

Active Learning for Hidden Markov Models

Brigham Anderson, Andrew Moore
brigham@cmu.edu, awm@cs.cmu.edu

Computer Science
Carnegie Mellon University

Outline

1. Active Learning

2. Hidden Markov Models

3. Active Learning + Hidden Markov Models

Notation

We Have:

1. Dataset, D
2. Model parameter space, W
3. Query algorithm, q

Dataset (*D*) Example

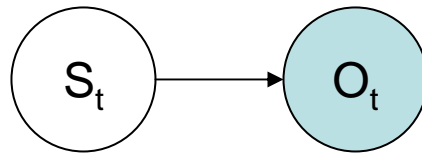
t	Sex	Age	Test A	Test B	Test C	Disease
0	M	40-50	0	1	1	?
1	F	50-60	0	1	0	?
2	F	30-40	0	0	0	?
3	F	60+	1	1	1	?
4	M	10-20	0	1	0	?
5	M	40-50	0	0	1	?
6	F	0-10	0	0	0	?
7	M	30-40	1	1	0	?
8	M	20-30	0	0	1	?

Notation

We Have:

1. Dataset, D ✓
2. Model parameter space, W
3. Query algorithm, q

Model Example



Probabilistic Classifier

Notation

T : Number of examples

O_t : Vector of features of example t

S_t : Class of example t

Model Example

Patient state (S_t)

S_t : DiseaseState

Patient Observations (O_t)

O_{t1} : Gender

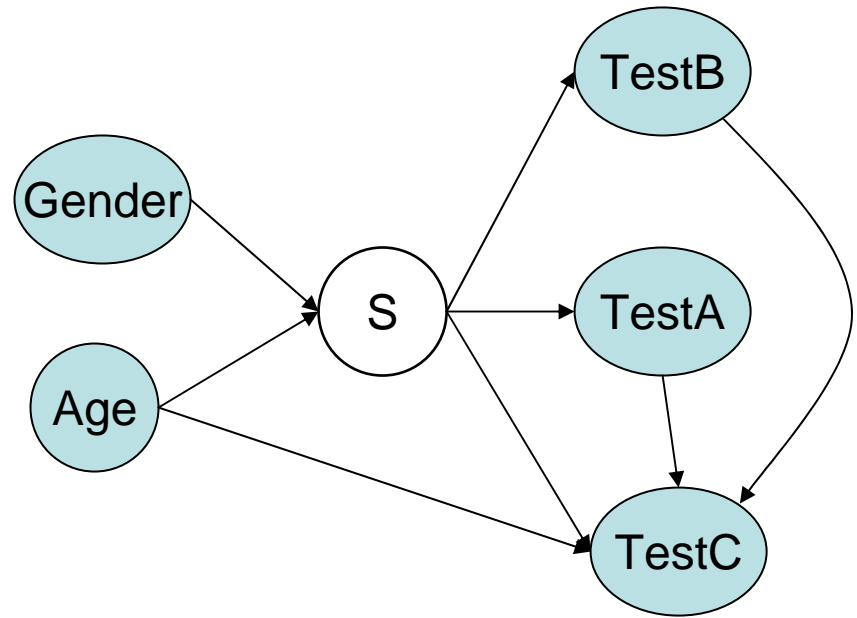
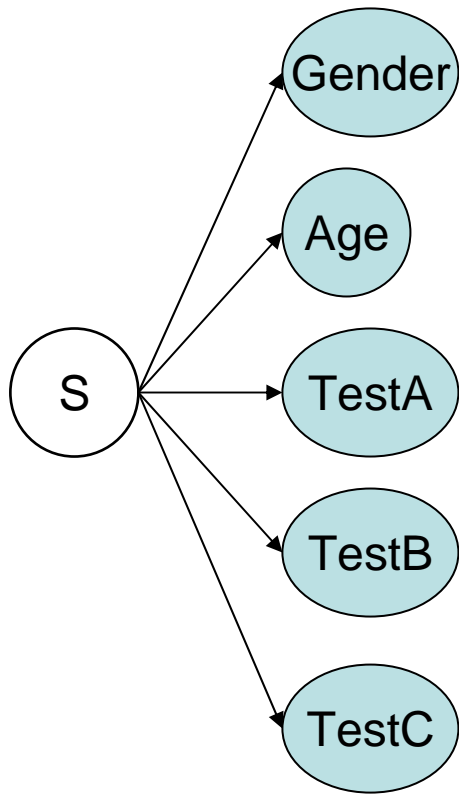
O_{t2} : Age

O_{t3} : TestA

O_{t4} : TestB

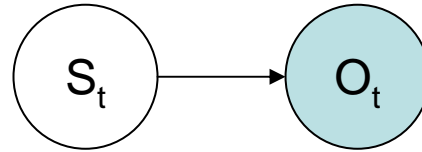
O_{t5} : TestC

Possible Model Structures



Model Space

Model:



Model Parameters:

$P(S_t)$

$P(O_t|S_t)$

Generative Model:

Must be able to compute $P(S_t=i, O_t=o_t | w)$

Model Parameter Space (W)

- W = space of possible parameter values
- Prior on parameters: $P(W)$
- Posterior over models: $P(W | D) \propto P(D | W)P(W)$
 $\propto \prod_t^T P(S_t, O_t | W)P(W)$

Notation

We Have:

1. Dataset, D ✓

2. Model parameter space, W ✓

3. Query algorithm, q



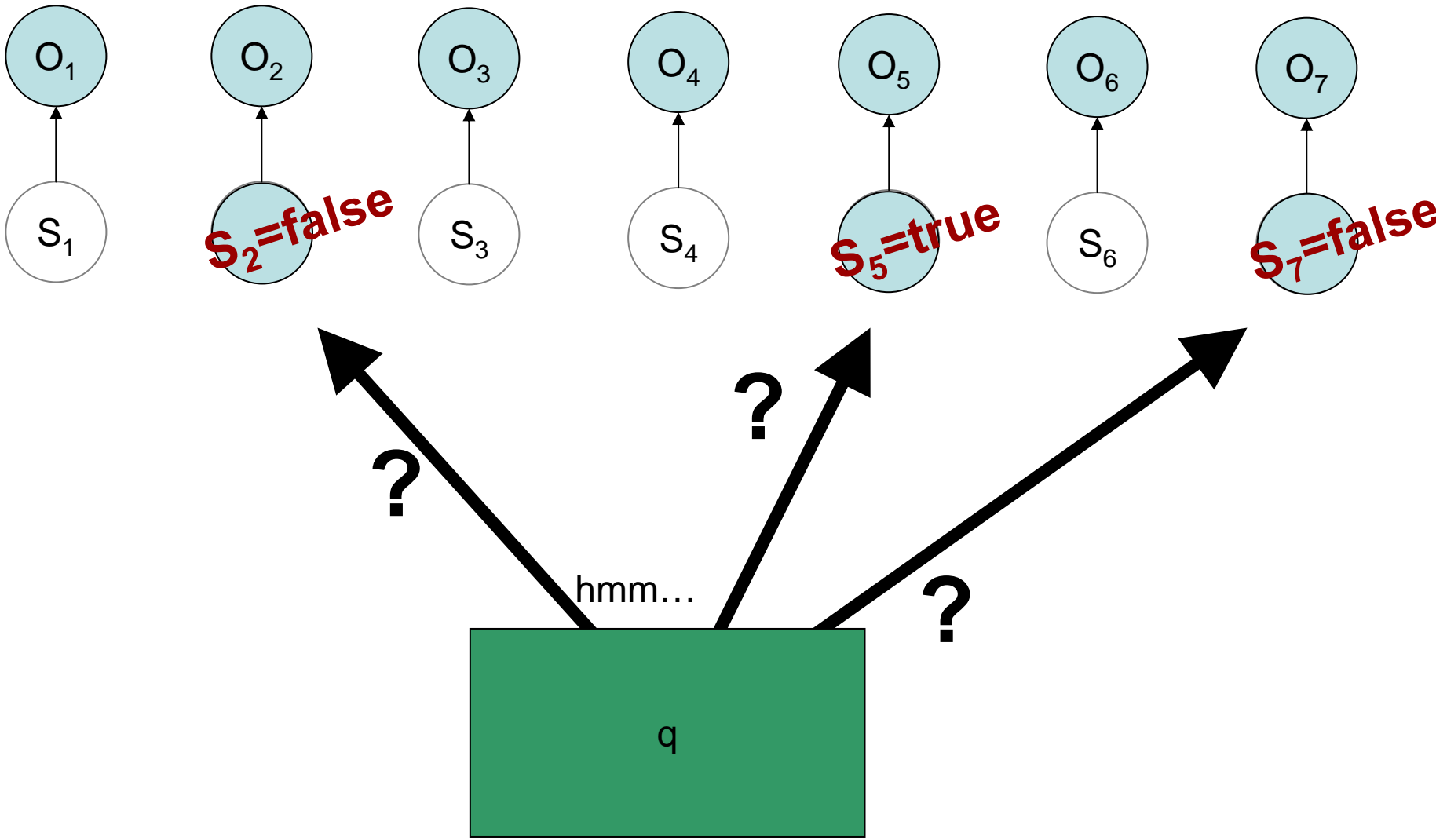
$q(W,D)$ returns t^* , the next sample to label

Game

while NotDone

- Learn $P(W | \mathbf{D})$
- q chooses next example to label
- Expert adds label to \mathbf{D}

Simulation



Active Learning Flavors

- Pool

("random access" to patients)

- Sequential

(must decide as patients walk in the door)

$q?$

- Recall: $q(W, D)$ returns the “most interesting” unlabelled example.
- Well, what makes a doctor curious about a patient?

A Sequential Algorithm for Training Text Classifiers

David D. Lewis (*lewis@research.att.com*) and William A. Gale (*gale@research.att.com*)

AT&T Bell Laboratories; Murray Hill, NJ 07974; USA

In W. Bruce Croft and C. J. van Rijsbergen, eds., SIGIR 94: Proceedings of Seventeenth Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, Springer-Verlag, London, pp. 3–12.

Abstract

The ability to cheaply train text classifiers is critical to their use in information retrieval, content analysis, natural language processing, and other tasks involving data which is partly or fully textual. An algorithm for sequential sampling during machine learning of statistical classifiers was developed and tested on a newswire text categorization task. This method, which we call uncertainty sampling, reduced by as much as 500-fold the amount of training data that would have to be manually classified to achieve a given level of effectiveness.

Score Function

$$\text{score}_{\text{uncert}}(S_t) = \text{uncertainty}(P(S_t | O_t))$$

$$= H(S_t)$$

$$= \sum_i P(S_t = i) \log P(S_t = i)$$

Uncertainty Sampling Example

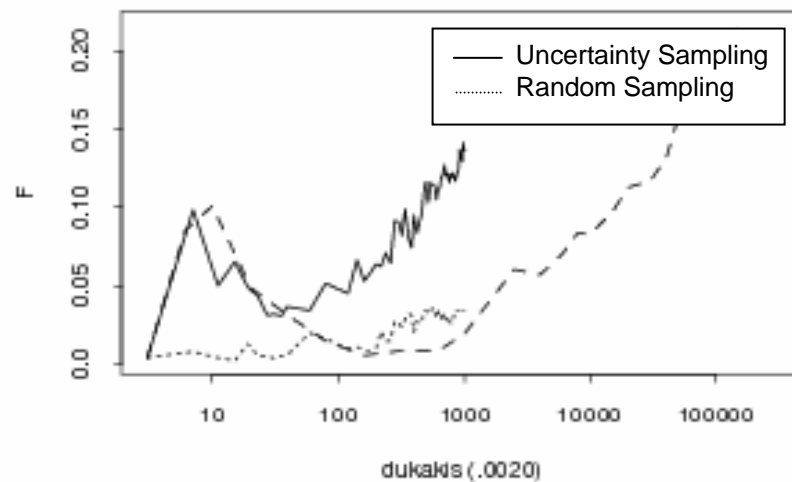
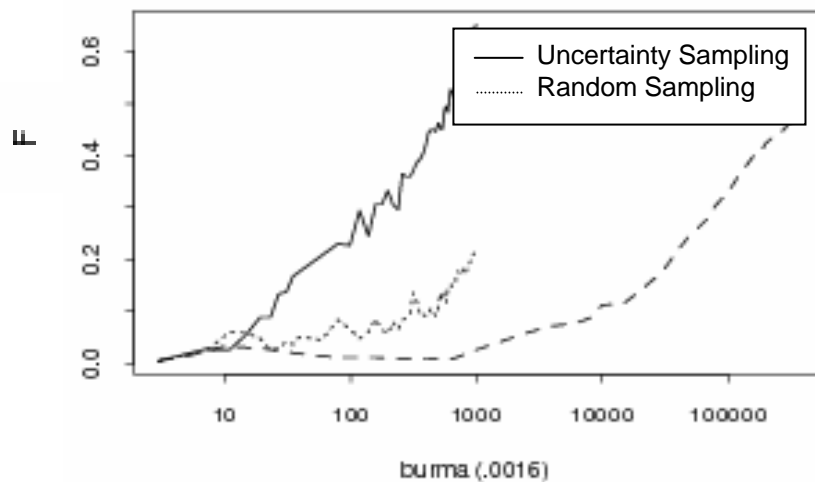
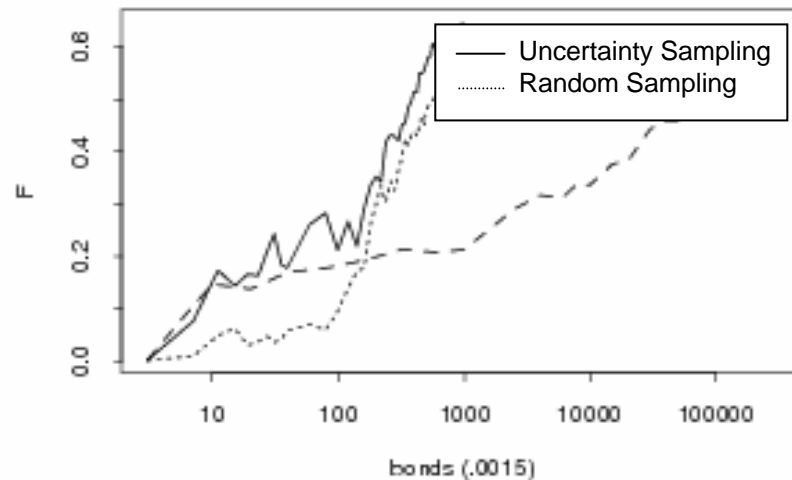
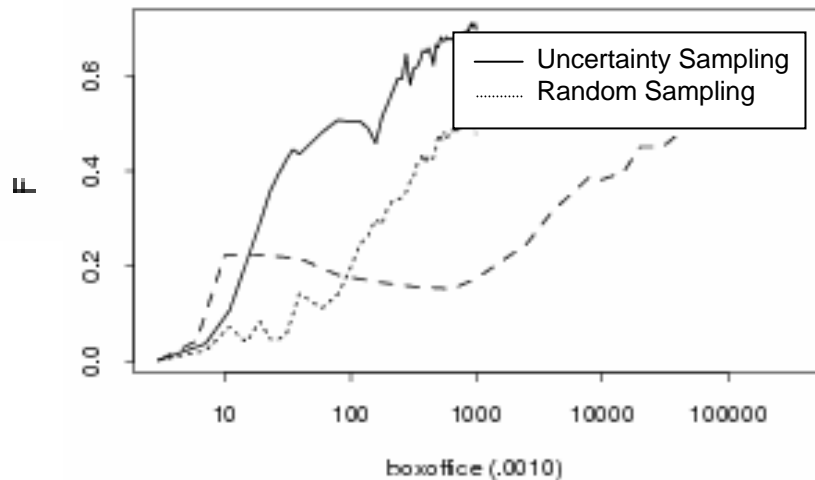
t	Sex	Age	Test A	Test B	Test C	S_t	$P(S_t)$	$H(S_t)$
1	M	20-30	0	1	1	?	0.02	0.043
2	F	20-30	0	1	0	?	0.01	0.024
3	F	30-40	1	0	0	?	0.05	0.086
4	F	60+	1	1	0	FALSE	0.12	0.159
5	M	10-20	0	1	0	?	0.01	0.024
6	M	20-30	1	1	1	?	0.96	0.073

Uncertainty Sampling Example

t	Sex	Age	Test A	Test B	Test C	S_t	$P(S_t)$	$H(S_t)$
1	M	20-30	0	1	1	?	0.01	0.024
2	F	20-30	0	1	0	?	0.02	0.043
3	F	30-40	1	0	0	?	0.04	0.073
4	F	60+	1	1	0	FALSE	0.00	0.00
5	M	10-20	0	1	0	TRUE	0.06	0.112
6	M	20-30	1	1	1	?	0.97	0.059

A Sequential Algorithm for Training Text Classifiers

David D. Lewis (*lewis@research.att.com*) and William A. Gale (*gale@research.att.com*)

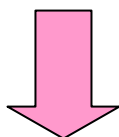


Uncertainty Sampling

GOOD: couldn't be easier

GOOD: often performs pretty well

BAD: $H(S_t)$ measures information gain about the ***samples***, not the ***model***



Sensitive to noisy samples

Can we do better than
uncertainty sampling?

Query by Committee

H. S. Seung*

Racah Institute of Physics and
Center for Neural Computation
Hebrew University
Jerusalem 91904, Israel
seung@mars.huji.ac.il

M. Opper†

Institut für Theoretische Physik
Justus-Liebig-Universität Giessen
D-6300 Giessen, Germany
manfred.opper@
physik.uni-giessen.dbp.de

H. Sompolinsky

Racah Institute of Physics and
Center for Neural Computation
Hebrew University
Jerusalem 91904, Israel
haim@galaxy.huji.ac.il

Abstract

We propose an algorithm called *query by committee*, in which a committee of students is trained on the same data set. The next query is chosen according to the principle of maximal disagreement. The algorithm is studied for two toy models: the high-low game and perceptron learning of another perceptron. As the number of queries goes to infinity, the committee algorithm yields asymptotically finite information gain. This leads to generalization error that decreases exponentially with the number of examples. This is in marked contrast to learning

Strategy #2: Query by Committee

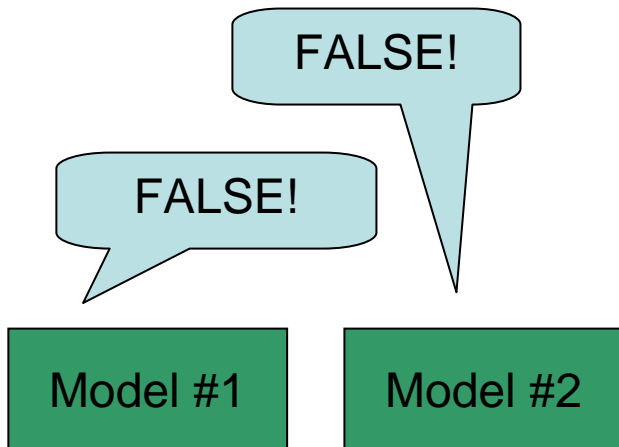
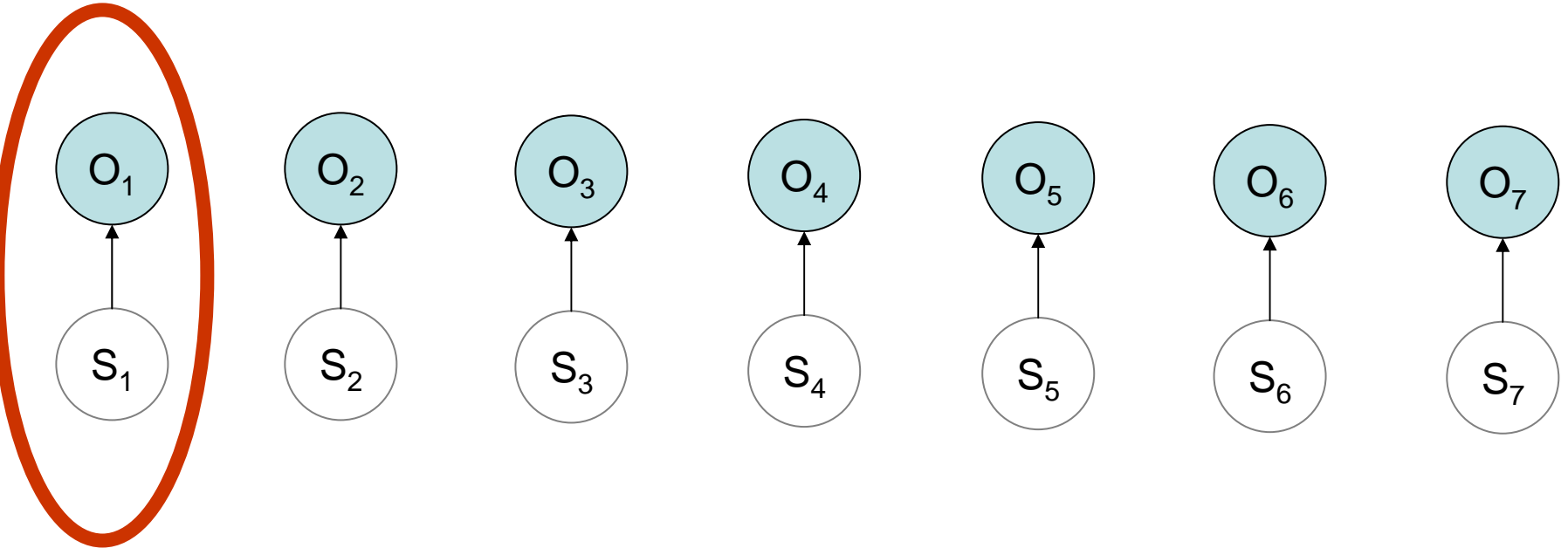
Temporary Assumptions:

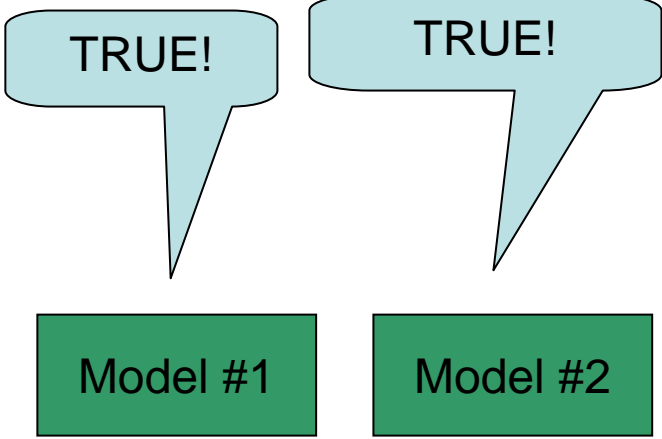
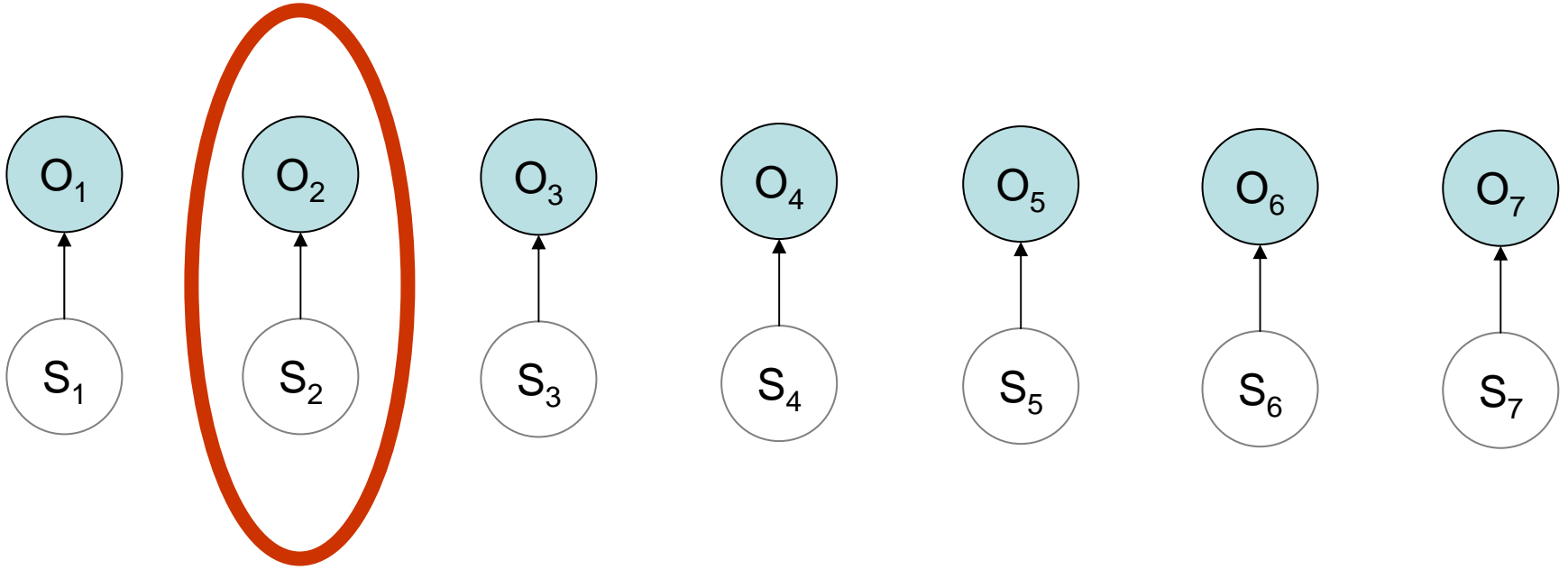
Pool → Sequential

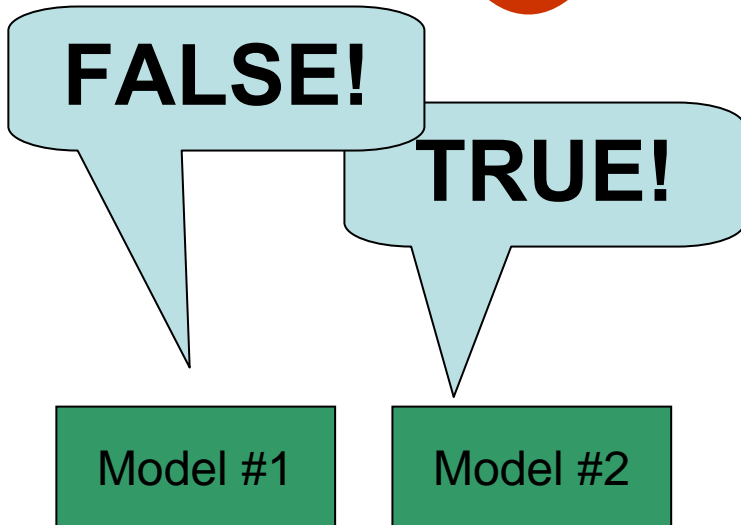
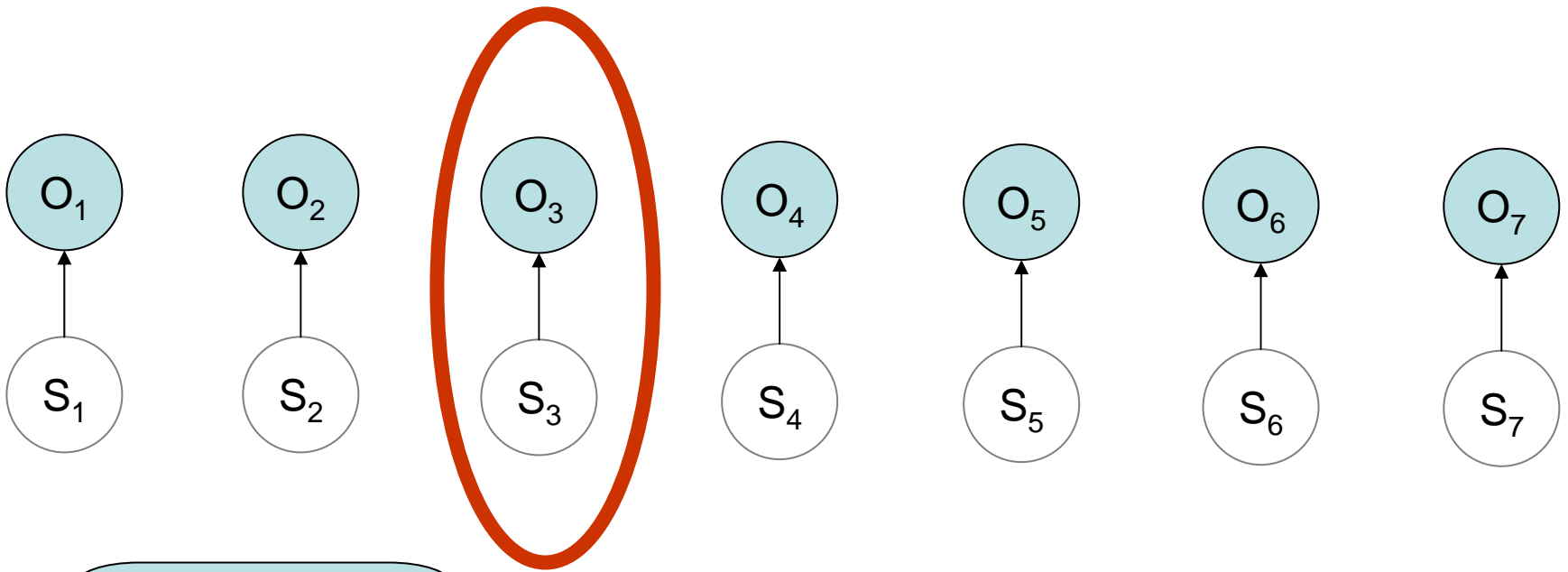
$P(W / \mathbf{D})$ → Version Space

Probabilistic → Noiseless

- QBC attacks the size of the “Version space”







Ooh, now we're going to learn something for sure!

One of them is definitely wrong.

The Original QBC Algorithm

As each example arrives...

1. Choose a committee, C , (usually of size 2) randomly from Version Space
2. Have each member of C classify it
3. If the committee disagrees, select it.

Query by Committee

H. S. Seung*

Racah Institute of Physics and
Center for Neural Computation
Hebrew University
Jerusalem 91904, Israel
seung@mars.huji.ac.il

M. Opper†

Institut für Theoretische Physik
Justus-Liebig-Universität Giessen
D-6300 Giessen, Germany
manfred.opper@
physik.uni-giessen.dbp.de

H. Sompolinsky

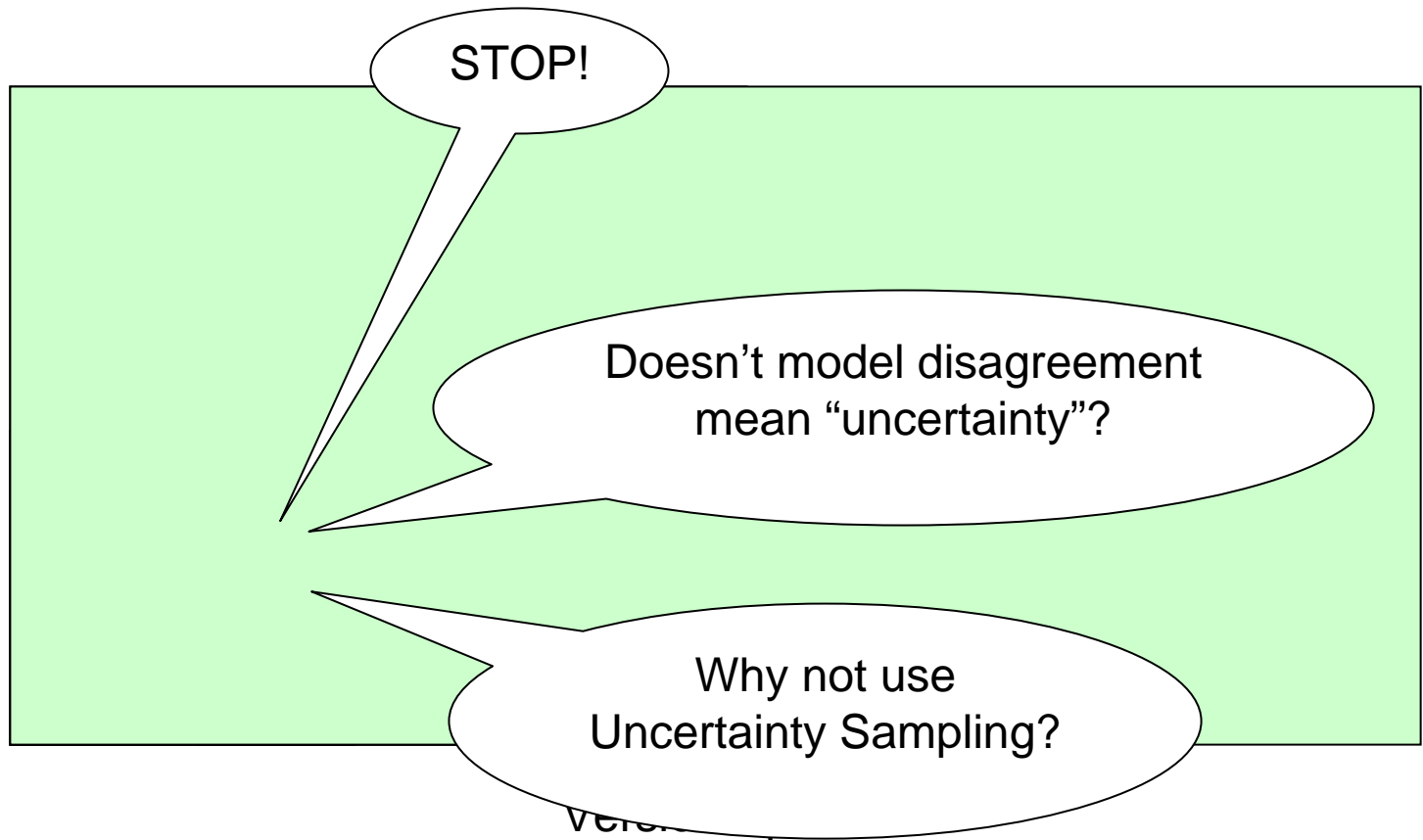
Racah Institute of Physics and
Center for Neural Computation
Hebrew University
Jerusalem 91904, Israel
haim@galaxy.huji.ac.il

Abstract

We propose an algorithm called *query by committee*, in which a committee of students is trained on the same data set. The next query is chosen according to the *principle of maximal disagreement*. The algorithm is studied for two toy models: the high-low game and perceptron learning of another perceptron. As the number of queries goes to infinity, the committee algorithm yields asymptotically finite information gain. This leads to generalization error that decreases exponentially with the number of examples. This is in marked contrast to learning from randomly chosen inputs, for which the in-

QBC:

Choose “Controversial” Examples



Remember our whiny objection to Uncertainty Sampling?

“ $H(S_t)$ measures information gain about the *samples* not the *model*.”

BUT... If the source of the sample uncertainty is model uncertainty, then they equivalent!

Why?

Symmetry of mutual information.

(1995)

Committee-Based Sampling For Training Probabilistic Classifiers

Ido Dagan and **Sean P. Engelson**

Department of Mathematics and Computer Science

Bar-Ilan University

52900 Ramat Gan, Israel


{dagan, engelson}@bimacs.cs.biu.ac.il

Abstract

In many real-world learning tasks, it is expensive to acquire a sufficient number of labeled examples for training. This paper proposes a general method for efficiently training probabilistic classifiers, by selecting for training only the more informative examples in a stream of unlabeled examples. The method, *committee-based sampling*, evaluates the informativeness of an example by measuring the degree of disagreement between several model variants. These variants (the committee) are drawn randomly from a probability distribution conditioned by the training set selected so far (Monte-Carlo sampling). The method is particularly attractive because it evaluates the expected information gain from

Dagan-Engelson QBC

For each example...  Note: Pool-based again

1. Choose a committee, C , (usually of size 2) randomly from $P(W / \mathbf{D})$  Note: No more Version space
2. Have each member C classify it
3. Compute the “Vote Entropy” to measure disagreement

$$VE(S_t) = \sum_i \frac{Votes(S_t = i)}{|C|} \log \frac{Votes(S_t = i)}{|C|}$$

How to Generate the Committee?

- This important point is not covered in the talk.
- Vague Suggestions:
 - Good conjugate priors for parameters
 - Importance sampling

OK, we could keep extending QBC,
but let's cut to the chase...

Information-based objective functions for active data selection

David J.C. MacKay

Computation and Neural Systems*

California Institute of Technology 139-74

Pasadena CA 91125

mackay@hope.caltech.edu

Appeared in *Neural Computation* 4 4 pp. 589-603

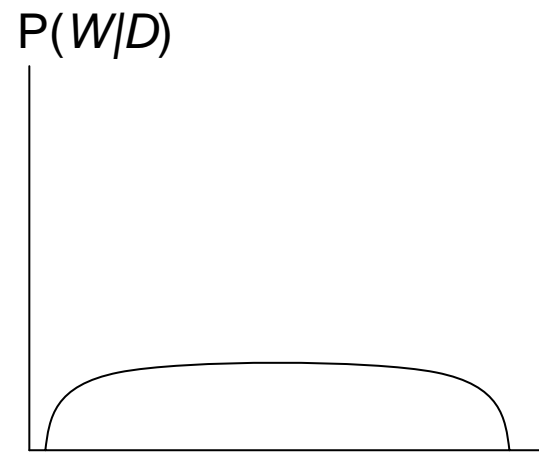
Abstract

Informative with
respect to what?



Learning can be made more efficient if we can actively select particularly salient data points. Within a Bayesian learning framework, objective functions are discussed which measure the *expected informativeness* of candidate measurements. Three alternative specifications of what we want to gain information about lead to three different criteria for data selection. All these criteria depend on the assumption that the hypothesis space is correct, which may prove to be their main weakness.

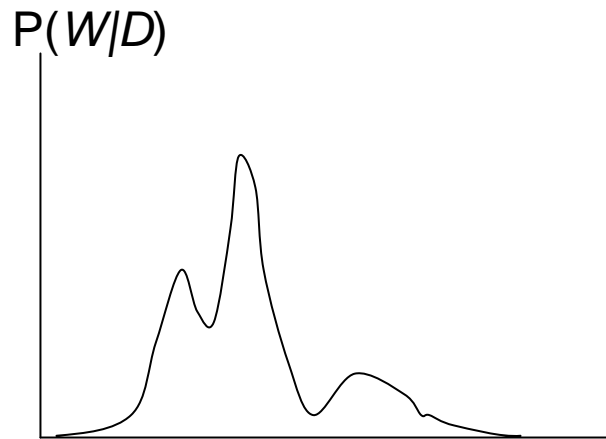
Model Entropy



W



$H(W) = \text{high}$



W



...better...



W



$H(W) = 0$


Information-Gain

- Choose the example that is expected to most reduce $H(W)$
- I.e., Maximize $H(W) - H(W | S_t)$

Current model
space entropy



Expected model space
entropy if we learn S_t



Score Function

$$\text{score}_{IG}(S_t) = MI(S_t; W)$$

$$= H(W) - H(W | S_t)$$

We usually can't just sum over all models to get $H(S_t|W)$

$$H(W) = -\int_w P(w) \log P(w) dw$$

...but we can sample from $P(W | \mathbf{D})$

$$\begin{aligned} H(W) &\propto H(C) \\ &= -\frac{1}{|C|} \sum_{c \in C} P(c) \log P(c) \end{aligned}$$

Conditional Model Entropy

$$H(W) = \int_w P(w) \log P(w) dw$$

$$H(W | S_t = i) = \int_w P(w | S_t = i) \log P(w | S_t = i) dw$$

$$H(W | S_t) = \sum_i P(S_t = i) \int_w P(w | S_t = i) \log P(w | S_t = i) dw$$

Score Function

$$\text{score}_{IG}(S_t) = H(C) - H(C | S_t)$$

t	Sex	Age	Test A	Test B	Test C	S_t
1	M	20-30	0	1	1	?
2	F	20-30	0	1	0	?
3	F	30-40	1	0	0	?
4	F	60+	1	1	1	?
5	M	10-20	0	1	0	?
6	M	20-30	0	0	1	?

$P(S_t)$
0.02
0.01
0.05
0.12
0.01
0.02

Score = $H(C) - H(C S_t)$
0.53
0.58
0.40
0.49
0.57
0.52

Amazing Entropy Fact

Symmetry of Mutual Information

$$\begin{aligned} \text{MI}(A;B) &= H(A) - H(A|B) \\ &= H(B) - H(B|A) \end{aligned}$$

Score Function

$$\begin{aligned} \text{score}_{IG}(S_t) &= H(C) - H(C | S_t) \\ &= H(S_t) - H(S_t | C) \end{aligned}$$

Familiar?

Uncertainty Sampling & Information Gain

$$\text{score}_{\text{Uncertain}}(S_t) = \underline{H(S_t)}$$

$$\text{score}_{\text{InfoGain}}(S_t) = \underline{H(S_t)} - H(S_t | C)$$

- The information gain framework is cleaner than the QBC framework, and easy to build on
- For instance, we don't need to restrict S_t to be the "class" variable

Any Missing Feature is Fair Game

t	Sex	Age	Test A	Test B	Test C	S_t
1	M	20-30	0	1	?	?
2	?	?	0	?	?	?
3	F	30-40	?	?	0	TRUE
4	F	60+	1	?	1	?
5	?	?	?	?	?	FALSE
6	M	20-30	0	?	1	FALSE

Outline

1. Active Learning

2. Hidden Markov Models

3. Active Learning + Hidden Markov Models

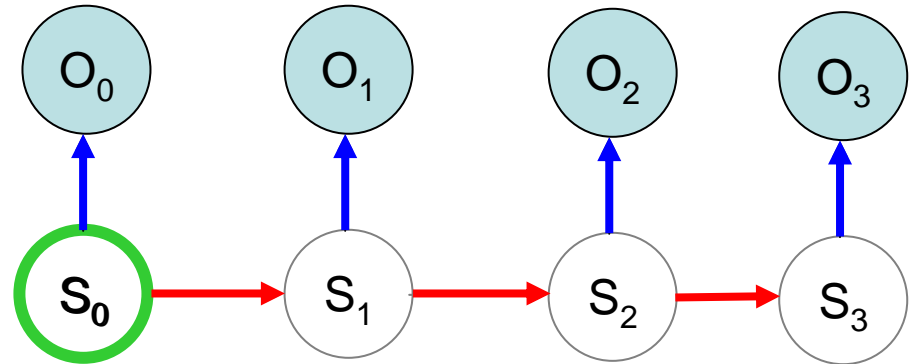
HMMs

Model parameters $W = \{\pi_0, \mathbf{A}, \mathbf{B}\}$

$$\pi_0 = \begin{bmatrix} P(S_0=1) \\ P(S_0=2) \\ \dots \\ P(S_0=n) \end{bmatrix}$$

$$\mathbf{A} = \begin{bmatrix} P(S_{t+1}=1|S_t=1) & \dots & P(S_{t+1}=n|S_t=1) \\ P(S_{t+1}=1|S_t=2) & \dots & P(S_{t+1}=n|S_t=2) \\ \dots & & \\ P(S_{t+1}=1|S_t=n) & \dots & P(S_{t+1}=n|S_t=n) \end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix} P(O=1|S=1) & \dots & P(O=m|S=1) \\ P(O=1|S=2) & \dots & P(O=m|S=2) \\ \dots & & \\ P(O=1|S=n) & \dots & P(O=m|S=n) \end{bmatrix}$$



HMM Light Switch

INPUT

Binary stream of motion / no-motion



OUTPUT

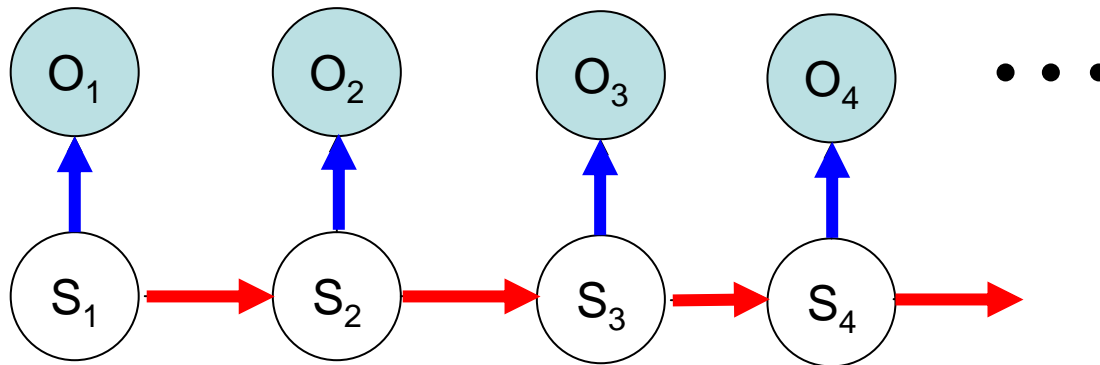
Probability distribution over

- Absent,
- Meeting,
- Computer, and
- Other

E.g.,

“There is an 86% chance that the user is in a meeting right now.”

Light Switch HMM



$$\mathbf{A} = \begin{bmatrix} P(S_{t+1}=A|S_t=A) & \dots & P(S_{t+1}=O|S_t=A) \\ P(S_{t+1}=A|S_t=M) & \dots & P(S_{t+1}=O|S_t=M) \\ P(S_{t+1}=A|S_t=C) & \dots & P(S_{t+1}=O|S_t=C) \\ P(S_{t+1}=A|S_t=O) & \dots & P(S_{t+1}=O|S_t=O) \end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix} P(O_t=1 | S_t = \text{Absent}) \\ P(O_t=1 | S_t = \text{Meeting}) \\ P(O_t=1 | S_t = \text{Computer}) \\ P(O_t=1 | S_t = \text{Other}) \end{bmatrix}$$

Canonical HMM Tasks

1. State Estimation

“For each timestep today, what were the probabilities of each state?”

$$P(S_t | O_1 O_2 O_3 \dots O_T, W)$$

Forward-Backward Algorithm

HMM Light Switch

t	O_t
1	0
2	0
3	1
4	1
5	1
6	0
...	...



$P(S_t = \text{Absent})$	$P(S_t = \text{Meeting})$	$P(S_t = \text{Computer})$	$P(S_t = \text{Other})$
1.00	0.0	0.00	0.0
1.00	0.0	0.00	0.0
0.0	0.10	0.80	0.10
0.0	0.11	0.80	0.09
0.0	0.12	0.80	0.08
0.0	0.10	0.78	0.12
...

Outline

1. Active Learning
2. Hidden Markov Models
3. Active Learning + Hidden Markov Models

Active Learning!



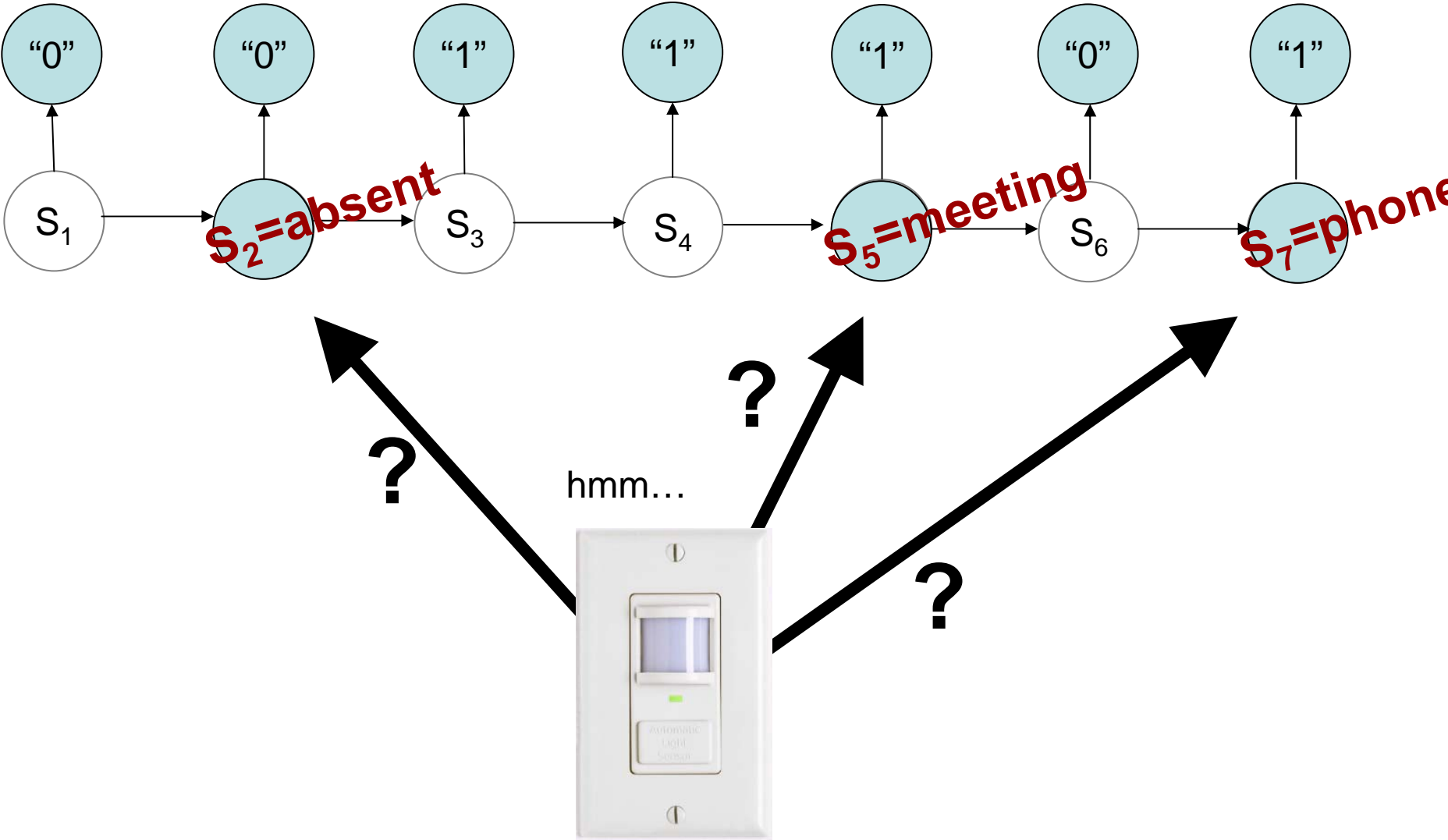
Good Morning Sir!

Here's the video footage of

Good Morning Sir!

Can you tell me what you are doing
in this frame of video?

HMMs and Active Learning



- Note: the dependencies between states does not affect the basic algorithm!
- ...the only change in how we compute $P(S_t|O_{1:T})$
 - (we have to use Forward-Backward.)

HMM Active Learning

1. Choose a committee, C , randomly from $P(W / \mathbf{D})$
2. Run Forward-Backward for each member of c
3. For each timestep, compute $H(S_t) - H(S_t|C)$

Done!

Actively Selecting *Excerpts*

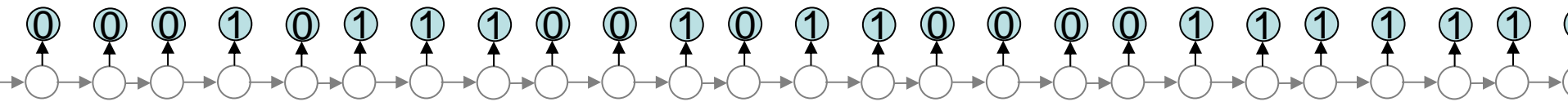


Good Morning Sir!

I'm still trying to learn your HMM.

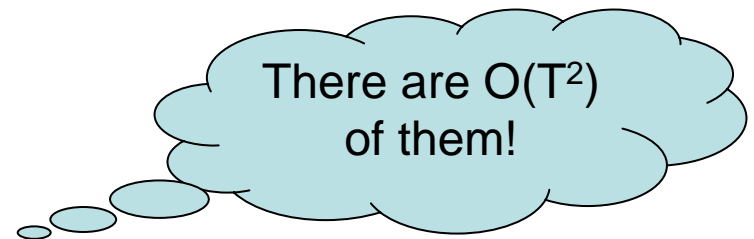
Could you please label the following **scene** from yesterday...

- Finding the optimal “scene” is useful for
 - Selecting scenes from video
 - Selecting utterances from audio
 - Selecting excerpts from text
 - Selecting sequences from DNA



Which sequence should I get labeled?

hmm...



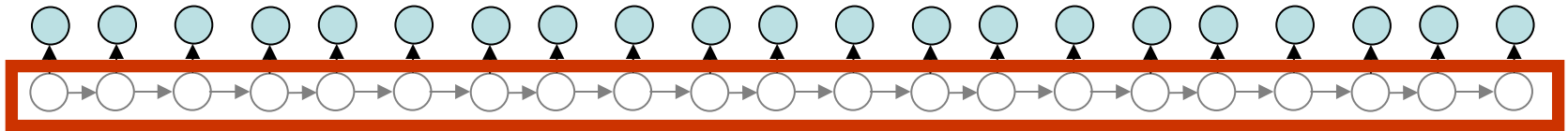
Excerpt Selection

- Let's maximize $H(\mathbf{S}) - H(\mathbf{S}|C)$

Note boldface \mathbf{S} = sequence $(S_t, S_{t+1}, \dots, S_{t+L})$

Trick question:

Which subsequence maximizes $H(\mathbf{S}) - H(\mathbf{S}|C)$?



Uncertainty Sampling for HMMs

We have to include the cost incurred
when we force an expert to sit down and
label 1000 examples...

$$\textit{score}(\mathbf{S}) = H(\mathbf{S}) - H(\mathbf{S} | W) - \alpha|\mathbf{S}|$$

What is the Entropy of a Sequence?

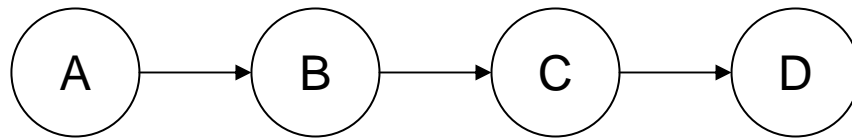
- $H(\mathbf{S}_{1:4}) = H(S_1, S_2, S_3, S_4) = ?$

Amazing Entropy Fact

The Chain Rule

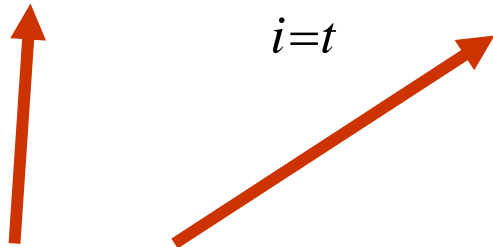
$$H(A,B,C,D) = H(A) + H(B|A) + H(C|A,B) + H(D|A,B,C)$$

...and even better:



$$H(A,B,C,D) = H(A) + H(B|A) + H(C|B) + H(D|C)$$

Entropy of a Sequence

$$H(S_t, S_{t+1}, \dots, S_{t+L}) = H(S_t) + \sum_{i=t}^{t+L} H(S_{i+1} | S_i)$$


We still get the components of these expressions, $P(S_t = i | O_{1:T})$, and $P(S_{t+1}=i | S_t = j, O_{1:T})$, from a Forward-Backward run.

Score of a Sequence

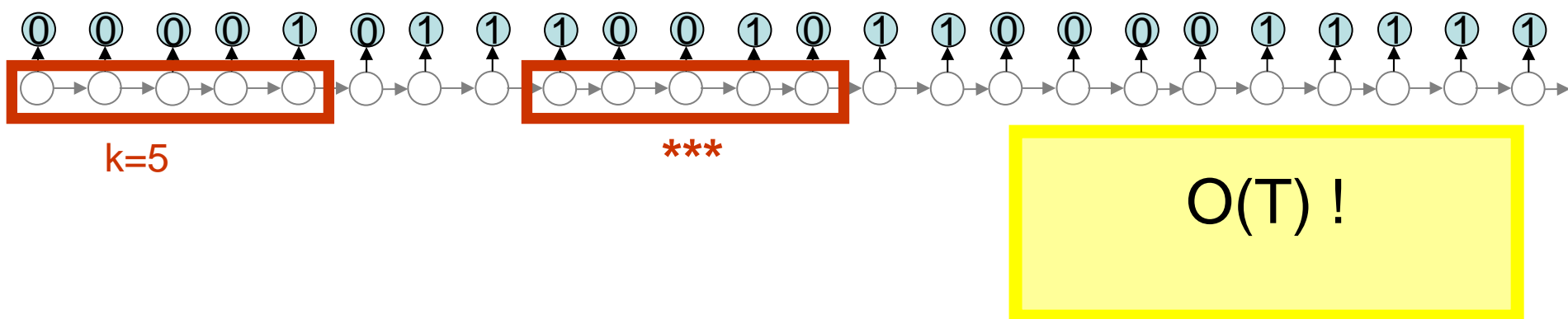
$$\text{score}_{seqIG}(\mathbf{S}) = H(\mathbf{S}) - H(\mathbf{S} | C) - \alpha|\mathbf{S}|$$

$$= \left[H(S_t) + \sum_{i=t}^{t+L} H(S_{i+1} | S_i) \right] - \left[H(S_t | C) + \sum_{i=t}^{t+L} H(S_{i+1} | S_i, C) \right] - \alpha|\mathbf{S}|$$

Finding Best Excerpt of Length k

Find Best Sequence of Length k

1. Draw committee C from $P(W | D)$
2. Run Forward-Backward for each c
3. Scan the entire sequence using $\text{score}_{\text{seqIG}}(\mathbf{S})$



Find Best Excerpt of
Any Length

Find Best Sequence of *Any* Length

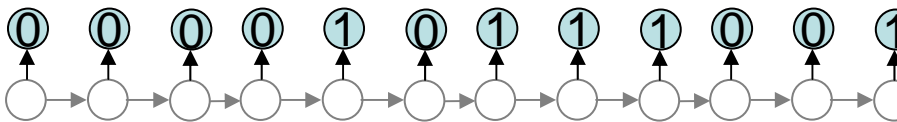
1. Score all possible
2. Pick the best one

Hmm...

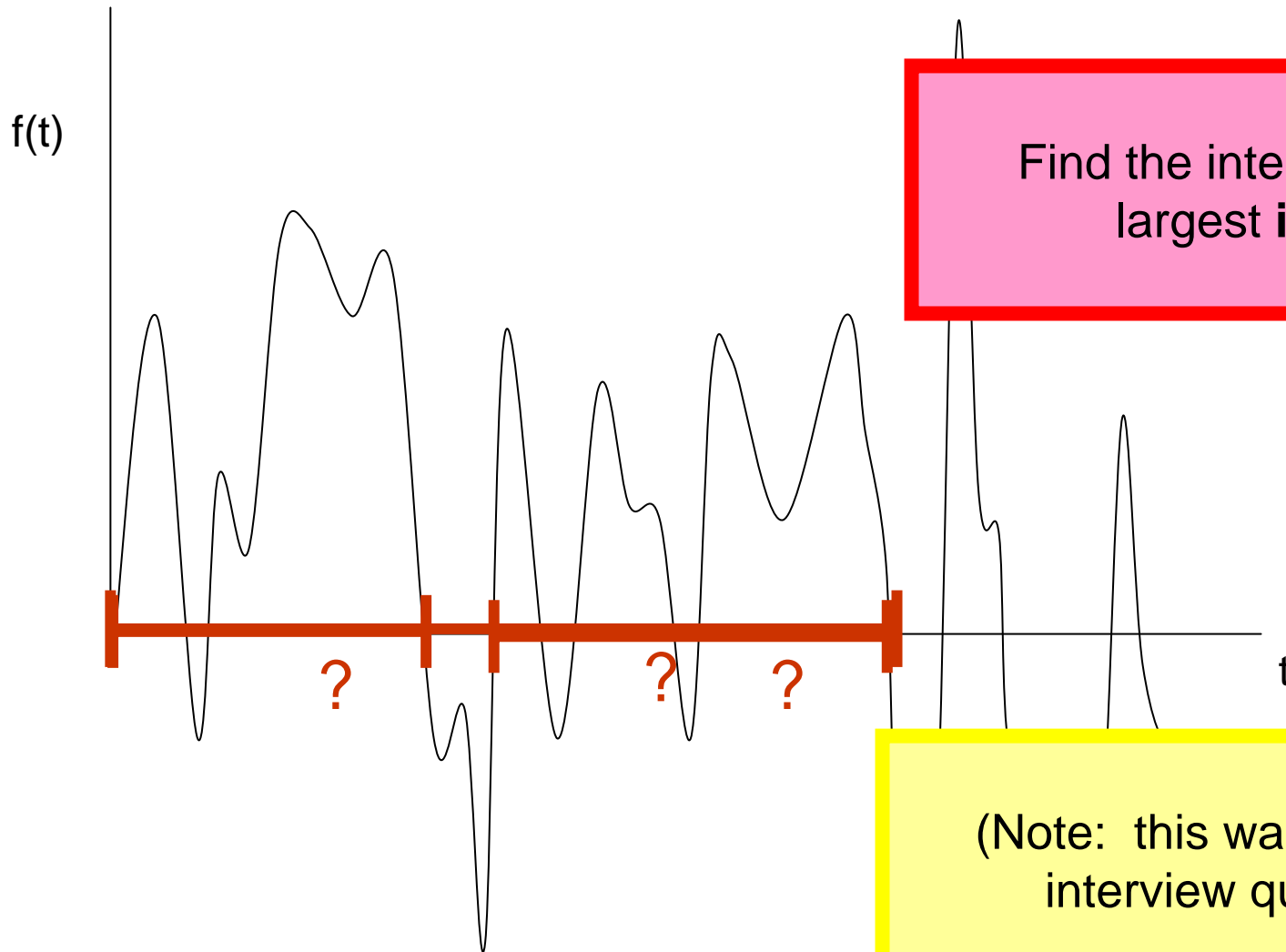
That's $O(T^2)$.

We could cleverly cache
some of the computation as we go...

But we're still going to be $O(T^2)$



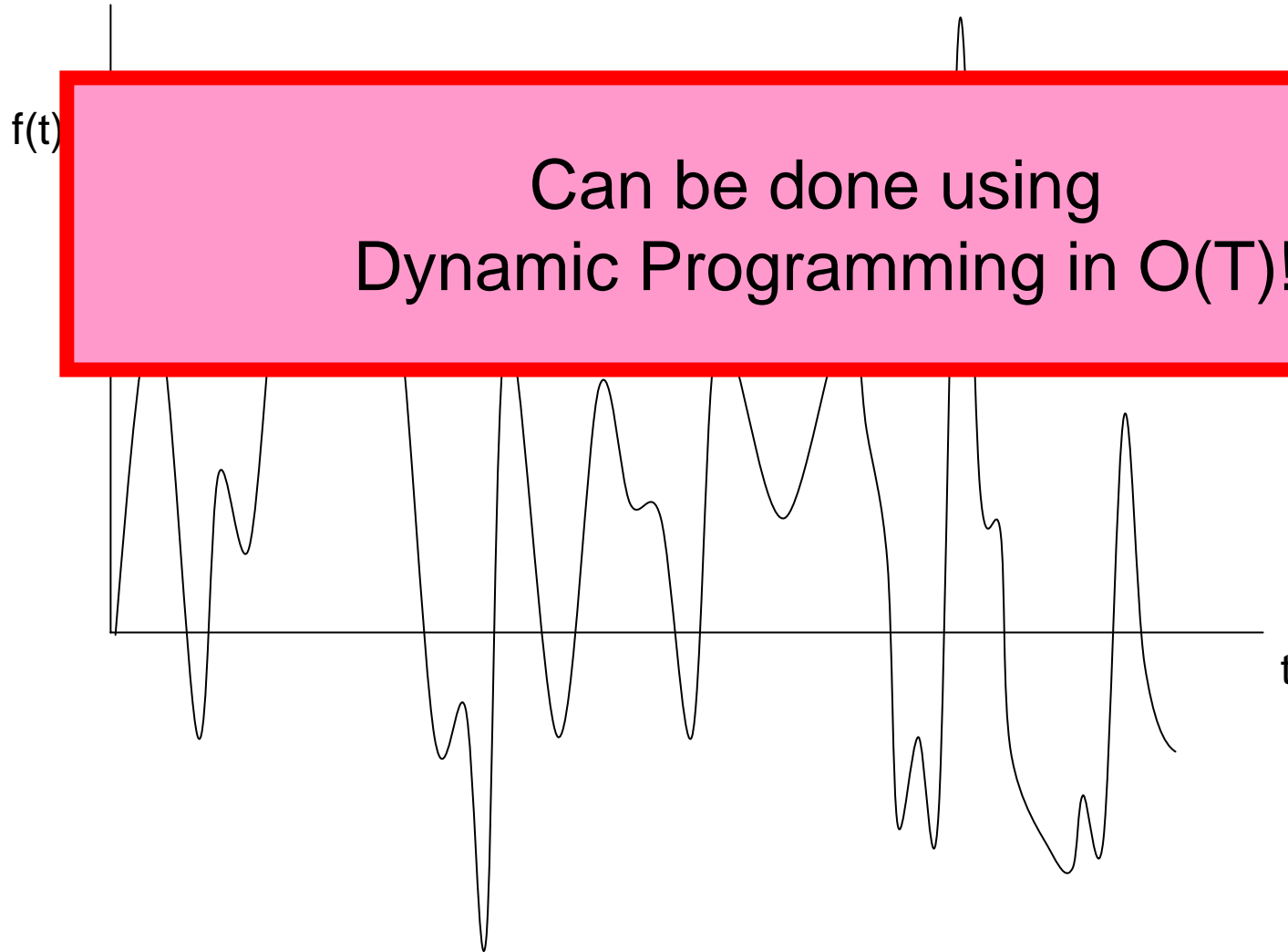
Similar Problem



Find the interval that has largest **integral**

(Note: this was a Google interview question!)

Similar Problem



$$\text{state}(t) = \left[\begin{array}{l} [a^*, b^*] : \text{ best interval so far} \\ a_{\text{temp}} : \text{ start of best interval ending at } t \\ \text{sum}(a^*, b^*) \\ \text{sum}(a_{\text{temp}}, t) \end{array} \right]$$

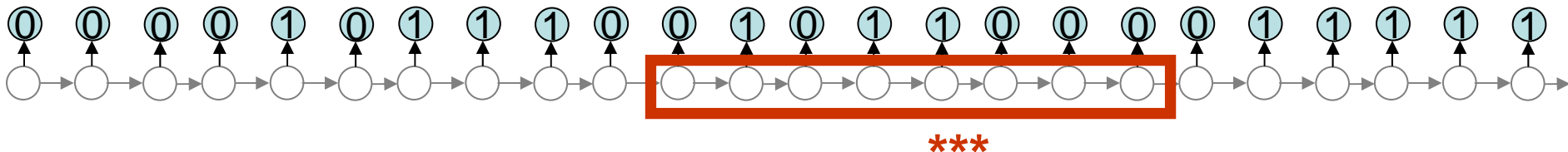
Rules:

if ($\text{sum}(a_{\text{temp}}, t-1) + y(t) < 0$)
then $a_{\text{temp}} = t$

if ($\text{sum}(a_{\text{temp}}, t) > \text{sum}(a^*, b^*)$)
then $[a^*, b^*] = [a_{\text{temp}}, t]$

Find Best Sequence of Any Length

1. Draw committee C from $P(W/D)$
2. Run Forward-Backward for each c
3. Find best-scoring interval using DP



Not Just HMMs

The “max-MI Excerpt” can be applied to any sequential process with the Markov property

E.g., Kalman filters

Aside: Active *Diagnosis*

- What if we're not trying to learn a model?
- What if we have a good model already, and we just want to learn the most about the sequence itself?
- E.g., “An HMM is trying to translate a news broadcast. It doesn't want to learn the model, it just wants the best transcription possible.”

$$\begin{aligned} MI(\mathbf{S}; S_t) &= H(\mathbf{S}) - H(\mathbf{S} | S_t) \\ &= H(S_t) - H(S_t | \mathbf{S}) \\ &= H(S_t) - 0 \\ &= H(S_t) \end{aligned}$$

 **Uncertainty Sampling!**

...we can use the same DP trick
to find the optimal subsequence too

Conclusion

- Uncertainty sampling is *sometimes* correct
- QBC is an approximation to Information Gain
- Finding the most-informative subsequence of a Markov time series is $O(T)$

Light Switch HMM

