Hasso Plattner Institut
Digital Engineering · Universität Potsdam

Introduction to Probabilistic Machine Learning

Ralf Herbrich

Inference & Decision Making

Overview

1. Inference Methods
   - Bayesian Inference
   - Maximum Likelihood Estimation
2. Decision Making

Overview

1. **Inference Methods**
   - Bayesian Inference
   - Maximum Likelihood Estimation
2. Decision Making

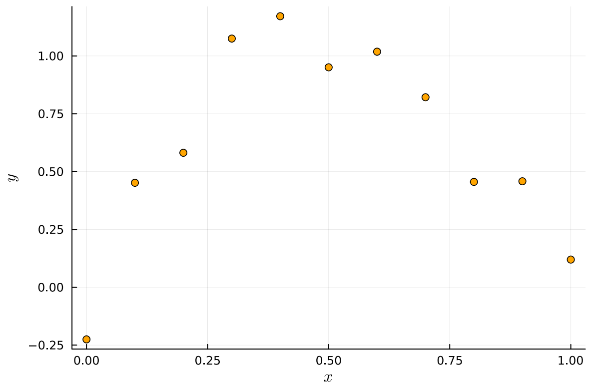
Probabilistic Machine Learning: Terminology

- **Given**:
  1. **Training Data**: $D \in (\mathcal{X} \times \mathcal{Y})^n$ of n (labelled) examples from the input space $\mathcal{X}$ and output space $\mathcal{Y}$
     - Binary Classification: $\mathcal{Y} = \{0,1\}$
     - Regression: $\mathcal{Y} = \mathbb{R}$
  2. **Prior belief over functions from $\mathcal{X}$ to $\mathcal{Y}$**: $p(f), \quad f \in \mathcal{F}$
     - Space of functions, $\mathcal{F}$, is also called *hypothesis space*.
  3. **Likelihood of function**: $p(D|f) =: \ell(f)$
     - Link between data and functions
     - Normalizes over D but not over f – never say "likelihood of data"!
     - Models all assumptions how data/labels are generated from a function
- **Key Questions in Machine Learning**:
  - **Prediction**: What is $p(y|x, D)$ for an example x and having seen D?
  - **Decision Making**: What prediction $\hat{y}$ shall be made for an example x having seen D?

Probabilistic Machine Learning: Polynomial Regression

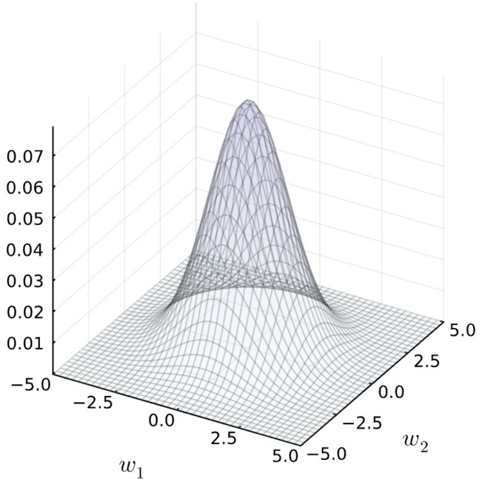
Training Data

$D \in \mathbb{R}^2$



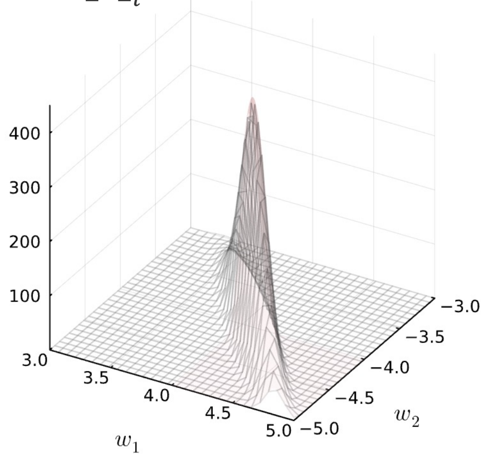
Prior

$$p(\boldsymbol{w}) = \mathcal{N}(w_1; 0,1) \cdot \mathcal{N}(w_2; 0,1)$$



Likelihood

$$\ell(\boldsymbol{w}) = \prod_i \mathcal{N}\left(y_i; w_1 x_i + w_2 x_i^2, \sigma^2\right)$$



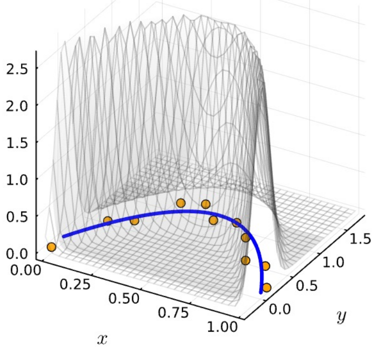
$$\boxed{f(x; \boldsymbol{w}) = w_1 \cdot x + w_2 \cdot x^2}$$

Predictions and Predictive Distributions

- Predictive distribution using the sum rule of probability

$$p(y|x,D) = \int p(y|x,f) \cdot p(f|D)\, df$$

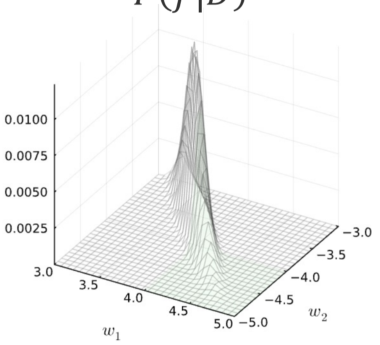
$P(y|x,f)$



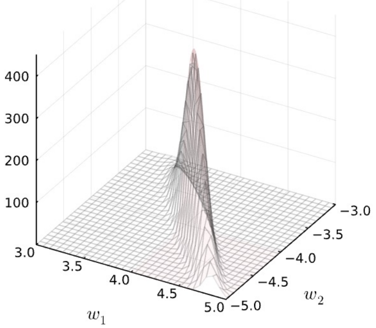
- But how do we get $p(f|D)$? Bayes' rule!

$$p(f|D) = \frac{p(D|f) \cdot p(f)}{p(D)}$$

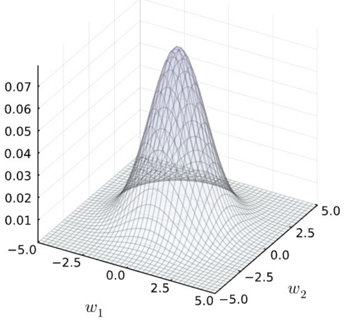
$P(f|D)$



$P(D|f)$



$P(f)$



Overview

1. Inference Methods
   - **Bayesian Inference**
   - Maximum Likelihood Estimation
2. Decision Making

Probabilistic Machine Learning: Bayesian Inference

- **Two computational difficulties**:
  1. **Posterior** $p(f|D)$ requires the *multiplication* of likelihood with prior which often results in a distribution which is no longer in a family with very few parameters.

$$p(f|D) = \frac{p(D|f) \cdot p(f)}{p(D)} \propto \ell(f) \cdot p(f)$$

  2. **Predictive distribution** $p(y|x, D)$ requires the *summation* of the data distribution over all prediction functions. This is only feasible for a small number of parametric distributions.

$$p(y|x, D) = \int p(y|x, f) \cdot p(f|D)\, df$$

Probability Distributions: Conjugacy

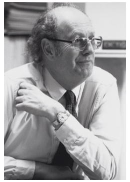
- **Bayes Rule for Random Variables**. *For any probability distribution p over two random variables X and Θ, it holds*

$$\underbrace{p(\theta|x)}_{\text{Posterior}} = \frac{\overbrace{\underbrace{p(x|\theta)}_{\text{Likelihood}} \cdot \underbrace{p(\theta)}_{\text{Prior}}}^{p(x,\theta)}}{p(x)}$$

- **Conjugacy**. *A family $\{p(x,\theta)\}_{x,\theta}$ is conjugate if the posterior $p(\theta|x)$ is part of the same family as the prior $p(\theta)$ for any value of x.*

Likelihood $p(x\|\theta)$	**Prior** $p(\theta)$	**Posterior** $p(\theta\|x)$
$\mathrm{Ber}(x;\theta)$	$\mathrm{Beta}(\theta;\alpha,\beta)$	$\mathrm{Beta}\big(\theta;\alpha+x,\beta+(1-x)\big)$
$\mathrm{Bin}(x;n,\theta)$	$\mathrm{Beta}(\theta;\alpha,\beta)$	$\mathrm{Beta}\big(\theta;\alpha+x,\beta+(n-x)\big)$
$\mathcal{N}(x;\theta,\sigma^2)$	$\mathcal{N}(\theta;m,s^2)$	$\mathcal{N}\left(\theta;x\cdot\frac{s^2}{s^2+\sigma^2}+m\cdot\frac{\sigma^2}{s^2+\sigma^2},s^2\cdot\frac{\sigma^2}{s^2+\sigma^2}\right)$

- **Big Advantage**: Computing the exact posterior is computationally efficient!



Howard Raiffa
(1924 – 2016)

Robert Osher Schlaifer
(1914 – 1994)

Normal Distribution: Representations

- **Two Parameterizations (for different purposes)**:
  - **Scale-Location Parameters**

$$\mathcal{N}(x;\mu,\sigma^2) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

  - **Natural Parameters**

$$\mathcal{G}(x;\tau,\rho) = \sqrt{\frac{\rho}{2\pi}}\cdot\exp\left(-\frac{\tau^2}{2\rho}\right)\cdot\exp\left(\tau\cdot x-\rho\cdot\frac{x^2}{2}\right)$$

Two divisions only!

- **Conversions**

$$\mathcal{N}(x;\mu,\sigma^2) = \mathcal{G}\left(x;\frac{\mu}{\sigma^2},\frac{1}{\sigma^2}\right). \qquad \mathcal{G}(x;\tau,\rho) = \mathcal{N}\left(x;\frac{\tau}{\rho},\frac{1}{\rho}\right)$$

- **Two Special Cases (in terms of σ^2)**

1. **Constant function**: $c(x) = 1 = \exp(0) = \lim_{\sigma^2\to\infty}\exp\left(-\frac{x^2}{\sigma^2}\right) = \frac{\mathcal{G}(x;0,0)}{\mathcal{N}(0;0,0)}$
2. **Dirac Delta**: $\delta(x) = \lim_{\sigma^2\to 0}\mathcal{N}(x;0,\sigma^2)$

Normal Distributions and the Product Rule

- **Theorem (Multiplication)**. *Given two one-dimensional Gaussian distributions* $\mathcal{G}(x;\tau_1,\rho_1)$ *and* $\mathcal{G}(x;\tau_2,\rho_2)$ *we have*

Gaussian density

$$\mathcal{G}(x;\tau_1,\rho_1)\cdot\mathcal{G}(x;\tau_2,\rho_2)=\mathcal{G}(x;\boxed{\tau_1+\tau_2},\boxed{\rho_1+\rho_2})\cdot\boxed{\mathcal{N}(\mu_1;\mu_2,\sigma_1^2+\sigma_2^2)}$$

Additive updates!

- **Theorem (Division)**. *Given two one-dimensional Gaussian distributions* $\mathcal{G}(x;\tau_1,\rho_1)$ *and* $\mathcal{G}(x;\tau_2,\rho_2)$ *we have*

Gaussian density

$$\frac{\mathcal{G}(x;\tau_1,\rho_1)}{\mathcal{G}(x;\tau_2,\rho_2)}=\mathcal{G}(x;\boxed{\tau_1-\tau_2},\boxed{\rho_1-\rho_2})\cdot\frac{1}{\boxed{\mathcal{N}\left(\frac{\tau_1-\tau_2}{\rho_1-\rho_2};\frac{\tau_2}{\rho_2},\frac{1}{\rho_1-\rho_2}+\frac{1}{\rho_2}\right)}}$$

Subtractive updates!

Probability Distributions: Exponential Family

- **Exponential Family**. *A family of distributions is said to belong to the exponential family if the probability density/mass function in terms of the parameterisation* $\boldsymbol{\theta}$ *is*

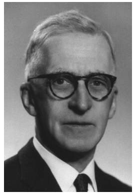
$$p(x) = \exp\left(\sum_i \eta_i(\boldsymbol{\theta}) \cdot T_i(x) - A(\boldsymbol{\theta})\right)$$

  - The η_i's are called canonical parameters and the T_i's are called sufficient statistics.

Distribution $p(x)$	Canonical Parameters $\eta(\theta)$	Sufficient Statistic $T(x)$
$\mathrm{Bin}(x; n, \pi)$	$\log\left(\frac{\pi}{1-\pi}\right)$	x
$\mathrm{Beta}(\pi; \alpha, \beta)$	$[\alpha, \beta]$	$[\log(\pi), \log(1-\pi)]$
$\mathcal{N}(x; \mu, \sigma^2)$	$\left[\frac{\mu}{\sigma^2}, \frac{1}{\sigma^2}\right]$	$\left[x, -\frac{x^2}{2}\right]$

- **Big Advantage**: Closed and efficient under multiplication (Bayes' rule!)

$$p(x; \boldsymbol{\eta}_1) \cdot p(x; \boldsymbol{\eta}_2) = p(x; \boldsymbol{\eta}_1 + \boldsymbol{\eta}_2)$$



Edwin Pitman (1897 - 1993)

Georges Darmois (1888 - 1960)

Bernhard Koopman (1900 - 1991)

Overview

1. Inference Methods
   - Bayesian Inference
   - **Maximum Likelihood Estimation**
2. Decision Making

Maximum Likelihood/Maximum A-Posteriori Inference

1. **Maximum Likelihood**. Find the most *likely* function $f_{\mathrm{ML}}(D)$ given the data D and approximate $p(f|D)$ by a single point distribution around
$$f_{\mathrm{ML}}(D) = \operatorname*{argmax}_f p(D|f)$$
2. **Maximum A Posterior**. Find the most *probable* function $f_{\mathrm{MAP}}(D)$ given the data D and prior approximate $p(f|D)$ by a single point distribution around
$$f_{\mathrm{MAP}}(D) = \operatorname*{argmax}_f p(D|f) \cdot p(f)$$

- **Pros**:
    1. Learning = optimization in the hypothesis space ("gradient descent")
    2. Storing the model = storing the function parameters
- **Cons**:
    1. The posterior/likelihood is "peaked" around a single best predictor (convergence)
    2. No model uncertainty after learning from data

Sir Ronald Fisher (1890 – 1962)

Maximum Likelihood/Maximum A-Posteriori Inference

1. **Maximum Likelihood**. Find the most *likely* function $f_{\mathrm{ML}}(D)$ given the data D and approximate $p(f|D)$ by a single point distribution around
$$f_{\mathrm{ML}}(D) = \underset{f}{\operatorname{argmax}}\, p(D|f)$$
2. **Maximum A Posterior**. Find the most *probable* function $f_{\mathrm{MAP}}(D)$ given the data D and prior approximate $p(f|D)$ by a single point distribution around
$$f_{\mathrm{MAP}}(D) = \max_{f} p(D|f) \cdot p(f)$$

- **Pros**:
  1. Learning = optimization in the hypothesis space ("gradient descent")
  2. Storing the model = storing the function parameters
- **Cons**:
  1. The posterior/likelihood is "peaked" around a single best predictor (convergence)
  2. No model uncertainty after learning from data

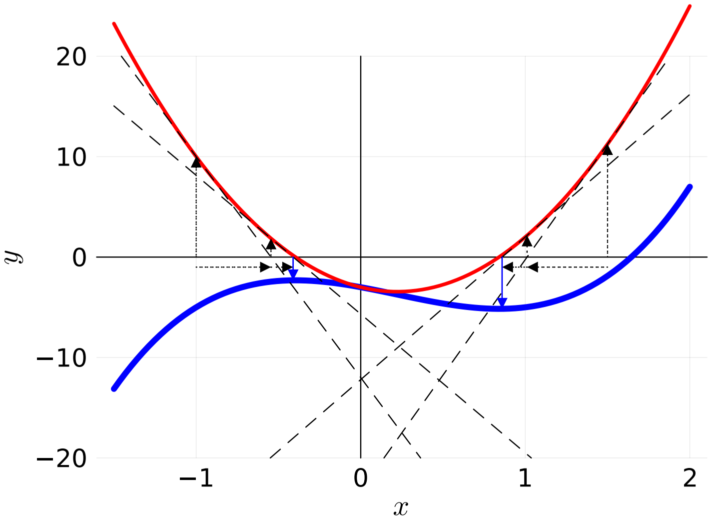
Sir Ronald Fisher (1890 – 1962)

Newton-Raphson Algorithm

- **Problem**: Find the local extrema of a function $f: \mathbb{R} \to \mathbb{R}$
- **Idea**: Find the zeros of the first derivative f' of the function!
- **Newton-Raphson Algorithm**: Approximate f' at a point x_t with a linear function $g(x) = ax + b$ and find update x_{t+1} such that $g(x_{t+1}) = 0$

$$a = f''(x_t)$$
$$b = f'(x_t) - f''(x_t) \cdot x_t$$

$$x_{t+1} = -\frac{b}{a} = \frac{f''(x_t) \cdot x_t - f'(x_t)}{f''(x_t)}$$

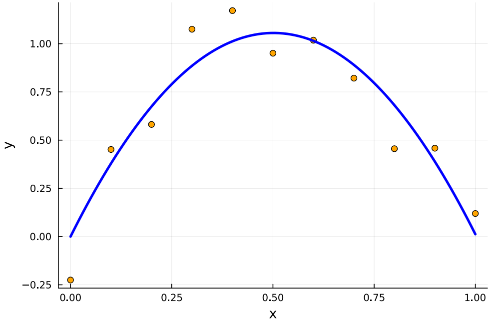


$$\boxed{x_{t+1} = x_t - \frac{f'(x_t)}{f''(x_t)}}$$

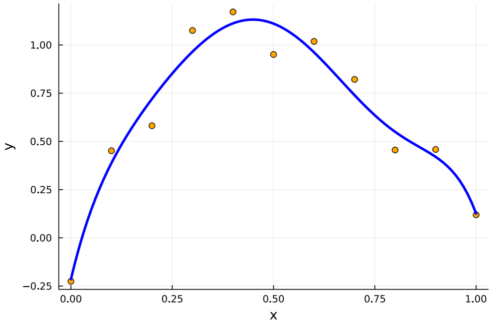
Sir Isaac Newton
(1643 – 1727)

Maximum A-Posteriori Inference: Polynomial Regression

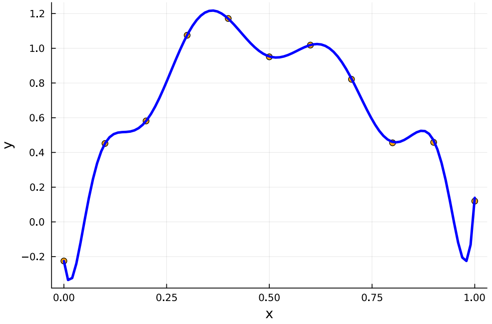
$$f(x) = w_1 x + w_2 x^2$$



$$f(x) = w_0 + w_1 x + w_2 x^2 + w_3 x^3 + w_4 x^4 + w_5 x^5 + w_6 x^6$$

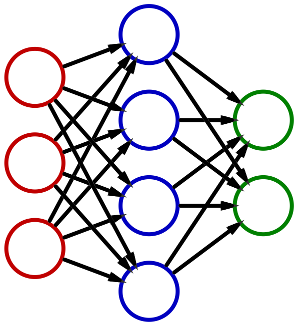


$$f(x) = \sum_{i=0}^{10} w_i \cdot x^i$$



Relation to Deep Learning

- **Deep Learning** is maximum likelihood inference on a layered function model
  - **Neural Networks**: $f(\boldsymbol{x}) = h\left(\boldsymbol{W}_L \cdots h\left(\boldsymbol{W}_2 h(\boldsymbol{W}_1 \boldsymbol{x})\right)\right)$ where h is a sigmoid
    - Number of layers: L
    - Each element of each vector is called a “neuron”
    - Each product of the inner products is called a “synapse”



- **Maximum Likelihood** optimization via gradient descent (w.r.t. $\boldsymbol{W}_1, \boldsymbol{W}_2, \ldots, \boldsymbol{W}_L$)
  - Application of the chain rule of differentiation = back propagation
  - Predicting and gradient computations are matrix multiplications; today, they are sped up using GPUs (which parallelize matrix multiplication)
- **Regularization** for the Deep Learning algorithms are equivalent to prior assumptions on $p(\boldsymbol{W}_1, \boldsymbol{W}_2, \ldots, \boldsymbol{W}_L)$!

Overview

1. Inference Methods
   - Bayesian Inference
   - Maximum Likelihood Estimation
2. **Decision Making**

Loss Functions

- **Decision Theory**: *Decision theory is concerned with the theory of making decisions based on uncertain outcomes and assigning numerical consequences to the outcome.*
  - **Answers** the second **key question of machine learning**: What prediction $\hat{y}$ shall be made for an example x having seen D?
  - Requires knowledge of the numerical consequence of taking an action (***loss*** or ***utility* function**)
- **Loss Function**: *A loss function* $l: \mathcal{Y} \times \mathcal{A} \to \mathbb{R}$ *is a function mapping the outcome space* $\mathcal{Y}$ *and an action space* $\mathcal{A}$ *to a real number representing the "cost" associated with taking the action* $a \in \mathcal{A}$ *when the true state of the world is* $y \in \mathcal{Y}$.
  - Losses are given by the domain problem; there are no "true" losses!
  - **Example**:
    1. Giving a treatment after a cancer test (economic costs?!)
    2. Deciding which advertisement to show on a search result page (bids!)

		Actions	
		treat	nothing
Outcomes	Cancer	0	**1000**
	No cancer	**1**	0

Optimal Decisions

- **Expected Loss Minimization**. *Given a predictive model $p(y|x)$ and a loss function $l: \mathcal{Y} \times \mathcal{A} \to \mathbb{R}$, the optimal action $a(x)$ is determined by minimizing the expected loss*

$$a(x) \coloneqq \mathrm{argmin}_{a \in \mathcal{A}} E_{y \sim p(y|x)}[l(y,a)]$$

- Optimal decisions require (yet again) solving an optimization problem!
  - **Example**: If $\mathcal{Y} = \mathcal{A} = \mathbb{R}$ and $l(y, a(x)) = (y - a(x))^2$ then $a(x) = E_{y \sim p(y|x)}[y]$
  - **Proof**: Taking the first derivative and setting it to zero gives

$$\frac{d}{da(x)} E_{y \sim p(y|x)}[l(y, a(x))] = \sum_y p(y|x) \cdot \frac{d}{da(x)} (y - a(x))^2$$

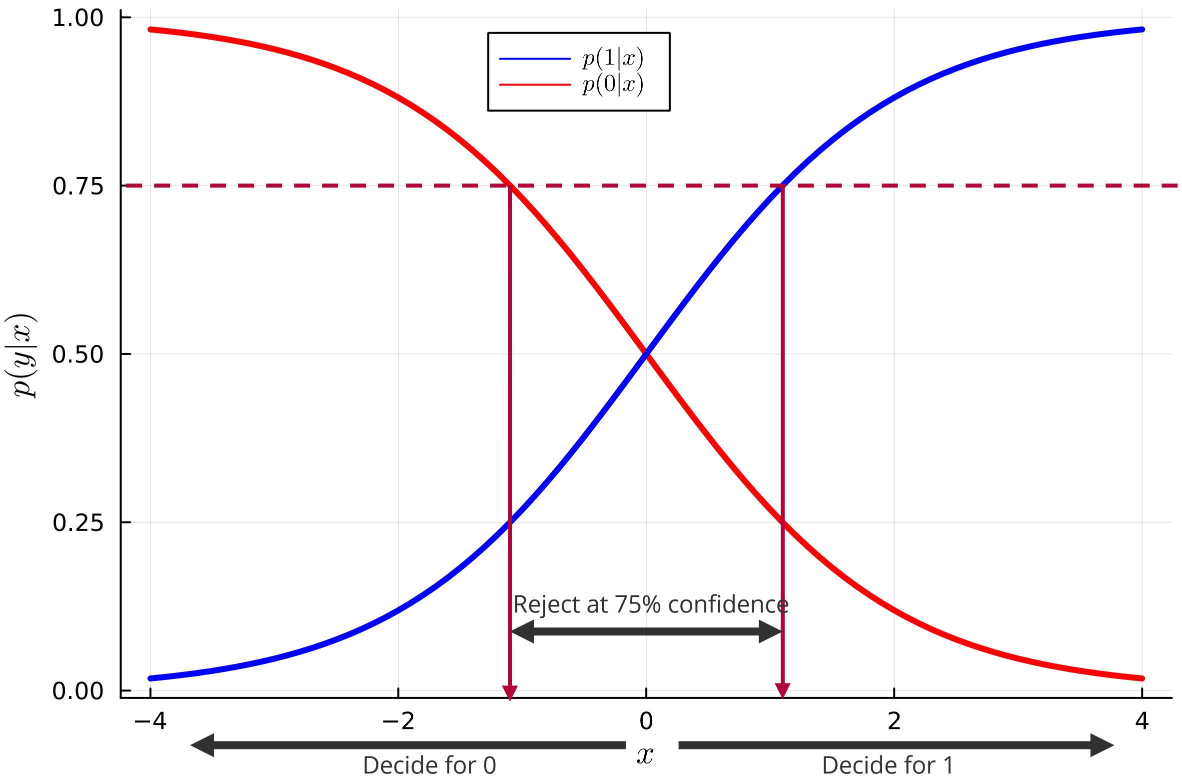
$$0 = \sum_y p(y|x) \cdot (2(a(x) - y))$$

$$0 = 2 \cdot \left(\sum_y p(y|x) \cdot a(x) - \sum_y p(y|x) \cdot y \right)$$

$$0 = a(x) - E_{y \sim p(y|x)}[y]$$

- **Reinforcement Learning** is optimizing the expected loss over an (infinite) sequence of predictions, not just for one prediction!

Example: Binary Classification



Introduction to Probabilistic Machine Learning

Unit 2 - Inference & Decision Making

Summary

1. **Inference Methods**
   - Inference is the task of inferring what we know about the plausibility of a prediction function in light of training data
   - Bayesian Inference is the only consistent inference technique requiring huge summations, but it is (usually) computationally too hard
   - Maximum Likelihood Estimation is often easier and reduces machine learning to parameter optimization – but we are losing model uncertainty
2. **Decision Making**
   - In order to make automatic decisions, we require domain-specific loss functions
   - Decision making requires optimization (again!)

See you next week!