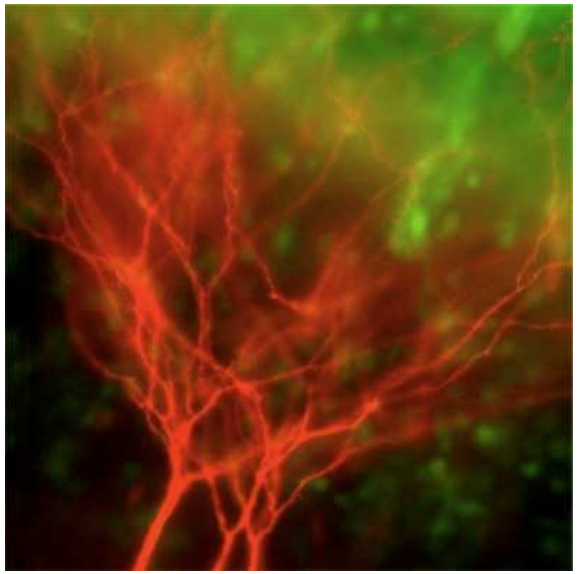
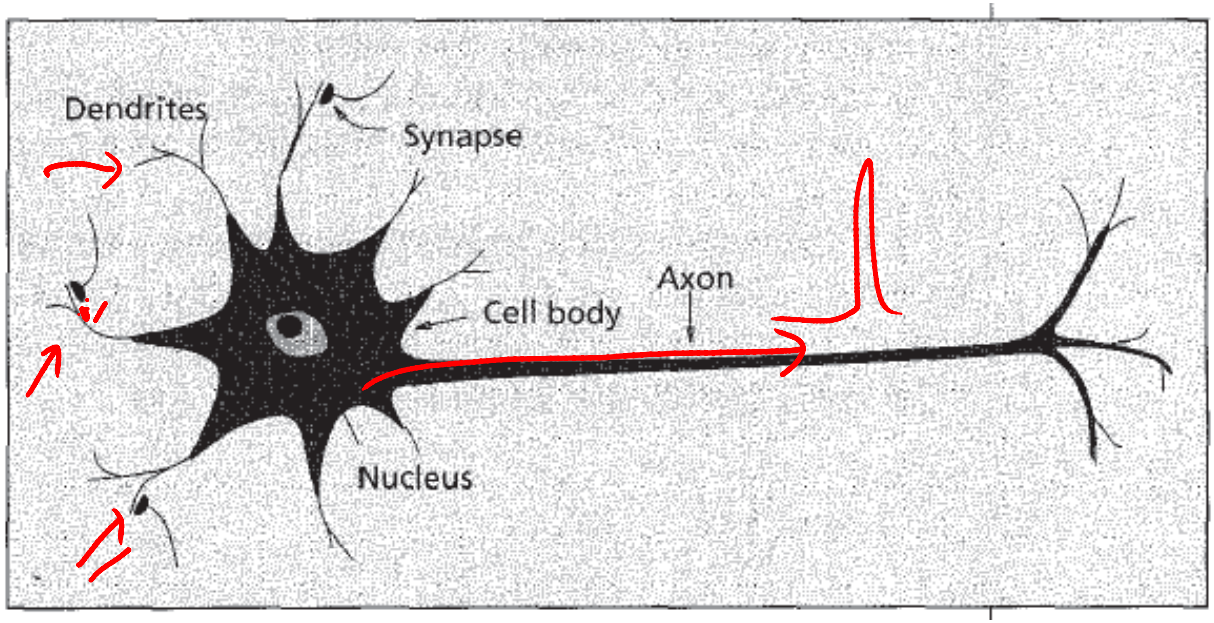
Outline of the lecture

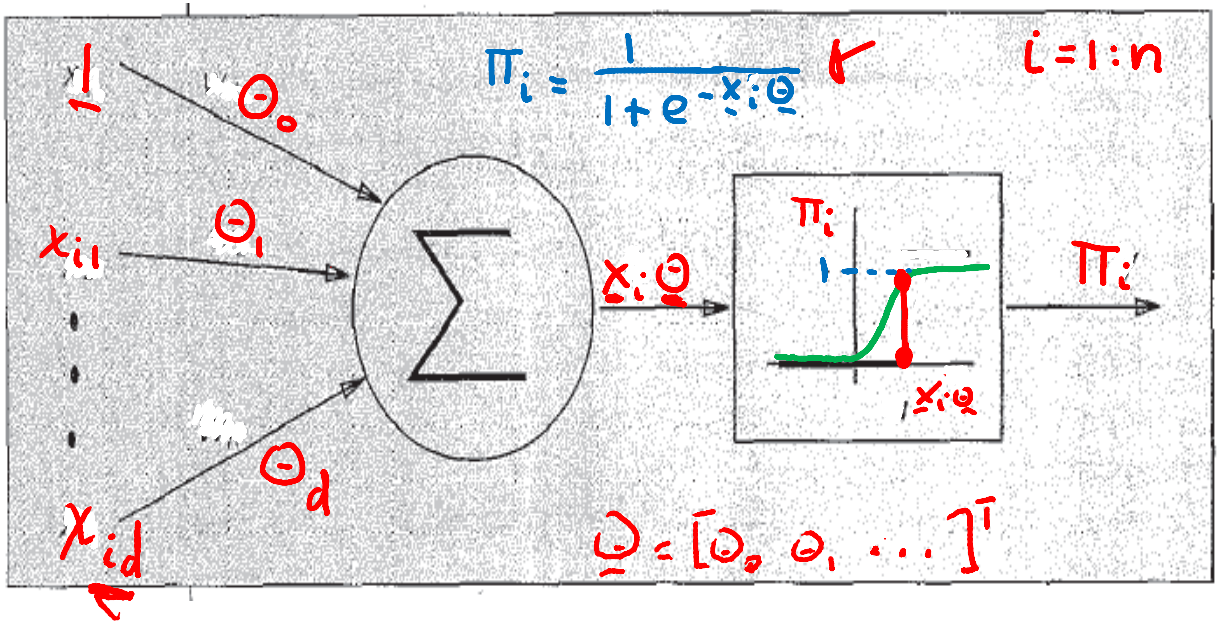
This lecture describes the construction of binary classifiers using a technique called Logistic Regression. The objective is for you to learn:

- ❑ How to apply logistic regression to discriminate between two classes.
- ❑ How to formulate the logistic regression likelihood.
- ❑ How to derive the gradient and Hessian of logistic regression on your own.
- ❑ How to incorporate the gradient vector and Hessian matrix into Newton's optimization algorithm so as to come up with an algorithm for logistic regression, which we'll call IRLS.
- ❑ How to do Bayesian logistic regression with Monte Carlo and importance sampling.

McCulloch-Pitts model of a neuron



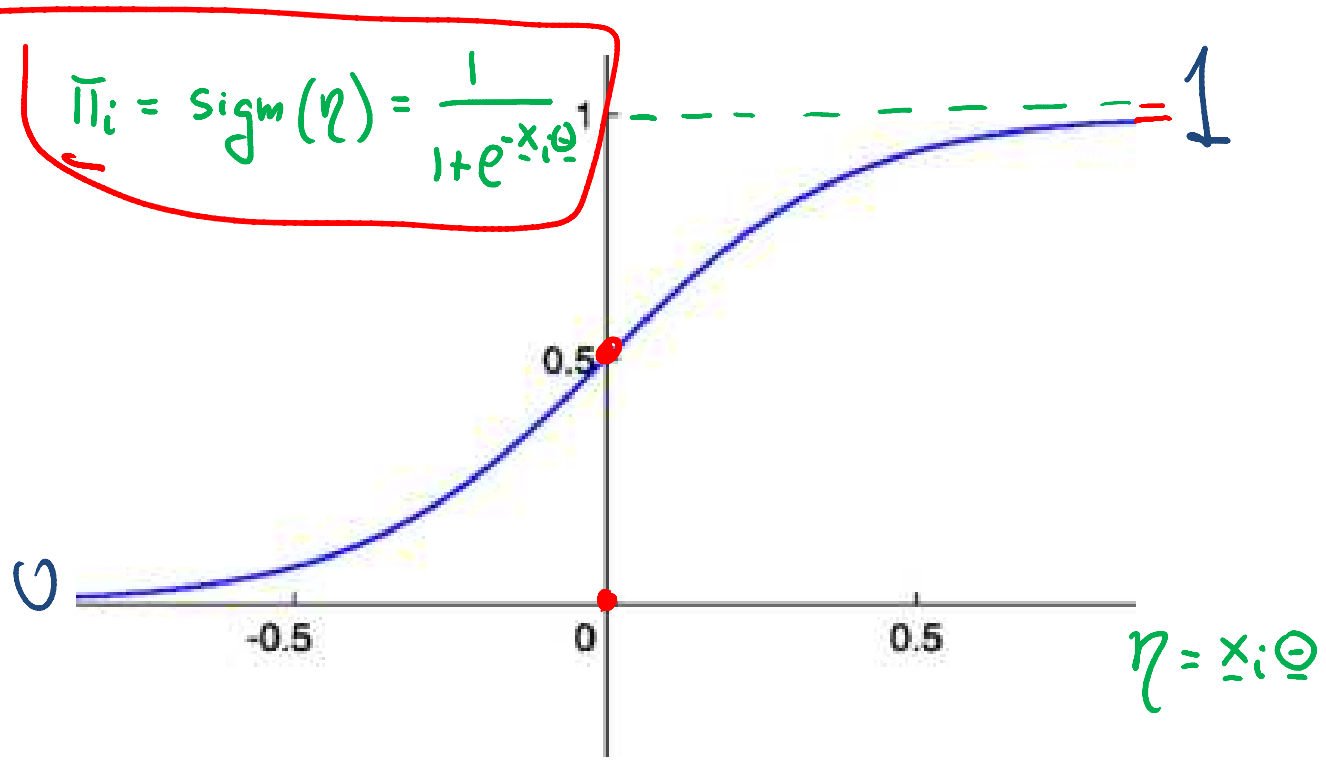




Sigmoid function

$\text{sigm}(\eta)$ refers to the **sigmoid** function, also known as the **logistic** or **logit** function:

$$\text{sigm}(\eta) = \frac{1}{1+e^{-\eta}} = \frac{e^{\eta}}{e^{\eta}+1}$$



Linear separating hyper-plane

$$P(Y_i = 1 \mid X_i, \Theta) = \text{sigm}(X_i \Theta) = \frac{1}{2} \quad \pi_i$$

when

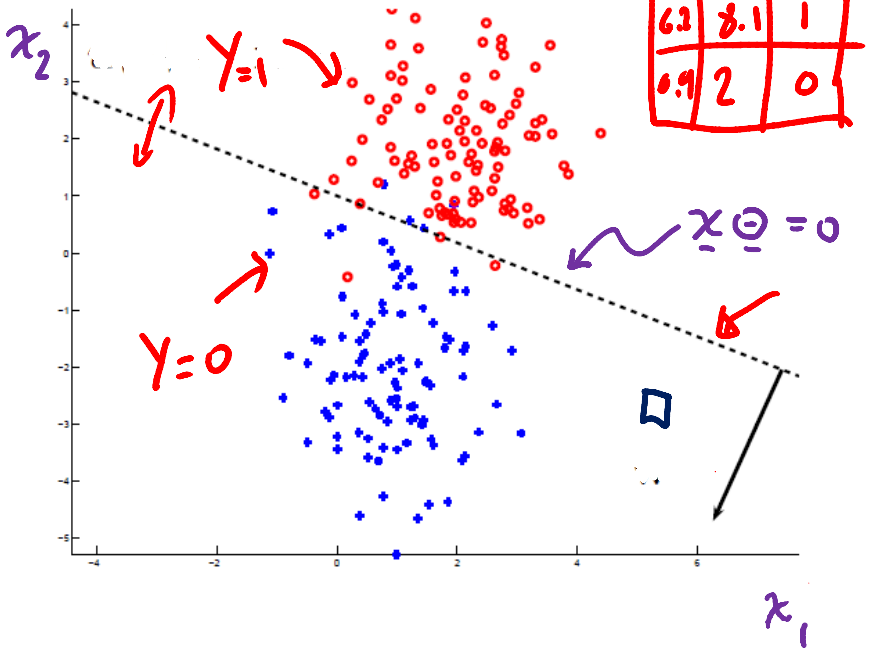
$X_i \Theta = 0$

EQUATION OF A PLANE.

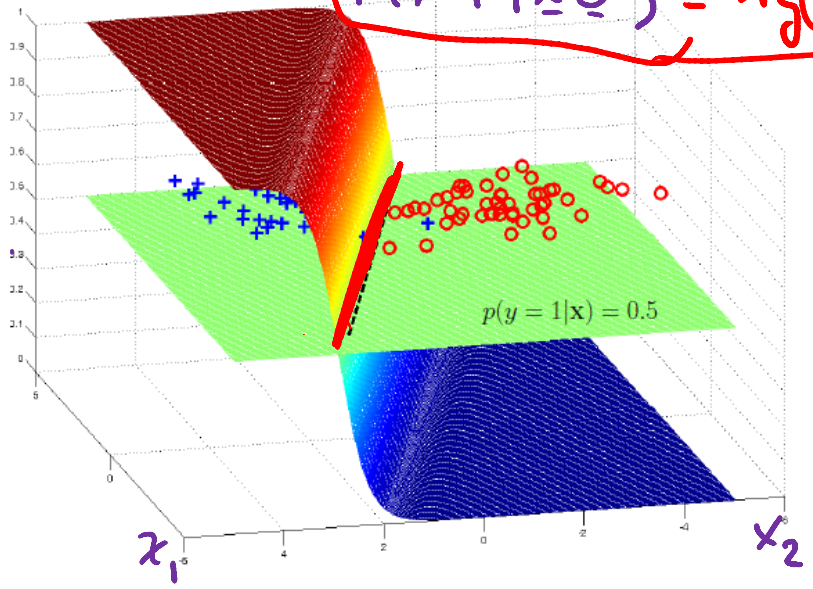
i.e. $\frac{1}{1+e^{-0}} = \frac{1}{2}$

Data D

x_1	x_2	y
1	2.3	0
6.2	8.1	1
0.1	2	0



$P(Y=1 \mid X\Theta) = \text{sig}(x_i \theta)$



[Greg Shakhnarovich]

Logistic regression

The logistic regression model specifies the probability of a binary output $y_i \in \{0, 1\}$ given the input $\mathbf{x}_i$ as follows:

$$p(\mathbf{y}|\mathbf{X}, \boldsymbol{\theta}) = \prod_{i=1}^{n} \text{Ber}(y_i|\text{sigm}(\mathbf{x}_i\boldsymbol{\theta})) = \prod_{i=1}^{n} \pi_i^{y_i}(1-\pi_i)^{1-y_i} = \begin{cases} \pi_i & y_i = 1 \\ 1-\pi_i & y_i = 0 \end{cases}$$

$$= \prod_{i=1}^{n} \left[\frac{1}{1+e^{-\mathbf{x}_i\boldsymbol{\theta}}}\right]^{y_i} \left[1 - \frac{1}{1+e^{-\mathbf{x}_i\boldsymbol{\theta}}}\right]^{1-y_i}$$

where $\mathbf{x}_i\boldsymbol{\theta} = \theta_0 + \sum_{j=1}^{d} \theta_j x_{ij}$

$$-\log P(y|X,\theta) = J(\theta) = \sum_{i=1}^{n} y_i \log \pi_i + (1-y_i)\log(1-\pi_i)$$

cross-entropy

Gradient and Hessian of binary logistic regression

The gradient and Hessian of the negative loglikelihood, $J(\boldsymbol{\theta}) = -\log p(\mathbf{y}|\mathbf{X}, \boldsymbol{\theta})$, are given by:

$$\begin{aligned} \mathbf{g}(\mathbf{w}) &= \frac{d}{d\boldsymbol{\theta}} J(\boldsymbol{\theta}) = \sum_{i=1}^{n} \mathbf{x}_i^T (\pi_i - y_i) = \mathbf{X}^T(\boldsymbol{\pi} - \mathbf{y}) \\ \mathbf{H} &= \frac{d}{d\boldsymbol{\theta}} \mathbf{g}(\boldsymbol{\theta})^T = \sum_i \pi_i(1-\pi_i)\mathbf{x}_i\mathbf{x}_i^T = \mathbf{X}^T \mathrm{diag}(\pi_i(1-\pi_i))\mathbf{X} \end{aligned}$$

where $\pi_i = \mathrm{sigm}(\mathbf{x}_i\boldsymbol{\theta})$

One can show that $\mathbf{H}$ is positive definite; hence the NLL is **convex** and has a unique global minimum.

To find this minimum, we turn to batch optimization.

Iteratively reweighted least squares (IRLS)

For binary logistic regression, recall that the gradient and Hessian of the negative log-likelihood are given by

$$\begin{aligned}
\mathbf{g}_k &= \mathbf{X}^T(\boldsymbol{\pi}_k - \mathbf{y}) \\
\mathbf{H}_k &= \mathbf{X}^T\mathbf{S}_k\mathbf{X} \\
\mathbf{S}_k &:= \text{diag}(\pi_{1k}(1-\pi_{1k}), \ldots, \pi_{nk}(1-\pi_{nk})) \\
\pi_{ik} &= \text{sigm}(\mathbf{x}_i\boldsymbol{\theta}_k)
\end{aligned}$$

The Newton update at iteration $k+1$ for this model is as follows (using $\eta_k = 1$, since the Hessian is exact):

$$\begin{aligned}
\boldsymbol{\theta}_{k+1} &= \boldsymbol{\theta}_k - \mathbf{H}^{-1}\mathbf{g}_k \\
&= \boldsymbol{\theta}_k + (\mathbf{X}^T\mathbf{S}_k\mathbf{X})^{-1}\mathbf{X}^T(\mathbf{y} - \boldsymbol{\pi}_k) \\
&= (\mathbf{X}^T\mathbf{S}_k\mathbf{X})^{-1}\left[(\mathbf{X}^T\mathbf{S}_k\mathbf{X})\boldsymbol{\theta}_k + \mathbf{X}^T(\mathbf{y} - \boldsymbol{\pi}_k)\right] \\
&= (\mathbf{X}^T\mathbf{S}_k\mathbf{X})^{-1}\mathbf{X}^T\left[\mathbf{S}_k\mathbf{X}\boldsymbol{\theta}_k + \mathbf{y} - \boldsymbol{\pi}_k\right]
\end{aligned}$$

Iteratively reweighted least squares (IRLS)

```
from __future__ import division
import numpy as np

def logistic(a):
        return 1.0 / (1 + np.exp(-a))

def irls(X, y):
        theta = np.zeros(X.shape[1])
        theta_ = np.inf
        while max(abs(theta-theta_)) > 1e-6:
                a = np.dot(X, theta)
                pi = logistic(a)
                SX = X * (pi - pi*pi).reshape(-1,1)
                XSX = np.dot(X.T, SX)
                SXtheta = np.dot(SX, theta)
                theta_ = theta
                theta = np.linalg.solve(XSX, np.dot(X.T, SXtheta + y - pi))
        return theta
```

Bayesian logistic regression

The logistic regression model specifies the probability of a binary output $y_i \in \{0, 1\}$ given the input $\mathbf{x}_i$ as follows:

data $D = ((x_1, y_1) \cdots (x_n, y_n))$

$$p(\mathbf{y}|\mathbf{X}, \boldsymbol{\theta}) = \prod_{i=1}^{n} \text{Ber}(y_i|\text{sigm}(\mathbf{x}_i\boldsymbol{\theta}))$$

$$P(y|x) = \int P(y|x,\theta)P(\theta)d\theta = \prod_{i=1}^{n} \left[\frac{1}{1+e^{-\mathbf{x}_i\boldsymbol{\theta}}}\right]^{y_i} \left[1 - \frac{1}{1+e^{-\mathbf{x}_i\boldsymbol{\theta}}}\right]^{1-y_i}$$

We also assume a Gaussian prior

$$p(\theta) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(\theta - \mu)'(\theta - \mu)\right)$$

Posterior $P(\theta|D) \propto P(y|x,\theta)\,P(\theta)$

$P(\theta|D) = \dfrac{P(y|x,\theta)\,P(\theta)}{P(y|x)}$

Bayesian logistic regression: Predictive distribution

Data $D = (x_{1:n}, y_{1:n})$

$P(AB) = P(A|B)P(B)$

$P(B) = \sum_A P(AB)$

New input x_{n+1} we want to predict y_{n+1}

$$P(y_{n+1} | x_{n+1}, D) = \int P(y_{n+1}, \theta | x_{n+1}, D)\, d\theta$$

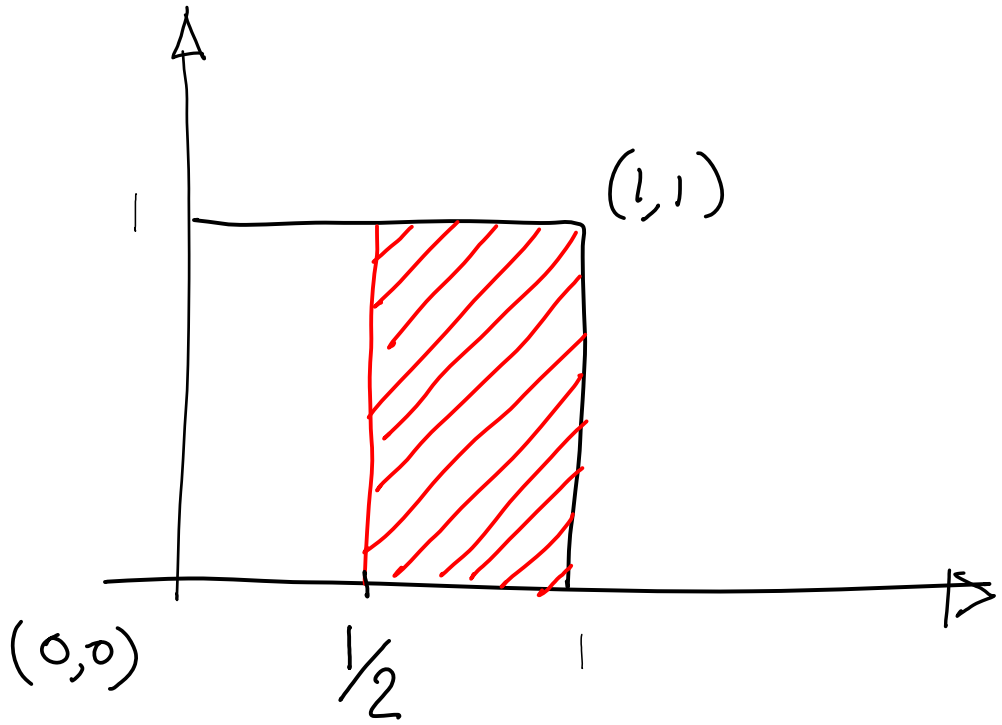
$$= \int P(y_{n+1} | \theta, x_{n+1}, \cancel{D})\, P(\theta | x_{n+1}, D)\, d\theta$$

$$= \int \underbrace{\pi_{n+1}^{y_{n+1}} (1-\pi_{n+1})^{1-y_{n+1}}}_{P(y_{n+1}|\theta, x_{n+1})}\, P(\theta|D)\, d\theta \approx \frac{1}{N} \sum_{i=1}^{N} P(y_{n+1} | \theta^{(i)}, x_{n+1})$$

$$\pi_{n+1} = \frac{1}{1+e^{-x_{n+1}\theta}}$$

The idea

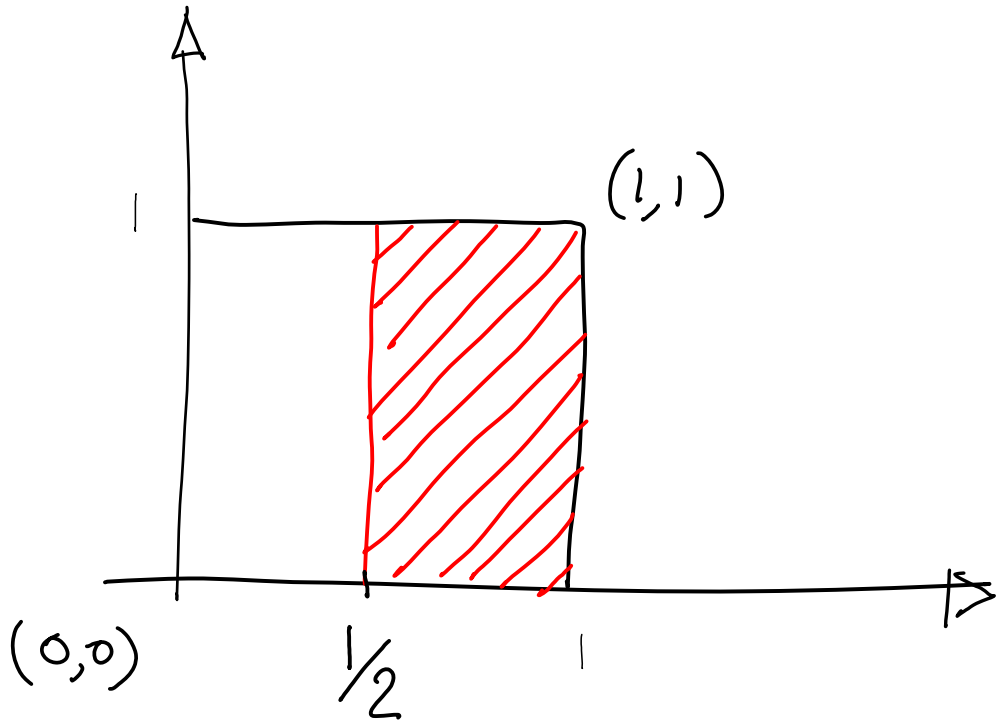
What is the probability that a dart thrown uniformly at random will hit the red area?



$$P(\text{area}) =$$

The idea

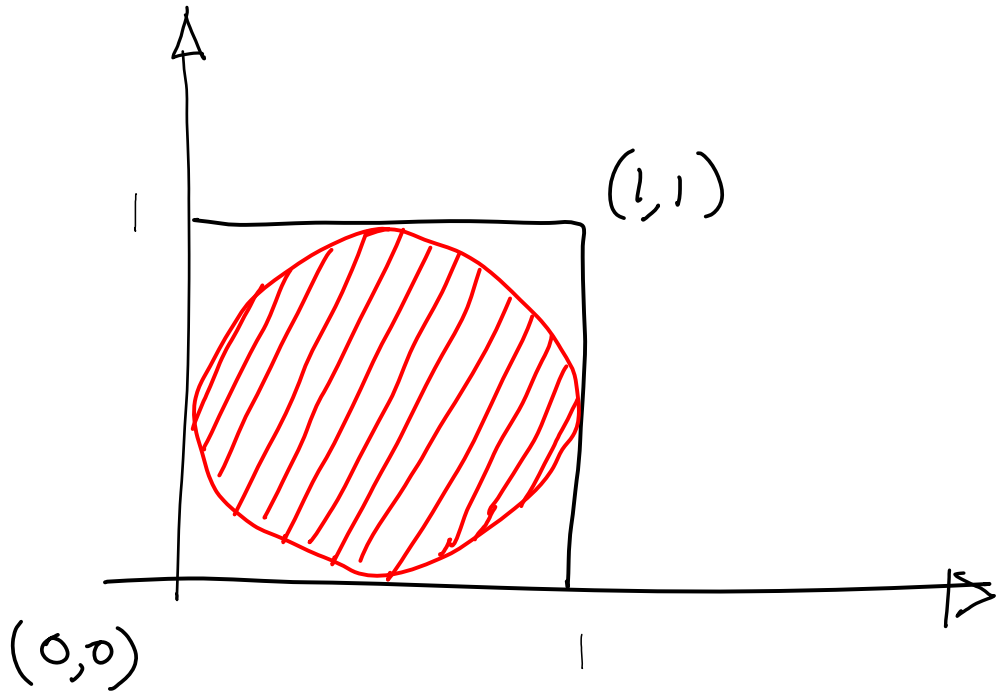
What is the probability that a dart thrown uniformly at random will hit the red area?



$$P(\text{area}) = 1/2$$

The idea

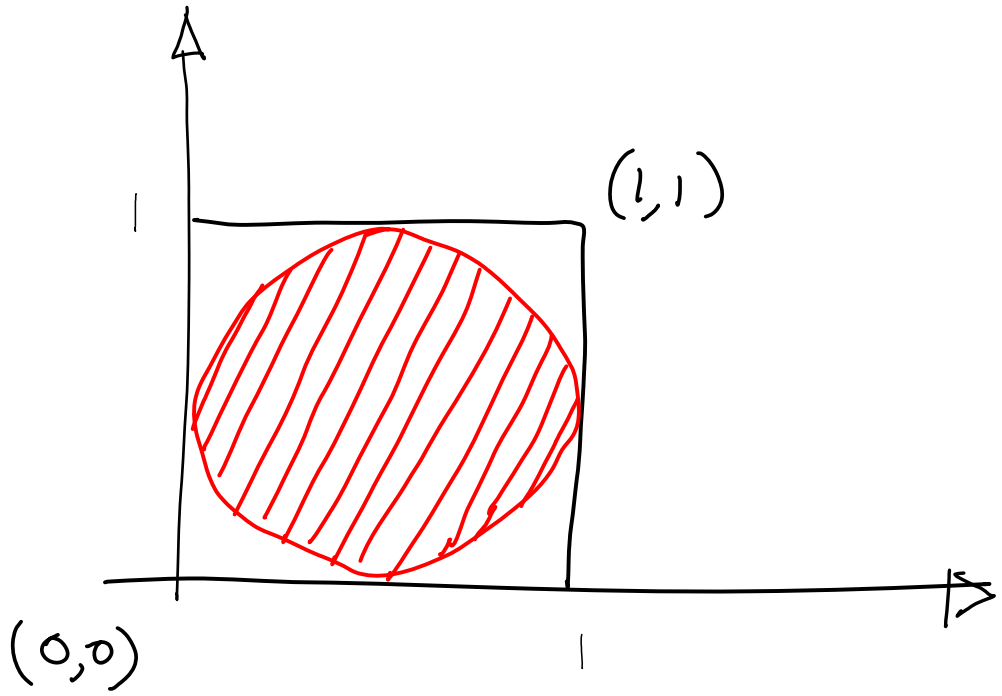
What is the probability that a dart thrown uniformly at random will hit the red area?



$$P(\text{area}) =$$

The idea

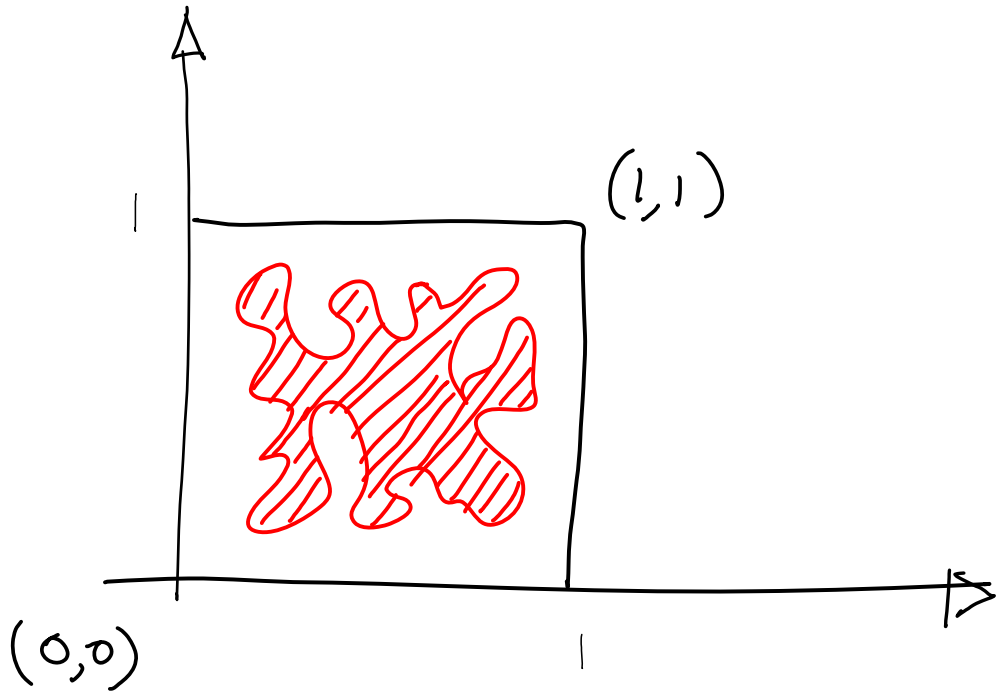
What is the probability that a dart thrown uniformly at random will hit the red area?



$$P(\text{area}) = \pi\left(\frac{1}{2}\right)^2 = \frac{\pi}{4}$$

The idea

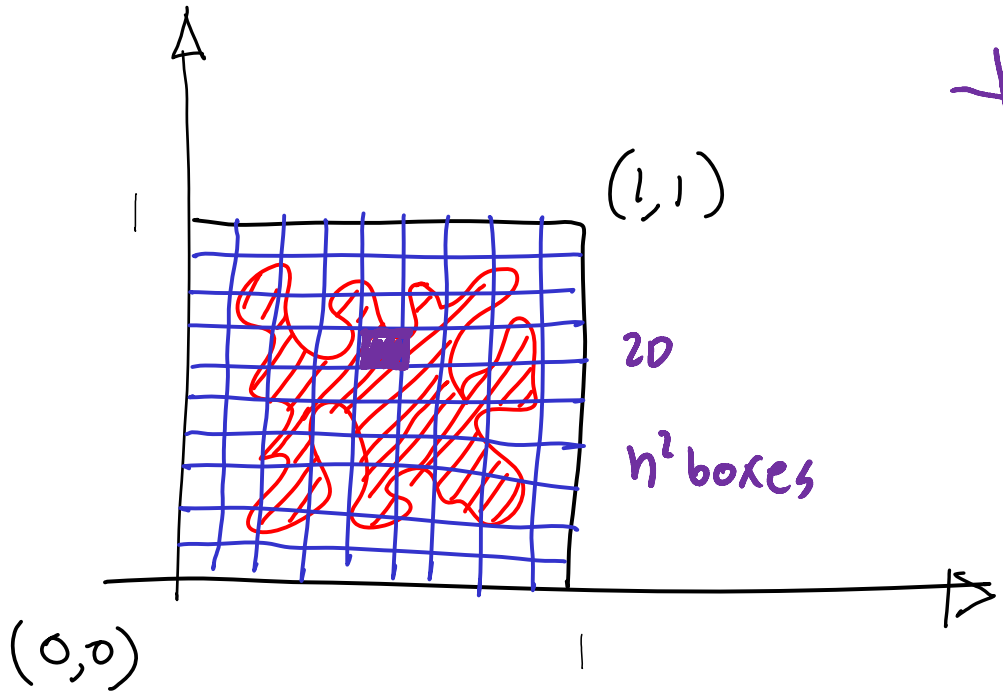
What is the probability that a dart thrown uniformly at random will hit the red area?

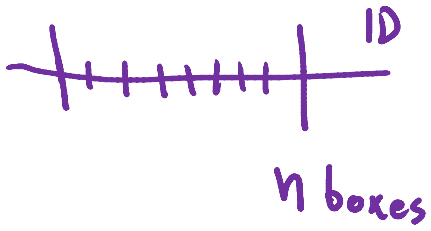


$$P(\text{area}) =$$

The idea

What is the probability that a dart thrown uniformly at random will hit the red area?

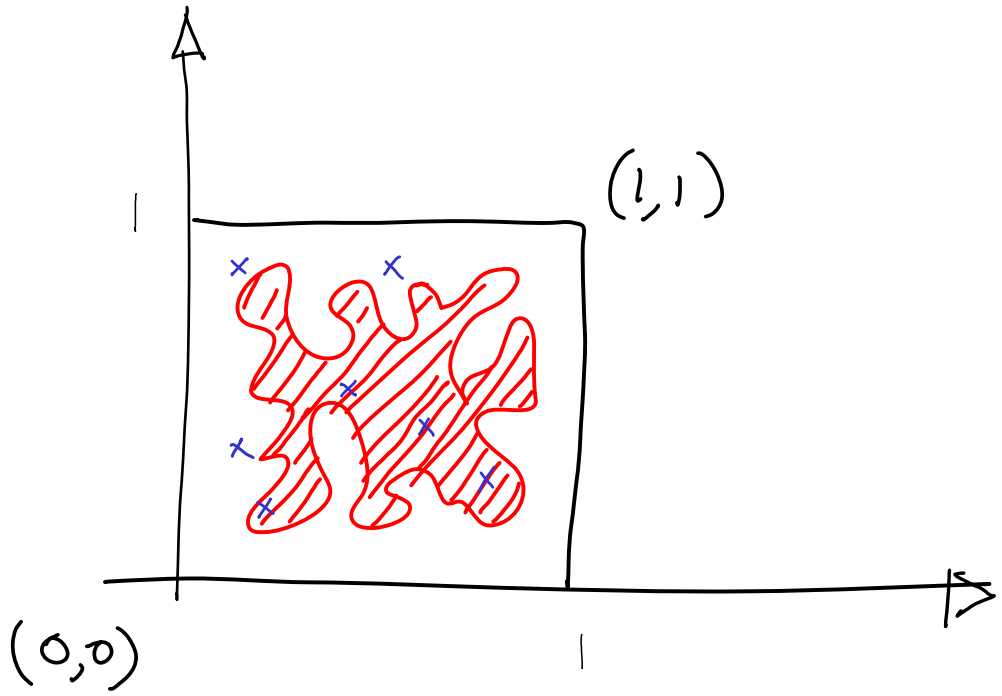




$$P(\text{area}) = \frac{\#\text{ red boxes}}{\#\text{ boxes}}$$

The idea

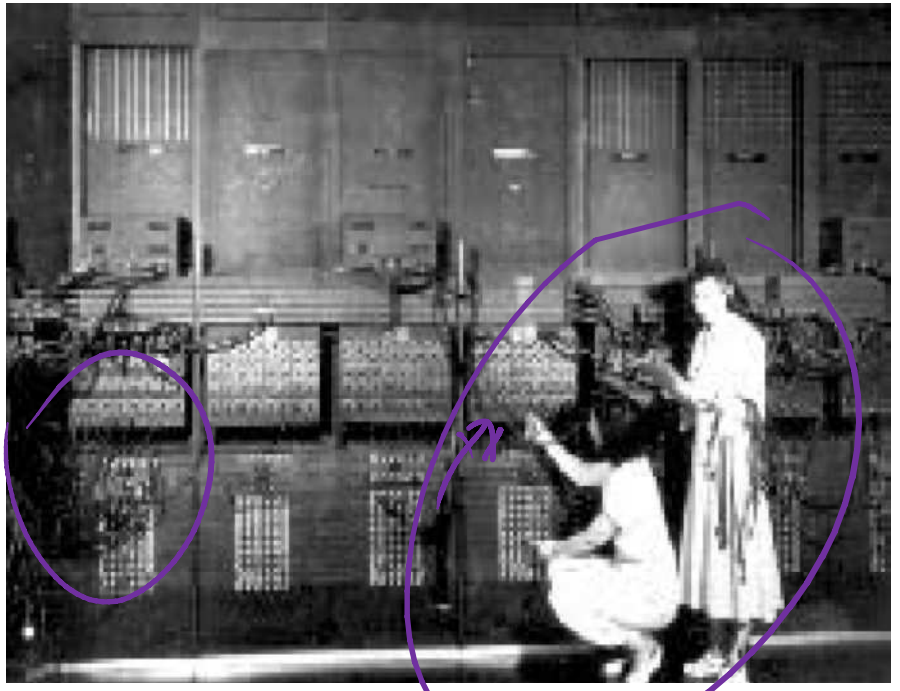
What is the probability that a dart thrown uniformly at random will hit the red area?



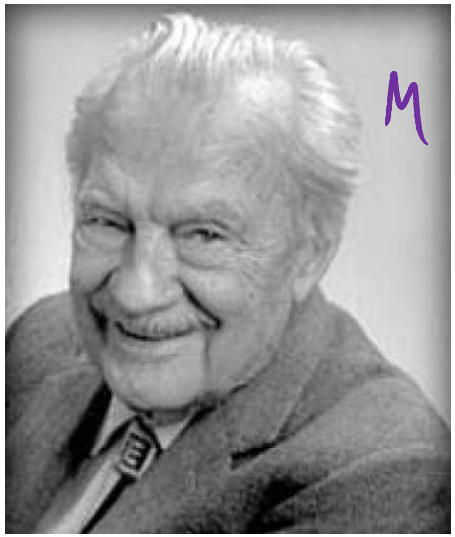
4/7

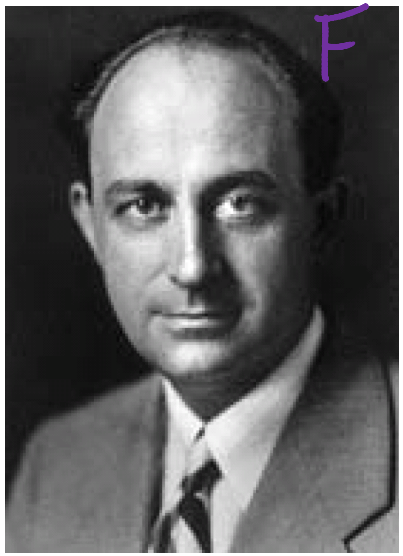
$$P(\text{area}) = \frac{\#\text{ darts in }}{\#\text{ darts in }\square}$$

History of the Monte Carlo method: The bomb and ENIAC



History of the Monte Carlo method





Monte Carlo Integration

Suppose we want to compute

$$I = \int f(x)\, P(x \mid \text{data})\, dx$$

$P(x \mid \text{data})$: posterior

Monte Carlo Integration

Suppose we want to compute

$$I = \int f(x)\, P(x \mid \text{data})\, dx$$

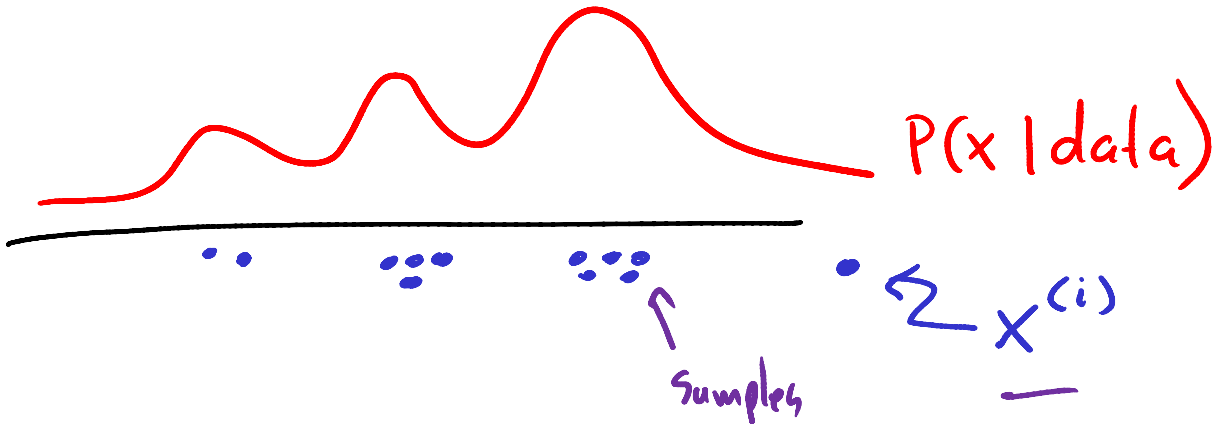
(i) Simulate $x^{(i)}\Big|_{i=1}^{N}$ from $P(x \mid \text{data})$

Monte Carlo Integration

Suppose we want to compute

$$I = \int f(x)\, P(x \mid \text{data})\, dx$$

(i) Simulate $x^{(i)}\Big|_{i=1}^{N}$ from $P(x \mid \text{data})$

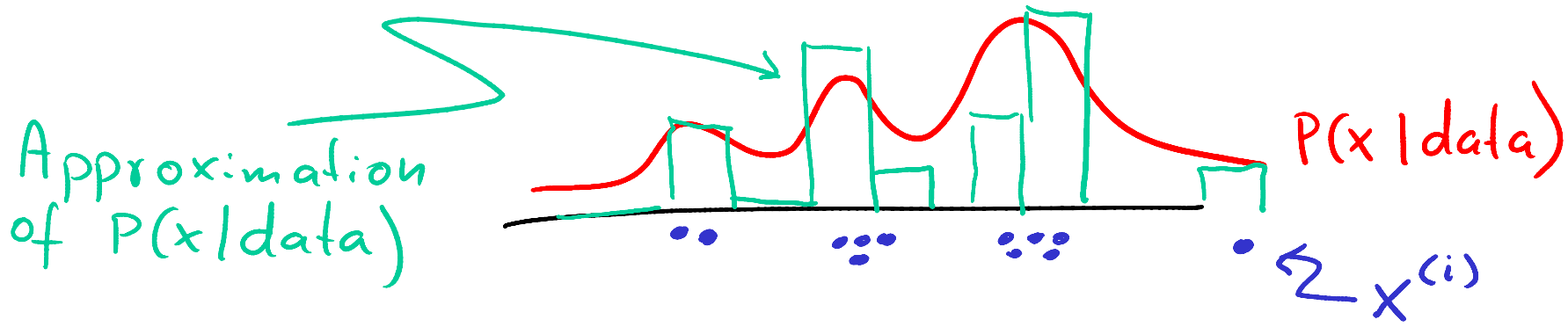


Monte Carlo Integration

Suppose we want to compute

$$I = \int f(x)\, P(x \mid \text{data})\, dx$$

(i) Simulate $x^{(i)}\Big|_{i=1}^{N}$ from $P(x \mid \text{data})$

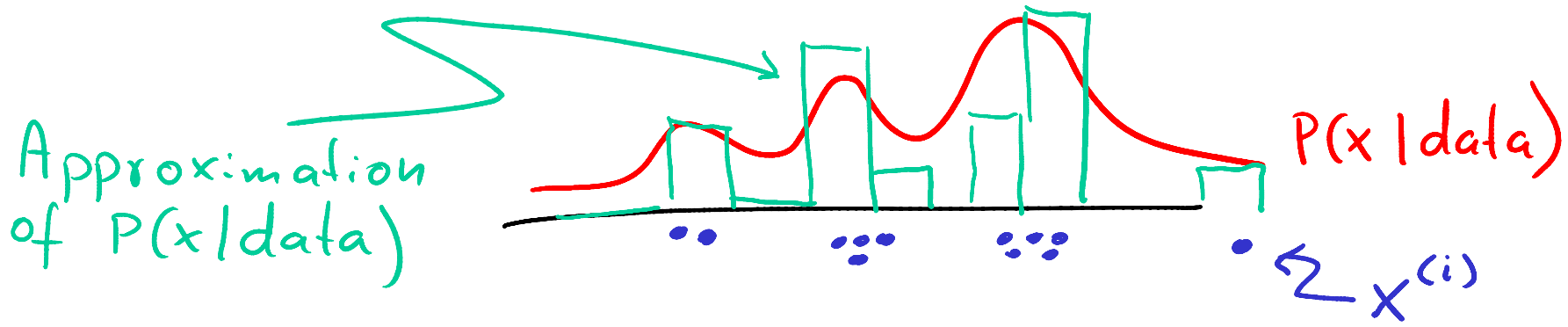


Monte Carlo Integration

Suppose we want to compute

$$I = \int f(x)\, P(x \mid \text{data})\, dx .$$

(i) Simulate $x^{(i)}\Big|_{i=1}^{N}$ from $P(x \mid \text{data})$



(ii) Replace nasty integral with simple sum:

$$I \approx \frac{1}{N} \sum_{i=1}^{N} f(x^{(i)})$$

Importance sampling

$$f(\theta) = \left(\frac{1}{1+e^{x\theta}}\right)^{y}\left(1-\frac{1}{1+e^{-x\theta}}\right)^{1-y}$$

$$I = \int f(\theta)\, P(\theta|D)\, d\theta$$

Introduce $q(\theta)$ (easy to sample)

$q(\theta) \sim N(0, \Sigma)$

$$= \int \left[f(\theta) \frac{P(\theta|D)}{q(\theta)} \right] q(\theta)\, d\theta = \int f(\theta)\, w(\theta)\, q(\theta)\, d\theta$$

$\theta^{(i)} \sim q(\theta)$

$$\approx \frac{1}{N} \sum_{i=1}^{N} f(\theta^i) w(\theta^i)$$

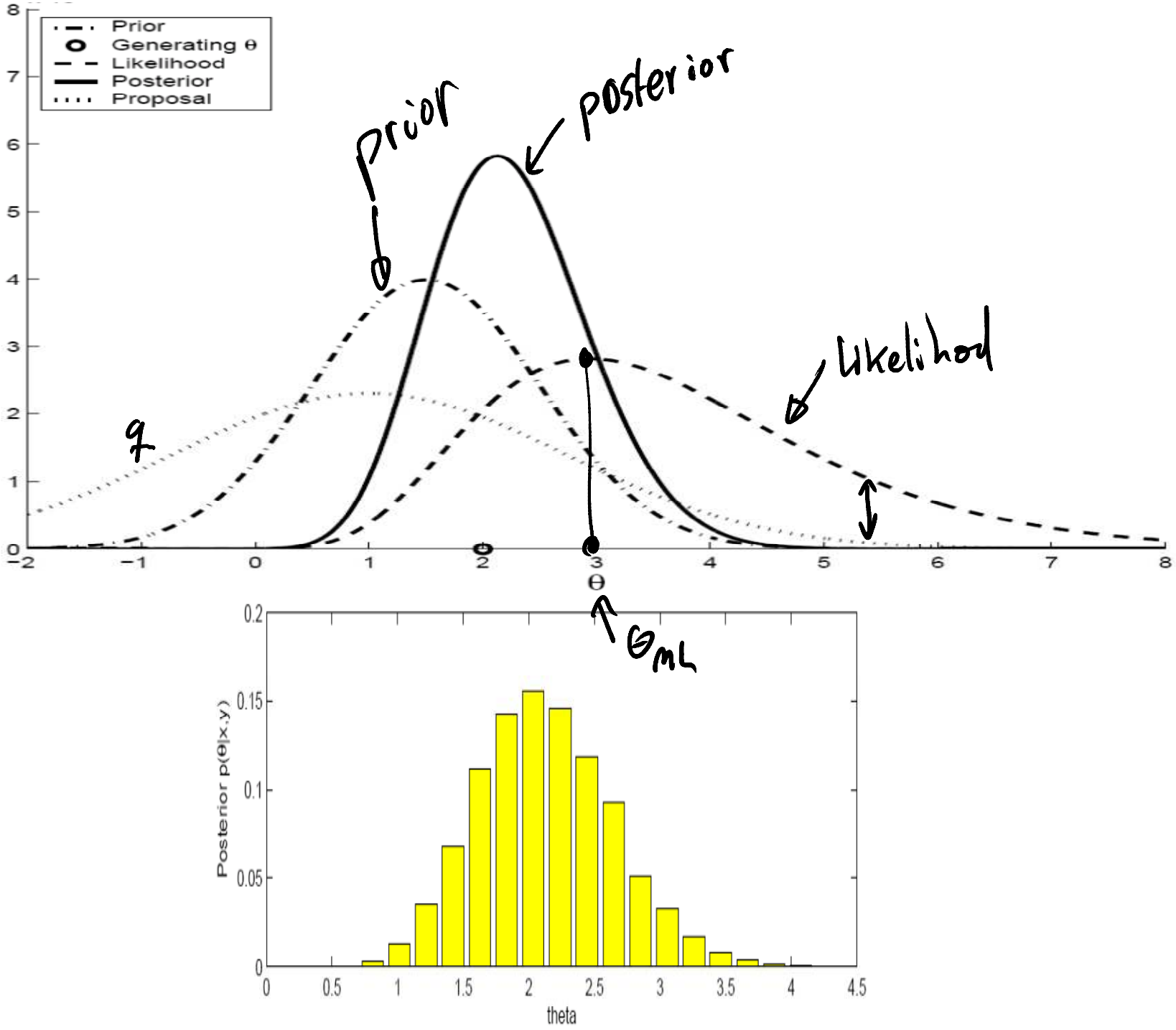
Importance sampling $Z = \int g(\theta)\, d\theta$

$$P(\theta|D) = \frac{P(D|\theta)\,P(\theta)}{P(D)} = \frac{P(D|\theta)P(\theta)}{Z} = \frac{g(\theta)}{Z}$$

$$I = \int f(\theta)\,P(\theta|D)\,d\theta = \frac{1}{Z}\int f(\theta)\,g(\theta)\,d\theta$$

$$= \frac{\int f(\theta)\,g(\theta)\,d\theta}{\int g(\theta)\,d\theta} = \frac{\int f(\theta)\frac{g(\theta)}{q(\theta)}\,q(\theta)\,d\theta}{\int \frac{g(\theta)}{q(\theta)}\,q(\theta)\,d\theta} \qquad \theta^{(i)} \sim q(\theta)$$

$$\approx \frac{\sum w(\theta^{(i)})\,f(\theta^{(i)})}{\sum w(\theta^{(i)})}$$


Prior
Generating θ
Likelihood
Posterior
Proposal
Prior
posterior
likelihood
q
θ
θML
Posterior p(θ|x,y)
theta

Example: Logistic Regression

$$p(y_{T+1}|x_{1:T+1}) = \int_{\Theta} p(y_{T+1}|x_{T+1},\theta)p(\theta|x_{1:T},y_{1:T})d\theta$$

$$p(y_{T+1}|x_{1:T+1}) = \frac{1}{N}\sum_{i=1}^{N} p(y_{T+1}|x_{T+1},\theta^{(i)})$$