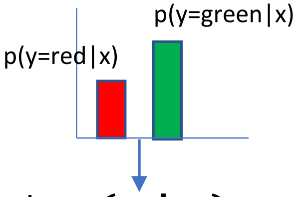
Probabilistic Machine Learning (3): Parameter Estimation via Maximum Likelihood

CS771: Introduction to Machine Learning

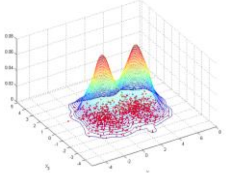
Piyush Rai

Probabilistic ML: Some Motivation

- In many ML problems, we want to model and reason about data probabilistically
- At a high-level, this is the density estimation view of ML, e.g.,



  - Given input-output pairs $\{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_N, y_N)\}$ estimate the conditional $p(y|\boldsymbol{x})$
  - Given inputs $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N\}$, estimate the distribution $p(\boldsymbol{x})$ of the inputs



  - Note 1: These dist. will depend on some parameters $\boldsymbol{\theta}$ (to be estimated), and written as

$$p(y|\boldsymbol{x}, \theta) \qquad \text{or} \qquad p(\boldsymbol{x}|\theta)$$

  - Note 2: These dist. sometimes assumed to have a specific form, but sometimes not
- Assuming the form of the distribution to be known, the goal in estimation is to use the observed data to estimate the parameters of these distributions

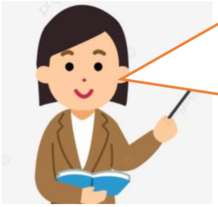
Probabilistic Modeling: The Basic Idea

- Assume N observations $\boldsymbol{y} = \{y_1, y_2, \dots, y_N\}$, generated from a presumed prob. model

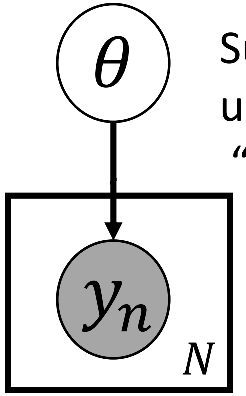
$$y_n \sim p(y|\theta) \qquad \forall n$$

(assumed independently & identically distributed (i.i.d.))

- Here $p(y|\theta)$ is a conditional distribution, conditioned on params $\boldsymbol{\theta}$ (to be learned)
  - Note: $\boldsymbol{\theta}$ may be fixed unknown or an unknown random variable (we will study both cases)



The parameters $\boldsymbol{\theta}$ may themselves depend on other unknown/known parameters (called hyperparameters), which may depend on other unknowns, and so on. ☺ This is essentially "hierarchical" modeling (will see various examples later)



Such diagrams are usually called the "plate notation"

The Predictive dist. tells us how likely each possible value of a new observation $\boldsymbol{y}_*$ is. Example: if $\boldsymbol{y}_*$ denotes the outcome of a coin toss, then what is $p(\boldsymbol{y}_* = \text{"}\boldsymbol{head}\text{"}|\boldsymbol{y})$, given $\boldsymbol{N}$ previous coin tosses $\boldsymbol{y} = \{y_1, y_2, \dots, y_N\}$

- Some of the tasks that we may be interested in
  - Parameter estimation: Estimating the unknown parameters $\boldsymbol{\theta}$ (and other unknowns $\boldsymbol{\theta}$ depends on)
  - Prediction: Estimating the **predictive distribution** of new data, i.e., $p(\boldsymbol{y}_*|\boldsymbol{y})$ - this is also a conditional distribution (conditioned on past data $\boldsymbol{y} = \{y_1, y_2, \dots, y_N\}$, as well as $\boldsymbol{\theta}$ and other things)

Parameter Estimation in Probabilistic Models

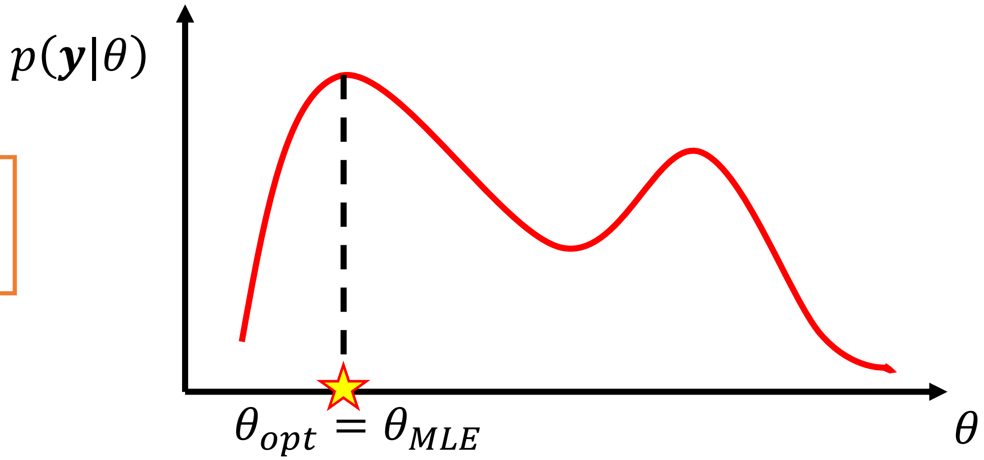
- Since data is assumed to be i.i.d., we can write down its total probability as

$$p(\boldsymbol{y}|\theta) = p(y_1, y_2, \ldots, y_N|\theta) = \prod_{n=1}^{N} p(y_n|\theta)$$

This now is an optimization problem essentially ($\boldsymbol{\theta}$ being the unknown)

- $p(\boldsymbol{y}|\theta)$ called "likelihood" - probability of observed data as a function of params $\boldsymbol{\theta}$

How do I find the best $\boldsymbol{\theta}$?

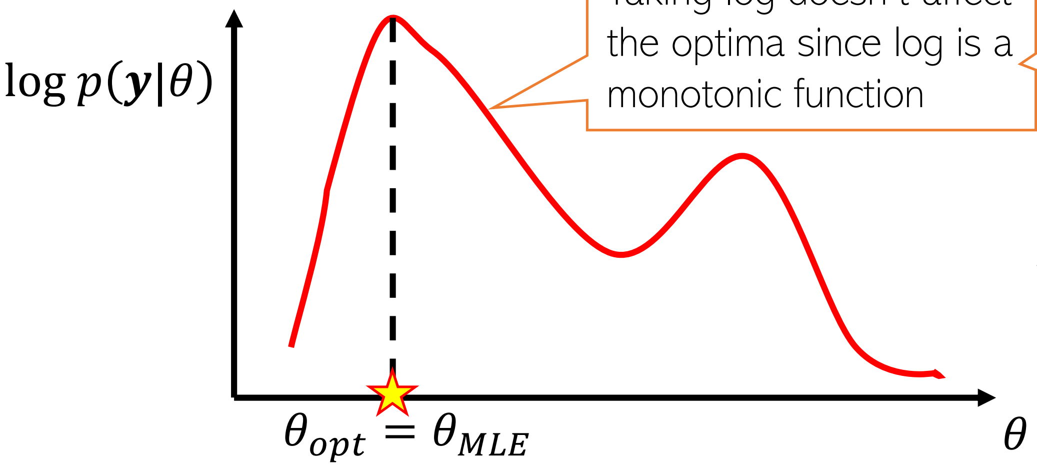


Well, one option is to find the $\boldsymbol{\theta}$ that maximizes the likelihood (probability of the observed data) – basically, which value of $\boldsymbol{\theta}$ makes the observed data most likely to have come from the assumed distribution $p(\boldsymbol{y}|\boldsymbol{\theta})$ --- Maximum Likelihood Estimation (MLE)

- In parameter estimation, the goal is to find the "best" $\boldsymbol{\theta}$, given observed data $\boldsymbol{y}$
- Note: Instead of finding single best, sometimes may be more informative to learn a distribution for $\boldsymbol{\theta}$ (can tell us about uncertainty in our estimate of $\boldsymbol{\theta}$ – more later)

Maximum Likelihood Estimation (MLE)

- The goal in MLE is to find the optimal $\boldsymbol{\theta}$ by maximizing the likelihood
- In practice, we maximize the log of the likelihood (log-likelihood in short)



Taking log doesn't affect the optima since log is a monotonic function

Leads to simpler algebra/calculus, and also yields better numerical stability when implementing it om computer (dealing with log of probabilities)

$$LL(\theta) = \log p(\boldsymbol{y}|\theta) = \log \prod_{n=1}^{N} p(y_n|\theta)$$

$$= \sum_{n=1}^{N} \log p(y_n|\theta)$$

- Thus the MLE problem is

$$\boxed{\theta_{MLE} = \underset{\theta}{\operatorname{argmax}}\, LL(\theta) = \underset{\theta}{\operatorname{argmax}} \sum_{n=1}^{N} \log p(y_n|\theta)}$$

- This is now an optimization (maximization problem). Note: $\boldsymbol{\theta}$ may have constraints

Maximum Likelihood Estimation (MLE)

Negative Log-Likelihood (NLL)

- The MLE problem can also be easily written as a <u>minimization</u> problem

$$\theta_{MLE} = \underset{\theta}{\operatorname{argmax}} \sum_{n=1}^{N} \log p(y_n|\theta) = \underset{\theta}{\operatorname{argmin}} \sum_{n=1}^{N} -\log p(y_n|\theta)$$

- Thus MLE can also be seen as minimizing the negative log-likelihood (NLL)

$$\boxed{\theta_{MLE} = \underset{\theta}{\operatorname{argmin}}\, NLL(\theta)}$$

Indeed. It may overfit. Several ways to prevent it: Use regularizer or other strategies to prevent overfitting. Alternatives, use "prior" distributions on the parameters $\boldsymbol{\theta}$ that we are trying to estimate (which will kind of act as a regularizer as we will see shortly)

- NLL is analogous to a loss function
  - The negative log-lik $(-\log p(y_n|\theta))$ is akin to the loss on each data point

Such priors have various other benefits as we will see later

- Thus doing MLE is akin to <u>minimizing training loss</u>

Does it mean MLE could overfit? If so, how to prevent this?

MLE: An Example

- Consider a sequence of N coin toss outcomes (observations)
- Each observation y_n is a binary random variable. Head: $y_n = 1$, Tail: $y_n = 0$
- Each y_n is assumed generated by a **Bernoulli distribution** with param $\theta \in (0,1)$

Probability of a head

$$p(y_n|\theta) = \text{Bernoulli}(y_n|\theta) = \theta^{y_n}(1-\theta)^{1-y_n}$$

- Here θ the unknown param (probability of head). Want to estimate it using MLE
- Log-likelihood: $\sum_{n=1}^{N} \log p(y_n|\theta) = \sum_{n=1}^{N} [y_n \log\theta + (1-y_n)\log(1-\theta)]$
- Maximizing log-lik (or minimizing NLL) w.r.t. θ will give a closed form expression

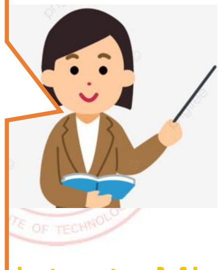
Take deriv. set it to zero and solve. Easy optimization

I tossed a coin 5 times – gave 1 head and 4 tails. Does it means θ = 0.2?? The MLE approach says so. What is I see 0 head and 5 tails. Does it mean θ = 0?

$$\theta_{MLE} = \frac{\sum_{n=1}^{N} y_n}{N}$$

Thus MLE solution is simply the fraction of heads! ☺ Makes intuitive sense!

Indeed – if you want to trust MLE solution. But with small number of training observations, MLE may overfit and may not be reliable. We will soon see better alternatives that use prior distributions!



Coming up next

- Prior distributions and their role in parameter estimation
  - Maximum-a-Posteriori (MAP) Estimation
  - Fully Bayesian inference
- Probabilistic modeling for regression and classification problems