

Ngrams

Jean Mark Gawron

Linguistics
San Diego State University
gawron@mail.sdsu.edu
<http://www.rohan.sdsu.edu/~gawron>

February 5, 2020

A card deck

Let's say we have a deck of standard cards and we are interested in what suits and ranks we get when we draw a random card (returning the card to the deck after each experiment).

Our mathematical description is as follows:

- 1 We chose a *variable* — call it **card** — to denote the outcomes of our various card-drawing experiments.
- 2 There are 52 possible outcomes to an experiment:

$$\{\spadesuit A, \spadesuit K, \spadesuit Q, \dots\}$$

- 3 The variable card is a **random variable**
- 4 The important thing about a random variable is that it can take different **values**. For example:

$$\text{card} = \diamond J$$

- 5 The term usually comes up when we know something or want to know something about the **probabilities** of certain events.

Card deck (ctd.)

1 There are numerous classes of outcomes of interest:

- Suits: ♠, ♦, ♣, ♥
- Ranks: Ace, King, Queen, Jack, 10, ...

Random Variables

Let Ω be a set of outcomes (called the **sample space**): $\Omega = \{x_1, x_2, \dots, x_n\}$ and let χ be some random variable that takes values in Ω :

$$\chi = x_1, \chi = x_2, \dots, \chi = x_n$$

$P(\chi = x_j)$ is the probability that χ is equal to x_j :

$$0 \leq P(\chi = x_j) \leq 1 \\ \sum_{x_j \in \Omega} P(\chi = x_j) = 1$$

For any two events e_1, e_2 ,

$$P(e_1 \cup e_2) = \sum_{x_j \in e_1 \cup e_2} P(\chi = x_j)$$

Subsets of Ω are called **events**. The probability of an event e_1 is defined:

$$P(e_1) = \sum_{x_j \in e_1} P(\chi = x_j)$$

With e_1, e_2 disjoint,

$$P(e_1 \cup e_2) = P(e_1) + P(e_2)$$

Example

In our card-drawing example,

$$\Omega = \{\spadesuit A, \spadesuit K, \spadesuit Q, \dots\}$$

X the random variable, was **card**, and it took values in Ω

$$\text{card} = \spadesuit A, \text{card} = \heartsuit Q, \text{card} = \clubsuit 7, \dots$$

Assuming the cards are being dealt fairly:

$$P(\text{card} = \clubsuit 7) = \frac{1}{52} = 0.019$$

And similarly for any other card.

Suit events are subsets of Ω :

$$\clubsuit = \{\clubsuit A, \clubsuit K, \clubsuit Q, \dots\}$$

So too for **rank events**:

$$\text{king} = \{\clubsuit K, \heartsuit K, \diamondsuit K, \spadesuit K\}$$

Example, ctd.

$$\begin{aligned}P(\clubsuit) &= \sum_{x_i \in \clubsuit} P(\chi = x_i) = \frac{13}{52} = \frac{1}{4} = .25 \\P(\heartsuit) &= \sum_{x_i \in \heartsuit} P(\chi = x_i) = \frac{13}{52} = \frac{1}{4} = .25 \\P(\text{King}) &= \sum_{x_i \in \text{King}} P(\chi = x_i) = \frac{4}{52} = \frac{1}{13} = .077\end{aligned}$$

For disjoint events \clubsuit, \heartsuit , we get

$$P(\clubsuit \cup \heartsuit) = .25 + .25 = .5$$

Compare $P(\clubsuit \cup \text{king})$

$$\begin{aligned}P(\text{king} \cup \clubsuit) &\neq \frac{4}{52} + \frac{13}{52} = .077 + .25 = .327 \\P(\text{king} \cup \clubsuit) &= \frac{16}{52} = .308\end{aligned}$$

$$\begin{aligned}\clubsuit &= \{\clubsuit A, \clubsuit K, \clubsuit Q, \dots\} \\ \text{king} &= \{\clubsuit K, \heartsuit K, \diamondsuit K, \spadesuit K\}\end{aligned}$$

Example, ctd.

$$\begin{aligned}P(\clubsuit) &= \sum_{x_i \in \clubsuit} P(\chi = x_i) = \frac{13}{52} = \frac{1}{4} = .25 \\P(\heartsuit) &= \sum_{x_i \in \heartsuit} P(\chi = x_i) = \frac{13}{52} = \frac{1}{4} = .25 \\P(\text{King}) &= \sum_{x_i \in \text{King}} P(\chi = x_i) = \frac{4}{52} = \frac{1}{13} = .077\end{aligned}$$

For disjoint events \clubsuit, \heartsuit , we get

$$P(\clubsuit \cup \heartsuit) = .25 + .25 = .5$$

Compare $P(\clubsuit \cup \text{king})$

$$\begin{aligned}P(\text{king} \cup \clubsuit) &\neq \frac{4}{52} + \frac{13}{52} = .25 + .077 = .327 \\P(\text{king} \cup \clubsuit) &= \frac{16}{52} = .308\end{aligned}$$

$$\begin{aligned}\clubsuit &= \{\clubsuit A, \clubsuit K, \clubsuit Q, \dots\} \\ \text{king} &= \{\clubsuit K, \heartsuit K, \diamondsuit K, \spadesuit K\}\end{aligned}$$

Joint Probabilities

Consider $P(\clubsuit \cap \text{king}) = P(\clubsuit, \text{king})$

$$\begin{aligned}\clubsuit &= \{\clubsuit A, \clubsuit K, \clubsuit Q, \dots\} \\ \text{king} &= \{\clubsuit K, \heartsuit K, \diamondsuit K, \spadesuit K\} \\ \clubsuit \cap \text{king} &= \{\clubsuit K\} \\ P(\text{king} \cap \clubsuit) &= \frac{1}{52} = .019\end{aligned}$$

Estimating the probabilities of points in the sample space

Suppose we are talking about coin tosses

$$\Omega = \{ H, T \}$$

You know the answer is .5

How do we get that answer? We assume that all the points in Ω (cards in the deck, for example) had equal probability. Since there are 2 points here and the probabilities have to add to 1, that means

$$\hat{P}(X = \text{head}) = .5$$

But that assumes it's a fair coin.

Suppose you toss the coin 100 times and you get 45 Heads. Now we have hard evidence it's not a fair coin. Our sample size is 100. The count of heads events is 45. Using relative frequency to estimate the probability of heads, we get:

$$\hat{P}(X = \text{head}) = \frac{45}{100} = .45$$

That's good practice: Estimate the probabilities in the sample space with experiments. But is there a justification for using **relative frequency**?

Binomial Distribution

Using the coin from the previous slide. Let's assume the relative frequency estimate, $P(H) = .45$. Given **probability** for a head, what is the probability of getting exactly 45 heads on 100 tosses?

$$\begin{aligned}P(r | p) &= \binom{n}{r} p^r (1 - p)^{n-r} \\P(45 | p) &= \binom{100}{45} p^{45} (1 - p)^{55} \\&= 0.08\end{aligned}$$

Here p is the probability of heads, r the number of heads in the sample, and n the total sample size. This is called a **binomial distribution**, because there are only two outcomes.

Binomial Distribution

We have 6 coin flips (fair coin $p(H) = .5$). What's the probability that exactly 2 of them will be H?

$$\begin{aligned}L(p; x) &= \frac{n!}{x!(n-x)!} p^x (1-p)^{n-x} \\L(.5; 2) &= \frac{6!}{2!4!} * .5^2 * .5^4 \\L(.5; 2) &= 15 * .15625 \\&\propto .15625 \\&= \end{aligned}$$

A not so fair coin ($p(H) = .3$)

$$\begin{aligned}L(.4; 2) &= \frac{6!}{2!4!} * .4^2 * .6^4 \\&= 15 * .0576 \\&\propto .0576\end{aligned}$$

Python

```
from scipy.special import comb
def binomial (n, r, p):
    return comb(n, r) * (p**r) * (1-p)**(n - r)
```

```
>>> binomial(100, 45, .45)\
0.07998750025326758\
>>> binomial(100, 45, .55)\
0.04847429662643074\
>>> binomial(100, 45, ..35)\
0.009589759011793606
```

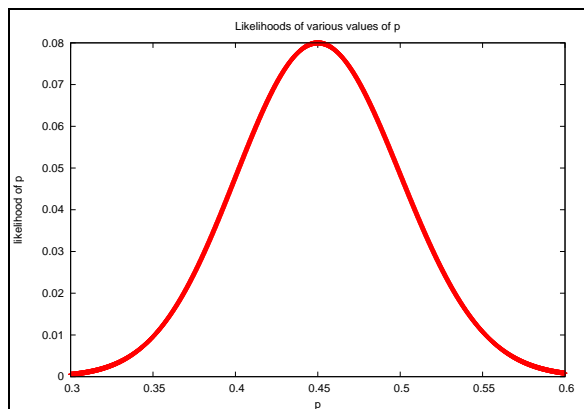
Maximum Likelihood Estimate (MLE)

Back to our experiment: 45 heads; 100 tosses.

We might consider a number of different values for the probability of heads with that coin (a number of different **models** for that coin), and ask for each, what is the likelihood of getting exactly 45 heads?

Using the formula for a binomial distribution, we can plot a graph whose x-axis is the probability of a head in each model, and whose y axis is that is the probability of getting 45 heads with that model. (the value on the y-axis is called the **likelihood**).

Likelihood Graph



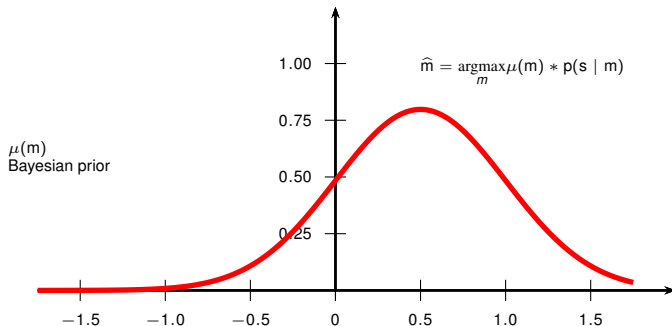
Likelihood: $p(s | m)$ (prob of sample given model)

MLE Drawbacks

- The probability of 45 heads in 100 throws is .08!
- 0.08 is not that high. The probability that a fair coin would produce this sample (that $p = 0.5$) is about 0.047, which is not that much lower.
- Try the experiment with a fair coin. Very rarely will you get 50 heads. The probability of getting exactly 50 heads with a fair coin is also about 0.08.
- The MLE may be the **best** guess of the true model, but it is still not all that likely to be right.
- Bigger sample size helps discriminate models better.
- Rare events are still hard to model well.

Alternative: Bayesian Estimate

The Bayesian estimate balances the likelihood of a sample given a model, $p(s | m)$, against some **prior** probability distribution over models, $\mu(m)$. If we have a prior belief that coins are fair, we choose a prior distribution favoring the fair coin model, with model probability dropping off rapidly as it moves away from fair:



Joint Distributions

Joint distributions can be defined in terms of two random variables χ and γ :

$$p(x, y) = P(\chi = x, \gamma = y)$$

So now we have probabilities for paired outcomes.

The marginal probability of outcome x is the sum of the probabilities of outcomes in which x is involved, which means summing over all the y 's x is paired with; and similarly for y :

$$p_{\chi}(x) = \sum_{\gamma} p(x, y) \quad p_{\gamma}(y) = \sum_{\chi} p(x, y)$$

Conditional Probabilities

Conditional Probabilities are defined by the **Chain Rule**:

$$(a) \quad p_{X|Y}(x | y) = \frac{p(x, y)}{p(y)}$$

$$(b) \quad p(x, y) = p_{X|Y}(x | y) * P(y)$$

Conditional Prob Distributions

Each way of fixing y defines a probability distribution:

$$\begin{aligned}\sum_x p(x \mid \gamma = y) &= \sum_x \frac{p(x, y)}{p(y)} \\ &= \frac{\sum_x p(x, y)}{p_\gamma(y)} = \frac{p(y)}{p(y)} \\ &= 1\end{aligned}$$

So $p(x \mid \gamma = y)$ is a probability distribution.

Chain Rule: review

$$(1) \quad P(x_1, x_2) = P(x_1) * P(x_2 | x_1)$$

$$(n-1) \quad P(x_1, x_2, \dots, x_n) = P(x_1) * P(x_2 | x_1) \cdots * P(x_n | x_1, \dots, x_{n-1})$$

Chain Rule: review

$$(1) \quad P(x_1, x_2) = P(x_1) * P(x_2 | x_1)$$

$$(2) \quad P(\boxed{x_1, x_2}, x_3) = \boxed{P(x_1, x_2)} * P(x_3 | \boxed{x_1, x_2})$$

⋮

$$(n-1) \quad P(x_1, x_2, \dots, x_n) = P(x_1) * P(x_2 | x_1) \cdots * P(x_n | x_1, \dots, x_{n-1})$$

Chain Rule: review

$$(1) \quad P(x_1, x_2) = P(x_1) * P(x_2 | x_1)$$

$$(2) \quad P(\boxed{x_1, x_2}, x_3) = \boxed{P(x_1) * P(x_2 | x_1)} * P(x_3 | \boxed{x_1, x_2})$$

⋮

$$(n-1) \quad P(x_1, x_2, \dots, x_n) = P(x_1) * P(x_2 | x_1) \cdots * P(x_n | x_1, \dots, x_{n-1})$$

Chain Rule: review

$$(1) \quad P(x_1, x_2) = P(x_1) * P(x_2 | x_1)$$

$$(2) \quad P(x_1, x_2, x_3) = P(x_1) * P(x_2 | x_1) * P(x_3 | x_1, x_2)$$

$$(3) \quad P(x_1, x_2, x_3, x_4) = P(x_1, x_2, x_3) * P(x_4 | x_1, x_2, x_3)$$

⋮

$$(n-1) \quad P(x_1, x_2, \dots, x_n) = P(x_1) * P(x_2 | x_1) \cdots * P(x_n | x_1, \dots, x_{n-1})$$

Chain Rule: review

$$\begin{aligned} (1) \quad & P(x_1, x_2) = P(x_1) * P(x_2 | x_1) \\ (2) \quad & P(\boxed{x_1, x_2}, x_3) = \boxed{P(x_1) * P(x_2 | x_1)} * P(x_3 | \boxed{x_1, x_2}) \\ (3) \quad & P(\boxed{x_1, x_2, x_3}, x_4) = \boxed{P(x_1) * P(x_2 | x_1) * P(x_3 | x_1, x_2)} * P(x_4 | \boxed{x_1, x_2, x_3}) \\ & \vdots \\ (n-1) \quad & P(x_1, x_2, \dots, x_n) = P(x_1) * P(x_2 | x_1) \cdots * P(x_n | x_1, \dots, x_{n-1}) \end{aligned}$$

Conditional Probs and Relative frequencies

The chain rule tells us how to estimate conditional probabilities using using relative frequencies:

$$\begin{aligned} P(a | b) &= \frac{P(a, b)}{P(b)} \approx \frac{\frac{|a, b|}{|S|}}{\frac{|b|}{|S|}} \\ &= \frac{|a, b|}{|b|} \end{aligned}$$

Instead of dividing the frequency by the size of the entire sample as we do for $P(a,b)$, we divide by the size of the sample restricted to b .

Predicting words

We are interested in predicting the n th word given some history of $n - 1$ words:

Consider a history of 6 words:

I want to make a long-distance _____

Motivations

Jurasky & Martin (pp. 83,84)

Apps

Speech and handwriting recognition:

how do you wreck a nice beach?

how do you recognize speech?

Choosing among the outputs of a Chinese statistical MT system:

he briefed to reporters on the chief contents of of the statement.

he briefed reporters on the chief contents of of the statement.

he briefed to reporters on the main contents of of the statement.

he briefed reporters on the main contents of of the statement.

Spelling correction: errors that are valid words:

They are leaving in about 15 **minuets** to go to her house.

The design **an** construction of the system will take more than a year.

More subfields/apps

- 1 augmentative communication
- 2 authorship identification
- 3 predictive text input (cell phone texting)
- 4 part-of-speech tagging
- 5 language generation
- 6 word similarity

The problem

We start with the problem: How do we formulate the probability of a *string of words of length n*?

We use the chain rule for the joint probability of n events:

$$P(x_1, x_2, \dots, x_n) = P(x_1) * P(x_2 | x_1) \cdots * P(x_n | x_1, \dots, x_{n-1})$$

We abbreviate word sequences

$$w_1, w_2, \dots, w_n$$

as w_1^n .

We think of n-word word sequence w_1^n as a joint event consisting of word 1 occurring in the first position, word 2 occurring in the second position, and so on. So the Chain rule applies!

Word sequences

It follows immediately that:

$$P(w_1^n) = P(w_1) * P(w_2 | w_1) * \dots * P(w_n | w_1^{n-1})$$

The problem with this formulation of the probability can be seen by looking at the last term:

$$\text{Prob}(w_n | w_1^{n-1})$$

This is the probability of the last word given the entire sequence of words before it.

How would we compute such a thing?

Sparseness

How to compute:

$$\text{Prob}(w_n \mid w_1^{n-1})?$$

It's easy!

$$\frac{|w_1^n|}{|w_1^{n-1}|}$$

For instance:

$$\frac{| \text{I want to make a long-distance call} |}{| \text{I want to make a long-distance} |}$$

To do this right we need some corpus large enough to give us a representative sampling of the 5-word string *I want to make a long-distance* in which there is hopefully a representative sample of the 6-word string *I want to make a long-distance call*.

Not enough data

The problem is that there isn't enough data in the world to get such representative samples in most cases for even moderately small n .

Consider a very small n . Consider Shakespeare.

Word token Count	884,647	
Word form Types	29,066	including lots of proper nouns
Number of bigram types	$29,066^2 = 844$ million	
Number of bigram tokens	884,647	

In any corpus of this size, we're very unlikely to see most of the rarer bigrams.

Shakespeare's trigrams

Word token Count	884,647
Word form Types	29,066
Number of trigram types	$29,066^3 = 25,636,000,000,000$
Number of trigram tokens	884,647

We get only a vanishing small sample of the entire space of trigrams. We're likely to encounter only the most common ones.

Other problems

- 1 After training on 1.5 million words from IBM Laser Patent Text Corpus, Bahl, et al (1983) reported that 23% of the trigrams in unseen data were new!
- 2 Ughh! This is “the long tail of language”
- 3 Still worse, we need samples build out of independent events; In fact trigrams in a text aren't independent:
- 4 Each partially overlaps (and partly helps determine) the next one:

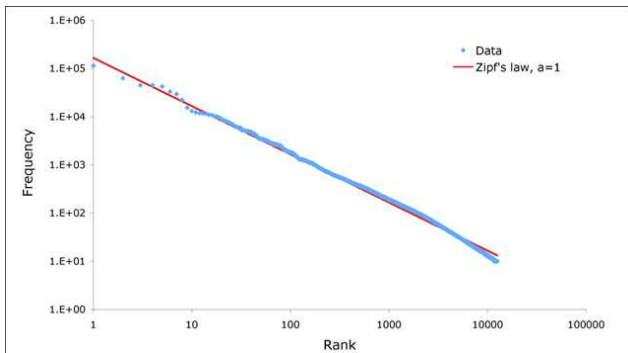
I	want	to			
	want	to	make		
		to	make	a	
			make	a	long
- 5 Content words tend to clump (a word's appearance in a document is one of the best predictors of its later appearance).
- 6 Auctorial tendencies (the Shakespeare corpus).

Zipf's Law

"The long tail of language"

- 1 Frequency of a word: How many times it occurs in a sample of a certain size
- 2 Rank: Most frequent word has rank 1; least frequent in a vocab of 20K has rank 20K.
- 3 Zipf's Law:

$$f \cdot r = k$$



Zipf's Law II

Word Frequency	Frequency of Frequency
1	3993
2	1292
3	664
4	410
5	243
6	199
7	172
8-10	304
11- 50	540
51-100	99
≥ 100	102
	<hr/> 8018

Almost all words are rare.
— Manning and Schuetze (1999)

Frequency of Frequencies in *Tom Sawyer*
(Manning & Schuetze, Table 1.2, p. 22)

Zipf's Law III

Rank	Word	Frequency	Use
1	the	3332	determiner (article)
2	and	2972	conjunction
3	a	1775	determiner
4	to	1725	preposition, infinitive marker
5	of	1440	preposition
6	was	1161	auxiliary verb
7	it	1027	personal/expletive pronoun
8	in	906	preposition
9	that	877	complementizer, demonstrative
10	he	877	personal pronoun
11	I	783	personal pronoun
12	his	772	possessive pronoun
13	you	686	personal pronoun
14	Tom	679	proper noun
	18229	out of 71,370 (25.6%)	Common words in <i>Tom Sawyer</i> (Manning & Schuetze, Table 1.1, p. 21)

A simplifying Assumption

Reduce the amount of history looked at (a Markov assumption)

$$P(w_n | w_1, \dots, w_{n-1}) = P(w_n | w_{n-1})$$

The chain rule calculation goes from:

$$P(w_1^n) = P(w_1) * P(w_2 | w_1) * P(w_3 | w_1, w_2) \dots P(w_n | w_1^{n-1})$$

to

$$P(w_1^n) = P(w_1) * P(w_2 | w_1) * P(w_3 | w_2) \dots P(w_n | w_{n-1})$$

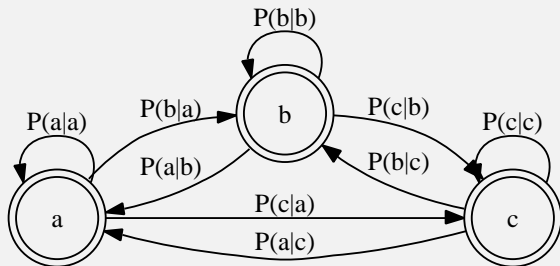
Markov Assumptions

The assumption that the probability of an event is determined by some finite amount of history is called a **Markov Assumption**

A consequence of the Markov assumption is that a Probability model can be completely described by a very simple kind of probabilistic finite-state automaton called a **Markov Chain**

A Simple bigram Markov chain

Bigram for a simple language with a vocabulary of three words: a, b, and c.

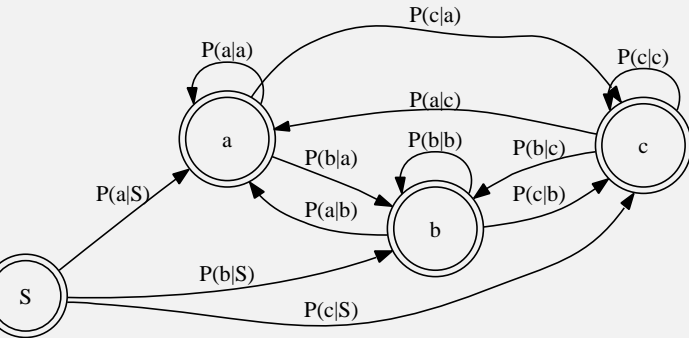


String probability

String probabilities for abc and cba.

$$P(abc) = P(a | S) * P(b | c) * P(c | b)$$

$$P(cba) = P(c | S) * P(b | c) * P(a | b)$$



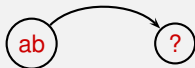
Markov Properties

- Markov models can encode dependencies on histories of any finite length – at the cost of more states. Consider a vocab of size 3:

1 bigram: 3 words = 3 possible histories = 3 states

2 trigram $3 * 3$ histories = 9 states

- In a trigram model, where should we go if we're in state "ab" and we see a "b"?



- Markov Chains: For any given emission, there is exactly one path through the network
- Hidden Markov Models: For any given emission there are an arbitrary number of paths through the network
- Ngram models are Markov chains.

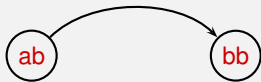
Markov Properties

- Markov models can encode dependencies on histories of any finite length – at the cost of more states. Consider a vocab of size 3:

1 bigram: 3 words = 3 possible histories = 3 states

2 trigram $3 * 3$ histories = 9 states

- In a trigram model, where should we go if we're in state "ab" and we see a "b"?



- Markov Chains: For any given emission, there is exactly one path through the network
- Hidden Markov Models: For any given emission there are an arbitrary number of paths through the network
- Ngram models are Markov chains.

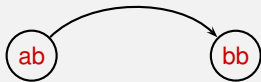
Markov Properties

- Markov models can encode dependencies on histories of any finite length – at the cost of more states. Consider a vocab of size 3:

1 bigram: 3 words = 3 possible histories = 3 states

2 trigram $3 * 3$ histories = 9 states

- In a trigram model, where should we go if we're in state "ab" and we see a "b"? And what is the probability? $P(b | b)$? $P(b | a)$? $P(b | ab)$?



- Markov Chains: For any given emission, there is exactly one path through the network
- Hidden Markov Models: For any given emission there are an arbitrary number of paths through the network
- Ngram models are Markov chains.

Markov Properties

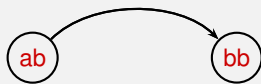
- Markov models can encode dependencies on histories of any finite length – at the cost of more states. Consider a vocab of size 3:

1 bigram: 3 words = 3 possible histories = 3 states

2 trigram $3 * 3$ histories = 9 states

- In a trigram model, where should we go if we're in state "ab" and we see a "b"? And what is the probability? $P(b | b)$? $P(b | a)$? $P(b | ab)$?

$P(b | ab)$



- Markov Chains: For any given emission, there is exactly one path through the network
- Hidden Markov Models: For any given emission there are an arbitrary number of paths through the network
- Ngram models are Markov chains.