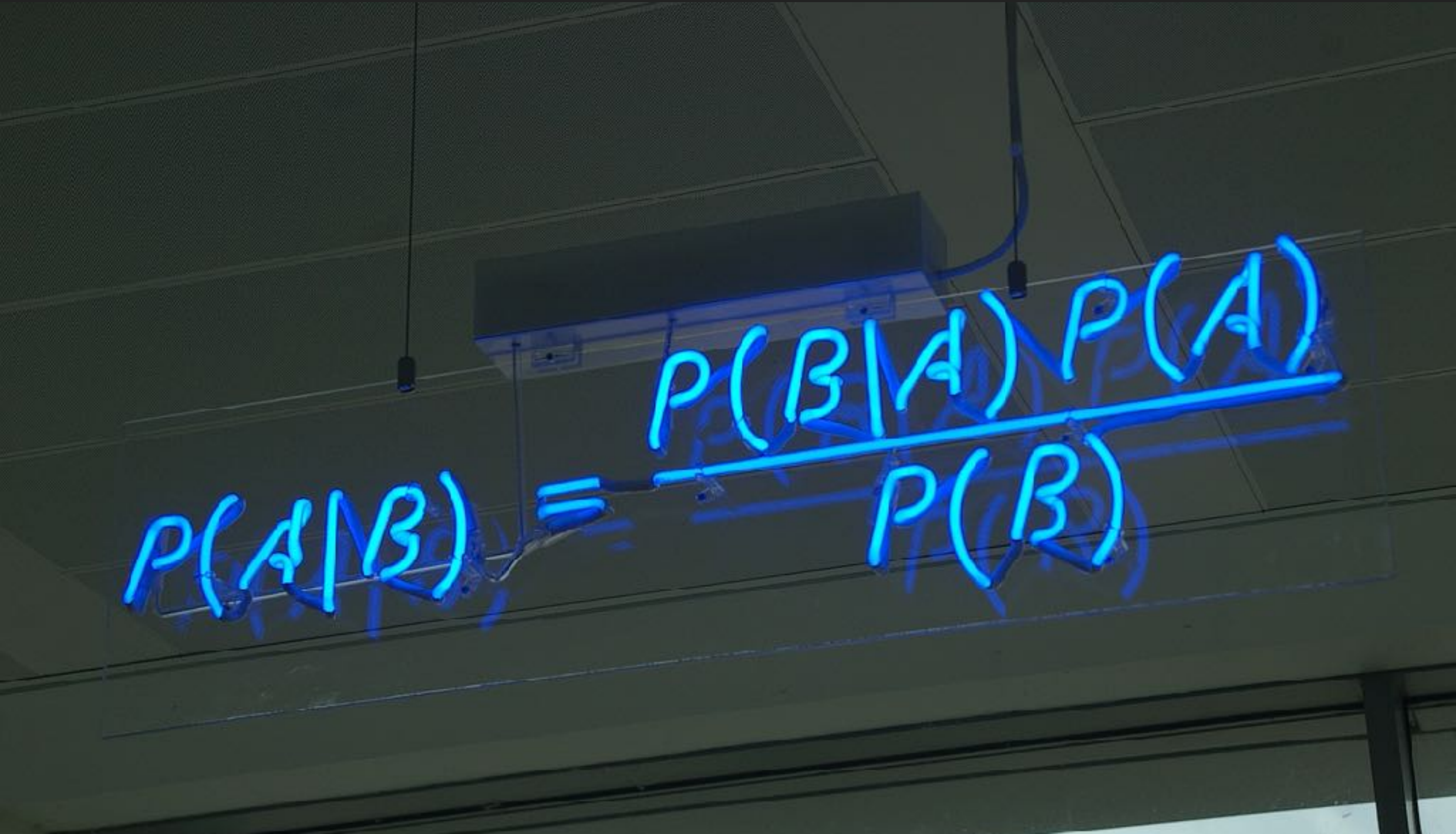
ASTC02 - PROF. HANNO REIN

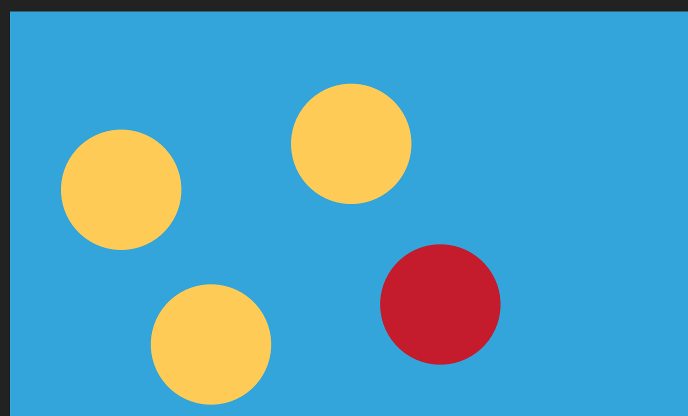
BAYESIAN STATISTICS

BAYES'S THEOREM

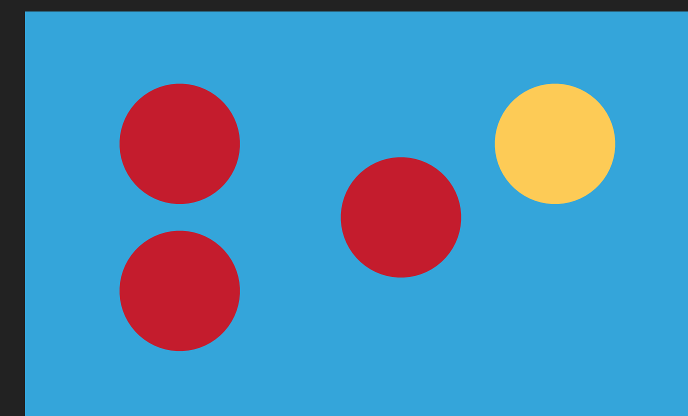


COOKIE PROBLEM

Two bowls of cookies.



30 vanilla
10 chocolate



10 vanilla
30 chocolate

Choose a bowl randomly, then choose a cookie randomly. It’s vanilla. What is the probability it came from bowl 1?

COOKIE PROBLEM

This is a hard question! Asking a different question is easy:

What is the probability of a vanilla cookie in bowl 1?

$$p(\text{vanilla}|\text{bowl 1}) = 3/4$$

$$p(\text{bowl 1}|\text{vanilla}) = ?$$

COOKIE PROBLEM

Let's call B_1 the hypothesis that the cookie came from bowl 1 and V for vanilla. Bayes's theorem gives us:

$$p(B_1|V) = \frac{p(B_1)p(V|B1)}{p(V)}$$

COOKIE PROBLEM

$$p(B_1|V) = \frac{p(B_1)p(V|B1)}{p(V)}$$

$$p(B_1) = \frac{1}{2}$$

$$p(V|B_1) = \frac{3}{4}$$

$$p(V) = \frac{1}{2}$$

$$p(B_1|V) = \frac{\frac{1}{2}\frac{3}{4}}{\frac{1}{2}} = \frac{3}{4}$$

DIACHRONIC INTERPRETATION

One way of thinking about Bayes's theorem. Suppose we have a hypothesis H and some data D.

$$p(H|D) = \frac{p(H)\; p(D|H)}{p(D)}$$

These terms now have names and can be interpreted as follows.

DIACHRONIC INTERPRETATION

$$p(H|D) = \frac{p(H)\;p(D|H)}{p(D)}$$

$p(H)$ Prior

$p(H|D)$ Posterior

$p(D|H)$ Likelihood

$p(D)$ Normalization constant

LINE FITTING

$p(D|H)$ Likelihood

One data point:

$$p(y_i|x_i, \sigma_{yi}, m.b) = \frac{1}{\sqrt{2\pi\sigma_{yi}^2}} \exp\left(-\frac{(y_i - mx_i - b)^2}{2\sigma_{yi}^2}\right)$$

Multiple data points:

$$\mathcal{L} = \prod_{i=1}^{N} p(y_i|x_i, \sigma_{yi}, m, b)$$

LINE FITTING

Multiple data points:

$$\mathcal{L} = \prod_{i=1}^{N} p(y_i | x_i, \sigma_{yi}, m, b)$$

Log likelihood:

$$\ln \mathcal{L} = K - \sum_{i=0}^{N} \frac{(y_i - m x_i - b)^2}{2\sigma_{yi}^2}$$

$$= K - \frac{1}{2}\chi^2$$

BAYES THEOREM

$$p(m, b|\left\{y_i\right\}_{i=1}^N, I) = \frac{p(\left\{y_i\right\}_{i=1}^N|m, b, I)\; p(m, b|I)}{p(\left\{y_i\right\}_{I=1}^N|I)}$$

I Short hand for all period knowledge

$\left\{y_i\right\}_{i=1}^N$ Short hand for all data

BAYES THEOREM

$$p(m,b|\left\{y_i\right\}_{i=1}^{N},I) = \frac{p(\left\{y_i\right\}_{i=1}^{N}|m,b,I)\;p(m,b|I)}{p(\left\{y_i\right\}_{I=1}^{N}|I)}$$

We want to know this!

It’s a distribution!

We can *sample* the distribution!

BAYES THEOREM

$$p(m, b|\left\{y_i\right\}_{i=1}^N, I) = \frac{p(\left\{y_i\right\}_{i=1}^N|m, b, I)\; p(m, b|I)}{p(\left\{y_i\right\}_{I=1}^N|I)}$$

Let's ignore the normalization constant (does not depend on m or b)

METROPOLIS HASTINGS ALGORITHM

1) Start at a random place (ideally close to a sensible value)
2) Predict random new step
3) a) Choose new step if it's better
   b) 'Sometimes' choose even if it's worse
4) Keep track of path. Path is the posterior distribution!

METROPOLIS HASTINGS ALGORITHM

We need:

1) Likelihood function
2) MH algorithm
3) Starting point

METROPOLIS HASTINGS ALGORITHM

```
def lnlike(state):
    m, b = state
    s = 0.
    for i in range(N):
        s -= (y[i]-m*x[i]-b)**2/(2.*sigma[i]**2)
    return s
```

METROPOLIS HASTINGS ALGORITHM

BUG! (EXP IS MISSING)

```
samples = []
state = np.array([2.,34.])  # m, b
lnlike_state = lnlike(state)
while len(samples)<10000:
    eps = 0.01
    state_n = state + eps*np.random.normal(size=2)
    lnlike_state_n = lnlike(state_n)
    if np.random.rand() < lnlike_state_n - lnlike_state:
        state = state_n
        lnlike_state = lnlike_state_n
    samples.append(state_n.copy())
```

METROPOLIS HASTINGS ALGORITHM

```
m,b = np.average(samples,axis=0)
print(m,b)
m,b = np.max(samples,axis=0)
print(m,b)
```

```
2.23711063552 34.0430047992
2.27749746482 34.090323576
```

```
import corner
figure = corner.corner(samples)
```

```
fig, ax = plt.subplots(1,1)
ax.errorbar(data[:,0],data[:,1],data[:,2],fmt="o")
xs = np.linspace(20,300,1000)
for k in range(10):
    ki = np.random.randint(len(samples))
    ax.plot(xs,samples[ki][0]*xs+samples[ki][1],color="gray",alpha=0.1)
ax.plot(xs,m*xs+b);
```