Machine Learning: Linear Models

Forest Agostinelli
University of South Carolina

Topics Covered in This Class

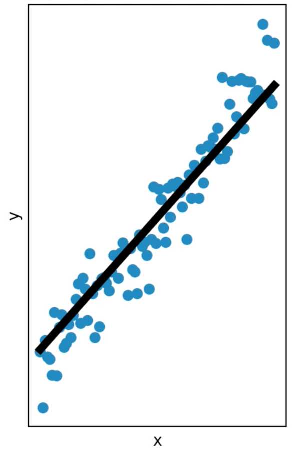
- **Part 1: Search**
  - Pathfinding
    - Uninformed search
    - Informed search
  - Adversarial search
  - Optimization
    - Local search
    - Constraint satisfaction
- **Part 2: Knowledge Representation and Reasoning**
  - Propositional logic
  - First-order logic
  - Prolog
- **Part 3: Knowledge Representation and Reasoning Under Uncertainty**
  - Probability
  - Bayesian networks
- **Part 4: Machine Learning**
  - Supervised learning
    - Inductive logic programming
    - Linear models
    - Deep neural networks
    - PyTorch
  - Reinforcement learning
    - Markov decision processes
    - Dynamic programming
    - Model-free RL
  - Unsupervised learning
    - Clustering
    - Autoencoders

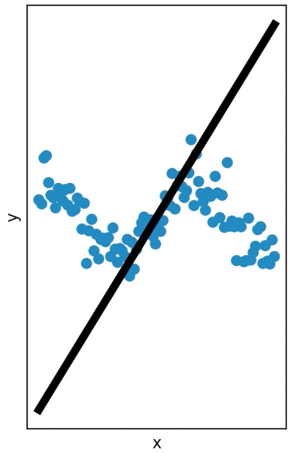
Outline

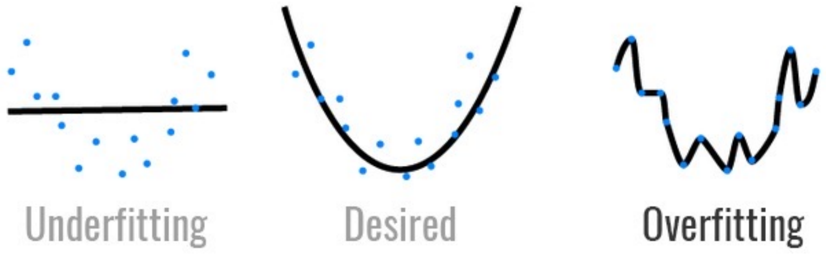
- Linear regression
    - Gradient descent
- Logistic regression (probabilistic classification)
    - Gradient descent

Hypothesis Space

- It is important that the hypothesis space be appropriate for the task at hand
- For example, if the observations have a linear input/output relationship, it is best to use a linear model
- However, if the observations have a non-linear input/output relationship, then a linear model will provide a poor explanation of the data
- On the other hand, if your hypothesis space is too large, then you may learn unnecessarily complicated hypotheses

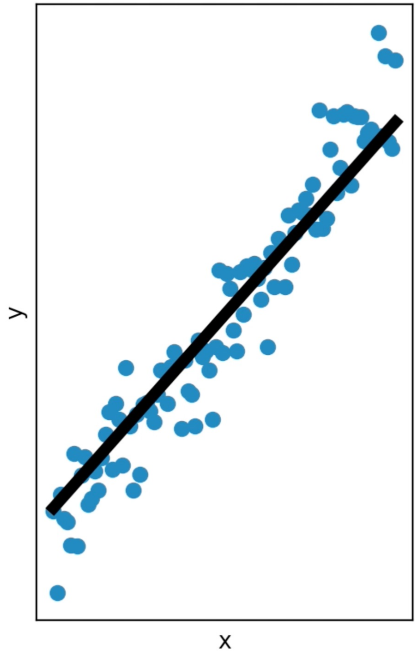






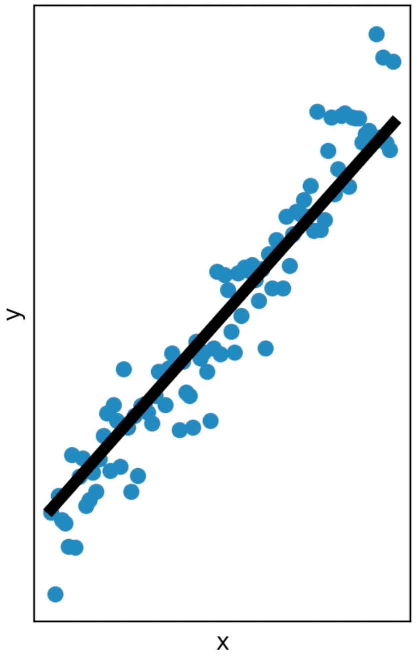
Regression and Classification

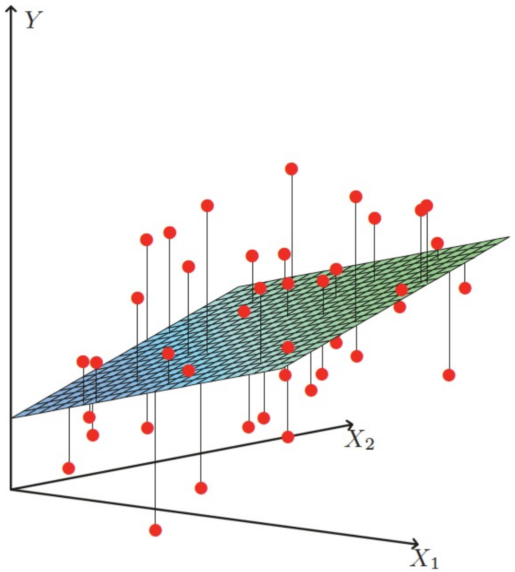
- Learn the relationship between the input $\boldsymbol{x} \in \mathbb{R}^p$ and output $\boldsymbol{y} \in \mathbb{R}^q$
  - $\boldsymbol{y} = f(\boldsymbol{x})$
- The input $\boldsymbol{x}$ is also known as the features or predictors
- **Regression**: $\boldsymbol{y}$ is a continuous variable
- **Classification**: $\boldsymbol{y}$ is a categorical variable



Linear Regression

- Limits model of input/output relationship to a line
- Learning a function $f(\boldsymbol{x}, \boldsymbol{\theta})$ with parameters $\boldsymbol{\theta}$
  - Linear model $\boldsymbol{\theta} = [\boldsymbol{w}, b]$
- $f(\boldsymbol{x}, \boldsymbol{w}, b) = \boldsymbol{w}^T \boldsymbol{x} + b = \sum_i w_i x_i + b$
- Examples (**may not truly be linear!**)
  - Yield of tomatoes as a function of health
  - Expression of a gene as a function of drug concentration





Linear Regression

- Assume 1 dimensional output
- Data
  - Inputs: $\boldsymbol{x}_1, \dots, \boldsymbol{x}_N$ where $\boldsymbol{x}_i \in \mathbb{R}^{p \times 1}$
  - Outputs: $y_1, \dots, y_N$ where $y_i \in \mathbb{R}$
- Data matrix
  - $\boldsymbol{X} \in \mathbb{R}^{N \times p}$
  - The i^{th} row contains example $\boldsymbol{x}_i$
- Vector of outputs
  - $\boldsymbol{y} \in \mathbb{R}^{N \times 1}$
- Parameters
  - $\boldsymbol{w} \in \mathbb{R}^{p \times 1}$ (weights)
  - $b \in \mathbb{R}$ (biases)
- Loss
  - $\mathcal{L}(\boldsymbol{\theta}) = \sum_n \left(y_n - f(\boldsymbol{x}, \boldsymbol{\theta})\right)^2$

Linear Regression: Analytical Solution

- $\mathcal{L}(\boldsymbol{\theta}) = \sum_n \left(y_n - f(\boldsymbol{x}, \boldsymbol{\theta})\right)^2 = ||\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y}||_2^2$
- $\nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}) = 2\boldsymbol{X}^T(\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y}) = 0$
- $\boldsymbol{w}^* = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}$

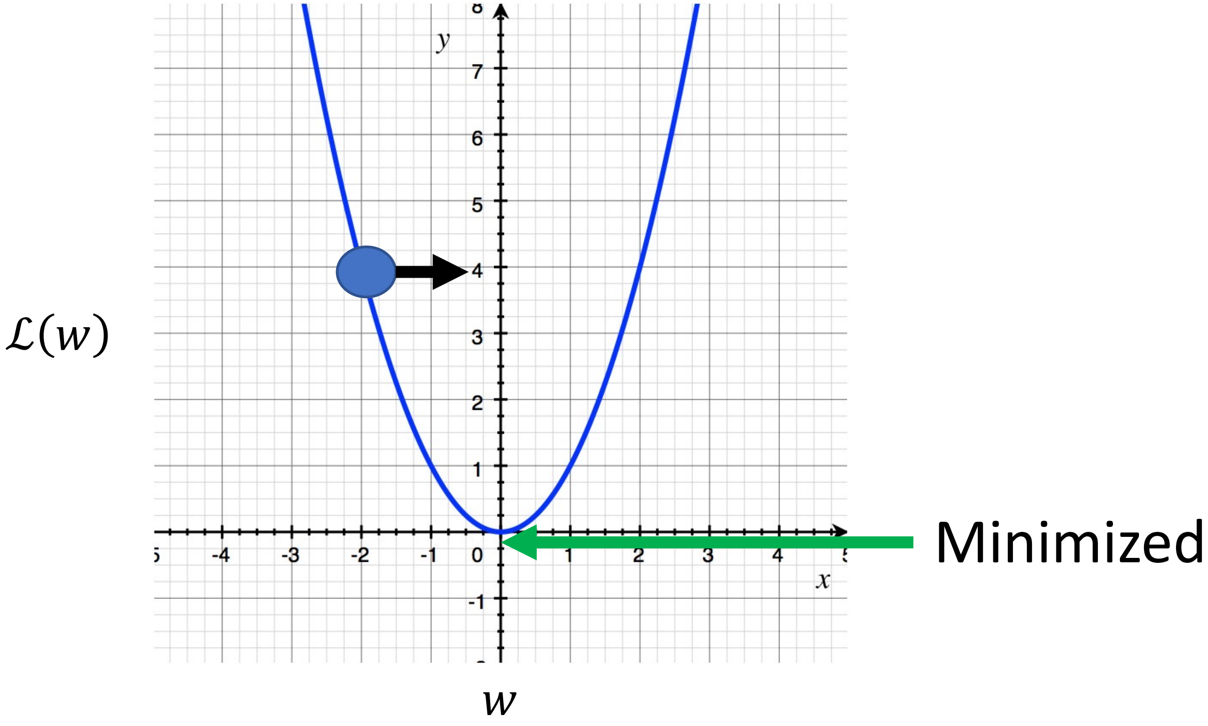
- Say there is no analytical solution, what kind of problem can this be posed as?
  - Optimization problem
  - We can do something similar to hill-climbing search where we want to minimize the loss

Linear Regression: Gradient Descent

- $\mathcal{L}(\boldsymbol{\theta}) = \frac{1}{2n}\sum_n \left(y_n - f(\boldsymbol{x},\boldsymbol{\theta})\right)^2$
- Gradient – A vector of partial derivatives
  - $\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}) = [\frac{\partial\mathcal{L}(\theta)}{\partial\theta_0}, \dots, \frac{\partial\mathcal{L}(\theta)}{\partial\theta_{p+1}}]$
  - $\boldsymbol{w} = \boldsymbol{w} - \alpha\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta})$
- Where α is the learning rate
  - This determines how big of a step we take in that direction

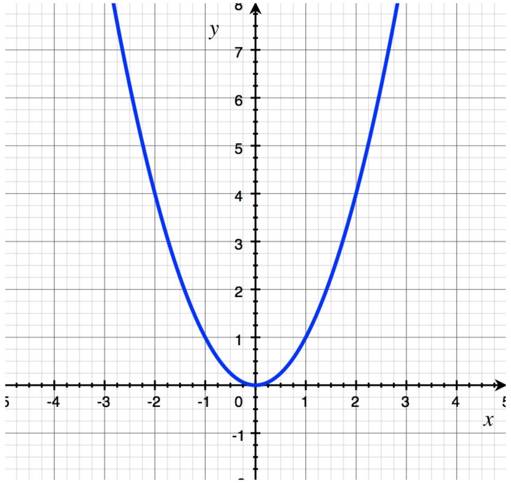
Gradient Descent: 1D Example

- One dimensional example
- $\mathcal{L}(w) = w^2$
- $\dfrac{\partial \mathcal{L}(w)}{\partial w} = 2w$



Derivatives

- The rate of change of a function at an infinitesimally small point
- $\frac{\partial f(x)}{\partial x} = \lim_{h \to 0} \frac{f(x+h)-f(x)}{h}$
- $\frac{\partial x}{\partial x} = 1$
- $\frac{\partial (xc)}{\partial x} = c$
- $\frac{\partial c}{\partial x} = 0$
- $\frac{\partial (f_1(x)+f_2(x))}{\partial x} = \frac{\partial f_1(x)}{\partial x} + \frac{\partial f_2(x)}{\partial x}$
- $\frac{\partial x^n}{\partial x} = nx^{n-1}$



Derivatives

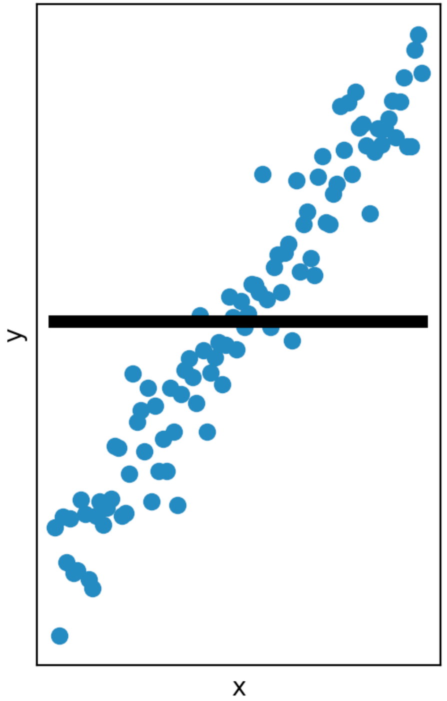
- $\frac{\partial \ln(x)}{\partial x} = \frac{1}{x}$
- $\frac{\partial a^x}{\partial x} = a^x \ln a$
- $\frac{\partial e^x}{\partial x} = e^x$
- $\frac{\partial}{\partial x} f_1(x) f_2(x) = f_1(x) \frac{\partial}{\partial x} f_2(x) + f_2(x) \frac{\partial}{\partial x} f_1(x)$
- $\frac{\partial}{\partial x} \frac{f_1(x)}{f_2(x)} = \frac{f_2(x) \frac{\partial}{\partial x} f_1(x) - f_1(x) \frac{\partial}{\partial x} f_2(x)}{f_2(x)^2}$
- $\frac{\partial}{\partial x} \frac{1}{f(x)} = -\frac{1}{f(x)^2} \frac{\partial}{\partial x} f(x)$
- $\frac{\partial}{\partial x} \sigma(x) = \frac{\partial}{\partial x} \frac{1}{1+e^{-x}}$
  - $\sigma(x)\big(1 - \sigma(x)\big) = \sigma(x)\sigma(-x)$

Derivatives: Chain Rule

- $g = u^2$
- $u = f(x)$
- $\frac{\partial g}{\partial x} = \frac{\partial g}{\partial u}\frac{\partial u}{\partial x}$

Linear Regression: 1D with No Bias

- $y_n = 3x + \epsilon_n$
- $\epsilon_n \sim \mathcal{N}(0{,}0.5)$
- $f(x_n, w) = wx_n$

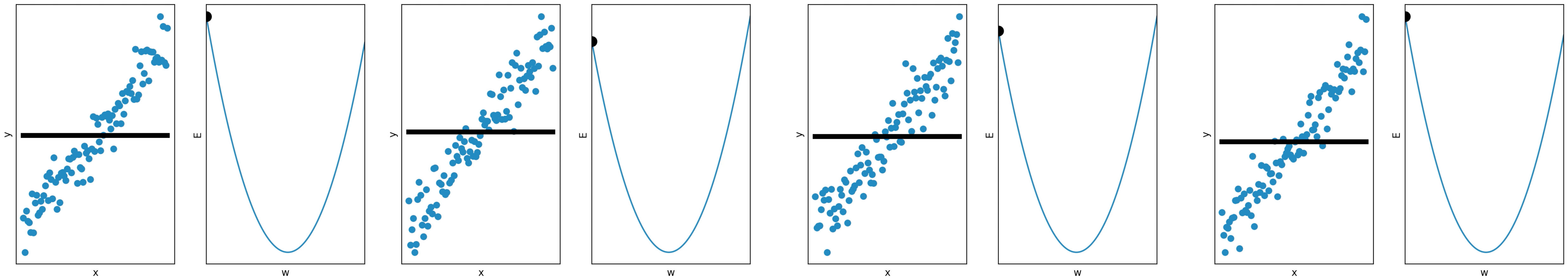


- $\mathcal{L}(w) = \frac{1}{2n}\sum_n (y_n - wx_n)^2$
- What is $\frac{\partial \mathcal{L}(w)}{\partial w}$?
- $\frac{\partial \mathcal{L}(w)}{\partial w} = -\frac{1}{n}\sum_n (y_n - wx_n)\, x_n$

Linear Regression: 1D with No Bias

- $y_n = 3x + \epsilon_n$
- $\epsilon_n \sim \mathcal{N}(0,0.5)$
- $f(x_n, w) = wx_n$

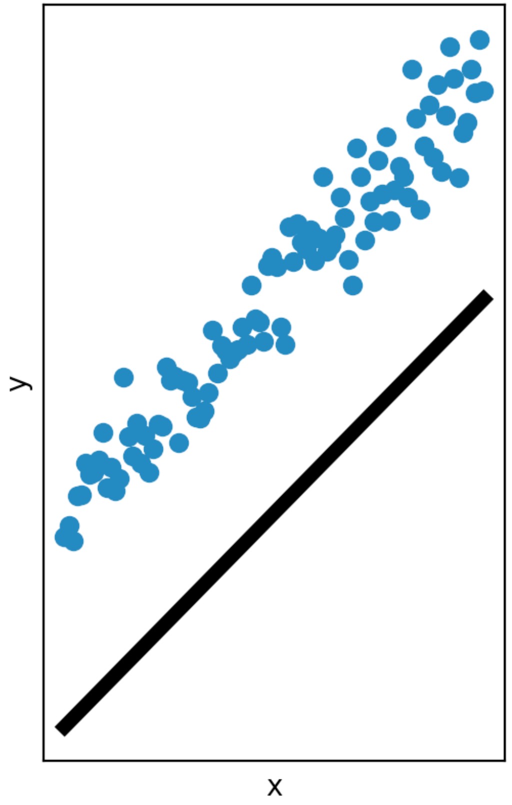
- $\mathcal{L}(w) = \frac{1}{2n}\sum_n (y_n - wx_n)^2$
- $\frac{\partial \mathcal{L}(w)}{\partial w} = -\frac{1}{n}\sum_n (y_n - wx_n)\, x_n$
- $w = w - \alpha \frac{\partial \mathcal{L}(w)}{\partial w}$



$\alpha = 1.0$ $\alpha = 0.1$ $\alpha = 5.0$ $\alpha = 10.0$

Linear Regression: 1D with Bias

- $y_n = 3x + 3 + \epsilon_n$
- $\epsilon_n \sim \mathcal{N}(0,0.5)$
- $f(x_n, w, b) = wx_n + b$

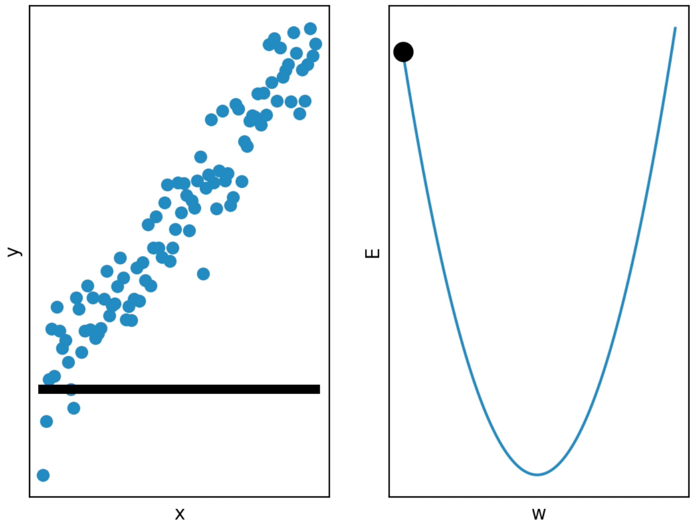


- $\mathcal{L}(w,b) = \frac{1}{2n}\sum_n (y_n - (wx_n + b))^2$
- What is $\frac{\partial \mathcal{L}(w,b)}{\partial w}$ and $\frac{\partial \mathcal{L}(w,b)}{\partial b}$?
- $\frac{\partial \mathcal{L}(w,b)}{\partial w} = -\frac{1}{n}\sum_n (y_n - (wx_n + b))\, x_n$
- $\frac{\partial \mathcal{L}(w,b)}{\partial b} = -\frac{1}{n}\sum_n (y_n - (wx_n + b))$

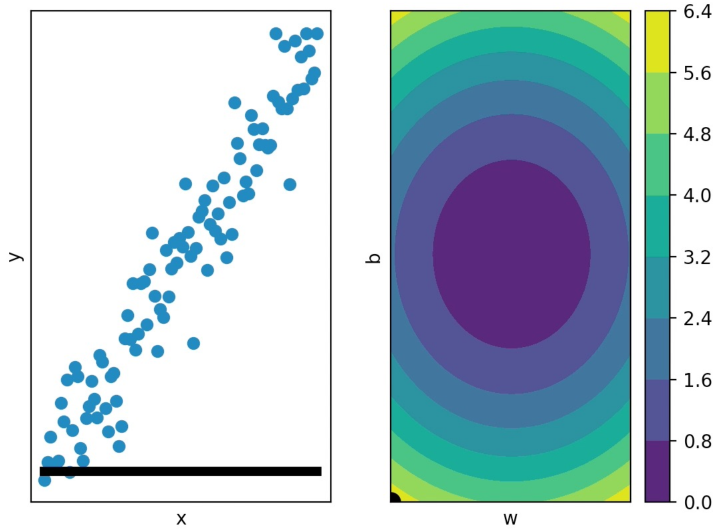
Linear Regression: 1D with Bias

- $y_n = 3x + 3 + \epsilon_n$
- $\epsilon_n \sim \mathcal{N}(0,0.5)$
- $f(x_n, w, b) = wx_n + b$

- $\mathcal{L}(w,b) = \frac{1}{2n}\sum_n (y_n - (wx_n + b))^2$
- $\frac{\partial \mathcal{L}(w,b)}{\partial w} = -\frac{1}{n}\sum_n (y_n - (wx_n + b))\, x_n$
- $\frac{\partial \mathcal{L}(w,b)}{\partial b} = -\frac{1}{n}\sum_n (y_n - (wx_n + b))$
- $w = w - \alpha \frac{\partial \mathcal{L}(w,b)}{\partial w}$
- $b = b - \alpha \frac{\partial \mathcal{L}(w,b)}{\partial b}$



No bias $\alpha = 0.5$



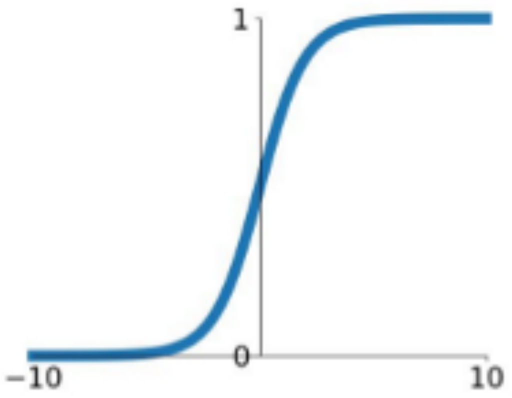
Bias $\alpha = 0.5$

Binary Classification

- We would like to differentiate between 2 classes
  - Dog/cat
  - Disease/no disease
  - Pedestrian/no pedestrian
- We are given an input vector $\boldsymbol{x}$ and want to predict y
- Suppose we compute a value, $\boldsymbol{w}_0^T\boldsymbol{x}$, for class 0 and $\boldsymbol{w}_1^T\boldsymbol{x}$ for class 1
- One way to make decisions
  - If $\boldsymbol{w}_1^T\boldsymbol{x} > \boldsymbol{w}_0^T\boldsymbol{x}$ then label this as class 1
  - Otherwise, label as class 0

Binary Classification

- However, what if we are interested in probabilistic decisions?
  - $P(y = 1|\boldsymbol{x})$
  - $P(y = 0|\boldsymbol{x}) = 1 - P(y = 1|\boldsymbol{x})$
- If values are guaranteed to be positive and have a sum greater than zero, then we can obtain a probability by dividing each value by their sum
  - Ensures normalized values are positive and sum to 1 (obeys the laws of probability)
- We can do this by exponentiating the values $\boldsymbol{w}^T\boldsymbol{x}$
  - $P(y = 1|\boldsymbol{x}) = \frac{e^{w_1^T x}}{e^{w_1^T x} + e^{w_0^T x}} = \frac{1}{1+e^{(w_0 - w_1)^T x}} = \frac{1}{1+e^{-w^T x}}$
- This gives us the logistic function
  - $\sigma(a) = \frac{1}{1+e^{-a}}$



Derivative of Logistic Function

- Show
  - $\frac{\partial}{\partial x}\sigma(x) = \frac{\partial}{\partial x}\frac{1}{1+e^{-x}} = \sigma(x)\big(1-\sigma(x)\big) = \sigma(x)\sigma(-x)$
  - $1-\sigma(x) = 1 - \frac{1}{1+e^{-x}} = \frac{1+e^{-x}}{1+e^{-x}} - \frac{1}{1+e^{-x}} = \frac{e^{-x}}{1+e^{-x}} = \frac{1}{\frac{1}{e^{-x}}+1} = \frac{1}{e^{x}+1}$
- Using
  - $\frac{\partial}{\partial x}\frac{1}{f(x)} = -\frac{1}{f(x)^2}\frac{\partial}{\partial x}f(x)$
  - $\frac{\partial(xc)}{\partial x} = c$
  - $\frac{\partial e^x}{\partial x} = e^x$
  - $\frac{\partial c}{\partial x} = 0$
  - $\frac{\partial(f_1(x)+f_2(x))}{\partial x} = \frac{\partial f_1(x)}{\partial x} + \frac{\partial f_2(x)}{\partial x}$
- $\frac{\partial}{\partial x}\frac{1}{1+e^{-x}} = -\frac{1}{(1+e^{-x})^2}\frac{\partial}{\partial x}(1+e^{-x}) = -\frac{1}{(1+e^{-x})^2}\frac{\partial}{\partial x}1 - \frac{1}{(1+e^{-x})^2}\frac{\partial}{\partial x}e^{-x}$
- $= \frac{e^{-x}}{(1+e^{-x})^2} = \frac{1}{(1+e^{-x})}\frac{e^{-x}}{(1+e^{-x})} = \sigma(x)\big(1-\sigma(x)\big)$

Likelihood

- Likelihood: the joint probability of the observed data given as a function of the parameters of a statistical model
  - Observed data: $((\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots (\boldsymbol{x}_N, y_N))$
  - Parameters: $\boldsymbol{w}$
- $l = \prod_{i=1}^{N} P(y_i|\boldsymbol{x}_i; \boldsymbol{w})$
- $P(y_i|\boldsymbol{x}_i; \boldsymbol{w})$ if $y_i = 1$
  - $\frac{1}{1+e^{-w^T x}}$
- $P(y_i|\boldsymbol{x}_i; \boldsymbol{w})$ if $y_i = 0$
  - $1 - \frac{1}{1+e^{-w^T x}}$

Maximum Likelihood

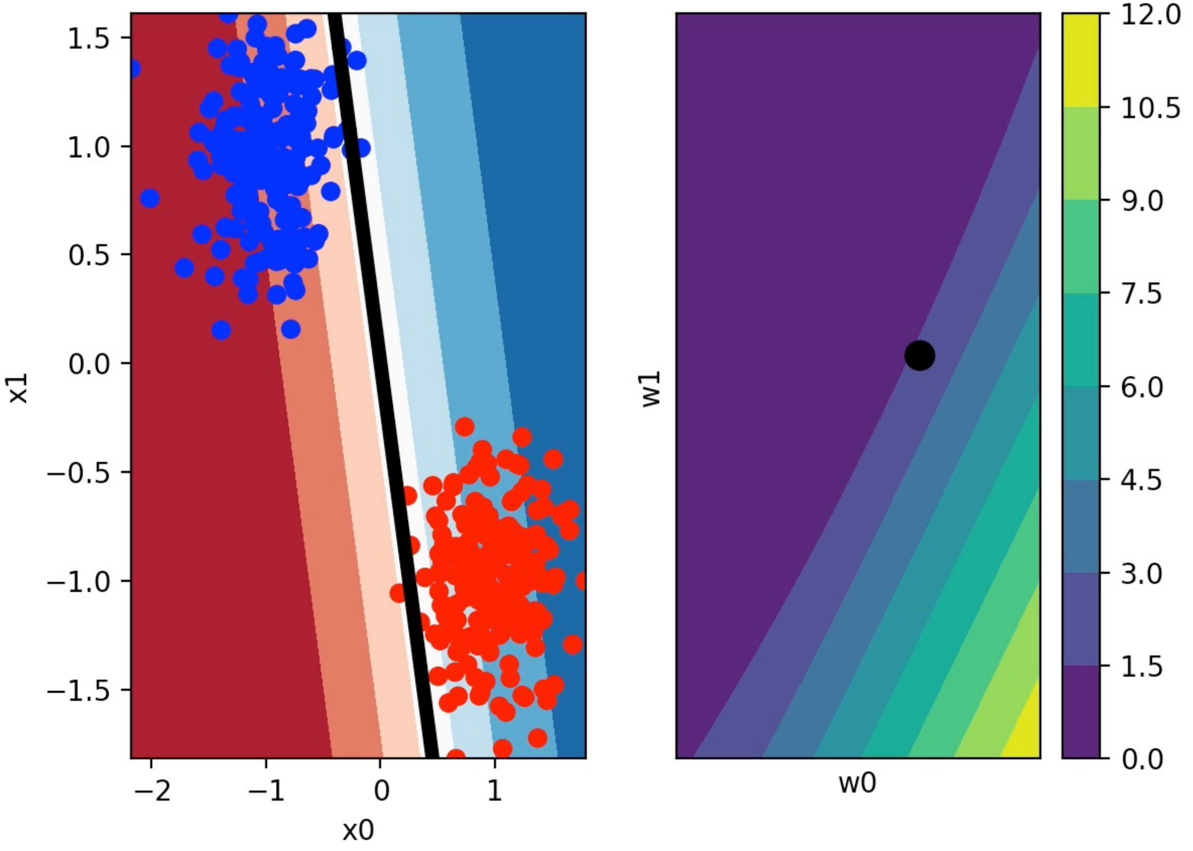
- We would like to find **w** that maximizes the likelihood
- $l = \prod_{i=1}^{N} P(y_i|\boldsymbol{x}_i; \boldsymbol{w})$
- For numerical stability, we take the log of the likelihood
- $ll = \sum_{i=1}^{N} \log P(y_i|\boldsymbol{x}_i; \boldsymbol{w})$
- $= \sum_{i=1}^{N} \mathrm{y_i} \log P(y_i = 1|\boldsymbol{x}_i; \boldsymbol{w}) + (1 - \mathrm{y_i}) \log(1 - P(y_i = 1|\boldsymbol{x}_i; \boldsymbol{w}))$
- $= \sum_{i=1}^{N} \mathrm{y_i} \log(\frac{1}{1+e^{-\boldsymbol{w}^T\boldsymbol{x}}}) + (1 - \mathrm{y_i}) \log(1 - \frac{1}{1+e^{-\boldsymbol{w}^T\boldsymbol{x}}})$

Logistic Regression: Gradient Descent

- Because of the non-linearity, we cannot find an analytical solution as we did with linear regression
- Because we want to maximize the log-likelihood, we perform gradient descent on the negative log-likelihood
- $L(\boldsymbol{w}) = -\left(\sum_{i=1}^{N} \mathrm{y_i} \log \sigma(\boldsymbol{w}^T \boldsymbol{x}_i) + (1-\mathrm{y_i}) \log(1-\sigma(\boldsymbol{w}^T \boldsymbol{x}_i))\right)$
- $\frac{\partial}{\partial w_i}\left(y \log \sigma(\boldsymbol{w}^T \boldsymbol{x}) + (1-y) \log(1-\sigma(\boldsymbol{w}^T \boldsymbol{x}))\right)$
  - $\left(\frac{y}{\sigma(\boldsymbol{w}^T \boldsymbol{x})} \sigma(\boldsymbol{w}^T \boldsymbol{x})(1-\sigma(\boldsymbol{w}^T \boldsymbol{x})) - \frac{1-y}{1-\sigma(\boldsymbol{w}^T \boldsymbol{x})} \sigma(\boldsymbol{w}^T \boldsymbol{x})(1-\sigma(\boldsymbol{w}^T \boldsymbol{x}))\right) x_i$
  - $\left(\frac{y}{\sigma(\boldsymbol{w}^T \boldsymbol{x})} - \frac{1-y}{1-\sigma(\boldsymbol{w}^T \boldsymbol{x})}\right) \sigma(\boldsymbol{w}^T \boldsymbol{x})(1-\sigma(\boldsymbol{w}^T \boldsymbol{x})) x_i$
  - $\left(\frac{y(1-\sigma(\boldsymbol{w}^T \boldsymbol{x}))}{\sigma(\boldsymbol{w}^T \boldsymbol{x})(1-\sigma(\boldsymbol{w}^T \boldsymbol{x}))} - \frac{\sigma(\boldsymbol{w}^T \boldsymbol{x})(1-y)}{\sigma(\boldsymbol{w}^T \boldsymbol{x})(1-\sigma(\boldsymbol{w}^T \boldsymbol{x}))}\right) \sigma(\boldsymbol{w}^T \boldsymbol{x})(1-\sigma(\boldsymbol{w}^T \boldsymbol{x})) x_i$
  - $\left(y - y\sigma(\boldsymbol{w}^T \boldsymbol{x}) - \sigma(\boldsymbol{w}^T \boldsymbol{x}) + y\sigma(\boldsymbol{w}^T \boldsymbol{x})\right) x_i$
  - $\left(y - \sigma(\boldsymbol{w}^T \boldsymbol{x})\right) x_i$
- $\frac{\partial L(\boldsymbol{w})}{\partial w_i} = -\sum_{i=1}^{N}\left(y - \sigma(\boldsymbol{w}^T \boldsymbol{x})\right) x_i = \sum_{i=1}^{N}(\sigma(\boldsymbol{w}^T \boldsymbol{x}) - y) x_i$

Logistic Regression: Gradient Descent

- The input is two dimensional
- $P(y = 1|\boldsymbol{x}) = \frac{1}{1+e^{-(w_0 x_0 + w_1 x_1)}}$
- No bias b
- We can plot the decision boundary between the positive and negative class as when $w_0 x_0 + w_1 x_1$ is 0
  $P(y = 1|\boldsymbol{x}) = 0.5$
  - $w_0 x_0 + w_1 x_1 = 0$
  - $x_1 = -w_0 x_0 / w_1$



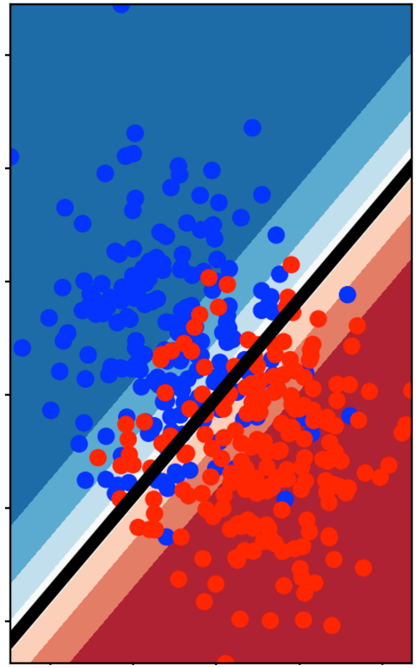
Classes Cannot Always be Perfectly Separated

- In many real-world applications, the classes are not perfectly separated
  - Data could be inherently noisy
  - The predictors may not be informative enough
  - The machine learning model may not be expressive enough
  - The training algorithm used may not be appropriate
- What could happen if your data contains more of one class than another?
  - For example, you want to learn if someone has a rare disease from medical tests
  - Since most people do not have the disease, most examples are of people that do not have the disease

Balanced vs Unbalanced Data

- One should always ensure that they balance their datasets!
  - Every gradient step can sample an equal number of states from each class
  - Or weight the contributions to the loss for each class to account for data being unbalanced
- Is this enough?

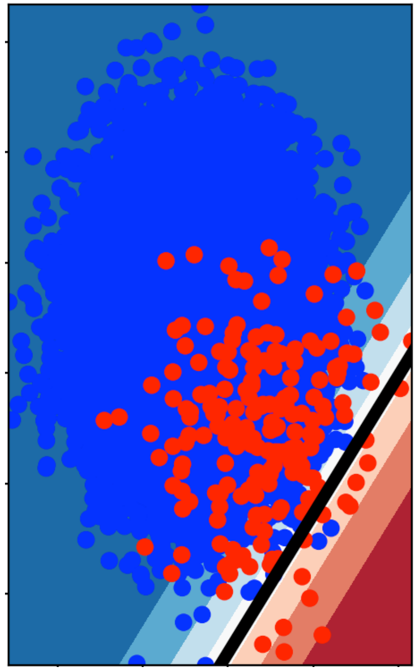
Decision boundary with balanced data



$$P(y = 1|\boldsymbol{x}) = \frac{1}{1 + e^{-(w_0 x_0 + w_1 x_1 + b)}}$$

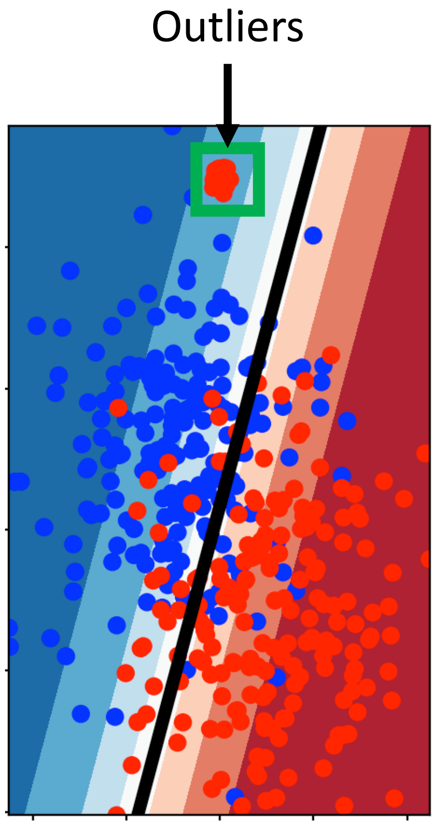
Has bias b

Decision boundary with unbalanced data



Balanced vs Unbalanced Data

- Even if the classes themselves are balanced, there may be outliers within those classes
- If these are not explicitly accounted for, the model may ignore them entirely
- For example, a rare disease that affects older people much more than children



Softmax Regression

- If we have more than two classes, we can generalize logistic regression
- We got the logistic function from this equation $\frac{e^{\boldsymbol{w}_1^T\boldsymbol{x}}}{e^{\boldsymbol{w}_1^T\boldsymbol{x}}+e^{\boldsymbol{w}_0^T\boldsymbol{x}}}$
- If we have C classes, the probability of class i is $\frac{e^{\boldsymbol{w}_i^T\boldsymbol{x}}}{\sum_{j=1}^{C} e^{\boldsymbol{w}_j^T\boldsymbol{x}}}$

Linear Models: Limitations

- Many interesting problems have a non-linear relationship between the inputs and outputs
- Linear models cannot handle these cases