begin our derivations, we will discuss the distributions that we use to gain some intuition on what kind of events they try to model. Figure 1 shows a plot of the three distributions with mean 10 and where applicable $\sigma = 1$. We see that the normal and double exponential have similar shapes with the mode equal to the mean and most of their probability mass around the mean although the double exponential has heavier tails than the normal. On the other hand the exponential distribution is neither symmetric nor concentrated around its mean. Its mean and variance depend only on one parameter and it allocates probability mass only to positive numbers. The normal distribution is mostly used to model measurements of things that happen because many small additive effects are contributing to them [2], the double exponential is practically

[3] We are willing to assume everything to make the analysis tractable. . .

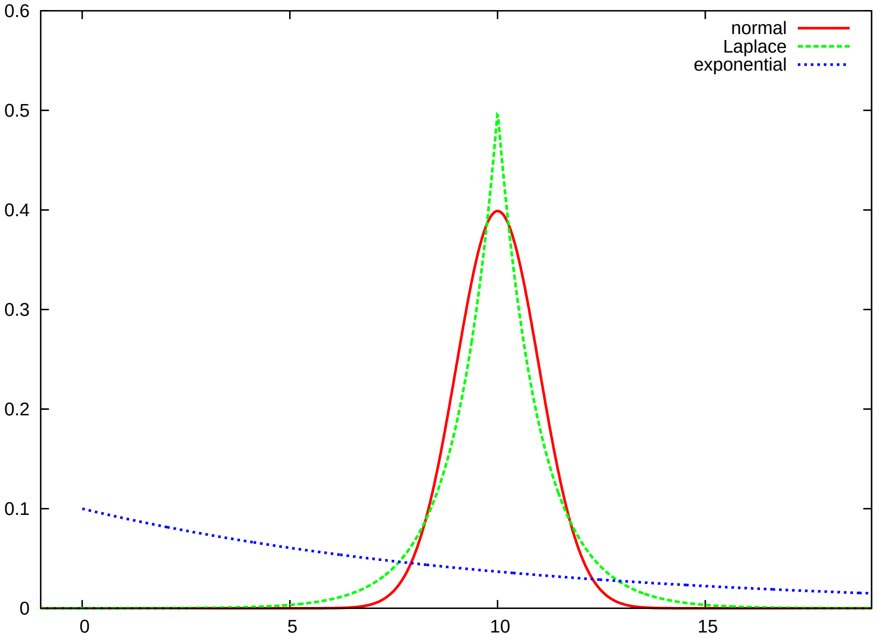

Figure 1: The normal(10,1), Laplace(10,1) and exponential(10) distributions.

used as an alternative to the normal[4] and the exponential distribution is used to model waiting times between events following a Poisson distribution. Our conclusion is that none of these distributions are actually good models for the term counts but they may still be useful.

(a) Recall the Normal distribution $p(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$. The scoring function becomes

$$\prod_{j:q[j],d[j]>0} \frac{\frac{1}{\sigma_j\sqrt{2\pi}} e^{-\frac{(d[j]-\mu_j)^2}{2\sigma_j^2}}}{\frac{1}{\sigma_j\sqrt{2\pi}} e^{-\frac{(d[j]-\nu_j)^2}{2\sigma_j^2}}} \times \frac{\frac{1}{\sigma_j\sqrt{2\pi}} e^{-\frac{\nu_j^2}{2\sigma_j^2}}}{\frac{1}{\sigma_j\sqrt{2\pi}} e^{-\frac{\mu_j^2}{2\sigma_j^2}}} =$$

$$\prod_{j:q[j],d[j]>0} e^{-\frac{(d[j]-\mu_j)^2+\nu_j^2}{2\sigma_j^2}+\frac{(d[j]-\nu_j)^2+\mu_j^2}{2\sigma_j^2}} =$$

$$\prod_{j:q[j],d[j]>0} e^{\frac{d[j](\mu_j-\nu_j)}{\sigma_j^2}} \overset{\text{rank}}{=} \sum_{j:q[j],d[j]>0} d[j]\frac{(\mu_j-\nu_j)}{\sigma_j^2}.$$

[4] For example, a Bayesian interpretation of ridge regression is that we do linear regression with a normal prior on the coefficients while a Bayesian interpretation of the Lasso is that we do linear regression with a Laplace prior on the coefficients [4].

We see that the term frequency comes up in the formula, but not anything resembling the IDF. Note that the Poisson model naturally yielded an IDF factor. Each term j such that $q[j] > 0$, $d[j] > 0$ is weighted by the difference $\mu_j - \nu_j$ which we expect to be positive. The bigger this difference, the more weight term j will get.

(b) For the Laplace distribution $p(x) = \frac{1}{2\sigma} e^{-\frac{|x-\mu|}{\sigma}}$, the scoring function becomes

$$\prod_{j:q[j],d[j]>0} \frac{\frac{1}{2\sigma_j} e^{-\frac{|d[j]-\mu_j|}{\sigma_j}}}{\frac{1}{2\sigma_j} e^{-\frac{|d[j]-\nu_j|}{\sigma_j}}} \times \frac{\frac{1}{2\sigma_j} e^{-\frac{\nu_j}{\sigma_j}}}{\frac{1}{2\sigma_j} e^{-\frac{\mu_j}{\sigma_j}}} =$$

$$\prod_{j:q[j],d[j]>0} e^{\frac{-|d[j]-\mu_j|+|d[j]-\nu_j|-\nu_j+\mu_j}{\sigma_j}} \overset{\text{rank}}{=}$$

$$\sum_{j:q[j],d[j]>0} \frac{-|d[j]-\mu_j|+|d[j]-\nu_j|-\nu_j+\mu_j}{\sigma_j}$$

We have three cases

$$\frac{-|d[j]-\mu_j|+|d[j]-\nu_j|-\nu_j+\mu_j}{\sigma_j} = \begin{cases} 0 & \text{if } d[j] < \nu_j < \mu_j \\ \frac{2(d[j]-\nu_j)}{\sigma_j} & \text{if } \nu_j < d[j] < \mu_j \\ \frac{2(\mu_j-\nu_j)}{\sigma_j} & \text{if } \nu_j < \mu_j < d[j] \end{cases}$$

This scoring function is funny because when the term frequency is less than what we expect even for general documents it doesn't contribute to the scoring function. When the term frequency is somewhere between what we expect for relevant and general documents, it contributes to the score by the amount of the difference from the mean of general documents. When the term frequency is more than what we expect to find in relevant documents the contribution is a constant independent of $d[j]$ (resistance to spam?). Also notice that μ_j appears explicitly in only one case which makes it easy to provide approximations to the scoring function. For example, if we assume that we always have $d[j] < \mu_j$ then the scoring function can be easily evaluated.

(c) Finally the scoring function for the exponential distribution $p(x) = \frac{1}{\mu} e^{-\frac{x}{\mu}}, x > 0$ becomes

$$\prod_{j:q[j],d[j]>0} \frac{\frac{1}{\mu_j} e^{-\frac{d[j]}{\mu_j}}}{\frac{1}{\nu_j} e^{-\frac{d[j]}{\nu_j}}} \times \frac{\frac{1}{\nu_j} \lim_{x\to 0^+} e^{-\frac{x}{\nu_j}}}{\frac{1}{\mu_j} \lim_{x\to 0^+} e^{-\frac{x}{\mu_j}}} = \prod_{j:q[j],d[j]>0} e^{\frac{d[j]}{\nu_j} - \frac{d[j]}{\mu_j}} \overset{\text{rank}}{=}$$

$$\sum_{j:q[j],d[j]>0} d[j] \frac{\mu_j - \nu_j}{\mu_j \nu_j}$$

Despite the differences between the normal and the exponential, this scoring function looks like the one from (a). Here, we divide by the geometric mean $\mu_j \nu_j$ of the variances (μ_j^2 and ν_j^2) instead of the single variance that we had in (a).

2. The probability distribution of the multinomial distribution is

$$P(\vec{A}=\vec{d}) = \frac{(\sum_{j=1}^{m} d[j])!}{\prod_{j=1}^{m} d[j]!}\prod_{j=1}^{m}\theta_j^{d[j]}$$

   where θ_j is the probability of outcome j (term j in our case). Plugging in the multinomial in (1) we get

$$\frac{\frac{(\sum_{j=1}^{m} d[j])!}{\prod_{j=1}^{m} d[j]!}\prod_{j=1}^{m}\theta_{jr}^{d[j]}}{\frac{(\sum_{j=1}^{m} d[j])!}{\prod_{j=1}^{m} d[j]!}\prod_{j=1}^{m}\theta_{jg}^{d[j]}} = \prod_{j=1}^{m}\left(\frac{\theta_{jr}}{\theta_{jg}}\right)^{d[j]} = \prod_{j:d[j]>0}\left(\frac{\theta_{jr}}{\theta_{jg}}\right)^{d[j]} =$$

$$\prod_{j:d[j]>0,q[j]=0}\left(\frac{\theta_{jr}}{\theta_{jg}}\right)^{d[j]}\prod_{j:d[j]>0,q[j]>0}\left(\frac{\theta_{jr}}{\theta_{jg}}\right)^{d[j]} =$$

$$\beta^{\sum_{j:q[j]=0} d[j]}\prod_{j:d[j]>0,q[j]>0}\left(\frac{\theta_{jr}}{\theta_{jg}}\right)^{d[j]} = \beta^{\delta}\prod_{j:d[j]>0,q[j]>0}\left(\frac{\theta_{jr}}{\theta_{jg}}\right)^{d[j]} \overset{\text{rank}}{=}$$

$$\sum_{j:d[j]>0,q[j]>0} d[j]\log\left(\frac{\theta_{jr}}{\theta_{jg}}\right) = \sum_{j:d[j]>0,q[j]>0} d[j]\log\left(1+\frac{\alpha}{\theta_{jg}}\right).$$

   Now if we use the estimate

$$\hat{\theta}_{jg} = \frac{n_j}{N}$$

   where n_j is the number of documents that contain term j and N is the number of documents in the corpus then the scoring function will have an IDF

$$\sum_{j:d[j]>0,q[j]>0} d[j]\log\left(1+\frac{\alpha N}{n_j}\right).$$

   However this estimate of θ_{jg} is very rough compared to something like:

$$\hat{\theta}_{jg} = \frac{1}{|C|}\sum_{d\in C}\frac{TF_j(d)}{\sum_k TF_k(d)}$$

   where the sum in the denominator is over all terms in document d and C is the corpus.

A Correction for the Previous Lecture

Let us recall the scoring function from the previous lecture:

$$\prod_{\substack{j:q[j]=1\\ d[j]=1}}\frac{P(A_j=d[j]|R_q=y)}{P(A_j=d[j])}\times\frac{P(A_j=0)}{P(A_j=0|R_q=y)} \tag{2}$$

Last time, for the binary-attribute variable case we said that the Croft-Harper assumption was to set for all attributes A_j that are shared between the query and the document

$$P(A_j = 1 | R_q = y) = \alpha,$$

where α is a constant $\in [0, 1]$. However α is really a function of the document d, the query q and the attribute A_j since it is a constant only for the *shared* attributes, so we should have written it as $\alpha_{d,q,j}$. To see why this gives rise to a potentially different scoring function, remember that Croft and Harper set

$$P(A_j = 1) = \frac{n_j}{N}, \quad P(A_j = 0) = 1 - \frac{n_j}{N}$$

where n_j is the number of documents that have attribute A_j and N is the total number of documents, and notice that taking the log of (2) leads to

$$\sum_{\substack{j:q[j]=1 \\ d[j]=1}} \log \left(\left(\frac{N}{n_j} - 1 \right) \frac{\alpha_{d,q,j}}{1 - \alpha_{d,q,j}} \right) = \sum_{\substack{j:q[j]=1 \\ d[j]=1}} \log \left(\frac{N}{n_j} - 1 \right) + \sum_{\substack{j:q[j]=1 \\ d[j]=1}} \log \frac{\alpha_{d,q,j}}{1 - \alpha_{d,q,j}}$$

so we cannot ignore the quantity $\frac{\alpha_{d,q,j}}{1-\alpha_{d,q,j}}$, like we did last time. But, if we make the additional assumption that $\alpha_{d,q,j} = \frac{1}{2}$ for the shared attributes, then we can drop the second sum and derive the scoring function that we got last time.

References

[1] Abraham Bookstein and D. R. Swanson. Probabilistic models for automatic indexing. Journal of the American Society for Information Science, 25:312–318 (1974).

[2] G. Casella and R. L. Berger. Statistical Inference, 2nd Edition.

[3] Stephen P. Harter. A probabilistic approach to automatic keyword indexing, part I: On the distribution of specialty words in a technical literature. Journal of the American Society for Information Science 26(4).

[4] T. Hastie, R. Tibshirani and J. Friedman. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer, 2001.