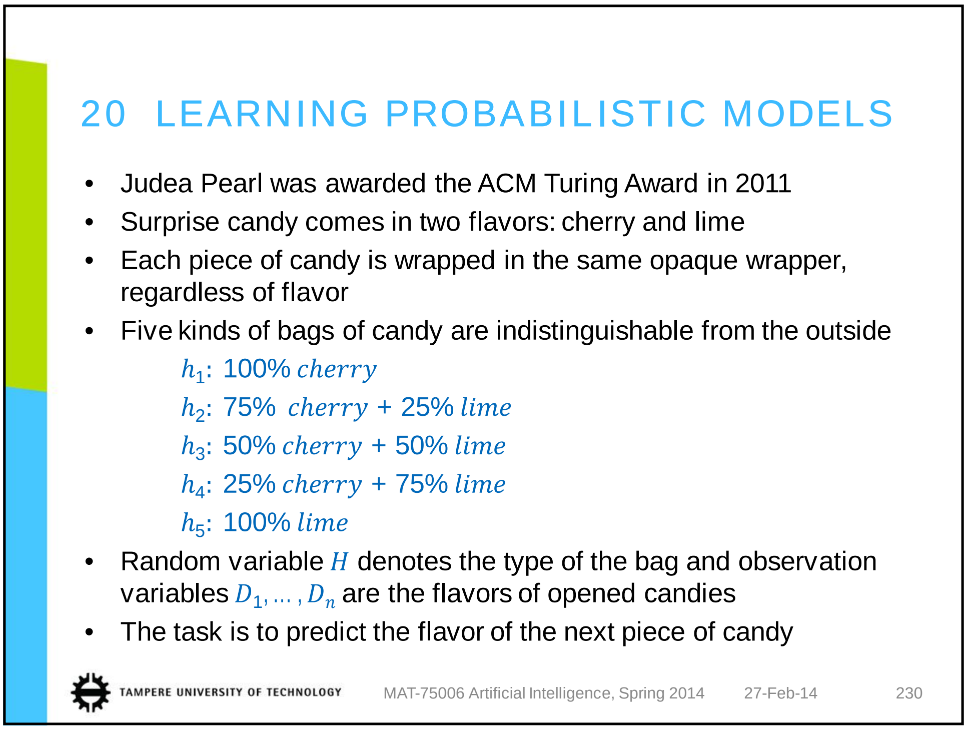


20 LEARNING PROBABILISTIC MODELS

- Judea Pearl was awarded the ACM Turing Award in 2011
- Surprise candy comes in two flavors: cherry and lime
- Each piece of candy is wrapped in the same opaque wrapper, regardless of flavor
- Five kinds of bags of candy are indistinguishable from the outside

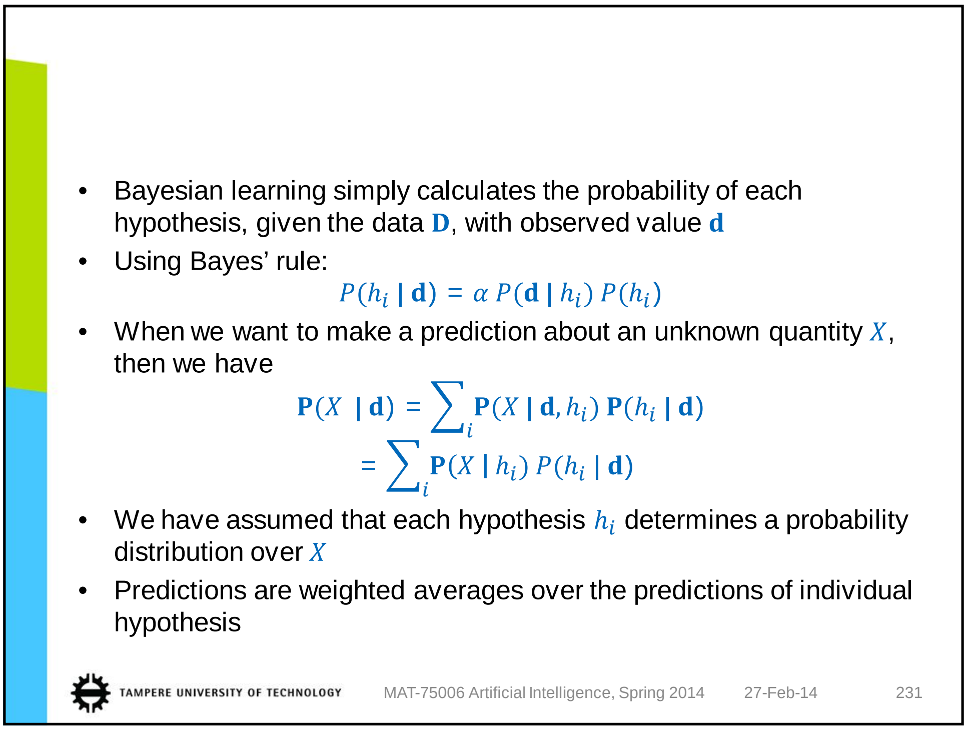
  h_1: 100% $cherry$

  h_2: 75% $cherry$ + 25% $lime$

  h_3: 50% $cherry$ + 50% $lime$

  h_4: 25% $cherry$ + 75% $lime$

  h_5: 100% $lime$
- Random variable H denotes the type of the bag and observation variables $D_1, \dots, D_n$ are the flavors of opened candies
- The task is to predict the flavor of the next piece of candy



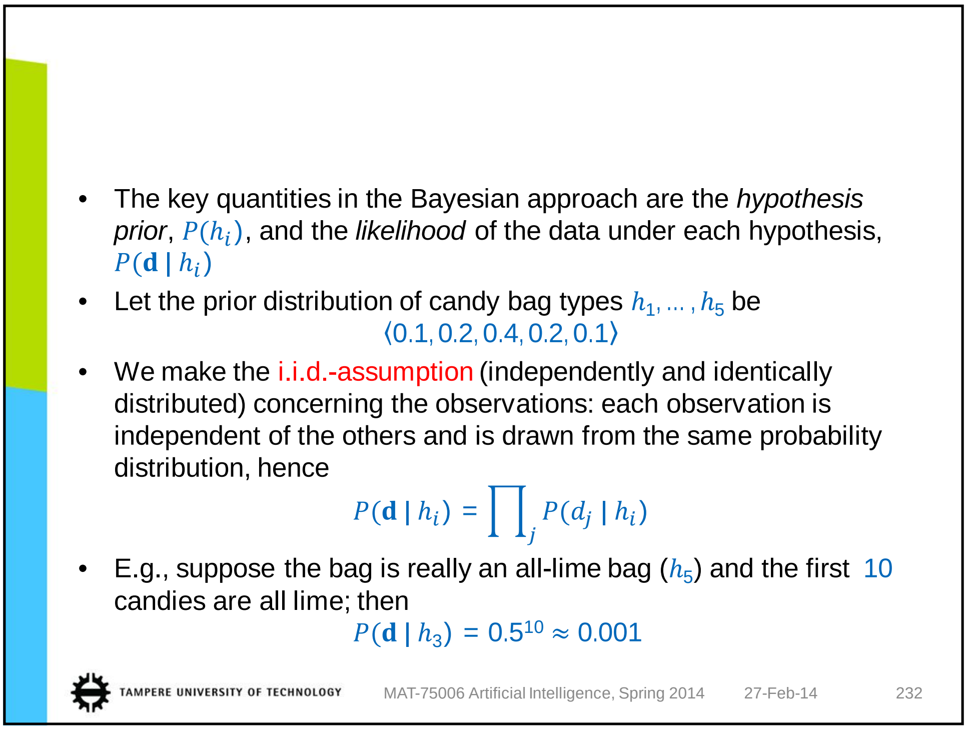
- Bayesian learning simply calculates the probability of each hypothesis, given the data $\mathbf{D}$, with observed value $\mathbf{d}$
- Using Bayes' rule:

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

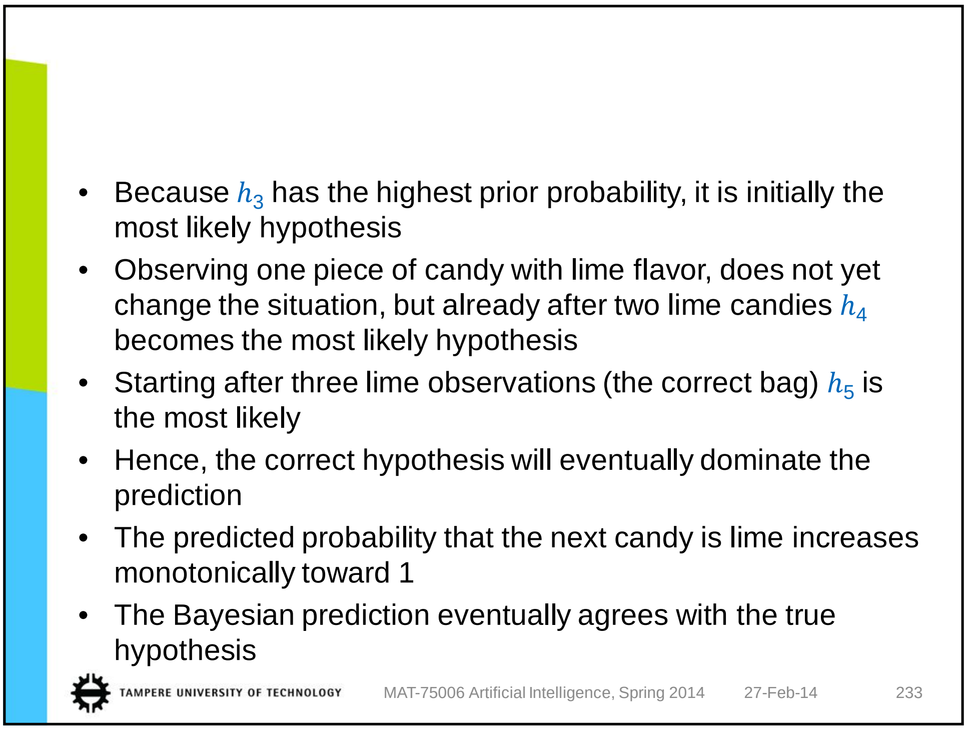
- When we want to make a prediction about an unknown quantity X, then we have

$$\mathbf{P}(X \mid \mathbf{d}) = \sum_i \mathbf{P}(X \mid \mathbf{d}, h_i)\, \mathbf{P}(h_i \mid \mathbf{d})$$
$$= \sum_i \mathbf{P}(X \mid h_i)\, P(h_i \mid \mathbf{d})$$

- We have assumed that each hypothesis h_i determines a probability distribution over X
- Predictions are weighted averages over the predictions of individual hypothesis



- The key quantities in the Bayesian approach are the *hypothesis prior*, $P(h_i)$, and the *likelihood* of the data under each hypothesis, $P(\mathbf{d} \mid h_i)$
- Let the prior distribution of candy bag types $h_1, \dots, h_5$ be $$\langle 0.1, 0.2, 0.4, 0.2, 0.1 \rangle$$
- We make the i.i.d.-assumption (independently and identically distributed) concerning the observations: each observation is independent of the others and is drawn from the same probability distribution, hence $$P(\mathbf{d} \mid h_i) = \prod_j P(d_j \mid h_i)$$
- E.g., suppose the bag is really an all-lime bag (h_5) and the first 10 candies are all lime; then $$P(\mathbf{d} \mid h_3) = 0.5^{10} \approx 0.001$$



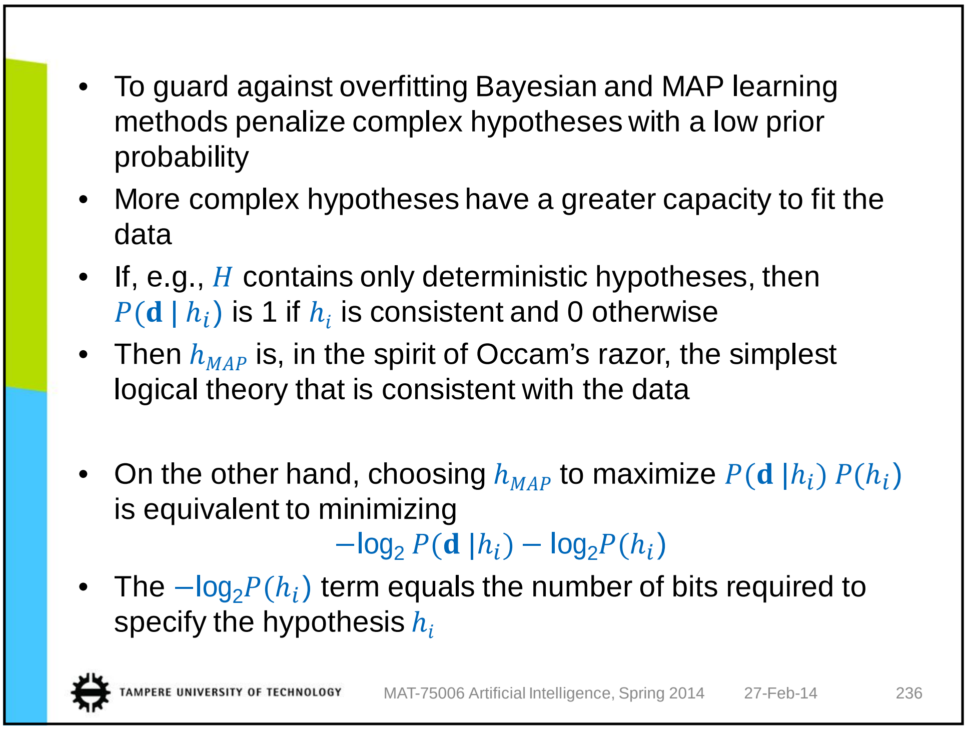
- Because h_3 has the highest prior probability, it is initially the most likely hypothesis
- Observing one piece of candy with lime flavor, does not yet change the situation, but already after two lime candies h_4 becomes the most likely hypothesis
- Starting after three lime observations (the correct bag) h_5 is the most likely
- Hence, the correct hypothesis will eventually dominate the prediction
- The predicted probability that the next candy is lime increases monotonically toward 1
- The Bayesian prediction eventually agrees with the true hypothesis

- Any fixed prior that does not rule out the true hypothesis, the posterior probability of any false hypothesis will (under certain technical conditions) eventually vanish
- This happens simply because the probability of generating “uncharacteristic” data indefinitely is vanishingly small
- The Bayesian prediction is optimal: any other prediction is expected to be correct less often (given the hypothesis prior)
- For real learning problems, the hypothesis space is usually very large or infinite
- In some cases, the summation (or integration, in the continuous case) over the hypothesis class can be carried out tracktably, but in most cases we must resort to approximate or simplified methods

- A very common approximation is to make predictions based on a single most probable; i.e., the one that maximizes the value

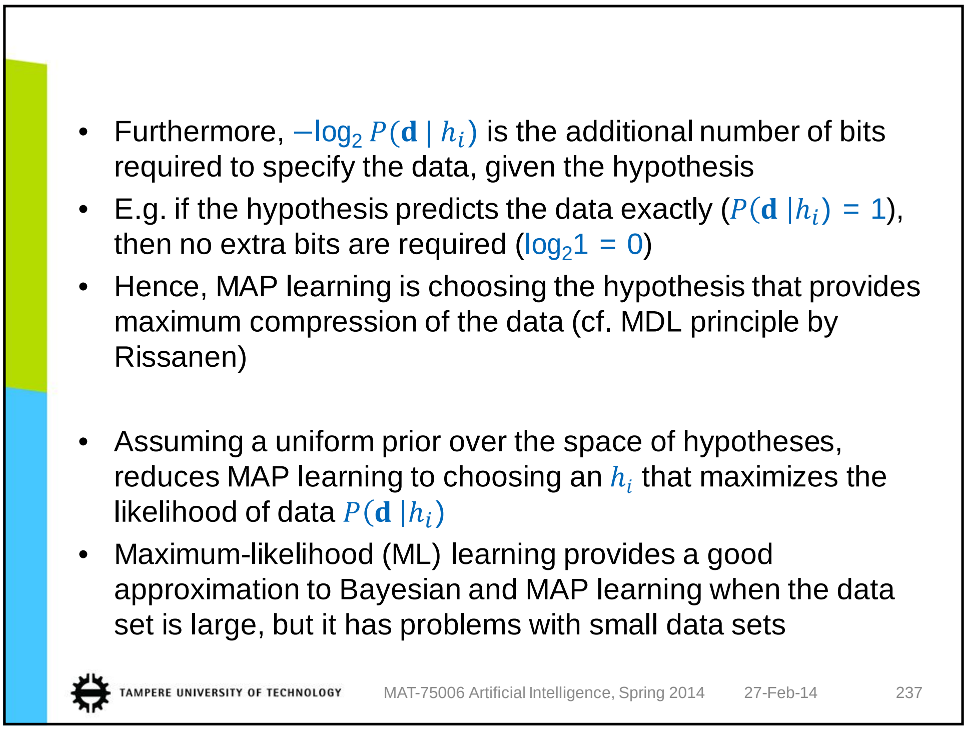
$$P(h_i \mid \mathbf{d})$$

- This is often called a *maximum a posteriori* (MAP) hypothesis h_{MAP}
- As more data arrive, the MAP prediction $\mathbf{P}(X \mid h_{MAP})$ and Bayesian prediction $\mathbf{P}(X \mid \mathbf{d})$ become closer, because the competitors to MAP hypothesis become less and less probable
- Instead of a summation (or integration) we now have to solve an optimization problem
- In our candy bag example, after three pieces of candy $h_{MAP} = h_5$ and the fourth candy is predicted to have lime flavor with probability 1.0, while the true Bayesian probability (averaged over all hypotheses) would be 0.8



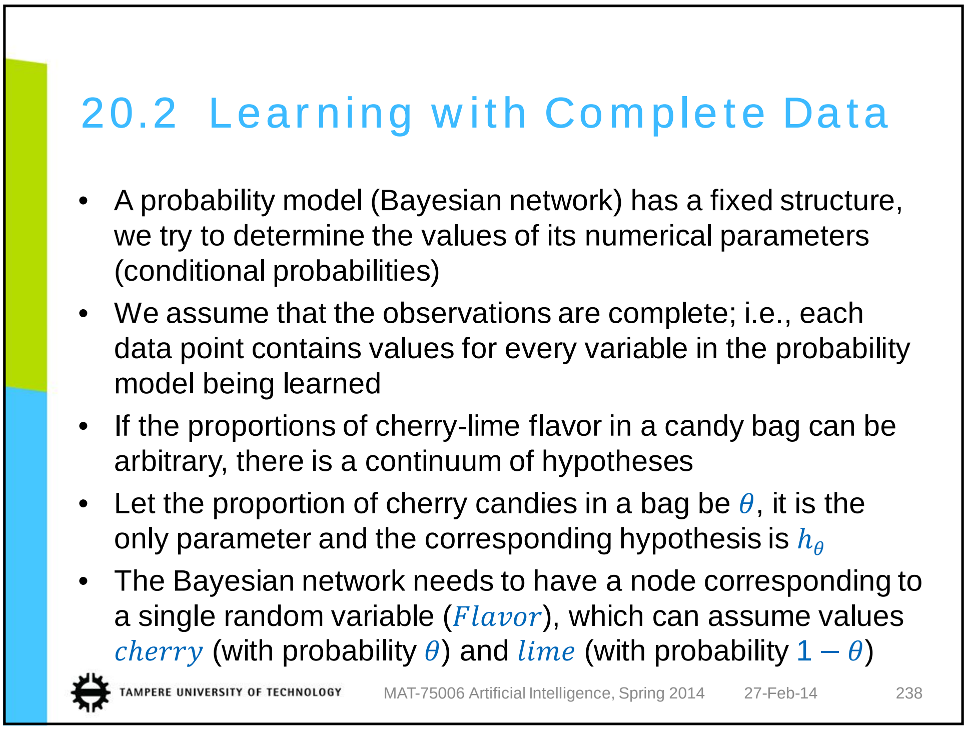
- To guard against overfitting Bayesian and MAP learning methods penalize complex hypotheses with a low prior probability
- More complex hypotheses have a greater capacity to fit the data
- If, e.g., H contains only deterministic hypotheses, then $P(\mathbf{d} \mid h_i)$ is 1 if h_i is consistent and 0 otherwise
- Then h_{MAP} is, in the spirit of Occam's razor, the simplest logical theory that is consistent with the data

- On the other hand, choosing h_{MAP} to maximize $P(\mathbf{d} \mid h_i)\, P(h_i)$ is equivalent to minimizing
$$-\log_2 P(\mathbf{d} \mid h_i) - \log_2 P(h_i)$$
- The $-\log_2 P(h_i)$ term equals the number of bits required to specify the hypothesis h_i



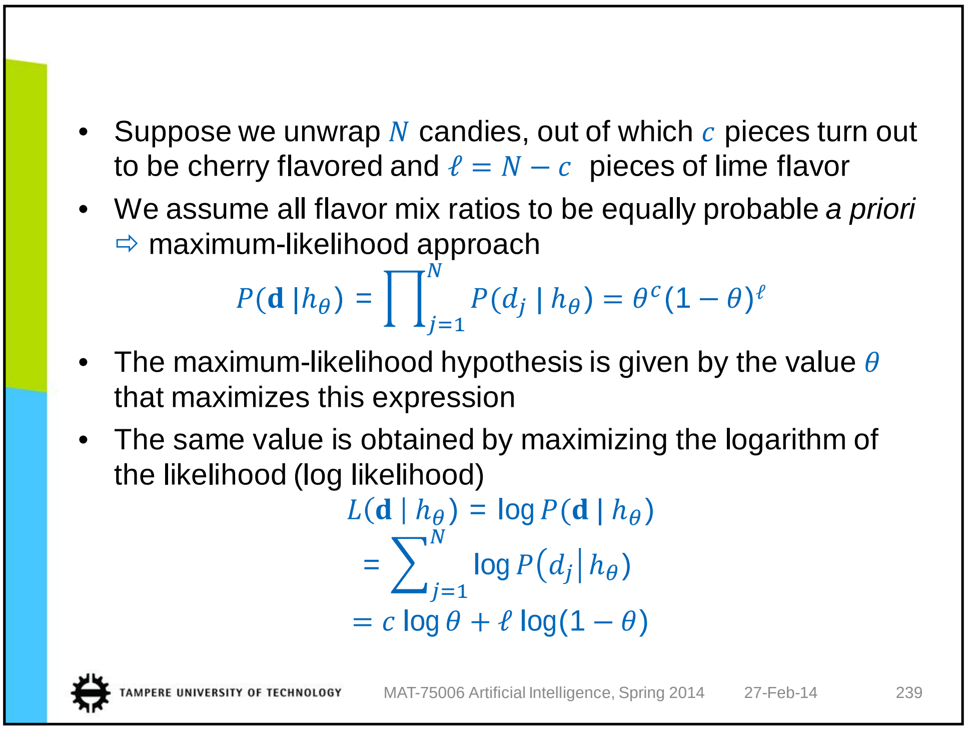
- Furthermore, $-\log_2 P(\mathbf{d} \mid h_i)$ is the additional number of bits required to specify the data, given the hypothesis
- E.g. if the hypothesis predicts the data exactly ($P(\mathbf{d} \mid h_i) = 1$), then no extra bits are required ($\log_2 1 = 0$)
- Hence, MAP learning is choosing the hypothesis that provides maximum compression of the data (cf. MDL principle by Rissanen)

- Assuming a uniform prior over the space of hypotheses, reduces MAP learning to choosing an h_i that maximizes the likelihood of data $P(\mathbf{d} \mid h_i)$
- Maximum-likelihood (ML) learning provides a good approximation to Bayesian and MAP learning when the data set is large, but it has problems with small data sets



20.2 Learning with Complete Data

- A probability model (Bayesian network) has a fixed structure, we try to determine the values of its numerical parameters (conditional probabilities)
- We assume that the observations are complete; i.e., each data point contains values for every variable in the probability model being learned
- If the proportions of cherry-lime flavor in a candy bag can be arbitrary, there is a continuum of hypotheses
- Let the proportion of cherry candies in a bag be θ, it is the only parameter and the corresponding hypothesis is h_θ
- The Bayesian network needs to have a node corresponding to a single random variable ($Flavor$), which can assume values $cherry$ (with probability θ) and $lime$ (with probability $1-\theta$)



- Suppose we unwrap N candies, out of which c pieces turn out to be cherry flavored and $\ell = N - c$ pieces of lime flavor
- We assume all flavor mix ratios to be equally probable *a priori* ⇨ maximum-likelihood approach

$$P(\mathbf{d}\,|h_\theta) = \prod_{j=1}^{N} P(d_j\,|\,h_\theta) = \theta^c(1-\theta)^\ell$$

- The maximum-likelihood hypothesis is given by the value θ that maximizes this expression
- The same value is obtained by maximizing the logarithm of the likelihood (log likelihood)

$$\begin{aligned} L(\mathbf{d}\,|\,h_\theta) &= \log P(\mathbf{d}\,|\,h_\theta) \\ &= \sum_{j=1}^{N} \log P\left(d_j \middle|\, h_\theta\right) \\ &= c\log\theta + \ell\log(1-\theta) \end{aligned}$$

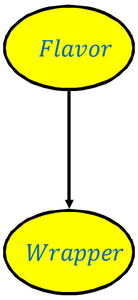
- Thus, the product over the data reduces to a sum over the data, which is usually easier to maximize
- Differentiating L with respect to θ and setting the resulting expression to zero gives the maximum-likelihood value of θ

$$dL(d \mid h_\theta)/d\theta = c/\theta - \ell/(1-\theta) = 0$$

$$\Rightarrow \theta = c/(c+\ell) = c/N$$

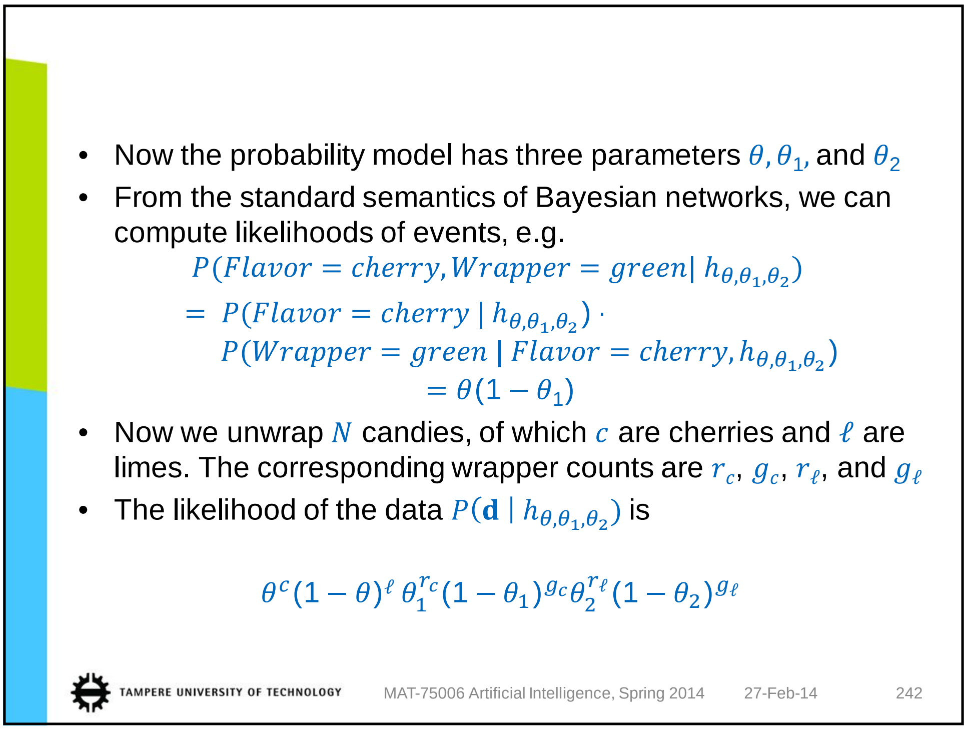
- The maximum-likelihood hypothesis h_{ML} asserts that the actual proportion of cherries in the bag is equal to as the observed proportion in the candies unwrapped so far
- This approach can be used more generally to disclose the values of more than one parameters

- A significant problem of the approach: if some events have not yet been observed (in a small enough data set), then h_{ML} assigns zero probability to those events
- Let us change the example so that, depending on the flavor of the candy, it is wrapped by a probabilistic rule either in red or green wrapper



$P(cherry)$
θ

$Flavor$	$P(red \mid F)$
$cherry$	θ_1
$lime$	θ_2

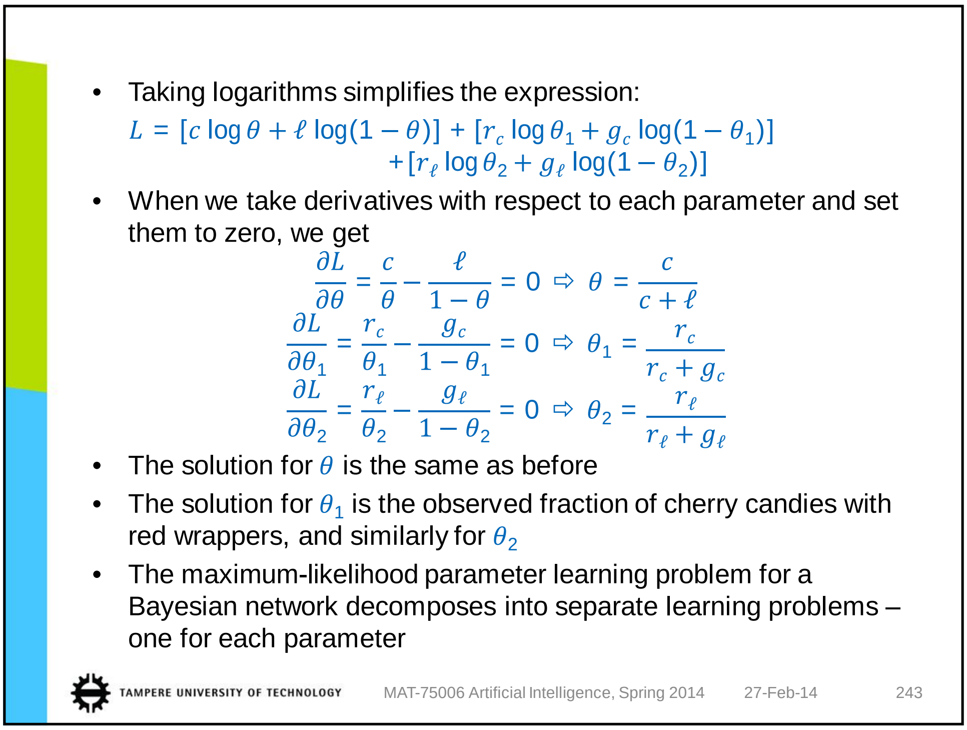


- Now the probability model has three parameters θ, θ_1, and θ_2
- From the standard semantics of Bayesian networks, we can compute likelihoods of events, e.g.

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

- Now we unwrap N candies, of which c are cherries and ℓ are limes. The corresponding wrapper counts are r_c, g_c, r_ℓ, and g_ℓ
- The likelihood of the data $P(\mathbf{d} \mid h_{\theta,\theta_1,\theta_2})$ is

$$\theta^c(1-\theta)^\ell\, \theta_1^{r_c}(1-\theta_1)^{g_c}\theta_2^{r_\ell}(1-\theta_2)^{g_\ell}$$



- Taking logarithms simplifies the expression:

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

- When we take derivatives with respect to each parameter and set them to zero, we get

$$
\begin{aligned}
\frac{\partial L}{\partial \theta} &= \frac{c}{\theta} - \frac{\ell}{1-\theta} = 0 \Rightarrow \theta = \frac{c}{c+\ell} \\
\frac{\partial L}{\partial \theta_1} &= \frac{r_c}{\theta_1} - \frac{g_c}{1-\theta_1} = 0 \Rightarrow \theta_1 = \frac{r_c}{r_c+g_c} \\
\frac{\partial L}{\partial \theta_2} &= \frac{r_\ell}{\theta_2} - \frac{g_\ell}{1-\theta_2} = 0 \Rightarrow \theta_2 = \frac{r_\ell}{r_\ell+g_\ell}
\end{aligned}
$$

- The solution for θ is the same as before
- The solution for θ_1 is the observed fraction of cherry candies with red wrappers, and similarly for θ_2
- The maximum-likelihood parameter learning problem for a Bayesian network decomposes into separate learning problems – one for each parameter

Naive Bayes models

- The class variable C is the root of the Bayesian network and the attribute variables X_i are its leaves
- The naïve assumption is that the attributes are conditionally independent of each other given the class
- Assuming Boolean variables, the parameters are

$$\theta = P(C = true)$$
$$\theta_{i1} = P(X_i = true \mid C = true)$$
$$\theta_{i2} = P(X_i = true \mid C = false)$$

- The maximum-likelihood parameter values are found exactly the same way as before
- Once the model has been trained, then observation $[x_1, \dots, x_n]$, for which the class variable C is unobserved, can be classified

$$\mathbf{P}(C \mid x_1, \dots, x_n) = \alpha\, \mathbf{P}(C) \prod_i \mathbf{P}(x_i \mid C)$$

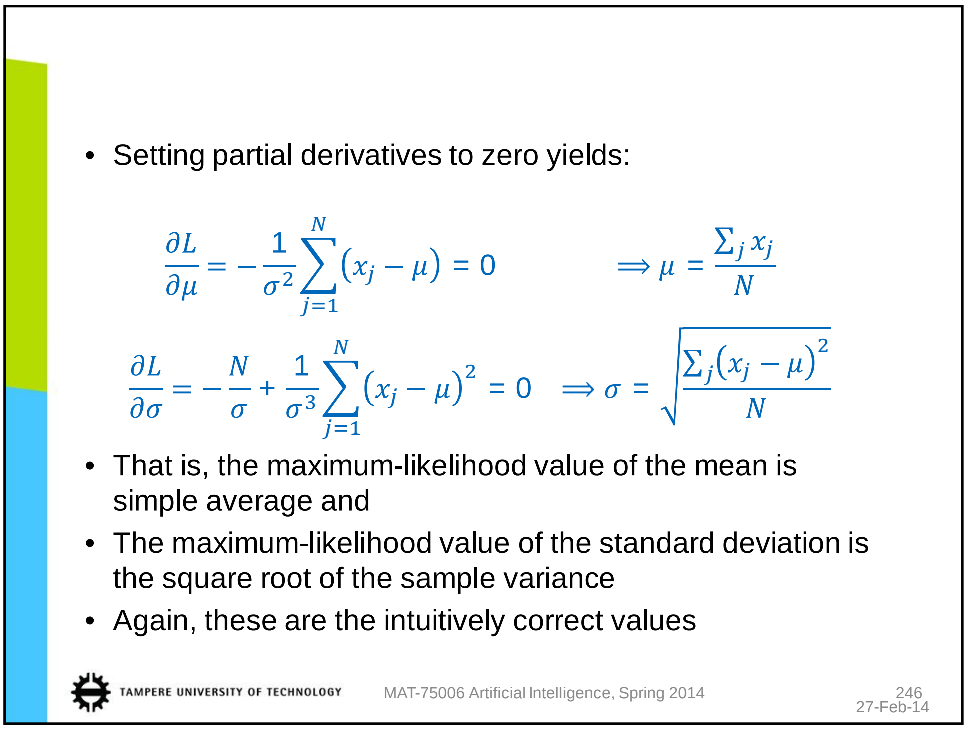
Continuous models

- Let us examine learning the parameters of a Gaussian density function on a single variable
- The data are generated by:

$$P(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-(x-\mu)^2}{2\sigma^2}}$$

- The parameters of the model are mean μ and standard deviation σ
- Let the observed values be $x_1, \dots, x_N$
- Then the log likelihood is

$$L = \sum_{j=1}^{N} \log \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-(x_j-\mu)^2}{2\sigma^2}}$$
$$= N\left(-\log\sqrt{2\pi} - \log\sigma\right) - \sum_{j=1}^{N} \frac{-(x_j-\mu)^2}{2\sigma^2}$$

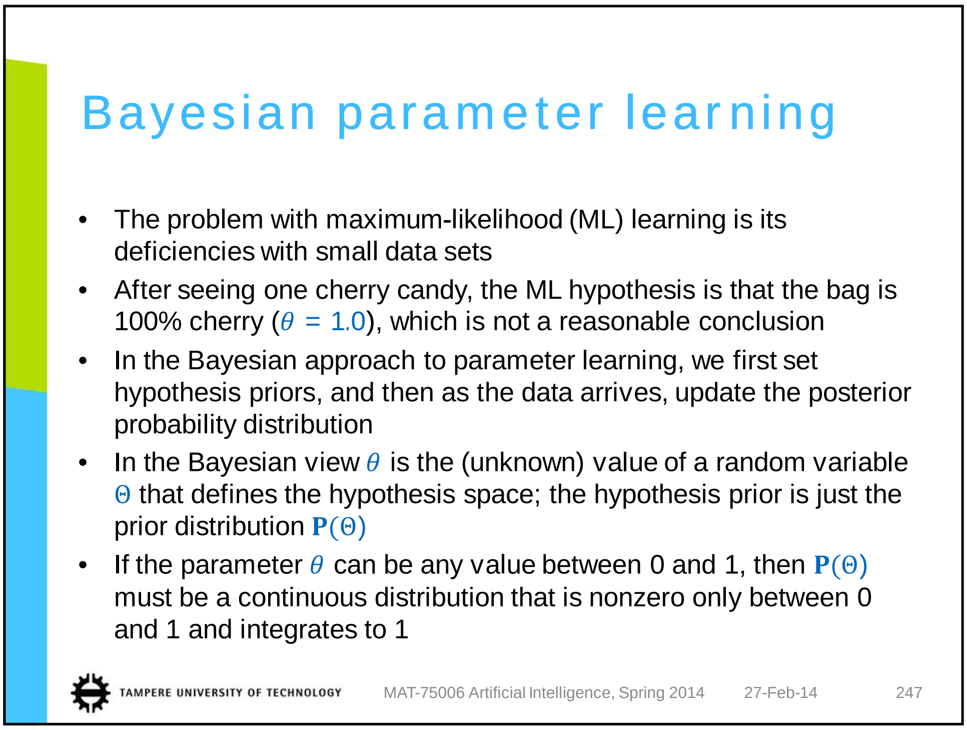


- Setting partial derivatives to zero yields:

$$\frac{\partial L}{\partial \mu} = -\frac{1}{\sigma^2}\sum_{j=1}^{N}(x_j - \mu) = 0 \qquad \Longrightarrow \mu = \frac{\sum_j x_j}{N}$$

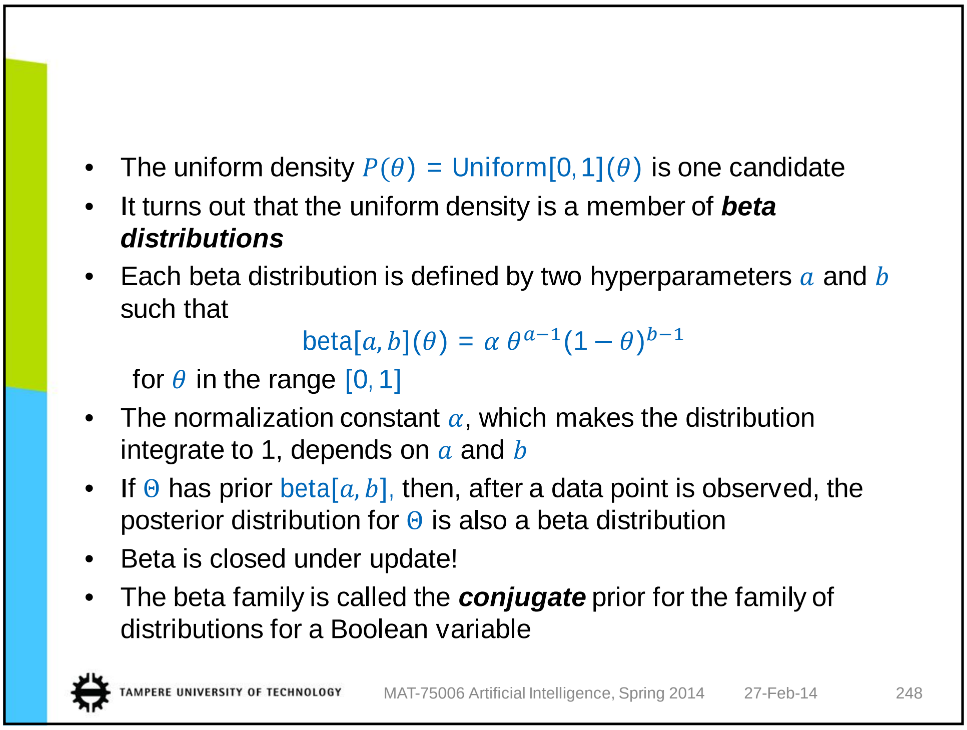
$$\frac{\partial L}{\partial \sigma} = -\frac{N}{\sigma} + \frac{1}{\sigma^3}\sum_{j=1}^{N}(x_j - \mu)^2 = 0 \quad \Longrightarrow \sigma = \sqrt{\frac{\sum_j (x_j - \mu)^2}{N}}$$

- That is, the maximum-likelihood value of the mean is simple average and
- The maximum-likelihood value of the standard deviation is the square root of the sample variance
- Again, these are the intuitively correct values

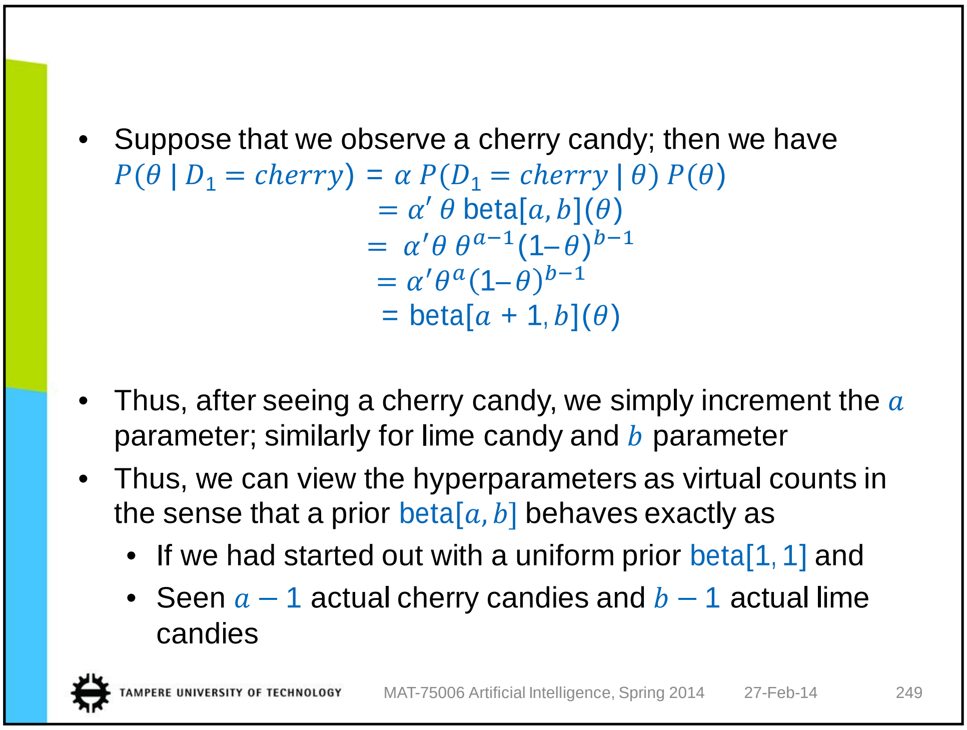


Bayesian parameter learning

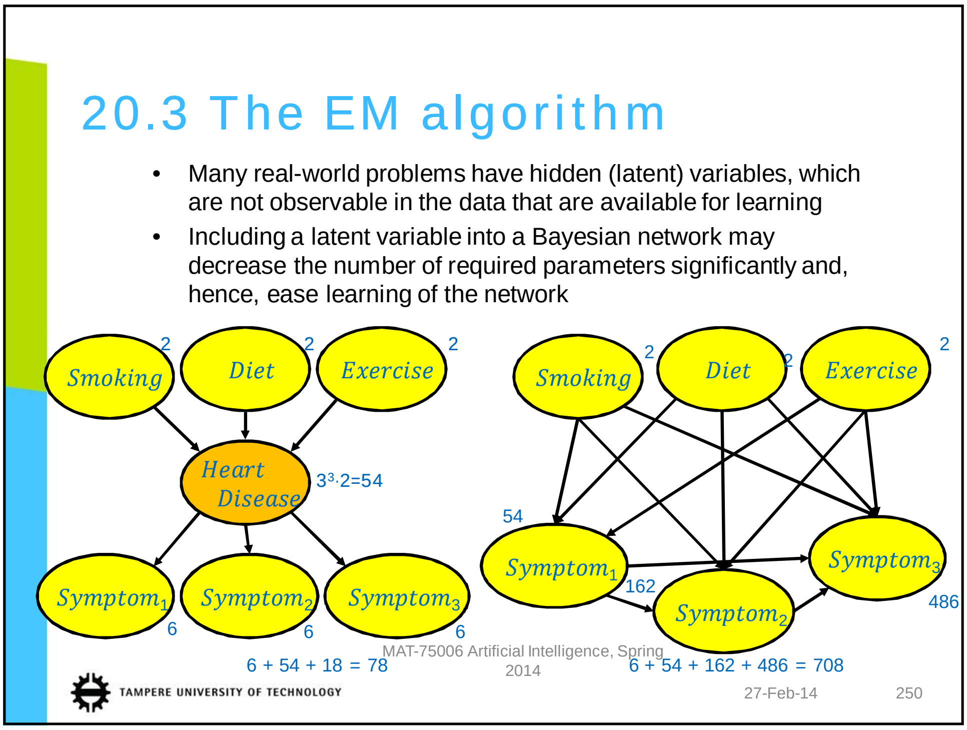
- The problem with maximum-likelihood (ML) learning is its deficiencies with small data sets
- After seeing one cherry candy, the ML hypothesis is that the bag is 100% cherry ($\theta = 1.0$), which is not a reasonable conclusion
- In the Bayesian approach to parameter learning, we first set hypothesis priors, and then as the data arrives, update the posterior probability distribution
- In the Bayesian view θ is the (unknown) value of a random variable Θ that defines the hypothesis space; the hypothesis prior is just the prior distribution $\mathbf{P}(\Theta)$
- If the parameter θ can be any value between 0 and 1, then $\mathbf{P}(\Theta)$ must be a continuous distribution that is nonzero only between 0 and 1 and integrates to 1



- The uniform density $P(\theta) = \text{Uniform}[0, 1](\theta)$ is one candidate
- It turns out that the uniform density is a member of ***beta distributions***
- Each beta distribution is defined by two hyperparameters a and b such that
$$\text{beta}[a, b](\theta) = \alpha\, \theta^{a-1}(1-\theta)^{b-1}$$
  for θ in the range $[0, 1]$
- The normalization constant α, which makes the distribution integrate to 1, depends on a and b
- If Θ has prior $\text{beta}[a, b]$, then, after a data point is observed, the posterior distribution for Θ is also a beta distribution
- Beta is closed under update!
- The beta family is called the ***conjugate*** prior for the family of distributions for a Boolean variable

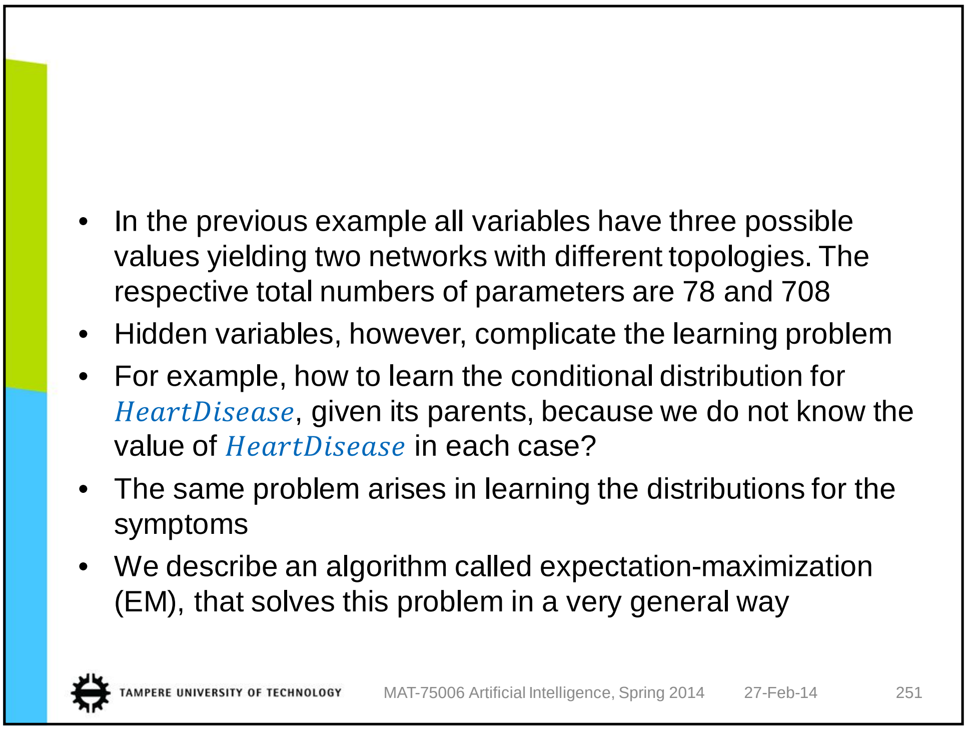


- Suppose that we observe a cherry candy; then we have
$$\begin{aligned} P(\theta \mid D_1 = cherry) &= \alpha\, P(D_1 = cherry \mid \theta)\, P(\theta) \\ &= \alpha'\, \theta\, \text{beta}[a, b](\theta) \\ &= \alpha' \theta\, \theta^{a-1}(1-\theta)^{b-1} \\ &= \alpha' \theta^{a}(1-\theta)^{b-1} \\ &= \text{beta}[a+1, b](\theta) \end{aligned}$$
- Thus, after seeing a cherry candy, we simply increment the a parameter; similarly for lime candy and b parameter
- Thus, we can view the hyperparameters as virtual counts in the sense that a prior $\text{beta}[a, b]$ behaves exactly as
  - If we had started out with a uniform prior $\text{beta}[1, 1]$ and
  - Seen $a-1$ actual cherry candies and $b-1$ actual lime candies

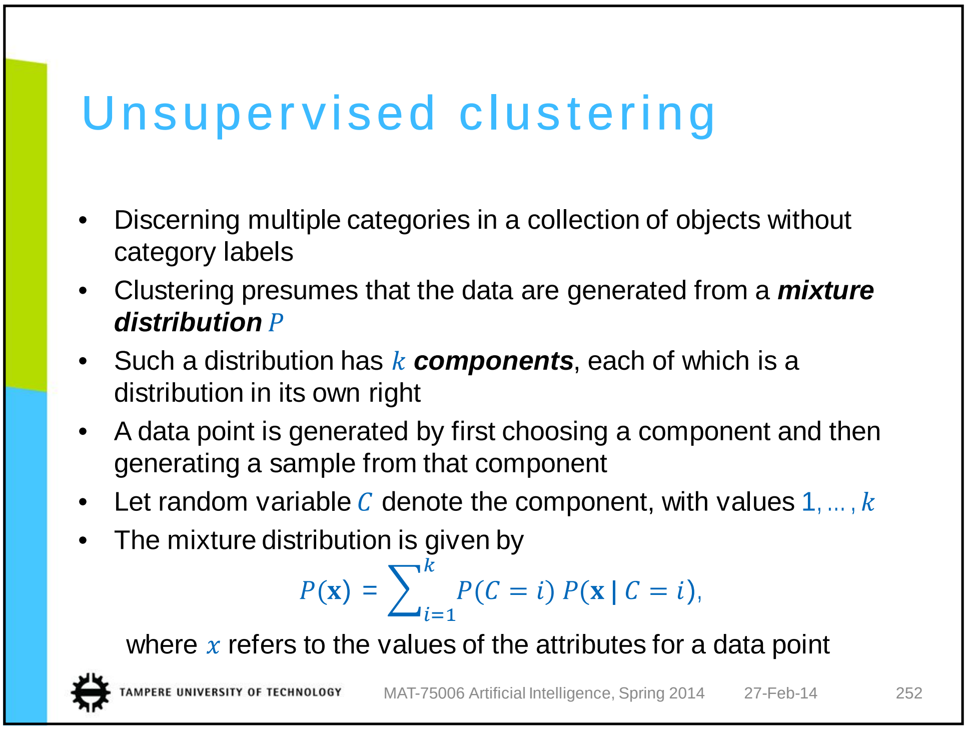


20.3 The EM algorithm

- Many real-world problems have hidden (latent) variables, which are not observable in the data that are available for learning
- Including a latent variable into a Bayesian network may decrease the number of required parameters significantly and, hence, ease learning of the network



- In the previous example all variables have three possible values yielding two networks with different topologies. The respective total numbers of parameters are 78 and 708
- Hidden variables, however, complicate the learning problem
- For example, how to learn the conditional distribution for $HeartDisease$, given its parents, because we do not know the value of $HeartDisease$ in each case?
- The same problem arises in learning the distributions for the symptoms
- We describe an algorithm called expectation-maximization (EM), that solves this problem in a very general way

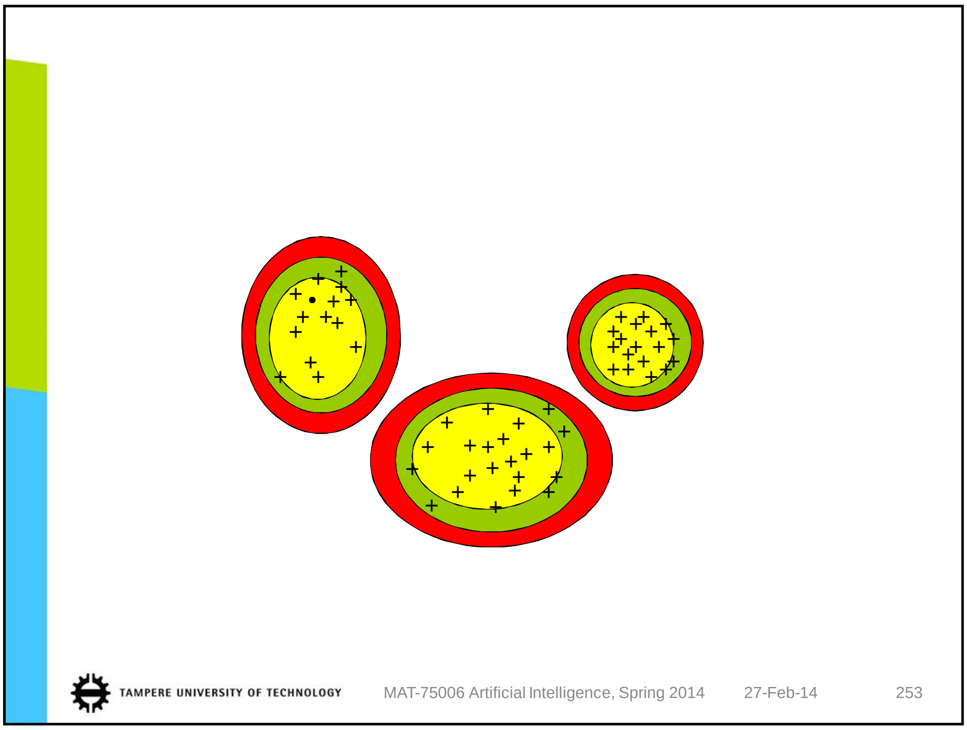


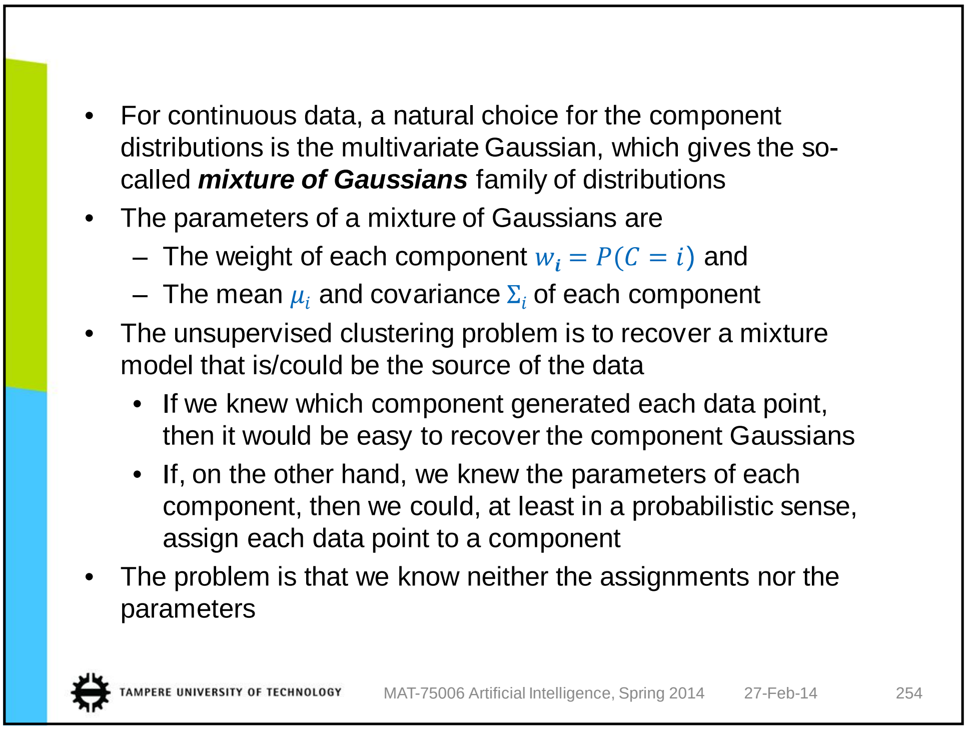
Unsupervised clustering

- Discerning multiple categories in a collection of objects without category labels
- Clustering presumes that the data are generated from a ***mixture distribution*** P
- Such a distribution has k ***components***, each of which is a distribution in its own right
- A data point is generated by first choosing a component and then generating a sample from that component
- Let random variable C denote the component, with values $1, \dots, k$
- The mixture distribution is given by

$$P(\mathbf{x}) = \sum_{i=1}^{k} P(C=i)\, P(\mathbf{x} \mid C=i),$$

where x refers to the values of the attributes for a data point





- For continuous data, a natural choice for the component distributions is the multivariate Gaussian, which gives the so-called ***mixture of Gaussians*** family of distributions
- The parameters of a mixture of Gaussians are
  - The weight of each component $w_{\boldsymbol{i}} = P(C = i)$ and
  - The mean μ_i and covariance Σ_i of each component
- The unsupervised clustering problem is to recover a mixture model that is/could be the source of the data
  - If we knew which component generated each data point, then it would be easy to recover the component Gaussians
  - If, on the other hand, we knew the parameters of each component, then we could, at least in a probabilistic sense, assign each data point to a component
- The problem is that we know neither the assignments nor the parameters