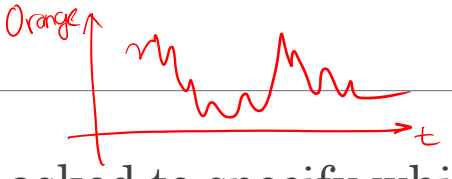
Machine Learning Basics

ADVANCED ARTIFICIAL INTELLIGENCE
JUCHEOL MOON

The Task



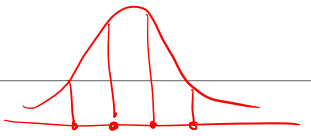
- Classification
  - The computer program is asked to specify which of k categories some input belongs to.
  - To solve this task, the learning algorithm is usually asked to produce a function f: $\mathbb{R}^n \rightarrow \{1, \cdots, k\}$

Open-set

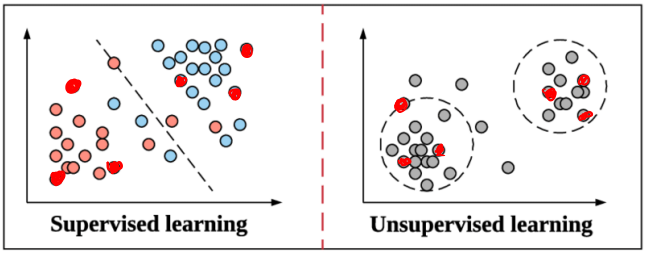
- Regression
  - The computer program is asked to predict a numerical value given some input.
  - To solve this task, the learning algorithm is asked to output a function f: $\mathbb{R}^n \rightarrow \mathbb{R}$

2

The Experience



- Unsupervised learning
  - Observing several examples of a random vector x, and attempting to learn the probability distribution $p(x)$
- Supervised learning
  - Observing several examples of a random vector x and a vector y, and learning to predict y from x, usually by estimating $p(y|x)$



3

3

Learning Paradigm

- The chain rule of probability
  - $p(x_1, x_2, x_3) = p(x_1|x_2, x_3)\,p(x_2|x_3)\,p(x_3)$
  - $p(\vec{x}) = \prod_{i}^{n} p(x_i|x_1, \cdots, x_{i-1})$

$\mathbf{x} \rightarrow \vec{x}$: vector

x : scalar

- Solving the unsupervised problem of modeling $p(x)$ by splitting it into n supervised learning problems
- Solving supervised learning problem of learning $p(y|x)$ by using unsupervised learning technologies
  - $p(y|\vec{x}) = \frac{p(\vec{x}, y)}{p(\vec{x})} = \frac{p(\vec{x}, y)}{\sum_{y'} p(\vec{x}, y')}$

$p(y)$ \ x	1	2	3
	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$

$p(y|x=1)$

4

4

Linear Regression

- The goal is to build a system that can take a vector $\vec{x} \in R^n$ as input and predict the value of a scalar $y \in R$ as its output.
  - Let $\hat{y}$ be the value that our model predicts y should take on. We define the output to be

$$\hat{y} = w_1 x_1 + w_2 x_2 + \cdots + w_n x_n = \vec{w}^T \vec{x}$$

$$\vec{w} = \begin{bmatrix} w_1 \\ \vdots \\ w_n \end{bmatrix} \quad \vec{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$$

$(n \times 1)^T$ $n \times 1$

$1 \times n$ $n \times 1$

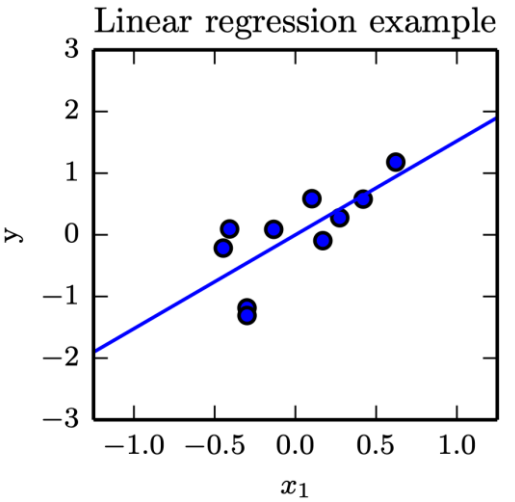
  where $\vec{w} \in \mathbb{R}^n$ is a vector of parameters.
  - Parameters are values that control the behavior of the system.

5

5

Linear Regression

- We can think of $\vec{w}$ as a set of weights that determine how each feature affects the prediction.
  - If a feature x_i receives a positive weight w_i, then increasing the value of that feature increases the value of our prediction $\hat{y}$.



6

6

Performance measure

- We are interested in how well the machine learning algorithm performs on data that it has not seen before, since this determines how well it will work when deployed in the real world.
  - We therefore evaluate these performance measures using a **test set** of data that is separate from the data used for training the machine learning system.

Training	Test

7

7

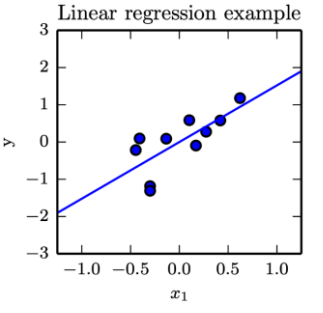
Linear Regression

- One way of measuring the performance of the model

Mean Squared Error (MSE)

$$MSE = \frac{1}{m}\sum_i (\hat{y} - y)_i^2 = \frac{1}{m}\|\hat{y} - y\|_2^2$$

where m is the number of examples



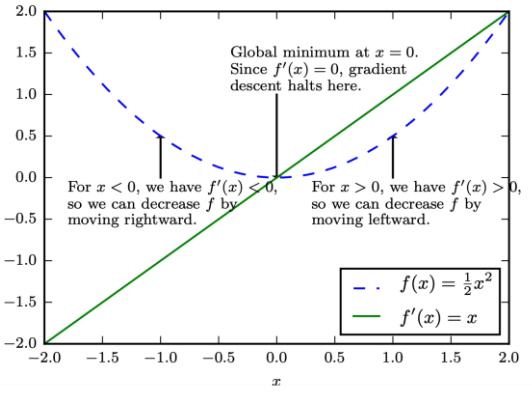
- We need to design an algorithm that will improve the weights $\vec{w}$ in a way that reduces MSE_{test}
  - Minimize MSE_{train} $\Rightarrow$ Minimize MSE_{test} (Assumption)

8

8

Gradient-Based Optimization

- The function we want to minimize (or maximize) is called the **objective** function.
  - $x^* = \arg\min_x f(x)$
  - Quadratic objective function
    - Find x such that $f'(x) = 0$

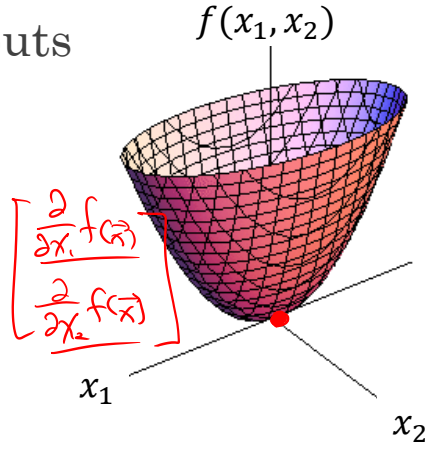


9

9

Gradient-Based Optimization

- The function we want to minimize (or maximize) is called the **objective** function.
  - For functions with multiple inputs
  - $\vec{x}^* = \arg\min_{\vec{x}} f(\vec{x})$
  - Quadratic objective function
    - $\nabla_{\vec{x}} f(\vec{x}) = \begin{bmatrix} \frac{\partial}{\partial x_1} f(\vec{x}) \\ \vdots \\ \frac{\partial}{\partial x_n} f(\vec{x}) \end{bmatrix}$
    - Find x such that $\nabla_{\vec{x}} f(\vec{x}) = \vec{0}$



10

10

Notations

- Scalars

  $a, b, c, \cdots, x, y, z$

- Vectors (a, b, x, y, z in text books)

  $\vec{a}, \vec{b}, \cdots, \vec{x}, \vec{y}, \vec{z}$

- Matrices (A, B, X, Y ⋯)

  $\vec{A}, \vec{B}, \cdots, \vec{X}, \vec{Y}, \vec{Z}$

11

11

Dimensions

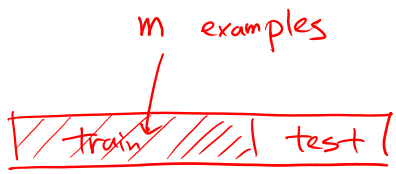
- $\vec{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$ $(n \times 1)$ $\qquad \vec{w} = \begin{bmatrix} w_1 \\ \vdots \\ w_n \end{bmatrix}$
- $\hat{y} = w_1 x_1 + w_2 x_2 + \cdots + w_n x_n = \vec{w} \cdot \vec{x} = \vec{w}^T \vec{x}$
- For m examples,
  - $\vec{X} = \begin{bmatrix} x_1^{(1)} & \cdots & x_n^{(1)} \\ & & \\ x_1^{(m)} & & x_n^{(m)} \end{bmatrix}$ $\qquad \vec{y} = \begin{bmatrix} y^{(1)} \\ \vdots \\ y^{(m)} \end{bmatrix}$

12

12

Dimensions

- $\hat{\vec{y}} = \begin{bmatrix} \hat{y}^{(1)} \\ \vdots \\ \hat{y}^{(m)} \end{bmatrix} = \begin{bmatrix} w_1 x_1^{(1)} + \cdots + w_n x_n^{(1)} \\ \ddots \\ w_1 x_1^{(m)} + \cdots + w_n x_n^{(m)} \end{bmatrix}$



- $\vec{w}^T \vec{X} = \begin{bmatrix} w_1 & \cdots & w_n \end{bmatrix} \begin{bmatrix} x_1^{(1)} & \cdots & x_n^{(1)} \\ & \ddots & \\ x_1^{(m)} & \cdots & x_n^{(m)} \end{bmatrix}$ $(1 \times n) \cdot (m \times n)$

- $\vec{X}\vec{w} = (m \times n)(n \times 1) = (m \times 1) = \begin{bmatrix} x_1^{(1)} & \cdots & x_n^{(1)} \\ & \ddots & \\ x_1^{(m)} & \cdots & x_n^{(m)} \end{bmatrix} \begin{bmatrix} w_1 \\ \vdots \\ w_n \end{bmatrix} = \begin{bmatrix} w_1 x_1^{(1)} + \cdots + w_n x_n^{(1)} \\ \ddots \\ w_1 x_1^{(m)} + \cdots + w_n x_n^{(m)} \end{bmatrix}$

13

Linear Regression

- Mean Squared Error
  - $MSE_{train} = \frac{1}{m} \| \hat{\vec{y}}^{train} - \vec{y}^{train} \|_2^2$
- Find $\vec{w}$ such that $\nabla_{\vec{w}} MSE_{train} = 0$ (give from train)

$$\nabla_{\vec{w}} \frac{1}{m} \| \hat{\vec{y}} - \vec{y} \|_2^2 = \frac{1}{m} \nabla_{\vec{w}} \| \vec{X}\vec{w} - \vec{y} \|_2^2 = 0$$

$$\nabla_{\vec{w}} \| \vec{X}\vec{w} - \vec{y} \|_2^2 = \nabla_{\vec{w}} (\vec{X}\vec{w} - \vec{y})^T (\vec{X}\vec{w} - \vec{y})$$

14

Linear Regression

- $\nabla_{\vec{w}}\left(\vec{X}train\vec{w} - \vec{y}train\right)^{\mathsf{T}}\left(\vec{X}train\vec{w} - \vec{y}train\right)$

$= D_{\vec{w}}\left(\vec{w}^{\mathsf{T}}\vec{X}^{\mathsf{T}} - \vec{y}^{\mathsf{T}}\right)\left(\vec{X}\vec{w} - \vec{y}\right)$

$= D_{\vec{w}}\left(\vec{w}^{\mathsf{T}}\vec{X}^{\mathsf{T}}\vec{X}\vec{w} - \vec{w}^{\mathsf{T}}\vec{X}^{\mathsf{T}}\vec{y} - \vec{y}^{\mathsf{T}}\vec{X}\vec{w} + \vec{y}^{\mathsf{T}}\vec{y}\right)$

$= D_{\vec{w}}\left(\vec{w}^{\mathsf{T}}\vec{X}^{\mathsf{T}}\vec{X}\vec{w}\right) - D_{\vec{w}}\left(\vec{w}^{\mathsf{T}}\vec{X}^{\mathsf{T}}\vec{y} + \vec{y}^{\mathsf{T}}\vec{X}\vec{w}\right)$

15

Linear Regression

- $\frac{\partial \vec{a}^{\mathsf{T}}\vec{B}\vec{a}}{\partial \vec{a}} = \left(\vec{B} + \vec{B}^{\mathsf{T}}\right)\vec{a}$
- $\nabla_{\vec{w}}\left(\vec{w}^{\mathsf{T}}\vec{X}train^{\mathsf{T}}\vec{X}train\vec{w}\right)$ ($\vec{a}^{\mathsf{T}}$, B, $\vec{a}$)

$= \left(\vec{X}^{\mathsf{T}}\vec{X} + \left(\vec{X}^{\mathsf{T}}\vec{X}\right)^{\mathsf{T}}\right)\vec{w}$

$= \left(\vec{X}^{\mathsf{T}}\vec{X} + \vec{X}^{\mathsf{T}}\vec{X}\right)\vec{w} = 2\vec{X}^{\mathsf{T}}\vec{X}\vec{w}$

16

Linear Regression

- $\frac{\partial \vec{a}^T \vec{b}}{\partial \vec{a}} = \frac{\partial \vec{b}^T \vec{a}}{\partial \vec{a}} = \vec{b}$
- $\nabla_{\vec{w}} \left(\vec{w}^T \vec{X}train^T \vec{y}train + \vec{y}train^T \vec{X}train \vec{w} \right)$

$= \vec{X}^T \vec{y} + \left(\vec{y}^T \vec{X} \right)^T = \vec{X}^T \vec{y} + \vec{X}^T \vec{y}$

$= 2 \vec{X}^T \vec{y}$

17

Linear Regression

- $\nabla_{\vec{w}} MSE_{train} = 2\vec{X}train^T \vec{X}train \vec{w} - 2\vec{X}train^T \vec{y}train = 0$

$\vec{X}^T \vec{X} \vec{w} = \vec{X}^T \vec{y}$

$\vec{w} = (\vec{X}^T \vec{X})^{-1} \vec{X}^T \vec{y}$

given

18

Generalization

- The central challenge in machine learning is that we must perform well on new & unseen inputs.
- Training error
  - Minimize MSE_{train}
- Test error (generalization error)

$$MSE_{test} = \frac{1}{m} \| \hat{y}^{test} - y^{test} \|_2^2$$

19

19

Data generating process

- i.i.d. assumptions
  - Examples in each dataset are independent from each other
  - The train set and test set are identically distributed, drawn from the same probability distribution as each other
  - p_{data}: data generating distribution
    - The same distribution is used to generate every train example and every test example

CA People	NV people	AZ people

20

20

Overfitting and Underfitting

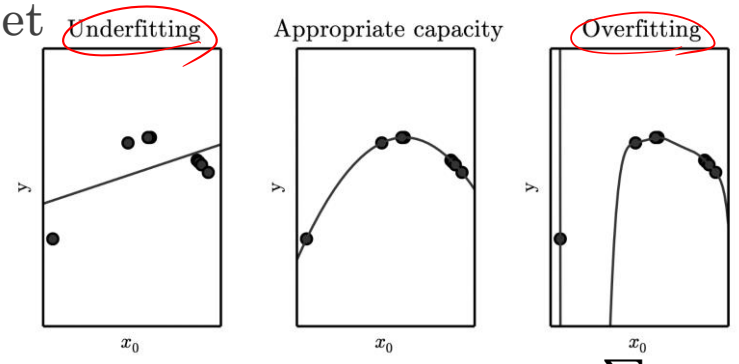
- How well a machine learning algorithm will perform are its ability to
  - Make the training error small
  - Make the gap between training and test error small
- When the model is not able to obtain a sufficiently low error value on the training set
  - Underfitting
- When the gap between the training error and test error is too large
  - Overfitting

21

21

Capacity

- Model's capacity is its ability to fit a wide variety of functions
  - Models with low capacity may struggle to fit the training set

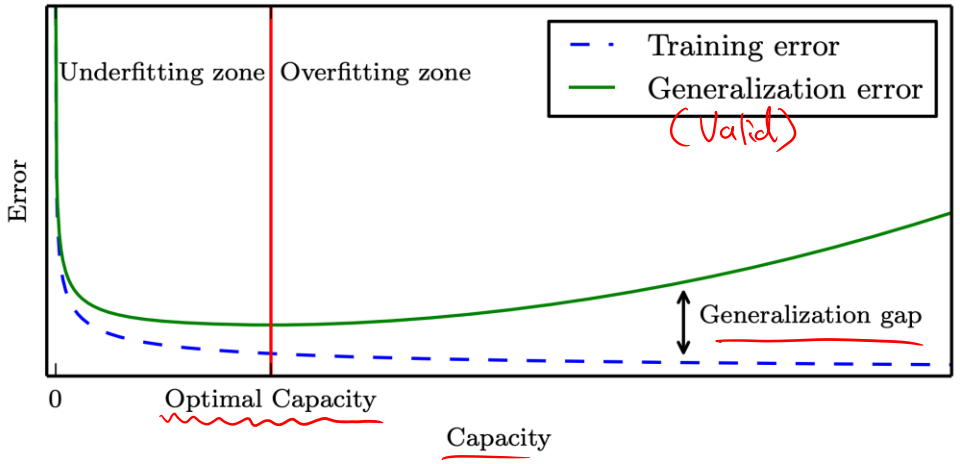


22

22

Capacity and Error

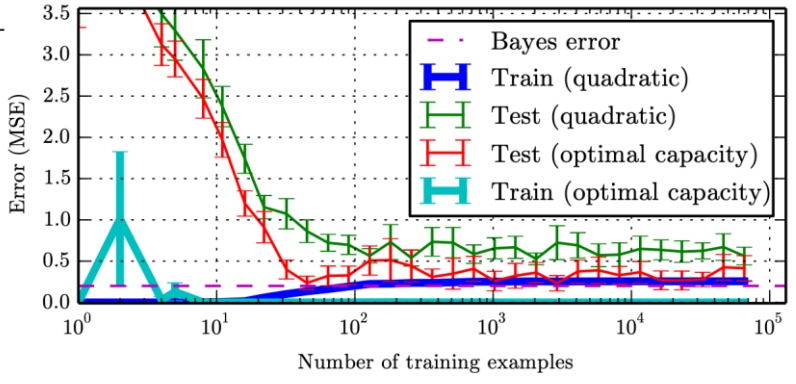
- Learning algorithm can choose from when varying the parameters in order to reduce a training objective



23

Training Set Size

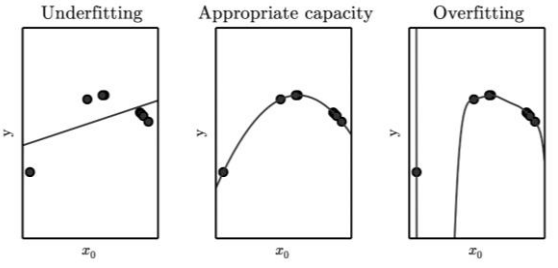
- The ideal model will incur some error on many problems, because there may still be some noise in the distribution.
  - The error incurred by an ideal model is called the Bayes error



24

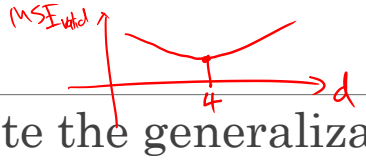
Hyperparameters

- In the polynomial regression
  - $\sum_{i=1}^{d} w_i x^i$
  - the degree of the polynomial d is a hyperparameter.
  - If learned on the training set, hyperparameters would always choose the maximum possible model capacity, resulting in overfitting



25

Validation set



- It can be used to estimate the generalization error of a learner, after the learning process has completed
  - The test examples are not used in any way to make choices about the model
  - Split the training data into two disjoint subsets (80/20)
    - One is used to learn the parameters.
    - The other subset is used to update the hyperparameters to be updated accordingly

$d=2 \rightarrow \vec{w}$

$MSE_{valid}^{d=1} = 1$

$MSE_{valid}^{d=2} = 0.9$

$MSE_{test} = 0.6$

Training	Validation	Test

train

$MSE_{test}^{d=1} =$

26

Point Estimator

- $\{x^{(1)}, \ldots, x^{(m)}\}$ i.i.d data points
- A point estimator is a function of the data
  - $\hat{\theta}_m =$
- Assume the true parameter θ is fixed but unknown
- Bias
  - $Bias(\hat{\theta}_m) =$
  - where the expectation is over the data

27

Variance and MSE

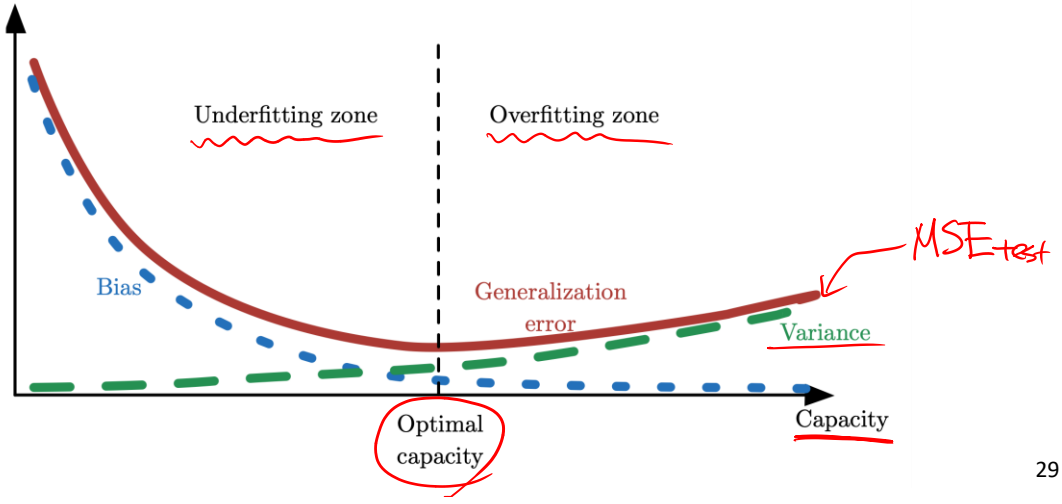
- Variance
  - $Var(\hat{\theta}_m) =$

- $MSE =$

28

Trading off Bias and Variance

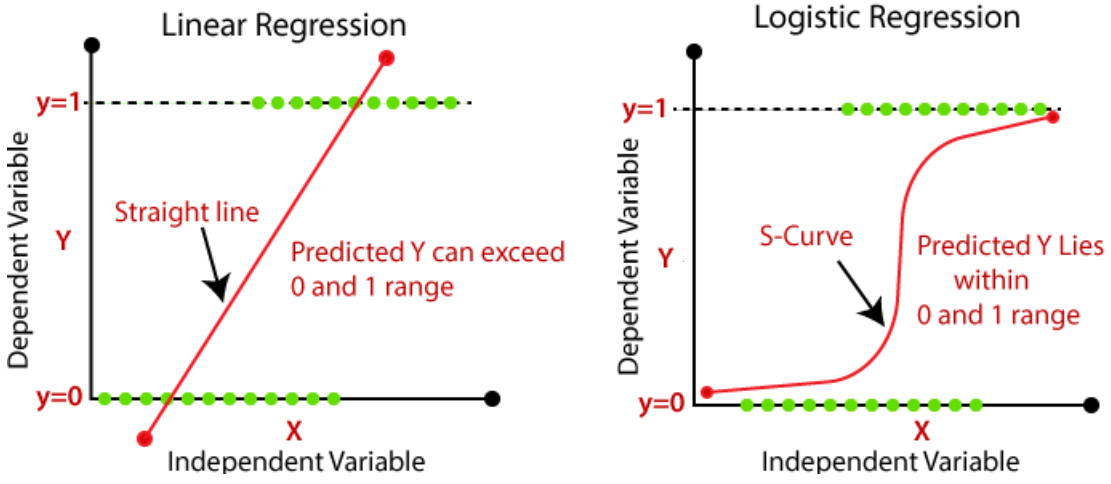
- As capacity increases, bias tends to decrease and variance tends to increase



29

Logistic Regression

- Binary classification



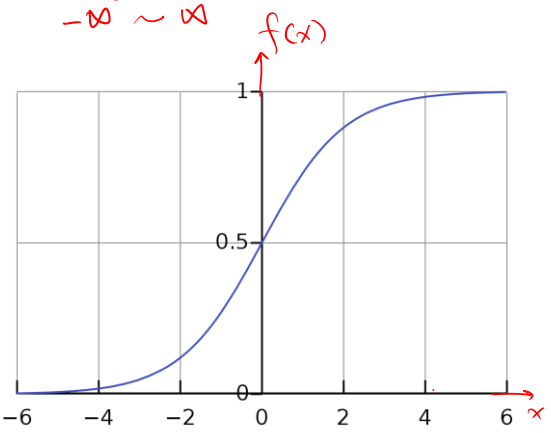
30

Logistic Regression

$\hat{y} = \vec{w}^T \vec{x}$ ⇐ linear

$-\infty \sim \infty$

- Sigmoid function
  - $f(x) = \frac{e^x}{e^x+1} = \frac{1}{1+e^{-x}}$
- Definition
  - $\hat{y} = \frac{1}{1+e^{-\vec{w}^T\vec{x}}}$
- Minimize MSE
  - $MSE = \frac{1}{m}\sum_i^m \left(y^{(i)} - \hat{y}^{(i)}\right)^2$
  - $= \frac{1}{m}\sum_i^m \left(y^{(i)} - \frac{1}{1+e^{-\vec{w}^T\vec{x}}}\right)^2$



31

31

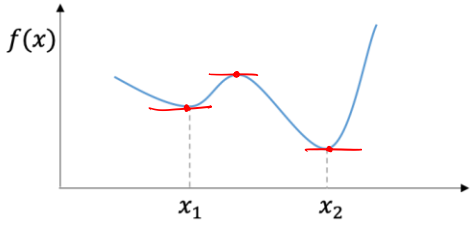
Logistic Regression

- Minimize MSE
  - $MSE = \frac{1}{m}\sum_i^m \left(y^{(i)} - \frac{1}{1+e^{-\vec{w}^T\vec{x}^{(i)}}}\right)^2$

$y^{(i)}$: 0 or 1

$\frac{1}{1+e^{-\vec{w}^T\vec{x}^{(i)}}}$: $0 < \sim < 1$

- Partial derivative
  - $\frac{\partial}{\partial w_1}\left(y - \frac{1}{1+e^{-\vec{w}^T\vec{x}}}\right)^2 =$ ✱
  - Non-convex problem

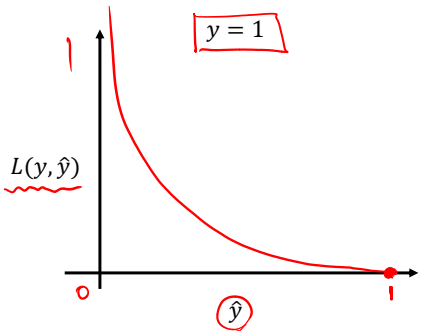


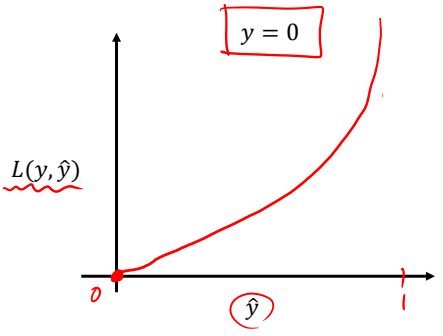
32

32

Cost function

- The cost function used by a machine learning algorithm often decomposes as a sum over training examples of some per-example loss function
  - $J(\vec{w}) = \frac{1}{m}\sum_{i}^{m} L(y^{(i)}, \hat{y}^{(i)})$
- Loss function for logistic regression



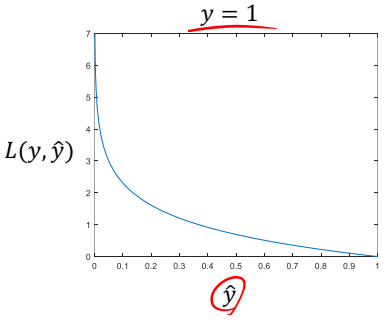


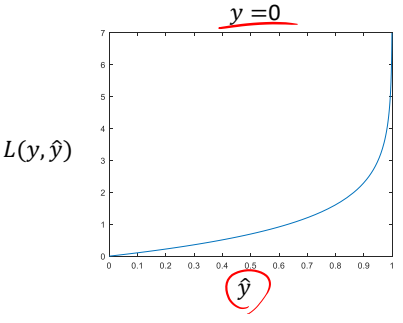
33

Loss function

- Cross-entropy function
  - $L(y, \hat{y}) = \begin{cases} -\log \hat{y} & \text{if } y=1 \\ -\log(1-\hat{y}) & \text{else } (y=0) \end{cases}$

$$= -(y \log \hat{y} + (1-y)\log(1-\hat{y}))$$

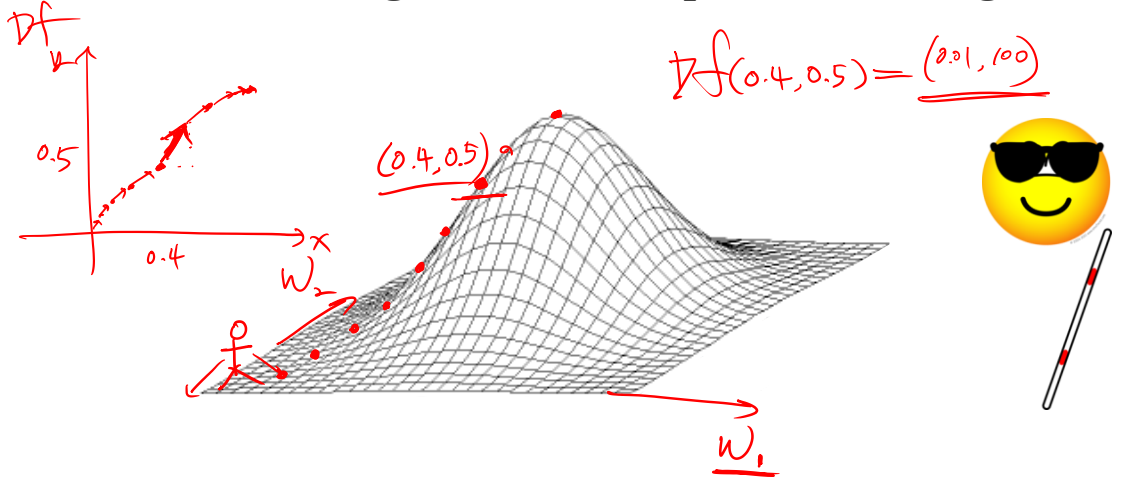




34

Gradient Descent

- Like climbing Everest in thick fog with amnesia
- The value of the gradient at a point is a tangent vector



36

Gradient Descent

- $\hat{y} = \frac{1}{1+e^{-\vec{w}^T\vec{x}}}$ where $\vec{w} = \begin{bmatrix} w_1 \\ \vdots \\ w_n \end{bmatrix}$
- Goal
  - Minimize $J(\vec{w}) = \frac{1}{m}\sum_i L(y^{(i)}, \hat{y}^{(i)})$
    $= \frac{1}{m}\sum_i^m -\left(y^{(i)}\log\hat{y}^{(i)} + (1-y^{(i)})\log(1-\hat{y}^{(i)})\right)$
- How can we update $\vec{w}$ to reduce $L(y, \hat{y})$?

$\vec{w} \leftarrow \vec{w} - \varepsilon \nabla_{\vec{w}} L(y, \hat{y})$ ε: learning rate

$w_j \leftarrow w_j - \varepsilon \frac{\partial}{\partial w_j} L(y^{(i)}, \hat{y}^{(i)})$ $\vec{w}^0 \rightarrow \vec{w}^1 \rightarrow \vec{w}^2$

37

Gradient Descent

- $L(y, \hat{y}) = -(y \log \hat{y} + (1-y) \log(1-\hat{y}))$
- Let $z = \vec{w}^T \vec{x}$ $\quad \hat{y} = \sigma(z)$ $\quad \sigma$: sigmoid function

- $\frac{\partial}{\partial w_i} L = \frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial z} \cdot \frac{\partial z}{\partial w_i} = -\left(\frac{y}{\hat{y}} - \frac{(1-y)}{(1-\hat{y})}\right) \hat{y}(1-\hat{y})\, x_i$
- $\frac{\partial L}{\partial \hat{y}} = -\left(\frac{y}{\hat{y}} - \frac{(1-y)}{(1-\hat{y})}\right)$
- $\frac{\partial \hat{y}}{\partial z} = \frac{\partial \sigma(z)}{\partial z} = \sigma(z)(1-\sigma(z)) = \hat{y}(1-\hat{y})$
- $\frac{\partial z}{\partial w_i} = x_i$

38

38

Sigmoid Function

- $\sigma(x) = \frac{1}{1+e^{-x}}$

$$\sigma'(x) = \frac{d}{dx}(1+e^{-x})^{-1}$$

$$= -(1+e^{-x})^{-2}(-e^{-x}) = \frac{e^{-x}}{(1+e^{-x})^2}$$

$$= \frac{1}{1+e^{-x}} \cdot \frac{e^{-x}}{1+e^{-x}} = \frac{1}{1+e^{-x}} \cdot \frac{1+e^{-x}-1}{1+e^{-x}}$$

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

($\frac{1}{1+e^{-x}} = \sigma(x)$, $\frac{1}{1+e^{-x}} = \sigma(x)$)

39

39

Gradient Descent

- Step 2: How can we update $\vec{w}$ to reduce $J(\vec{w})$?
  - $J(\vec{w}) = \frac{1}{m}\sum_i L(y^{(i)}, \hat{y}^{(i)})$

$$\frac{\partial}{\partial w_j} J = \frac{\partial}{\partial w_j} \frac{1}{m} \sum_i L(y^{(i)}, \hat{y}^{(i)}) = \frac{1}{m} \sum_i -\left(\frac{y}{\hat{y}} - \frac{1-y}{1-\hat{y}}\right) \hat{y}(1-\hat{y}) x_j$$

$$\vec{w} \leftarrow \vec{w} - \varepsilon \nabla_{\vec{w}} J$$

40

40

Minibatch

- The gradient may be approximately estimated using a small set of samples.
  - $\vec{X} = \left[\vec{x}^{1T} \ \ldots \ \vec{x}^{m'T} \ \vec{x}^{m'+1T} \ \ldots \ \vec{x}^{2m'T} \ \vec{x}^{2m'+1T} \ \ldots \ \vec{x}^{mT}\right]^T$ $m' \ll m$
  - $\mathbb{B} = \left[\vec{x}^{1T} \ \cdots \ \vec{x}^{m'T}\right]^T$

$$g = \frac{1}{m'} \sum_i^{m'} L(y^{(i)}, \hat{y}^{(i)})$$

$$\vec{w} \leftarrow \vec{w} - \varepsilon \nabla_{\vec{w}} g$$

41

41

Performance measurement

- A confusion matrix is a specific table layout that allows visualization of the performance of an algorithm
  - Accuracy: $ACC = \frac{TP+TN}{TP+TN+FP+FN}$
  - Error rate: $error = 1 - ACC = \frac{FP+FN}{N}$

	Predicted Positive	Predicted Negative
Actual Positive	True positive (TP)	False negative (FN)
Actual Negative	False positive (FP)	True negative (TN)

42

42

Class Imbalance Problem

	Predicted Positive	Predicted Negative
Actual Positive	True positive (TP)	False negative (FN)
Actual Negative	False positive (FP)	True negative (TN)

- Data sets with imbalanced class distributions are quite common in many real applications.
  - In credit card fraud detection, fraudulent transactions are outnumbered by legitimate transactions
    - $ACC = \frac{TP+TN}{TP+TN+FP+FN} = \frac{198}{200} = 99\%$

	Predicted legitimate	Predicted fraud
Actual legitimate	197	1
Actual fraud	1	1

43

43

Alternative Metrics

- The accuracy measure treats every class as equally important, it may not be suitable for analyzing imbalanced data sets
  - True positive rate, sensitivity
    - $TPR = \frac{TP}{TP+FN} = \frac{197}{198} \simeq 0.995$
  - True negative rate, specificity
    - $TNR = \frac{TN}{TN+FP} = \frac{1}{2} = 0.5$

	Predicted Positive	Predicted Negative
Actual Positive	True positive (TP)	False negative (FN)
Actual Negative	False positive (FP)	True negative (TN)

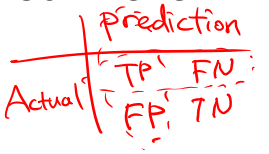
	Predicted legitimate	Predicted fraud
Actual legitimate	197	1
Actual fraud	1	1

44

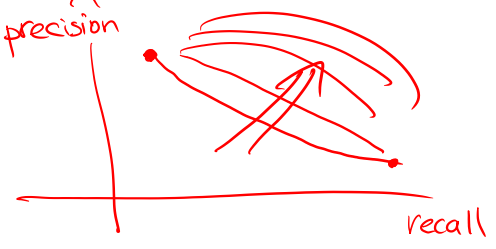
44

Alternative Metrics

- Successful detection of true class is considered more significant than detection of false class.



- $F_\beta = \frac{(\beta^2+1)TP}{(\beta^2+1)TP+\beta^2 FP+FN}$
  - If $\beta \to \infty$, $F_\beta = \frac{(1+\frac{1}{\beta^2})TP}{(1+\frac{1}{\beta^2})TP + 1 \cdot FP + \frac{1}{\beta^2}FN} = \frac{TP}{TP+FP}$
    - Precision, $p = \frac{TP}{TP+FP}$
  - If $\beta = 0$, $F_\beta = \frac{TP}{TP+FN}$
    - Recall, $r = \frac{TP}{TP+FN}$



45

45