6.01: Introduction to EECS I

Lecture 11

Discrete Probability and State Estimation

Antonio Torralba

April 29, 2008

Uncertainty

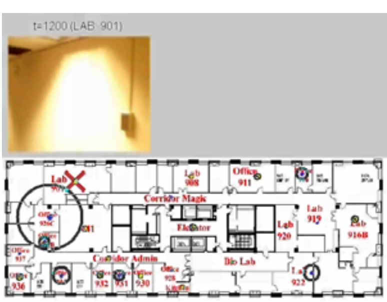
We have used the idea of state space to plan trajectories from a starting state to a goal.

We assumed:

- We knew the initial state
- Actions were executed without error

Unfortunately, things are not as ideal in real systems

Uncertainty



Initial state: unknown
Observations: low resolution and low rate video

What do we know? The floor plant, and we have learnt to recognize the rooms

Uncertainty

Uncertainty

Uncertainty

In this video, the model of the world is driving your interpretations.
But, the your model of *this* world might be wrong.

- Lecture
  - Probabilistic model of the state space
  - Probabilistic model of the observations
- Lab
  - Modeling the effect of actions

Probability

Probability theory allow us to assign numerical assessments of uncertainty to possible events.

$\mathcal{U}$ = universe = set of all possible atomic events

Atomic event = an outcome

Axioms

- $P(\mathcal{U}) = 1$
- $P(\{\}) = 0$
- $P(A \cup B) = P(A) + P(B) - P(A \cap B)$

Discrete Random Variables

A discrete random variable X takes a discrete set of values $x_1, x_2, \ldots, x_n$ with probabilities $p_1, p_2, \ldots, p_n$

Examples

- Fair coin: X = {head : 0.5, tails : 0.5}
- Biased coin: X = {head : 0.6, tails : 0.4}

Question: what is an atomic event when we flip two coins?

Pairs of random variables

We can consider two random variables together to understand how they interact.

C = cavity : {True, False}
A = Toothache : {True, False}

The event space is the cartesian product of the value spaces of the variables

C x A = {(T,T), (T,F), (F,T), (F,F)}

Joint Distribution

The joint distribution is a function from elements of the product space to probabilities

C x A = {(T,T), (T,F), (F,T), (F,F)}

		C = T	C = F
A	T	0.05	0.05
A	F	0.1	0.8

P(A=F, C =T) = 0.1

P(A=T, C =T) = 0.8

P(A=T, C =T) + P(A=F, C =T) + P(A=F, C =T) + P(A=F, C =F) = 1

Joint Distribution

C = cavity : {True, False}
A = Toothache : {True, False}

		C = T	C = F
A	T	0.05	0.05
A	F	0.1	0.8

P(A = T) = ?

P(A = T) = P(A=T, C=T) + P(A=T, C= F) = 0.05 + 0.05 = 0.1

P(C = T) = P(A=T, C=T) + P(A=F, C= T) = 0.05 + 0.1 = 0.15

These are called marginal probabilities

$$\underset{0.05}{P(A = T, C = T)} \neq \underset{0.015}{P(A = T) \cdot P(C = T)}$$

Equality holds if the random variables A and C are independent

Conditional Probability

What is the probability of having a cavity if the patient has toothache?

$$P(C = T \mid A = T) = ?$$

We are only uncertain about the value of C

		C = T	C = F
A	T	0.05	0.05
	F	0.1	0.8

$$P(C = T \mid A = T) = \frac{P(C = T, A = T)}{P(A = T)} = \frac{0.05}{0.1} = 0.5$$

Bayes' Rule

$$P(E_1 \mid E_2) = \frac{P(E_2 \mid E_1) \cdot P(E_1)}{P(E_2)}$$

Thomas Bayes (1702- 1761)

Verification:

$$\left.\begin{aligned} P(E_1 \mid E_2) &= \frac{P(E_2 , E_1)}{P(E_2)} \\ P(E_2 \mid E_1) &= \frac{P(E_2 , E_1)}{P(E_1)} \end{aligned}\right\} P(E_1 \mid E_2) \cdot P(E_2) = P(E_2 \mid E_1) \cdot P(E_1)$$

Sequences

We want to consider the case in which we have a sequence of states (random variables)

The random variables could represent:

- Position of the robot at time *t*

- A word at position *t* within a sentence

Last week, we introduced the idea of a state space, and its use for planning trajectories from some starting state to a goal.

A Model for the 6.01 course notes

Last week, we introduced the idea of a state space, and its use for planning trajectories from some starting state to a goal. Our assumptions in that work were that we knew the initial state, and that the actions could be executed without error. That is a useful idealization in many cases, but it is also very frequently false. Even navigation through a city can fail on both counts: sometimes we don't know where we are on a map, and sometimes, due to traffic or road work or bad driving, we fail to execute a turn we had intended to take.

In such situations, we have some information about where we are: we can make observations of our local surroundings, which give us useful information; and we know what actions we have taken and the consequences those are likely to have on our location. So, the question is: how can we take information from a sequence of actions and local observations and integrate it into some sort of estimate of where we are? What form should that estimate take?

We will consider this text as a sequence of random variables: W_t

Each variable is one word W_t which can take any value within a Dictionary.

A Model for the 6.01 course notes

1) Faculty select words from the 6.01 dictionary

state stable programming python conditional

2) Each word is selected randomly with some probability

P (W = “Stable”) = 0.1 ?

P (W = “Stable”) + P (W = “programming”) + P (W = “python”) + … = 1

3) The memoryless model of a 6.01 faculty:

To build a sequence, each word is selected independently of the previous word.

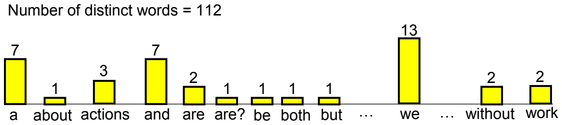
Estimation of P(W)

Last week, we introduced the idea of a state space, and its use for planning trajectories from some starting state to a goal. Our assumptions in that work were that we knew the initial state, and that the actions could be executed without error. That is a useful idealization in many cases, but it is also very frequently false. Even navigation through a city can fail on both counts: sometimes we don't know where we are on a map, and sometimes, due to traffic or road work or bad driving, we fail to execute a turn we had intended to take.

In such situations, we have some information about where we are: we can make observations of our local surroundings, which give us useful information; and we know what actions we have taken and the consequences those are likely to have on our location. So, the question is: how can we take information from a sequence of actions and local observations and integrate it into some sort of estimate of where we are? What form should that estimate take?

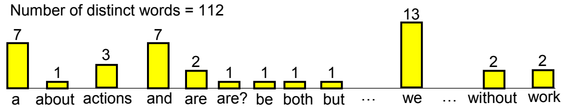
Number of words = 179

Number of distinct words = 112



Estimation of P(W)

Number of words = 179
Number of distinct words = 112



We aren't done yet, we need to translate counts into probabilities

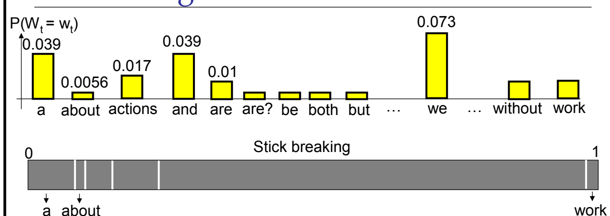
$$P(W = \text{“a”}) \approx \frac{\text{Counts (“a”)}}{\text{Number of words}} = 7 / 179 = 0.039$$

$$P(W = \text{“about”}) \approx \frac{\text{Counts (“about”)}}{\text{Number of words}} = 1 / 179 = 0.0056$$

This estimation guarantees that:

$$\sum_{w_i \text{ in Dictionary}} P(W = w_i) = 1$$

Generating Text



To	we	the	we	city	the	useful
a	actions	actions	question	to	those	can

We assume that $P(W_t=w_t)$ is stationary. It does not change with time.

Generating Text

To we the we city the useful a actions actions question to those can fail give a we and or we cases, take from planning In idea state to actions are In information is and a a without we What of some are the a planning navigation counts: state we of are: likely turn we Our sometimes, and can had work know taken and we know road sort a is driving, road of So, idea should have we for are a navigation some where know can where it surroundings, the planning our that have actions a local or taken false.

Properties of the memoryless model

- Words are drawn independently

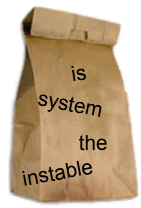
$$P(W_0=w_0, W_1=w_1, \ldots, W_N=w_N) \underset{\uparrow}{=} P(W_0=w_0) \cdot P(W_1=w_1) \cdot \ldots \cdot P(W_N=w_N)$$

Independence assumption

- Under this model, the probability of a sentence does not depend on word order!

Sentences are bags of words

P("the system is instable") = P("is system the instable")



Sequences

Each paragraph is a sequence of words

$$w_0, w_1, w_2, \ldots w_N$$

How do we decide which word to add next?

The words are not independent

$$P(W_{N+1}=w_{N+1} \mid W_0=w_0, W_1=w_1, \ldots W_N=w_N) \neq P(W_{N+1}=w_{N+1})$$

But, capturing all the dependencies is too complicated.

We need a simple approximation that still captures properties of the text without requiring a full model of our brains.

The bigram model of 6.01 course notes

Faculty starts a paragraph by randomly selecting one word from the dictionary:

Initial state distribution
$P(W_0=w_0)$

Each word depends only on the previous word:

State transition model

$$P(W_{t+1}=w_{t+1} \mid W_0=w_0, W_1=w_1, \ldots W_t=w_t) \underset{\uparrow}{=} P(W_{t+1}=w_{t+1} \mid W_t=w_t)$$

Markov sequence

The bigram model of 6.01 course notes

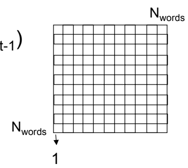
To build this model we need to estimate:

1) Initial state distribution

$$P(W_0 = w_0)$$

2) State transition model

$$P(W_t = w_t \mid W_{t-1} = w_{t-1})$$

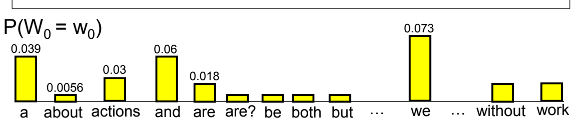


The number of parameters in the model is: Nwords + $(\text{Nwords})^2$

Bigram: Initial state distribution

Last week, we introduced the idea of a state space, and its use for planning trajectories from some starting state to a goal. Our assumptions in that work were that we knew the initial state, and that the actions could be executed without error. That is a useful idealization in many cases, but it is also very frequently false. Even navigation through a city can fail on both counts: sometimes we don't know where we are on a map, and sometimes, due to traffic or road work or bad driving, we fail to execute a turn we had intended to take.

In such situations, we have some information about where we are: we can make observations of our local surroundings, which give us useful information; and we know what actions we have taken and the consequences those are likely to have on our location. So, the question is: how can we take information from a sequence of actions and local observations and integrate it into some sort of estimate of where we are? What form should that estimate take?



Bigram: Transition Model

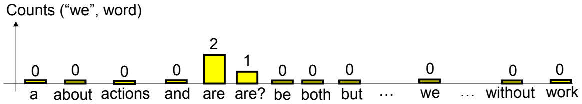
Last week, we introduced the idea of a state space, and its use for planning trajectories from some starting state to a goal. Our assumptions in that work were that we knew the initial state, and that the actions could be executed without error. That is a useful idealization in many cases, but it is also very frequently false. Even navigation through a city can fail on both counts: sometimes we don't know where we are on a map, and sometimes, due to traffic or road work or bad driving, we fail to execute a turn we had intended to take.

In such situations, we have some information about where we are: we can make observations of our local surroundings, which give us useful information; and we know what actions we have taken and the consequences those are likely to have on our location. So, the question is: how can we take information from a sequence of actions and local observations and integrate it into some sort of estimate of where we are? What form should that estimate take?

Bigram: Transition Model

Last week, we introduced the idea of a state space, and its use for planning trajectories from some starting state to a goal. Our assumptions in that work were that we knew the initial state, and that the actions could be executed without error. That is a useful idealization in many cases, but it is also very frequently false. Even navigation through a city can fail on both counts: sometimes we don't know where we are on a map, and sometimes, due to traffic or road work or bad driving, we fail to execute a turn we had intended to take.

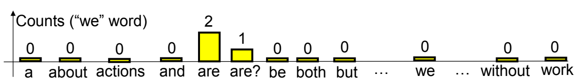
In such situations, we have some information about where we are, we can make observations of our local surroundings, which give us useful information; and we know what actions we have taken and the consequences those are likely to have on our location. So, the question is: how can we take information from a sequence of actions and local observations and integrate it into some sort of estimate of where we are? What form should that estimate take?



Estimation of state transition model

Number of words = 179

Number of distinct words = 112

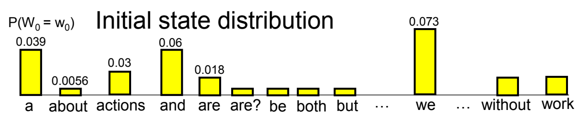


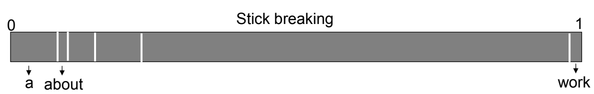
$$P(W_t = \text{"a"} \mid W_{t-1} = \text{"we"}) \approx \frac{\text{Counts ("we", "a")}}{\text{Counts ("we")}} = 0 / 13 = 0$$

$$P(W_t = \text{"are"} \mid W_{t-1} = \text{"we"}) \approx \frac{\text{Counts ("we", "are")}}{\text{Counts ("we")}} = 2 / 13 = 0.15$$

$$\sum_{\text{w in Dictionary}} P(W_t = w \mid W_{t-1} = \text{"we"}) = 1$$

Generating Text

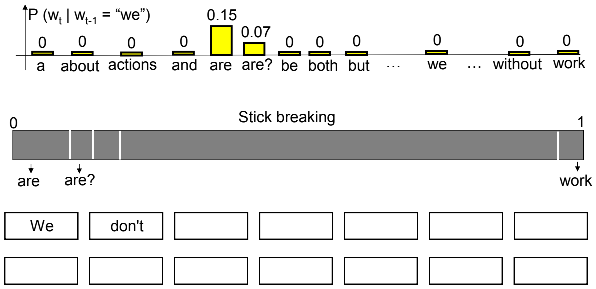




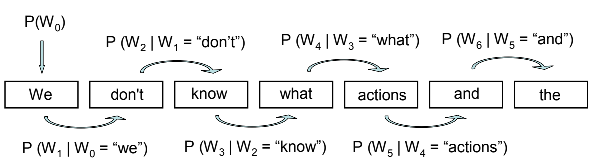
We						

Generating Text

State transition model



Generating Text



Markov chain

Generating text

We don't know what actions and the actions we had intended to execute a goal. Our assumptions in many cases, but it into some sort of our local observations and its use for planning trajectories from some information about where we are on our location. So, the initial state, and integrate it is also very frequently false. Even navigation through a goal. Our assumptions in many cases, but it into some sort of actions and sometimes, due to execute a turn we fail to execute a state to have some sort of our location. So, the question is: how can …

Comparison of the two models

Sentence = $\{W_0, W_1, W_2, \ldots, W_N\}$

Bag of words model

$P(W_0, \ldots, W_N) = P(W_0) \cdot P(W_1) \cdot \ldots \cdot P(W_N)$

Independence assumption

Bigram model

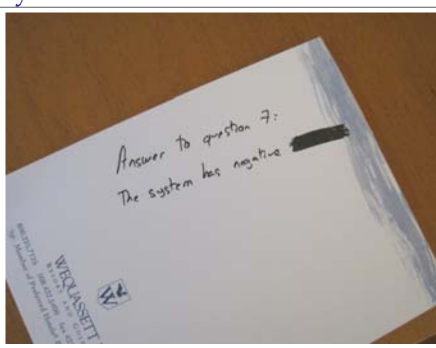
$P(W_0, \ldots, W_N) = P(W_0) \cdot P(W_1|W_0) \cdot \ldots \cdot P(W_N|W_{N-1})$

Markov assumption

Why is it useful to build models?

When is a model good enough / useful?

Noisy Observations



The + system + has + negative + (8 letters word)

Observation Model

The sentence is a sequence of words, but now the words are hidden. We only observe something that depends on the hidden words.

Observation Model:

$$P(O_t = o_t \mid W_t = w_t)$$

If observations are the number of letters on a word:

$$o_t = \text{length word } w_t$$

$$P(O_t = 1 \mid W_t = \text{“the”}) = 0$$

$$P(O_t = 2 \mid W_t = \text{“the”}) = 0$$

$$P(O_t = 3 \mid W_t = \text{“the”}) = 1$$

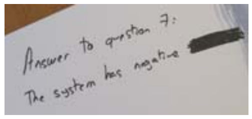
…

Estimation of the Hidden State

Let's start with the memoryless faculty model: words are independent of each other.

Can we predict the hidden word?

$P(W_5 = w_5 \mid O_5 = 8) = ?$



Intuition:
1. Select all words of 8 letters in the dictionary.
2. Then, take their frequencies and normalize them so that they sum to 1.
3. Select the word with the highest probability.

Estimation of the Hidden State

Intuition:
1. Select all words of 8 letters in the dictionary.
2. Then, take their frequencies and normalize them so that they sum to 1.
3. Select the word with the highest probability.

With math:

$$P(W_t = w_t \mid O_t = 8) = \frac{P(O_t = 8 \mid W_t = w_t)\, P(W_t = w_t)}{P(O_t = 8)}$$

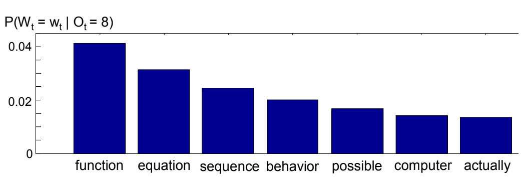
$P(O_t = 8) = ?$

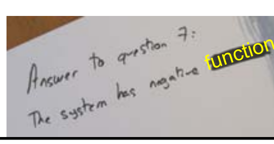
$$P(O_t = 8) = \underbrace{\sum_w P(O_t = 8 \mid W_t = w)\, P(W_t = w)}_{\text{Sum of probabilities of all words of length 8}}$$

Estimation of the Hidden State

We need a good model of 6.01, so I will use **all** the course notes:

53181 words, and 7463 distinct words





Hidden Markov Model

- Initial State Distribution

$$P(W_0 = w_0)$$

- State Transition Model (Bigram model)

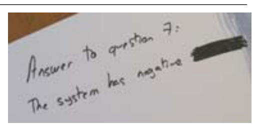
$$P(W_t = w_t \mid W_{t-1} = w_{t-1})$$

- Observation Model

$$P(O_t = o_t \mid W_t = w_t)$$

Estimation of the Hidden State

Can we predict the hidden word?



$$P(W_5 = w_5 \mid O_5 = 8, W_4 = \text{“negative”}) = ?$$

Bayes' rule

$$P(W_5 = w_5 \mid O_5 = 8, W_4 = \text{“negative”}) =$$

$$= \frac{P(O_5 = 8 \mid W_5 = w_5, W_4 = \text{“negative”}) \cdot P(W_5 = w_5 \mid W_4 = \text{“negative”})}{P(O_5 = 8 \mid W_4 = \text{“negative”})}$$

$$= \frac{P(O_5 = 8 \mid W_5 = w_5) \cdot P(W_5 = w_5 \mid W_4 = \text{“negative”})}{P(O_5 = 8 \mid W_4 = \text{“negative”})}$$

Estimation of the Hidden State

$$P(W_5 = w_5 \mid O_5 = 8, W_4 = \text{“negative”}) =$$

$$= \frac{P(O_5 = 8 \mid W_5 = w_5) \cdot P(W_5 = w_5 \mid W_4 = \text{“negative”})}{P(O_5 = 8 \mid W_4 = \text{“negative”})}$$

$$P(O_5 = 8 \mid W_4 = \text{“negative”}) = ?$$

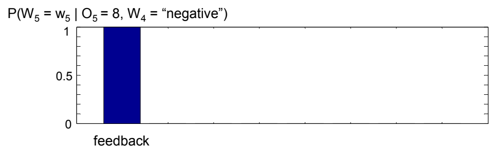
As before, we can calculate this with values we already know

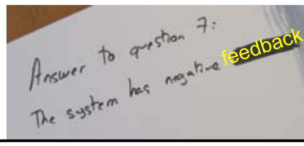
$$P(O_5 = 8 \mid W_4 = \text{“negative”}) = \sum_w P(O_5 = 8 \mid W_5 = w) \cdot P(W_5 = w \mid W_4 = \text{“negative”})$$

Estimation of the Hidden State

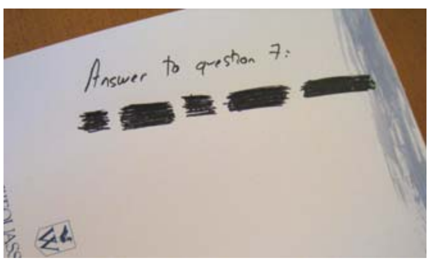
Using **all** the course notes: 7463 distinct words

$P(W_t{=}w_t \mid W_{t-1}{=}w_{t-1})$ Is a matrix with more than 50 million entries!





Estimation of the Hidden States

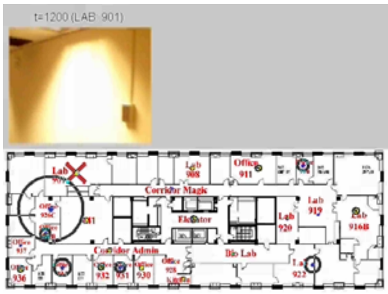


$o_1 = 3$, $o_2 = 6$, $o_3 = 3$, $o_4 = 7$, $o_5 = 8$

Applications

- Noisy image of an object
- Speech recognition
- Robot localization

Localization



Observations = images

States = location (offices, corridor, conference room)

Transition model = encode topology of the space