A Markov chain is a special sort of belief network:

What probabilities need to be specified?

P(S_0) specifies initial conditions
P(S_{i+1} | S_i) specifies the dynamics

What independence assumptions are made?
P(S_{i+1} | S_0, \ldots, S_i) = P(S_{i+1} | S_i).

Often S_t represents the state at time t. Intuitively S_t conveys all of the information about the history that can affect the future states.

"The future is independent of the past given the present."
A Markov chain is a special sort of belief network:

- $P(S_0)$ specifies initial conditions
- $P(S_{i+1}|S_i)$ specifies the dynamics

What independence assumptions are made?

- $P(S_{i+1}|S_0, \ldots, S_i) = P(S_{i+1}|S_i)$

Intuitively, $S_t$ represents the state at time $t$. Intuitively, $S_t$ conveys all of the information about the history that can affect the future states.

"The future is independent of the past given the present."
A Markov chain is a special sort of belief network:

What probabilities need to be specified?
- $P(S_0)$ specifies initial conditions
- $P(S_{i+1} | S_i)$ specifies the dynamics

What independence assumptions are made?

The future is independent of the past given the present.
A Markov chain is a special sort of belief network:

What probabilities need to be specified?
- $P(S_0)$ specifies initial conditions
- $P(S_{i+1}|S_i)$ specifies the dynamics

What independence assumptions are made?
- $P(S_{i+1}|S_0, \ldots, S_i) = P(S_{i+1}|S_i)$. 
A **Markov chain** is a special sort of belief network:

What probabilities need to be specified?

- $P(S_0)$ specifies initial conditions
- $P(S_{i+1}|S_i)$ specifies the dynamics

What independence assumptions are made?

- $P(S_{i+1}|S_0, \ldots, S_i) = P(S_{i+1}|S_i)$.
- Often $S_t$ represents the **state** at time $t$. Intuitively $S_t$ conveys all of the information about the history that can affect the future states.
A Markov chain is a special sort of belief network:

What probabilities need to be specified?

- $P(S_0)$ specifies initial conditions
- $P(S_{i+1}|S_i)$ specifies the dynamics

What independence assumptions are made?

- $P(S_{i+1}|S_0, \ldots, S_i) = P(S_{i+1}|S_i)$.
- Often $S_t$ represents the state at time $t$. Intuitively $S_t$ conveys all of the information about the history that can affect the future states.
- “The future is independent of the past given the present.”
A stationary Markov chain is when for all $i > 0$, $i' > 0$, $P(S_{i+1} | S_i) = P(S_{i'+1} | S_{i'}).$

We specify $P(S_0)$ and $P(S_{i+1} | S_i)$.

- Simple model, easy to specify
- Often the natural model
- The network can extend indefinitely
A distribution over states, $P$ is a stationary distribution if for each state $s$, $P(S_{i+1} = s) = P(S_i = s)$. 

Every Markov chain has a stationary distribution. A Markov chain is ergodic if, for any two states $s_1$ and $s_2$, there is a non-zero probability of eventually reaching $s_2$ from $s_1$. A Markov chain is periodic if there is a strict temporal regularity in visiting states. A Markov chain has period $n$ iff each state is only visited at time $t$ if $t \mod n = m$ for some $m$. An ergodic and aperiodic Markov chain has a unique stationary distribution $P$ and $P(s) = \lim_{i \to \infty} P(S_i = s)$ — equilibrium distribution.
A distribution over states, $P$ is a **stationary distribution** if for each state $s$, $P(S_{i+1}=s) = P(S_i=s)$.

Every Markov chain has a stationary distribution.
Stationary Distribution

- A distribution over states, $P$ is a stationary distribution if for each state $s$, $P(S_{i+1}=s) = P(S_i=s)$.
- Every Markov chain has a stationary distribution.
- A Markov chain is ergodic if, for any two states $s_1$ and $s_2$, there is a non-zero probability of eventually reaching $s_2$ from $s_1$. 

©D. Poole and A. Mackworth 2019
Artificial Intelligence, Lecture 8.5
Stationary Distribution

- A distribution over states, $P$ is a stationary distribution if for each state $s$, $P(S_{i+1} = s) = P(S_i = s)$.
- Every Markov chain has a stationary distribution.
- A Markov chain is ergodic if, for any two states $s_1$ and $s_2$, there is a non-zero probability of eventually reaching $s_2$ from $s_1$.
- A Markov chain is periodic if there is a strict temporal regularity in visiting states. A Markov chain has period $n$ iff each state is only visited at time $t$ if $t \mod n = m$ for some $m$. 
A distribution over states, $P$ is a **stationary distribution** if for each state $s$, $P(S_{i+1} = s) = P(S_i = s)$.

Every Markov chain has a stationary distribution.

A Markov chain is **ergodic** if, for any two states $s_1$ and $s_2$, there is a non-zero probability of eventually reaching $s_2$ from $s_1$.

A Markov chain is **periodic** if there is a strict temporal regularity in visiting states. A Markov chain has period $n$ iff each state is only visited at time $t$ if $t \mod n = m$ for some $m$.

An ergodic and aperiodic Markov chain has a unique stationary distribution $P$ and $P(s) = \lim_{i \rightarrow \infty} P(S_i = s)$ — equilibrium distribution
Consider the Markov chain:

- Domain of $S_i$ is the set of all web pages
- $P(S_0)$ is uniform; $P(S_0 = p_j) = 1/N$

\[ P(S_{i+1} = p_j \mid S_i = p_k) = \frac{(1 - d)}{N} + d \times \begin{cases} 
\frac{1}{n_k} & \text{if } p_k \text{ links to } p_j \\
\frac{1}{N} & \text{if } p_k \text{ has no links} \\
0 & \text{otherwise}
\end{cases} \]

where there are $N$ web pages and $n_k$ links from page $p_k$

- $d \approx 0.85$ is the probability someone keeps surfing web
Consider the Markov chain:

- Domain of $S_i$ is the set of all web pages
- $P(S_0)$ is uniform; $P(S_0 = p_j) = 1/N$

$$P(S_{i+1} = p_j \mid S_i = p_k) = (1 - d)/N + d \times \begin{cases} 1/n_k & \text{if } p_k \text{ links to } p_j \\ 1/N & \text{if } p_k \text{ has no links} \\ 0 & \text{otherwise} \end{cases}$$

where there are $N$ web pages and $n_k$ links from page $p_k$

- $d \approx 0.85$ is the probability someone keeps surfing web
- This Markov chain converges to a distribution over web pages (original $P(S_i)$ for $i = 52$ for 322 million links):

**Pagerank** - basis for Google’s initial search engine
A Hidden Markov Model (HMM) is a belief network:

The probabilities that need to be specified:
A Hidden Markov Model (HMM) is a belief network:

The probabilities that need to be specified:

- $P(S_0)$ specifies initial conditions
- $P(S_{i+1} | S_i)$ specifies the dynamics
- $P(O_i | S_i)$ specifies the sensor model
Filtering:

\[ P(S_i | o_1, \ldots, o_i) \]

What is the current belief state based on the observation history?
Filtering

Filtering:

\[ P(S_i|o_1, \ldots, o_i) \]

What is the current belief state based on the observation history?

\[ P(S_i|o_1, \ldots, o_i) \propto P(o_i|S_i o_1, \ldots, o_{i-1}) P(S_i|o_1, \ldots, o_{i-1}) \]

\[ = ??? \sum_{S_{i-1}} P(S_i S_{i-1}|o_1, \ldots, o_{i-1}) \]

\[ = ??? \]
Filtering:

\[ P(S_i|o_1, \ldots, o_i) \]

What is the current belief state based on the observation history?

\[ P(S_i|o_1, \ldots, o_i) \propto P(o_i|S_i o_1, \ldots, o_{i-1}) P(S_i|o_1, \ldots, o_{i-1}) \]

\[ = \sum_{S_{i-1}} P(S_i S_{i-1}|o_1, \ldots, o_{i-1}) \]

\[ = \sum_{S_{i-1}} P(S_i S_{i-1}|o_1, \ldots, o_{i-1}) \]

- Observe \( O_0 \), query \( S_0 \).
Filtering:

\[ P(S_i | o_1, \ldots, o_i) \]

What is the current belief state based on the observation history?

\[ P(S_i | o_1, \ldots, o_i) \propto P(o_i | S_i o_1, \ldots, o_{i-1}) P(S_i | o_1, \ldots, o_{i-1}) \]

\[ = \sum_{S_{i-1}} P(S_i S_{i-1} | o_1, \ldots, o_{i-1}) \]

\[ = \ldots \]

- Observe \( O_0 \), query \( S_0 \).
- then observe \( O_1 \), query \( S_1 \).
Filtering:

\[ P(S_i|o_1, \ldots, o_i) \]

What is the current belief state based on the observation history?

\[ P(S_i|o_1, \ldots, o_i) \propto P(o_i|S_i, o_1, \ldots, o_{i-1})P(S_i|o_1, \ldots, o_{i-1}) \]

\[ = \sum_{S_{i-1}} P(S_i, S_{i-1}|o_1, \ldots, o_{i-1}) \]

\[ = ??? \]

- Observe \( O_0 \), query \( S_0 \).
- then observe \( O_1 \), query \( S_1 \).
- then observe \( O_2 \), query \( S_2 \).
- ...
Example: localization

- Suppose a robot wants to determine its location based on its actions and its sensor readings: Localization
- This can be represented by the augmented HMM:
Example localization domain

- Circular corridor, with 16 locations:

- Doors at positions: 2, 4, 7, 11.
- Noisy Sensors
- Stochastic Dynamics
- Robot starts at an unknown location and must determine where it is.
Example Sensor Model

- \( P(\text{Observe Door} \mid \text{At Door}) = 0.8 \)
- \( P(\text{Observe Door} \mid \text{Not At Door}) = 0.1 \)
Example Dynamics Model

- \( P(\text{loc}_{t+1} = L|\text{action}_t = \text{goRight} \land \text{loc}_t = L) = 0.1 \)
- \( P(\text{loc}_{t+1} = L + 1|\text{action}_t = \text{goRight} \land \text{loc}_t = L) = 0.8 \)
- \( P(\text{loc}_{t+1} = L + 2|\text{action}_t = \text{goRight} \land \text{loc}_t = L) = 0.074 \)
- \( P(\text{loc}_{t+1} = L'|\text{action}_t = \text{goRight} \land \text{loc}_t = L) = 0.002 \) for any other location \( L' \).
  
  ▶ All location arithmetic is modulo 16.
  ▶ The action \text{goLeft} works the same but to the left.
**Example:** we can combine information from a light sensor and the door sensor **Sensor Fusion**

$S_t$ robot location at time $t$
$D_t$ door sensor value at time $t$
$L_t$ light sensor value at time $t$
Naive Bayes Classifier: User’s request for help

$H$ is the help page the user is interested in.

$H$ is the help page the user is interested in.
Naive Bayes Classifier: User’s request for help

$H$ is the help page the user is interested in.
What probabilities are required?
Naive Bayes Classifier: User’s request for help

H is the help page the user is interested in. What probabilities are required?

- \( P(h_i) \) for each help page \( h_i \). The user is interested in one best web page, so \( \sum_i P(h_i) = 1 \).
Naive Bayes Classifier: User’s request for help

$H$ is the help page the user is interested in.

What probabilities are required?

- $P(h_i)$ for each help page $h_i$. The user is interested in one best web page, so $\sum_i P(h_i) = 1$.
- $P(w_j \mid h_i)$ for each word $w_j$ given page $h_i$. There can be multiple words used in a query.
Naive Bayes Classifier: User’s request for help

**H** is the help page the user is interested in. What probabilities are required?

- $P(h_i)$ for each help page $h_i$. The user is interested in one best web page, so $\sum_i P(h_i) = 1$.
- $P(w_j \mid h_i)$ for each word $w_j$ given page $h_i$. There can be multiple words used in a query.
- Given a help query:
Naive Bayes Classifier: User’s request for help

\[ H \]

"able" \hspace{2cm} "absent" \hspace{2cm} "add" \hspace{2cm} \ldots \hspace{2cm} "zoom"

\( H \) is the help page the user is interested in.

What probabilities are required?

- \( P(h_i) \) for each help page \( h_i \). The user is interested in one best web page, so \( \sum_i P(h_i) = 1 \).
- \( P(w_j \mid h_i) \) for each word \( w_j \) given page \( h_i \). There can be multiple words used in a query.
- Given a help query: condition on the words in the query and display the most likely help page.
Sentence: \( w_1, w_2, w_3, \ldots \)

Set-of-words model:

- “a”
- “aardvark”
- ...
- “zzz”

Each variable is Boolean: \( true \) when word is in the sentence and \( false \) otherwise.

What probabilities are provided?

- \( P(\text{a}) \), \( P(\text{aardvark}) \), \( P(\text{zzz}) \)
Sentence: \( w_1, w_2, w_3, \ldots \)

Set-of-words model:

- Each variable is Boolean: \( true \) when word is in the sentence and \( false \) otherwise.
- What probabilities are provided?

---

“a”  “aardvark”  “zzz”
Simple Language Models: set-of-words

Sentence: $w_1, w_2, w_3, \ldots$

Set-of-words model:

- Each variable is Boolean: true when word is in the sentence and false otherwise.
- What probabilities are provided?
  - $P(\text{"a"})$, $P(\text{"aardvark"})$, $\ldots$, $P(\text{"zzz"})$
Sentence: \( w_1, w_2, w_3, \ldots \)

Set-of-words model:

- Each variable is Boolean: \textit{true} when word is in the sentence and \textit{false} otherwise.
- What probabilities are provided?
  - \( P(“a”) \), \( P(“aardvark”) \), \ldots, \( P(“zzz”) \)
- How do we condition on the question “how can I phone my phone”?
Sentence: $w_1, w_2, w_3, \ldots, w_n$.

Bag-of-words or unigram:

- Domain of each variable is the set of all words.
Simple Language Models: bag-of-words

Sentence: \( w_1, w_2, w_3, \ldots, w_n \).

Bag-of-words or unigram:

- Domain of each variable is the set of all words.
- What probabilities are provided?
Simple Language Models: bag-of-words

Sentence: $w_1, w_2, w_3, \ldots, w_n$.

Bag-of-words or unigram:

- Domain of each variable is the set of all words.
- What probabilities are provided?
  - $P(w_i)$ is a distribution over words for each position
Simple Language Models: bag-of-words

Sentence: $w_1, w_2, w_3, \ldots, w_n$. 
**Bag-of-words or unigram:**

$W_1 \ldots W_2 \ldots W_3 \ldots W_n$

- Domain of each variable is the set of all words.
- What probabilities are provided?
  - $P(w_i)$ is a distribution over words for each position
- How do we condition on the question “how can I phone my phone”? 

©D. Poole and A. Mackworth 2019
Artificial Intelligence, Lecture 8.5
14 / 23
Simple Language Models: bigram

Sentence: $w_1, w_2, w_3, \ldots, w_n$.

bigram:

![Diagram of bigram model]

- Domain of each variable is the set of all words.
Simple Language Models: bigram

Sentence: $w_1, w_2, w_3, \ldots, w_n$.  

bigram:

- Domain of each variable is the set of all words.
- What probabilities are provided?
Simple Language Models: bigram

Sentence: $w_1, w_2, w_3, \ldots, w_n$.

bigram:

- Domain of each variable is the set of all words.
- What probabilities are provided?
  - $P(w_i|w_{i-1})$ is a distribution over words for each position given the previous word
Sentence: \( w_1, w_2, w_3, \ldots, w_n \).

bigram:

\[ \begin{array}{cccc}
W_1 & \rightarrow & W_2 & \rightarrow & W_3 & \rightarrow & \ldots & \rightarrow & W_n \\
\end{array} \]

- Domain of each variable is the set of all words.
- What probabilities are provided?
  - \( P(w_i|w_{i-1}) \) is a distribution over words for each position given the previous word
- How do we condition on the question “how can I phone my phone”? 

\[\text{\textcopyright D. Poole and A. Mackworth 2019}\]

Artificial Intelligence, Lecture 8.5
Simple Language Models: trigram

Sentence: \(w_1, w_2, w_3, \ldots, w_n\).

trigram:

\[W_2 \ldots W_3 W_n W_1 W_4\]

Domain of each variable is the set of all words.
Simple Language Models: trigram

Sentence: \( w_1, w_2, w_3, \ldots, w_n \).

trigram:

Domain of each variable is the set of all words.
What probabilities are provided?
Simple Language Models: trigram

Sentence: \( w_1, w_2, w_3, \ldots, w_n \).

trigram:

![Diagram of trigram model](image)

Domain of each variable is the set of all words.

What probabilities are provided?

- \( P(w_i|w_{i-1}, w_{i-2}) \)

N-gram

- \( P(w_i|w_{i-1}, \ldots w_{i-n+1}) \) is a distribution over words given the previous \( n - 1 \) words
From: Google Books Ngram Viewer
(https://books.google.com/ngrams)
domain($W_i$) = \{"a", "aarvark", \ldots, "zzz", "\bot", "?"\} \\
domain(L_{ji}) = \{"a", "b", "c", \ldots, "z", "1", "2", \ldots \}
Beyond N-grams

- A person with a big hairy cat drank the cold milk.
- Who or what drank the milk?
Beyond N-grams

- A *person with a big hairy cat drank the cold milk.*
- Who or what drank the milk?

Simple syntax diagram:

```
s
  np  vp
    np
      np
        a
        person
      pp
        with
        np
          a
          big
          hairy
          cat
        np
          drank
            np
              the
              cold
              milk
```
Topic Model

- tools
- finance
- fish
- food

- "bolt"
- "nut"
- "salmon"
- "shark"
- "tuna"

Topics

Words
Topic Model

- **tools**
  - "bolt"
- **food**
  - "nut"
  - "tuna"

**topics**

**words**
Google’s rephil

900,000 topics
350,000,000 links
12,000,000 words

"aardvark" "Aaron's beard" "zzz"
Deep Belief Networks