Statistical Inference: n-gram Models over Sparse Data

B4 Sudayama

November 7, 2014

Overview

Introduction

Statistical inference in general consists of two parts:

- Given some data which is assumed to be genereated according to some distribution,
- make some inferences about that distribution.

In this chapter, language modeling is dealt with based on statistical inference.

- Given previous words,
- predict the next word.

Forming Equivalence Classes

The target feature is predicted on the basis of various classifactory features. One of those features the authors suggest is the past behavior. For that reasoning to be valid, Some assumptions are made implicitly:

- Each point is distributed roughly stationary.
- One point is independent of the other.

Tradeoff between Reliability and Discrimination

- Dividing the data into many bins (or equivalence classes) lead to greater discrimination.
- On the other hand, the estimation may not be statistically reiliable.

n-gram models

To give reasonable predictions, Markov assumption is made. In n-gram models, all histories that have the same last (n-1) words are placed in the same class. In the probability calculations at the start of a sentence, some special tokens are inserted.

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

MLE and smoothing

With fixed observed data, the choice of parameter values to give highest probability within a certain distribution is called as the maximum likelihood estimate.
Since the MLE assigns zero probability to unseen events, a little bit of probability mass is left over for previously unseen events. That process is referred to as smoothing.

common practice and some heuristics

- Usually n is chosen to be two (bigram) or three (trigram).
- n-grams are caluculated only for the most common k words and all other words are mapped to a single token.
- In the other case, such mapping is done for all words that is seen only once in the training corpus.
- With enough data, use higher order n-gram. Otherwise back off to lower order.

Notation

- N Number of training instances
- B Number of bins (or equivalnce classes)
- $C(w_1 \ldots w_n)$ freq of n-gram in training text
- N_r number of bins that have r training instances in them

Smoothing methods

Laplace's law: Adding one

$$P(w_1 \ldots w_n) = \frac{c(w_1 \ldots w_n) + 1}{N + B}$$

Lidstone's law: Adding some positive value λ

$$P(w_1 \ldots w_n) = \frac{c(w_1 \ldots w_n) + \lambda}{N + B\lambda}$$

Held out estimator

One way to estimate the probability mass to be left over for unseen events is held out estimation. With further text from the same source,

$$C_1(w_1 \ldots w_n) = \textit{ freq in training data}$$
$$C_2(w_1 \ldots w_n) = \textit{ freq in held out data}$$
$$T_r = \Sigma_{w_1 \ldots w_n \ : C1(\ldots)=r} C_2(w_1 \ldots w_n)$$

The average freq of those n-grams is T_r/N_r. Thus an estimate for one of these n-grams is:

$$P_{ho}(w_1 \ldots w_n) = \frac{T_r}{N_r N}$$

Pots of data

- one should always separate the data into a training and testing portion.
- Testing a succession of variant models can also lead to overtraining.
- In such cases, the test set is further divided.

Cross-validation

Rather than using some of the training data only for freq counts and some only for smoothing estimates, each part is used both as initial training and held out data. Such methods in statistics is called as cross-validation.

Good-Turing estimation

The count for each event is assumed to be binomial-distributed. By the underlying theorem, the following estimate is given for previously observed items:

$$P_{GT} = \frac{r^*}{N} \text{ where } r^* = (r+1)\frac{E(N_{r+1})}{E(N_r)}$$

In practice, that reestimation is used only for low frequency words since such words are numerous and the observed freq of frequencies is quite accurate.

back-off models

In back-off models, different model are consulted.
The most detailed model that is deemed to provide reliable information is used.