CIS 5300: NATURAL LANGUAGE PROCESSING

# Review of probabilities

Professor Chris Callison-Burch
Dr. Kuzman Ganchev

# What is a probability distribution?

- A mathematical object that we use to model an event in the world.
- Assigns a number to possible outcomes
- Example model:

Event: Coin flip

Outcome space: {heads, tails}

P(heads) = ½

P(tails) = ½

# Notation for coin flips.

- A random variable  is a variable that takes on a value according to some probability distribution.
- Assigns a number to possible outcomes
- Probabilities are:
  - Non-negative
  - Sum to 1
- Our probability models will have some parameters

# Notation for coin flips

Random variable c denotes a coin-flip:

- Event space: c ∈ {heads, tails}
- P(c = heads) = h
- P(c = tails) = 1 - h

- Event space: c ∈ {heads, tails, other}
- P(c = heads) = 0.6
- P(c = tails) = 0.4

This is an unfair coin.

# Example probability distributions

Dice roll

- Random variable d.
- Outcome space: {1 .. 6}
- Parameters $\boldsymbol{\theta}$
- $P(d = i) = \theta_i$

A fair dice would have $\boldsymbol{\theta} = [1/6, 1/6, 1/6, 1/6, 1/6, 1/6]$

# Example probability distribution

Roll a fair dice to pick a word:

| 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|
| "Sam | laughs | last | and | laughs | loudest" |

Define random variables:

- f = first letter of the chosen word
  - Outcome space: {S, l, a}
- s = second letter of the chosen word
  - Outcome space {a, n, o}

# Simple example: picking a letter

Roll a fair dice to pick a word:

1 2 3 4 5 6

"Sam laughs last and laughs loudest"

- f = first letter of the chosen word
- s = second letter of the chosen word

- E.g. P(f = l) = 4 / 6
- E.g. P(s = a) = 4 / 6

Sam laughs last and laughs loudest

Sam laughs last and laughs loudest

# Relationships between distributions

Roll a fair dice to pick a word:

1 2 3 4 5 6

"Sam laughs last and laughs loudest"

- f = first letter of the chosen word
- s = second letter of the chosen word

**Joint probability** p(f, s) distribution over both random variables at the same time.

- Outcome space: {(S, a), (S, n), (S, o), … (l, a), (l, n), (l, o)}

**Conditional probability** p(s |f) distribution of s given a particular value of f.

- Outcome space is same as p(s).

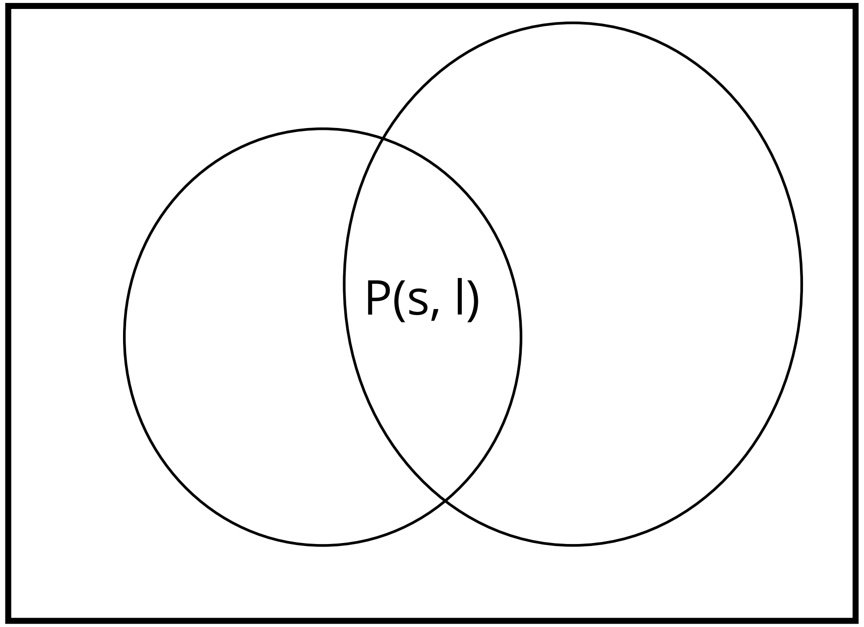
# Conditional probability

Conditional probability:

- P(s = a | f = l)

Is the amount of probability in the event f=l that is also shared with the event s=a.

- P(s = a | f = l) = P(s = a, f = l) / P(f = l)




Bayes Rule

- P(c | d) = P(d | c) * P(c) / P(d)

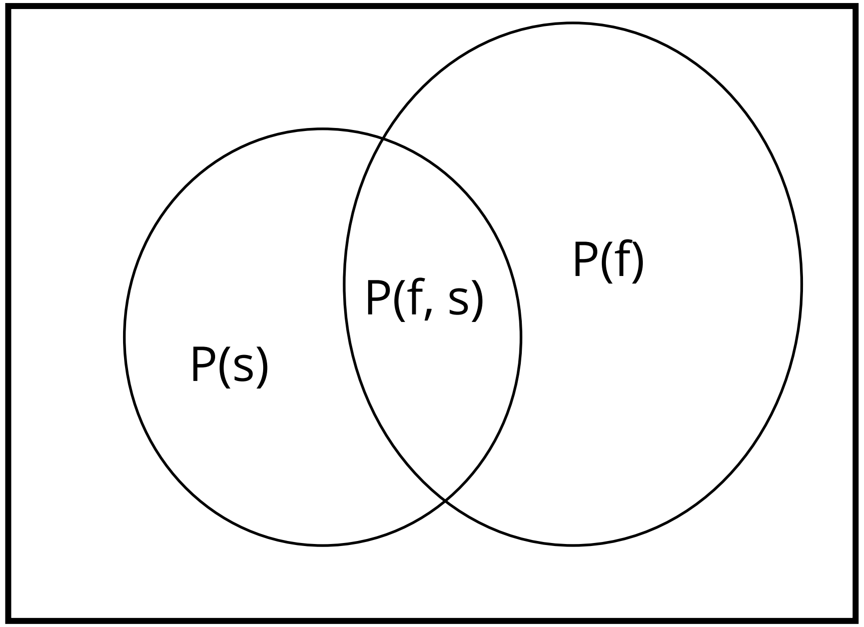
# Conditional probability

Conditional probability:

- P(s = a | f = l)

Is the amount of probability in the event f=l that is also shared with the event s=a.

- P(s = a | f = l) = P(s = a, f = l) / P(f = l)




Bayes Rule

- P(s | f) = P(f | s) * P(s) / P(f)

# Simple example: picking a letter

Roll a fair dice to pick a word:

"Sam laughs last and laughs loudest"

(1 Sam, 2 laughs, 3 last, 4 and, 5 laughs, 6 loudest)

- f = first letter of the chosen word
- s = second letter of the chosen word

- E.g. P(f = l) = 4 / 6
- E.g. P(s = a) = 4 / 6
- Joint P(f = l, s = a) = 3 / 6
- Conditional P(s = a | f = l) = 3 / 4

Sam laughs last and laughs loudest

Sam laughs last and laughs loudest

Sam laughs last and laughs loudest

laughs last laughs loudest

CIS 5300: NATURAL LANGUAGE PROCESSING

# Estimating model parameters


Professor Chris Callison-Burch

Dr. Kuzman Ganchev


# Probabilistic models in practice

Usual scenario:

- Gather some data.
- Define a probabilistic model.
- Use the data to estimate the parameters of the model.

Example:

- We flip a coin *n* times, and collect the observations: [heads, tails, tails, …]
- Model: each flip is independent, with probability p(heads) = h.
- This video: how do we select p(h)?

# Some more terminology

- The data we collect are called a **sample**.
- The procedure we use to choose model parameters is called an **estimator**.
- The **data likelihood** is the probability of the data under the model's distribution.
  E.g.
  - Data: [heads, heads, tails]
  - Model: p(heads) = h = 0.7
  - Likelihood = 0.7 * 0.7 * 0.3

  Usually look at log-likelihood (the natural log of the likelihood).

# Data likelihood

A couple more examples:

Data: D = [heads, heads, tails]

P(heads) = 0.5 ⇨ likelihood(D) = 0.5 * 0.5 * 0.5 = 0.125

P(heads) = 0.8 ⇨ likelihood(D) = 0.8 * 0.8 * 0.2 = 0.128

P(heads) = 0.6 ⇨ likelihood(D) = 0.6 * 0.6 * 0.4 = 0.144

Question: what P(h) gives the highest likelihood?

Choosing P(h) this way is called the maximum likelihood estimator.

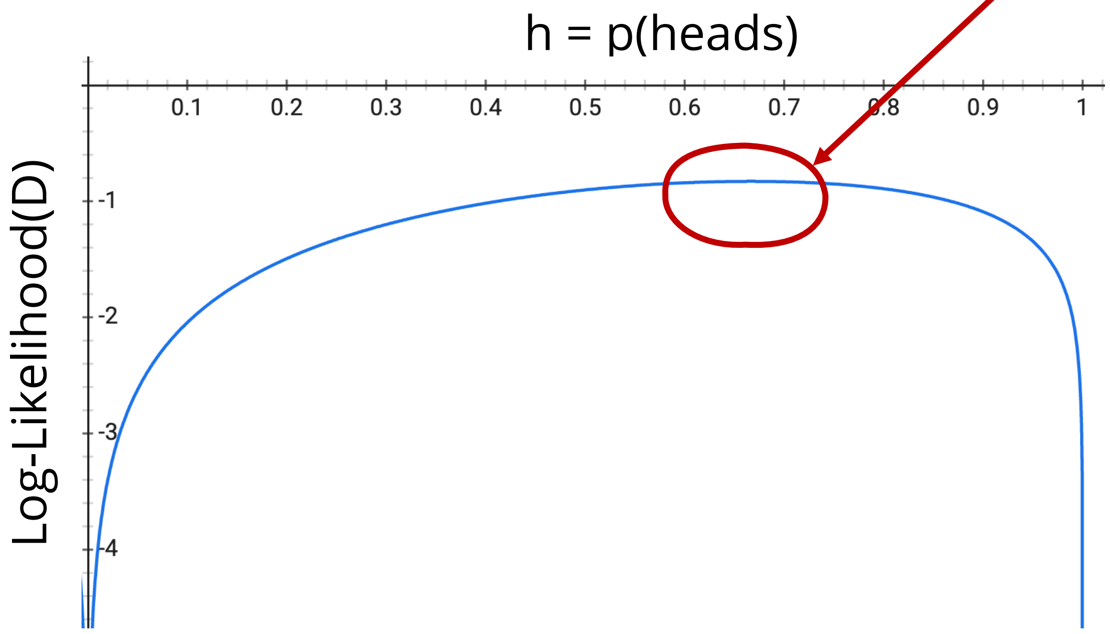
# Data likelihood

What is the highest log-likelihood?

Data: D = [heads, heads, tails]

Log-likelihood = ln(h * h * (1 - h))

Looks like max is somewhere here

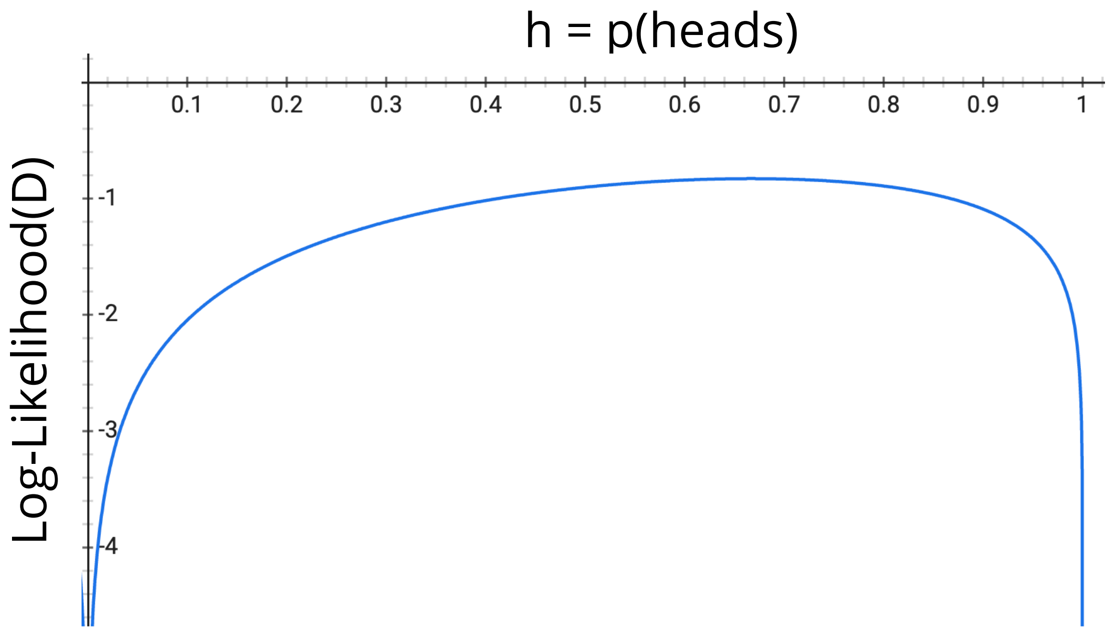


# How to find the max of a function

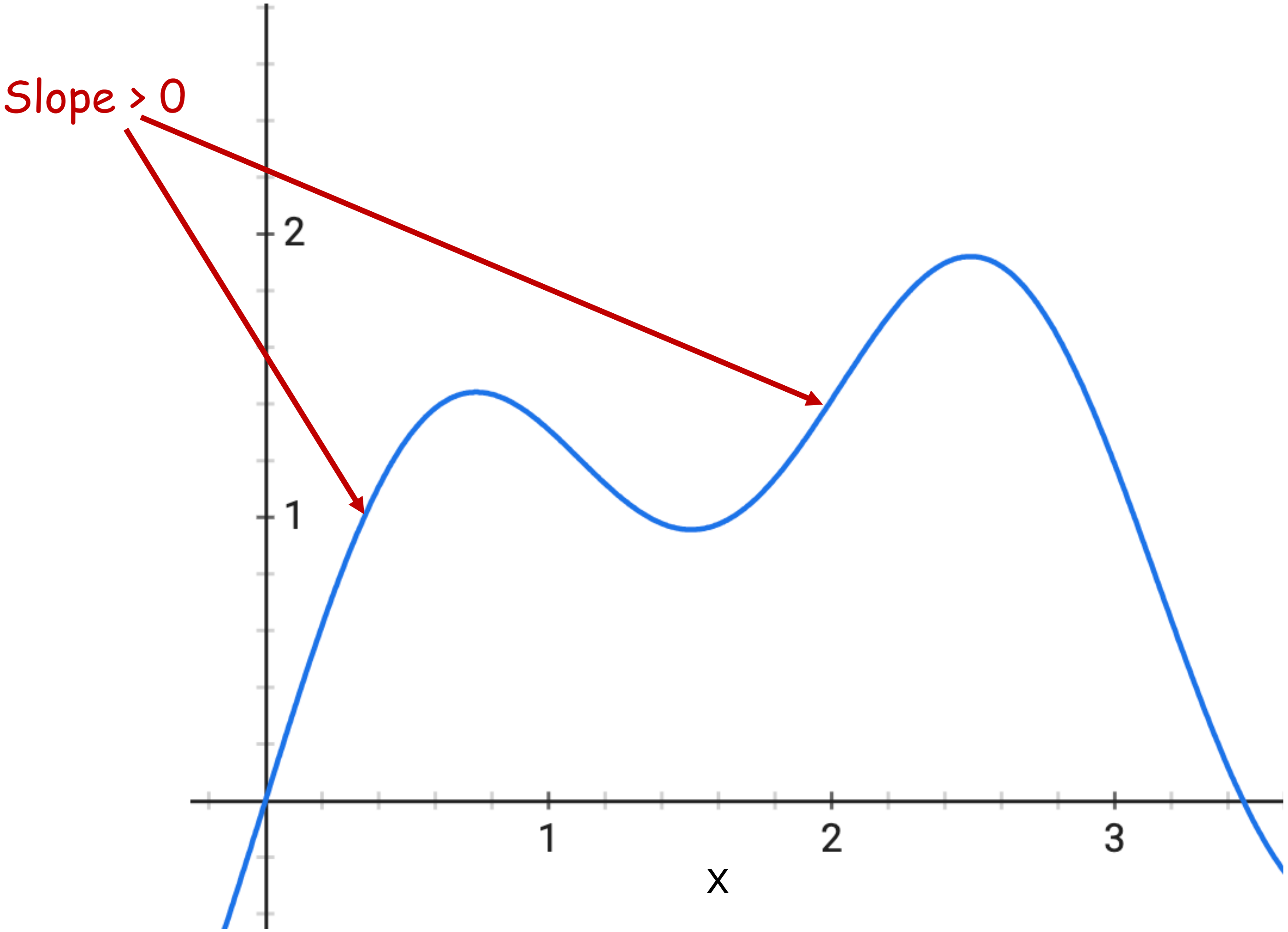
We are looking for the very top of the curve.

The top is always flat.

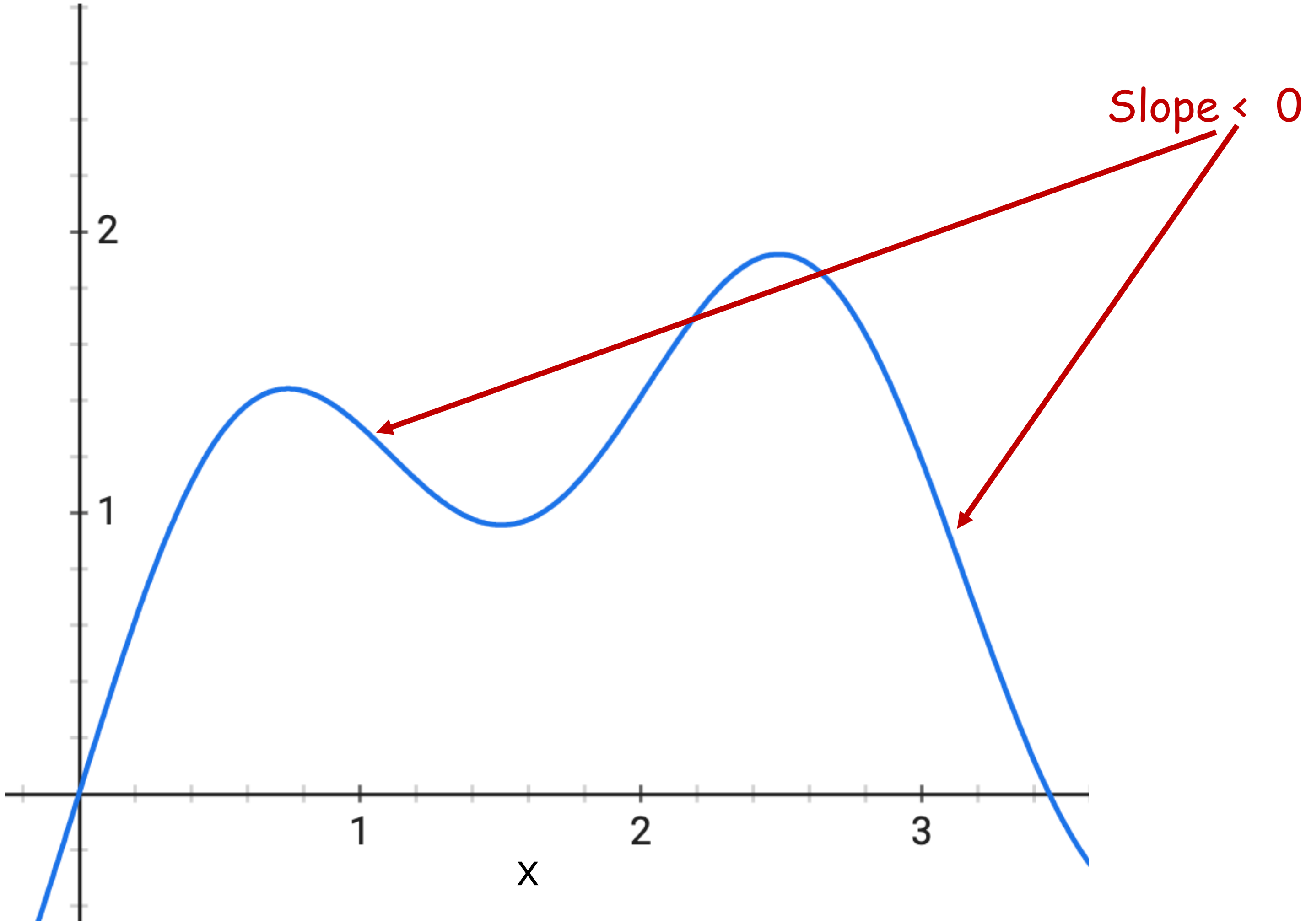
More formally, the slope is zero: derivative of the function is zero.



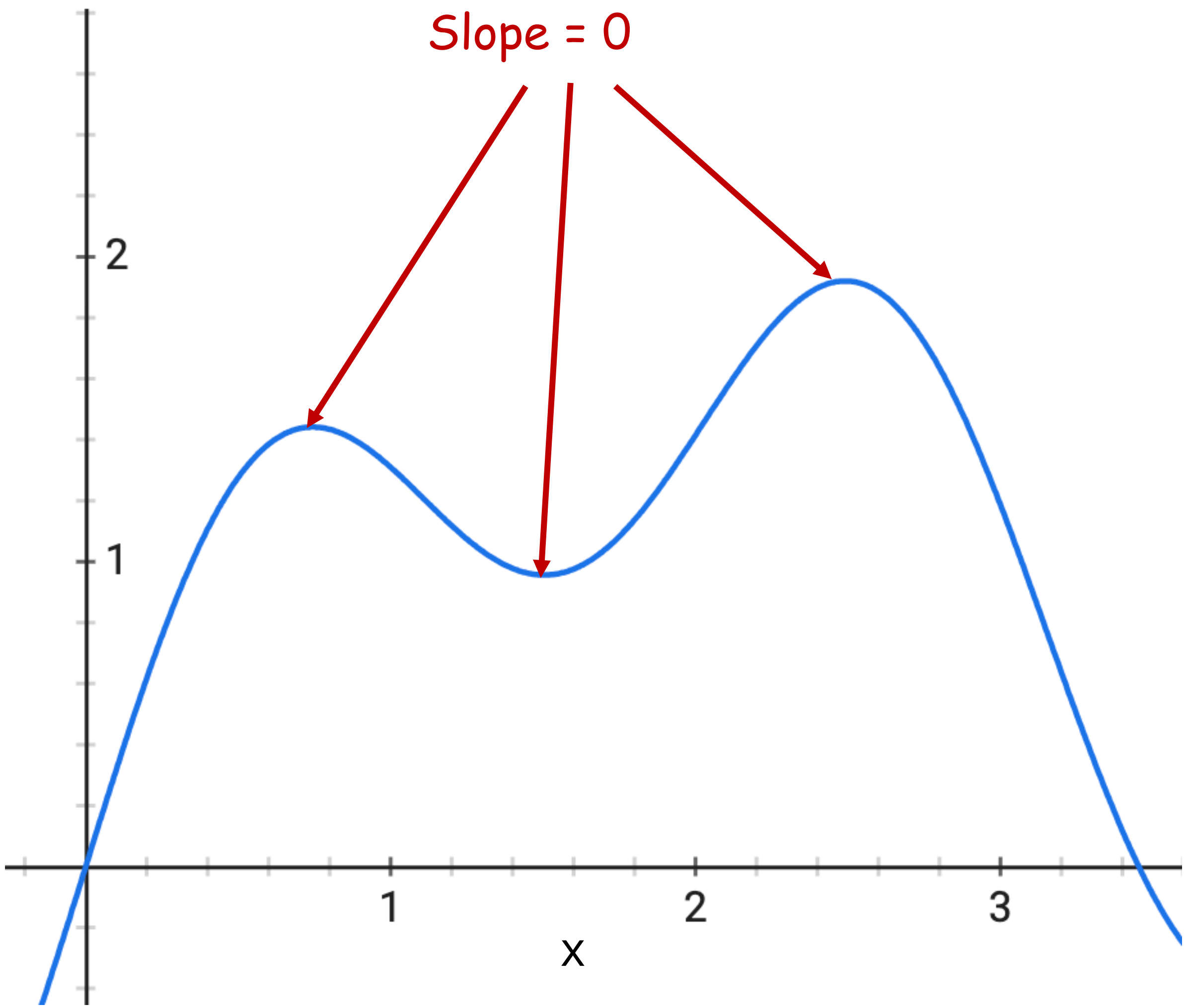
# Review of calculus



# Review of calculus



# Review of calculus



# Review of calculus

**Derivative notation:**

Partial derivative of f(x) with respect to x:

- $\partial f(x) / \partial x$

**Derivative of a logarithm**

Partial derivative of ln(x) with respect to x equals 1/x

- $\partial \ln(x) / \partial x = 1 / x$

We will also use the chain rule of calculus.

# Maximum likelihood for our example

Data: [heads, heads, tails]

$\mathcal{L}$ = Log-likelihood(Data)

$\mathcal{L} = \ln(h * h * (1 - h))$

$\mathcal{L} = 2 \ln(h) + \ln(1 - h)$

Taking the derivative:

$\partial\mathcal{L} / \partial h = 2 * 1/h + (-1) / (1-h)$

# Maximum likelihood for our example

Data: [heads, heads, tails]

Log-likelihood(Data) : $\mathcal{L} = 2 \ln(h) + \ln(1 - h)$

Derivative: $\partial\mathcal{L} / \partial h = 2 / h - 1 / (1-h)$

Setting the derivative to zero:

$2 / h - 1 / (1-h) = 0$

$2 / h = 1 / (1-h)$

$2 – 2h = h$

$2 = 3h \quad \Rightarrow \quad h = 2 / 3$

# Multinomial distribution

Multinomial distribution:

- Distribution over some discrete outcomes.
  - E.g. coin flip; dice; letters of the alphabet, words in the dictionary, etc.
- Parameters:
  - Probability of outcome i: $p(X=i) = \pi_i$
  - Where i ranges from 1 to k.
- Remember they are probabilities!
  - $\pi_i \geq 0$
  - $\sum \pi_i = 1$

# Maximum likelihood for multinomials

Data: [o1, o2, o3, …, on]

Log-Likelihood = $\ln(\pi_{o1}) + \ln(\pi_{o2}) + \ln(\pi_{o3}) + \ldots + \ln(\pi_{on})$

Log-Likelihood = $c_1 \ln(\pi_1) + c_2 \ln(\pi_2) + \ldots + c_k \ln(\pi_k)$

- Where $c_i$ counts how many times we see i in the data.

$$\mathcal{L} = \Sigma_i \, c_i \ln(\pi_i)$$

Optimization problem:

$$\max \Sigma_i \, c_i \ln(\pi_i)$$

Why doesn't this work?

# Maximum likelihood for multinomials

Data: [o1, o2, o3, ..., on]  ⇨ $c_i$ are the counts

Log-Likelihood: $\mathcal{L} = \Sigma_i \, c_i \ln(\pi_i)$

Optimization problem:

$$\max \Sigma_i \, c_i \ln(\pi_i)$$

The larger π, the larger $\mathcal{L}$.

So we can get $\mathcal{L}$ arbitrarily large by setting π arbitrarily high.

But these are probabilites!

# Maximum likelihood for multinomials

Data: [o1, o2, o3, ..., on] ⇨ $c_i$ are the counts

Log-Likelihood: $\mathcal{L}(\pi) = \Sigma_i\, c_i \ln(\pi_i)$

Optimization problem:

$$\max \Sigma_i\, c_i \ln(\pi_i) \quad \text{such that} \quad \Sigma\, \pi_i = 1$$

Make the constraint into a game:

$$\max_\pi \; \min_\lambda \; \Sigma_i\, c_i \ln(\pi_i) + \lambda\, (\Sigma\, \pi_i - 1)$$

Now if we choose π to not satisfy the constraint, we get get a very bad objective value.

# Maximum likelihood for multinomials

Data: [o1, o2, o3, ..., on] ⇨ $c_i$ are the counts

$\mathcal{L}(\pi, \lambda) = \Sigma_i\, c_i \ln(\pi_i) + \lambda\, (\Sigma\, \pi_i - 1)$

$\partial\mathcal{L} / \partial\, \pi_i = c_i / \pi_i + \lambda$

Setting derivative to zero:

$c_i / \pi_i + \lambda = 0$

$\pi_i = c_i / \lambda$

What about $\lambda$? How can we compute it?

$\lambda = \Sigma\, \pi_i$

# Note: it is not always easy to compute max likelihood

- Sometimes, we do not have a closed-form solution for maximum likelihood.
- We do not always observe everything we would like to.

# Complicated example

Roll a dice to pick a word:

1 2 3 4 5 6

"Sam laughs last and laughs loudest"

Data:

- f = first letter of the chosen word
- s = second letter of the chosen word
- F = [l, l, a, ...]; S = [a, a, n, ...]

Model:

- Probability distribution over [1, 2, 3, 4, 5, 6].

# Problem with small samples

Suppose we flip a coin just once.

Data: [heads]

Max likelihood estimate: p(heads) = 1; p(tails) = 0.

Is this a good model of the world?

# Add-1 smoothing

One way to overcome this, is to add 1, or ½ or something else to our counts.

Data = [heads]; counts = {heads: 2, tails: 1}

This turns out to be the same as having a prior belief about what the coin probability is.

This is a probability distribution over the model parameters.

Priors and other forms of regularization are very important for most models.