Assignment 4: Sequence Learning and Maximum Likelihood Estimation

Machine Learning

Fall 2019

Learning Objectives

- Understand the idea of maximum likelihood estimation and apply it to the Naïve Bayes model.
- Learn the basic ideas of sequence prediction and approach the problem using bigrams.
- Learn about word embeddings.

1 Placeholder for Companion Notebook

TODO

2 Word Embeddings

As we've seen in this assignment and the previous one, treating each word as a unique entity can result in missed opportunities for learning. For instance, when predicting the sentiment of a movie review, if we don't know that the words *terrific* and *fabulous* have similar meanings, we are going to need a lot of data to learn a good model of sentiment classification.

Word embeddings are one answer to this problem. A word embeddings takes each word and maps it into a vector space where words that have similar meanings are are nearby in the vector space (e.g., they have a small Euclidean distance between them). Methods have been proposed that allow one to learn these word embeddings from giant collections of raw, unlabeled text. Thus, we can develop high quality, numerical representations of the meanings of various words using unlabeled data, and then use a smaller collection of labeled data (for instance, movie reviews with corresponding sentiment values) along with these representations to learn the task at hand.

External Resource(s) (45 minutes)

We'd like you to get the basic idea of word embeddings so you can dig into some important issues regarding bias in machine learning. We think these resources are written at a level where you can get some important details without this becoming unmanageable.

- Watch Jordan Boyd-Graber's video on word2vec (after 15:00 he goes into a connection to the singular value decomposition (SVD), that you shouldn't worry about).
- If you prefer (or would also like) a written resource, consider An Intuitive Understanding of Word Embeddings: From Count Vectors to Word2Vec. This resource builds up to word2vec from simpler methods, which you can skip if you'd like. The most directly relevant path through the content is to read: Intro, What are Word Embeddings?, Prediction based Embedding, Word Embeddings use case scenarios.

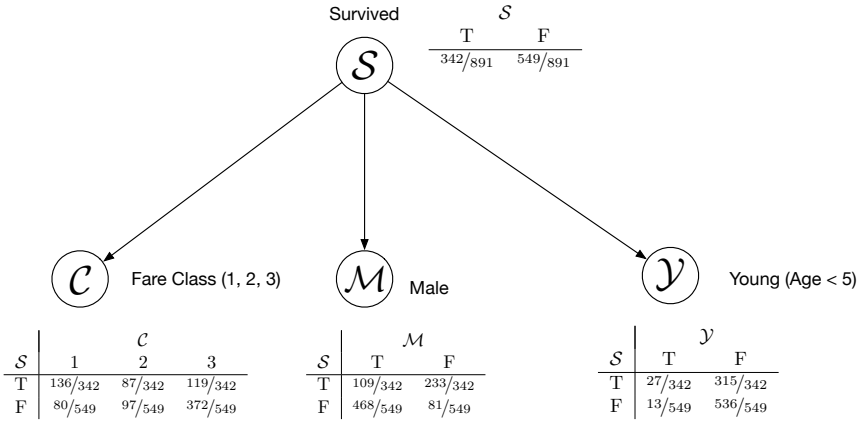
2.1 Bias in Word Embeddings

External Resource(s) (45 minutes)

- Read Text Embedding Models Contain Bias. Here's Why That Matters
- **(Optional)** if you are up for a denser read and want to spend more time on this read Man is to Computer Programmer as Woman is to Homemaker? Debiasing Word Embeddings (**to be clear, you will not be able to read this article in 45 minutes**)

3 Maximum Likelihood Parameter Estimation for Discrete Models

In assignment 2, you met the Naïve Bayes model. As a motivating example, we presented a Bayesian Network (BN) for the Titanic Dataset that modeled three features (*is male*, *is young*, and *fare class*) being generated by whether or not the passenger survived. Here is the BN corresponding to this model.



In assignment 2, we described how we determined the parameters in the conditional probability tables.

> "The probabilities in this BN were computed by looking at the training set and counting the appropriate passengers that fell into each category. For instance, to compute $p(\mathcal{Y}|\mathcal{S})$ since $p(\mathcal{Y}|\mathcal{S}) = \frac{p(\mathcal{Y},\mathcal{S})}{p(\mathcal{S})}$, we can approximate this probability by counting the number of passengers under 5 who survived and dividing by the total number who survived (note that there are some subtle and important modifications to this method of fitting these probabilities that we'll discuss in the next assignment). This process was repeated for each conditional probability. Since we assume that all of the features are conditionally independent given the output ($\mathcal{S}$ in this case), this process is done independently for each feature."

In the assignment 3 companion notebook on sentiment analysis, we employed very similar logic (for instance counting the number of reviews that had positive sentiment and a particular word and dividing it by the number of reviews with positive sentiment).

While these methods of parameter fitting (hopefully) seem logical, it helps to be rigorous about *why* these are the right probabilities to fit given the training data. In this section, we'll go over the math behind determining these probabilities. The goal will be to provide a general outline of a process for fitting parameters of a BN. We'll do so by analyzing the Naïve Bayes model in particular to help you get the "recipe" for how this works.

3.1 Formalizing the Problem

We can think of the numbers in the conditional probability tables as the parameters of our Bayesian Network. In order to compute sensible values for those parameters, we're going to choose the parameters values that agree as closely as possible with a set of training data. At a conceptual level this strategy should feel pretty familiar. In the last module, we did this again and again by tuning model parameters to accurately predict the training outputs as a function of the training inputs (last module the parameters were typically weights of a neural network or a logistic regression model).

Suppose we are given n training data points $\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_n}$. In the case of the titanic dataset $\mathbf{x_i}$ would be a four-dimensional vector consisting of whether the passenger survived ($\mathcal{S}$), the passenger's fare class ($\mathcal{C}$), whether the passenger was male ($\mathcal{M}$), and whether the passenger was under the age of 5 ($\mathcal{Y}$). Further, suppose our model is parameterized by parameters $\mathbf{\Theta}$, which provide the necessary information to compute the probability of any input $\mathbf{x}$. In other words, our model can compute $p(\mathbf{x_i}|\mathbf{\Theta})$ for any of the training points (or any other possible input for that matter).

✔ Understanding Check

In the case of the Titanic model, what would the parameters $\mathbf{\Theta}$ represent? (check solutions for the answer).

☆ Solution

The parameters in this case would represent all of the entries in the conditional probability tables in the BN. For example, the parameters would encode $p(\mathcal{S})$, $p(\mathcal{M}|\mathcal{S})$, etc.

Given our model of $p(\mathbf{x_i}|\mathbf{\Theta})$, we would now like to figure out the best parameters, $\mathbf{\Theta}^\star$ based on our training data. To do this, we can use the technique of maximum likelihood estimation (MLE). The maximum likelihood estimate of the parameters is given by the following formula.

$$\mathbf{\Theta}^\star = \arg\max_{\mathbf{\Theta}} p(\mathbf{x_1}, \mathbf{x_2}, \ldots \mathbf{x_n}|\mathbf{\Theta}) \tag{1}$$

This equation is known as maximum likelihood estimation because: (a) it involves a maximization and (b) it is a likelihood (probability of data given a hypothesis or model). Intuitively, Equation 1 captures the idea that we should choose the model parameters that makes the observed training data as likely as possible under our model.

3.2 Simplifications to Equation 1

It may seem that computing the probability in Equation 1 would be quite difficult. While in some cases it can be, there are some simplifying assumptions that we can apply to make our lives easier. One of the most common assumptions is that the training points are conditionally independent given $\mathbf{\Theta}$ (that is $X_i \perp\!\!\!\perp X_j \mid \mathbf{\Theta}$ for all $i \neq j$). This is known as the independently and identically distributed (i.i.d.) assumption.

✔ Understanding Check

Make sure you have a good conceptual sense of what the i.i.d. assumption is all about. Perhaps a good thing to think through would be why X_i and X_j are *not* independent (they are only conditionally independent given $\mathbf{\Theta}$.

If we apply the i.i.d. assumption to Equation 1, we derive the following equation.

$$\boldsymbol{\Theta}^\star = \arg\max_{\boldsymbol{\Theta}} p(\mathbf{x_1}|\boldsymbol{\Theta})p(\mathbf{x_2}|\boldsymbol{\Theta}) \dots p(\mathbf{x_n}|\boldsymbol{\Theta}) \tag{2}$$

To make things even easier, we can apply a log without changing the arg max. This works because log is a monotonic (continuously increasing) function, so $\arg\max_x f(x) = \arg\max_x \log f(x)$ for any monotonic function f.

$$\begin{aligned}\boldsymbol{\Theta}^\star &= \arg\max_{\boldsymbol{\Theta}} \log\left(p(\mathbf{x_1}|\boldsymbol{\Theta})p(\mathbf{x_2}|\boldsymbol{\Theta}) \dots p(\mathbf{x_n}|\boldsymbol{\Theta})\right) \\ &= \arg\max_{\boldsymbol{\Theta}} \sum_{i=1}^{n} \log p(\mathbf{x_i}|\boldsymbol{\Theta})\end{aligned} \tag{3}$$

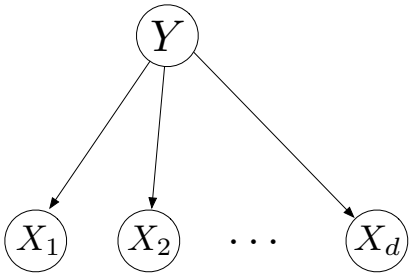
The importance of applying the log might not be apparent yet, but it actually did some useful work for us. Specifically, it broke apart our probability into multiple components (one for each data point). We can now optimize the sum of a bunch of terms rather than the product of a bunch of terms (the former turns out to be much easier). We'll see how this plays out in the next section.

3.3 Maximum Likelihood Estimation for Naïve Bayes

In this section you'll be applying the technique of MLE to the Naïve Bayes algorithm.

Exercise 1 (60 minutes)

The BN for the Naïve Bayes model is shown below.



In this BN, the variable $\mathcal{Y}$ represents some category of interest (e.g., survive versus not survive), and $X_1, X_2, \dots, X_d$ represent various features of a data point (e.g., age, sex, fare class). The rules of d-separation tell us that $X_i \perp\!\!\!\perp X_j \mid Y$ for all $i \neq j$. For simplicity, we'll assume that Y takes on values from the set $\{1, 2, \dots c\}$ and each X_i takes on values from the set $\{1, 2, \dots, r\}$. Extending your work to the case where each of the random variables takes values from some other discrete set is straightforward.

(a) Equation 3 can be written for this model as

$$\boldsymbol{\Theta}^\star = \arg\max_{\Theta} \sum_{i=1}^{n} \log p(X_1 = x_{i,1}, X_2 = x_{i,2}, \dots, X_d = x_{i,d}, Y = y_i|\boldsymbol{\Theta}) \ . \tag{4}$$

Using the concept of d-separation on the BN graph for Naïve Bayes (the figure above), simplify Equation 3. Hint: you'll want to break apart the big joint probability (the probability of all of the X_i's and Y using conditional independence).

Warning: spoiler alert if you look at part b.

☆ Solution

We know that each of the variables in a BN is conditionally independent given its parents. We can use this result to write the joint probability of all of our random variables the X's and Y in terms of the probability of each conditioned on its parents.

$$\boldsymbol{\Theta}^\star = \arg\max_{\boldsymbol{\Theta}} \sum_{i=1}^{n} \left(\log p(Y = y_i|\boldsymbol{\Theta}) + \sum_{j=1}^{d} \log p(X_j = x_{i,j}|Y = y_i, \boldsymbol{\Theta}) \right) \quad . \tag{5}$$

(b) The answer to part (a) is given here to help setup the next part of this question.

$$\boldsymbol{\Theta}^\star = \arg\max_{\boldsymbol{\Theta}} \sum_{i=1}^{n} \left(\log p(Y = y_i|\boldsymbol{\Theta}) + \sum_{j=1}^{d} \log p(X_j = x_{i,j}|Y = y_i, \boldsymbol{\Theta}) \right) \quad . \tag{6}$$

Suppose, $\boldsymbol{\Theta}$ consists of the following parameters.

- $\Theta(1), \Theta(2), \ldots, \Theta(c)$ represent the prior probability that y takes on a particular value (e.g., $\Theta(i) = p(Y = i)$). Note that in order for $\Theta(1), \ldots, \Theta(c)$ to represent a valid probability mass function $\sum_{i=1}^{c} \Theta(i) = 1$.
- $\Theta_j(k|i)$ represents the conditional probability that feature $X_j = k$ given $Y = i$. That is $\Theta_j(k|j) = p(X_j = k|Y = i)$. In order for the $\Theta_j(k|i)$'s to represent valid PMFs, $\sum_{k=1}^{r} \Theta_j(k|j) = 1$ for all j and i.

Suppose that ycount(i) represents the number of training points where $y = i$ (i.e., ycount$(i) = \sum_{j=1}^{n} \mathbb{I}[y_j = i]$ where $\mathbb{I}$ is the indicator function, which returns 1 if the condition is true and 0 otherwise).

Suppose that xcount$_j(k|i)$ represents the number of training points where $x_j = k$ and $y = i$ (i.e., xcount$_j(k|i) = \sum_{u=1}^{n} \mathbb{I}[x_{u,j} = k, y_j = i]$).

Rewrite Equation 6 in terms of the $\Theta(i)$'s, $\Theta_j(k|i)$'s, ycount's, and xcount's. Hint: replace summations over the data points with summations over the possible values that the random variables can take on.

Warning: spoiler alert if you look at part c.

☆ Solution

To help understand the logic of what we're going to do in this solution, consider the summation $\sum_{i=1}^{n} \log p(Y = y_i|\boldsymbol{\Theta})$. The value of the term we are summing will depend on $\boldsymbol{\Theta}(y_i)$ (the probability that our model assigns to the $Y = y_i$. Similarly, the term in the second part of the equation we are simplifying, $\sum_{i=1}^{n}\sum_{j=1}^{d} \log p(X_j = x_{i,j}|Y = y_i, \boldsymbol{\Theta})$, only depends on $\boldsymbol{\Theta}_j(x_{i,j} \mid y_i)$. Given these observations, we can rewrite the equation in the following way.

$$\begin{aligned}\boldsymbol{\Theta}^\star &= \arg\max_{\boldsymbol{\Theta}} \sum_{i=1}^{n} \left(\log \boldsymbol{\Theta}(y_i) + \sum_{j=1}^{d} \log \boldsymbol{\Theta}_j(x_{i,j} \mid y_i) \right) \\ &= \arg\max_{\boldsymbol{\Theta}} \left(\sum_{i=1}^{n} \log \boldsymbol{\Theta}(y_i) \right) + \left(\sum_{i=1}^{n}\sum_{j=1}^{d} \log \boldsymbol{\Theta}_j(x_{i,j} \mid y_i) \right) \quad \text{break it apart to make it easier to deal with}\end{aligned}$$

We can further change these equations by removing the summations over the n training points and instead summing over each of the possible values of y_i (weighting by the count) and each possible combination of the values of $(y_i, x_{i,j})$ (again weighting by the count). We already defined the relevant counts above, so now we'll just substitute them into the equation.

$$\boldsymbol{\Theta}^\star = \arg\max_{\boldsymbol{\Theta}} \left(\sum_{i=1}^{c} \text{ycount}(i) \log \Theta(i) \right) + \left(\sum_{j=1}^{d}\sum_{i=1}^{c}\sum_{k=1}^{r} \text{xcount}_j(k|i) \log \Theta_j(k|i) \right) \tag{7}$$

(c) The maximum likelihood equation for the model (the answer to part b) is as follows.

$$\boldsymbol{\Theta}^\star = \arg\max_{\boldsymbol{\Theta}} \left(\sum_{i=1}^{c} \text{ycount}(i) \log \Theta(i) \right) + \left(\sum_{j=1}^{d}\sum_{i=1}^{c}\sum_{k=1}^{r} \text{xcount}_j(k|i) \log \Theta_j(k|i) \right) \tag{8}$$

Since each of the various parameters only affects particular terms in these summations, we can break the maximization over the entire parameters space $\boldsymbol{\Theta}$ into a bunch of separate maximization problems. For example, the first summation in Equation 8 is only affected by $\Theta(1), \Theta(2), \ldots, \Theta(c)$, therefore

$$\Theta^\star(1), \ldots, \Theta^\star(c) = \arg\max_{\Theta(1),\ldots,\Theta(c)} \sum_{i=1}^{c} \text{ycount}(i) \log \Theta(i) \ . \tag{9}$$

One thing to remember about this equation is that not all values of $\Theta(1), \ldots, \Theta(c)$ are permissible. We know that these parameters have to specify a valid probability mass function, which requires that $\sum_{i=1}^{c} \Theta(i) = 1$ and each $\Theta(i) \geq 0$. In the language of numerical optimization, these equations are known as *constraints*.

Additionally, for i in the set $\{1, 2, \ldots, c\}$ and j in the set $\{1, 2, \ldots, d\}$

$$\Theta_j^\star(1|i), \ldots, \Theta_j^\star(r|i) = \arg\max_{\Theta_j(1|i),\ldots,\Theta_j(r|i)} \sum_{k=1}^{r} \text{xcount}_j(k|i) \log \Theta_j(k|i) \ . \tag{10}$$

For similar reasons to the ones we just stated for $\Theta(1), \ldots, \Theta(c)$, Equation 10 must satisfy $\sum_{k=1}^{r} \Theta_j(k|i) = 1$ and each $\Theta_j(k|i) \geq 0$ (the reason being, again, that these values must specify a valid PMF).

One way to derive the solution to these constrained optimization problems, is to use the technique of Lagrange Multipliers. Here is a walkthrough of using this strategy to solve the equations for the Naïve Bayes algorithm (the proof is in section 4.2). Instead of having you prove this directly (do prove it if you feel inclined), let's take as given the following theorem.

Suppose $c_1, \ldots c_m$ represent non-negative constants ($c_i \geq 0$). Further, suppose $q_1, \ldots q_m$ represents a PMF ($q_i \geq 0$ and $\sum_{i=1}^{m} q_i = 1$). If this is true then,

$$q_1^\star, \ldots, q_m^\star = \arg\max_{q_1,\ldots,q_m} \sum_{i=1}^{m} c_i \log q_i \tag{11}$$

$$q_i^\star = \frac{c_i}{\sum_{j=1}^{m} c_j} \tag{12}$$

These equation are pretty dense, so let's take a minute to unpack them. Equation 11 is stating that our goal is to compute values of $q_1, \ldots q_m$ that maximize $\sum_{i=1}^{m} c_i \log q_i$. We also require (but did not write explicitly in the equation) that $q_1, \ldots, q_m$ define a valid probability mass function ($q_i \geq 0$ and $\sum_{i=1}^{m} q_i = 1$). Equation 12 states, without proof, the solution to Equation 11 (again, it's not too crazy to prove this if you would like to try).

Using the theorem above, find the optimal values of the parameters of the Naïve Bayes model. In other words, compute $\Theta^\star(1), \ldots, \Theta^\star(c)$ and $\Theta_j^\star(1|i), \ldots, \Theta_j^\star(r|i)$ (for all appropriate values of i and j). Does the result match your intuitions about what the Θ values should be?

Hint: You should be able to pattern match Equation 12 to both Equation 9 and Equation 10.

☆ Solution

Equation 9 can be solved using Equation 12 in the following way.

$$\Theta^\star(i) = \frac{\text{ycount}(i)}{\sum_{j=1}^{c} \text{ycount}(j)} \tag{13}$$

Equation 10 can be solved using Equation 12 in the following way.

$$\Theta_j^\star(k|i) = \frac{\text{xcount}_j(k|i)}{\sum_{u=1}^{r} \text{xcount}_j(u|i)} \tag{14}$$

The answers both seem logical since in the case of computing the $\Theta^\star(i)$ we just counted the number of training points that had $Y = i$ (this matches what we did in previous assignments). In the case of computing $\Theta_j^\star(k|i)$ we calculated the proportion of training inputs where $X_j = k$ and divided by the sum of the number of training points that where X_j took on some other value (the summation in the denominator should be the same as the total number of training points).