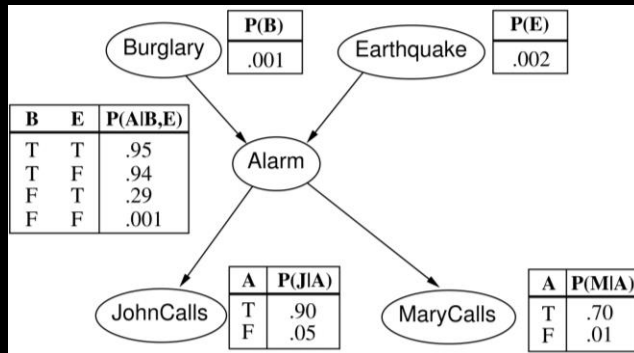


Learning Probabilistic Models

Previously...

- We saw how to infer the probability of an event from a Bayes Net



e.g., $P(j \wedge m \wedge a \wedge \neg b \wedge \neg e)$

$$\begin{aligned} &= P(j|a)P(m|a)P(a|\neg b, \neg e)P(\neg b)P(\neg e) \\ &= 0.9 \times 0.7 \times 0.001 \times 0.999 \times 0.998 \\ &\approx 0.00063 \end{aligned}$$

- But where do the numbers in these probability tables come from?

Outline

- Learning without hidden variables
- Learning with hidden variables
- GMMs

Learning **without** hidden variables

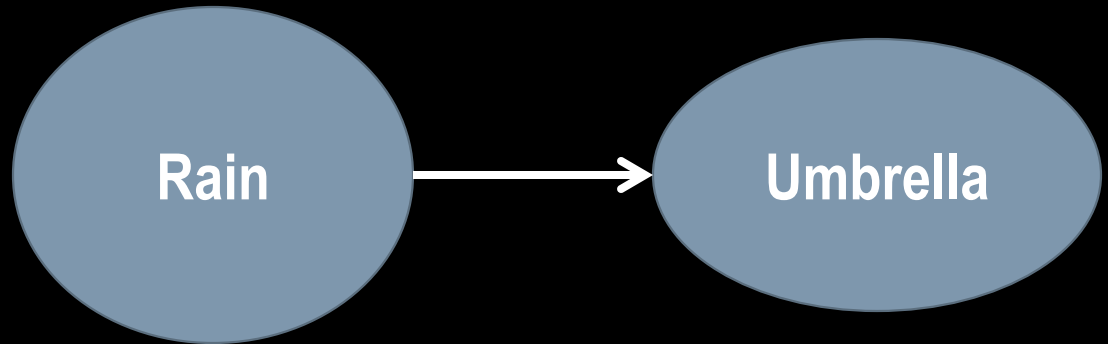
The goal

- Given training data, estimate a parameter
 - E.g., CPT for a network



- We'll look at this in the context of Bayesian networks
- Determine $P(B \mid A=a)$
 - Where does this number come from?

Bayesian networks



- Parameters needed for this network's CPT:
 - $P(\text{rain}=\text{yes})$
 - $P(\text{umbrella} \mid \text{rain} = \{\text{yes}, \text{no}\})$

Learning in a Bayesian framework

- Want to find the “best” hypothesis h_i
 - Best = the one that’s most likely *given the data set d*
 - $\operatorname{argmax}_{h_i} P(h_i | d)$
 - “hypothesis” is candidate for the value of a parameter
- How do we compute $\operatorname{argmax}_{h_i} P(h_i | d)$?

Using Bayes' rule

- Want $P(h_i|d)$
 - $P(h_i|d) = P(d|h_i) * P(h_i) / P(d)$
- What does each term mean?

Using Bayes' rule

- Want $P(h_i|d) = P(d|h_i) * P(h_i) / P(d)$
- What does each term mean?
 - How likely are the data under this hypothesis?
 - How likely is this hypothesis?
 - How likely are the data at all?

Can rewrite it

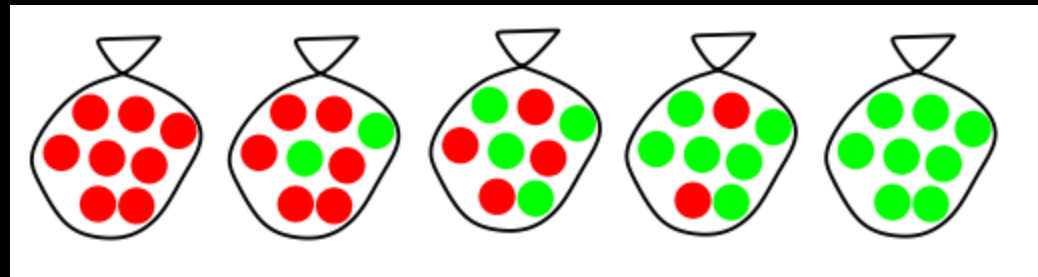
- $P(h_i|d) = P(d|h_i) * P(h_i) / P(d)$
 - Insight: $P(d)$ is constant across hypotheses
 - Note this is *unconditional* probability of the data
 - The *conditional* probability of the data ($P(d|h_i)$) is *not* constant across hypotheses
 - A constant (η) does not affect the result of the argmax
 - So we don't need to worry about it
- $P(h_i|d) = \eta P(d|h_i) * P(h_i)$

Computing the first term

- $P(h_i|d) = \eta \mathbf{P(d|h_i)} * P(h_i)$
- Want to know how likely the data are under this hypothesis
 - Approach: for each datapoint, compute how likely it is under each hypothesis
- Assumptions
 - Data are an unbiased sample of the underlying distribution
 - Number of hypotheses is enumerable
 - Data are **IID** (!!)
 - **I**ndependent
 - **I**dentically **D**istributed

Example from book

- Observe pieces of candy one at a time
- Know they were drawn from one of the following five bags:
 - H_1 : 100% cherry
 - H_2 : 75% cherry, 25% lime
 - H_3 : 50% cherry, 50% lime
 - H_4 : 25% cherry, 75% lime
 - H_5 : 100% lime
- Data: a set of candies drawn from the bag: **C** (cherry), **L** (lime)



Computing likelihood of data

- Suppose we observe $d = \text{CCL}$
- Maximize $P(h \mid d)$
 - Rewrite as $P(d \mid h)$
- $P(\text{CCL} \mid h) = P(\text{C} \mid h) * P(\text{C} \mid h) * P(\text{L} \mid h)$
 - What is being assumed for this equation to be true?

Independence!

- $P(d_1, d_2, d_3 \mid h) = P(d_1 \mid h) * P(d_2 \mid h) * P(d_3 \mid h)$
 - We cannot separate probabilities like this in general
 - Only works when data are independent
- Remember we assumed IID
 - Independent
 - Identically distributed
- Makes problem tractable

Using independence

- $P(\text{CCL} \mid h) = P(\text{C} \mid h) * P(\text{C} \mid h) * P(\text{L} \mid h)$
- Let's say $h = P(\text{C}) = 1.0$
 - What is $P(\text{L} \mid h)$?
- $P(\text{CCL} \mid h) = P(\text{C} \mid h) * P(\text{C} \mid h) * P(\text{L} \mid h)$
 $1.0 * 1.0 * 0.0 = 0$

Example

- Compute $P(\text{LCL} \mid h)$, where...
 - $h_1 = P(L) = 0.25$
 - $P(\text{LCL} \mid h_1) = ?$
 - $h_2 = P(L) = 0.5$
 - $P(\text{LCL} \mid h_2) = ?$
 - $h_3 = P(L) = 0.75$
 - $P(\text{LCL} \mid h_3) = ?$
- Which hypothesis is most likely?

Example

- Compute $P(\text{LCL} | h)$, where...
 - $h_1 = P(L) = 0.25$
 - $P(\text{LCL} | h_1) = 1/4 * 3/4 * 1/4 = 3 / 64 = 0.047$
 - $h_2 = P(L) = 0.5$
 - $P(\text{LCL} | h_2) = 1/2 * 1/2 * 1/2 = 1/8 = 0.125$
 - $h_3 = P(L) = 0.75$
 - $P(\text{LCL} | h_3) = 3/4 * 1/4 * 3/4 = 9/64 = 0.1406$
- Which hypothesis is most likely?
 - h_3

You just did
Maximum
Likelihood
Estimation
(MLE)!

What if we knew additional information?


- 25% of the bags are 25% lime (h_1)
- 50% of the bags are 50% lime (h_2)
- 25% of the bags are 75% lime (h_3)

- Now which is more likely?

Recall

- Maximize $P(h \mid d)$

- Rewrite as $P(d \mid h) * P(h)$

 We ignored this part before – will use it now

- How likely the hypothesis is matters
 - If some types of bags are more common, that *should* affect our guess

Doing the computation

Recall:

25% of the bags are 25% lime (h_1)

50% of the bags are 50% lime (h_2)

25% of the bags are 75% lime (h_3)

- Compute $P(\text{LCL} | h) * P(h)$, where...
 - $h_1 = P(L) = 0.25 \rightarrow 0.047 * 0.25 = 0.0112$
 - $h_2 = P(L) = 0.5 \rightarrow 0.125 * 0.5 = 0.0625$
 - $h_3 = P(L) = 0.75 \rightarrow 0.141 * 0.25 = 0.0352$
- Now which is most likely?

Doing the computation

- Compute $P(\text{LCL} | h) * P(h)$, where...
 - $h_1 = P(L) = 0.25 \rightarrow 0.047 * 0.25 = 0.0112$
 - $h_2 = P(L) = 0.5 \rightarrow 0.125 * 0.5 = 0.0625 \leftarrow$
 - $h_3 = P(L) = 0.75 \rightarrow 0.141 * 0.25 = 0.0352$
- h_2 (50% lime) is the most likely hypothesis once we consider $P(h)$ (i.e., some types of bags are more likely than others)

We just did **Maximum A Posteriori (MAP)** estimation

$$P(h_i)$$

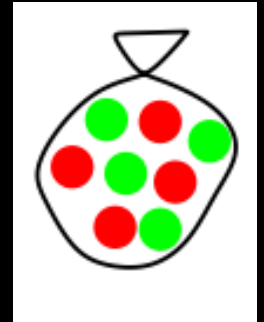
- Does not depend on data
- Is your *prior belief* of how likely a hypothesis is to be true
- Can encode anything you like, e.g.
 - Distribution of different types of bags of candy
 - Whether you think more complex hypotheses are less likely

Review: Two ways of doing estimation

- $\operatorname{argmax}_{h_i} P(h_i|d) = \eta P(d|h_i)$ (MLE)
- $\operatorname{argmax}_{h_i} P(h_i|d) = \eta P(d|h_i) * P(h_i)$ (MAP)
- MAP is same as MLE, but weighted by hypothesis likelihood

Continuous parameter estimation

- Before, we were deciding among a set of discrete hypotheses
- Now, say you now have 1 bag of candy (forget about multiple bags)
- Say you reach into bag and get 6 cherry and 9 lime
 - What is $P(\text{next candy is cherry})$?
 - Intuitively, $6/(6+9) = 40\%$
 - But why does this work?



More formally

- Let's say seen **c** cherry and **l** lime candies
- Let Θ be proportion of bag that is cherry
- Probability of seeing **c** cherry candies
 - Θ^c
 - imagine if $P(\text{C})=0.7$, then $P(\text{CCC}) = 0.7 * 0.7 * 0.7$
- Probability of seeing **l** lime candies
 - $(1 - \Theta)^l$

Figuring out probability of the data

- Probability of cherries: Θ^c
- Probability of limes: $(1 - \Theta)^l$
- How to compose the two?
 - $P(c \text{ cherries}, l \text{ limes}) = P(c \text{ cherries}) * P(l \text{ limes})$
 - What assumption is being used here?
 - Assuming samples are **IID**

Figuring out probability of the data

- Probability of cherries: Θ^c
- Probability of limes: $(1 - \Theta)^l$
- How to compose the two?
 - Multiply them together:
 - $P(c \text{ cherries}) * P(l \text{ limes}) = \Theta^c(1 - \Theta)^l$
- To get Θ^* , the best estimate of probability:
 - Compute $\Theta^* = \underset{\Theta}{\operatorname{argmax}} (\Theta^c(1 - \Theta)^l)$

Computing Θ^*

- Goal is to pick Θ to make the data most likely
- $\Theta^* = \underset{\Theta}{\operatorname{argmax}} \Theta^c (1 - \Theta)^I$

How?

- Trial and error:
 - Let's try $\Theta = 0$
 - $\Theta = 0.01$
 - $\Theta = 0.02$
- What's the problem with this method?

Better approach: use calculus!

- Must check
 - Maximum value (1)
 - Minimum (0)
 - Where derivative is equal to 0
- It turns out, working with the **log likelihood** is easier than directly with probabilities:

$$\underset{\Theta}{\operatorname{argmax}} \log(\Theta^c (1 - \Theta)^I)$$

- Θ^* for log likelihood is the same as for normal probabilities

Advantages of using log likelihood

- Add instead of multiply
 - Faster
 - Easier to take derivatives
- Probabilities can be very small (slight errors in precision make a big deal)
 - Logs increase (absolute) magnitude of such numbers
 - Logs can help avoid floating point precision errors

Optimizing with log likelihood

- How to optimize $\operatorname{argmax}_{\Theta} \log(\Theta^c (1 - \Theta)^l)$?

- Apply log rules:

$$\operatorname{argmax}_{\Theta} [c \log \Theta + l \log (1 - \Theta)]$$

- Take derivate w.r.t to Θ and set equal to 0:

- $c / \Theta - l / (1 - \Theta) = 0$

- Solve for Θ :

- $\Theta = c / (c + l)$

- Recall, in our example, $\Theta = 6 / (6 + 9) = 0.4$

The overall method

- **Question:** Given some data, what probabilistic model accounts for it?
- **Process:**
 1. Write down likelihood of data in terms of Θ
 - (the parameter we want to estimate)
 2. Goal is to maximize the data likelihood $P(d \mid h_{\Theta})$
 - (want the data to be predicted well)
 3. Take the log of the data likelihood for ease
 4. Determine the Θ for which the log likelihood is maximized

Estimating Bayes net parameters

- Given some data and a network structure

- Need to estimate CPTs



- The CPT values for a variable, given its parents, are the observed frequencies of the variable values (from the data) for each setting of the parent values

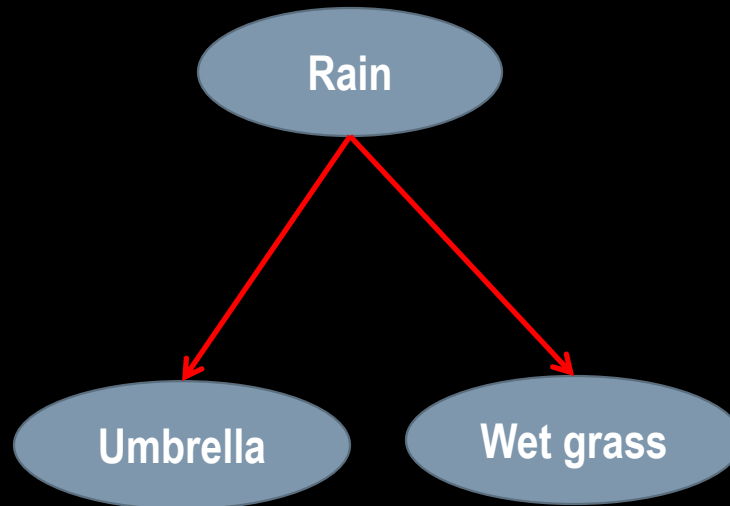
Just like computing $\Theta = c / (c + l)$

- $P(B \mid A = a_1) = P(B, A = a_1) / P(A = a_1)$

Learning with hidden variables (EM)

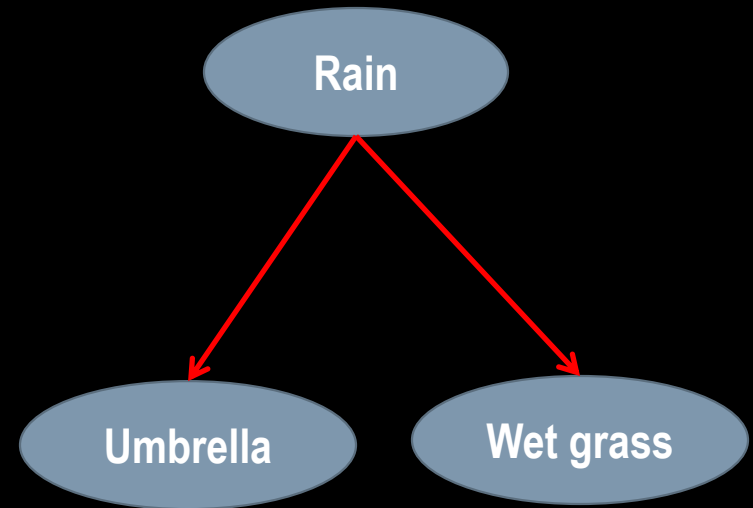
A more complex example

- All 3 nodes are observable and binary



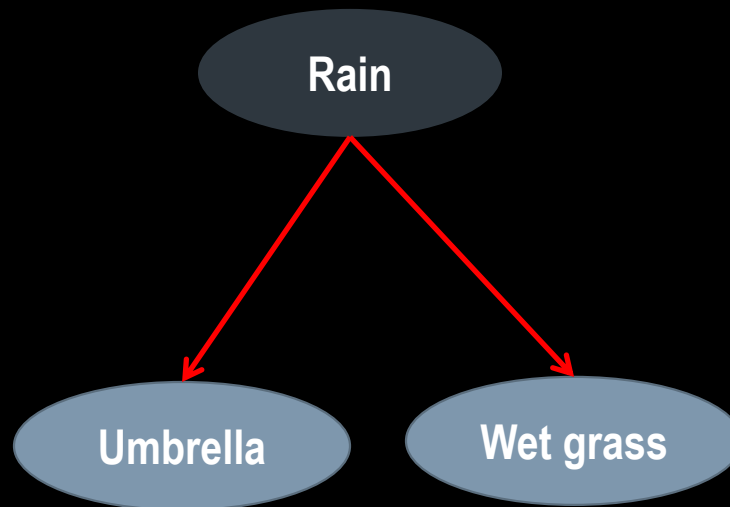
What parameters do we need?

- $P(\text{Rain})$
- $P(\text{Wet} \mid \text{Rain})$
- $P(\text{Wet} \mid \sim\text{Rain})$
- $P(\text{Umbrella} \mid \text{Rain})$
- $P(\text{Umbrella} \mid \sim\text{Rain})$
- How do we compute them?
 - MLE

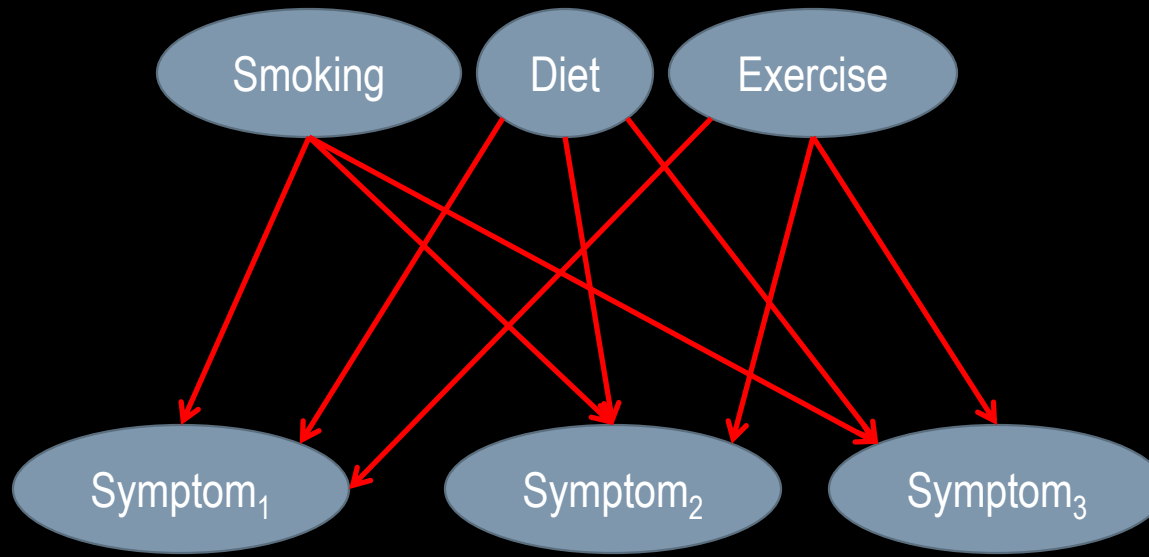


A trickier example

- We know if person has umbrella or grass is wet
- But Rain is *latent* (i.e. hidden)
 - Means we do not know if it rained



Many things are hidden



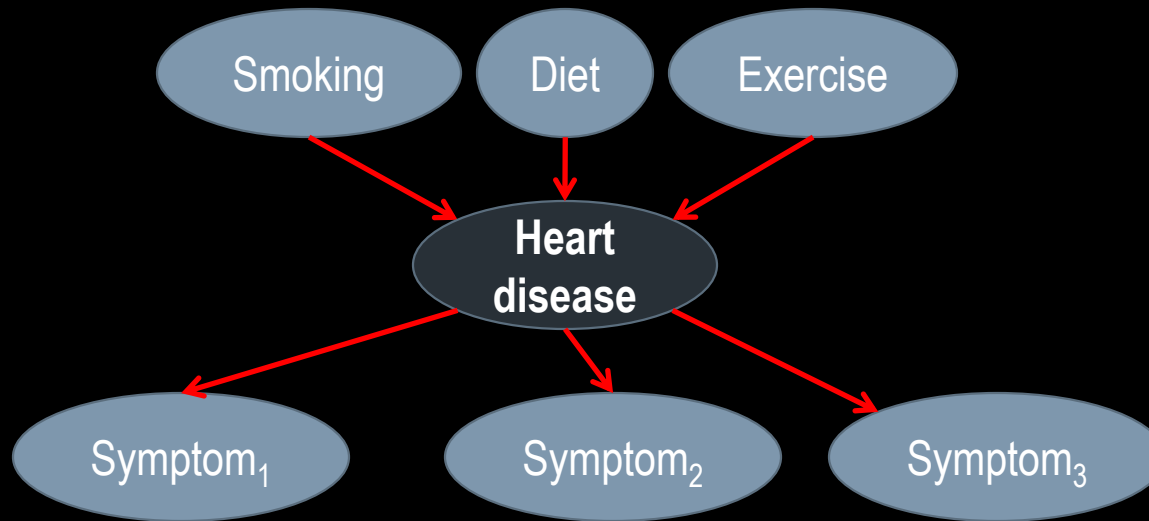
Assuming binary variables

24 parameters in CPTs

Assuming ternary variables

81 parameters in CPTs

Many things are hidden



Assuming binary variables

14 parameters (was 24)

Assuming ternary variables

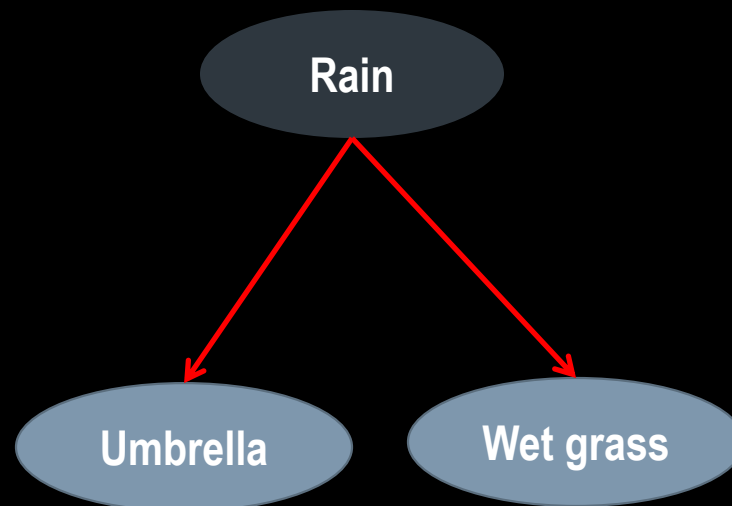
36 parameters (was 81)

Latent variables

- Unobservable (from our model's standpoint)
- Benefits
 - Can make our lives easier
 - Are often how things really work
- Drawbacks
 - Harder to estimate parameters (we don't directly see their values in the data)

A trickier example

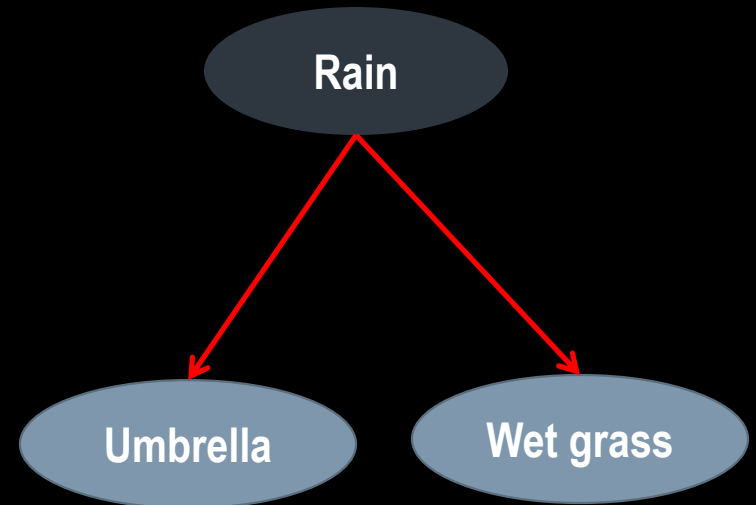
- All nodes are binary, but Rain is *latent*



- **Why can't we use our previous method?**

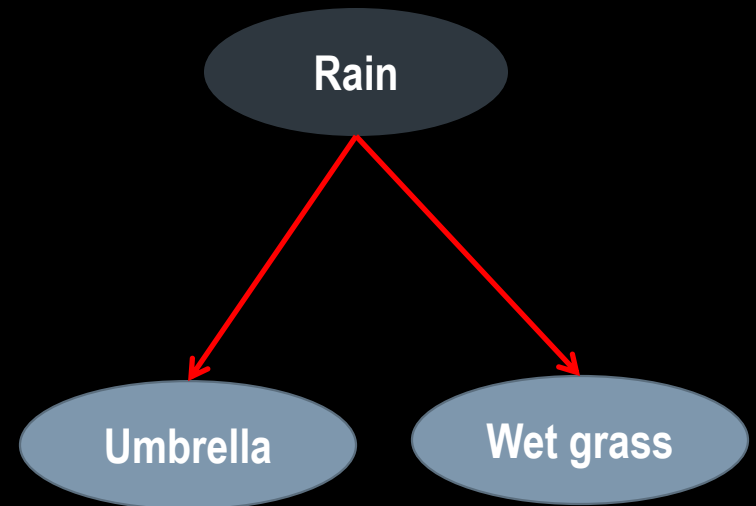
What parameters do we want to estimate?

- $P(\text{Wet grass} \mid \text{rain=yes})$
- $P(\text{Wet grass} \mid \text{rain=no})$
- $P(\text{Umbrella} \mid \text{rain=yes})$
- $P(\text{Umbrella} \mid \text{rain=no})$
- $P(\text{Rain})$



Rain is *latent*! Hard to condition on what you cannot see

- $P(\text{Rain})$
- $P(\text{Umbrella} \mid \text{Rain})$
- $P(\text{Umbrella} \mid \sim \text{Rain})$
- $P(\text{Wet} \mid \text{Rain})$
- $P(\text{Wet} \mid \sim \text{Rain})$



A trick

- We know the counts of each case relative to Wet and Umbrella
- Can think of task as *partitioning* the cases into whether they occurred when Rain=true vs. Rain=false
- If we could only get the right partition, we'd be done
- So, let's guess a partitioning, and improve our guess iteratively
 - This is essentially the **Expectation Maximization (EM)** algorithm

Intuition of the EM algorithm

1. Start with a guess
2. Partition the observations based on what latent they “came from” (**Expectation**)
3. Recompute your model of the world with MLE (**Maximization**)
4. Repeat steps 2 and 3 until convergence

Sketch of EM algorithm

1. Create an initial model, $\theta^{(0)}$.
 - Arbitrarily, randomly, or with a small set of training examples.
2. Use the model $\theta^{(i)}$ to obtain another model $\theta^{(i+1)}$ such that

$$\sum_j \log P_{\theta^{(i+1)}}(x_j) > \sum_j \log P_{\theta^{(i)}}(x_j)$$

3. Repeat the above step until reaching a local maximum
 - Guaranteed to find a better model after each iteration until max
- Beware: Where you end depends on where you start
 - You only get a *Local* maximum

General form of the EM algorithm

- \mathbf{x} : the observed values at all the examples
- \mathbf{Z} : the hidden variables
- θ : the parameters of the probability model
- At each iteration i , update θ using this equation:

$$\theta^{i+1} = \underset{\theta}{\operatorname{argmax}} \sum_z P(Z = z | x, \theta^{(i)}) L(x, Z = z | \theta)$$

Diagram annotations:

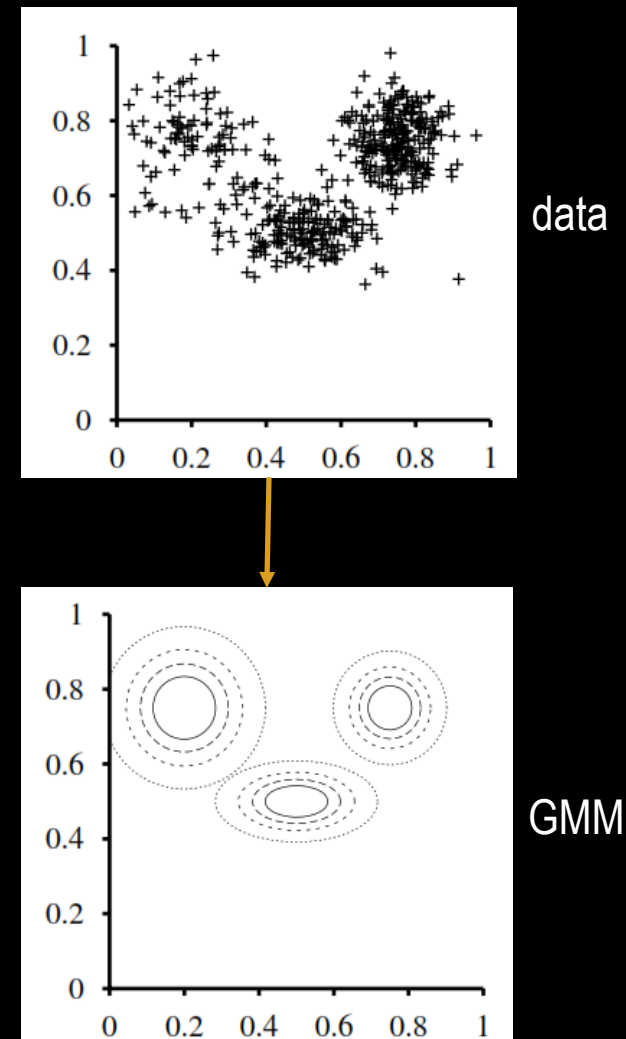
- A bracket under θ is labeled "Maximization".
- A bracket under the entire sum $\sum_z P(Z = z | x, \theta^{(i)}) L(x, Z = z | \theta)$ is labeled "Expectation".
- An arrow points from the text "Log likelihood" to the term $L(x, Z = z | \theta)$.

Break

Gaussian Mixture Models (GMMs)

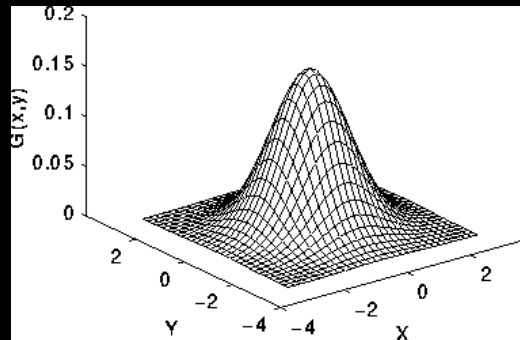
Example: Learning a Gaussian Mixture Model (GMM)

- A GMM is simply a set of Gaussians
 - The set jointly defines a probability distribution over the space
- GMMs are very popular in robotics
 - Example later
- Lets say we want to model a set of data points with a GMM
- Problem: we don't know which data point came from which Gaussian
 - i.e. we don't know how to partition the data



GMM model

- Each component C is a Gaussian with mean μ , variance Σ , and weight w



- The probability at a point x for a GMM with k Gaussians is:

$$P(x) = \sum_{i=1}^k P(C = i)P(x|C = i)$$

The weight parameter for the i th Gaussian w_i


Probability of x in the
 i th Gaussian, given μ_i and Σ_i

EM for a GMM

- **Expectation:** Compute $p_{ij} = P(C = i|x_j)$
 - (the probability x_j was generated from the i th Gaussian)

$$p_{ij} = P(C = i|x_j) = \eta P(x_j|C = i)P(C = i)$$

Probability of x_j in the i th Gaussian,
given current μ_i and Σ_i



The current weight
parameter for the i th
Gaussian w_i

- Define $n_i = \sum_j p_{ij}$ as how well the i th Gaussian accounts for the datapoints

EM for a GMM

- **Maximization:** Compute the following steps in sequence

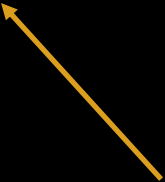
$$\mu_i \leftarrow \sum_j p_{ij} x_j / n_i$$

$$\Sigma_i \leftarrow \sum_j p_{ij} (x_j - \mu_i)(x_j - \mu_i)^T / n_i$$

$$w_i \leftarrow n_i / N$$

- Iterate EM until convergence

Total number of data points

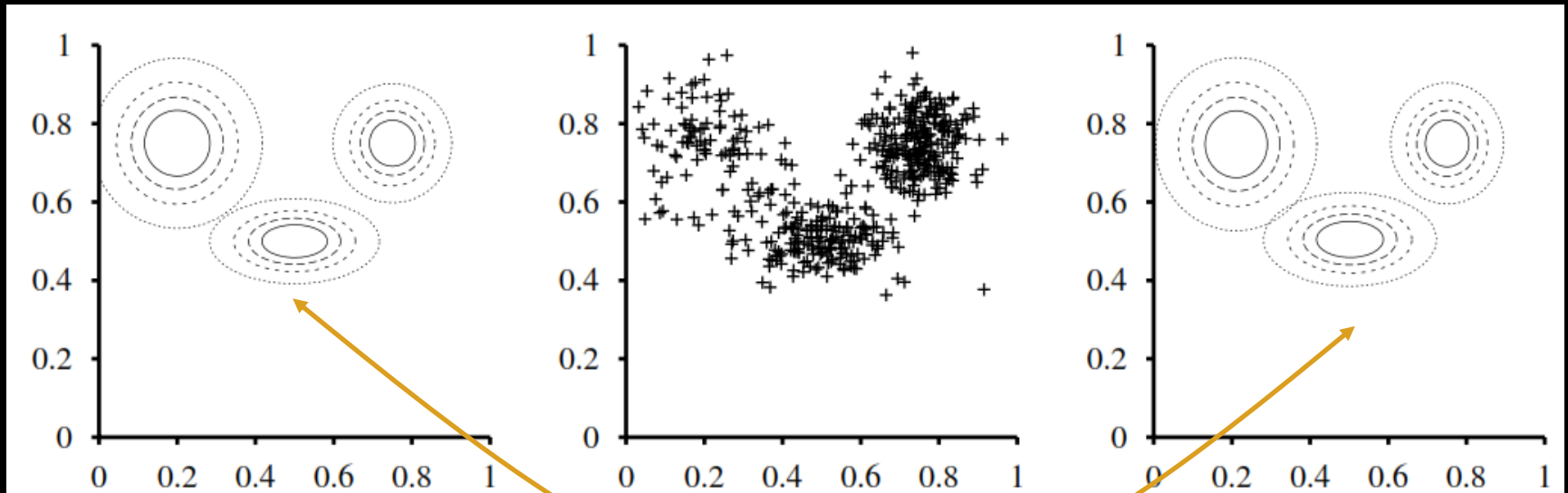


EM for GMMs results

Original GMM

Data generated by sampling
from original GMM

GMM recovered
from data using EM



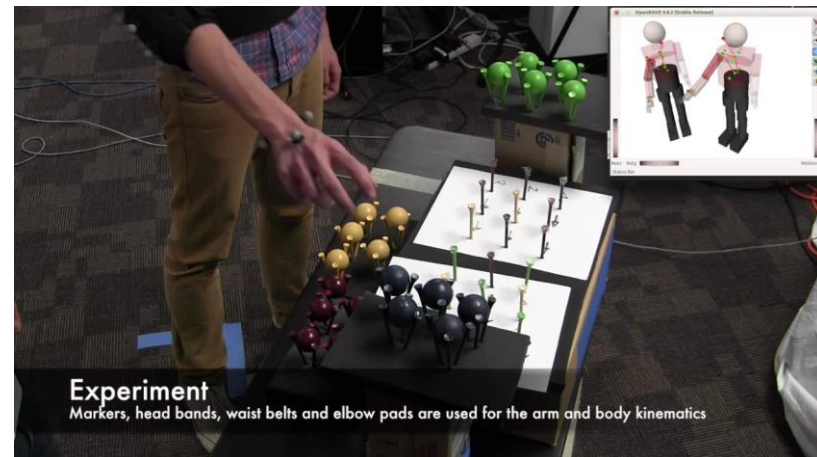
Almost identical!

Going Further: GMMs for Human-robot Collaboration



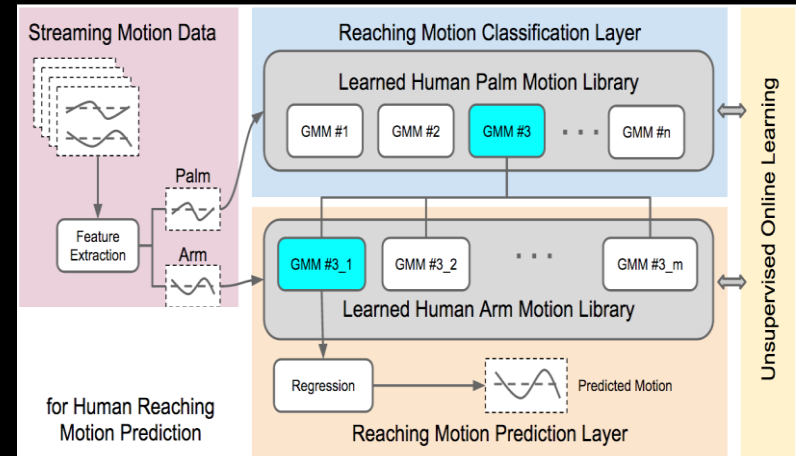
Ruikun Luo
PhD Student

- Problem Statement:
 - Recognition and early prediction
 - Observed beginning part, predict remainder
 - Human reaching motion
 - Industrial manipulation tasks
- Main Contributions:
 - Recognition and prediction work best with different features, necessitating a two-layer framework
 - Proposed an Unsupervised Online Learning Algorithm that outperforms supervised methods



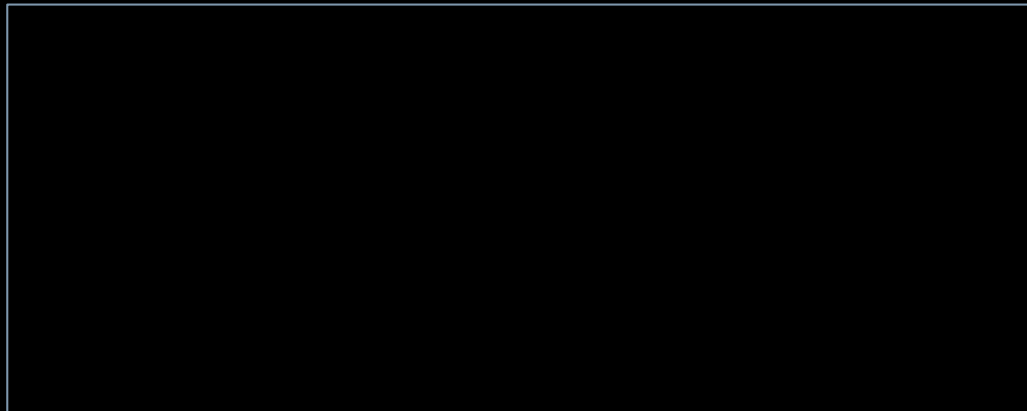
Unsupervised Online Learning for Early Prediction

- Two-layer GMM library for recognition and prediction of human reaching
- Incorporate new data through incremental GMM updates
- Observes portion of trajectory, predicts remainder



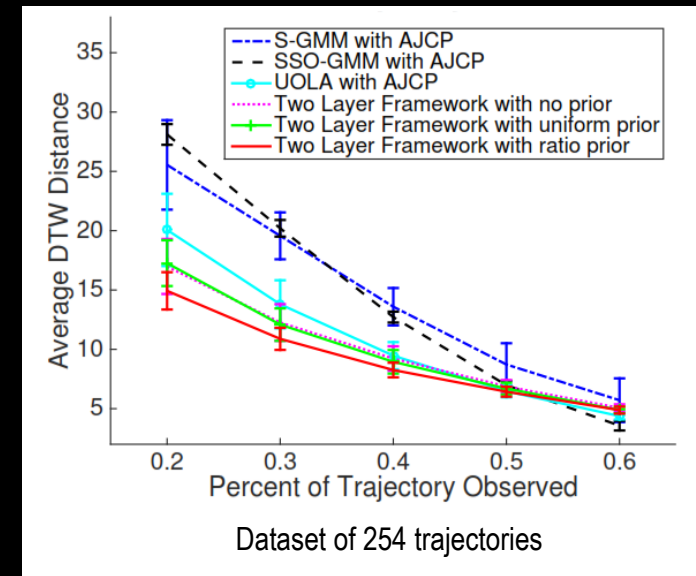
Supervised GMM

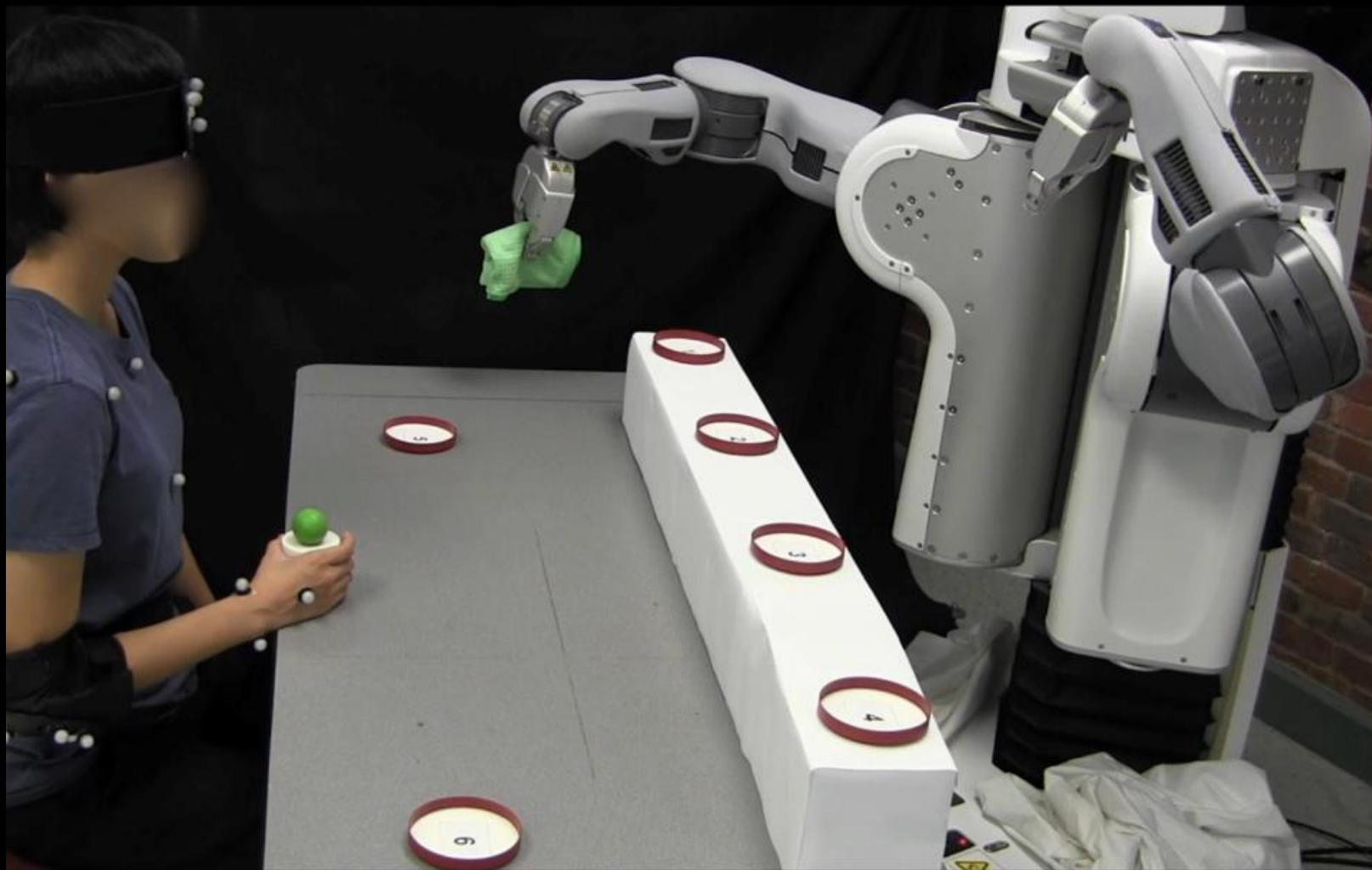
Our Method

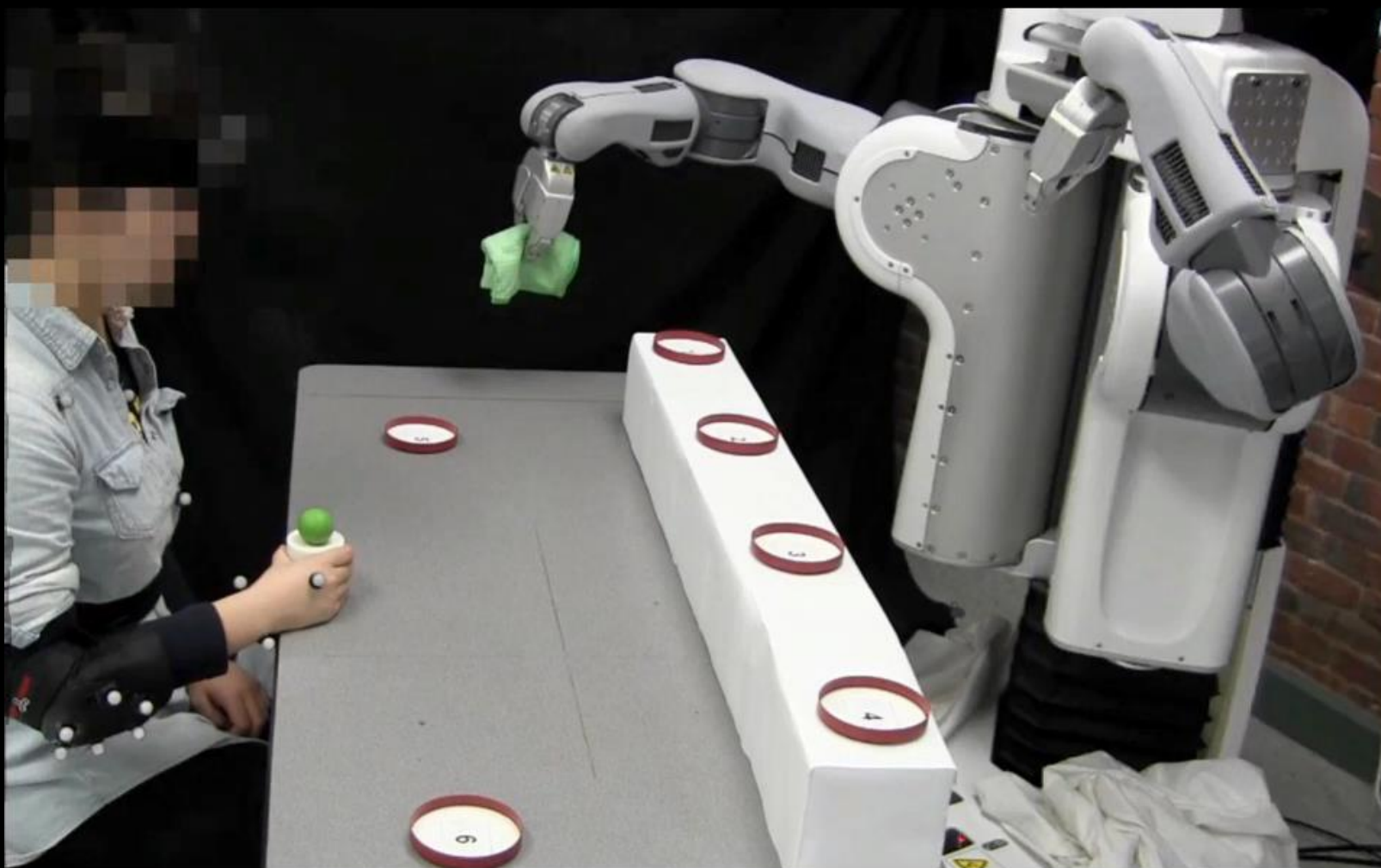


Blue: Recorded trajectory

Magenta: Predicted trajectory







Initial prediction is correct (0.1s)

Summary

- Statistical learning goal: Compute $\operatorname{argmax}_h P(d|h) * P(h)$
- MLE and MAP allow us to define the probability of a variable from data
 - when all variables are observable
- Use log-likelihood for mathematical/computation convenience
- EM allows us to estimate the probability of hidden variables from data
 - a fundamental algorithm in machine learning
- Fitting GMMs to data is an application of EM which is useful for robotics

Homework

- Read AI book Ch. 15.1-15.3
- I'm away next week, watch recorded lectures online (will post on Piazza)