COMP4121 Advanced Algorithms

Aleks Ignjatović

School of Computer Science and Engineering
University of New South Wales Sydney

The Hidden Markov Models and the Viterbi Algorithm and its applications

The Problem of Speech Recognition

We start with an example, the problem of speech recognition.

The Problem of Speech Recognition

We start with an example, the problem of speech recognition.

- A **phoneme** is an “ele[illegible]article” of human speech, so to speak.

The Problem of Speech Recognition

We start with an example, the problem of speech recognition.

- A **phoneme** is an “ele[illegible]article” of human speech, so to speak.
- Spoken words of a par[illegible]uage are understood by a listener by parsing them into a sequence [illegible]s.

The Problem of Speech Recognition

We start with an example, the problem of speech recognition.

- A **phoneme** is an “ele article” of human speech, so to speak.
- Spoken words of a par uage are understood by a listener by parsing them into a sequence s.
- There is no general consensus what exactly the phonemes are, even for a specific language such as English.

The Problem of Speech Recognition

We start with an example, the problem of speech recognition.

- A **phoneme** is an “elementary particle” of human speech, so to speak.
- Spoken words of a particular language are understood by a listener by parsing them into a sequence of phonemes.
- There is no general consensus what exactly the phonemes are, even for a specific language such as English.
- We can think of them as the basic sounds of a language, which have the property that replacing one phoneme in a word with a different phoneme would change the meaning of the word.

We start with an example, the problem of speech recognition.

- A **phoneme** is an “elementary particle” of human speech, so to speak.
- Spoken words of a particular language are understood by a listener by parsing them into a sequence of phonemes.
- There is no general consensus what exactly the phonemes are, even for a specific language such as English.
- We can think of them as the basic sounds of a language, which have the property that replacing one phoneme in a word with a different phoneme would change the meaning of the word.
- For example, if you replace the last phoneme /s/ in ‘kiss” with phoneme /l/ you obtain a different word, “kill”.

We start with an example, the problem of speech recognition.

- A **phoneme** is an “elementary particle” of human speech, so to speak.
- Spoken words of a particular language are understood by a listener by parsing them into a sequence of phonemes.
- There is no general consensus what exactly the phonemes are, even for a specific language such as English.
- We can think of them as the basic sounds of a language, which have the property that replacing one phoneme in a word with a different phoneme would change the meaning of the word.
- For example, if you replace the last phoneme /s/ in “kiss” with phoneme /l/ you obtain a different word, “kill”.
- However, we identify variants of the same sound as a single phoneme, for example the “k” sound in words “cat”, “kit” and “skill” are considered all to be slight variants of a single basic phoneme /k/, and replacing one such variant with another would not cause the listener to mistakenly understand the word being pronounced.

The Problem of Speech Recognition

- Humans are very good at understanding speech, but machines got good at that only quite recently.

The Problem of Speech Recognition

- Humans are very good at understanding speech, but machines got good at that only quite recently.
- So let us now consider being spoken, for example "I love algorithms".

The Problem of Speech Recognition

- Humans are very good at understanding speech, but machines got good at that only quite recently.
- So let us now consider being spoken, for example “I love algorithms”.
- We understand this se breaking the speech into constituent phonemes.

The Problem of Speech Recognition

- Humans are very good at understanding speech, but machines got good at that only quite recently.
- So let us now consider [illegible] being spoken, for example “I love algorithms”.
- We understand this se[illegible]reaking the speech into constituent phonemes.
- I tried to reproduce the International Phonetic Alphabet (IPA) transcription in LaTeX, but could only get an approximation; so the utterance of “I love algorithms” would look approximately like this:

aI lav algorIθmz .

The Problem of Speech Recognition

- Humans are very good at understanding speech, but machines got good at that only quite recently.
- So let us now consider [illegible] being spoken, for example "I love algorithms".
- We understand this se[illegible]breaking the speech into constituent phonemes.
- I tried to reproduce the International Phonetic Alphabet (IPA) transcription in LaTeX, but could only get an approximation; so the utterance of "I love algorithms" would look approximately like this:

  aI lav algorIθmz .

- Neither humans nor machines have a direct access to the sequence of the phonemes spoken.

The Problem of Speech Recognition

- Humans are very good at understanding speech, but machines got good at that only quite recently.
- So let us now consider [illegible] being spoken, for example "I love algorithms".
- We understand this se[illegible]breaking the speech into constituent phonemes.
- I tried to reproduce the International Phonetic Alphabet (IPA) transcription in LaTeX, but could only get an approximation; so the utterance of "I love algorithms" would look approximately like this:

  aɪ lav algorɪθmz .

- Neither humans nor machines have a direct access to the sequence of the phonemes spoken.
- Humans have access only to the resulting acoustic waves produced, as detected by the hearing organs.

The Problem of Speech Recognition

- Humans are very good at understanding speech, but machines got good at that only quite recently.
- So let us now consider a sentence being spoken, for example "I love algorithms".
- We understand this sentence by breaking the speech into constituent phonemes.
- I tried to reproduce the International Phonetic Alphabet (IPA) transcription in LaTeX, but could only get an approximation; so the utterance of "I love algorithms" would look approximately like this:

$$\texttt{aI lav algorI}\theta\texttt{mz}.$$

- Neither humans nor machines have a direct access to the sequence of the phonemes spoken.
- Humans have access only to the resulting acoustic waves produced, as detected by the hearing organs.
- Machines only have access to the resulting digitally sampled acoustic waveforms.

The Problem of Speech Recognition

- Humans are very good at understanding speech, but machines got good at that only quite recently.
- So let us now consider a sentence being spoken, for example "I love algorithms".
- We understand this sentence by breaking the speech into constituent phonemes.
- I tried to reproduce the International Phonetic Alphabet (IPA) transcription in LaTeX, but could only get an approximation; so the utterance of "I love algorithms" would look approximately like this:

  `aI lav algorIθmz .`

- Neither humans nor machines have a direct access to the sequence of the phonemes spoken.
- Humans have access only to the resulting acoustic waves produced, as detected by the hearing organs.
- Machines only have access to the resulting digitally sampled acoustic waveforms.
- From such waveforms they have to "decide" what phonemes were spoken.

The Problem of Speech Recognition

- In order to recognise a spoken sentence a machine has to:

The Problem of Speech Recognition

- In order to recognise a spoken sentence a machine has to:
  1. partition the waveform into disjoint pieces, so that each piece should correspo[illegible]gle phoneme

- In order to recognise a spoken sentence a machine has to:
  1. partition the waveform into disjoint pieces, so that each piece should correspo[illegible]gle phoneme
  (this is tricky in[illegible]uring out where the boundaries between the waveforms c[illegible] phonemes are)

The Problem of Speech Recognition

- In order to recognise a spoken sentence a machine has to:
  1. partition the waveform into disjoint pieces, so that each piece should correspo[illegible]gle phoneme
     (this is tricky in[illegible]uring out where the boundaries between the waveforms o[illegible] phonemes are)
  2. then it has to d[illegible]extraction from each piece, thus producing feature vectors $\vec{f}[1], \vec{f}[2], \ldots$
     (this is usually done in the frequency domain, by applying a transform of the kind of the Discrete Fourier Transform);

The Problem of Speech Recognition

- In order to recognise a spoken sentence a machine has to:
  1. partition the waveform into disjoint pieces, so that each piece should correspo[illegible]gle phoneme
     (this is tricky in[illegible]uring out where the boundaries between the waveforms o[illegible] phonemes are)
  2. then it has to d[illegible]xtraction from each piece, thus producing feature vectors $\vec{f}[1], \vec{f}[2], \ldots$
     (this is usually done in the frequency domain, by applying a transform of the kind of the Discrete Fourier Transform);
  3. finally, it has to decide for each feature vector $\vec{f}[i]$ to which phoneme it corresponds, i.e., it has to decide what phonemes would produce as their signatures each of the feature vectors $\vec{f}[i]$.

- In order to recognise a spoken sentence a machine has to:
  1. partition the waveform into disjoint pieces, so that each piece should correspo[illegible]gle phoneme (this is tricky in[illegible]uring out where the boundaries between the waveforms o[illegible] phonemes are)
  2. then it has to d[illegible]xtraction from each piece, thus producing feature vectors $\vec{f}[1], \vec{f}[2], \ldots$ (this is usually done in the frequency domain, by applying a transform of the kind of the Discrete Fourier Transform);
  3. finally, it has to decide for each feature vector $\vec{f}[i]$ to which phoneme it corresponds, i.e., it has to decide what phonemes would produce as their signatures each of the feature vectors $\vec{f}[i]$.

- Note that some phonemes are only slightly different from one another and the phonemes also vary a lot from speaker to speaker;

The Problem of Speech Recognition

- In order to recognise a spoken sentence a machine has to:
  1. partition the waveform into disjoint pieces, so that each piece should correspo[illegible]gle phoneme
     (this is tricky in[illegible]uring out where the boundaries between the waveforms c[illegible] phonemes are)
  2. then it has to d[illegible]xtraction from each piece, thus producing feature vectors $\vec{f}[1], \vec{f}[2], \ldots$
     (this is usually done in the frequency domain, by applying a transform of the kind of the Discrete Fourier Transform);
  3. finally, it has to decide for each feature vector $\vec{f}[i]$ to which phoneme it corresponds, i.e., it has to decide what phonemes would produce as their signatures each of the feature vectors $\vec{f}[i]$.

- Note that some phonemes are only slightly different from one another and the phonemes also vary a lot from speaker to speaker;
- It is hard to determine with absolute certainty from a feature vector $\vec{f}[i]$ what phoneme "caused" a waveform that resulted in that particular feature vector.

The Problem of Speech Recognition

The Problem of Speech Recognition

- One can model human speech as a Markov chain, whose states are the phonemes.

The Problem of Speech Recognition

- One can model human speech as a Markov chain, whose states are the phonemes.
- An utterance is an evolution of a Markov chain through such states, i.e., phonemes.

The Problem of Speech Recognition

- One can model human speech as a Markov chain, whose states are the phonemes.
- An utterance is an ev[illegible]f a Markov chain through such states, i.e., phonemes.
- This Markov chain is [illegible]om the listener (in our case a machine)

The Problem of Speech Recognition

- One can model human speech as a Markov chain, whose states are the phonemes.
- An utterance is an evolution of a Markov chain through such states, i.e., phonemes.
- This Markov chain is [illegible] from the listener (in our case a machine)
- The machine has access to a corresponding sequence of manifestations caused [illegible]ing these phonemes, namely the corresponding waveforms.

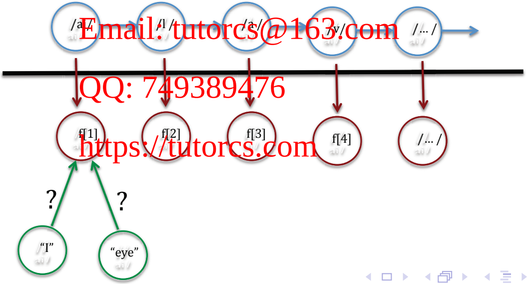
- One can model human speech as a Markov chain, whose states are the phonemes.
- An utterance is an evolution of a Markov chain through such states, i.e., phonemes.
- This Markov chain is hidden from the listener (in our case a machine)
- The machine has access to a corresponding sequence of manifestations caused by emitting these phonemes, namely the corresponding waveforms.
- These waveforms are processed into feature vectors $\vec{f}[1], \vec{f}[2], \ldots, \vec{f}[10]$.

The Problem of Speech Recognition

- One can model human speech as a Markov chain, whose states are the phonemes.
- An utterance is an evolution of a Markov chain through such states, i.e., phonemes.
- This Markov chain is hidden from the listener (in our case a machine)
- The machine has access only to a corresponding sequence of manifestations caused by producing these phonemes, namely the corresponding waveforms.
- These waveforms are processed into feature vectors $\vec{f}[1], \vec{f}[2], \ldots, \vec{f}[10]$.
- Given a sequence of manifestations $\vec{f}[1], \vec{f}[2], \ldots, \vec{f}[10]$ how can we decide what sequence of phonemes caused it?

The Problem of Speech Recognition

- One can model human speech as a Markov chain, whose states are the phonemes.
- An utterance is an evolution of a Markov chain through such states, i.e., phonemes.
- This Markov chain is hidden from the listener (in our case a machine)
- The machine has access only to a corresponding sequence of manifestations caused by producing these phonemes, namely the corresponding waveforms.
- These waveforms are processed into feature vectors $\vec{f}[1], \vec{f}[2], \ldots, \vec{f}[10]$.
- Given a sequence of manifestations $\vec{f}[1], \vec{f}[2], \ldots, \vec{f}[10]$ how can we decide what sequence of phonemes caused it?



The Viterbi Algorithm

- Main idea: we should … way that maximises the likelihood that we are correct!

The Viterbi Algorithm

- Main idea: we shoul[illegible] a way that maximises the likelihood that we are correct!
- This is what the Vit[illegible]hm does.

- Main idea: we shoul[illegible] way that maximises the likelihood that we are correct!
- This is what the Vit[illegible]hm does.
- It is a dynamic programming algorithm which produces a sequence of states that are most likely to have as a consequence the sequence of "observations" $\vec{f}[1], \vec{f}[2], \dots, \vec{f}[10]$.

The Viterbi Algorithm

- Main idea: we should [illegible] way that maximises the likelihood that we are correct!
- This is what the Vit[illegible]hm does.
- It is a dynamic programming algorithm which produces a sequence of states that are most likely to have as a consequence the sequence of "observations" $\vec{f}[1], \vec{f}[2], \ldots, \vec{f}[10]$.

- The Viterbi algorithm has a huge range of applications, besides speech recognition, for example, for decoding the transmitted convolutional codes in presence of noise in communication channels, which is used in digital telecommunications such as the CDMA mobile phone network and 802.11 wireless LANs; it is also used in many other fields such as bioinformatics and computational linguistics.

Hidden Markov Models: the general setup

- We are given a finite Markov chain, consisting of a set of its states $S = \{s_1, s_2, \ldots, s_K\}$

Hidden Markov Models: the general setup

- We are given a finite Markov chain, consisting of a set of its states $S = \{s_1, s_2, \ldots, s_K\}$
- We are also given the initial probabilities $\pi_1, \pi_2, \ldots, \pi_K$ of states,

Hidden Markov Models: the general setup

- We are given a finite Markov chain, consisting of a set of its states $S = \{s_1, s_2, \ldots, s_K\}$
- We are also given the initial probabilities $\pi_1, \pi_2, \ldots, \pi_K$ of states,
- π_i is the probability that the chain starts in state s_i.

Hidden Markov Models: the general setup

- We are given a finite Markov chain, consisting of a set of its states $S = \{s_1, s_2, \ldots, s_K\}$
- We are also given the i[illegible]abilities $\pi_1, \pi_2, \ldots, \pi_K$ of states,
- π_i is the probability th[illegible]n starts in state s_i.
- We are also given a ma[illegible]$(p(i,j) : 1 \leq i,j \leq K)$ of size $K \times K$ of transition probabilities [illegible]$(s_i \to s_j)$.

Hidden Markov Models: the general setup

- We are given a finite Markov chain, consisting of a set of its states $S = \{s_1, s_2, \ldots, s_K\}$
- We are also given the i[illegible]abilities $\pi_1, \pi_2, \ldots, \pi_K$ of states,
- π_i is the probability th[illegible]n starts in state s_i.
- We are also given a ma[illegible]$(p(i,j) : 1 \leq i, j \leq K)$ of size $K \times K$ of transition probabilities[illegible]$(s_i \rightarrow s_j)$.
- So if at an instant t th[illegible]n a state s_i, then $p(i,j)$ is the probability that at the next instant $t+1$ the chain will be in state s_j).

Hidden Markov Models: the general setup

- We are given a finite Markov chain, consisting of a set of its states $S = \{s_1, s_2, \ldots, s_K\}$
- We are also given the initial probabilities $\pi_1, \pi_2, \ldots, \pi_K$ of states,
- π_i is the probability that the chain starts in state s_i.
- We are also given a matrix $M = (p(i,j) \, : \, 1 \leq i, j \leq K)$ of size $K \times K$ of transition probabilities $p(i,j) = P(s_i \to s_j)$.
- So if at an instant t the chain is in a state s_i, then $p(i,j)$ is the probability that at the next instant $t+1$ the chain will be in state s_j).
- We have no direct access to the states of such a Markov chain.

Hidden Markov Models: the general setup

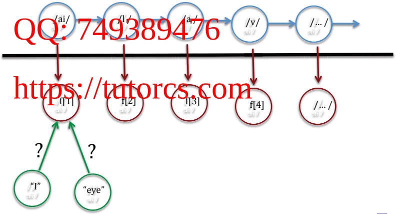
- We are given a finite Markov chain, consisting of a set of its states $S = \{s_1, s_2, \ldots, s_K\}$
- We are also given the initial probabilities $\pi_1, \pi_2, \ldots, \pi_K$ of states,
- π_i is the probability that the chain starts in state s_i.
- We are also given a matrix $M = (p(i,j) : 1 \leq i,j \leq K)$ of size $K \times K$ of transition probabilities $p(i,j) = P(s_i \to s_j)$.
- So if at an instant t the chain is in a state s_i, then $p(i,j)$ is the probability that at the next instant $t+1$ the chain will be in state s_j).
- We have no direct access to the states of such a Markov chain.
- Instead we have a set of observables $O = \{o_1, o_2, \ldots, o_N\}$ which are the possible manifestations of the states $S = \{s_1, s_2, \ldots, s_K\}$

Hidden Markov Models: the general setup

- We are given a finite Markov chain, consisting of a set of its states $S = \{s_1, s_2, \ldots, s_K\}$
- We are also given the initial probabilities $\pi_1, \pi_2, \ldots, \pi_K$ of states,
- π_i is the probability that the chain starts in state s_i.
- We are also given a matrix $M = (p(i,j) : 1 \leq i,j \leq K)$ of size $K \times K$ of transition probabilities $p(i,j) = P(s_i \to s_j)$.
- So if at an instant t the chain is in a state s_i, then $p(i,j)$ is the probability that at the next instant $t+1$ the chain will be in state s_j).
- We have no direct access to the states of such a Markov chain.
- Instead we have a set of observables $O = \{o_1, o_2, \ldots, o_N\}$ which are the possible manifestations of the states $S = \{s_1, s_2, \ldots, s_K\}$.
- We are also given the *emission matrix* E of size $K \times N$ where entry $e(i,k)$ represents the probability that when the chain is in state s_i the observable outcome will be o_k.

Hidden Markov Models: the general setup

- We are given a finite Markov chain, consisting of a set of its states $S = \{s_1, s_2, \ldots, s_K\}$
- We are also given the initial probabilities $\pi_1, \pi_2, \ldots, \pi_K$ of states,
- π_i is the probability that the chain starts in state s_i.
- We are also given a matrix $P = (p(i,j) : 1 \leq i, j \leq K)$ of size $K \times K$ of transition probabilities $p(i,j) = P(s_i \rightarrow s_j)$.
- So if at an instant t the chain is in a state s_i, then $p(i,j)$ is the probability that at the next instant $t+1$ the chain will be in state s_j).
- We have no direct access to the states of such a Markov chain.
- Instead we have a set of observables $O = \{o_1, o_2, \ldots, o_N\}$ which are the possible manifestations of the states $S = \{s_1, s_2, \ldots s_K\}$
- We are also given the *emission matrix* E of size $K \times N$ where entry $e(i,k)$ represents the probability that when the chain is in state s_i the observable outcome will be o_k.



A quick digression: what is likelihood?

- Likelihood is, in a sense, an inverse of probability.

A quick digression: what is likelihood?

- Likelihood is, in a sense, an inverse of probability.
- Assume you have a (possibly biased) coin, so that the probability of a tail is p.

A quick digression: what is likelihood?

- Likelihood is, in a sense, an inverse of probability.
- Assume you have a (possibly biased) coin, so that the probability of a tail is p.
- Thus, the probability o[illegible] $1 - p$.

A quick digression: what is likelihood?

- Likelihood is, in a sense, an inverse of probability.
- Assume you have a (possibly biased) coin, so that the probability of a tail is p.
- Thus, the probability [illegible] $1 - p$.
- The probability of gett[illegible]e $TTHTTH$ is equal to $p \cdot p \cdot (1-p) \cdot p \cdot p \cdot (1$ [illegible] $- p)^2$.

A quick digression: what is likelihood?

- Likelihood is, in a sense, an inverse of probability.
- Assume you have a (possibly biased) coin, so that the probability of a tail is p.
- Thus, the probability [illegible] $1 - p$.
- The probability of gett[illegible]e $TTHTTH$ is equal to $p \cdot p \cdot (1 - p) \cdot p \cdot p \cdot (1$ [illegible] $- p)^2$.
- But now we can invert the question: assume we have a coin with an unknown probability p of getting a tail.

A quick digression: what is likelihood?

- Likelihood is, in a sense, an inverse of probability.
- Assume you have a (possibly biased) coin, so that the probability of a tail is p.
- Thus, the probability of a head is $1 - p$.
- The probability of getting the outcome $TTHTTH$ is equal to $p \cdot p \cdot (1-p) \cdot p \cdot p \cdot (1-p) = p^4(1-p)^2$.
- But now we can invert the question: assume we have a coin with an unknown probability p of getting a tail.
- Assume we have tossed it 6 times and got the outcome $TTHTTH$.

A quick digression: what is likelihood?

- Likelihood is, in a sense, an inverse of probability.
- Assume you have a (possibly biased) coin, so that the probability of a tail is p.
- Thus, the probability o[illegible] $1 - p$.
- The probability of gett[illegible]e $TTHTTH$ is equal to $p \cdot p \cdot (1 - p) \cdot p \cdot p \cdot (1$ [illegible] $- p)^2$.
- But now we can invert the question: assume we have a coin with an unknown probability p of getting a tail.
- Assume we have tossed it 6 times and got the outcome $TTHTTH$.
- We can now ask for what value of p is such an outcome MOST LIKELY?

A quick digression: what is likelihood?

- Likelihood is, in a sense, an inverse of probability.
- Assume you have a (possibly biased) coin, so that the probability of a tail is p.
- Thus, the probability [illegible] $1 - p$.
- The probability of gett[illegible] $TTHTTH$ is equal to $p \cdot p \cdot (1 - p) \cdot p \cdot p \cdot (1$ [illegible] $- p)^2$.
- But now we can invert the question: assume we have a coin with an unknown probability p of getting a tail.
- Assume we have tossed it 6 times and got the outcome $TTHTTH$.
- We can now ask for what value of p is such an outcome MOST LIKELY?
- This boils down to finding p for which $f(p) = p^4(1 - p)^2$ attains maximal value.

- Likelihood is, in a sense, an inverse of probability.
- Assume you have a (possibly biased) coin, so that the probability of a tail is p.
- Thus, the probability of getting a head is $1 - p$.
- The probability of getting the outcome $TTHTTH$ is equal to $p \cdot p \cdot (1-p) \cdot p \cdot p \cdot (1-p) = p^4(1-p)^2$.
- But now we can invert the question: assume we have a coin with an unknown probability p of getting a tail.
- Assume we have tossed it 6 times and got the outcome $TTHTTH$.
- We can now ask for what value of p is such an outcome MOST LIKELY?
- This boils down to finding p for which $f(p) = p^4(1-p)^2$ attains maximal value.
- Differentiating $f(p)$ and setting its first derivative to zero we obtain

$$4p^3(1-p)^2 - 2p^4(1-p) = 0$$

- Likelihood is, in a sense, an inverse of probability.
- Assume you have a (possibly biased) coin, so that the probability of a tail is p.
- Thus, the probability of a head is $1-p$.
- The probability of getting the outcome $TTHTTH$ is equal to $p \cdot p \cdot (1-p) \cdot p \cdot p \cdot (1-p) = p^4(1-p)^2$.
- But now we can invert the question: assume we have a coin with an unknown probability p of getting a tail.

- Assume we have tossed it 6 times and got the outcome $TTHTTH$.
- We can now ask for what value of p is such an outcome MOST LIKELY?

- This boils down to finding p for which $f(p) = p^4(1-p)^2$ attains maximal value.
- Differentiating $f(p)$ and setting its first derivative to zero we obtain

$$4p^3(1-p)^2 - 2p^4(1-p) = 0$$

- Since we can rule out values $p=0$ and $p=1$, this implies $4(1-p)-2p=0$ which yields $p=2/3$ and $1-p=1/3$ which sounds right: the probability of getting a tail is twice as high as the probability of getting a head, and indeed, we have twice as many tails than heads in the observed outcome.

- So assume we have an observed outcome, such as $TTHTTH$ in the previous example, generated by [illegible] process with unknown parameters, such as the probability p of ge[illegible] in the previous example. Maximum Likelihood estimation [illegible]he unknown parameters so that the probability of getting [illegible]come is the largest.

- So assume we have an observed outcome, such as $TTHTTH$ in the previous example, generated by [illegible] process with unknown parameters, such as the probability p of ge[illegible] in the previous example. Maximum Likelihood estimation [illegible]he unknown parameters so that the probability of getting [illegible]come is the largest.
- Maximum likelihood often produces very good estimates of the unknown parameters, but sometimes, if the sample is small, it can produce a very bad estimate.

Maximum likelihood estimation

- So assume we have an observed outcome, such as $TTHTTH$ in the previous example, generated by [illegible] process with unknown parameters, such as the probability p of ge[illegible] in the previous example. Maximum Likelihood estimation [illegible]he unknown parameters so that the probability of getting s[illegible]come is the largest.
- Maximum likelihood often produces very good estimates of the unknown parameters, but sometimes, if the sample is small, it can produce a very bad estimate.
- As an example, consider the following problem.

- So assume we have an observed outcome, such as $TTHTTH$ in the previous example, generated by [illegible] process with unknown parameters, such as the probability p of ge[illegible] in the previous example. Maximum Likelihood estimation [illegible]he unknown parameters so that the probability of getting s[illegible]come is the largest.
- Maximum likelihood often produces very good estimates of the unknown parameters, but sometimes, if the sample is small, it can produce a very bad estimate.
- As an example, consider the following problem.
- Assume that I have given you a box which contains n balls which are numbered consecutively 1 to n, but I do not tell you what n is, i.e., how many balls there are inside.

Maximum likelihood estimation

- So assume we have an observed outcome, such as $TTHTTH$ in the previous example, generated by [illegible] process with unknown parameters, such as the probability p of ge[illegible] in the previous example. Maximum Likelihood estimation [illegible]he unknown parameters so that the probability of getting s[illegible]come is the largest.
- Maximum likelihood often produces very good estimates of the unknown parameters, but sometimes, if the sample is small, it can produce a very bad estimate.
- As an example, consider the following problem.
- Assume that I have given you a box which contains n balls which are numbered consecutively 1 to n, but I do not tell you what n is, i.e., how many balls there are inside.
- You are allowed to draw one single ball and look at its number, and then you have to estimate how many balls there are inside.

Maximum likelihood estimation

- So assume we have an observed outcome, such as $TTHTTH$ in the previous example, generated by [illegible] process with unknown parameters, such as the probability p of ge[illegible] in the previous example. Maximum Likelihood estimation [illegible] the unknown parameters so that the probability of getting [illegible]come is the largest.
- Maximum likelihood often produces very good estimates of the unknown parameters, but sometimes, if the sample is small, it can produce a very bad estimate.
- As an example, consider the following problem.
- Assume that I have given you a box which contains n balls which are numbered consecutively 1 to n, but I do not tell you what n is, i.e., how many balls there are inside.
- You are allowed to draw one single ball and look at its number, and then you have to estimate how many balls there are inside.
- Assume that you drew the ball numbered k. Since all balls are equally likely, if there are n balls inside, then the probability of drawing any particular ball is $1/n$.

Maximum likelihood estimation

- Thus, the event that you drew ball k has the highest possible probability if $1/n$ is as large as possible, i.e., when n is as small as possible.

Maximum likelihood estimation

- Thus, the event that you drew ball k has the highest possible probability if $1/n$ is as large as possible, i.e., when n is as small as possible.
- Since you know that t[illegible] least k balls inside, the MLE estimate for the number of balls in the [illegible] k, i.e., the MLE estimator in this case is $N(X) = X$, or simply [illegible]f the sample.

- Thus, the event that you drew ball k has the highest possible probability if $1/n$ is as large as possible, i.e., when n is as small as possible.
- Since you know that there are at least k balls inside, the MLE estimate for the number of balls in the box is k, i.e., the MLE estimator in this case is $N(X) = X$, or simply [illegible]f the sample.
- What is the mean of this estimator? The expected value of X, i.e., $\mu = E(X)$ is then given by

$$\mu = \sum_{i=1}^{n} \left(i \times \frac{1}{n} \right) = \frac{n(n+1)}{2n} = \frac{n+1}{2}.$$

Thus, in this case the MLE estimator is extremely biased, because its expected value is only about one half of the true value n of the number of balls inside the box!

Maximum likelihood estimation

- Thus, the event that you drew ball k has the highest possible probability if $1/n$ is as large as possible, i.e., when n is as small as possible.
- Since you know that there are at least k balls inside, the MLE estimate for the number of balls in the box is k, i.e., the MLE estimator in this case is $N(X) = X$, or simply the value of the sample.
- What is the mean of this estimator? The expected value of X, i.e., $\mu = E(X)$ is then given by

$$\mu = \sum_{i=1}^{n} \left(i \times \frac{1}{n} \right) = \frac{n(n+1)}{2n} = \frac{n+1}{2}.$$

  Thus, in this case the MLE estimator is extremely biased, because its expected value is only about one half of the true value n of the number of balls inside the box!
- If you instead use the estimator $Y(X) = 2X - 1$, then the expected value of Y is

$$\sum_{i=1}^{n} \frac{2i-1}{n} = \frac{2\sum_{i=1}^{n} i}{n} - \frac{\sum_{i=1}^{n} 1}{n} = \frac{2n(n+1)}{2n} - 1 = n$$

  and so this estimator is unbiased – much better than the MLE.

- Thus, the event that you drew ball k has the highest possible probability if $1/n$ is as large as possible, i.e., when n is as small as possible.
- Since you know that t[illegible] least k balls inside, the MLE estimate for the number of balls in the [illegible] k, i.e., the MLE estimator in this case is $N(X) = X$, or simply [illegible]f the sample.
- What is the mean of t[illegible]r? The expected value of X, i.e., $\mu = E(X)$ is then given by

$$\mu = \sum_{i=1}^{n} \left(i \times \frac{1}{n} \right) = \frac{n(n+1)}{2n} = \frac{n+1}{2}.$$

Thus, in this case the MLE estimator is extremely biased, because its expected value is only about one half of the true value n of the number of balls inside the box!

- If you instead use the estimator $Y(X) = 2X - 1$, then the expected value of Y is

$$\sum_{i=1}^{n} \frac{2i-1}{n} = \frac{2\sum_{i=1}^{n} i}{n} - \frac{\sum_{i=1}^{n} 1}{n} = \frac{2n(n+1)}{2n} - 1 = n$$

and so this estimator is unbiased – much better than the MLE.
- This does not happen for large samples; it can be shown that as the size of the sample increases, ML estimate approaches the best possible estimate.

Hidden Markov Models

- Assume now that we a … sequence of observable outcomes $(y_1, y_2, \ldots, y_T)$.

- Assume now that we a[illegible] sequence of observable outcomes $(y_1, y_2, \ldots, y_T)$.

- The goal is to find the sequence $(x_1, x_2, \ldots, x_T)$ of states of the Markov chain for which the likelihood that such a sequence is the cause of the observed sequence of outcomes $(y_1, y_2, \ldots, y_T)$ is the highest.

Hidden Markov Models

- Assume now that we are given a sequence of observable outcomes $(y_1, y_2, \ldots, y_T)$.
- The goal is to find the sequence $(x_1, x_2, \ldots, x_T)$ of states of the Markov chain for which the likelihood that such a sequence is the cause of the observed sequence of outcomes $(y_1, y_2, \ldots, y_T)$ is the highest.
- In other words, we are looking for *the most likely cause* of the sequence of observations $(y_1, y_2, \ldots, y_T)$.

- Assume now that we a[illegible] sequence of observable outcomes $(y_1, y_2, \ldots, y_T)$.
- The goal is to find the sequence $(x_1, x_2, \ldots, x_T)$ of states of the Markov chain for which the likelihood that such a sequence is the cause of the observed sequence of outcomes $(y_1, y_2, \ldots, y_T)$ is the highest.
- In other words, we are looking for *the most likely cause* of the sequence of observations $(y_1, y_2, \ldots, y_T)$.
- In our speech recognition example we are given a sequence of feature vectors $(\vec{f}[1], \vec{f}[2], \ldots, \vec{f}[10])$ extracted from the sound waveform and we have to find what sequence of phonemes $(\phi_1, \phi_2, \ldots, \phi_{10})$ is the most likely cause of the sequence of the feature vectors $(\vec{f}[1], \vec{f}[2], \ldots, \vec{f}[10])$.

The Viterbi Algorithm

- Theoretically, we could solve this problem by brute force:

The Viterbi Algorithm

- Theoretically, we could solve this problem by brute force:
- we could go through all possible sequences of states $\vec{x} = (x_1, x_2, \ldots, x_T)$ of the same length as the seq[illegible]bservables $\vec{y} = (y_1, y_2, \ldots, y_T)$ and compute the corresponding prob[illegible]at each of such sequences might be the cause for the given sequence [illegible]bles $\vec{y} = (y_1, y_2, \ldots, y_T)$ and then select the sequence $\vec{x}$ for which s[illegible]ility is the largest.

The Viterbi Algorithm

- Theoretically, we could solve this problem by brute force:
- we could go through all possible sequences of states $\vec{x} = (x_1, x_2, \ldots, x_T)$ of the same length as the sequence of observables $\vec{y} = (y_1, y_2, \ldots, y_T)$ and compute the corresponding probability that each of such sequences might be the cause for the given sequence of observables $\vec{y} = (y_1, y_2, \ldots, y_T)$ and then select the sequence $\vec{x}$ for which such probability is the largest.
- This could, theoretically, be done as follows. We first find the probability $P(\vec{x})$ that a sequence $\vec{x}$ occurred at all, as the following product:

$$P(\vec{x}) = \pi(x_1) \cdot p(x_1 \to x_2) \cdot p(x_2 \to x_3) \ldots p(x_{T-1} \to x_T)$$

The Viterbi Algorithm

- Theoretically, we could solve this problem by brute force:
- we could go through all possible sequences of states $\vec{x} = (x_1, x_2, \ldots, x_T)$ of the same length as the seq[illegible]bservables $\vec{y} = (y_1, y_2, \ldots, y_T)$ and compute the corresponding pro[illegible]at each of such sequences might be the cause for the given sequence [illegible]les $\vec{y} = (y_1, y_2, \ldots, y_T)$ and then select the sequence $\vec{x}$ for which s[illegible]ility is the largest.
- This could, theoretical[illegible] as follows. We first find the probability $P(\vec{x})$ that a sequence $\vec{x}$ occurred at all, as the following product:

$$P(\vec{x}) = \pi(x_1) \cdot p(x_1 \to x_2) \cdot p(x_2 \to x_3) \ldots p(x_{T-1} \to x_T)$$

- This is because $\pi(x_1)$ is the probability that the starting state (phoneme in our example) is x_1, times the probability $p(x_1 \to x_2)$ that system transitions from state x_1 into state x_2 and so fourth.

The Viterbi Algorithm

- Theoretically, we could solve this problem by brute force:
- we could go through all possible sequences of states $\vec{x} = (x_1, x_2, \ldots, x_T)$ of the same length as the sequence of observables $\vec{y} = (y_1, y_2, \ldots, y_T)$ and compute the corresponding probability that each of such sequences might be the cause for the given sequence of observables $\vec{y} = (y_1, y_2, \ldots, y_T)$ and then select the sequence $\vec{x}$ for which such probability is the largest.
- This could, theoretically, be done as follows. We first find the probability $P(\vec{x})$ that a sequence $\vec{x}$ occurred at all, as the following product:

$$P(\vec{x}) = \pi(x_1) \cdot p(x_1 \to x_2) \cdot p(x_2 \to x_3) \ldots p(x_{T-1} \to x_T)$$

- This is because $\pi(x_1)$ is the probability that the starting state (phoneme in our example) is x_1, times the probability $p(x_1 \to x_2)$ that system transitions from state x_1 into state x_2 and so fourth.
- Next, we compute the probability $P^*(\vec{x}, \vec{y})$ that this particular sequence of states $\vec{x} = (x_1, \ldots, x_T)$ will cause the sequence of observable outcomes $\vec{y} = (y_1, \ldots, y_T)$, as the product of the probabilities $e(x_1, y_1) \cdot \ldots \cdot e(x_T, y_T)$ where each $e(x_i, y_i)$ is the "emission" probability that state x_i produces the observable manifestation y_i.

- Theoretically, we could solve this problem by brute force:
- we could go through all possible sequences of states $\vec{x} = (x_1, x_2, \ldots, x_T)$ of the same length as the sequence of observables $\vec{y} = (y_1, y_2, \ldots, y_T)$ and compute the corresponding probability that each of such sequences might be the cause for the given sequence of observables $\vec{y} = (y_1, y_2, \ldots, y_T)$ and then select the sequence $\vec{x}$ for which such probability is the largest.
- This could, theoretically, be done as follows. We first find the probability $P(\vec{x})$ that a sequence $\vec{x}$ occurred at all, as the following product:

$$P(\vec{x}) = \pi(x_1) \cdot p(x_1 \to x_2) \cdot p(x_2 \to x_3) \ldots p(x_{T-1} \to x_T)$$

- This is because $\pi(x_1)$ is the probability that the starting state (phoneme in our example) is x_1, times the probability $p(x_1 \to x_2)$ that system transitions from state x_1 into state x_2 and so fourth.
- Next, we compute the probability $P^*(\vec{x}, \vec{y})$ that this particular sequence of states $\vec{x} = (x_1, \ldots, x_T)$ will cause the sequence of observable outcomes $\vec{y} = (y_1, \ldots, y_T)$, as the product of the probabilities $e(x_1, y_1) \cdot \ldots \cdot e(x_T, y_T)$ where each $e(x_i, y_i)$ is the "emission" probability that state x_i produces the observable manifestation y_i.
- Finally, the probability $\mathcal{P}(\vec{x}, \vec{y})$ that sequence $\vec{x}$ is the cause of the sequence of observable outcomes $\vec{y}$ is the product of the probabilities $P(\vec{x})$ that sequence $\vec{x}$ occurred at all and the probability $P^*(\vec{x}, \vec{y})$ that, if it did occur, it also caused the right observable outcome $\vec{y}$, so in total $\mathcal{P}(\vec{x}, \vec{y}) = P(\vec{x}) \cdot P^*(\vec{x}, \vec{y})$.

- So given the sequence … s $\vec{y} = (y_1, y_2, \ldots, y_T)$ observed, we could now simply pick the sequen… s $\vec{x} = (x_1, x_2, \ldots, x_T)$ for which the value of $\mathcal{P}(\vec{x}, \vec{y})$ is the largest.

- So given the sequence s $\vec{y} = (y_1, y_2, \ldots, y_T)$ observed, we could now simply pick the sequen s $\vec{x} = (x_1, x_2, \ldots, x_T)$ for which the value of $\mathcal{P}(\vec{x}, \vec{y})$ is the largest.
- However, this is not a feasible algorithm: if the total number of states is K and the observed sequence is of length T then there would be in total K^T many sequences to try, i.e., exponentially many.

The Viterbi Algorithm

- So given the sequence [illegible]s $\vec{y} = (y_1, y_2, \ldots, y_T)$ observed, we could now simply pick the sequen[illegible]s $\vec{x} = (x_1, x_2, \ldots, x_T)$ for which the value of $\mathcal{P}(\vec{x}, \vec{y})$ is the largest.
- However, this is not a feasible algorithm: if the total number of states is K and the observed sequence is of length T then there would be in total K^T many sequences to try, i.e., exponentially many.
- This is no good for real time (or almost real time) algorithms such as speech recognition or decoding the convolutional codes used in mobile telephony.

The Viterbi Algorithm

- So given the sequence [illegible]s $\vec{y} = (y_1, y_2, \ldots, y_T)$ observed, we could now simply pick the sequen[illegible]s $\vec{x} = (x_1, x_2, \ldots, x_T)$ for which the value of $\mathcal{P}(\vec{x}, \vec{y})$ is the largest.

- However, this is not a feasible algorithm: if the total number of states is K and the observed sequence is of length T then there would be in total K^T many sequences to try, i.e., exponentially many.

- This is no good for real time (or almost real time) algorithms such as speech recognition or decoding the convolutional codes used in mobile telephony.

- Instead, the Viterbi algorithm solves this problem in time $O(T \times K^2)$ using dynamic programming.

The Viterbi Algorithm

- **The Viterbi Algorithm:** We solve all subproblems $\mathcal{S}(i,k)$ for every $1 \leq i \leq T$ and every $1 \leq k \leq K$:

  Subproblem $\mathcal{S}(i,k)$: *equence of states* $(x_1, \ldots x_i)$ *such that* $x_i = s_k$ *and such that the prob* *serving the outcome* $(y_1, \ldots, y_i)$ *is maximal.*

The Viterbi Algorithm

- **The Viterbi Algorithm:** We solve all subproblems $\mathcal{S}(i,k)$ for every $1 \leq i \leq T$ and every $1 \leq k \leq K$:

  Subproblem $\mathcal{S}(i,k)$: [illegible] *sequence of states* $(x_1, \ldots x_i)$ *such that* $x_i = s_k$ *and such that the prob*[illegible]*serving the outcome* $(y_1, \ldots, y_i)$ *is maximal.*

- Thus, we solve subpro[illegible]l truncations $(x_1, \ldots, x_i)$, $1 \leq i \leq T$, but also insisting that the last state x_i of such a truncated subsequence be state s_k.

The Viterbi Algorithm

- **The Viterbi Algorithm:** We solve all subproblems $\mathcal{S}(i,k)$ for every $1 \leq i \leq T$ and every $1 \leq k \leq K$:

  Subproblem $\mathcal{S}(i,k)$: *Find a sequence of states* $(x_1, \ldots x_i)$ *such that* $x_i = s_k$ *and such that the probability of observing the outcome* $(y_1, \ldots, y_i)$ *is maximal.*

- Thus, we solve subproblems not only for all truncations $(x_1, \ldots, x_i)$, $1 \leq i \leq T$, but also insisting that the last state x_i of such a truncated subsequence be state s_k.

- Let us denote by $L(i,k)$ the largest possible probability for which there exists a sequence $(x_1, x_2, \ldots, x_i)$ ending with $x_i = s_k$ causing the subsequence of observable outcomes $(y_1, y_2, \ldots, y_i)$; such probabilities must satisfy the following recursion:

$$L(1,k) = \pi_k \cdot e(s_k, y_1)$$

$$L(i,k) = \max_{m \in S} \left(L(i-1,m)\, p(s_m \to s_k)\, e(s_k, y_i)\right);$$

$$\sigma(i,k) = \arg\max_{m \in S} \left(L(i-1,m)\, p(s_m \to s_k)\, e(s_k, y_i)\right).$$

- **The Viterbi Algorithm:** We solve all subproblems $\mathcal{S}(i,k)$ for every $1 \leq i \leq T$ and every $1 \leq k \leq K$:

  Subproblem $\mathcal{S}(i,k)$: [illegible] *sequence of states* $(x_1, \ldots x_i)$ *such that* $x_i = s_k$ *and such that the prob*[illegible]*serving the outcome* $(y_1, \ldots, y_i)$ *is maximal.*

- Thus, we solve subpro[illegible]l truncations $(x_1, \ldots, x_i)$, $1 \leq i \leq T$, but also insisting that the last state x_i of such a truncated subsequence be state s_k.

- Let us denote by $L(i,k)$ the largest possible probability for which there exists a sequence $(x_1, x_2, \ldots, x_i)$ ending with $x_i = s_k$ causing the subsequence of observable outcomes $(y_1, y_2, \ldots, y_i)$; such probabilities must satisfy the following recursion:

$$L(1,k) = \pi_k \cdot e(s_k, y_1)$$

$$L(i,k) = \max_{m \in S} \left(L(i-1,m)\, p(s_m \to s_k)\, e(s_k, y_i) \right);$$

$$\sigma(i,k) = \arg\max_{m \in S} \left(L(i-1,m)\, p(s_m \to s_k)\; e(s_k, y_i) \right).$$

- Here $\sigma(i,k)$ stores the value of m for which $L(i-1,m)\, p(s_m \to s_k)\, e(s_k, y_i)$ is the largest which allows us to reconstruct the sequence of states which maximize the probabilities we are tracking.

The Viterbi Algorithm

$$
\begin{aligned}
L(1,k) &= \pi_k\, e(s_k, y_1);\\
L(i,k) &= \max_{m \in S} \left(L(i-1,m)\, p(s_m \to s_k)\, e(s_k, y_i) \right);\\
\sigma(i,k) &= \arg\max_{m \in S} \left(L(i-1,m)\, p(s_m \to s_k)\, e(s_k, y_i) \right).
\end{aligned}
$$

- We can now obtain the sequence of states which is most likely cause of the given sequence of observations by backtracking:

$$
\begin{aligned}
x_T &= \arg\max_{m \in S} (L(T,m))\\
x_{i-1} &= \sigma(i, x_i), \quad i = T, T-1, \ldots, 2.
\end{aligned}
$$

The Viterbi Algorithm Example

- In California there are twice as many raccoons as possums. Having gotten a job with Google, you are in California observing a back yard. It is dusk, so the probability that you think you saw a raccoon when you are actually looking at a possum at a distance [illegible] the probability that you think you saw a possum while you are actually looking at a raccoon at a distance is 1/4. Raccoons move in packs; if a raccoon comes to your back yard the probability that the next animal to follow will also be a raccoon is 4/5. Possums are solitary animals, so if a possum comes to your back yard, this does not impact the probabilities what the next animal to come will be (a possum or a raccoon, but recall in California there are twice as many raccoons as possums!) You believe that you saw four animals coming in the following order: a raccoon, a possum, a possum, a raccoon ($rppr$). Given such a sequence of observations, what actual sequence of animals is most likely to cause such a sequence of your observations?

- In California there are twice as many raccoons as possums. Having gotten a job with Google, you are in California observing a back yard. It is dusk, so the probability that you th[illegible]w a raccoon when you are actually looking at a possum at a distance [illegible] probability that you think you saw a possum while you are [illegible]oking at a raccoon at a distance is 1/4. Raccoons move in pac[illegible]accoon comes to your back yard the probability that the ne[illegible]o follow will also be a raccoon is 4/5. Possums are solitary animals, so if a possum comes to your back yard, this does not impact the probabilities what the next animal to come will be (a possum or a raccoon, but recall in California there are twice as many raccoons as possums!) You believe that you saw four animals coming in the following order: a raccoon, a possum, a possum, a raccoon (*rppr*). Given such a sequence of observations, what actual sequence of animals is most likely to cause such a sequence of your observations?
- probabilities of initial states:

The Viterbi Algorithm Example

- In California there are twice as many raccoons as possums. Having gotten a job with Google, you are in California observing a back yard. It is dusk, so the probability that you th[illegible]w a raccoon when you are actually looking at a possum at a distance[illegible] probability that you think you saw a possum while you are [illegible]oking at a raccoon at a distance is 1/4. Raccoons move in pac[illegible]accoon comes to your back yard the probability that the ne[illegible]o follow will also be a raccoon is 4/5. Possums are solitary animals, so if a possum comes to your back yard, this does not impact the probabilities what the next animal to come will be (a possum or a raccoon, but recall in California there are twice as many raccoons as possums!) You believe that you saw four animals coming in the following order: a raccoon, a possum, a possum, a raccoon ($rppr$). Given such a sequence of observations, what actual sequence of animals is most likely to cause such a sequence of your observations?
- probabilities of initial states: $\pi(R) = 2/3$;

The Viterbi Algorithm Example

- In California there are twice as many raccoons as possums. Having gotten a job with Google, you are in California observing a back yard. It is dusk, so the probability that you th[illegible]w a raccoon when you are actually looking at a possum at a distance[illegible]e probability that you think you saw a possum while you are [illegible]oking at a raccoon at a distance is 1/4. Raccoons move in pack[illegible]accoon comes to your back yard the probability that the ne[illegible]to follow will also be a raccoon is 4/5. Possums are solitary animals, so if a possum comes to your back yard, this does not impact the probabilities what the next animal to come will be (a possum or a raccoon, but recall in California there are twice as many raccoons as possums!) You believe that you saw four animals coming in the following order: a raccoon, a possum, a possum, a raccoon ($rppr$). Given such a sequence of observations, what actual sequence of animals is most likely to cause such a sequence of your observations?
- probabilities of initial states: $\pi(R) = 2/3$; $\pi(P) = 1/3$.

The Viterbi Algorithm Example

- In California there are twice as many raccoons as possums. Having gotten a job with Google, you are in California observing a back yard. It is dusk, so the probability that you th[illegible]w a raccoon when you are actually looking at a possum at a distance [illegible] probability that you think you saw a possum while you are [illegible]oking at a raccoon at a distance is 1/4. Raccoons move in pac[illegible]accoon comes to your back yard the probability that the ne[illegible]to follow will also be a raccoon is 4/5. Possums are solitary animals, so if a possum comes to your back yard, this does not impact the probabilities what the next animal to come will be (a possum or a raccoon, but recall in California there are twice as many raccoons as possums!) You believe that you saw four animals coming in the following order: a raccoon, a possum, a possum, a raccoon ($rppr$). Given such a sequence of observations, what actual sequence of animals is most likely to cause such a sequence of your observations?
- probabilities of initial states: $\pi(R) = 2/3$; $\pi(P) = 1/3$.
- transition probabilities:
  $p(R \rightarrow R) = 4/5$;

The Viterbi Algorithm Example

- In California there are twice as many raccoons as possums. Having gotten a job with Google, you are in California observing a back yard. It is dusk, so the probability that you th[illegible]w a raccoon when you are actually looking at a possum at a distance [illegible]e probability that you think you saw a possum while you are [illegible]oking at a raccoon at a distance is 1/4. Raccoons move in pac[illegible]accoon comes to your back yard the probability that the ne[illegible]o follow will also be a raccoon is 4/5. Possums are solitary animals, so if a possum comes to your back yard, this does not impact the probabilities what the next animal to come will be (a possum or a raccoon, but recall in California there are twice as many raccoons as possums!) You believe that you saw four animals coming in the following order: a raccoon, a possum, a possum, a raccoon ($rppr$). Given such a sequence of observations, what actual sequence of animals is most likely to cause such a sequence of your observations?
- probabilities of initial states: $\pi(R) = 2/3$; $\pi(P) = 1/3$.
- transition probabilities:
  $p(R \to R) = 4/5$; $p(R \to P) = 1/5$;

The Viterbi Algorithm Example

- In California there are twice as many raccoons as possums. Having gotten a job with Google, you are in California observing a back yard. It is dusk, so the probability that you th[illegible]w a raccoon when you are actually looking at a possum at a distanc[illegible]e probability that you think you saw a possum while you are [illegible]oking at a raccoon at a distance is 1/4. Raccoons move in pac[illegible]accoon comes to your back yard the probability that the ne[illegible]to follow will also be a raccoon is 4/5. Possums are solitary animals, so if a possum comes to your back yard, this does not impact the probabilities what the next animal to come will be (a possum or a raccoon, but recall in California there are twice as many raccoons as possums!) You believe that you saw four animals coming in the following order: a raccoon, a possum, a possum, a raccoon ($rppr$). Given such a sequence of observations, what actual sequence of animals is most likely to cause such a sequence of your observations?
- probabilities of initial states: $\pi(R) = 2/3$; $\pi(P) = 1/3$.
- transition probabilities:
  $p(R \rightarrow R) = 4/5$; $p(R \rightarrow P) = 1/5$; $p(P \rightarrow P) = 1/3$;

The Viterbi Algorithm Example

- In California there are twice as many raccoons as possums. Having gotten a job with Google, you are in California observing a back yard. It is dusk, so the probability that you th[illegible]w a raccoon when you are actually looking at a possum at a distanc[illegible]e probability that you think you saw a possum while you are [illegible]oking at a raccoon at a distance is 1/4. Raccoons move in pac[illegible]accoon comes to your back yard the probability that the ne[illegible]to follow will also be a raccoon is 4/5. Possums are solitary animals, so if a possum comes to your back yard, this does not impact the probabilities what the next animal to come will be (a possum or a raccoon, but recall in California there are twice as many raccoons as possums!) You believe that you saw four animals coming in the following order: a raccoon, a possum, a possum, a raccoon ($rppr$). Given such a sequence of observations, what actual sequence of animals is most likely to cause such a sequence of your observations?
- probabilities of initial states: $\pi(R) = 2/3$; $\pi(P) = 1/3$.
- transition probabilities:
  $p(R \to R) = 4/5$; $p(R \to P) = 1/5$; $p(P \to P) = 1/3$; $p(P \to R) = 2/3$.

- In California there are twice as many raccoons as possums. Having gotten a job with Google, you are in California observing a back yard. It is dusk, so the probability that you th[illegible]w a raccoon when you are actually looking at a possum at a distanc[illegible] probability that you think you saw a possum while you are [illegible]oking at a raccoon at a distance is $1/4$. Raccoons move in pac[illegible]accoon comes to your back yard the probability that the ne[illegible]to follow will also be a raccoon is $4/5$. Possums are solitary animals, so if a possum comes to your back yard, this does not impact the probabilities what the next animal to come will be (a possum or a raccoon, but recall in California there are twice as many raccoons as possums!) You believe that you saw four animals coming in the following order: a raccoon, a possum, a possum, a raccoon ($rppr$). Given such a sequence of observations, what actual sequence of animals is most likely to cause such a sequence of your observations?
- probabilities of initial states: $\pi(R) = 2/3$; $\pi(P) = 1/3$.
- transition probabilities:
  $p(R \to R) = 4/5$; $p(R \to P) = 1/5$; $p(P \to P) = 1/3$; $p(P \to R) = 2/3$.
- emission probabilities:
  $O(r|P) = 1/3$;

The Viterbi Algorithm Example

- In California there are twice as many raccoons as possums. Having gotten a job with Google, you are in California observing a back yard. It is dusk, so the probability that you th[illegible]w a raccoon when you are actually looking at a possum at a distance[illegible] probability that you think you saw a possum while you are [illegible]oking at a raccoon at a distance is 1/4. Raccoons move in pac[illegible]accoon comes to your back yard the probability that the ne[illegible]to follow will also be a raccoon is 4/5. Possums are solitary animals, so if a possum comes to your back yard, this does not impact the probabilities what the next animal to come will be (a possum or a raccoon, but recall in California there are twice as many raccoons as possums!) You believe that you saw four animals coming in the following order: a raccoon, a possum, a possum, a raccoon ($rppr$). Given such a sequence of observations, what actual sequence of animals is most likely to cause such a sequence of your observations?
- probabilities of initial states: $\pi(R) = 2/3$; $\pi(P) = 1/3$.
- transition probabilities:
  $p(R \to R) = 4/5$; $p(R \to P) = 1/5$; $p(P \to P) = 1/3$; $p(P \to R) = 2/3$.
- emission probabilities:
  $O(r|P) = 1/3$; $O(p|P) = 2/3$;

The Viterbi Algorithm Example

- In California there are twice as many raccoons as possums. Having gotten a job with Google, you are in California observing a back yard. It is dusk, so the probability that you th[illegible]w a raccoon when you are actually looking at a possum at a distance[illegible] probability that you think you saw a possum while you are [illegible]oking at a raccoon at a distance is 1/4. Raccoons move in pac[illegible]accoon comes to your back yard the probability that the ne[illegible]to follow will also be a raccoon is 4/5. Possums are solitary animals, so if a possum comes to your back yard, this does not impact the probabilities what the next animal to come will be (a possum or a raccoon, but recall in California there are twice as many raccoons as possums!) You believe that you saw four animals coming in the following order: a raccoon, a possum, a possum, a raccoon ($rppr$). Given such a sequence of observations, what actual sequence of animals is most likely to cause such a sequence of your observations?
- probabilities of initial states: $\pi(R) = 2/3$; $\pi(P) = 1/3$.
- transition probabilities:
  $p(R \to R) = 4/5$; $p(R \to P) = 1/5$; $p(P \to P) = 1/3$; $p(P \to R) = 2/3$.
- emission probabilities:
  $O(r|P) = 1/3$; $O(p|P) = 2/3$; $O(p|R) = 1/4$;

The Viterbi Algorithm Example

- In California there are twice as many raccoons as possums. Having gotten a job with Google, you are in California observing a back yard. It is dusk, so the probability that you th[illegible]w a raccoon when you are actually looking at a possum at a distanc[illegible] probability that you think you saw a possum while you are [illegible]oking at a raccoon at a distance is 1/4. Raccoons move in pac[illegible]accoon comes to your back yard the probability that the ne[illegible]to follow will also be a raccoon is 4/5. Possums are solitary animals, so if a possum comes to your back yard, this does not impact the probabilities what the next animal to come will be (a possum or a raccoon, but recall in California there are twice as many raccoons as possums!) You believe that you saw four animals coming in the following order: a raccoon, a possum, a possum, a raccoon ($rppr$). Given such a sequence of observations, what actual sequence of animals is most likely to cause such a sequence of your observations?
- probabilities of initial states: $\pi(R) = 2/3$; $\pi(P) = 1/3$.
- transition probabilities:
  $p(R \to R) = 4/5$; $p(R \to P) = 1/5$; $p(P \to P) = 1/3$; $p(P \to R) = 2/3$.
- emission probabilities:
  $O(r|P) = 1/3$; $O(p|P) = 2/3$; $O(p|R) = 1/4$; $O(r|R) = 3/4$.

The Viterbi Algorithm Example

- In California there are twice as many raccoons as possums. Having gotten a job with Google, you are in California observing a back yard. It is dusk, so the probability that you th[illegible]w a raccoon when you are actually looking at a possum at a distance[illegible]e probability that you think you saw a possum while you are [illegible]oking at a raccoon at a distance is 1/4. Raccoons move in pac[illegible]accoon comes to your back yard the probability that the ne[illegible]o follow will also be a raccoon is 4/5. Possums are solitary animals, so if a possum comes to your back yard, this does not impact the probabilities what the next animal to come will be (a possum or a raccoon, but recall in California there are twice as many raccoons as possums!) You believe that you saw four animals coming in the following order: a raccoon, a possum, a possum, a raccoon ($rppr$). Given such a sequence of observations, what actual sequence of animals is most likely to cause such a sequence of your observations?
- probabilities of initial states: $\pi(R) = 2/3$; $\pi(P) = 1/3$.
- transition probabilities:
  $p(R \to R) = 4/5$; $p(R \to P) = 1/5$; $p(P \to P) = 1/3$; $p(P \to R) = 2/3$.
- emission probabilities:
  $O(r|P) = 1/3$; $O(p|P) = 2/3$; $O(p|R) = 1/4$; $O(r|R) = 3/4$.
- **Homework:** Execute the Viterbi algorithm by hand (using a calculator) to see what actual sequence of animals is most likely to cause the sequence of observations $rppr$.