STAT 3503/8109 Lecture 8 Notes

Edoardo Airoldi Scribe: Srikar Katta*

Fall 2020: October 19, 2020

Introduction

To wrap up the section on models and algorithms, we will discuss proteomics and text models. This proteomics data is very popular in forensic sciences that can help estimate how much of a certain compound is present in a sample. These text models are widely popular in journalism, the tech space, and finance. And when people talk about text, there is a wide array of strategies to approach text models, but we will talk about utilizing frequencies of words for text.

Proteomics

Introduction

The first thing to understand when discussing a model for a new kind of data is what exactly are the instruments generating the data. In this proteomics model, we take the following steps. First, we collect a sample (e.g., dirt or DNA). Then, the sample goes into a vial with a solution. Then we utilize “fractionation by hydrophobicity,” a technique that allows biologists and scientists to separate compounds based on their attraction to water (hydrophobicity is literally “fear of water”). First, a sample is sent into a tube coated with water on the inside. Then, the vial with the sample of dirt or DNA is poured into the tube. Based on compounds’ attraction to water, these compounds will slide through at different rates. So for example, if you poured a mixture of oil and juice, because oil is completely hydrophobic (hates water), it will slide down the fastest, while the juice would slide down soon after because it is binding with the water as they are dragged down the column due to gravity. While a simplistic example, this is the general process that is occurring. Once these compounds slide down the tube, they are separated based on the window of time at which they arrive. Now, once these compounds arrive, they get ionized with protons (get shocked with some positive charge) and gets pushed into a mass spectrometer, a tool that has two magnets around which the compounds rotate around at different speeds based on how charged they are. The first magnet then selects the fastest rotating compounds and “passes them over” to the second magnet. Based on the speed of rotation of compounds in the second magnet, the machine will decide the “mass to charge ratio,” a way to describe what amino acids are actually in the compound. Now, based on the charge, we can construct a sequence of

*Please share any comments or suggestions with Srikar Katta at srikar@temple.edu amino acids and will allow biologists to identify what actually exists. Assume the sequence of amino acids is "AG, P, G, Q", then the mass spectrometer allows biologists to identify the relative ratio of each of these amino acids in the compound, which they can use to recognize that the compound contains Arsenic.

The Model

Let i be a protein, k be a peptide in the protein, and l be a charge state. And y_{ikl}^{obs} is the abundance of a certain peptide in a sequence of proteins that is observed. Now, we have a theoretical abundance of this peptide, y_{ikl}. We assume this is normally distributed with mean γ_{ik} and variance σ_i^2. Consider the indicator $R_{ikl} \sim Bernoulli(1 - \pi^{rnd})$ that is some random chance that we will never see this peptide in the sequence. And there is another probability, $I_{ikl} \sim Bernoulli(1 - g(y_{ikl}, x_{ik}); \eta)$, which is a way of saying, "If I see this peptide in the sequence a little early on, I will be less likely to see it later." And this makes sense based on how the mass spectrometer works: the first magnet will take the fastest spinning and most abundant compounds and send it to the second magnet. So if the peptide was not sent to the second magnet early on, the chances that it will be sent to the second magnet are going to be less and less as time goes on. And now, we define a new indicator for seeing arsenic $O_{ikl} = R_{ikl} I_{ikl}$, which allows us to combine whether we see some peptide in the sequence because it is abundant or due to random chance. Now, we define some variables a_i, γ_{ik}, s_{ik} that inform how abundant y_{ikl} is in the compound. And then based on some random chance R_{ikl} and the theoretical abundance of y_{ikl} in past samples I_{ikl}, we can create an indicator O_{ikl} of whether we see y_{ikl} in our data or not. Then, we can define a set $Y^{obs} = \{y_{ikl}; O_{ikl} = 1\}$, which contains the theoretical abundances of a peptide y_{ikl} if we actually observe it (which is why we have the $O_{ikl} = 1$ term there). And we define another set of all the unobserved peptides y_{ikl}: $Y^{mis} = \{y_{ikl}; O_{ikl} = 0\}$. Then, we can define the likelihood as follows:

$$\begin{aligned} CompleteLikelihood &= \mathbb{P}(Y_{com} | \vec{\mu}, \vec{\sigma^2}, \vec{\tau^2}, \vec{r}, \vec{\lambda}) \\ &= \prod_{ik} \left[\frac{1}{\tau_i} \phi\left(\frac{\gamma_{ik} - \alpha_0 - \alpha_1 a_i}{\tau_i} \right) \binom{s_{ik} + r - 2}{s_{ik} - 1} \lambda^r (1 - \lambda)^{s_{ik} - 1} \prod_{l=1}^{s_{ik}} \left[\frac{1}{\sigma_i} \phi\left(\frac{y_{ikl} - \gamma_{ik}}{\sigma_i} \right) \right] \right], \end{aligned}$$

where $Y_{com} = Y^{obs} \cup Y^{mis}$ and $\phi(.)$ is the probability density function for the normal distribution with mean 0 and variance 1.

Text Models

Assume you are supposed bucket words in a series of essays on science based on their topic for an assignment. If this was in your native language, that should not be difficult at all. Obviously, words like "biology," "organism," "evolution," can be bucketed into one topic group while words like "statistics," "probability," "likelihood" can all go into another topic. But if you did not speak this language, bucketing these words may be difficult. One proposed solution might be to identify words that often occur together and bucket them into the same group. So, if you say "organism, evolution, biology" ten times throughout the essay, you might assume that these words go together. Intuitively, that makes sense. So, we can extend this intuition to building a model and a data generating process for text based on word frequencies.

Model Inputs

For these text models – often called topic models – we need to have a collection of texts (e.g., series of essays, set of emails, a bunch of books). And the scientist must decide the number of topics that these words in the essays should be bucketed into. Typically, this can be anywhere from 20 to 1000 but can go beyond 100000. But the number of topics, k, is simply an input into the model that represents the number of topics. Now, we run into another problem: there are many words in any language. In order to streamline analyses, we have to preprocess the data and specify which words should and should not be in classified (e.g., we may not want "and" in our data because it occurs too often and does not offer information about topics).

So, mathematically, suppose we M documents. Using some natural language processing library, we identify the number of unique words throughout all of the M documents. And then we remove words that are irrelevant to our task. For example, we may not want "and," "or," "that", etc if that is not relevant to the problem. Then, we can also specify words that are important for our task, such as "biology," "statistics," etc. Then, we might remove the top 10% most frequent words and least frequent words that will most likely not help with the topics. After removing unnecessary words and keeping relevant ones, we have a *vocabulary*, a set of *unique* words in our documents. A term is simply an element of our vocabulary. Now, we assign an index to our vocabulary so that each term in the vocabulary is associated with some entry of our vocabulary matrix. It is also important to make a distinction between words and terms: a word is a term in the document while a term is an element of the vocabulary. Because words have some sort of position associated with them in the text while terms do not, words and terms are not the same.

Under the Hood

Now, based on the number of topics, these text models will identify words that frequently occur together (co-occur), and will then lump them into the same topic. Now, within each topic, there is a proportion associated with each word that represents how often this word is to come up in this specific topic, and the sum of all the proportions within an individual topic should be 1. Now, each topic will also have a proportion associated with it: how many words are in this topic relative to all the words in our corpus of text? And so the sum of the proportion of topics will be 1.

Data Generating Process

As with any data generating process, we hope to create a synthetic set that resembles the true data. In this case, we want to create a document of text with the same composition as a document from our larger corpus of text. So, to do this, we first lay out a series of blank spaces that represent words. For each word, we first decide the topic (randomly choose based on the proportion associated with each topic) and then randomly select a word in that topic to fill this blank space. So, for example, suppose we have two topics: biology and statistics with proportions 0.75 and 0.25 respectively. So, 75% of our text relates to biology while 25% relates to statistics. Suppose the biology topic has the following words: biology, organism, evolution each with probability 0.5, 0.4, 0.1 respectively. And suppose the statistics topic has the following: statistics, probability, likelihood each with probability 0.2, 0.1, 0.7. Then, to randomly fill in the first blank word of our document, we choose a topic randomly. The probability we choose the biology topic is 75% while the probability we choose the statistics topic is 25%. Suppose we choose the biology topic; now we choose a word in the biology topic. The probability we choose "biology" is 50%, the probability we choose "organism," is 40%, and the probability we choose "evolution" is 10%. So, we randomly select a word for the first blank and repeat this process until we create a fake document composed entirely of words from these two topics.

Mathematically, we have M documents. For each document d, we have a certain number of words (e.g., "evolution and organisms are biology" has five words), referred to as N_d. Additionally, we have k topics, which we will represent as a k x 1 vector: $\vec{z}_{k\times 1}$. Each entry in the vector represents the proportion of words in the corpus associated with that document. Returning to the scientific articles example, $\vec{z}_{2\times 1} = \begin{pmatrix} \text{biology} \\ \text{statistics} \end{pmatrix} = \begin{pmatrix} 0.75 \\ 0.25 \end{pmatrix}$. Now, we will also define a vector $z_{d,n}$ which is going to represent the topic associated with the word in the n-th position of document d. So, if the topic of the second word in document 1 is "biology," then $\vec{z}_{1,2} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$. Now, even if we have that the topic of word 2 is "biology," we need to insert a word into that blank space. To do this, we define a new term, $\beta = \mathbb{P}(\text{term } j \text{ in vocabulary}|\text{topic } k)$, a $v \times k$ matrix, where v is the number of terms in the vocabulary and k is the number of topics. Each entry represents the proportion of the topic that some term composes. So, in our scientific articles example, our vocabulary would be the following: { biology, organism, evolution, statistics, probability, likelihood}. And our β matrix would be the following:

$$\beta = \begin{pmatrix} \text{biology} & \text{biology} \\ \text{organism} & \text{organism} \\ \text{evolution} & \text{evolution} \\ \text{statistics} & \text{statistics} \\ \text{probability} & \text{probability} \\ \text{likelihood} & \text{likelihood} \end{pmatrix}$$
$$= \begin{pmatrix} 0.5 & 0 \\ 0.4 & 0 \\ 0.1 & 0 \\ 0 & 0.2 \\ 0 & 0.1 \\ 0 & 0.7 \end{pmatrix},$$

where the first column is the proportion of each term that composes the biology topic and the second is the proportion of each term that composes the statistics topic. To choose the word, we need to choose a topic and then a word, so we multiply $\vec{z}_{1,2}$ and β to give us $w_{1,2}$, the word in document 1 and position 2. So, if we have a biology topic, then $\vec{z}_{1,2} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$. Then,

$$\begin{aligned}\beta\vec{z}_{1,2} &= \begin{pmatrix} 0.5 & 0 \\ 0.4 & 0 \\ 0.1 & 0 \\ 0 & 0.2 \\ 0 & 0.1 \\ 0 & 0.7 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} \\ &= \begin{pmatrix} 0.5 \\ 0.4 \\ 0.1 \\ 0 \\ 0 \\ 0 \end{pmatrix}.\end{aligned}$$

Now, we have a vector from which we can sample a word for document d and position n, $\vec{w}_{d,n}$, a $v \times 1$ vector that is 1 if a specific word is chosen. So, if we choose "biology," $\vec{w}_{1,2} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$. This was just a mathematical way of representing the process by which we select words.

Data Generating Process

Now that we have mathematical notation for the process, we can create a data generating process. We have M documents, so $d = 1...M$, terms from $1...v$, topics $1...k$, words/positions in document d from $1...N_d$. Then, we can write the GDP, given $M, v, k, \{N_1...N_M\}, \beta_{v \times k}$. Now, we can write this as an algorithm:

```
for $d = 1...M$
    sample $\vec{\theta}_d \sim Dirichlet_k(\alpha)$, where $\theta_d$ is a $k \times 1$ vector of topic proportions
    for $n = 1...N_d$
            sample topic $z_{d,n} \sim Multinomial(\theta_d, 1)$
            sample word $w_{d,n} \sim Multinomial(\beta\vec{z}_{d,n}, 1)$
```

In this way, we can recreate the set of documents. Even though we mention the Dirichlet distribution the probability distribution function really does not matter. All that you need to know is that it allows you to assign proportions that sum to 1 to a $k \times 1$ vector. And the multinomial distributions allow us to select a vector of 0s except 1, which represents the term choice for our word.

Likelihood

So, we can then write the likelihood as follows:

$$L(\vec{w}, \vec{z}, \vec{\theta}|\alpha, \beta) = \prod_{d=1}^{M} Dirichlet(\theta_d|\alpha) \left[\prod_{n=1}^{N_d} Multinomial(\vec{z}_{d,n}|\theta_d) Multinomial(w_{d,n}|\beta\vec{z}_{d,n}, 1) \right],$$

where $Dirichlet(\theta_d|\alpha)$ is the vector of topic proportions, $Multinomial(\vec{z}_{d,n}|\theta_d)$ is the topic for a single word, and $Multinomial(w_{d,n}|\beta\vec{z}_{d,n}, 1)$ is a term from the selected topic.

Conclusion

There are a multitude of applications for text models – from finance to government to journalism. Researchers have utilized similar methods for understanding authorship attribution and analyzing massive saves of text. This concludes the modeling section of the course, and moving forward, we will discuss techniques for estimation and inference ˘ two other essential components to one's work as a data scientist.