Machine Learning (II): Markov Chain, Statistical Language Models and Conditional Distribution

Le Wang

2019-02-05

Applications of Conditional Distributions

1. Transition Matrix and Income Mobility
2. Statistical Language Model

Application 1: Income Mobility or Intergenerational Mobility

Question: How can I understand intergenerational mobility or income mobility?

Application 1: Income Mobility or Intergenerational Mobility

Question: How can I understand intergenerational mobility or income mobility?

1. Moving from **last** generation to **current** one
2. Moving from **last** period to **current** period

We can discretize the income into many different categories, e.g., bottom quartile, 2nd, 3rd, top quartiles.

$$\Pr[X_t|X_{t-1}] = \frac{\Pr[X_t, X_{t-1}]}{\Pr[X_{t-1}]}$$

Note: You should already know how to discretize a variable, but we will introduce another way here.

Let's look at the `R` code to

1. What a transition matrix looks like
2. How we can construct a transition matrix from the data.

Markov Chain Monte Carlo

In some applications, sampling from the joint distribution is either infeaible or difficult. **Morkov Chain Monte Carlo** (MCMC) involves algorithms that use simulation to construct approximations to the joint distribution in several different ways, but as the name suggests, the principal device is the Markov chain.

A **Markov Chain** is a stochastic process for whic, given the current state, future states of the random variable Y are independent of past states.

$$\Pr[X_{t+1} = x | x_t, x_{t-1}, \ldots] = \Pr[X_{t+1} = x | x_t]$$

Statistical Language Models

Foundation of **natural language processing**.

Computer Science

1. Machine translation
2. Voice Recognition
3. Handwritting recognition
4. Spelling correction

Social Science

1. Sentiment Analysis/Opinion Minning
2. Document Classficiation

Statistical Language Models (Motivation)

Question: Which stentence is **reasonable**?

1. I have known John approximately seven years
2. approximately seven years I have known John
3. Have approximately seven years known John I

Statistical Language Models (Solutions)

Solution 1:

Before 70s, scientists are trying to figure out the answer by examining grammar etc..

Statistical Language Models (Solutions)

Solution 1:

Before 70s, scientists are trying to figure out the answer by examining grammar etc..

Solution 2:

Frederick Jelinek: Whether a sentence is reasonable depends on the probability of its occurrence!

$$P(S) = P(w_1, w_2, w_3, \ldots, w_n)$$

A sentence S consists of n words in the above order. Nice idea, but how the heck to implement (or calculate) this?!

Statistical Language Model (Solution)

$$P(S) = P(w_1, w_2, w_3, \ldots, w_n)$$

Statistical Language Model (Solution)

$$
\begin{aligned}
P(S) &= P(w_1, w_2, w_3, \ldots, w_n) \\
&= P(w_n | w_1 w_2, w_3, \ldots, w_{n-1}) \cdot \\
& P(w_1, w_2, w_3, \ldots, w_{n-1})
\end{aligned}
$$

Statistical Language Model (Solution)

$$
\begin{aligned}
P(S) &= P(w_1, w_2, w_3, \ldots, w_n) \\
&= P(w_n | w_1 w_2, w_3, \ldots, w_{n-1}) \cdot \\
&P(w_1, w_2, w_3, \ldots, w_{n-1}) \\
&= P(w_n | w_1 w_2, w_3, \ldots, w_{n-1}) \cdot P(w_{n-1} | w_1 w_2, w_3, \ldots, w_{n-2}) \\
&\cdot P(w_1, w_2, w_3, \ldots, w_{n-2})
\end{aligned}
$$

Statistical Language Model (Solution)

$$
\begin{aligned}
P(S) &= P(w_1, w_2, w_3, \ldots, w_n) \\
&= P(w_n | w_1 w_2, w_3, \ldots, w_{n-1}) \cdot \\
& P(w_1, w_2, w_3, \ldots, w_{n-1}) \\
&= P(w_n | w_1 w_2, w_3, \ldots, w_{n-1}) \cdot P(w_{n-1} | w_1 w_2, w_3, \ldots, w_{n-2}) \\
& \cdot P(w_1, w_2, w_3, \ldots, w_{n-2}) \\
&= P(w_n | w_1 w_2, w_3, \ldots, w_{n-1}) \cdots P(w_3 | w_1 w_2) \cdot P(w_2 | w_1) \cdot p(w_1)
\end{aligned}
$$

Some improvement: Easy to calculate $P(w_1)$ and $P(w_2 \mid w_1)$, not too bad to calculate $P(w_3 \mid w_1, w_2)$..

Statistical Language Model (Solution)

$$
\begin{aligned}
P(S) &= P(w_1, w_2, w_3, \ldots, w_n) \\
&= P(w_n | w_1 w_2, w_3, \ldots, w_{n-1}) \cdot \\
&P(w_1, w_2, w_3, \ldots, w_{n-1}) \\
&= P(w_n | w_1 w_2, w_3, \ldots, w_{n-1}) \cdot P(w_{n-1} | w_1 w_2, w_3, \ldots, w_{n-2}) \\
&\cdot P(w_1, w_2, w_3, \ldots, w_{n-2}) \\
&= P(w_n | w_1 w_2, w_3, \ldots, w_{n-1}) \cdots P(w_3 | w_1 w_2) \cdot P(w_2 | w_1) \cdot p(w_1)
\end{aligned}
$$

Some improvement: Easy to calculate $P(w_1)$ and $P(w_2 \mid w_1)$, not too bad to calculate $P(w_3 \mid w_1, w_2)$..

But how to calculate $P(w_n | w_1 w_2, w_3, \ldots, w_{n-1})$??????

Statistical Language Model (Solution)

$$P(S) = P(w_n|w_1 w_2, w_3, \ldots, w_{n-1}) \cdots P(w_3|w_1 w_2) \cdot P(w_2|w_1) \cdot p(w_1)$$

Statistical Language Model (Solution)

$$P(S) = P(w_n|w_1 w_2, w_3, \ldots, w_{n-1}) \cdots P(w_3|w_1 w_2) \cdot P(w_2|w_1) \cdot p(w_1)$$

Andrey Markov's approach is applied: What if the probability of a word's occurrence depends only on the last word?! (**Markov Chain**!)

$$\begin{aligned} P(S) &= P(w_n|w_1 w_2, w_3, \ldots, w_{n-1}) \cdots P(w_3|w_1 w_2) \cdot P(w_2|w_1) \cdot p(w_1) \\ &= P(w_n|w_{n-1}) \cdots P(w_3|w_2) \cdot P(w_2|w_1) \cdot p(w_1) \end{aligned}$$

We know how to calculate every one of them! **Bigram Model**!

How do we implement this idea?

0. Find a Corpus.

How do we implement this idea?

0. Find a Corpus.
1. Calculate how many times w_{n-1}, w_n appear in the text, and then calculate the number of w_{n-1} appearing in the text.

How do we implement this idea?

0. Find a Corpus.
1. Calculate how many times w_{n-1}, w_n appear in the text, and then calculate the number of w_{n-1} appearing in the text.
2. $P(w_n|w_{n-1}) = \frac{\#(w_{n-1},w_n)}{\#w_{n-1}}$

Many linguisists question the method, but it has worked very well.

1. Only after 2-year development, Google Voice and Rosetta was ranked No. 1 in NIST's 2001 evaluations: National Institute of Standards and Technology
2. Kai-Fu Lee (was a Ph.D student then) was able to employ a statistical language mdoel to simplify a 997-word speech recognition problem to 20-word recognition problem.

Further Extension: N-gram Model

$$P(S) = P(w_n|w_1 w_2, w_3, \ldots, w_{n-1}) \cdots P(w_3|w_1 w_2) \cdot P(w_2|w_1) \cdot p(w_1)$$

We can employ $k-1$ Markov Chain Assumption: k-gram Model!
(but the notation in the literature is usually N-gram model)

$$\begin{aligned} P(S) &= P(w_n|w_1 w_2, w_3, \ldots, w_{n-1}) \cdots P(w_3|w_1 w_2 \\ &= P(w_n|w_{n-1}, \boldsymbol{w_{n-2}}, \boldsymbol{w_{n-3}}, \boldsymbol{w_{n-k+1}}) \cdots \end{aligned}$$

$$P(w_3|w_2) \cdot P(w_2|w_1) \cdot p(w_1)$$

In practice, $k = 3$.

Statistical Language Models (Issues)

1. **What if** you do not observe some (w_{n-1}, w_n)?
   1.1 Increase the sample size (trainning a 3-gram model with Chinese language requires $200,000^3 = 8 \times 10^{15}$ parameters, 10 billion meaningful websites)
   1.2 I.J. Good and Alan Turing's Good-Turing Estimate: take into account unseen events (by assigning some probability to things that you have not seen). Discount the probability for what you actually see.
2. **Choice of Corpus**:
   2.1 Tecent's early choice: People's Daily (Ren Min Ri Bao): best, official language terrible performance!
   2.2 Move to crappy websites with a lot of noises, but it actually does a lot better!