ECE595 / STAT598: Machine Learning I
Course Overview

Spring 2020

Stanley Chan

School of Electrical and Computer Engineering
Purdue University

PURDUE
UNIVERSITY

Machine Learning: Your Interpretation?

https://www.kaggle.com/benhamner/nips-2015-papers/discussion/22778

Machine Learning: Your Interpretation?

https://www.kaggle.com/benhamner/nips-2015-papers/discussion/22778

Machine Learning: Your Interpretation?

https://www.kaggle.com/benhamner/nips-2015-papers/discussion/22778

Machine Learning: Your Interpretation?

https://www.kaggle.com/benhamner/nips-2015-papers/discussion/22778

Machine Learning: Your Interpretation?

https://www.kaggle.com/benhamner/nips-2015-papers/discussion/22778

Machine Learning: Your Interpretation?

https://www.kaggle.com/benhamner/nips-2015-papers/discussion/22778

Machine Learning: Your Interpretation?

https://www.kaggle.com/benhamner/nips-2015-papers/discussion/22778

8

Machine Learning: Your Interpretation?

https://www.kaggle.com/benhamner/nips-2015-papers/discussion/22778

Elements of Learning?

Elements of Learning?

- Data

Elements of Learning?

- Data
- Computer

Elements of Learning?

- Data
- Computer
- Algorithm

What is Learning? What is NOT learning?

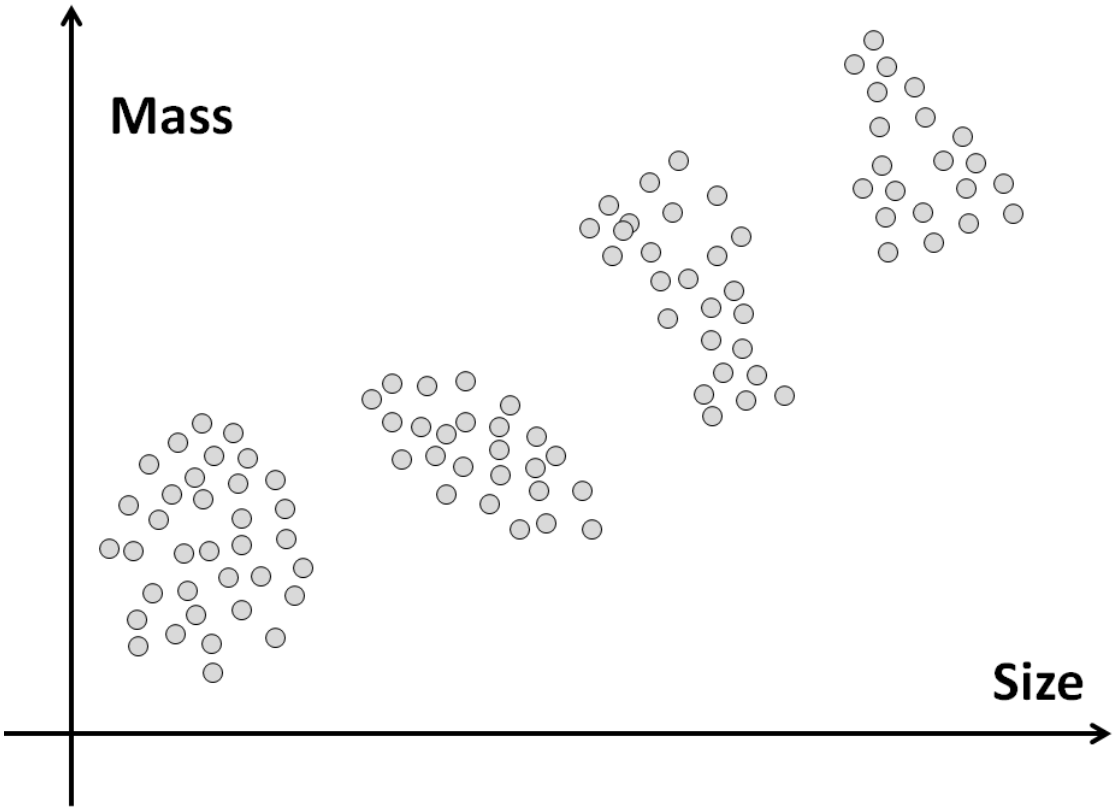
What is Learning? What is NOT learning?

What is Learning? What is NOT learning?

- You are given a bag of US coins.
- Your task: Build a classifier.
- Four classes: Penny, Nickel, Dime, Quarter.

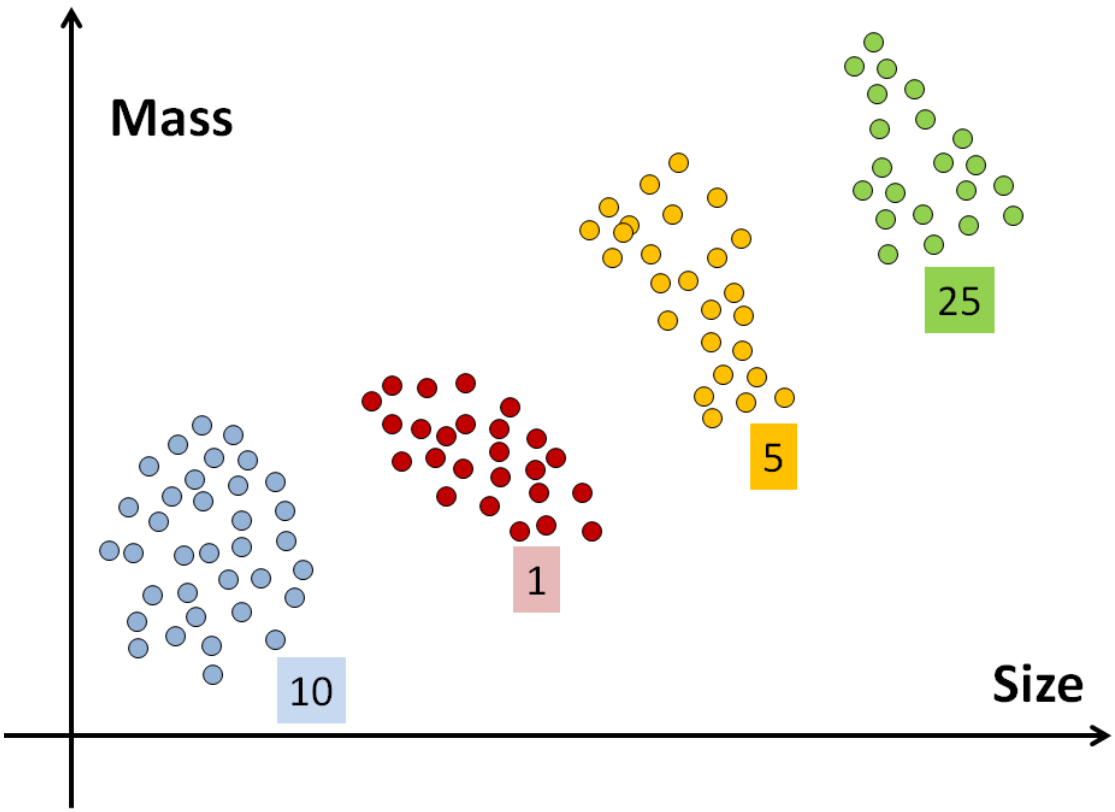
Approach 1: Learning

Approach 1: Learning



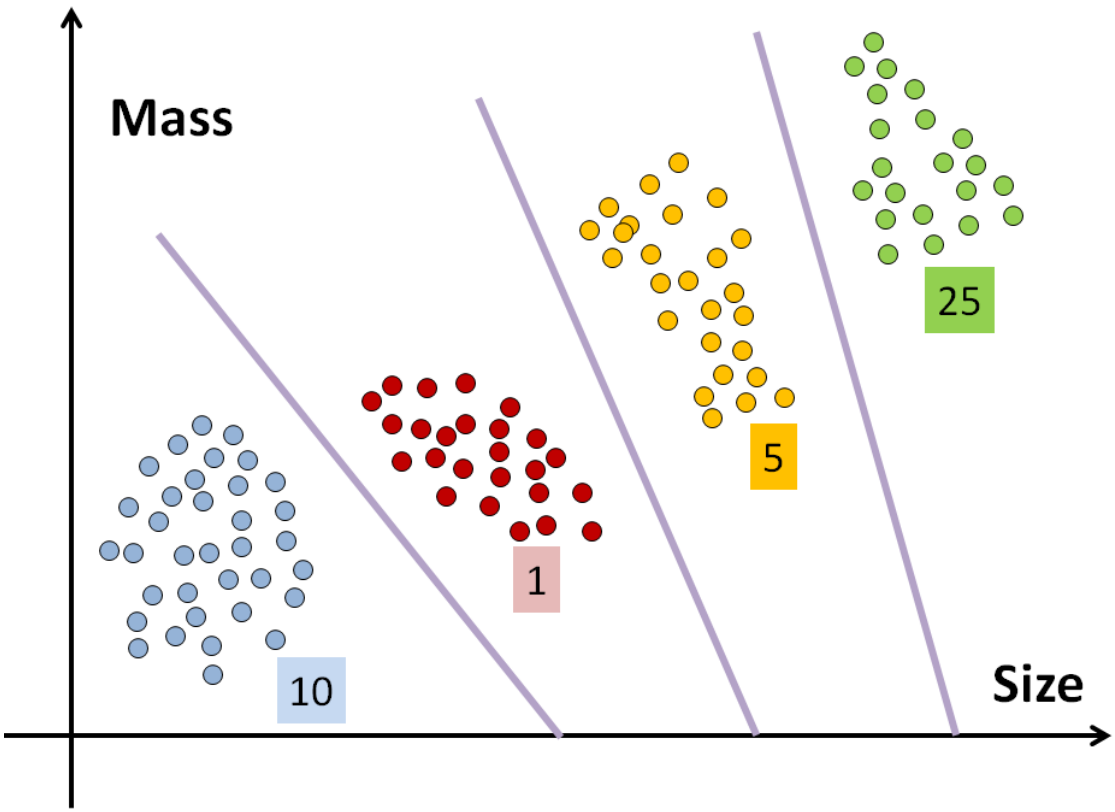
- You measure mass and size.

Approach 1: Learning



- You measure mass and size.
- Put each coin to its class. Plot a 2D histogram.

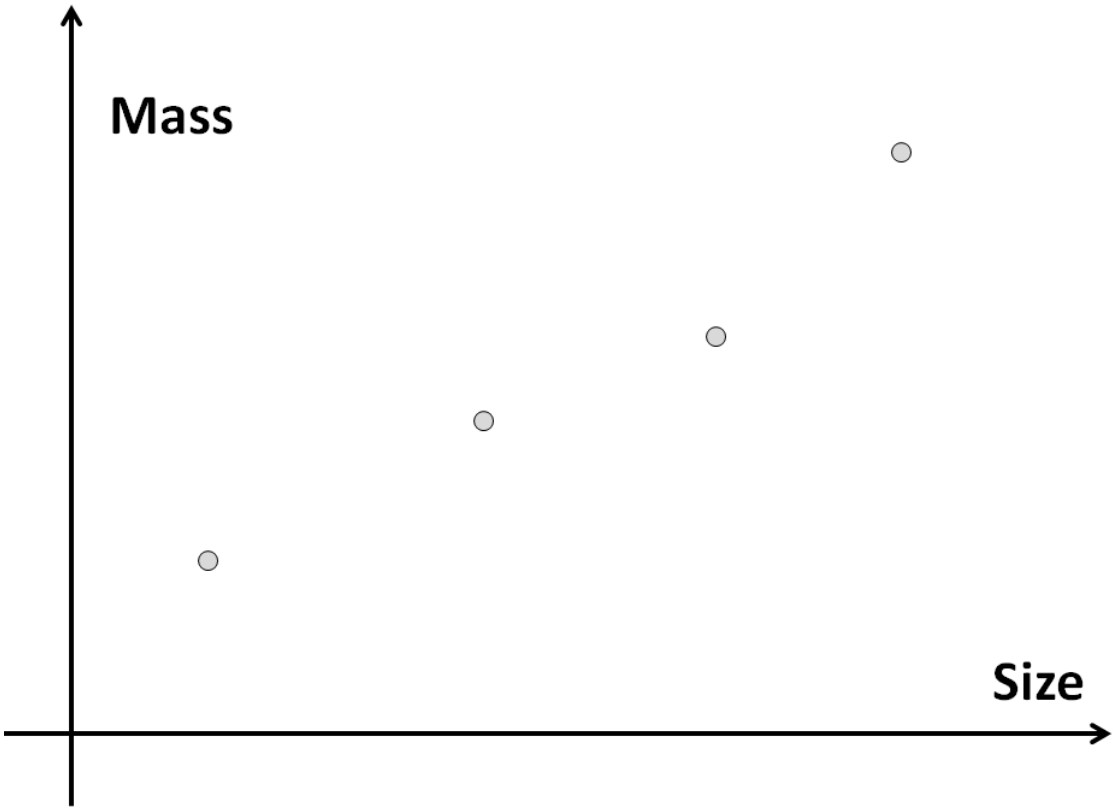
Approach 1: Learning



- You measure mass and size.
- Put each coin to its class. Plot a 2D histogram.
- Create the classifier.

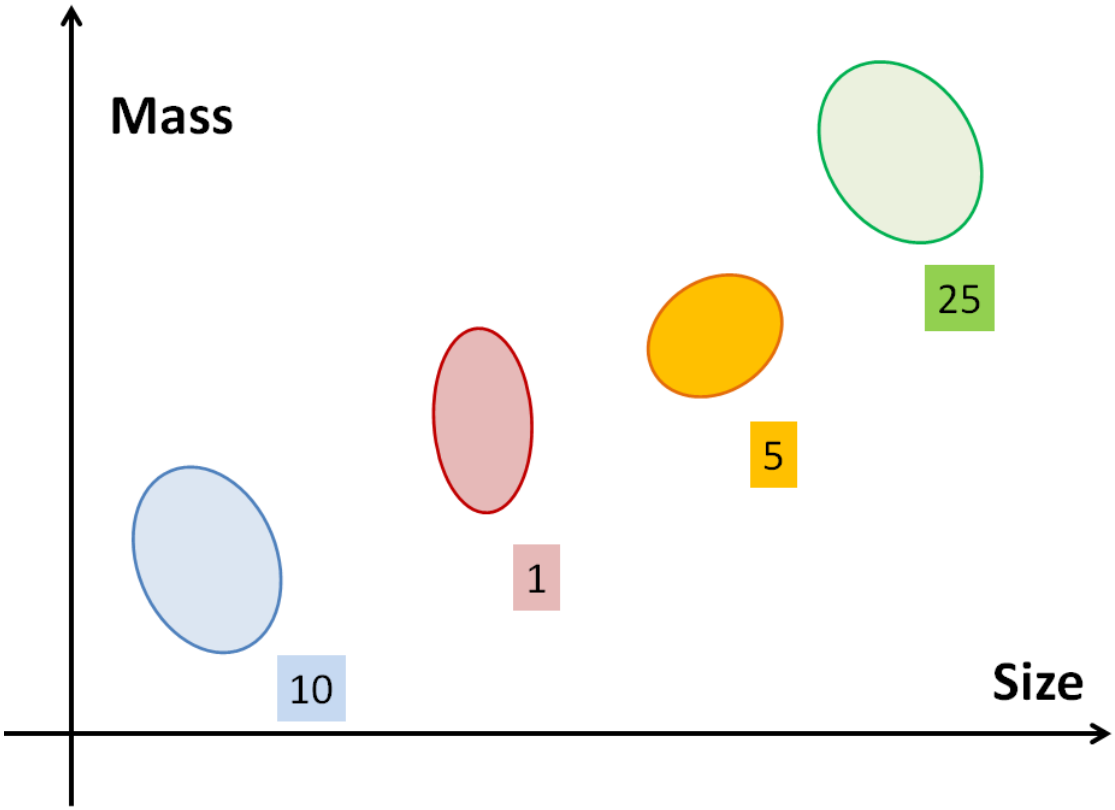
Approach 2: Design

Approach 2: Design



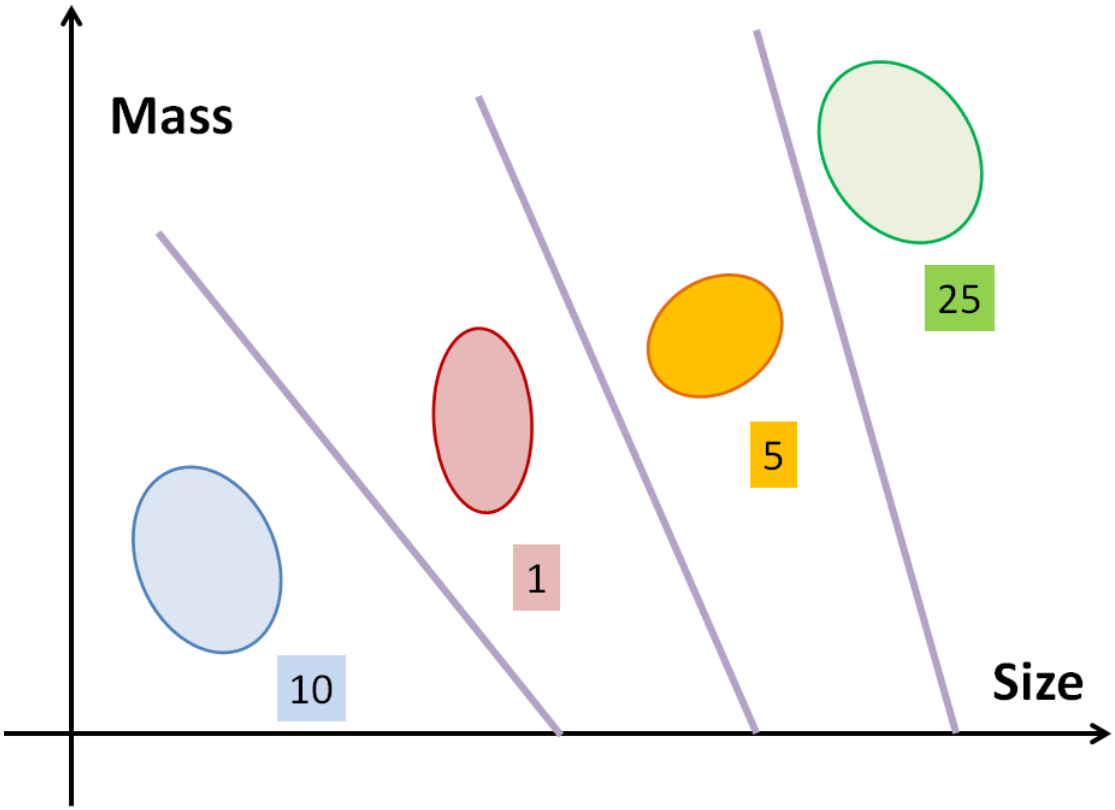
- You go to United States Mint to ask ideal mass and size of the coins.

Approach 2: Design



- You go to United States Mint to ask ideal mass and size of the coins.
- You ask them to give you the measurement error. Plot 2D distribution.

Approach 2: Design



- You go to United States Mint to ask ideal mass and size of the coins.
- You ask them to give you the measurement error. Plot 2D distribution.
- Create the classifier.

Which one fits learning? Which one fits design?

- Determining the age at which a particular medical test should be performed.

Which one fits learning? Which one fits design?

- Determining the age at which a particular medical test should be performed.
- Classifying numbers into primes and non-primes.

Which one fits learning? Which one fits design?

- Determining the age at which a particular medical test should be performed.
- Classifying numbers into primes and non-primes.
- Detecting potential fraud in credit card charges.

Which one fits learning? Which one fits design?

- Determining the age at which a particular medical test should be performed.
- Classifying numbers into primes and non-primes.
- Detecting potential fraud in credit card charges.
- Determining the time it would take a falling object to hit the ground.

Which one fits learning? Which one fits design?

- Determining the age at which a particular medical test should be performed.
- Classifying numbers into primes and non-primes.
- Detecting potential fraud in credit card charges.
- Determining the time it would take a falling object to hit the ground.
- Determining the optimal cycle for traffic lights in a busy intersection.

Machine Learning Model

Machine Learning Model

- Data points $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$.

Machine Learning Model

- Data points $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$.
- Labels $y_1, \ldots, y_N$.

Machine Learning Model

- Data points $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$.
- Labels $y_1, \ldots, y_N$.

- Where does a data point $\boldsymbol{x}_n$ come from?

Machine Learning Model

- Data points $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$.
- Labels $y_1, \ldots, y_N$.

- Where does a data point $\boldsymbol{x}_n$ come from?
- How is a label y_n defined?

Machine Learning Model

- Data points $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$.
- Labels $y_1, \ldots, y_N$.

- Where does a data point $\boldsymbol{x}_n$ come from?
- How is a label y_n defined?
- What do we mean by a learning algorithm?

Machine Learning Model

- Data points $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$.
- Labels $y_1, \ldots, y_N$.

- Where does a data point $\boldsymbol{x}_n$ come from?
- How is a label y_n defined?
- What do we mean by a learning algorithm?
- What is a classifier?

Machine Learning Model

- Data points $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$.
- Labels $y_1, \ldots, y_N$.

- Where does a data point $\boldsymbol{x}_n$ come from?
- How is a label y_n defined?
- What do we mean by a learning algorithm?
- What is a classifier?
- How to evaluate a classifier?

What is learning?

"The activity or process of gaining knowledge or skill by studying, practicing, being taught, or experiencing something."

Merriam Webster dictionary

"A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E."

Tom Mitchell

38

What is machine learning?

- For many problems, it's difficult to program the correct behavior by hand
  - recognizing people and objects
  - understanding human speech
- Machine learning approach: program an algorithm to automatically learn from data, or from experience
- Why might you want to use a learning algorithm?

What is machine learning?

- For many problems, it's difficult to program the correct behavior by hand
  - recognizing people and objects
  - understanding human speech
- Machine learning approach: program an algorithm to automatically learn from data, or from experience
- Why might you want to use a learning algorithm?
  - hard to code up a solution by hand (e.g. vision, speech)
  - system needs to adapt to a changing environment (e.g. spam detection)
  - want the system to perform *better* than the human programmers
  - privacy/fairness (e.g. ranking search results)

40

What is machine learning?

- It's similar to statistics...
  - Both fields try to uncover patterns in data
  - Both fields draw heavily on calculus, probability, and linear algebra, and share many of the same core algorithms
- But it's not statistics!
  - Stats is more concerned with helping scientists and policymakers draw good conclusions; ML is more concerned with building autonomous agents
  - Stats puts more emphasis on interpretability and mathematical rigor; ML puts more emphasis on predictive performance, scalability, and autonomy

Relations to AI

- Nowadays, "machine learning" is often brought up with "artificial intelligence" (AI)
- AI does not always imply a learning based system
  - Symbolic reasoning
  - Rule based system
  - Tree search
  - etc.
- Learning based system → learned based on the data → more flexibility, good at solving pattern recognition problems.

42

Relations to human learning

- Human learning is:
  - Very data efficient
  - An entire multitasking system (vision, language, motor control, etc.)
  - Takes at least a few years :)
- For serving specific purposes, machine learning doesn't have to look like human learning in the end.
- It may borrow ideas from biological systems, e.g., neural networks.
- It may perform better or worse than humans.

43

What is machine learning?

- Types of machine learning
  - **Supervised learning:** have labeled examples of the correct behavior
  - **Reinforcement learning:** learning system (agent) interacts with the world and learns to maximize a scalar reward signal
  - **Unsupervised learning:** no labeled examples – instead, looking for "interesting" patterns in the data

44

History of machine learning

- 1957 — Perceptron algorithm (implemented as a circuit!)
- 1959 — Arthur Samuel wrote a learning-based checkers program that could defeat him
- 1969 — Minsky and Papert's book *Perceptrons* (limitations of linear models)
- 1980s — Some foundational ideas
  - Connectionist psychologists explored neural models of cognition
  - 1984 — Leslie Valiant formalized the problem of learning as PAC learning
  - 1988 — Backpropagation (re-)discovered by Geoffrey Hinton and colleagues
  - 1988 — Judea Pearl's book *Probabilistic Reasoning in Intelligent Systems* introduced Bayesian networks

45

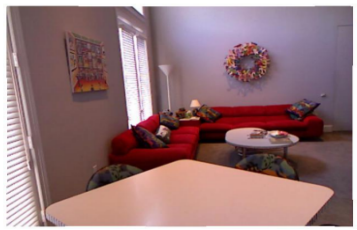
History of machine learning

- 1990s — the "AI Winter", a time of pessimism and low funding
- But looking back, the '90s were also sort of a golden age for ML research
  - Markov chain Monte Carlo
  - variational inference
  - kernels and support vector machines
  - boosting
  - convolutional networks
  - reinforcement learning
- 2000s — applied AI fields (vision, NLP, etc.) adopted ML
- 2010s — deep learning
  - 2010–2012 — neural nets smashed previous records in speech-to-text and object recognition
  - increasing adoption by the tech industry
  - 2016 — AlphaGo defeated the human Go champion
  - 2018-now — generating photorealistic images and videos
  - 2020 — GPT3 language model
- now — increasing attention to ethical and societal implications

Computer vision: Object detection, semantic segmentation, pose estimation, and almost every other task is done with ML.

Figure 4. More results of **Mask R-CNN** on COCO test images, using ResNet-101-FPN and running at 5 fps, with 35.7 mask AP (Table 1).

Instance segmentation - ▸ Link



DAQUAR 1553
What is there in front of the sofa?
Ground truth: table
IMG+BOW: table (0.74)
2-VIS+BLSTM: table (0.88)
LSTM: chair (0.47)

COCOQA 5078
How many leftover donuts is the red bicycle holding?
Ground truth: three
IMG+BOW: two (0.51)
2-VIS+BLSTM: three (0.27)
BOW: one (0.29)

Speech: Speech to text, personal assistants, speaker identification...

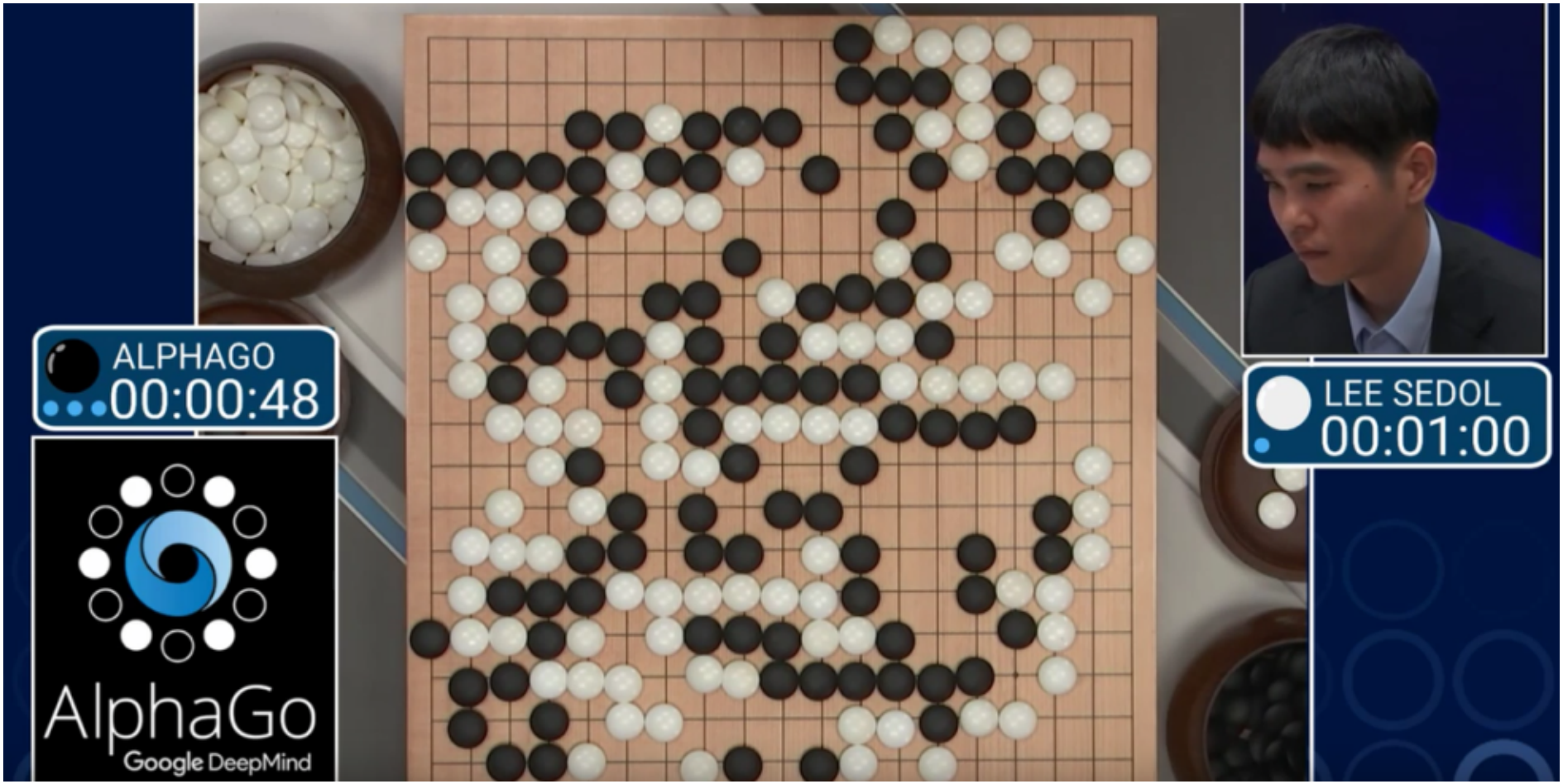
NLP: Machine translation, sentiment analysis, topic modeling, spam filtering.

Real world example: The New York Times

LDA analysis of 1.8M New York Times articles:

music band songs rock album jazz pop song singer night	book life novel story books man stories love children family	art museum show exhibition artist artists paintings painting century works	game knicks nets points team season play games night coach	show film television movie series says life man character know
theater play production show stage street broadway director musical directed	clinton bush campaign gore political republican dole presidential senator house	stock market percent fund investors funds companies stocks investment trading	restaurant sauce menu food dishes street dining dinner chicken served	budget tax governor county mayor billion taxes plan legislature fiscal

Playing Games

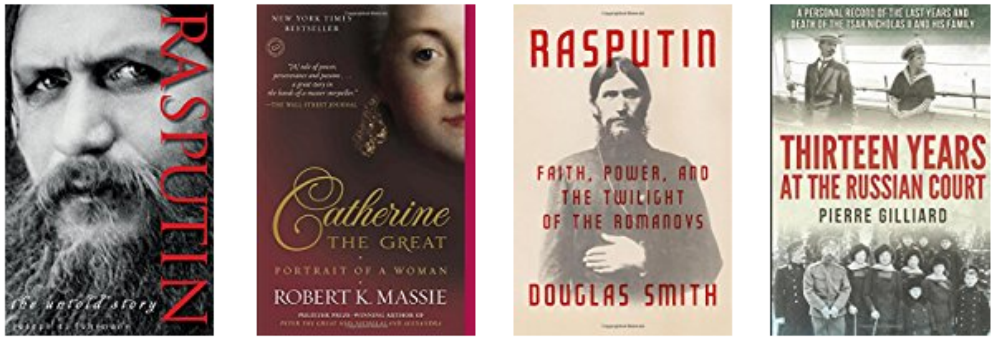


DOTA2 - ▸ Link

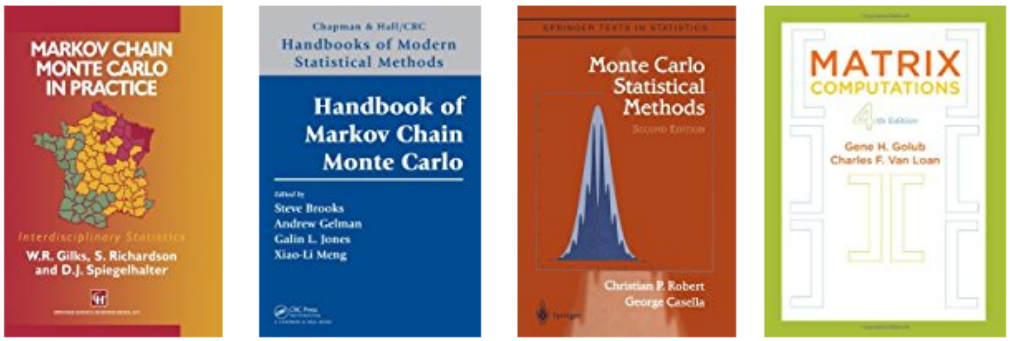
50

E-commerce & Recommender Systems : Amazon, netflix, ...

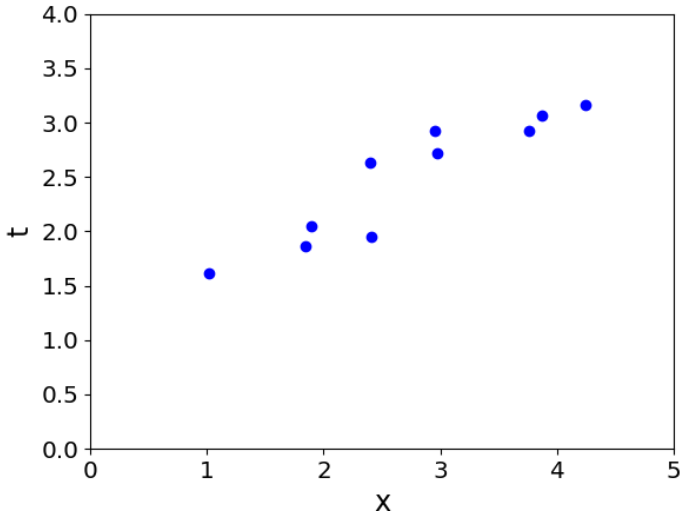
Inspired by your shopping trends



Related to items you've viewed See more



Supervised Learning Setup



In supervised learning:

- There is input $\mathbf{x} \in \mathcal{X}$, typically a vector of features (or covariates)
- There is target $t \in \mathcal{T}$ (also called response, outcome, output, class)
- Objective is to learn a function $f : \mathcal{X} \to \mathcal{T}$ such that $t \approx y = f(\mathbf{x})$ based on some data $\mathcal{D} = \{(\mathbf{x}^{(i)}, t^{(i)}) \text{ for } i = 1, 2, ..., N\}$.

52

Linear Regression - Model

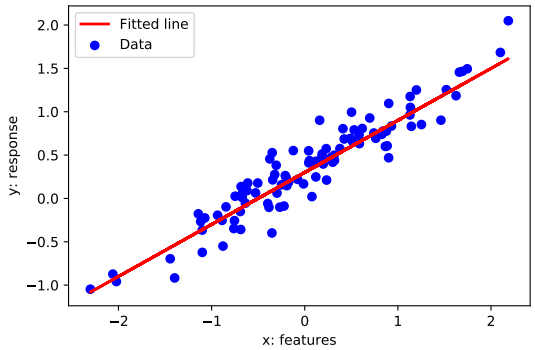
- Model: In linear regression, we use a *linear* function of the features $\mathbf{x} = (x_1, \ldots, x_D) \in \mathbb{R}^D$ to make predictions y of the target value $t \in \mathbb{R}$:

$$y = f(\mathbf{x}) = \sum_j w_j x_j + b$$

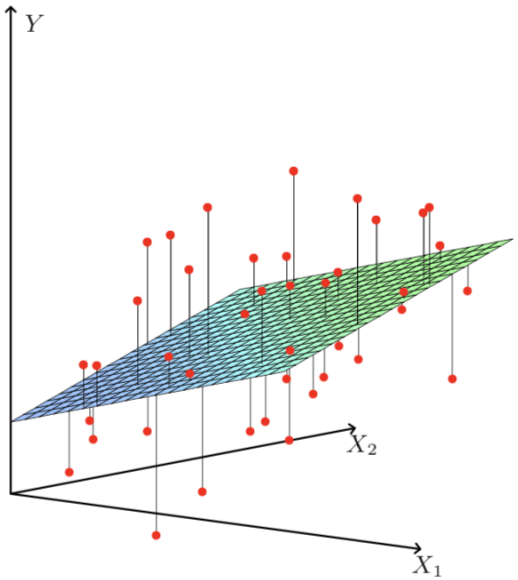
  - y is the prediction
  - $\mathbf{w}$ is the weights
  - b is the bias (or intercept)
- $\mathbf{w}$ and b together are the parameters
- We hope that our prediction is close to the target: $y \approx t$.

53

What is Linear? 1 feature vs D features



- If we have only 1 feature:
  $y = wx + b$ where $w, x, b \in \mathbb{R}$.
- y is linear in x.



- If we have D features:
  $y = \mathbf{w}^\top \mathbf{x} + b$ where $\mathbf{w}, \mathbf{x} \in \mathbb{R}^D$,
  $b \in \mathbb{R}$
- y is linear in $\mathbf{x}$.

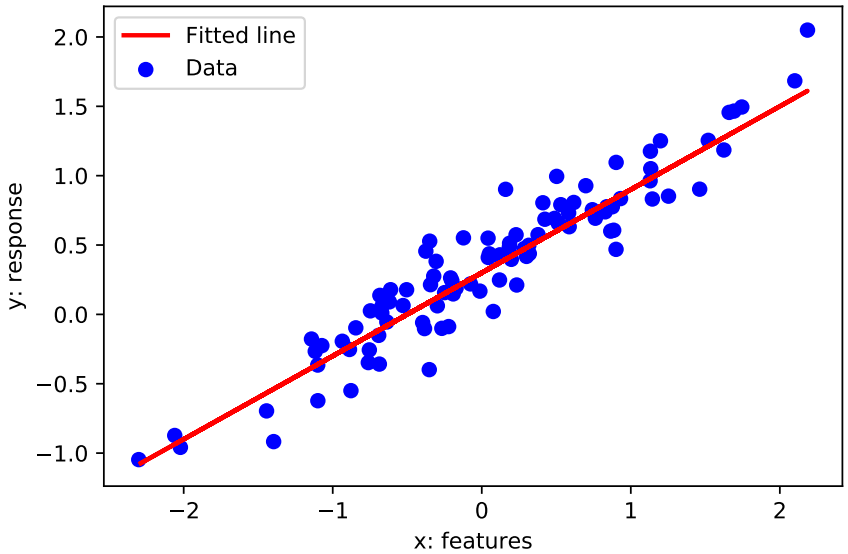
Relation between the prediction y and inputs $\mathbf{x}$ is linear in both cases.

54

Linear Regression

We have a dataset $\mathcal{D} = \{(\mathbf{x}^{(i)}, t^{(i)}) \text{ for } i = 1, 2, ..., N\}$ where,

- $\mathbf{x}^{(i)} = (x_1^{(i)}, x_2^{(i)}, ..., x_D^{(i)})^\top \in \mathbb{R}^D$ are the inputs (e.g. age, height)
- $t^{(i)} \in \mathbb{R}$ is the target or response (e.g. income)
- predict $t^{(i)}$ with a linear function of $\mathbf{x}^{(i)}$:



- $t^{(i)} \approx y^{(i)} = \mathbf{w}^\top \mathbf{x}^{(i)} + b$
- Different $(\mathbf{w}, b)$ define different lines.
- We want the "best" line $(\mathbf{w}, b)$.
- How to quantify "best"?

55

Linear Regression - Loss Function

- A loss function $\mathcal{L}(y, t)$ defines how bad it is if, for some example $\mathbf{x}$, the algorithm predicts y, but the target is actually t.
- Squared error loss function:

$$\mathcal{L}(y, t) = \frac{1}{2}(y - t)^2$$

- $y - t$ is the residual, and we want to make this small in magnitude
- The $\frac{1}{2}$ factor is just to make the calculations convenient.
- Cost function: loss function averaged over all training examples

$$\begin{aligned}\mathcal{J}(\mathbf{w}, b) &= \frac{1}{2N}\sum_{i=1}^{N}\left(y^{(i)} - t^{(i)}\right)^2 \\ &= \frac{1}{2N}\sum_{i=1}^{N}\left(\mathbf{w}^\top \mathbf{x}^{(i)} + b - t^{(i)}\right)^2\end{aligned}$$

- Terminology varies. Some call "cost" *empirical* or *average loss*.

Vectorization

- Notation-wise, $\frac{1}{2N}\sum_{i=1}^{N}\left(y^{(i)}-t^{(i)}\right)^2$ gets messy if we expand $y^{(i)}$:

$$\frac{1}{2N}\sum_{i=1}^{N}\left(\sum_{j=1}^{D}\left(w_j x_j^{(i)}+b\right)-t^{(i)}\right)^2$$

- The code equivalent is to compute the prediction using a for loop:

```
y = b
for j in range(M):
    y += w[j] * x[j]
```

- Excessive super/sub scripts are hard to work with, and Python loops are slow, so we vectorize algorithms by expressing them in terms of vectors and matrices.

$$\mathbf{w}=(w_1,\ldots,w_D)^\top \qquad \mathbf{x}=(x_1,\ldots,x_D)^\top$$

$$y=\mathbf{w}^\top\mathbf{x}+b$$

- This is simpler and executes much faster:

```
y = np.dot(w, x) + b
```

57

Vectorization

Why vectorize?

- The equations, and the code, will be simpler and more readable. Gets rid of dummy variables/indices!
- Vectorized code is much faster
  - Cut down on Python interpreter overhead
  - Use highly optimized linear algebra libraries (hardware support)
  - Matrix multiplication very fast on GPU (Graphics Processing Unit)

Switching in and out of vectorized form is a skill you gain with practice

- Some derivations are easier to do element-wise
- Some algorithms are easier to write/understand using for-loops and vectorize later for performance

58

Vectorization

- We can organize all the training examples into a design matrix $\mathbf{X}$ with one row per training example, and all the targets into the target vector $\mathbf{t}$.

one feature across all training examples

$$\mathbf{X} = \begin{pmatrix} \mathbf{x}^{(1)\top} \\ \mathbf{x}^{(2)\top} \\ \mathbf{x}^{(3)\top} \end{pmatrix} = \begin{pmatrix} 8 & 0 & 3 & 0 \\ 6 & -1 & 5 & 3 \\ 2 & 5 & -2 & 8 \end{pmatrix}$$

one training example (vector)

- Computing the predictions for the whole dataset:

$$\mathbf{X}\mathbf{w} + b\mathbf{1} = \begin{pmatrix} \mathbf{w}^T\mathbf{x}^{(1)} + b \\ \vdots \\ \mathbf{w}^T\mathbf{x}^{(N)} + b \end{pmatrix} = \begin{pmatrix} y^{(1)} \\ \vdots \\ y^{(N)} \end{pmatrix} = \mathbf{y}$$

59

Vectorization

- Computing the squared error cost across the whole dataset:

$$\mathbf{y} = \mathbf{X}\mathbf{w} + b\mathbf{1}$$

$$\mathcal{J} = \frac{1}{2N}\|\mathbf{y} - \mathbf{t}\|^2$$

- Sometimes we may use $\mathcal{J} = \frac{1}{2}\|\mathbf{y} - \mathbf{t}\|^2$, without a normalizer. This would correspond to the sum of losses, and not the averaged loss. The minimizer does not depend on N (but optimization might!).
- We can also add a column of 1's to design matrix, combine the bias and the weights, and conveniently write

$$\mathbf{X} = \begin{bmatrix} 1 & [\mathbf{x}^{(1)}]^\top \\ 1 & [\mathbf{x}^{(2)}]^\top \\ 1 & \vdots \end{bmatrix} \in \mathbb{R}^{N \times (D+1)} \quad \text{and} \quad \mathbf{w} = \begin{bmatrix} b \\ w_1 \\ w_2 \\ \vdots \end{bmatrix} \in \mathbb{R}^{D+1}$$

Then, our predictions reduce to $\mathbf{y} = \mathbf{X}\mathbf{w}$.

60

Solving the Minimization Problem

We defined a cost function. This is what we'd like to minimize.

Two commonly applied mathematical approaches:

- Algebraic, e.g., using inequalities:
  - to show z^* minimizes $f(z)$, show that $\forall z, f(z) \geq f(z^*)$
  - to show that $a = b$, show that $a \geq b$ and $b \geq a$
- Calculus: minimum of a smooth function (if it exists) occurs at a critical point, i.e. point where the derivative is zero.
  - multivariate generalization: set the partial derivatives to zero (or equivalently the gradient).

Solutions may be direct or iterative

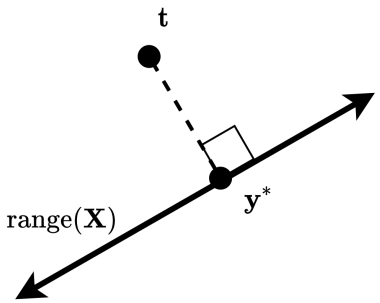
- Sometimes we can directly find provably optimal parameters (e.g. set the gradient to zero and solve in closed form). We call this a direct solution.
- We may also use optimization techniques that iteratively get us closer to the solution. We will get back to this soon.

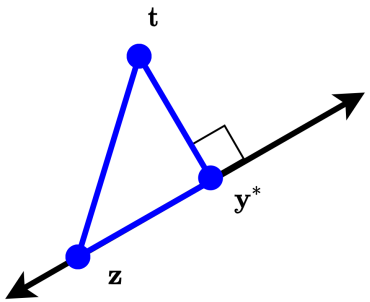
61

Direct Solution I: Linear Algebra

- We seek $\mathbf{w}$ to minimize $\|\mathbf{X}\mathbf{w} - \mathbf{t}\|^2$, or equivalently $\|\mathbf{X}\mathbf{w} - \mathbf{t}\|$

  range$(\mathbf{X}) = \{\mathbf{X}\mathbf{w} \,|\, \mathbf{w} \in \mathbb{R}^D\}$ is a D-dimensional subspace of $\mathbb{R}^N$.
- Recall that the closest point $\mathbf{y}^* = \mathbf{X}\mathbf{w}^*$ in subspace range$(\mathbf{X})$ of $\mathbb{R}^N$ to arbitrary point $\mathbf{t} \in \mathbb{R}^N$ is found by orthogonal projection.





- We have $(\mathbf{y}^* - \mathbf{t}) \perp \mathbf{X}\mathbf{w}, \ \forall \mathbf{w} \in \mathbb{R}^D$
- Why is $\mathbf{y}^*$ the closest point to $\mathbf{t}$?
  - Consider any $\mathbf{z} = \mathbf{X}\mathbf{w}$
  - By Pythagorean theorem and the trivial inequality $(x^2 \geq 0)$:

$$\begin{aligned}\|\mathbf{z} - \mathbf{t}\|^2 &= \|\mathbf{y}^* - \mathbf{t}\|^2 + \|\mathbf{y}^* - \mathbf{z}\|^2 \\ &\geq \|\mathbf{y}^* - \mathbf{t}\|^2\end{aligned}$$

62

Direct Solution I: Linear Algebra

- From the previous slide, we have $(\mathbf{y}^* - \mathbf{t}) \perp \mathbf{X}\mathbf{w},\ \forall \mathbf{w} \in \mathbb{R}^D$
- Equivalently, the columns of the design matrix $\mathbf{X}$ are all orthogonal to $(\mathbf{y}^* - \mathbf{t})$, and we have that:

$$\begin{aligned} \mathbf{X}^\top(\mathbf{y}^* - \mathbf{t}) &= \mathbf{0} \\ \mathbf{X}^\top\mathbf{X}\mathbf{w}^* - \mathbf{X}^\top\mathbf{t} &= \mathbf{0} \\ \mathbf{X}^\top\mathbf{X}\mathbf{w}^* &= \mathbf{X}^\top\mathbf{t} \\ \mathbf{w}^* &= (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{t} \end{aligned}$$

- While this solution is clean and the derivation easy to remember, like many algebraic solutions, it is somewhat ad hoc.
- On the hand, the tools of calculus are broadly applicable to differentiable loss functions...

63

Direct Solution II: Calculus

- Partial derivative: derivative of a multivariate function with respect to one of its arguments.

$$\frac{\partial}{\partial x_1} f(x_1, x_2) = \lim_{h \to 0} \frac{f(x_1 + h, x_2) - f(x_1, x_2)}{h}$$

- To compute, take the single variable derivative, pretending the other arguments are constant.
- Example: partial derivatives of the prediction y

$$\frac{\partial y}{\partial w_j} = \frac{\partial}{\partial w_j} \left[\sum_{j'} w_{j'} x_{j'} + b \right] = x_j \qquad \frac{\partial y}{\partial b} = \frac{\partial}{\partial b} \left[\sum_{j'} w_{j'} x_{j'} + b \right] = 1$$

64

Direct Solution II: Calculus

- For loss derivatives, apply the chain rule:

$$\frac{\partial \mathcal{L}}{\partial w_j} = \frac{\mathrm{d}\mathcal{L}}{\mathrm{d}y}\frac{\partial y}{\partial w_j} = \frac{\mathrm{d}}{\mathrm{d}y}\left[\frac{1}{2}(y-t)^2\right] \cdot x_j = (y-t)x_j$$

$$\frac{\partial \mathcal{L}}{\partial b} = \frac{\mathrm{d}\mathcal{L}}{\mathrm{d}y}\frac{\partial y}{\partial b} = y - t$$

- For cost derivatives, use linearity and average over data points:

$$\frac{\partial \mathcal{J}}{\partial w_j} = \frac{1}{N}\sum_{i=1}^{N}(y^{(i)} - t^{(i)})\, x_j^{(i)} \qquad \frac{\partial \mathcal{J}}{\partial b} = \frac{1}{N}\sum_{i=1}^{N} y^{(i)} - t^{(i)}$$

- Minimum must occur at a point where partial derivatives are zero.

$$\frac{\partial \mathcal{J}}{\partial w_j} = 0 \quad (\forall j), \qquad \frac{\partial \mathcal{J}}{\partial b} = 0.$$

(if $\partial \mathcal{J}/\partial w_j \neq 0$, you could reduce the cost by changing w_j)

65

Direct Solution II: Calculus

- The derivation on the previous slide gives a system of linear equations, which we can solve efficiently.
- As is often the case for models and code, however, the solution is easier to characterize if we vectorize our calculus.
- We call the vector of partial derivatives the gradient
- Thus, the "gradient of $f : \mathbb{R}^D \to \mathbb{R}$", denoted $\nabla f(\mathbf{w})$, is:

$$\left(\frac{\partial}{\partial w_1} f(\mathbf{w}), \ldots, \frac{\partial}{\partial w_D} f(\mathbf{w})\right)^\top$$

- The gradient points in the direction of the greatest rate of increase.
- Analogue of second derivative (the "Hessian" matrix): $\nabla^2 f(\mathbf{w}) \in \mathbb{R}^{D \times D}$ is a matrix with $[\nabla^2 f(\mathbf{w})]_{ij} = \frac{\partial^2}{\partial w_i \partial w_j} f(\mathbf{w})$.

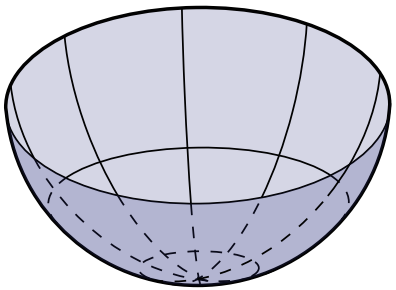
66

Aside: The Hessian Matrix

- Analogue of second derivative (the Hessian): $\nabla^2 f(\mathbf{w}) \in \mathbb{R}^{D \times D}$ is a matrix with $[\nabla^2 f(\mathbf{w})]_{ij} = \frac{\partial^2}{\partial w_i \partial w_j} f(\mathbf{w})$.
  - Recall from multivariable calculus that for continuously differentiable f, $\frac{\partial^2}{\partial w_i \partial w_j} f = \frac{\partial^2}{\partial w_j \partial w_i} f$, so the Hessian is symmetric.
- The second derivative test in single variable calculus: a critical point is a local minimum if the second derivative is positive.
- The multivariate analogue involves the eigenvalues of the Hessian.
  - Recall from linear algebra that the eigenvalues of a symmetric matrix (and therefore the Hessian) are real-valued.
  - If all of the eigenvalues are positive, we say the Hessian is positive definite.
  - A critical point ($\nabla f(\mathbf{w}) = \mathbf{0}$) of a continuously differentiable function f is a local minimum if the Hessian is positive definite.

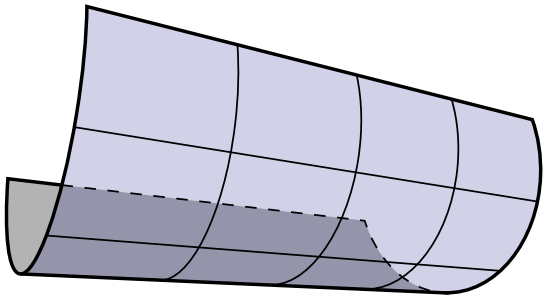
Aside: The Hessian Matrix

- Visualization:[1]



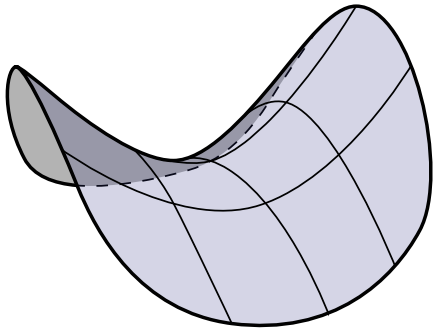
$x^2 + y^2$

(definite)



x^2

(semidefinite)



$x^2 - y^2$

(indefinite)

[1] Image source: mkwiki.org

Direct Solution II: Calculus

- We seek $\mathbf{w}$ to minimize $\mathcal{J}(\mathbf{w}) = \frac{1}{2}\|\mathbf{X}\mathbf{w} - \mathbf{t}\|^2$
- Taking the gradient with respect to $\mathbf{w}$ (**see course notes for additional details**) we get:

$$\nabla_{\mathbf{w}}\mathcal{J}(\mathbf{w}) = \mathbf{X}^\top\mathbf{X}\mathbf{w} - \mathbf{X}^\top\mathbf{t} = \mathbf{0}$$

- We get the same optimal weights as before:

$$\mathbf{w}^* = (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{t}$$

- Linear regression is one of only a handful of models in this course that permit direct solution.

69