

Probability

Chen Yu
Indiana University

Probability

$P(A)$ as the fraction of possible worlds in which A is true.

The axioms of probability

(1) $0 \leq P(A) \leq 1$

(2) $p(A \vee B) = p(A) + P(B) - P(A \wedge B)$

Two theorems from the Axioms:

$$p(\neg A) = 1 - P(A)$$

$$p(A) = p(A \wedge B) + p(A \wedge \neg B)$$

Example

- The results of the race are captured by a random variable R , with sample space {win,lose}. A horse Harry won 20 of 100 races, then we could say
 $P(R=\text{win}) = 20/100=0.2$;
 $P(R=\text{lose})=20/100=0.8$;
- To represent richer situations, we might care what the weather was like at each race (with a random variable W with values rain or shine). To model how these two variables interact, we need a **joint** probability distribution that combines the two random variables R and W :
<win,rain> <lose, rain> <win, shine> <lose, shine>

Example

- It rained in 30 of Harry's races, and he won 15 of them:
 $P(R=\text{win}, W=\text{rain})=15/100=0.15$;
 $P(R=\text{win}, W=\text{shine})=5/100=0.05$;
 $P(R=\text{lose}, W=\text{rain})=15/100=0.15$;
 $P(R=\text{lose}, W=\text{shine})=65/100=0.65$;
- Given a joint distribution, we can derive the probability distributions for each single random variable (called marginal probability):
 $P(R=\text{win}) = P(R=\text{win}, W \in \{\text{rain}, \text{shine}\})$
 $= P(R=\text{win}, W=\text{rain}) + P(R=\text{win}, W=\text{shine})$

Conditional Probability

- If it is raining, what is the probability that Harry will win?
 $P(R=\text{win} | W=\text{rain}) = P(R=\text{win}, W=\text{rain}) / P(W=\text{rain})$
 $= 15/30 = 0.5$
- The conditional probability is written as $P(A|B)$
 $P(A|B) = P(A, B) / P(B)$
- Almost all our knowledge about the world is in the form of conditional probabilities, it is relative to a certain context.

More Realistic

- To make the above example more realistic, we should include many other variables as we use to represent the world. For instance, we might represent what Harry had for lunch, or what Harry's current health is, or how Harry did in his last race.
- The key issues to formulate any problem is selecting and identifying what random variables to use, and when we formulate knowledge, knowing what the dependencies are.

Independence

- The variables, such as when we got up in the morning, and how we got to the racetrack, should not affect how Harry performs in the race, unless a few of us believe in lucky omens. This is the notion of independence.
- Two random variables are independent if the probability distribution of one isn't affected by the values of the other.
e.g. G states whether we drove or walked to the racetrack. We believe:
 $P(R|G)=P(R)$
- Making independence assumptions is critical in building probabilistic models for it allows us to ignore certain information that is not relevant.

Independence

- An equivalent way to define independence relates the joint distribution to the marginal distributions. In particular, random variables X and Y are independent if
 $P(X,Y)=P(X)*P(Y)$
- A new random variable indicates whether I attend the race or not (at home). I go to the track 60 times, and Harry wins 12 races. This means that the probability that Harry wins if I attend the race is $12/60=.2$. The same as if I don't attend., and the same probability over all races. Thus we can prove that the random variable R is independent of the random variable A
 $P(R=win|A=track) = 12/60 = .2$
 $P(R=win|A=home)=8/40 = .2$
 $P(R|A) = P(R)$

Bayes Rule

- $P(A|B)=(P(B|A)*P(A))/P(B)$
- Let's assume that we do know the probability that he wins any races (.2), the probability that he wins when its raining (.5), and the probability that it rains (.3). We can calculate the probability that it rained on a day when Harry won a race:
 $P(rain|win)=(P(win|rain)*P(rain))/P(win)$
 $= .5*.3/.2 = .75$

Joint Distribution

Joint distribution counts	R=win	R=lose	Marginal counts for W
W=rain	15	15	30
W=shine	5	65	70
Marginal counts of R	20	80	100

Building such a table may be impossible in most case.

Chain Rule

- $P(X_1, X_2, \dots, X_n) = P(X_1 | X_2, \dots, X_n) * P(X_2 | X_3 \dots X_n) \dots P(X_{n-1} | X_n) * P(X_n)$
- $P(A, B, C) = P(A | B, C) * P(B | C) * P(C)$
- The chain rule will be important when we estimate the probability of sequential data.

Using Probability in Language Processing

- To use probability theory for natural language applications, we need to know what the probability distributions are for language.
- We don't know what these distributions are.
- The simplest estimation technique is to count up the number of occurrences of each outcome in a training corpus, and assume that these numbers accurately reflect the true probability distribution. This method is called maximum likelihood estimation (MLE).

Law of large numbers

- The more data you use for estimate, the better the estimate actually gets.
- The other issue is how to balance a tradeoff between the accuracy to which a probability distribution can model the phenomena versus the amount of data we would require to adequately estimate it.

A very basic model

- Introduce one random variable C with its set of outcomes being the part of speech tags.
- We estimate the probability distribution of this random variable using the MLE by counting up what tags are identified in the corpus for each word.
e.g. the corpus contains 100,000 words of which 33,000 are nouns.
 $P(C=\text{noun})=.33$ $P(C=\text{Verb})=.22$ $P(C=\text{adj})=0.11$

A simple conditional Probability model

- We can do much better by using conditional probability model, where the probability of the tag is conditional on what the word is $P(C|W)$.
 $P(C|W)=P(C,W)/P(W)$
 $P(C=t,W=w)$ is estimated by $\#(\text{word } w \text{ with tag } t) / \#(\text{words in corpus})$
 $P(W=w)$ is estimated by $\#(\text{word } w \text{ occurs}) / \#(\text{words in corpus})$
- $P(C|W)$ is estimated by $\#(\text{word } w \text{ with tag } t) / \#(\text{words in corpus})$

Example

- The word “level” occurs 100 times in the corpus, and 53 times as a verb, 28 times as an adjective, and 19 times as noun.
 $P(C=\text{verb}|W=\text{level})=53/100=.53$
 $P(C=\text{adj}|W=\text{level})=28/100=.28$
- We can define and obtain such an estimate for every word that appears in the corpus.
- How well this approach works?

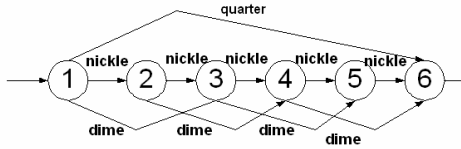
Sequential data

- Language is inherently a sequential phenomena.
- Words occur in sequence over time, and the words that appeared so far constrain the interpretation of words that follow.
- Only certain sequences of words are considered to be grammatical in a language.

Finite State Machines

- One of the simplest models of sequential processes is the finite state machine.
- FSM consists of a set of states and a set of connections called transitions that allow movement between states.
- An FSM defines a set of sequences that it is said to accept by the following methods: a sequence is accepted by an FSM if we can start at the starting state, and find a path through the FSM where the outputs on the transitions generate the sequence and end in an end state.

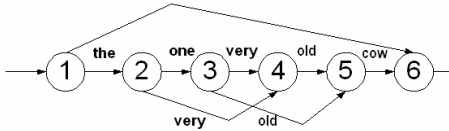
Gumball Machine



The set of sequences that an FSM accepts is called its Language.

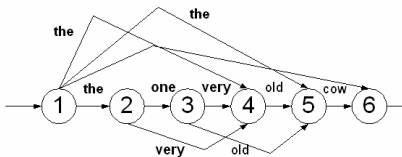
An FSM accepting a set of noun phrases

- The one very old cow
- The one old cow
- The very old cow



Deterministic and Non-deterministic FSMs

- In the previous example, given a word, we know the next state.
- Example, if we add two more phrases: the cow and the old cow

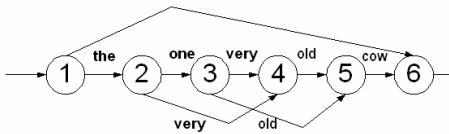


Weighted FSMs

- In many application, we are not only interested in whether a sequence is a valid sentence in a language but also computing the probability of a sequence.
e.g. speech recognition
- A WFSM is an FSM with numbers on the transitions.

Old Cow Example

- We have the FSM

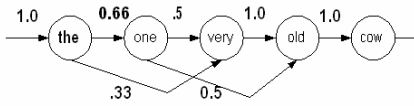


- We collect the following data:
The very old cow
The one old cow
The one very old cow

Transition Probability

- We can use this data to estimate the probability of each transition.
- First, we label the sequences with the states they pass through.
The/1 very/3 old/4 cow/5
The/1 one/2 old/3 cow/5
The/1 one/2 very/3 old/4 cow/5
- We have three sentences in the training data pass through state 1, two going on the state 2 and one to state 3. So the probability from 1 to 2 is .66 and from 1 to 3 is .33.

Weighted FSMs



A deterministic weighted finite state machine has a nice property in that we can very quickly determine whether a sequence is accepted by the machine and we can also compute the probability of a sequence by simply multiplying the probabilities on each of the transitions together.


$$P(\text{The one old cow}) = P(1 \rightarrow 2) * P(2 \rightarrow 3) * P(3 \rightarrow 5) = 0.66 * .5 * 1.0 = 0.33$$

Language Model

- A language model is a probability distribution over word sequences.
 $P(\text{"I can can the peaches in a can"}) \approx 0$
 $P(\text{"I can do it"}) = 0.001$
- What's a language model for?
 speech recognition, web mining...
 statistical model of sequential data.

Speech Recognition

$$\begin{aligned} & \operatorname{argmax}_{\text{wordsequence}} P(\text{wordsequence}|\text{speech}) \\ &= \operatorname{argmax}_{\text{wordsequence}} \frac{P(\text{speech}|\text{wordsequence}) \times P(\text{wordsequence})}{P(\text{speech})} \\ &= \operatorname{argmax}_{\text{wordsequence}} P(\text{speech}|\text{wordsequence}) \times P(\text{wordsequence}) \end{aligned}$$


 Language model

How language models work?

- Compute $P(\text{"I can do it"})$
 - The first step – decompose probability
- $$P(\text{"I can do it"}) = P(\text{"I"}) \times P(\text{"can"}|\text{"I"}) \times P(\text{"do"}|\text{"I can"}) \times P(\text{"it"}|\text{"I can do"})$$

Independence assumption

- Assume that each word depends only on the previous word.
 $P(\text{"do"}|\text{"I can"}) = P(\text{"do"}|\text{"can"})$
 $P(\text{"it"}|\text{"I can do"}) = P(\text{"it"}|\text{"do"})$
- Estimate those probabilities from real data by counting
 $P(\text{"it"}|\text{"do"}) = C(\text{"do it"})/C(\text{"do"})$
- Markov property: given the current word, the next word and past words are independent.

N-gram

- Sequence of n words
unigram
bigram
trigram
- Larger n: more information about the context.
- Smaller n: more instances n training data, better statistical estimates (more reliability)

Selecting an n

- Vocabulary = 20,000 word
- Bigrams: 400,000,000 bins
- Trigrams: 8,000,000,000,000 bins
