Statistical Inference: n-gram Models over Sparse Data

Statistical Inference

- **Statistical inference** is the process of deducing properties of the distribution of your data using statistical methods.
  - What is the probability that an English sentence contains a gerund?
  - How often are prepositional phrases used?
- **Language modeling** is the task of estimating the probability of the next word in a sentence given its preceding words.
  - What is the probability that the next word of "Every morning I eat bacon and" is "eggs"?
- The **Shannon Game** is the task of guessing the next letter in a text.
  - The same techniques can be used in both, however the probability space of words may be richer than the probability space of letters.

Reliability vs. Discrimination

- To make any decisions about our data, we must first define the problem formally.
- The objective of any task is to predict a **target feature** from **classificatory features**.
  - Features are any distinguishing characteristics between our data points which we can use to separate them into equivalence classes.
- For this, it's often reasonable to make **independence assumptions**.
  - While our our classificatory features may not be independent of each other, we do not see these dependencies to have a significant impact.
- The trade-off: more features give us higher **discrimination**, but as each equivalence class gets smaller, we lose **reliability**.

n-gram Models

We can define the basic language modeling task as follows:

$P(w_n|w_1, \ldots, w_{n-1})$ Where w_i represents the *i*-th word in the sentence.

Here, we call $w_1, \ldots, w_n$ the **history**, the words that we have already observed.

Histories will generally be long unique sequences, meaning that we have not seen the exact sentence before.

Even when the histories are not unique, the next word could very well end differently the next time it's seen.

I like to go to the mall.

I like to go to the museum.

Markov Assumption

- One method for handling histories is to accept the **Markov Assumption**.
- The Markov Assumption is that next state of a process only depends on its local context.
  - This means we only need to consider the last few words rather than the entire history.
  - We will expand on this in later slides that describe Markov Models.
- If we only consider the last *n-1* words, before predicting the *n*-th, we can represent our dataset as a list of n-grams.
  - The *n*-th word is our target feature.
  - The first *n-1* words are our classificatory features.
- n-gram models for *n = 2, 3,* or *4* are often called **bigram**, **trigram**, **four-gram**.

Statistical Estimators on n-grams

In general, the overall problem can be solved by the frequentist approach:

$$P(w_n \mid w_1, \ldots, w_{n-1}) = \frac{P(w_1, \ldots, w_n)}{P(w_1, \ldots, w_{n-1})}$$

Calculate the rate of occurrence of the full sequence, divided by the rate of occurrence of the sequence without the last word.

Fails with sparsity of dataset; can't predict novel sentences.

Using Maximum Likelihood Estimation, we can approximate P to be consistent with our relative frequencies of the n-grams.

Pick a distribution function P which maximizes the probability of our training set.

Laplace's Law

Sparsity is an issue in natural languages.

- Languages provide the basis for such a wide variety of syntactically and grammatically correct sentences, that there are potentially an infinite number of sentences.
- Datasets are finite and will therefore be missing a vast majority of sentence examples.

Predicting sentences that haven't been seen before using MLE is not possible.

- They all have relative frequency zero.

$$P_{\text{Laplace}}(X) = \frac{\text{freq}(X) + 1}{N + |\Omega|}$$

Laplace's Law seeks to resolve this issue by increasing relative frequency of all outcomes in a probability space without changing the order of probabilities.

Laplace's Law (Continued)

- Effectively, Laplace's Law adds one example of every possible outcome to the space while also increasing the size of the observed dataset by how many added samples were included.
  - This is equivalent to the Bayesian estimator with a uniform prior probability assumed.
- For natural languages with sparse datasets and large vocabularies, this works very poorly, actually.
  - All unseen outcomes have equal probability. For example:
    - "I like dogs."
    - "I like histopathology."
- Also, may predict unseen occurrences with probabilities too closely to rarely-but-seen occurrences.

Lidstone's Law and Jeffreys-Perks Law

Lidstone's Law considers scaling the increase to the probability by a factor λ.

$$P_{\text{Lidstone}}(X) = \frac{\text{freq}(X) + \lambda}{N + \lambda|\Omega|}$$

This allows you to further separate rare-but-seen occurrences from never-before-seen occurrences.

The **Jeffreys-Perks Law** is equivalent to the Lidstone's Law with λ = 0.5

$$P_{\text{Jeffreys-Perks}}(X) = \frac{\text{freq}(X) + \frac{1}{2}}{N + \frac{|\Omega|}{2}}$$

Validation Methods

- How do we know that we've picked good parameters?
  - Our choice of distribution parameters, our lambda for Lidstone's Law, etc.
- The parameters we learned on our training set will likely work very well for data we've already observed.
- However, we need to ensure that the models we've learned will work on new data.
  - We already know the results of our training set.
  - Predictions should be able to generalize to the problem space.
- One method for this is test the learned models on new instances from the problem space.

Testing Sets

Testing sets can be formed by holding out a portion of your dataset from the training of your models.

These instances will not be used to calculate relative frequencies or estimate other parameters.

Once you've estimated your distributions on the remaining data, you can validate them against the testing set.

The idea is that the learned content from your training set should generalize well to unseen data.

Problems with a Single Test Set

How did we choose the test set?

- Was our choice fair and representative of other unseen data?
- Was it a poor choice?
- Should we pick different test sets until the model performs well on it?
  - NO!

The purpose behind using a test set is to prepare the model for unseen data.

- Any choice of testing set should perform roughly the same for a robust model.