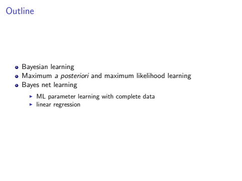
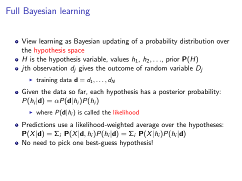


Outline

- Bayesian learning
- Maximum *a posteriori* and maximum likelihood learning
- Bayes net learning
  - ML parameter learning with complete data
  - linear regression

We have already seen at probabilistic inference that knowledge is uncertain. We can deal with uncertainty, but to do so we need to set up a probabilistic model of the world. We will show that the Bayesian approach is also a very effective and general solution to the problem of noise and overfitting. Since the Bayesian approach is very calculation-intensive, we present two other approaches that give almost as accurate results but are easier to compute. We also look at how the parameters of the Bayesian network can be determined in a discrete or continuous case.

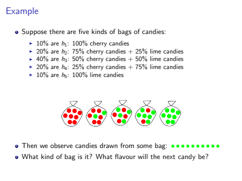
└Full Bayesian learning



Full Bayesian learning

- 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 $\mathbf{P}(H)$
- jth observation d_j gives the outcome of random variable D_j
  - training data $\mathbf{d} = d_1, \ldots, d_N$
- Given the data so far, each hypothesis has a posterior probability: $P(h_i|\mathbf{d}) = \alpha P(\mathbf{d}|h_i)P(h_i)$
  - where $P(\mathbf{d}|h_i)$ is called the likelihood
- Predictions use a likelihood-weighted average over the hypotheses: $\mathbf{P}(X|\mathbf{d}) = \Sigma_i\ \mathbf{P}(X|\mathbf{d}, h_i)P(h_i|\mathbf{d}) = \Sigma_i\ \mathbf{P}(X|h_i)P(h_i|\mathbf{d})$
- No need to pick one best-guess hypothesis!

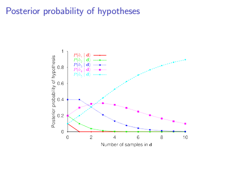
Think of Bayesian learning as a change in the probability distribution over the hypothesis space. For this we have a hypothesis variable H, which has values – the hypotheses – and these values have prior probabilities even before the learning process, i.e. H has a probability distribution. The results of experiments – the observations – are related to random variables. All the observations will be denoted by d. By applying the Bayes-rule, the conditional probability of each hypothesis in case of the observation can be calculated, where the conditional probability of the sample in case of hypothesis – called likelihood – plays an important role. If we want to predict something according to the sample, we need a weighted sum of probability distribution of this certain something in case of hypothesis. It is important: there is no need to select the most probable hypothesis, we use all of them.



Example

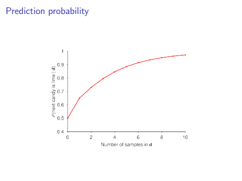
- Suppose there are five kinds of bags of candies:
  - 10% are h_1: 100% cherry candies
  - 20% are h_2: 75% cherry candies + 25% lime candies
  - 40% are h_3: 50% cherry candies + 50% lime candies
  - 20% are h_4: 25% cherry candies + 75% lime candies
  - 10% are h_5: 100% lime candies
- Then we observe candies drawn from some bag: ●●●●●●●●●●
- What kind of bag is it? What flavour will the next candy be?

A candy factory sells its candies in different configurations, there are different proportions of cherry and lemon candies in the bags. (Lets think of sack-sized bags – we have thousands of candies in a bag, not only 8, as you see in the picture – to take away the difference between sampling with and without replacement.) We have priori probabilities about compositions. We are taking the candies out of the bag one by one and each one will be lemon. What kind of bag it is, and what will be the next candy?

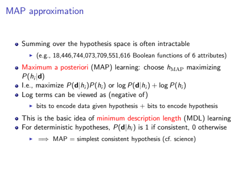


Posterior probability of hypotheses

Before we take the first candy out of the bag, we can only use the prior probabilities, and according to them h_3 is the most probable hypothesis. But even h_1 has a 10% chance. However, by pulling out the first candy, the probability of h_1 falls to 0%. As more and more lemon candies come out, h_5 becomes the most likely hypothesis. If we are curious about the probability of hypothesis h_3 – where there is the same amount of cherry and lemon candies – then the chance of continuously pulling a lemon candies is 0.5^n, which approximates to zero.


Prediction probability

By using the formula at the bottom of page 3 the likelihood that subsequent candies will also be lemon is increasing from candy to candy.

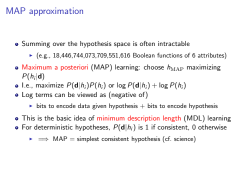


MAP approximation

- Summing over the hypothesis space is often intractable
  - (e.g., 18,446,744,073,709,551,616 Boolean functions of 6 attributes)
- Maximum a posteriori (MAP) learning: choose h_{MAP} maximizing $P(h_i|\mathbf{d})$
- I.e., maximize $P(\mathbf{d}|h_i)P(h_i)$ or $\log P(\mathbf{d}|h_i) + \log P(h_i)$
- Log terms can be viewed as (negative of)
  - bits to encode data given hypothesis + bits to encode hypothesis
- This is the basic idea of minimum description length (MDL) learning
- For deterministic hypotheses, $P(\mathbf{d}|h_i)$ is 1 if consistent, 0 otherwise
  - $\implies$ MAP $=$ simplest consistent hypothesis (cf. science)

The problem with the Bayesian method is that it would take a huge amount of computation to determine accurate forecasts. For simplicity, we only use one term, the one that is most probable, that is the one for which $P(h_i|d)$ is maximal. This could be dangerous as the Bayesian method gives an 80% chance for the fourth candy, whereas with this method it is 100%, however for large datasets the difference will be negligible.

We know from earlier that $P(h_i|d)$ is proportional to $P(d|h_i)P(h_i)$, and we need to take its maximum. The logarithm of the maximal value is also maximal, and this leads us to the concept of information, because $\log P(d|h_i) + \log P(h_i)$ is maximal iff $-\log P(d|h_i) - \log P(h_i)$ is minimal, which is the entropy of the data in case of the hypothesis plus the entropy of the hypothesis h_i. This sum should be minimized, i.e. described in the shortest form.



MAP approximation

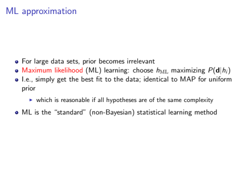
- Summing over the hypothesis space is often intractable
  - (e.g., 18,446,744,073,709,551,616 Boolean functions of 6 attributes)
- Maximum a posteriori (MAP) learning: choose h_{MAP} maximizing $P(h_i|\mathbf{d})$
- I.e., maximize $P(\mathbf{d}|h_i)P(h_i)$ or $\log P(\mathbf{d}|h_i) + \log P(h_i)$
- Log terms can be viewed as (negative of)
  - bits to encode data given hypothesis + bits to encode hypothesis
- This is the basic idea of minimum description length (MDL) learning
- For deterministic hypotheses, $P(\mathbf{d}|h_i)$ is 1 if consistent, 0 otherwise
  - $\Longrightarrow$ MAP = simplest consistent hypothesis (cf. science)

If our hypotheses does not contain uncertainty (there are only cherry candies or only lemon candies in the bag), then the conditional probability can only be 1 or 0. The former means that the hypothesis is consistent with the data, so Occams razor is a good method for prediction.

AI #2

2020-05-01

└ML approximation

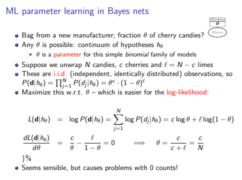


ML approximation

- For large data sets, prior becomes irrelevant
- Maximum likelihood (ML) learning: choose h_{ML} maximizing $P(\mathbf{d}|h_i)$
- 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

If the prior probabilities of our hypotheses are the same, then we only need to select the h_i for which $P(d|h_i)$ will be maximal. It the sample is big enough, it is worth using this simplification. This is the reason why this method is very often used in statistics.

└ML parameter learning in Bayes nets



ML parameter learning in Bayes nets

- Bag from a new manufacturer; fraction θ of cherry candies?
- Any θ is possible: continuum of hypotheses h_θ
    - θ is a parameter for this simple *binomial* family of models
- Suppose we unwrap N candies, c cherries and $\ell = N - c$ limes
- These are i.i.d. (independent, identically distributed) observations, so $P(\mathbf{d}|h_\theta) = \prod_{j=1}^{N} P(d_j|h_\theta) = \theta^c \cdot (1-\theta)^\ell$
- Maximize this w.r.t. θ – which is easier for the log-likelihood:

$$L(\mathbf{d}|h_\theta) = \log P(\mathbf{d}|h_\theta) = \sum_{j=1}^{N} \log P(d_j|h_\theta) = c\log\theta + \ell\log(1-\theta)$$

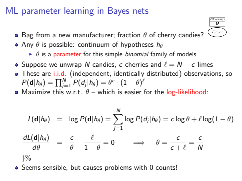
$$\frac{dL(\mathbf{d}|h_\theta)}{d\theta} = \frac{c}{\theta} - \frac{\ell}{1-\theta} = 0 \qquad \Longrightarrow \qquad \theta = \frac{c}{c+\ell} = \frac{c}{N}$$

}%

- Seems sensible, but causes problems with 0 counts!

Suppose that a bag of candy comes from a completely different manufacturer. We dont know what the cherry-lemon ratio will be here. The proportion of cherry candies among all candies is denoted by theta. Thus we will have an infinite number of h_θ hypotheses. Assuming that all θ values can occur with the same probability, it is worthwhile using an ML approach. The associated Bayes network contains only one vertex for which the prior probability is θ. Consider a sample of size N containing c cherries and l lemon candies ($c + l = N$). The probability of this sample in case of h_θ can be calculated by the binomial theorem, assuming that candies have the same taste with the same probability and members of the sample are independent of each other. The question is when does this value have a maximum. To do this, we consider its logarithm – so the product is converted into a sum. The derivative at the maximum is 0, so we get that the best approximation is the ratio in the sample.

ML parameter learning in Bayes nets



ML parameter learning in Bayes nets

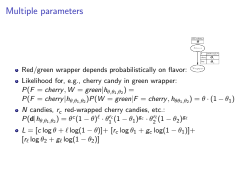
- Bag from a new manufacturer; fraction θ of cherry candies?
- Any θ is possible: continuum of hypotheses h_θ
  - θ is a parameter for this simple *binomial* family of models
- Suppose we unwrap N candies, c cherries and $\ell = N - c$ limes
- These are i.i.d. (independent, identically distributed) observations, so $P(\mathbf{d}|h_\theta) = \prod_{j=1}^{N} P(d_j|h_\theta) = \theta^c \cdot (1-\theta)^\ell$
- Maximize this w.r.t. θ – which is easier for the log-likelihood:

$$\begin{aligned} L(\mathbf{d}|h_\theta) &= \log P(\mathbf{d}|h_\theta) = \sum_{j=1}^{N} \log P(d_j|h_\theta) = c\log\theta + \ell\log(1-\theta) \\ \frac{dL(\mathbf{d}|h_\theta)}{d\theta} &= \frac{c}{\theta} - \frac{\ell}{1-\theta} = 0 \qquad \Longrightarrow \qquad \theta = \frac{c}{c+\ell} = \frac{c}{N} \end{aligned}$$

}%

- Seems sensible, but causes problems with 0 counts!

If the sample is small, there may be cases that have not yet occurred, so we would assign zero probability to this. In this case, it is worth using all sorts of tricks to complete the calculations.



Multiple parameters

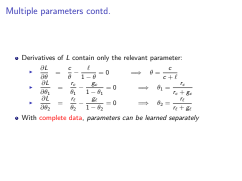
- Red/green wrapper depends probabilistically on flavor:

- Likelihood for, e.g., cherry candy in green wrapper:
  $P(F = cherry, W = green|h_{\theta,\theta_1,\theta_2}) =$
  $P(F = cherry|h_{\theta,\theta_1,\theta_2})P(W = green|F = cherry, h_{\theta\theta_1,\theta_2}) = \theta \cdot (1 - \theta_1)$
- N candies, r_c red-wrapped cherry candies, etc.:
  $P(\mathbf{d}|h_{\theta,\theta_1,\theta_2}) = \theta^c(1-\theta)^\ell \cdot \theta_1^{r_c}(1-\theta_1)^{g_c} \cdot \theta_2^{r_\ell}(1-\theta_2)^{g_\ell}$
- $L = [c \log\theta + \ell\log(1-\theta)] + [r_c \log\theta_1 + g_c \log(1-\theta_1)] +$
  $[r_\ell \log\theta_2 + g_\ell \log(1-\theta_2)]$

Suppose the candy manufacturer does not wrap the candies uniformly: the color of the wrapper indicates the taste of the candy, but does not necessarily identify it. This means that we have three parameters here, the previous θ is supplemented by θ_1, which specifies the probability that the cherry candy is wrapped in red paper, and we also have a θ_2, which specifies the probability that the lemon candy is wrapped in red paper. The Bayes-network in this case consists of two vertices. The conditional probability of an elementary event can easily be given, which can be traced back to the parameters.

Consider a pattern in which all four elementary events can occur. The probability of this can be written up easily, and we can similarly write up the logarithm of this probability as well.

Multiple parameters contd.

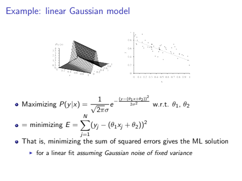


Multiple parameters contd.

- Derivatives of L contain only the relevant parameter:
  - $\frac{\partial L}{\partial \theta} = \frac{c}{\theta} - \frac{\ell}{1-\theta} = 0 \implies \theta = \frac{c}{c+\ell}$
  - $\frac{\partial L}{\partial \theta_1} = \frac{r_c}{\theta_1} - \frac{g_c}{1-\theta_1} = 0 \implies \theta_1 = \frac{r_c}{r_c+g_c}$
  - $\frac{\partial L}{\partial \theta_2} = \frac{r_\ell}{\theta_2} - \frac{g_\ell}{1-\theta_2} = 0 \implies \theta_2 = \frac{r_\ell}{r_\ell+g_\ell}$
- With complete data, *parameters can be learned separately*

We have three parameters, and we looking for a maximum in accordance with them all, so we take the derivative according to all three directions, and taking each zero, we get the most probable parameters. This method can be extended to any discrete Bayesian network.

└Example: linear Gaussian model

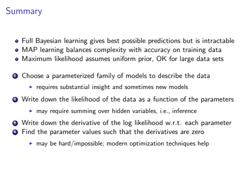


Example: linear Gaussian model

- Maximizing $P(y|x) = \frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{(y-(\theta_1 x+\theta_2))^2}{2\sigma^2}}$ w.r.t. θ_1, θ_2
- = minimizing $E = \sum_{j=1}^{N}(y_j - (\theta_1 x_j + \theta_2))^2$
- That is, minimizing the sum of squared errors gives the ML solution
  - for a linear fit *assuming Gaussian noise of fixed variance*

In real life, we usually need to use continuous probabilistic models, and very often our data is normally distributed. The normal distribution has a Gaussian density function, which has two parameters, mu and sigma – the mean and the standard deviation, respectively.

Consider a linear Gaussian model where both parent X and child Y are continuous and Y is a linear combination of X ($y = \theta_1 x + \theta_2$). Thus, the mean of the variable Y depends on the value of X, while the standard deviation remains the same. The conditional probability – or its logarithm presented here should be maximized, i.e., the value behind the minus sign should be minimized. As we are working with a sample, these probabilities are multiplied, so their logarithms are added together, therefore the sum of squares shown here must be minimized – that is the sum of the error squares, for which linear regression can be used here.



Summary

- Full Bayesian learning gives best possible predictions but is intractable
- MAP learning balances complexity with accuracy on training data
- Maximum likelihood assumes uniform prior, OK for large data sets

1. Choose a parameterized family of models to describe the data
   - requires substantial insight and sometimes new models
2. Write down the likelihood of the data as a function of the parameters
   - may require summing over hidden variables, i.e., inference
3. Write down the derivative of the log likelihood w.r.t. each parameter
4. Find the parameter values such that the derivatives are zero
   - may be hard/impossible; modern optimization techniques help

We have seen three methods today, in the order shown here, going from the more complex one towards the simplest, but special conditions are needed to use the simpler ones successfully. Here is the recipe using which the specific model can be constructed.