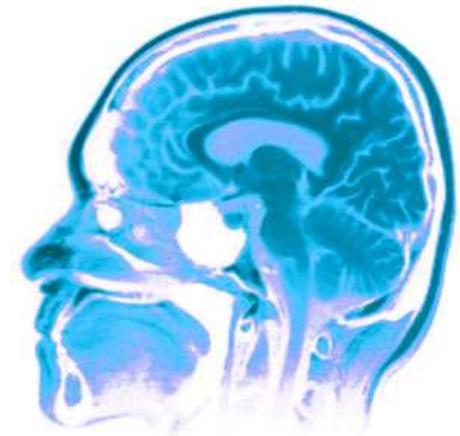




# CPS<sup>C</sup>540



## Logistic Regression and Neuron Models



Nando de Freitas

*March, 2013*

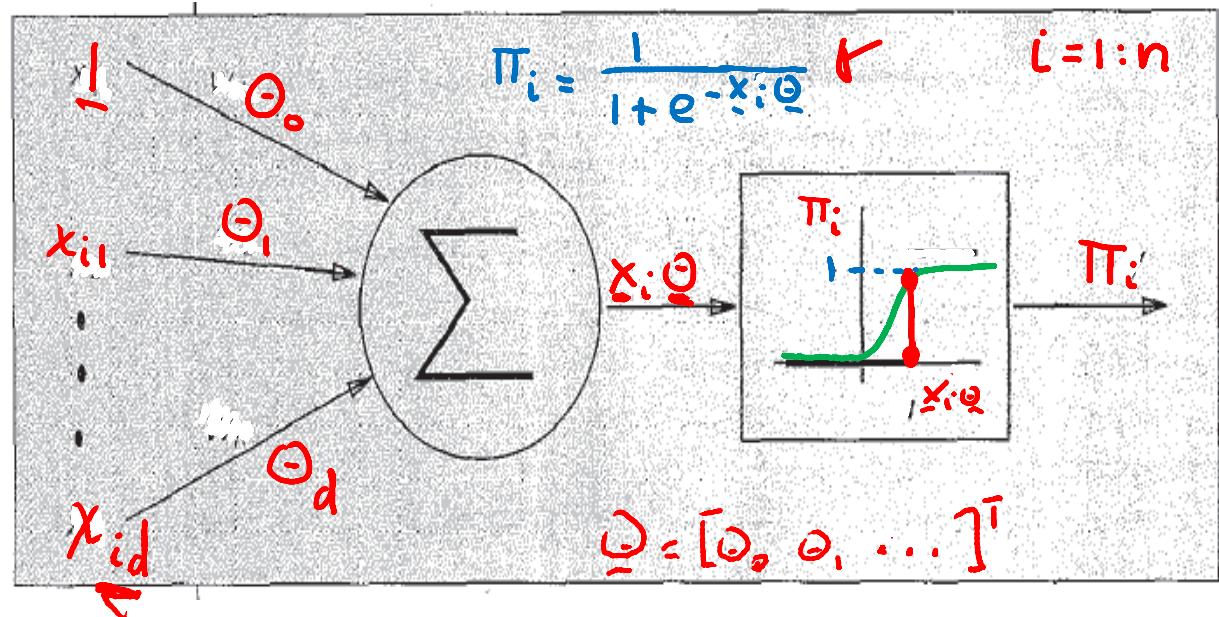
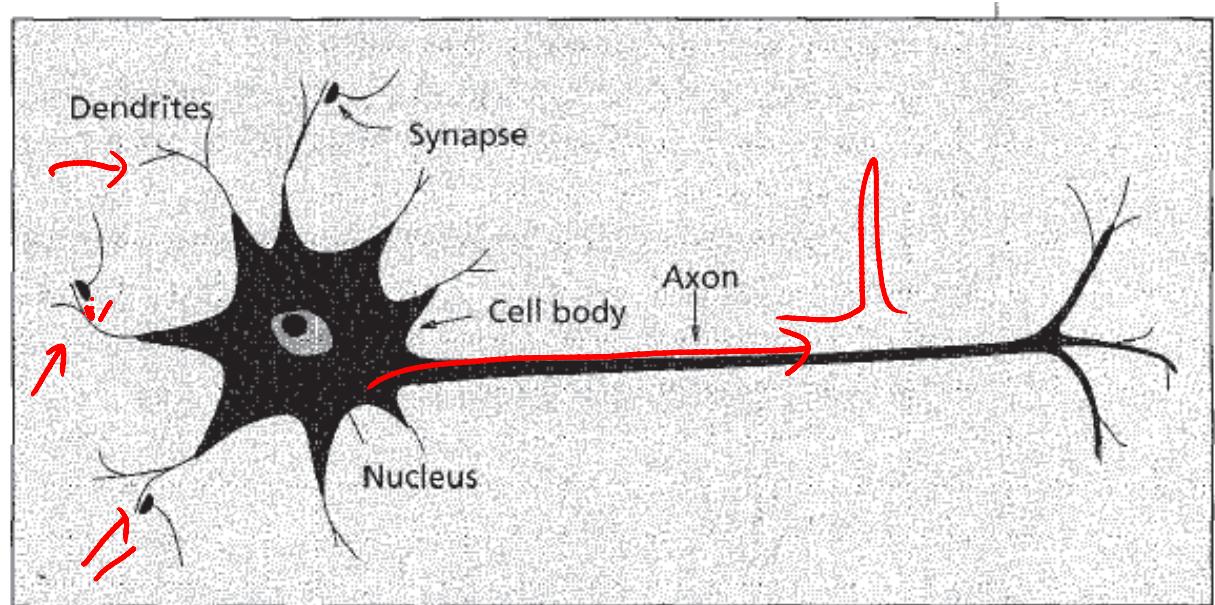
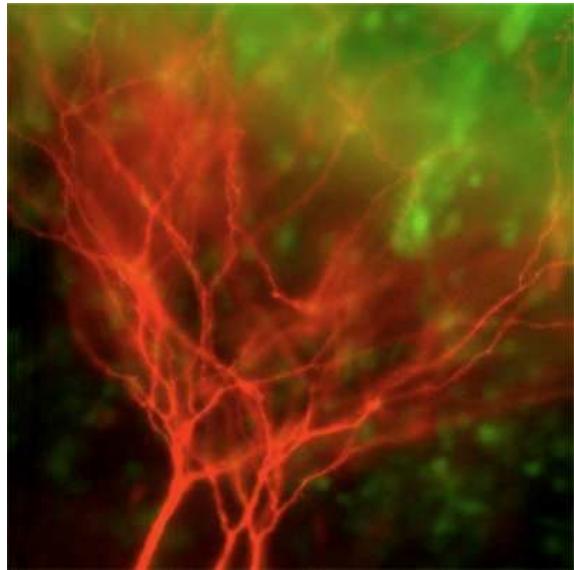
*University of British Columbia*

# 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

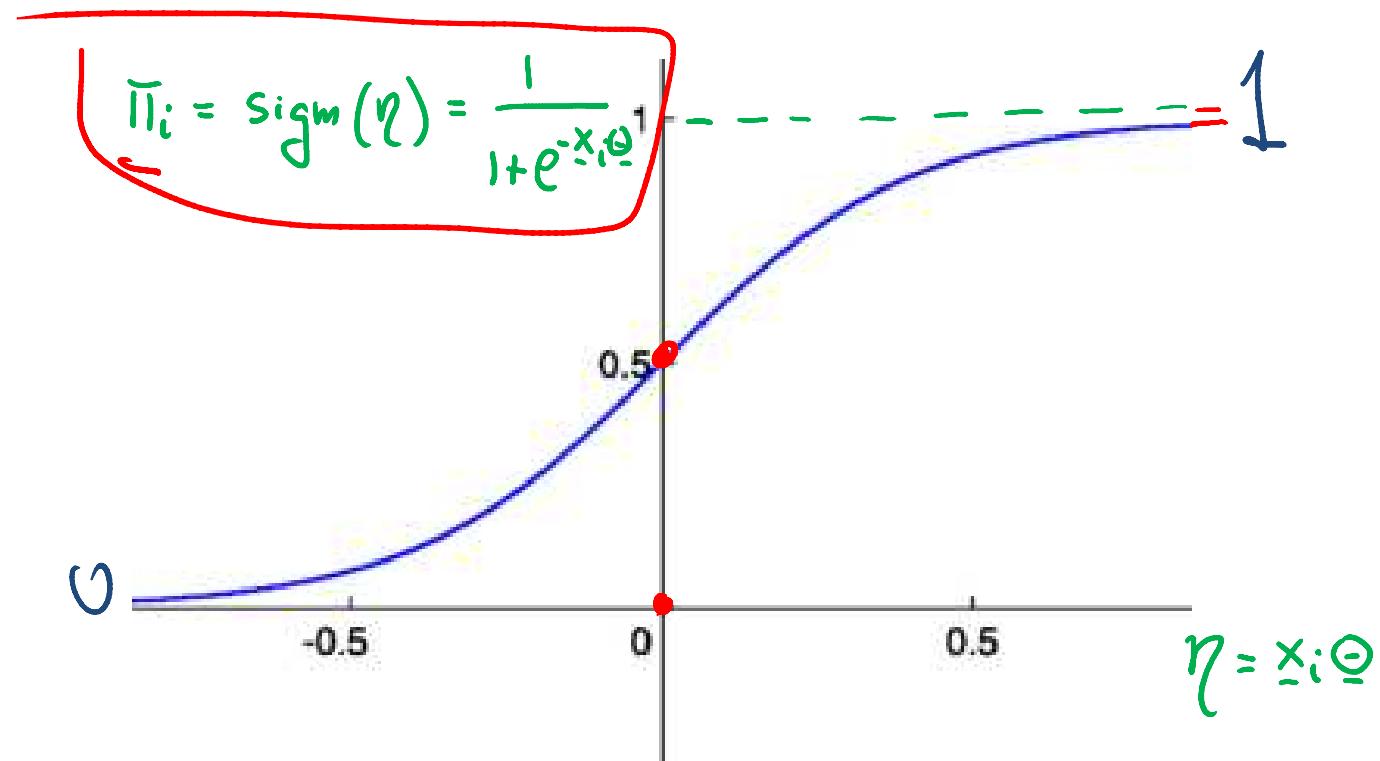


# Sigmoid function

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

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

$\frac{\partial L}{\partial \theta_j}$



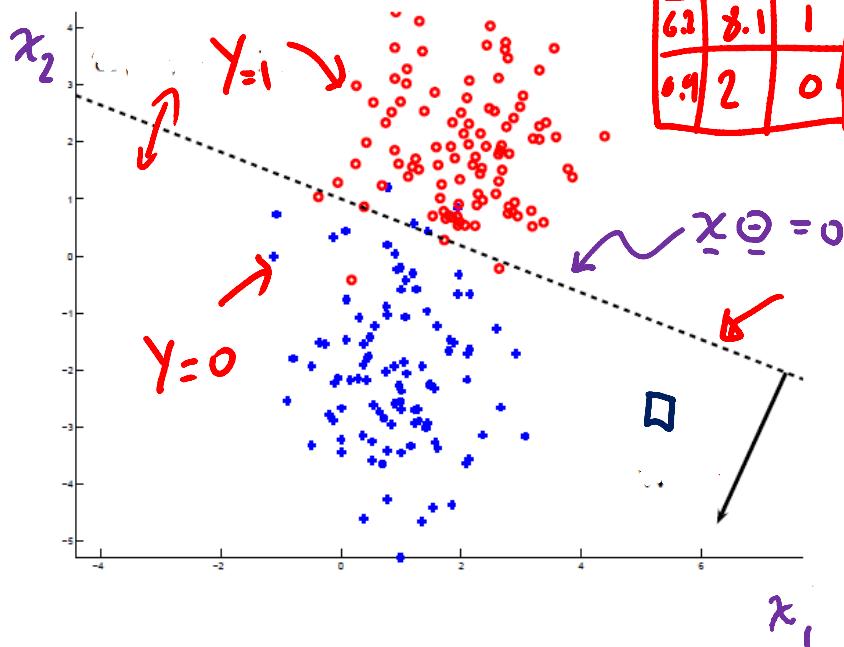
# Linear separating hyper-plane

$$\pi_i P(y_i=1 | \underline{x}_i, \underline{\Theta}) = \text{sigm}(\underline{x}_i \underline{\Theta}) = \frac{1}{1+e^{-\underline{x}_i \underline{\Theta}}}$$

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

Data D

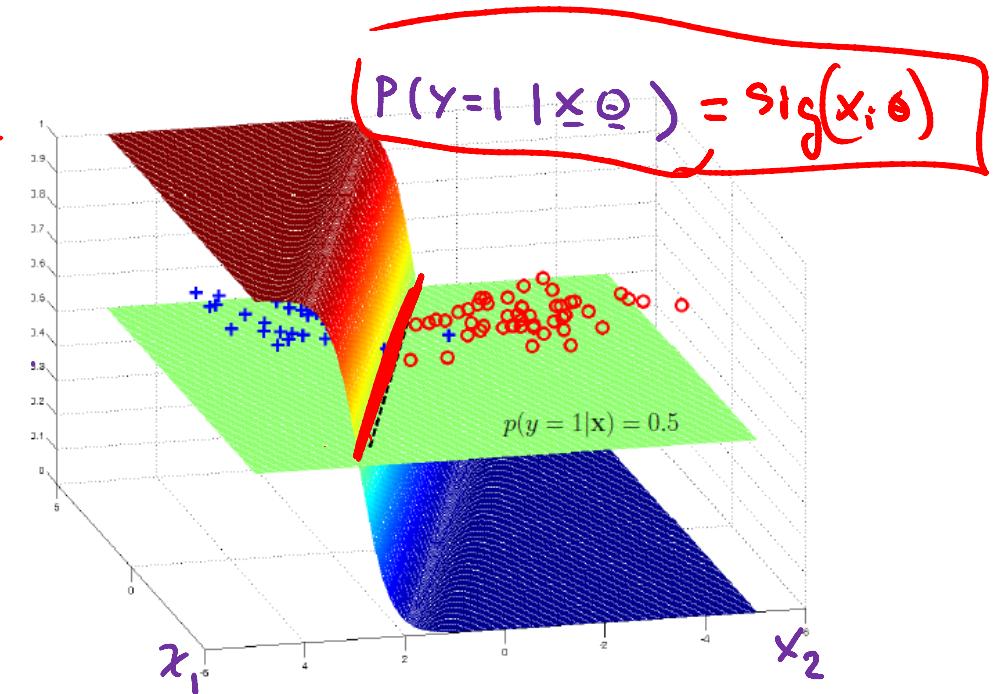
$x_1$	$x_2$	$y$
1	2.3	0
6.3	3.1	1
0.9	2	0



When

$$\underline{x}_i \underline{\Theta} = 0$$

EQUATION OF A  
PLANE.



[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:

$$\begin{aligned}
 p(\mathbf{y}|\mathbf{X}, \boldsymbol{\theta}) &= \prod_{i=1}^n \text{Ber}(y_i | \text{sigm}(\mathbf{x}_i \boldsymbol{\theta})) = \prod_{i=1}^n \underbrace{\pi_i}_{\substack{\text{def} \\ \text{n} \times 1}}^{y_i} \underbrace{(1-\pi_i)}_{1-y_i}^{1-y_i} = \begin{cases} \pi_i & y_i = 1 \\ 1-\pi_i & y_i = 0 \end{cases} \\
 &= \prod_{i=1}^n \left[ \frac{1}{1 + e^{-\mathbf{x}_i \boldsymbol{\theta}}} \right]^{y_i} \left[ 1 - \frac{1}{1 + e^{-\mathbf{x}_i \boldsymbol{\theta}}} \right]^{1-y_i}
 \end{aligned}$$

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

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

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:

$$\left\{ \begin{array}{lcl} \mathbf{g}(\mathbf{w}) & = & \frac{d}{d\boldsymbol{\theta}} J(\boldsymbol{\theta}) = \sum_{i=1}^n \mathbf{x}_i^T (\pi_i - y_i) = \underbrace{\mathbf{X}^T (\boldsymbol{\pi} - \mathbf{y})}_{\text{gradient}} \\ \mathbf{H} & = & \frac{d}{d\boldsymbol{\theta}} \mathbf{g}(\boldsymbol{\theta})^T = \sum_i \pi_i(1-\pi_i) \mathbf{x}_i \mathbf{x}_i^T = \mathbf{X}^T \text{diag}(\pi_i(1-\pi_i)) \mathbf{X} \end{array} \right.$$

$\Theta_{t+1} = \Theta_t - \gamma \times \mathbf{X}^T [\pi(\Theta_t) - \mathbf{y}]$

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

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

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

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

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

# 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:

$$\text{data } D = \{(x_1, y_1), \dots, (x_n, y_n)\}$$

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

We also assume a Gaussian prior

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

Posterior  $P(\boldsymbol{\theta}|D) \propto P(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta}) P(\boldsymbol{\theta})$

$$P(\boldsymbol{\theta}|D) \propto \frac{P(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta}) P(\boldsymbol{\theta})}{P(\mathbf{y}|\mathbf{x})}$$

# Bayesian logistic regression: Predictive distribution

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

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

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

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

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

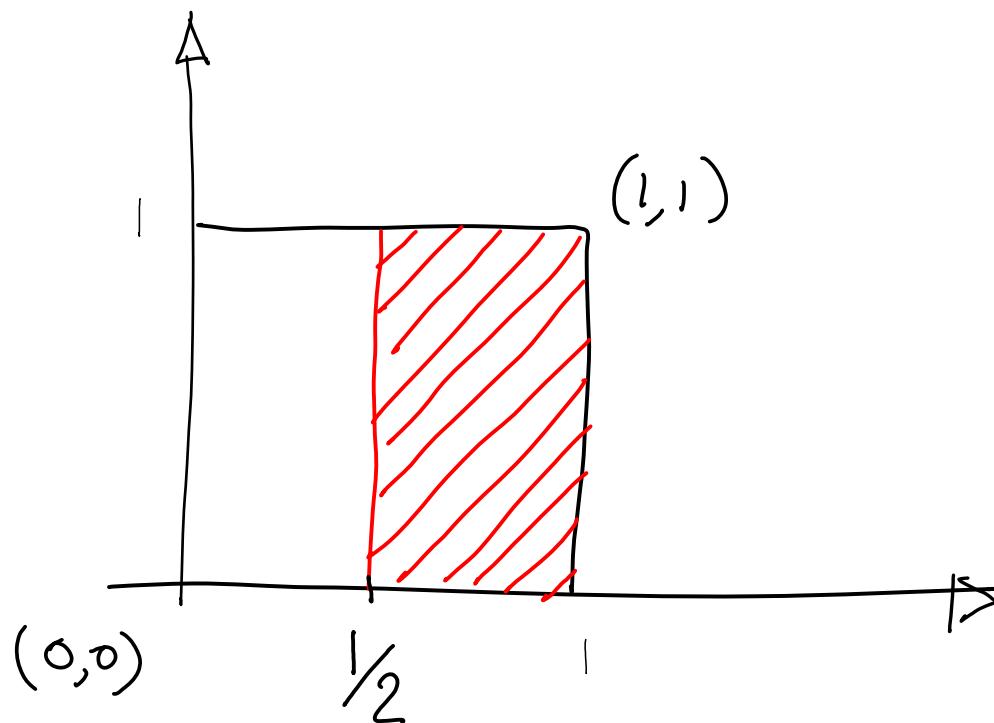
$$= \int P(y_{n+1} | \theta, x_{n+1}, \cancel{D}) P(\theta | x_{n+1}, D) d\theta$$

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

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

# The idea

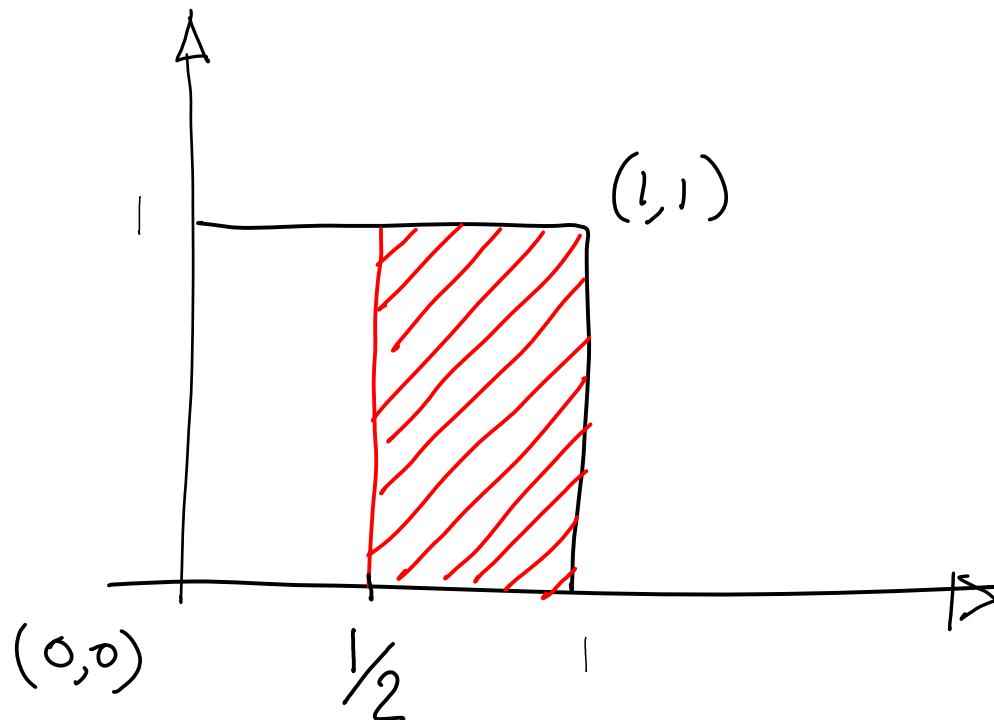
What is the probability that a dart thrown uniformly at random will hit the red area?



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

# The idea

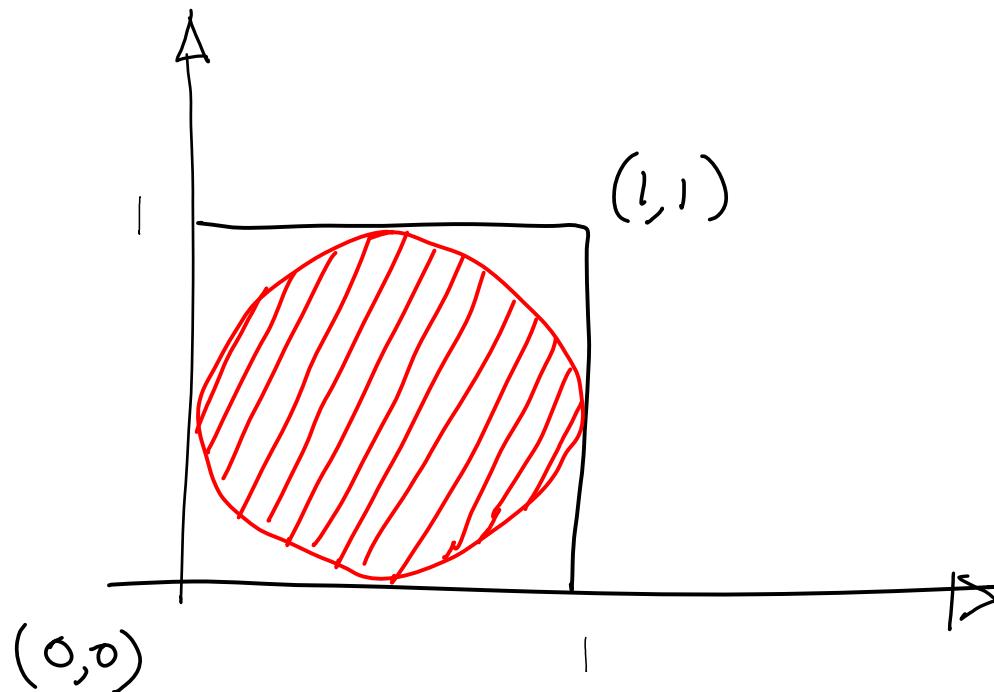
What is the probability that a dart thrown uniformly at random will hit the red area?



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

# The idea

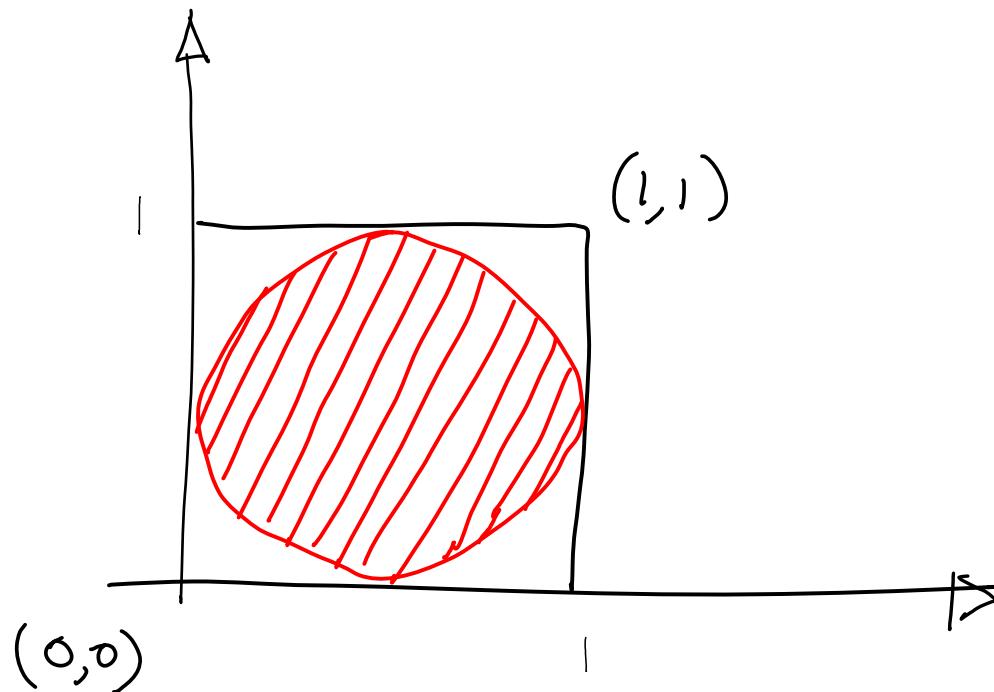
What is the probability that a dart thrown uniformly at random will hit the red area?



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

# The idea

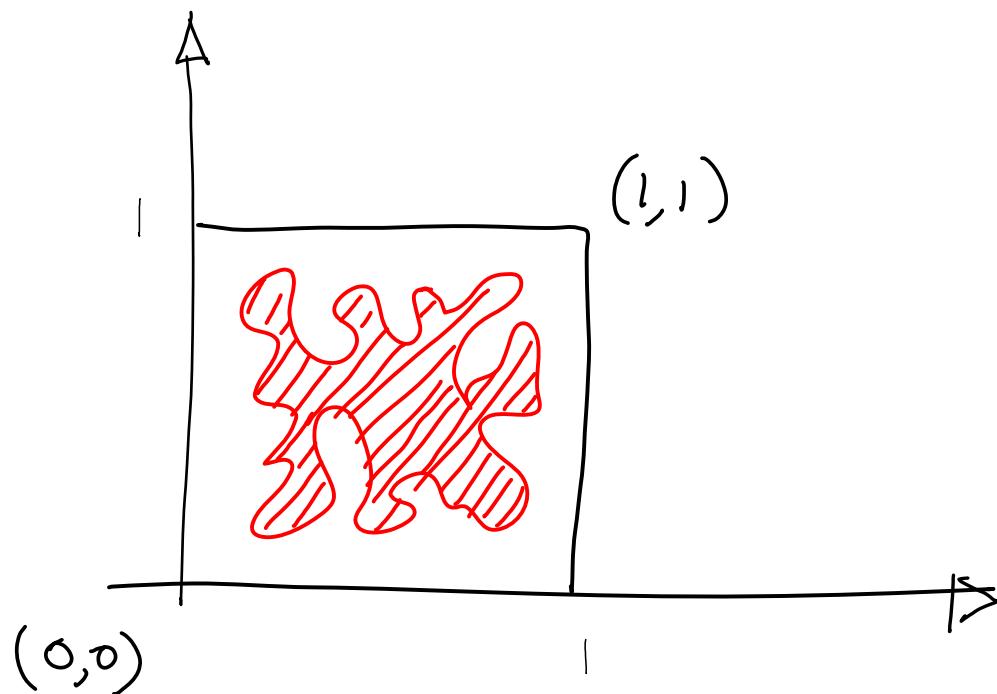
What is the probability that a dart thrown uniformly at random will hit the red area?



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

# The idea

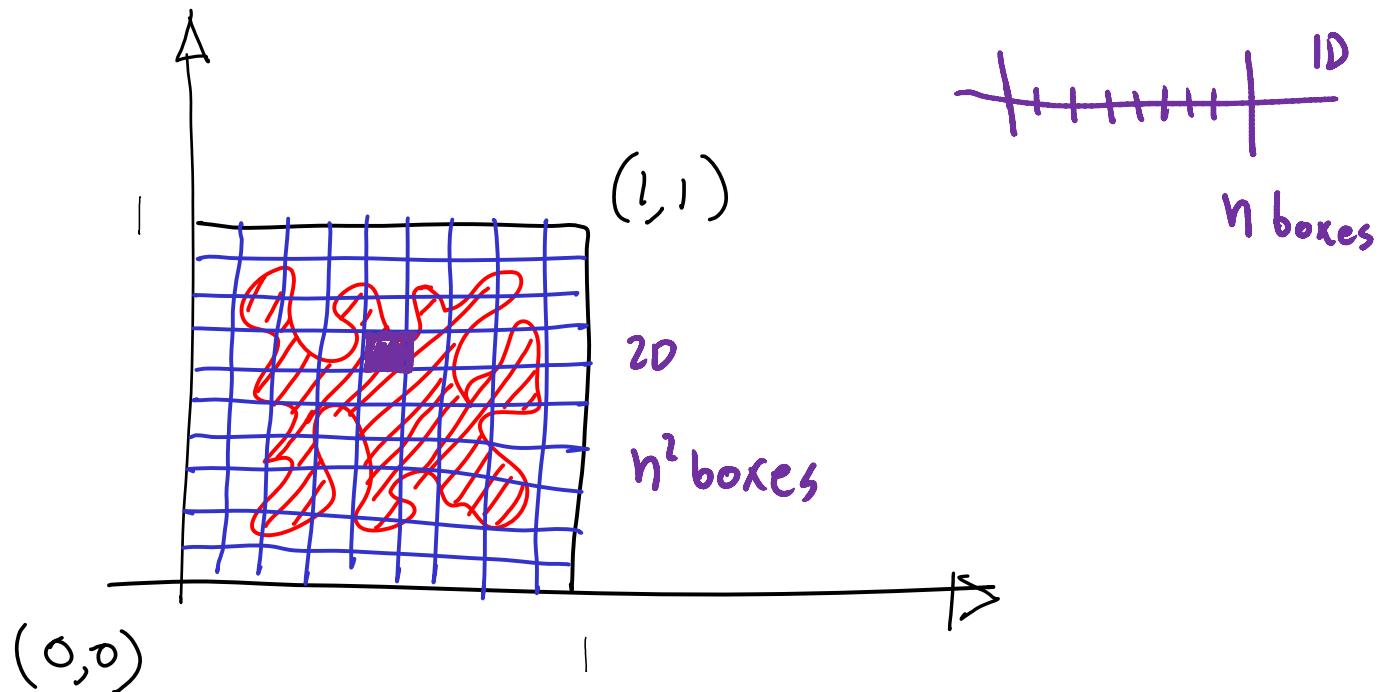
What is the probability that a dart thrown uniformly at random will hit the red area?



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

# The idea

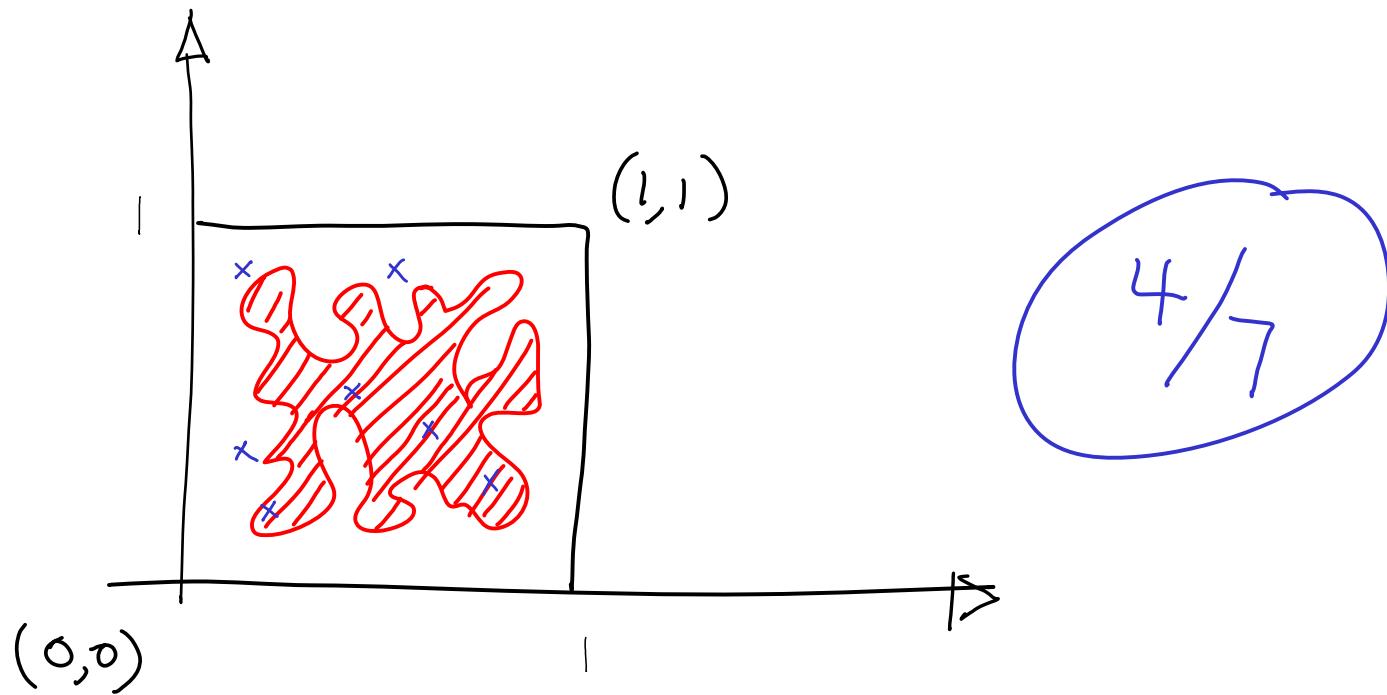
What is the probability that a dart thrown uniformly at random will hit the red area?



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

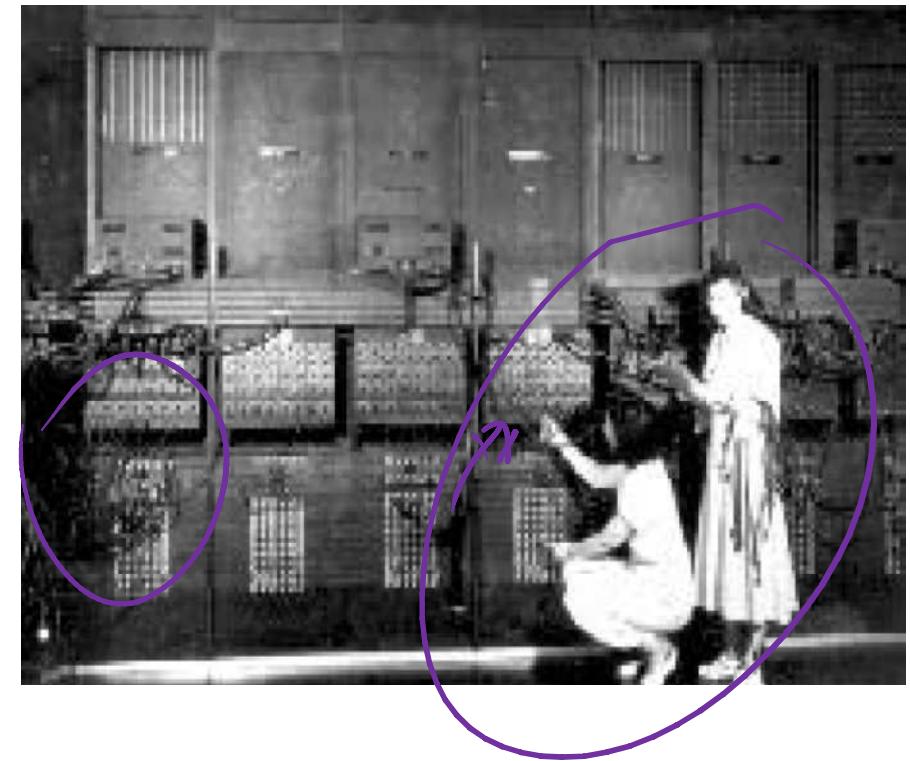
# The idea

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

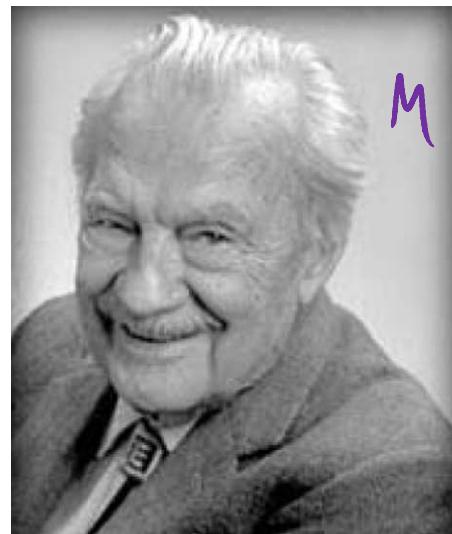


$$P(\text{area}) = \frac{\#\text{ darts in } \text{red blob}}{\#\text{ darts in } \square}$$

# History of the Monte Carlo method: The bomb and ENIAC



# History of the Monte Carlo method



# Monte Carlo Integration

Suppose we want to compute

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

posterior

# Monte Carlo Integration

Suppose we want to compute

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

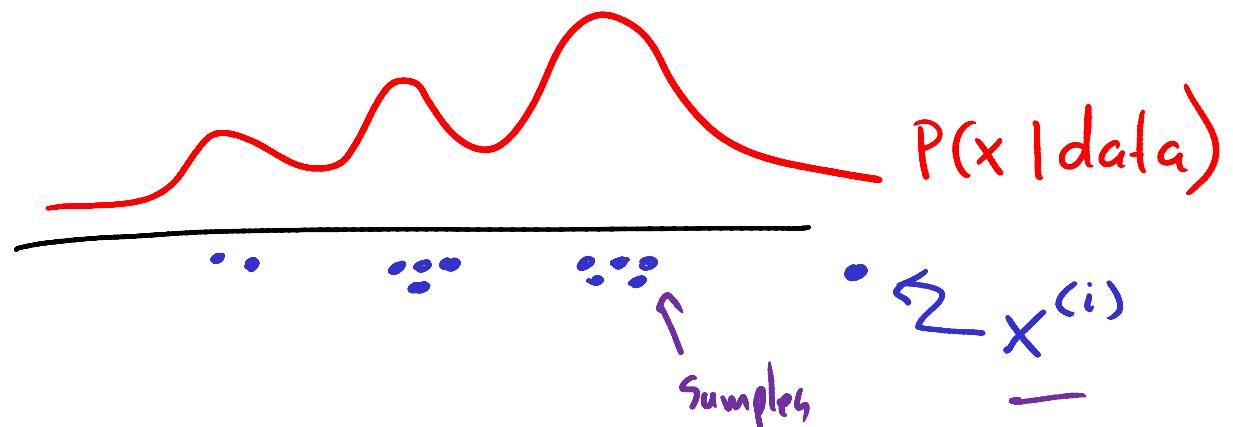
(i) Simulate  $x^{(i)} \Big|_{i=1}^N$  from  $P(x | \text{data})$

# Monte Carlo Integration

Suppose we want to compute

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

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

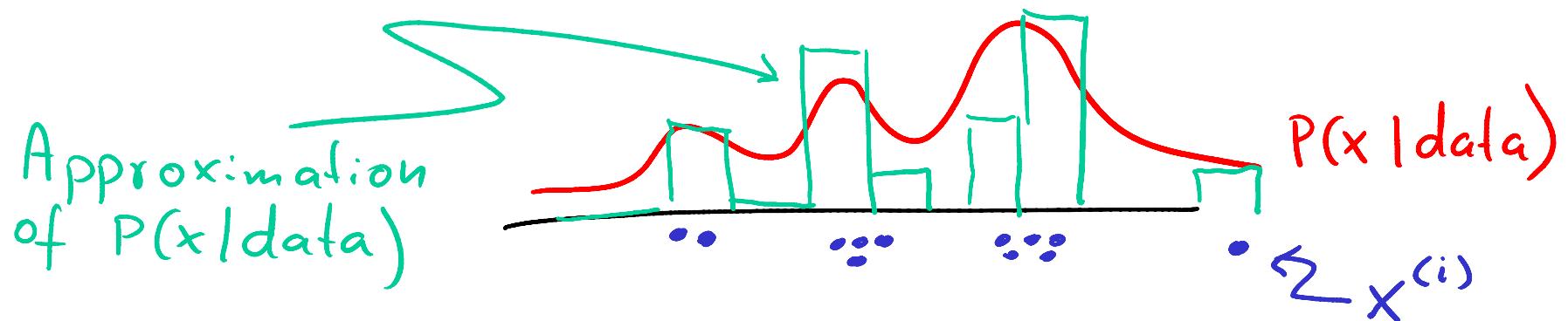


# Monte Carlo Integration

Suppose we want to compute

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

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

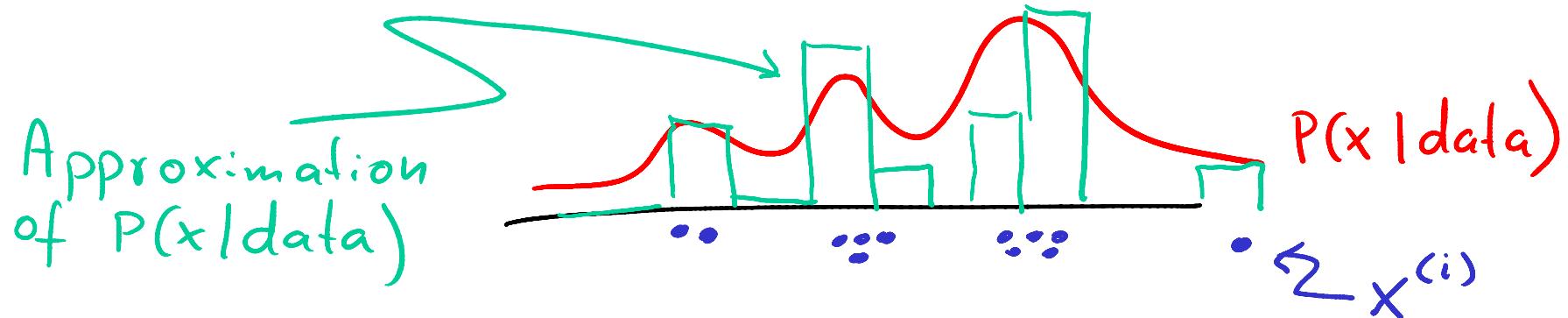


# Monte Carlo Integration

Suppose we want to compute

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

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



(ii) Replace nasty integral with simple sum:  $I \approx \frac{1}{N} \sum_{i=1}^N f(x^{(i)})$   $\blacksquare$

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

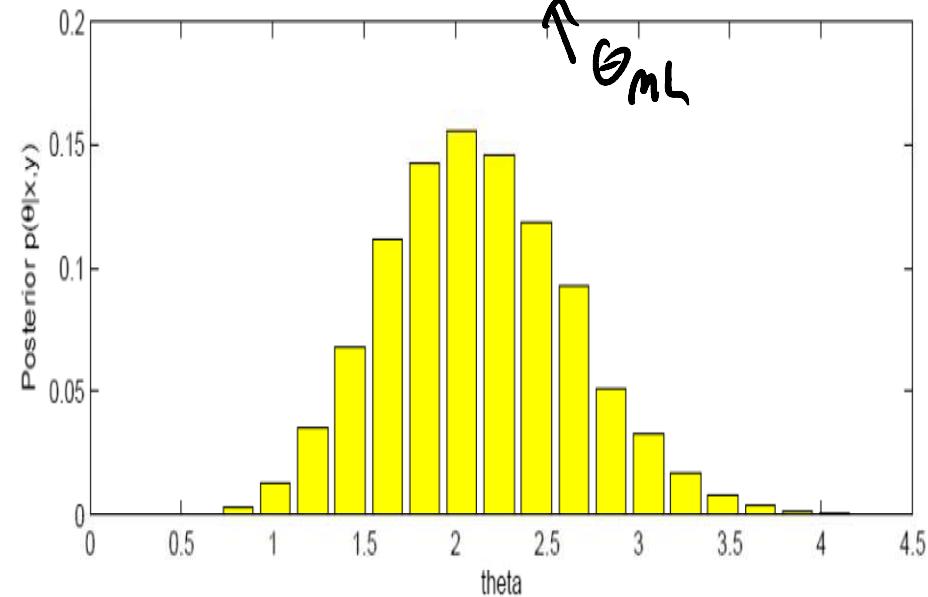
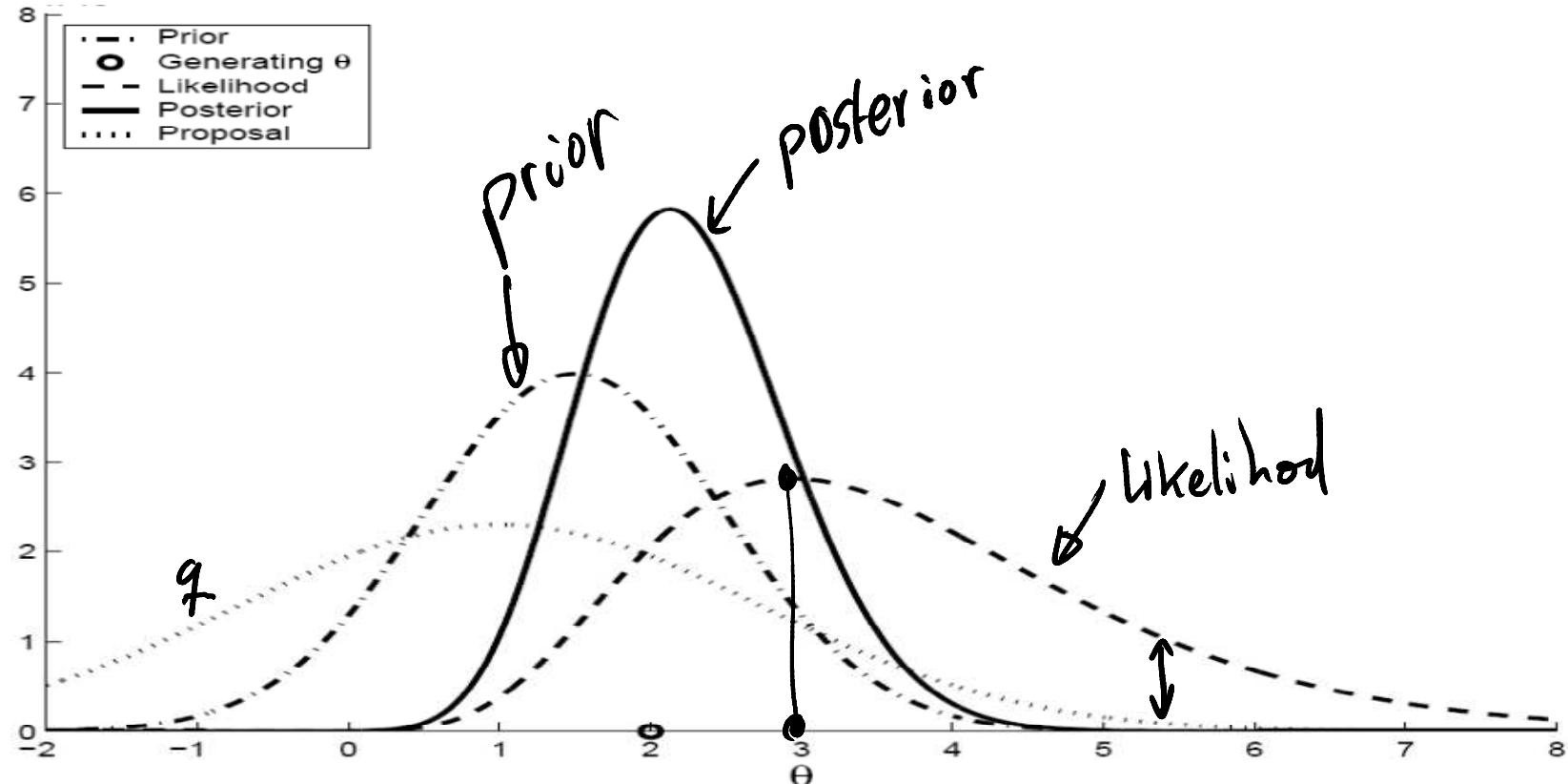
$$= \int \left[ f(\theta) \frac{P(\theta | D)}{q(\theta)} \right] q(\theta) d\theta = \int f(\theta) w(\theta) q(\theta) d\theta$$

$$\approx \frac{1}{n} \sum_{i=1}^n f(\theta^i) w(\theta^i) \quad \theta^i \sim q(\theta)$$

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

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

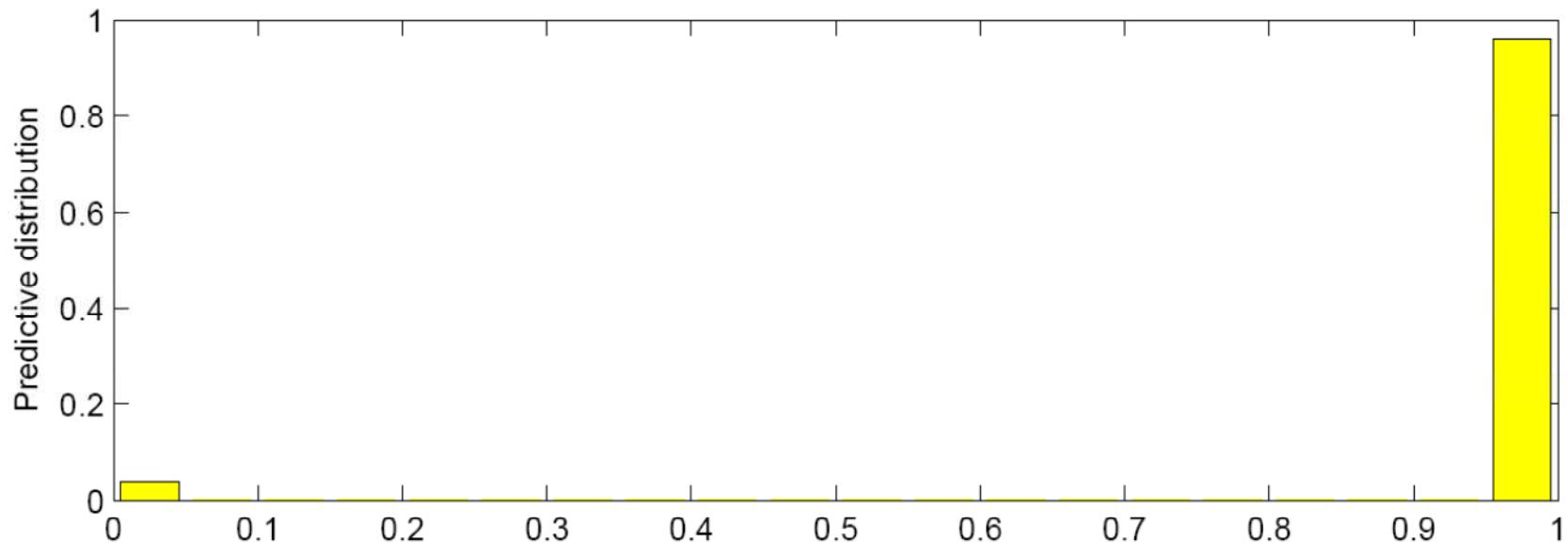
$$\begin{aligned} I &= \int f(\theta) P(\theta|D) d\theta = \frac{1}{Z} \int f(\theta) g(\theta) d\theta \\ &= \frac{\int f(\theta) g(\theta) d\theta}{\int g(\theta) d\theta} \approx \frac{\int \frac{f(\theta) g(\theta)}{q(\theta)} q(\theta) d\theta}{\int \cancel{f(\theta)} \frac{g(\theta)}{q(\theta)} q(\theta) d\theta} \quad \theta^{(i)} \sim q(\theta) \\ &\approx \frac{\sum w(\theta^{(i)}) f(\theta^{(i)})}{\sum w(\theta^{(i)})} \end{aligned}$$



# Example: Logistic Regression

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

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



# Next lecture

In the next lecture, we consider a generalization of logistic regression, with many logistic units, called multi-layer perceptron (MLP). MLPs are the most commonly used type of artificial neural networks and are an important component of state-of-the-art speech recognition.