COL333/671: Introduction to AI

Semester I, 2024-25

Learning with Probabilities

Rohan Paul

Outline

- Last Class
  - CSPs
- This Class
  - Bayesian Learning, MLE/MAP, Learning in Probabilistic Models.
- Reference Material
  - Please follow the notes as the primary reference on this topic. Supplementary reading on topics covered in class from AIMA Ch 20 sections 20.1 – 20.2.4.

Acknowledgement

These slides are intended for teaching purposes only. Some material has been used/adapted from web sources and from slides by Doina Precup, Dorsa Sadigh, Percy Liang, Mausam, Parag, Emma Brunskill, Alexander Amini, Dan Klein, Anca Dragan, Nicholas Roy and others.

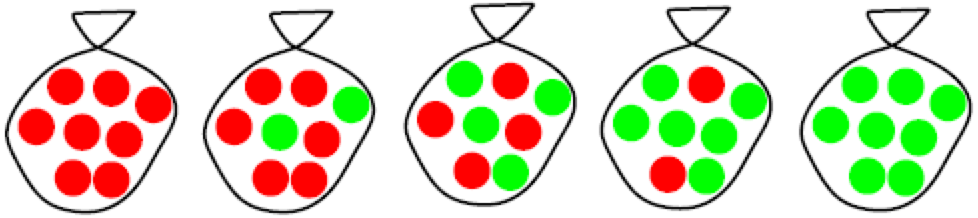
Learning Probabilistic Models

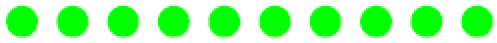
- Models are useful for making optimal decisions.
  - Probabilistic models express a theory about the domain and can be used for decision making.
- How to acquire these models in the first place?
  - Solution: data or experience can be used to build these models
- Key question: how to learn from data?
  - Bayesian view of learning (learning task itself is probabilistic inference)
  - Learning with complete and incomplete data.
  - Essentially, rely on counting.

Example: *Which candy bag is it?*

Suppose there are five kinds of bags of candies:

10% are h_1: 100% cherry candies
20% are h_2: 75% cherry candies + 25% lime candies
40% are h_3: 50% cherry candies + 50% lime candies
20% are h_4: 25% cherry candies + 75% lime candies
10% are h_5: 100% lime candies



Then we observe candies drawn from some bag: 

What kind of bag is it? What flavour will the next candy be?

Statistics Probability

Bayesian Learning – in a nutshell

View learning as Bayesian updating of a probability distribution over the hypothesis space

H is the hypothesis variable, values $h_1, h_2, \ldots$, prior $\mathbf{P}(H)$

ith observation x_i gives the outcome of random variable X_i
training data $\mathbf{X} = x_1, \ldots, x_N$

Given the data so far, each hypothesis has a posterior probability:

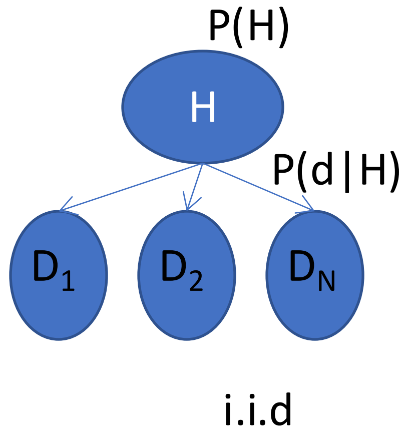
$$P(h_k|\mathbf{X}) = \alpha P(\mathbf{X}|h_k)P(h_k)$$

where $P(\mathbf{X}|h_k)$ is called the likelihood

Predictions use a likelihood-weighted average over the hypotheses:

$$\mathbf{P}(X_{N+1}|\mathbf{X}) = \Sigma_k \, \mathbf{P}(X_{N+1}|\mathbf{X}, h_k)P(h_k|\mathbf{X}) = \Sigma_k \, \mathbf{P}(X_{N+1}|h_k)P(h_k|\mathbf{X})$$

No need to pick one best-guess hypothesis!



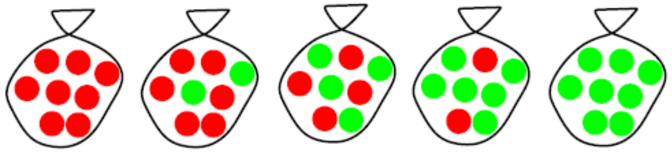
In these slides X and d used interchangeably.

Posterior Probability of Hypothesis given Ovservations

Now, we are getting observations incrementally, how does our belief change?

Suppose there are five kinds of bags of candies:

- 10% are h_1: 100% cherry candies
- 20% are h_2: 75% cherry candies + 25% lime candies
- 40% are h_3: 50% cherry candies + 50% lime candies
- 20% are h_4: 25% cherry candies + 75% lime candies
- 10% are h_5: 100% lime candies



Then we observe candies drawn from some bag: 

What kind of bag is it? What flavour will the next candy be?

Probability of a bag of a certain type given observations.

$$P(h_i \mid d_1, \ldots, d_N)$$

Bayes Rule

$$P(h_i|\mathbf{d}) = \alpha P(\mathbf{d}\,|\,h_i)P(h_i)$$

IID assumption

$$P(d\,|\,h_i) = \prod_j P(d_j\,|\,h_i)$$

Posterior Probability of Hypothesis given Ovservations

$$P(h_k|\mathbf{X}) = \alpha P(\mathbf{X}|h_k)P(h_k)$$

$$P(h_1 \mid \text{5 limes}) = \alpha P(\text{5 limes} \mid h_1)P(h_1) = \alpha \cdot 0.0^5 \cdot 0.1 = 0$$
$$P(h_2 \mid \text{5 limes}) = \alpha P(\text{5 limes} \mid h_2)P(h_2) = \alpha \cdot 0.25^5 \cdot 0.2 = 0.000195\alpha$$
$$P(h_3 \mid \text{5 limes}) = \alpha P(\text{5 limes} \mid h_3)P(h_3) = \alpha \cdot 0.5^5 \cdot 0.4 = 0.0125\alpha$$
$$P(h_4 \mid \text{5 limes}) = \alpha P(\text{5 limes} \mid h_4)P(h_4) = \alpha \cdot 0.75^5 \cdot 0.2 = 0.0475\alpha$$
$$P(h_5 \mid \text{5 limes}) = \alpha P(\text{5 limes} \mid h_5)P(h_5) = \alpha \cdot 1.0^5 \cdot 0.1 = 0.1\alpha$$

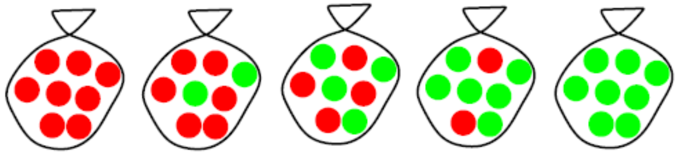
$$\alpha = 1/(0 + 0.000195 + 0.0125 + 0.0475 + 0.1) = 6.2424$$

$$P(h_1 \mid \text{5 limes}) = 0$$
$$P(h_2 \mid \text{5 limes}) = 0.00122$$
$$P(h_3 \mid \text{5 limes}) = 0.07803$$
$$P(h_4 \mid \text{5 limes}) = 0.29650$$
$$P(h_5 \mid \text{5 limes}) = 0.62424$$

Incremental Belief Update

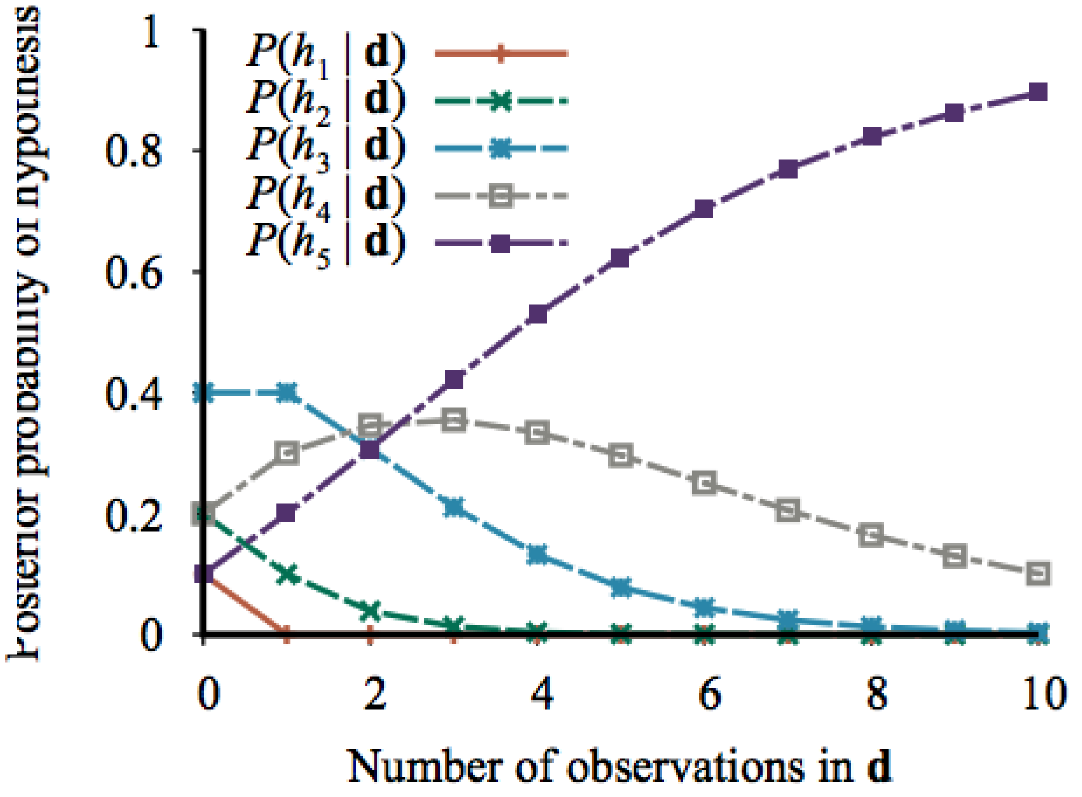
Suppose there are five kinds of bags of candies:

10% are h_1: 100% cherry candies
20% are h_2: 75% cherry candies + 25% lime candies
40% are h_3: 50% cherry candies + 50% lime candies
20% are h_4: 25% cherry candies + 75% lime candies
10% are h_5: 100% lime candies



Then we observe candies drawn from some bag: 

What kind of bag is it? What flavour will the next candy be?



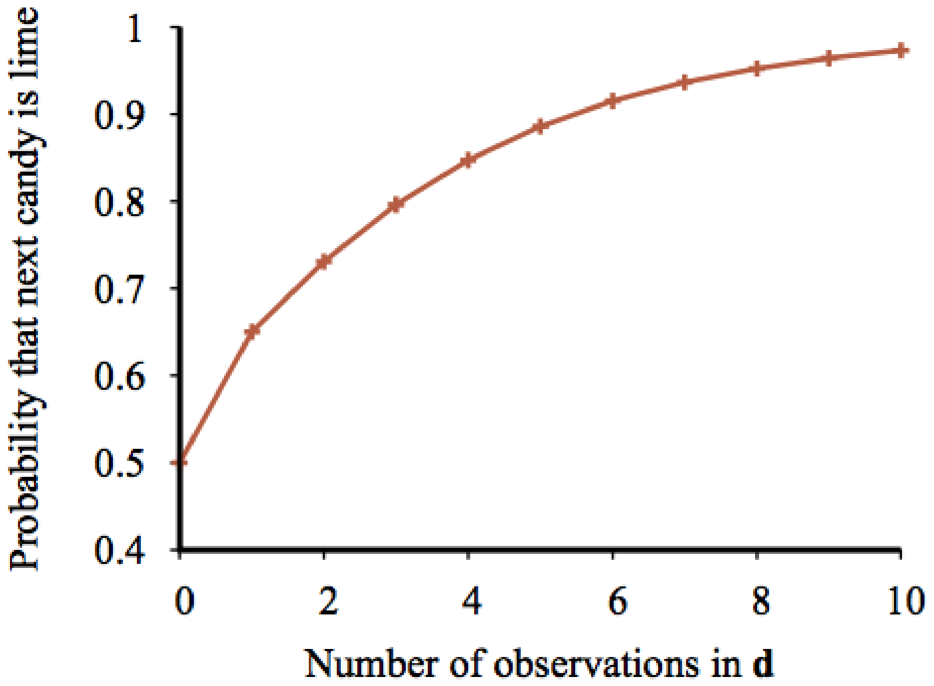
True hypothesis eventually dominates. Probability of indefinitely producing uncharacteristic data →0

Predictions given Belief over Hypotheses

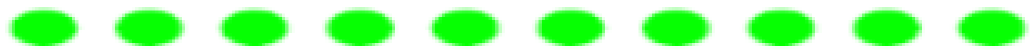
What is the probability that the next candy is of type lime?

$$\mathbf{P}(X_{N+1}|\mathbf{X}) = \Sigma_k\, \mathbf{P}(X_{N+1}|\mathbf{X}, h_k)P(h_k|\mathbf{X}) = \Sigma_k\, \mathbf{P}(X_{N+1}|h_k)P(h_k|\mathbf{X})$$

$$\begin{aligned}
P(\text{lime on 6} \mid \text{5 limes}) &= P(\text{lime on 6} \mid h_1)P(h_1 \mid \text{5 limes}) \\
&+ P(\text{lime on 6} \mid h_2)P(h_2 \mid \text{5 limes}) \\
&+ P(\text{lime on 6} \mid h_3)P(h_3 \mid \text{5 limes}) \\
&+ P(\text{lime on 6} \mid h_4)P(h_4 \mid \text{5 limes}) \\
&+ P(\text{lime on 6} \mid h_5)P(h_5 \mid \text{5 limes}) \\
&= 0 \times 0 \\
&+ 0.25 \times 0.00122 \\
&+ 0.5 \times 0.07830 \\
&+ 0.75 \times 0.29650 \\
&+ 1.0 \times 0.62424 \\
&= 0.88607
\end{aligned}$$



Observations

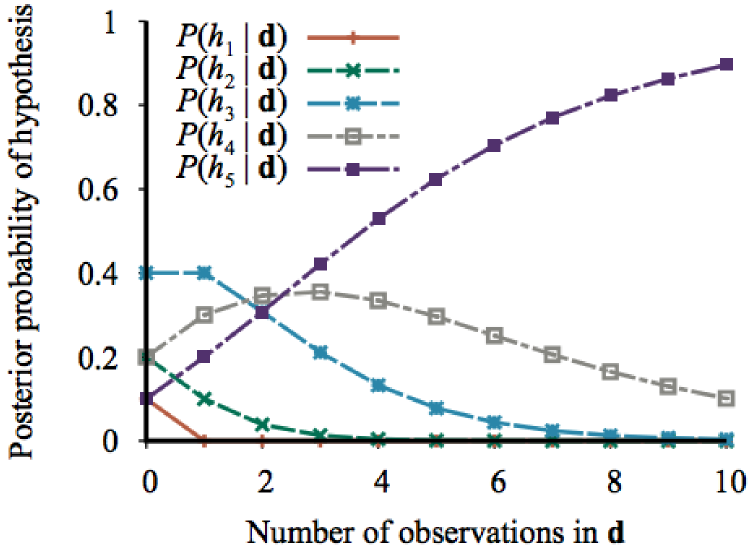


Bayesian Prediction – key ideas

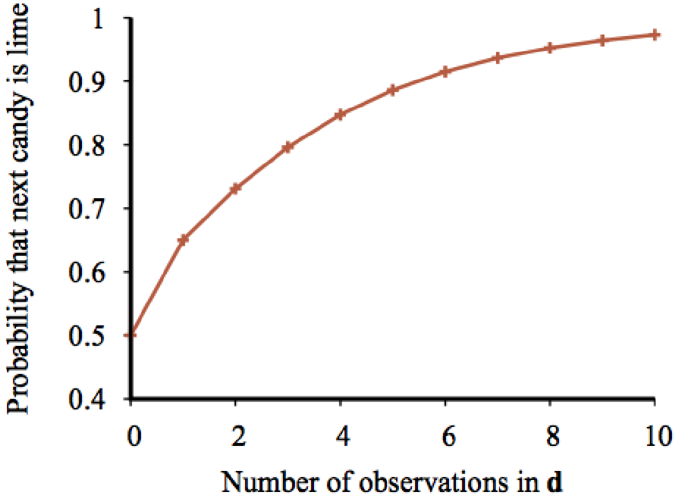
- Predictions are weighted average over the predictions of the individual hypothesis.
- Bayesian prediction eventually agrees with the true hypothesis.
- For any fixed prior that does not rule out the true hypothesis, the posterior probability of any false hypothesis will eventually vanish.
- Why keep all the hypothesis?
  - Learning from small data, early commitment to a hypothesis is risky, later evidence may lead to a different likely hypothesis.
  - Better accounting of uncertainty in making predictions.
  - Problem: maybe slow and intractable, cannot estimate and marginalize out the hypotheses.

Evidence arrives incrementally

Changing belief



Prediction by model averaging.



Marginalization over Hypothesis – *challenging!*

Ideally, one needs to marginalize or account for all the hypotheses.

Can we pick one good hypothesis and just use that for predications?

$$\mathbf{P}(X_{N+1}|\mathbf{X}) = \Sigma_k\, \mathbf{P}(X_{N+1}|\mathbf{X}, h_k)P(h_k|\mathbf{X}) = \Sigma_k\, \mathbf{P}(X_{N+1}|h_k)P(h_k|\mathbf{X})$$

$$\begin{aligned} P(\text{lime on 6} \mid \text{5 limes}) &= P(\text{lime on 6} \mid h_1)P(h_1 \mid \text{5 limes}) \\ &+ P(\text{lime on 6} \mid h_2)P(h_2 \mid \text{5 limes}) \\ &+ P(\text{lime on 6} \mid h_3)P(h_3 \mid \text{5 limes}) \\ &+ P(\text{lime on 6} \mid h_4)P(h_4 \mid \text{5 limes}) \\ &+ P(\text{lime on 6} \mid h_5)P(h_5 \mid \text{5 limes}) \end{aligned}$$

Maximum a-posteriori (MAP) Approximation

Make predictions based on a single most probable hypothesis

$$P(X \mid d) \approx P(X \mid h_{MAP})$$

What is the probability of a hypothesis given data?

$$P(h_{MAP}) = \underset{h_i}{\operatorname{argmax}}(P(h_i \mid d))$$

$$P(h_{MAP}) = \underset{h_i}{\operatorname{argmax}}(P(d \mid h_i)P(h_i))$$

$$= \underset{h_i}{\operatorname{argmax}}(\log(P(d \mid h_i)) + \log(P(h_i)))$$

- MAP learning chooses the hypothesis that provides maximum *compression* of the data.
  - $\log_2 P(h_i)$: the number of bits required to specify the hypothesis h_i.
  - $\log_2 P(\mathbf{d} \mid h_i)$: the additional number of bits required to specify the data, given the hypothesis.

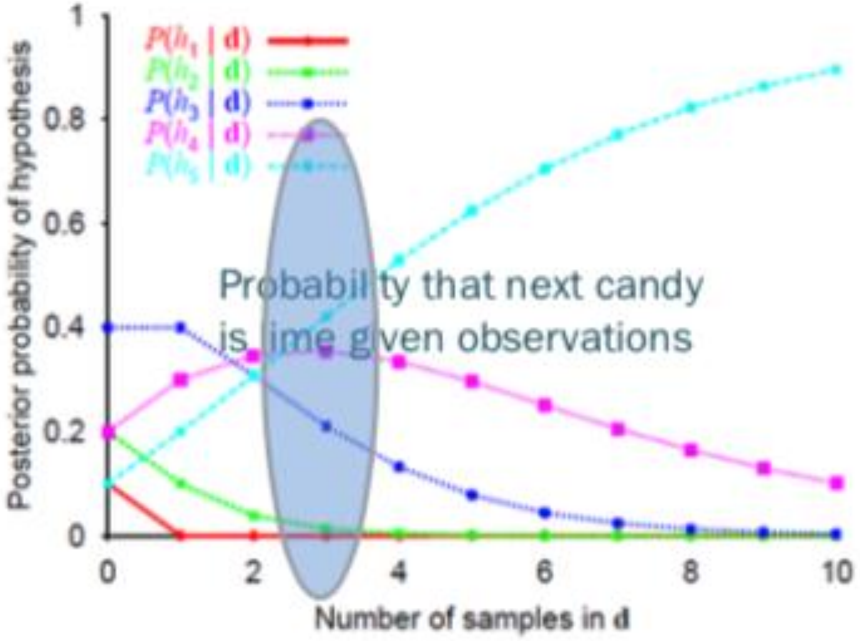
Estimate the best hypothesis given data while incorporating the prior knowledge.

The prior term says which hypothesis are likelier than others. Typically, the number of bits to encode hypothesis.

MAP Vs. Bayesian Estimation

EX> ● ● ● After three observations

MAP predict with probability 1 that next candy is lime
(pick h5)
Bayes will predict with probability 0.8 that net is lime



Difference between marginalization (accounting for all hypothesis) vs. committing to one and make predictions from it.

Maximum Likelihood Estimation

Assume uniform prior over the space of hypothesis

MAP with uniform prior: Maximum-likelihood hypothesis

Becomes irrelevant if uniform

$$P(h_{MAP}) = \underset{h_i}{\operatorname{argmax}}(\log(P(d \mid h_i)) + \log(P(h_i)))$$

$$P(h_{ML}) = \underset{h_i}{\operatorname{argmax}}(\log(P(d \mid h_i)))$$

Make predictions with the hypothesis that maximizes the data likelihood. Essentially, assuming a uniform prior with no preference of a hypothesis over another.

MLE is also called Maximum likelihood (ML) Approximation

Maximum Likelihood Approximation

For large data sets, prior becomes irrelevant

Maximum likelihood (ML) learning: choose h_{ML} maximizing $P(\mathbf{X}|h_k)$

I.e., simply get the best fit to the data; identical to MAP for uniform prior (which is reasonable if all hypotheses are of the same complexity)

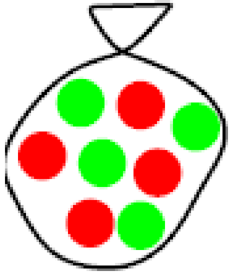
ML is the "standard" (non-Bayesian) statistical learning method

$$\theta_{ML} = \arg\max_{\theta} P(\mathbf{X}|\theta)$$

$$= \arg\max_{\theta} \prod_i P_\theta(X_i)$$

ML Estimation in General: Bernoulli Model

Hypothesis is the likelihood of generating a candy of a specific flavor.



Cherry, Lime, Lime, Cherry, Cherry, Lime, Cherry, Cherry

E.g., Bernoulli[θ] model:

$$P(X_i = 1) = \theta; \;\; P(X_i = 0) = 1 - \theta$$

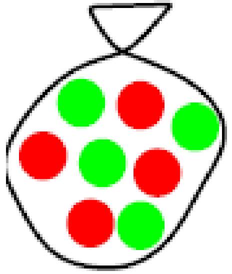
or $P(X_i = x_i) = \theta^{x_i}(1-\theta)^{1-x_i}$

Suppose we get a bag of candy from a new manufacturer; fraction θ of cherry candies

Any θ is possible: continuum of hypotheses h_θ

Similar problem to observing tosses of a biased coin and estimating the bias/fractional parameter.

ML Estimation in General: Estimation for Bernoulli Model



Cherry, Lime, Lime, Cherry, Cherry, Lime, Cherry, Cherry

Suppose we unwrap N candies, c cherries and $\ell = N - c$ limes
These are i.i.d. (independent, identically distributed) observations, so

$$P(\mathbf{X}|h_\theta) = \Pi_{i=1}^{N} P(x_i|h_\theta) = \theta^{\Sigma_i x_i}(1-\theta)^{N-\Sigma_i x_i} = \theta^c \cdot (1-\theta)^\ell$$

Maximize this w.r.t. θ—which is easier for the log-likelihood:

$$L(\mathbf{X}|h_\theta) = \log P(\mathbf{X}|h_\theta) = \Sigma_{i=1}^{N} \log P(x_i|h_\theta) = c\log\theta + \ell\log(1-\theta)$$

$$\frac{dL(\mathbf{X}|h_\theta)}{d\theta} = \frac{c}{\theta} - \frac{\ell}{1-\theta} = 0 \quad \Rightarrow \quad \theta = \frac{c}{c+\ell} = \frac{c}{N}$$

Even in the coin tossing problem, one would take the fraction as heads or tails over the total number of tosses.

MAP vs. MLE Estimation

- **Maximum likelihood estimate (MLE)**
  - Estimates the parameters that maximizes the data likelihood.
  - Relative counts give MLE estimates

$$\theta_{ML} = \arg\max_{\theta} P(\mathbf{X}|\theta)$$

$$= \arg\max_{\theta} \prod_{i} P_{\theta}(X_i)$$

- **Maximum a posteriori estimate (MAP)**
  - Bayesian parameter estimation
  - Encodes a prior over the parameters (not all parameters are equal prior values).
  - Combines the prior and the likelihood while estimating the parameters.

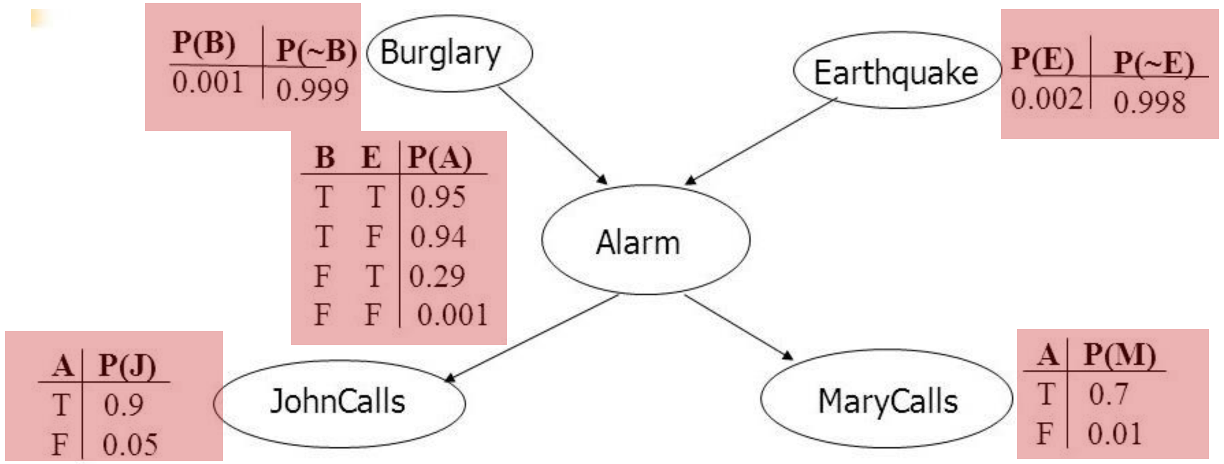
$$\theta_{MAP} = \arg\max_{\theta} P(\theta|\mathbf{X})$$

$$= \arg\max_{\theta} P(\mathbf{X}|\theta)P(\theta)/P(\mathbf{X})$$

$$= \arg\max_{\theta} P(\mathbf{X}|\theta)P(\theta)$$

ML Estimation in General: Learning Parameters for a Probability Model

- Probabilistic models require parameters (numbers in the conditional probability tables).
- We need these values to make predictions.
- Can we <u>learn</u> these from data (i.e., samples from the Bayes Net)?
- How to do this? Counting and averaging.



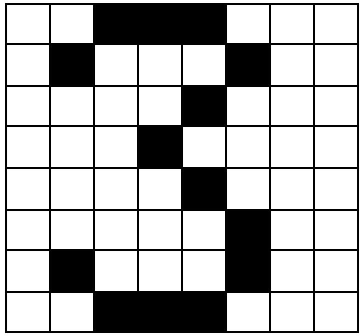
Can we use samples to estimate the values in the tables?

Learning Parameters for a Probability Model

Classification Problem

- Task: given inputs x, predict labels (classes) y
- Examples:
  - Spam detection (input: document, classes: spam / ham)
  - OCR (input: images, classes: characters)
  - Medical diagnosis (input: symptoms, classes: diseases)
  - Fraud detection (input: account activity, classes: fraud / no fraud)

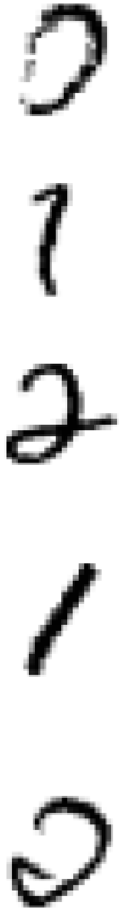
Bayes Net for Classification



- Input: images / pixel grids
- Output: a digit 0-9

- Setup:
  - Get a large collection of example images, each labeled with a digit
  - Note: someone has to hand label all this data!
  - Want to learn to predict labels of new, future digit images

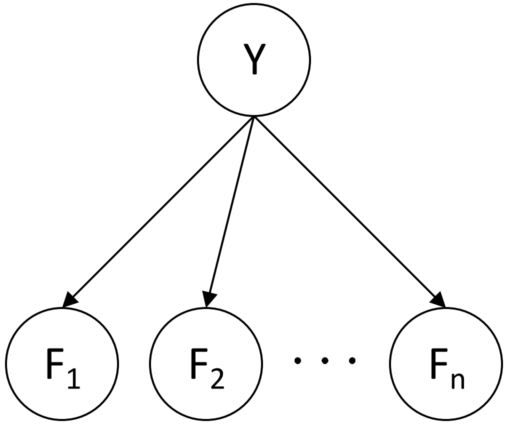
- Features: The attributes used to make the digit decision
  - Pixels: (6,8)=ON
  - Shape Patterns: NumComponents, AspectRatio, NumLoops
  - …

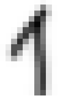


Bayes Net for Classification

- Naïve Bayes: Assume all features are independent effects of the label

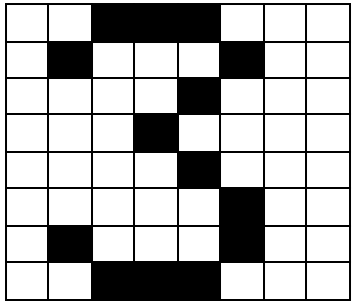
- Simple digit recognition:
  - One feature (variable) F_{ij} for each grid position <i,j>
  - Feature values are on / off, based on whether intensity is more or less than 0.5 in underlying image
  - Each input maps to a feature vector, e.g.





$$\rightarrow \langle F_{0,0} = 0 \;\; F_{0,1} = 0 \;\; F_{0,2} = 1 \;\; F_{0,3} = 1 \;\; F_{0,4} = 0 \;\; \ldots F_{15,15} = 0 \rangle$$

$$P(Y|F_{0,0} \ldots F_{15,15}) \propto P(Y) \prod_{i,j} P(F_{i,j}|Y)$$

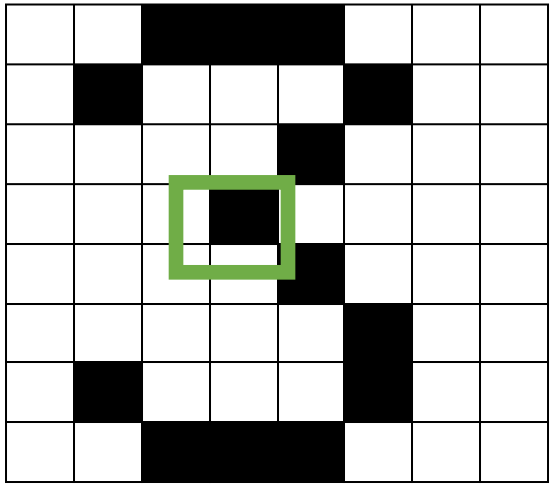


Parameter Estimation

- Need the estimates of local conditional probability tables.
  - $P(Y)$, the prior over labels
  - $P(F_i|Y)$ for each feature (evidence variable)
  - These probabilities are collectively called the ***parameters*** of the model and denoted by $\boldsymbol{\theta}$
    - Till now, the table values were provided.
    - Now, use data to acquire these values.

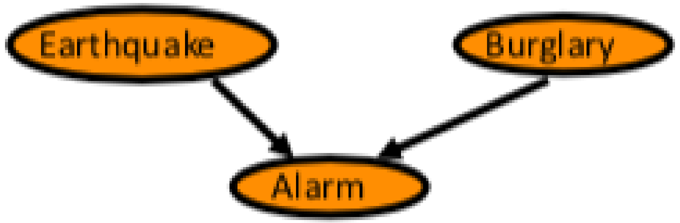
Parameter Estimation

- P(Y) – how frequent is the class-type for digit 3?
  - If you take a sample of images of numbers how frequent is this number
- $P(F_i|Y)$ – for digit 3 what fraction of the time the cell is on?
  - Conditioned on the class type how frequent is the feature
- Use **relative frequencies** from the data to estimate these values.



Parameter Estimation: Complete Data

E	B	A	#
0	0	0	1000
0	0	1	10
0	1	0	20
0	1	1	100
1	0	0	200
1	0	1	50
1	1	0	0
1	1	1	5



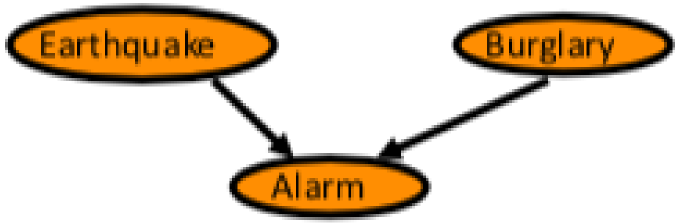
Note: The data is "complete". Each data point had values observed for "all" the variables in the model.

	Pr(A|E,B)
e,b	
$e,\bar{b}$	
$\bar{e},b$	
$\bar{e},\bar{b}$	

$P(a|\bar{e},\bar{b}) = ?$

$= 10/1010$

Parameter Estimation



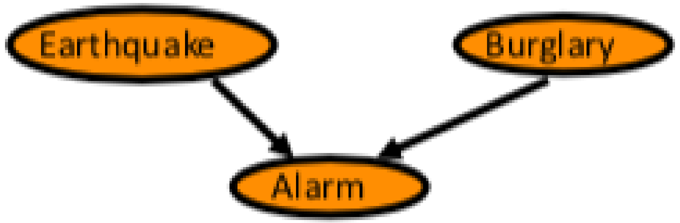
E	B	A	#
0	0	0	1000
0	0	1	10
0	1	0	20
0	1	1	100
1	0	0	200
1	0	1	50
1	1	0	0
1	1	1	5

	$\Pr(A \mid E,B)$
e,b	
$e,\bar{b}$	
$\bar{e},b$	
$\bar{e},\bar{b}$	~0.01

$P(a \mid \bar{e}, b) = ?$

$= 100/120$

Parameter Estimation



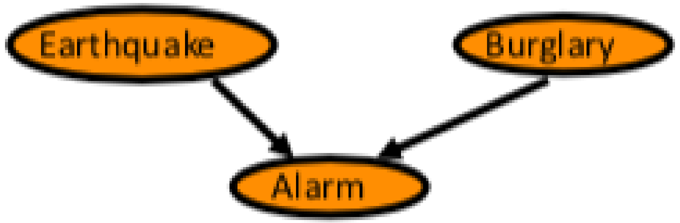
E	B	A	#
0	0	0	1000
0	0	1	10
0	1	0	20
0	1	1	100
1	0	0	200
1	0	1	50
1	1	0	0
1	1	1	5

	Pr(A|E,B)
e,b	
$e,\bar{b}$	
$\bar{e},b$	0.83
$\bar{e},\bar{b}$	~0.01

$P(a|e,\bar{b}) = ?$

$= 50/250$

Parameter Estimation



E	B	A	#
0	0	0	1000
0	0	1	10
0	1	0	20
0	1	1	100
1	0	0	200
1	0	1	50
1	1	0	0
1	1	1	5

	Pr(A|E,B)
e,b	
$e,\bar{b}$	0.2
$\bar{e},b$	0.83
$\bar{e},\bar{b}$	~0.01

$P(a|e, b) = ?$

$= 5/5$

Problem: values not seen in the training data

$P(\text{features}, C = 2)$

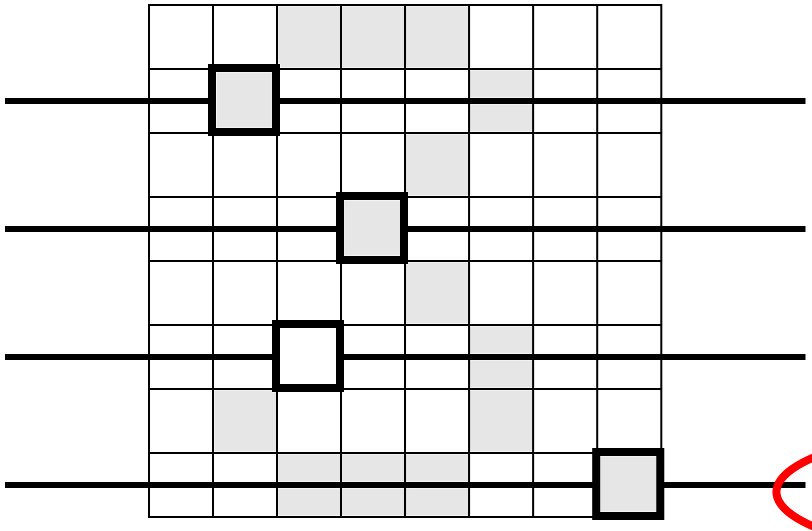
$P(C = 2) = 0.1$

$P(\text{on}|C = 2) = 0.8$

$P(\text{on}|C = 2) = 0.1$

$P(\text{off}|C = 2) = 0.1$

$P(\text{on}|C = 2) = 0.01$



$P(\text{features}, C = 3)$

$P(C = 3) = 0.1$

$P(\text{on}|C = 3) = 0.8$

$P(\text{on}|C = 3) = 0.9$

$P(\text{off}|C = 3) = 0.7$

$P(\text{on}|C = 3) = 0.0$

If one feature was not seen in the training data, the likelihood goes to zero. If we did not see this feature in the training data, does not mean we will not see this in training. Essentially overfitting to the training data set.

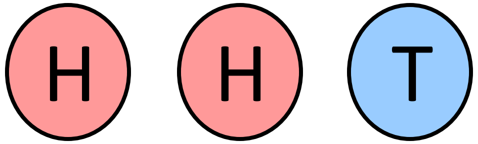
Laplace Smoothing

- Pretend that every outcome occurs **once more** than it is actually observed.

- If certain counts are **not** seen in training does not mean that they have zero probability of occurring in future.

- Another version of Laplace smoothing
  - instead of 1, add **k times**
  - k is an adjustable parameter.

- Essentially, encodes a **prior** (pseudo-counts).



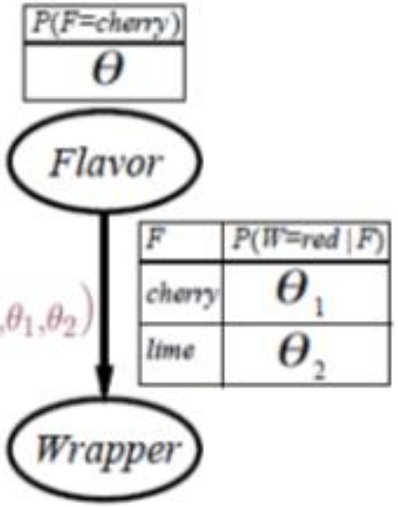
$$P_{LAP}(x) = \frac{c(x)+1}{\sum_x [c(x)+1]} = \frac{c(x)+1}{N+|X|}$$

$$P_{LAP,k}(x) = \frac{c(x)+k}{N+k|X|}$$

Learning Multiple Parameters

- Estimate latent parameters using MLE.
- There are two CPTs in this example.
- Observations are of both variables: Flavor and Wrapper.
- Take log likelihood.

Red/green wrapper depends probabilistically on flavor:



Likelihood for, e.g., cherry candy in green wrapper:

$$
\begin{aligned}
&P(F = cherry, W = green \mid h_{\theta,\theta_1,\theta_2}) \\
&= P(F = cherry \mid h_{\theta,\theta_1,\theta_2}) P(W = green \mid F = cherry, h_{\theta,\theta_1,\theta_2}) \\
&= \theta \cdot (1 - \theta_1)
\end{aligned}
$$

N candies, r_c red-wrapped cherry candies, etc.:

$$P(\mathbf{d} \mid h_{\theta,\theta_1,\theta_2}) = \theta^c (1-\theta)^\ell \cdot \theta_1^{r_c} (1-\theta_1)^{g_c} \cdot \theta_2^{r_\ell} (1-\theta_2)^{g_\ell}$$

N candies unwrapped, c are cherries and *l* are limes

Take logarithm

$$
\begin{aligned}
L = {} & [c \log \theta + \ell \log(1-\theta)] \\
& + [r_c \log \theta_1 + g_c \log(1-\theta_1)] \\
& + [r_\ell \log \theta_2 + g_\ell \log(1-\theta_2)]
\end{aligned}
$$

With complete data, the ML parameter learning problem for a Bayesian network <u>decomposes into separate learning problems</u>, one for each parameter

Learning Multiple Parameters

- Minimize data likelihood to estimate the parameters.

Derivatives of L contain only the relevant parameter:

$$\frac{\partial L}{\partial \theta} = \frac{c}{\theta} - \frac{\ell}{1-\theta} = 0 \qquad \Rightarrow \quad \theta = \frac{c}{c+\ell}$$

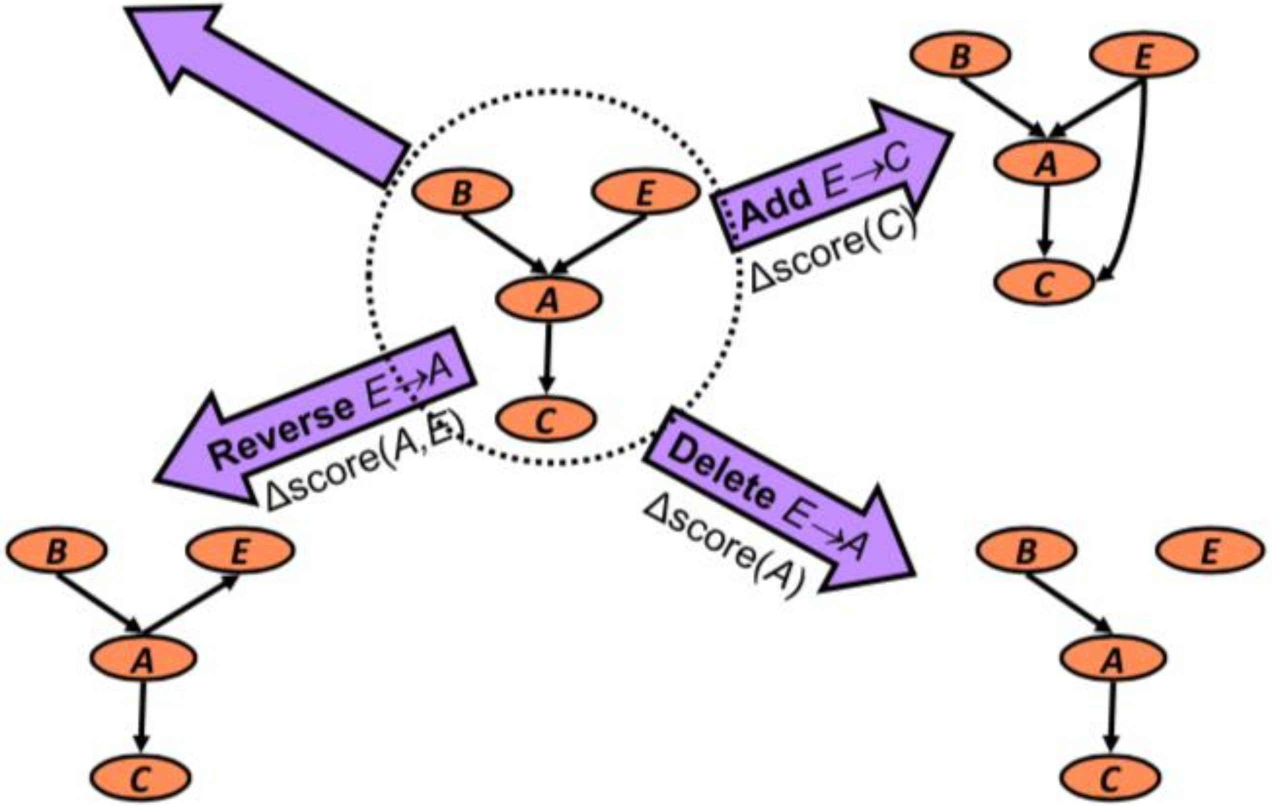
$$\frac{\partial L}{\partial \theta_1} = \frac{r_c}{\theta_1} - \frac{g_c}{1-\theta_1} = 0 \qquad \Rightarrow \quad \theta_1 = \frac{r_c}{r_c+g_c}$$

$$\frac{\partial L}{\partial \theta_2} = \frac{r_\ell}{\theta_2} - \frac{g_\ell}{1-\theta_2} = 0 \qquad \Rightarrow \quad \theta_2 = \frac{r_\ell}{r_\ell+g_\ell}$$

Maximum Likelihood Parameter Learning with complete data for a Bayes Net decomposes into separate learning problems, one for each parameter.

How to learn the structure of the Bayes Net?

- Problem: Estimate/learn the structure of the model
- Setup a search process (like local search, hill climbing etc.)
- For each structure, learn the parameters.
- How to score a solution?
  - Use Max. likelihood estimation.
  - Penalize complexity of the structure (don't want a fully connected model).
  - Additionally check for validity of the conditional independences.



Parameter Learning when some variables are not observed

- If we knew the missing value for B. Then we can estimate the CPTs.

- If we knew the CPTs then we can infer the probability of the missing value of B.

- It is a *chicken and egg* problem.

Examples:

A	B	C
0	1	1
1	0	0
1	1	1
1	?	0

Initialization:

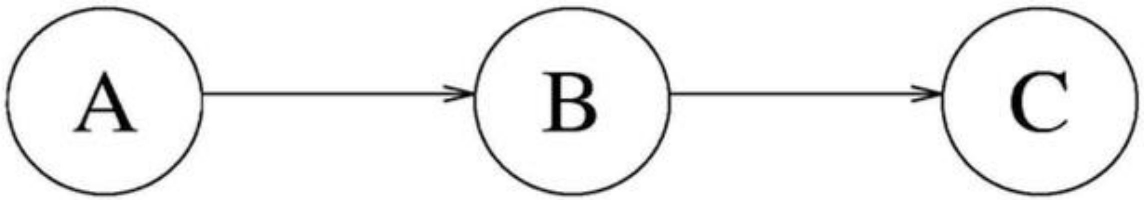
$P(A) =$ $\quad P(B|A) =$ $\quad P(B|\neg A) =$ $\quad P(C|B) =$ $\quad P(C|\neg B) =$

Data is incomplete. One sample has (A = 1, **B= ?** and C = 0)

Expectation Maximization

- Initialization
  - Initialize CPT parameter values (ignoring missing information)
- Expectation
  - Compute expected values of unobserved variables assuming current parameters values.
  - Involves BayesNet inference (exact or approximate)
- Maximization
  - Compute new parameters (of the CPTs) to maximize the probability of data (observed and estimated)
- Alternate the EM steps until convergence. Convergence is guaranteed.

Expectation Maximization



Examples:

0	1	1
1	0	0
1	1	1
1	?	0

Initialization: $P(B|A) = 0$ $P(C|B) = 0$

$P(A) = 0.75$ $P(B|\neg A) = 0$ $P(C|\neg B) = 0$

E-step: $P(? = 1) = P(B|A, \neg C) = \frac{P(A,B,\neg C)}{P(A,\neg C)} = \ldots = 0$

M-step: $P(B|A) =$ $P(C|B) =$

$P(A) =$ $P(B|\neg A) =$ $P(C|\neg B) =$

E-step: $P(? = 1) =$

Expectation Maximization

$$A \rightarrow B \rightarrow C$$

Examples:	A	B	C
	0	1	1
	1	0	0
	1	1	1
	1	0	0

Initialization: $P(B|A) = 0$ $\quad P(C|B) = 0$

$P(A) = 0.75$ $\quad P(B|\neg A) = 0$ $\quad P(C|\neg B) = 0$

E-step: $P(? = 1) = P(B|A, \neg C) = \frac{P(A,B,\neg C)}{P(A,\neg C)} = \ldots = 0$

M-step: $P(B|A) = 0.33$ $\quad P(C|B) = 1$

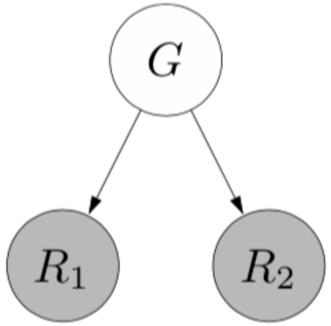
$P(A) = 0.75$ $\quad P(B|\neg A) = 1$ $\quad P(C|\neg B) = 0$

E-step: $P(? = 1) =$

Parameter Learning with Missing Data

Consider a problem of inferring the genre of a movie (Comedy or Drama) from the ratings given by two film reviewers R1 and R2.

Setting where only the ratings are observed.



What if we **don't observe** some of the variables?

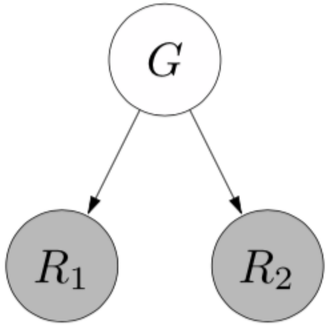
$$\mathcal{D}_{\text{train}} = \{(?, 4, 5), (?, 4, 4), (?, 5, 3), (?, 1, 2), (?, 5, 4)\}$$

Slide adapted from Dorsa Sadigh and Percy Liang

Maximum Marginal Likelihood

Variables: H is hidden, $E = e$ is observed

Example:



$H = G \quad E = (R_1, R_2) \quad e = (1, 2)$

$\theta = (p_G, p_R)$

Marginalize over the latent variables in the likelihood

Maximum marginal likelihood objective:

$$\max_\theta \prod_{e \in \mathcal{D}_{\text{train}}} \mathbb{P}(E = e; \theta)$$

$$= \max_\theta \prod_{e \in \mathcal{D}_{\text{train}}} \sum_h \mathbb{P}(H = h, E = e; \theta)$$

Expectation Maximization

Initialize θ

E-step:

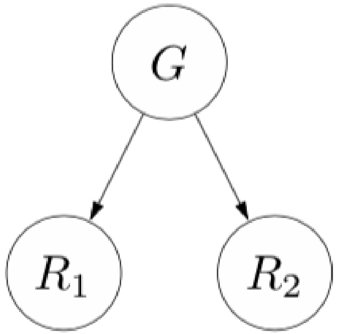
- Compute $q(h) = \mathbb{P}(H = h \mid E = e; \theta)$ for each h (use any probabilistic inference algorithm)
- Create weighted points: (h, e) with weight $q(h)$

M-step:

- Compute maximum likelihood (just count and normalize) to get θ

Repeat until convergence.

Expectation Maximization



$$\mathcal{D}_{\text{train}} = \{(?, 2, 2), (?, 1, 2)\}$$

θ:

g	$p_G(g)$
c	0.5
d	0.5

g	r	$p_R(r \mid g)$
c	1	0.4
c	2	0.6
d	1	0.6
d	2	0.4

E-step

(r_1, r_2)	g	$\mathbb{P}(G = g, R_1 = r_1, R_2 = r_2)$	$q(g)$
(2, 2)	c	$0.5 \cdot 0.6 \cdot 0.6 = 0.18$	$\frac{0.18}{0.18+0.08} = 0.69$
(2, 2)	d	$0.5 \cdot 0.4 \cdot 0.4 = 0.08$	$\frac{0.08}{0.18+0.08} = 0.31$
(1, 2)	c	$0.5 \cdot 0.4 \cdot 0.6 = 0.12$	$\frac{0.12}{0.12+0.12} = 0.5$
(1, 2)	d	$0.5 \cdot 0.6 \cdot 0.4 = 0.12$	$\frac{0.12}{0.12+0.12} = 0.5$

Estimated Fractional samples

(g=c, r_1=2, r_2=2) prob: 0.69
(g=d, r_1=2, r_2=2) prob: 0.31
(g=c, r_1=1, r_2=2) prob: 0.5
(g=d, r_1=1, r_2=2) prob: 0.5

M-step

θ:

g	count	$p_G(g)$
c	0.69 + 0.5	0.59
d	0.31 + 0.5	0.41

g	r	count	$p_R(r \mid g)$
c	1	0.5	0.21
c	2	0.5 + 0.69 + 0.69	0.79
d	1	0.5	0.31
d	2	0.5 + 0.31 + 0.31	0.69

Revising probabilities based on fractional samples.

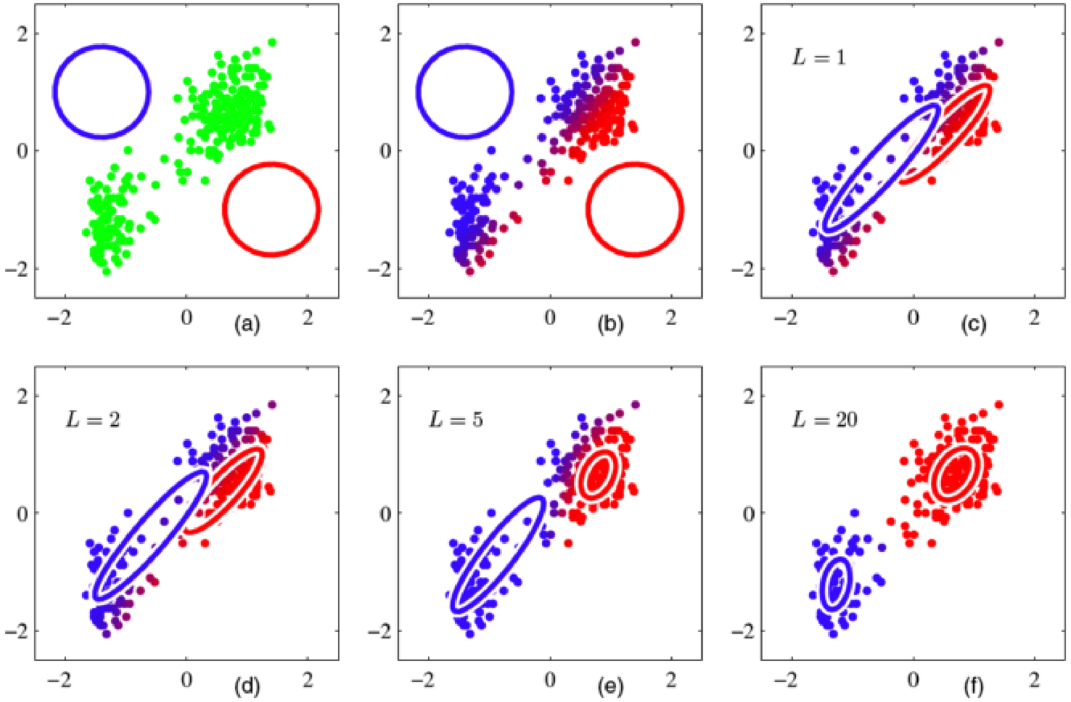
The CPTs for the two reviewers is the same.

Slide adapted from Dorsa Sadigh and Percy Liang

EM in Continuous Space: Gaussian Mixture Modeling

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

- Problem: clustering task where we want to discern multiple category in a collection of given points.
- Assume a mixture of components (Gaussian)
- Don't know which data point comes from which component.
- Use EM to iteratively determine the assignments and the parameters of the Gaussian components.



Web link: https://lukapopijac.github.io/gaussian-mixture-model/

Closely related: K-Means Clustering

***K*-means clustering problem:**

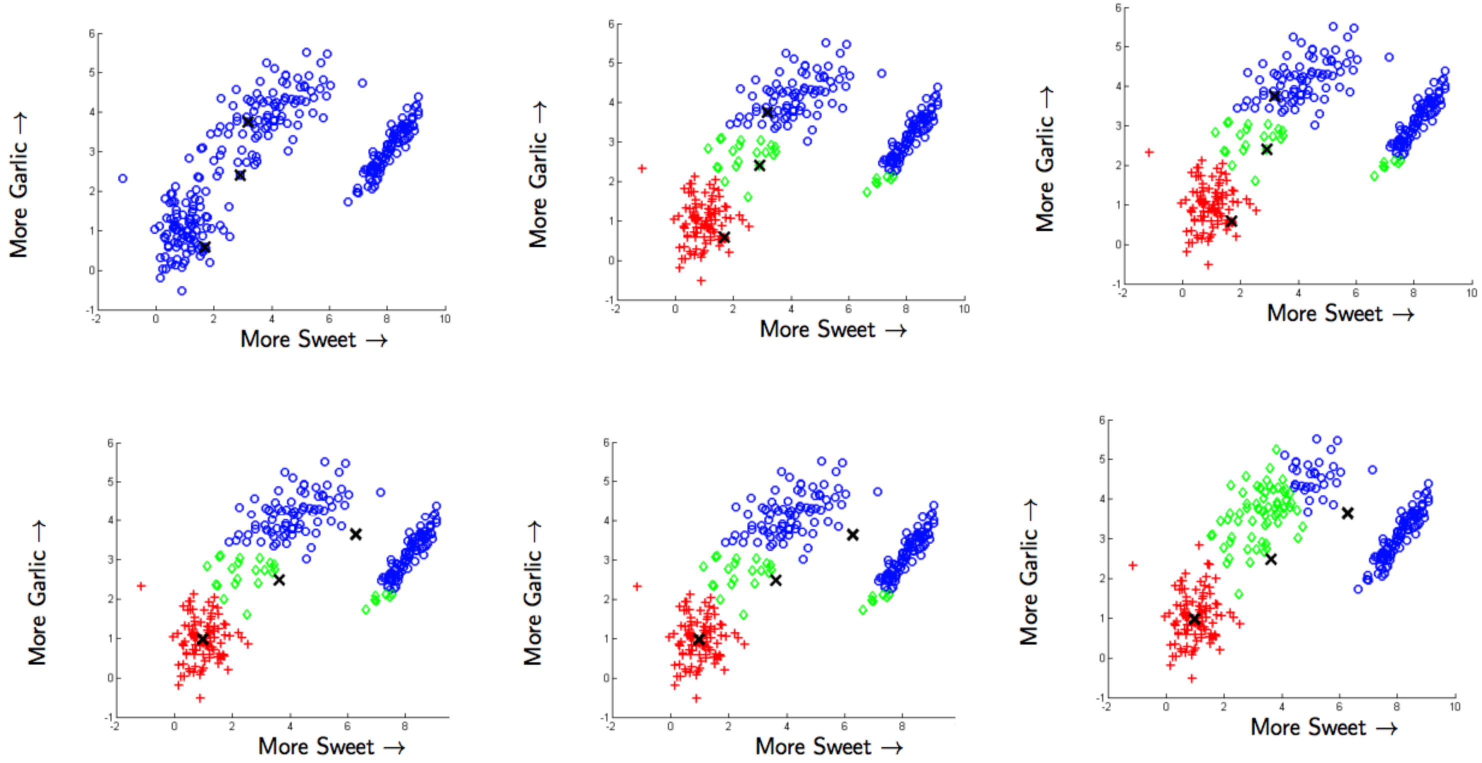
Partition the n observations into K sets ($K \le n$) $\mathbf{S} = \{S_1, S_2, \ldots, S_K\}$ such that the sets minimize the within-cluster sum of squares:

$$\underset{\mathbf{S}}{\arg\min} \sum_{i=1}^{K} \sum_{\mathbf{x}_j \in S_i} \left\| \mathbf{x}_j - \boldsymbol{\mu}_i \right\|^2$$

where μ_i is the mean of points in set S_i.

A GMM yields a probability distribution over the cluster assignment for each point; whereas K-Means gives a single hard assignment

- How many different sauces should the company make?
- How sweet/garlicy should these sauces be?
- Idea: We will segment the consumers into groups (in this case 3), we will then find the best sauce for each group



K-Means: Application

Goal of Segmentation is to partition an image into regions each of which has reasonably homogenous visual appearance.

Apply K-Means in the colour space.