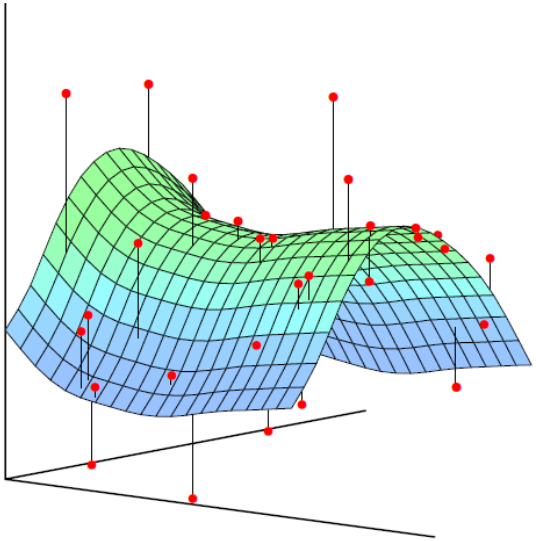
Machine Learning Overview



(James, Witten, Hastie, & Tibshirani, 2013)

ECE 411 Introduction to Machine Learning

Chau-Wai Wong, NC State University, Fall 2024

Acknowledgment: Some graphics and slides were adapted from the resources provided by the publisher.

Machine Learning (ML) in the News

How IBM built Watson, its *Jeopardy*-playing supercomputer by Dawn Kawamoto DailyFinance 02/08/2011

Learning from its mistakes According to David Ferrucci (PI of Watson DeepQA technology for IBM Research), Watson's software is wired for more that handling natural language processing.

"It's machine learning allows the computer to become smarter as it tries to answer questions — and to learn as it gets them right or wrong."

Modern ML Capabilities: Face Generation

- ◆ StyleGAN (2021): A *generative adversarial network* (*GAN*) capable of generate photorealistic images by progressively adding details to lower resolution intermediate images.

https://nvlabs.github.io/stylegan3/

Real images from the training set

StyleGAN3-T (ours), FID 3.67

- ◆ StyleGAN2 generated faces: https://thispersondoesnotexist.com/
- ◆ Can you tell which face is real? https://www.whichfaceisreal.com/

Modern ML Capabilities: Text to Image

- DALL·E (2021): 12-billion parameter version of *generative pretrained transformer (GPT 3)* trained to generate images from text descriptions.

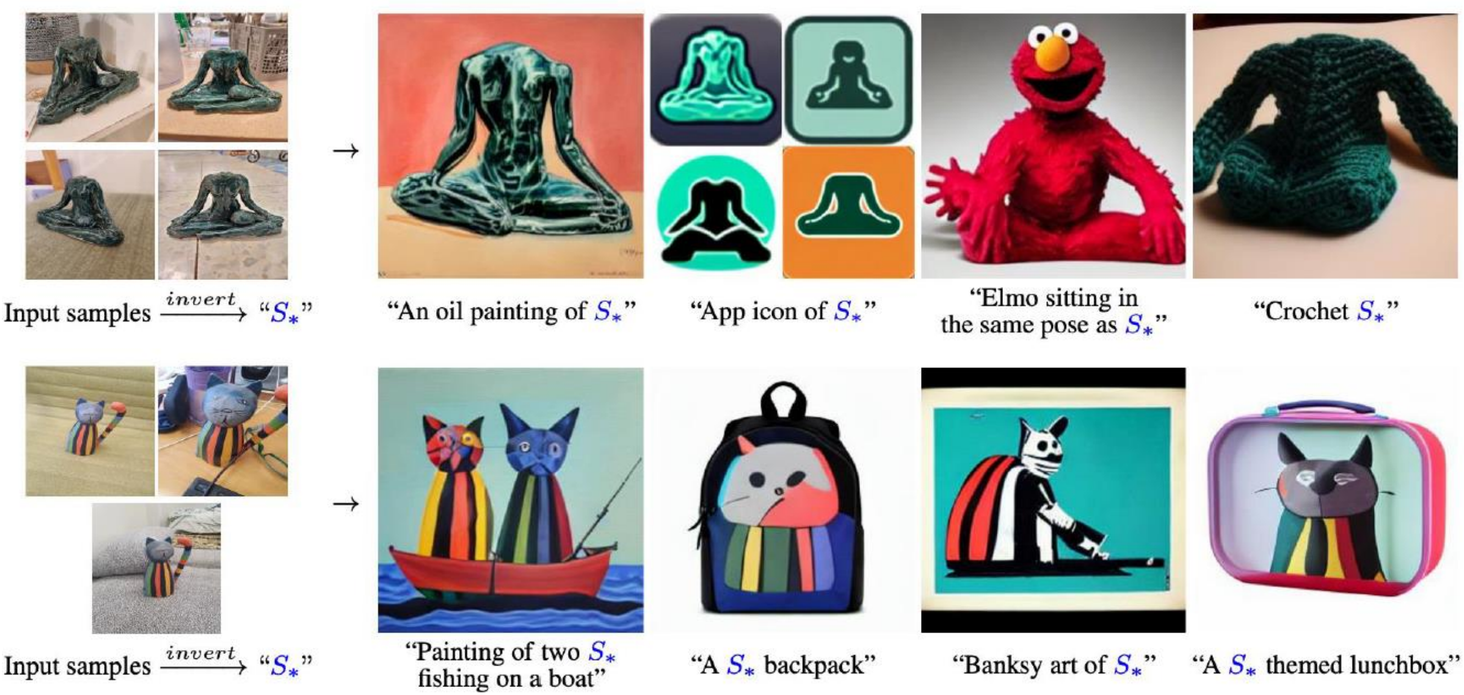
TEXT PROMPT an illustration of a baby daikon radish in a tutu walking a dog

AI-GENERATED IMAGES

More examples (including paper and code):
https://openai.com/blog/dall-e/

Modern ML Capabilities: Image Editing

- Diffusion models (2023): Learned to generate images by guided denoising.



Gal, R., Alaluf, Y., Atzmon, Y., Patashnik, O., Bermano, A. H., Chechik, G., and Cohen-Or, D., "An image is worth one word: Personalizing text-to-image generation using textual inversion," ICLR, 2023.

Modern ML Capabilities: Image to Video

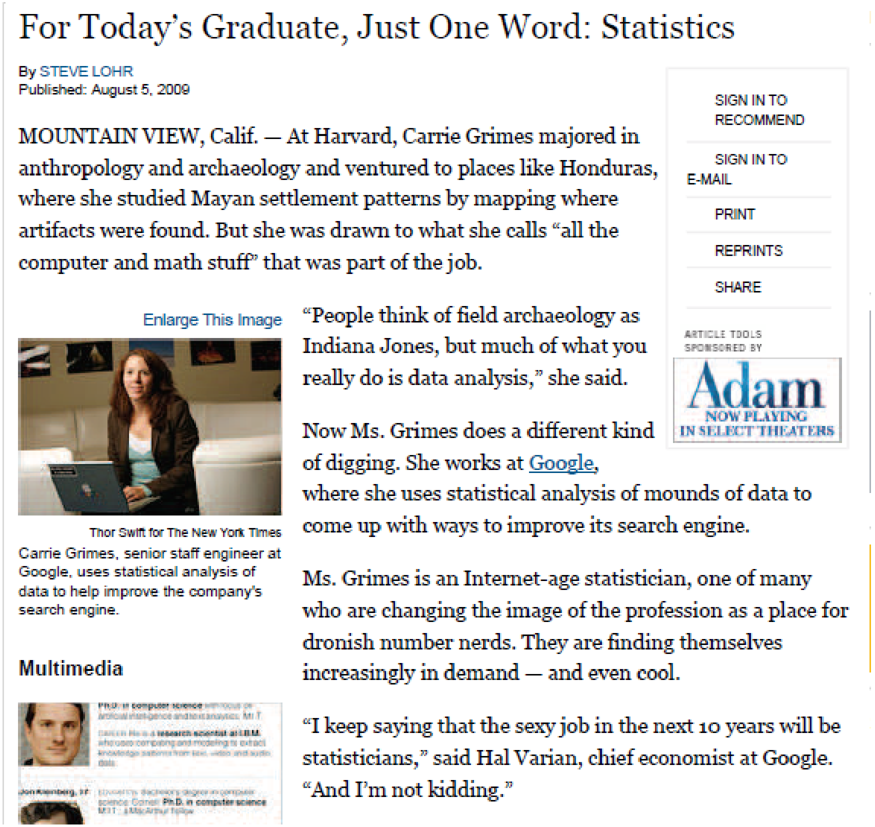
- OpenAI's Sora (2024): Diffusion model + transformer + latent space + tons of video training data → "World simulator"
- Can simulate the physical world in motion:
  - Complex scenes
  - Multiple characters
  - Specific motion types
  - Accurate subject details
- Not good at accurate physics, causality
- Can also do image2video, video extension
- Highly relevant to visual artists, designers, filmmakers

Visual Demo

OpenAI's Sora public release page and technical report.

Data Scientist is a Sexy Job

DSs deal with
unstructured data



For Today's Graduate, Just One Word: Statistics

By STEVE LOHR
Published: August 5, 2009

MOUNTAIN VIEW, Calif. — At Harvard, Carrie Grimes majored in anthropology and archaeology and ventured to places like Honduras, where she studied Mayan settlement patterns by mapping where artifacts were found. But she was drawn to what she calls "all the computer and math stuff" that was part of the job.

"People think of field archaeology as Indiana Jones, but much of what you really do is data analysis," she said.

Now Ms. Grimes does a different kind of digging. She works at Google, where she uses statistical analysis of mounds of data to come up with ways to improve its search engine.

Ms. Grimes is an Internet-age statistician, one of many who are changing the image of the profession as a place for dronish number nerds. They are finding themselves increasingly in demand — and even cool.

"I keep saying that the sexy job in the next 10 years will be statisticians," said Hal Varian, chief economist at Google. "And I'm not kidding."

Enlarge This Image

Thor Swift for The New York Times

Carrie Grimes, senior staff engineer at Google, uses statistical analysis of data to help improve the company's search engine.

Multimedia

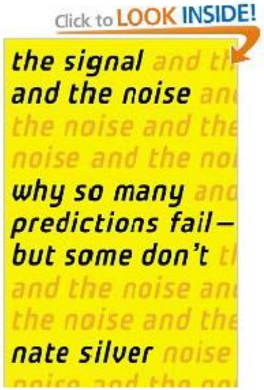
Quote of the Day, New York Times, August 5, 2009

"I keep saying that the sexy job in the next 10 years will be statisticians. And I'm not kidding."

— HAL VARIAN, chief economist at Google.

Machine Learning is a Part of Our Life

But it could generate wrong predictions :p

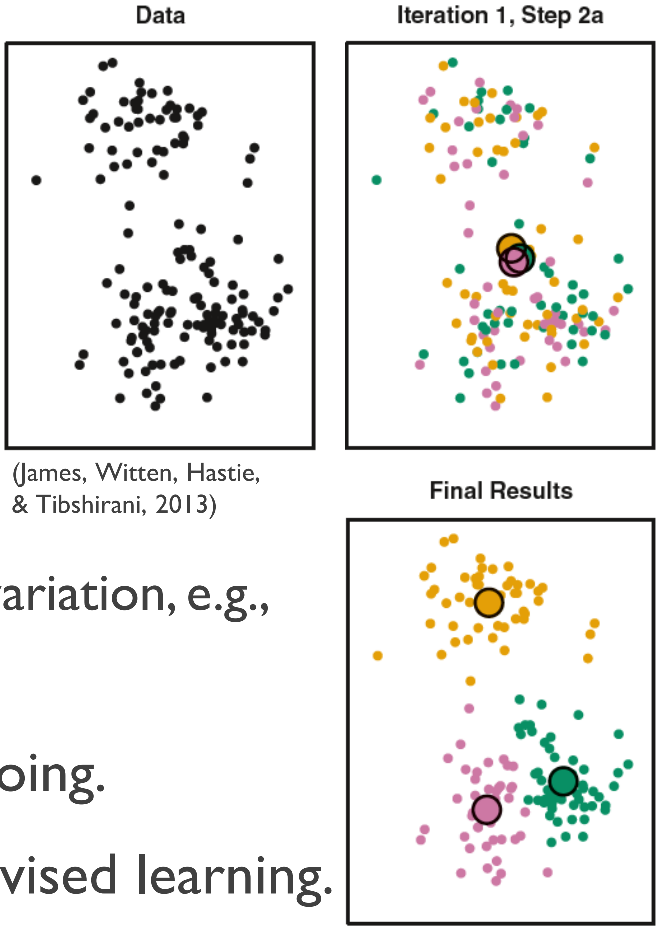


Machine Learning Philosophy

- It is important to understand the ideas behind the various techniques, in order to know how and when to use them.
- One has to understand the simpler/fundamental methods (2/3 of this course) first, e.g., *linear / logistic regression*, *principal component analysis (PCA)*, in order to grasp the more sophisticated ones.
- It is important to accurately assess the performance of a method, to know how well or how badly it is working. Simpler methods often perform as well as fancier ones!
- This is an exciting research area, having important applications in engineering, natural/social sciences, industry, finance, …
- Statistical machine learning is a fundamental pillar (interview questions!) in the training of a data scientist/machine learning engineer.

Machine Learning Paradigms: Unsupervised Learning

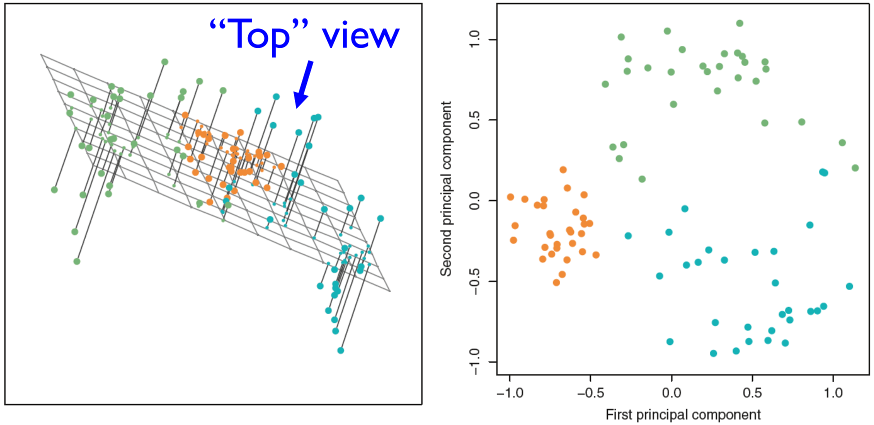
- ***Unsupervised Learning***: Learns from a set of unlabeled data to discover patterns (mathematical representation), without human supervision.
- Objective is fuzzy. For example, to find
  - Groups of samples that behave similarly, e.g., *k-nearest neighbors (kNN)*.
  - Linear combinations of features with the most variation, e.g., *principal component analysis (PCA)*.
- Difficult to judge how well the algorithm is doing.
- Can be useful as a preprocess. step for supervised learning.



(James, Witten, Hastie, & Tibshirani, 2013)

Machine Learning Paradigms: Unsupervised Learning

- Examples:
  - Movies grouped by ratings and behavioral data from viewers.
  - Groups of shoppers characterized by browsing & purchasing histories.
  - Subgroups of breast cancer patients grouped by gene expressions.
  - Tweets grouped by latent topics inferred from the use of words.
- Principal component analysis (PCA) can also be used for visualization:



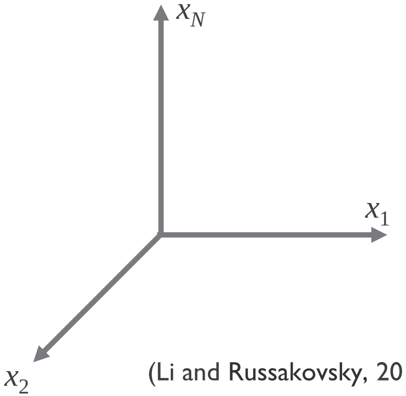
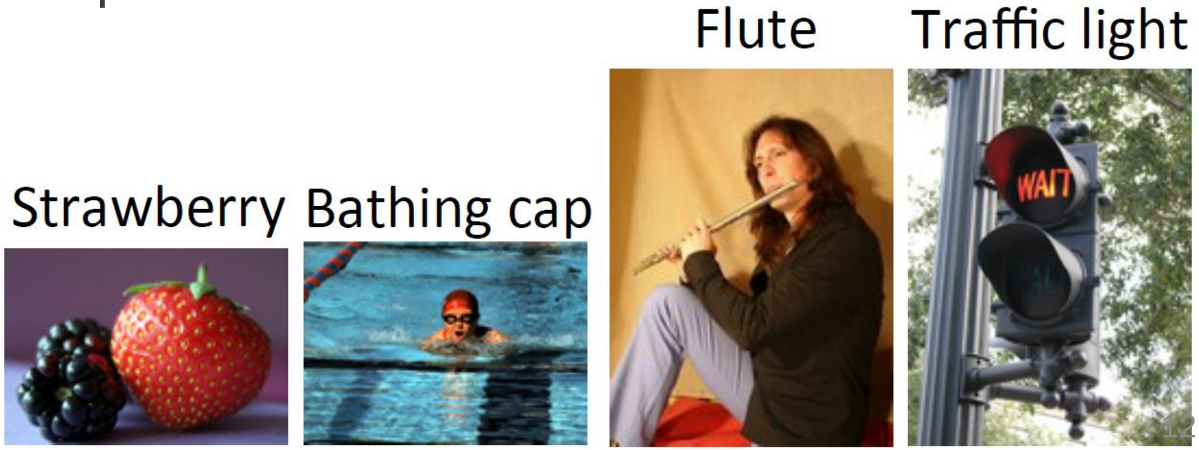
Each data point is a 3-D vector

Dim. reduction from 3-D to 2-D

Nonlinear dim. reduction tools: t-SNE, UMAP

Machine Learning Paradigms: Supervised Learning

- ***Supervised learning***: Learns an input–output mapping based on labeled data.
- Terminology:
  - Y: output / label, (outcome) measurement, response, target, dependent variable.
  - $\mathbf{X} = [X_1, \ldots, X_p]$: A vector of p inputs, features, predictor (measurements), regressors, covariates, independent variables.



(Li and Russakovsky, 2013)

Machine Learning Paradigms: Supervised Learning

- Major problems of supervised learning, ***regression*** vs. ***classification***:
  - In regression, *Y* is *quantitative*, e.g., price, blood pressure.
  - In classification, *Y* is *qualitative* / *categorical*, or a finite, unordered set, e.g., survived/died, cancer class of tissue sample).
    - A qualitative label is a member of a finite, unordered set.
    - Note: categorical ≠ ordinal. But one can consider ordinal numbers as categorical by ignoring relative relations.

Flute

Traffic light

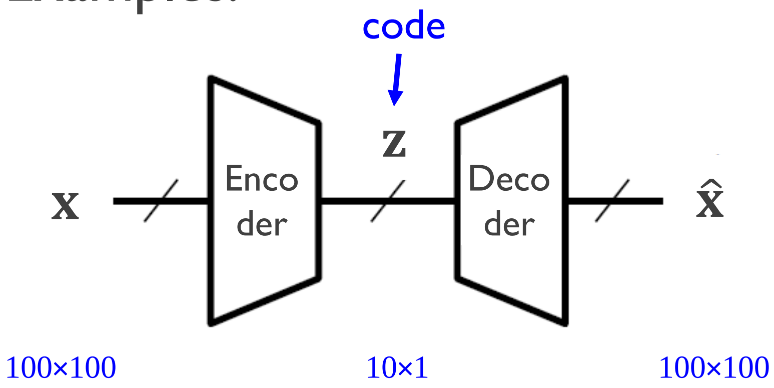
Strawberry

Bathing cap

(Li and Russakovsky, 2013)

Machine Learning Paradigms: Self-Supervised Learning

- ***Self-supervised learning***:[*] A representation learning method where a supervised task is created out of the unlabeled data.
- Used to reduce the data labelling cost and leverage the unlabeled data.
- Examples:



(i) Autoencoder

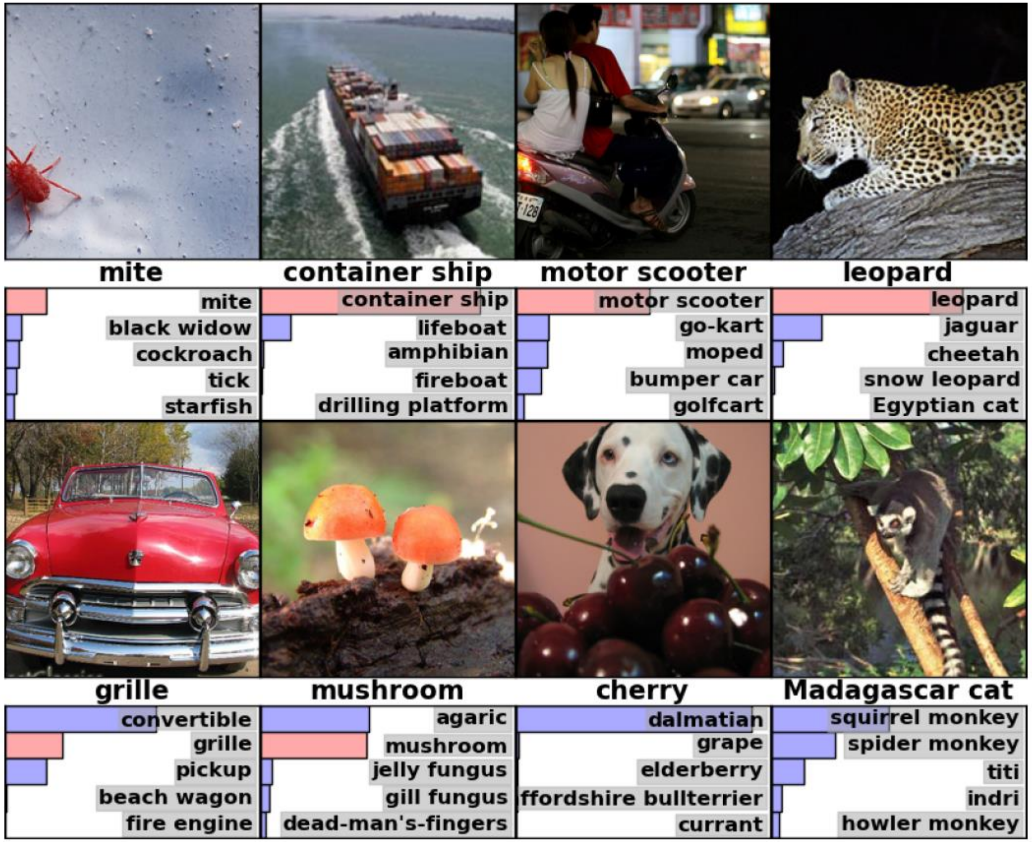
(predict, word) → miss
(miss, from) → word
(word, previous) → from

(ii) predicting missing word from the previous and next words.

---

[*] https://towardsdatascience.com/self-supervised-learning-methods-for-computer-vision-c25ec10a91bd

Supervised Learning: Classification



Goal of classification:

Assign a categorical/ qualitative label, or a class, to a given input.

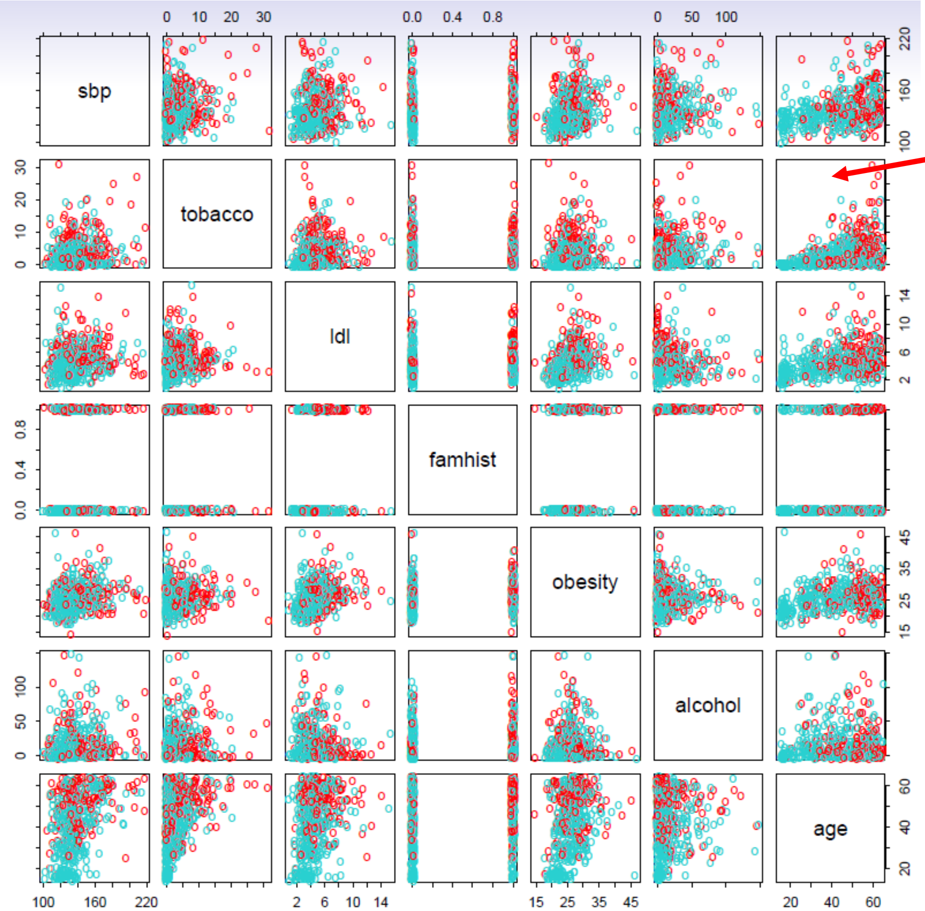
← Given an image, it returns the class label.

Optionally, provide a "confidence score."

Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton, **ImageNet** Classification with Deep Convolutional Neural Networks, *NeurIPS*, 2012.

Example: Predict whether someone will have a heart attack on the basis of demographic, diet and clinical measurements.

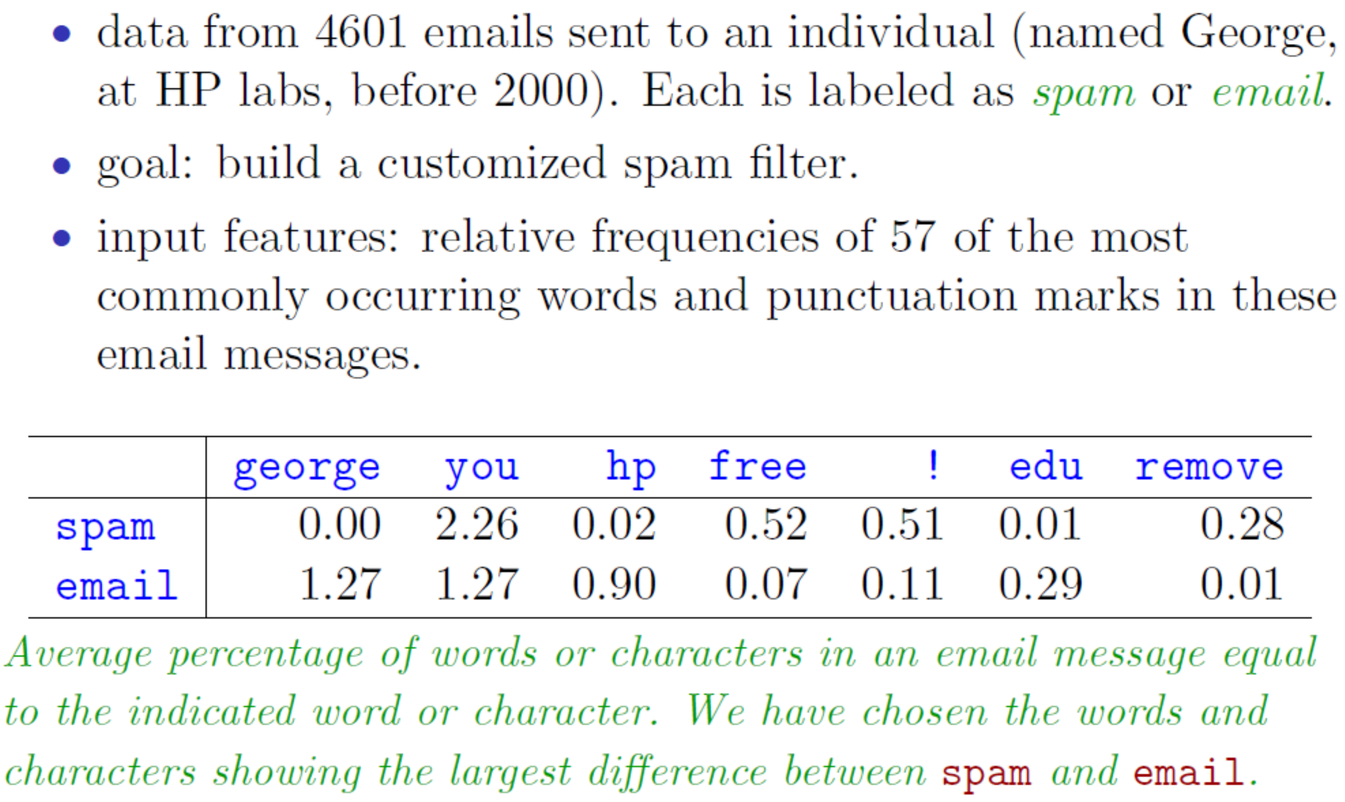
$y \in \{0, 1\}$



A scatter plot

$$\mathbf{x} = \begin{bmatrix} \text{sbp} \\ \text{tobacco} \\ \vdots \\ \text{age} \end{bmatrix}$$

Example: Spam detection (using naïve Bayes classifier)



- data from 4601 emails sent to an individual (named George, at HP labs, before 2000). Each is labeled as *spam* or *email*.
- goal: build a customized spam filter.
- input features: relative frequencies of 57 of the most commonly occurring words and punctuation marks in these email messages.

	`george`	`you`	`hp`	`free`	`!`	`edu`	`remove`
`spam`	0.00	2.26	0.02	0.52	0.51	0.01	0.28
`email`	1.27	1.27	0.90	0.07	0.11	0.29	0.01

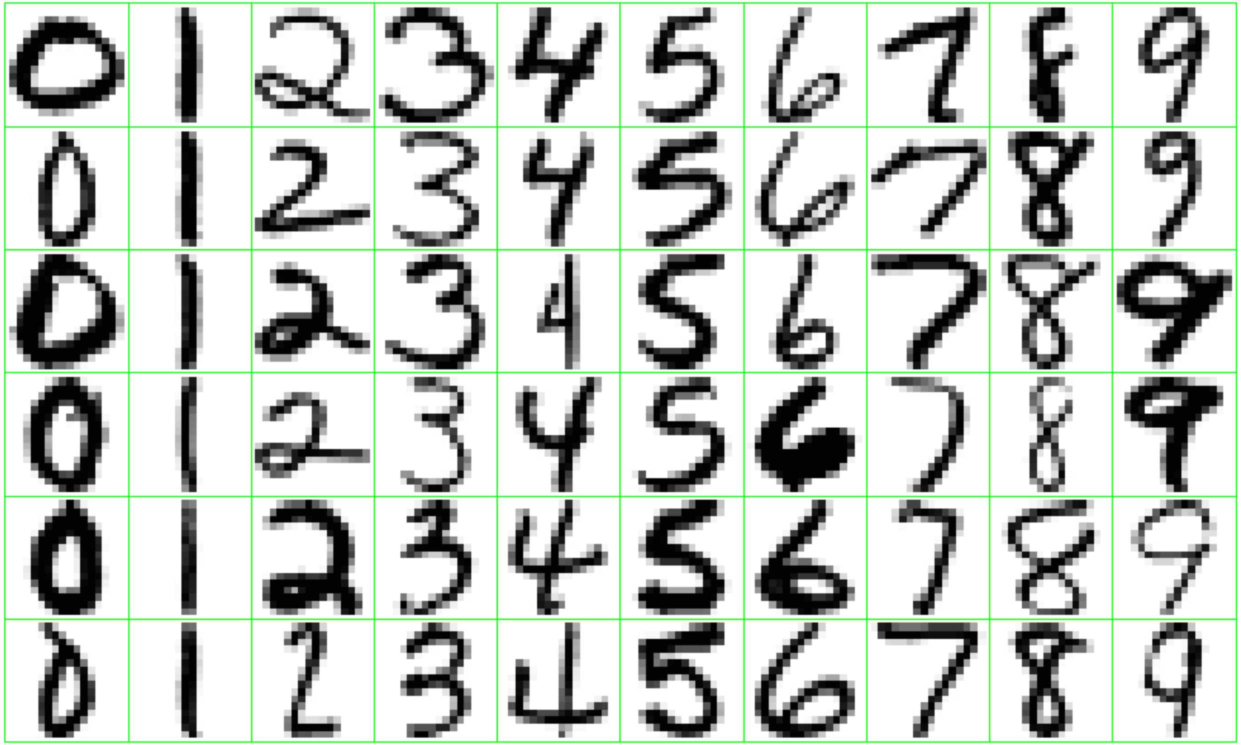
Average percentage of words or characters in an email message equal to the indicated word or character. We have chosen the words and characters showing the largest difference between `spam` *and* `email`.

$\mathbf{x} \in \mathbb{R}^{57}$

$y \in \{0, 1\}$

Example: Identify the numbers in a handwritten zip code.

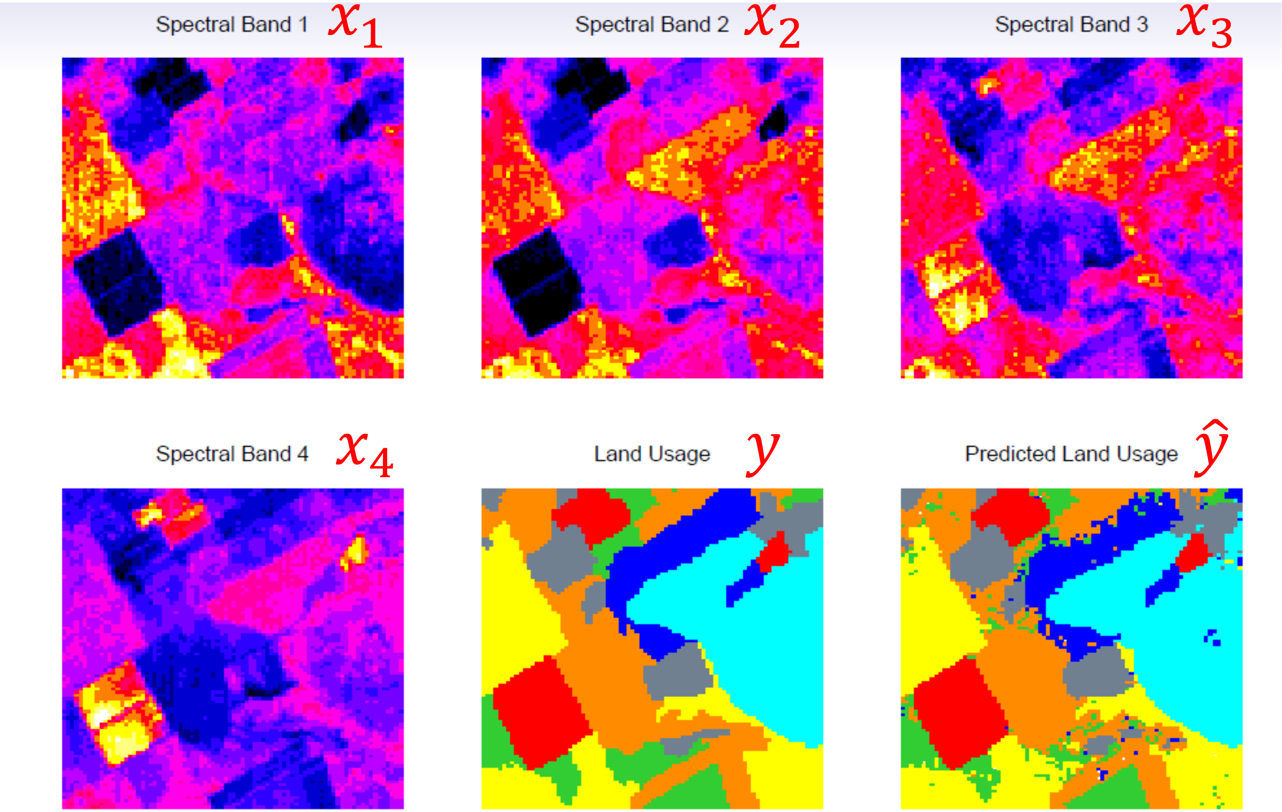
Modified National Institute of Standards and Technology (MNIST) dataset:



$\mathbf{x} \in \{0, \ldots, 255\}^{28 \times 28}$

$y \in \{"0", "1", \ldots, "9"\}$

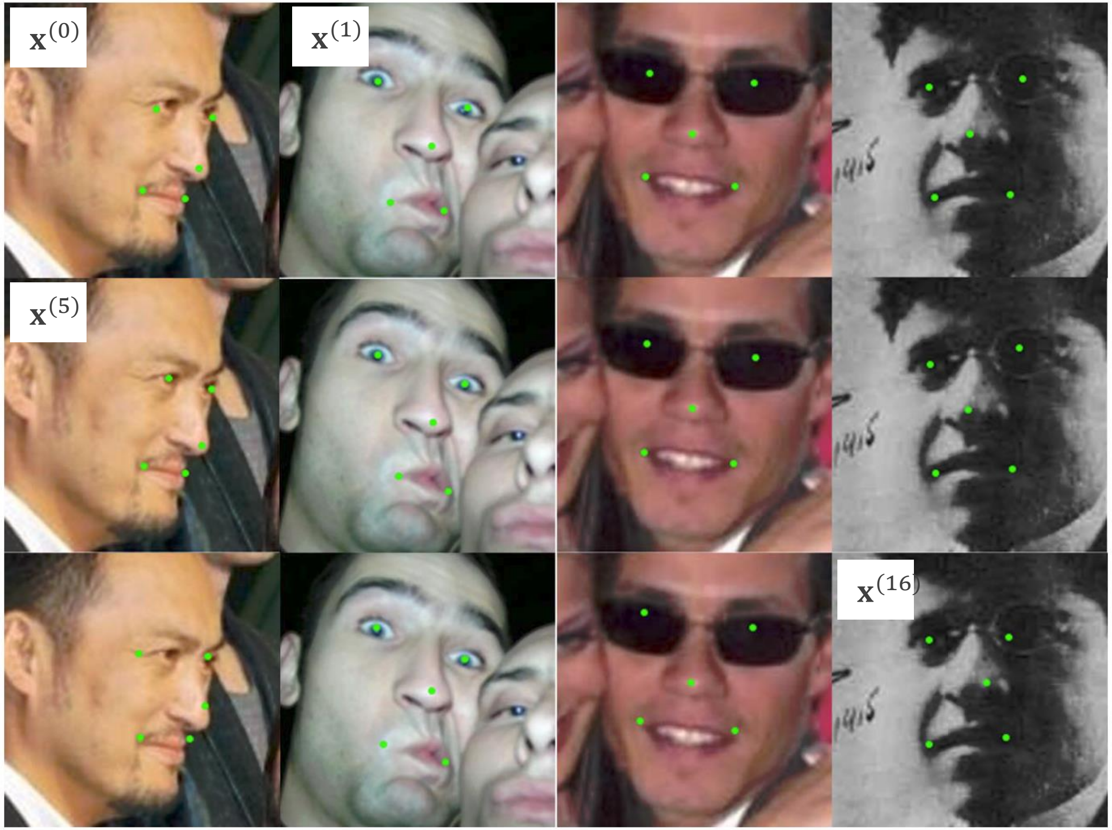
Example: Land use prediction via hyperspectral imaging.



Consumer cameras *vs* hyperspectral cameras: 3 *vs* >> 3 channels

$$Usage \in \{red\ soil,\ cotton,\ vegetation\ stubble,\ mixture,\ gray\ soil,\ damp\ gray\ soil\}$$

Supervised Learning: Regression



Goal of regression:

Assign a number to each input, e.g., horizontal coordinate of a nose tip.

Loosely, in ML, it is also called a "label."

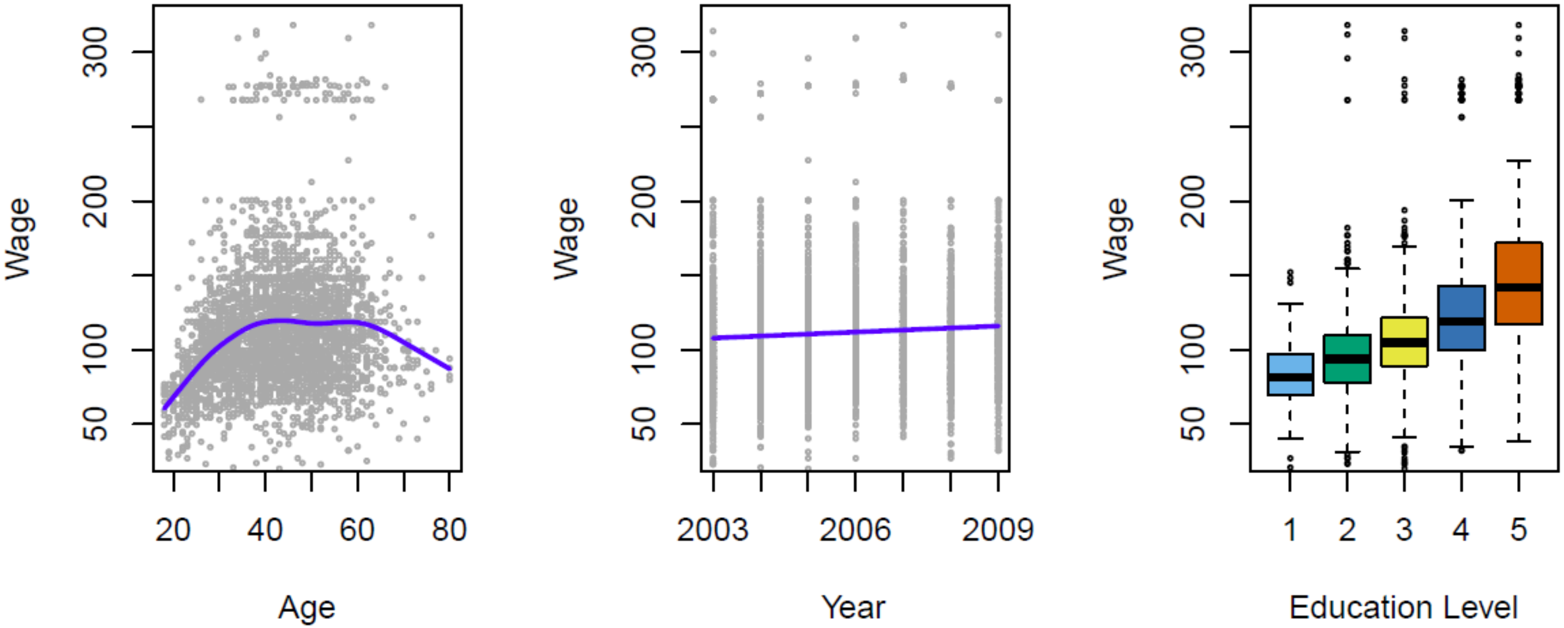
🡐 Given a facial image, it returns the 2-D location for each key point of the face.

Yi Sun, Xiaogang Wang, Xiaoou Tang, Deep Convolutional Network Cascade for Facial Point Detection, *CVPR*, 2013.

Example: Wage prediction—Income survey data for males from the central Atlantic region of the USA in 2009.

y:

x:



Supervised Learning: Definition

- **Terminologies:**
  - Training data: $\mathcal{D}_{\text{tr}} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$
  - Test data: $\mathcal{D}_{\text{te}} = \{(\mathbf{x}_i, y_i)\}_{i=n+1}^{n+m}$
  - True model f_{true}: $y = [f_{\text{true}}(\mathbf{x}) \text{ with noise}]$
  - Learned model f: $\hat{y} = f(\mathbf{x})$ (^ : hat/cap, estimated/predicted)
- **Goal**: Given a set of training data $\mathcal{D}_{\text{tr}}$ as the inputs, the learning task computes a learned model $f(\cdot)$ such that it can generate accurate predicted outputs

$$\hat{y}_i = f(\mathbf{x}_i), \quad i = n+1, \ldots, n+m,$$

from a set of new inputs $\{\mathbf{x}_i\}_{i=n+1}^{n+m}$ of the test data $\mathcal{D}_{\text{te}}$ whose labels $\{y_i\}_{i=n+1}^{n+m}$ have never been taken into account when the model is computed.

Quantifying the Accuracy of Prediction

- Quantify the accuracy of the learned model by a *loss function* (or cost/objective function), based on predicted output, $\hat{y}_i$, and the true output, y_i, namely, $L(\hat{\mathbf{y}}, \mathbf{y})$.

- A typical choice for the loss function for a continuous-valued output is the *mean squared error*:

$$L(\hat{\mathbf{y}}, \mathbf{y}) = \frac{1}{m} \sum_{i=n+1}^{n+m} (\hat{y}_i - y_i)^2$$

- Key ML assumption: Test data shouldn't have been seen before (at the training stage), or there will be overfit.

Simplest Example: Linear Model

Data: $(x_i, Y_i), \quad i = 1, \ldots, n$

Explicitly write out all *n* eqs:

random $\mathbb{E}[e_i] = 0$

Model: $Y_i = \beta_0 + \beta_1 x_i + e_i$

intercept

noise: measurement noise,
biological variation

dependent var.
/observation

independent var./predictor

$\boldsymbol{\beta} = [\beta_0, \beta_1]^T$ is the parameter vector/weights.

$\mathbb{E}[Y_i] = \beta_0 \quad + \beta_1 x_i =$ linear combination of unknowns β_0 and β_1 with known coefficient 1 and x_i.

Linear Model in Matrix-Vector Form

$$\boxed{\begin{array}{l} Y_i = \beta_0 + \beta_1 x_i + e_i, \\ i = 1, \ldots, n. \end{array}}$$

$$\mathbf{Y} = \begin{bmatrix} Y_1 \\ \vdots \\ Y_n \end{bmatrix}_{n\times 1}, \quad \mathbf{X} = \left[\begin{array}{c:c} \underbrace{\begin{matrix} 1 \\ \vdots \\ 1 \end{matrix}}_{\mathbb{1}} & \underbrace{\begin{matrix} x_1 \\ \vdots \\ x_n \end{matrix}}_{\mathbf{x}} \end{array}\right]_{n\times 2}, \quad \boldsymbol{\beta} = \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix}_{2\times 1}, \quad \mathbf{e} = \begin{bmatrix} e_1 \\ \vdots \\ e_n \end{bmatrix}_{n\times 1}$$

$$\mathbf{Y} = \underset{\substack{\uparrow \\ \text{data matrix}}}{\mathbf{X}}\boldsymbol{\beta} + \mathbf{e} \qquad \text{“Matrix–vector form”}$$

Linear Model with Multiple Predictors / Features

- Multiple (Linear) Regression Model:

$$Y_i = \sum_{j=1}^{p} x_{ij}\beta_j + e_i, \quad i = 1, \ldots, n.$$

vector of random elements

$$\mathbf{Y}_{n\times 1} = \mathbf{X}_{n\times p}\boldsymbol{\beta}_{p\times 1} + \mathbf{e}_{n\times 1}$$

Explicitly write out each element:

Linear Regression Example

$$Y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + e_i, \quad i = 1, \ldots, 50.$$

Y_i : grade

x_{i1} : time spent on HW

x_{i2} : time spent on review

Fill in the elements:

$$\begin{bmatrix} Y_1 \\ \vdots \\ Y_{50} \end{bmatrix} = \begin{bmatrix} 1 & x_{1,1} & x_{1,2} \\ \vdots & \vdots & \vdots \\ 1 & x_{50,1} & x_{50,2} \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \end{bmatrix} + \begin{bmatrix} e_1 \\ \vdots \\ e_{50} \end{bmatrix}$$

How to estimate model parameters $\beta_0, \beta_1,$ and β_2? Least-Squares!

Linear Regression Example

$$Y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + e_i, \quad i = 1, \ldots, 50.$$

Y_i : grade

x_{i1} : time spent on HW

x_{i2} : time spent on review

$$\begin{bmatrix} Y_1 \\ \vdots \\ Y_{50} \end{bmatrix} = \begin{bmatrix} 1 & x_{1,1} & x_{1,2} \\ \vdots & \vdots & \vdots \\ 1 & x_{50,1} & x_{50,2} \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \end{bmatrix} + \begin{bmatrix} e_1 \\ \vdots \\ e_{50} \end{bmatrix}$$

How to estimate model parameters β_0, β_1, and β_2? Least-Squares!

Least-Squares for Parameter Estimation

Problem Setup: $\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{e}$, where $\mathbf{X} = [x_{ij}]_{n \times p} \triangleq [\boldsymbol{\xi}_1, \cdots, \boldsymbol{\xi}_p]$.

Estimate $\boldsymbol{\beta}$ such that $J(\boldsymbol{b}) = \|\mathbf{Y} - \mathbf{X}\boldsymbol{b}\|^2$ is minimized.

$$\text{or } J(\boldsymbol{b}) = \sum_{i=1}^{n} (Y_i - \sum_{j=1}^{p} x_{ij} b_j)^2$$

This is called the *least-squares* procedure.

Least-Squares via Vector Calculus

$$\boxed{\text{Recall: } J(\boldsymbol{b}) = \|\mathbf{Y} - \mathbf{X}\boldsymbol{b}\|^2}$$

$$\textbf{Method 1: } \nabla_{\boldsymbol{b}} J(\boldsymbol{b}) = \Big|_{\boldsymbol{b}=\hat{\boldsymbol{\beta}}} 0,$$

$$\nabla_{\boldsymbol{b}} J(\boldsymbol{b}) = 2\left[-\mathbf{X}^T(\mathbf{Y} - \mathbf{X}\boldsymbol{b})\right] = \Big|_{\boldsymbol{b}=\hat{\boldsymbol{\beta}}} \mathbf{0}$$

$$\boxed{\mathbf{X}^T\mathbf{Y} = \mathbf{X}^T\mathbf{X}\hat{\boldsymbol{\beta}}} \qquad \boxed{\mathbf{X}^T(\mathbf{Y} - \mathbf{X}\hat{\boldsymbol{\beta}}) = \mathbf{0}}$$

(Error orthogonal to data)

Normal Equation (N.E.)

Vector calculus cheat sheet (p. 521–527): https://www.cs.cmu.edu/~epxing/Class/10701-08s/recitation/mc.pdf

Least-Squares via Partial Differentiation (optional)

If linear algebra is not used, the derivation can be much more involved:

Method 2 :

Recall: $J(\boldsymbol{b}) = \sum_{i=1}^{n} \left(Y_i - \sum_{j=1}^{p} x_{ij} b_j \right)^2$

$$\frac{\partial J}{\partial b_k} = \sum_{i=1}^{n} 2(Y_i - \sum_{j=1}^{p} x_{ij} b_j) \underbrace{\frac{\partial}{\partial b_k}\Big(- (\cdots + x_{ik} b_k + \cdots)\Big)}_{-x_{ik}}$$

$$= |_{b_j = \hat{\beta}_j} 0, \quad k = 1, \cdots, p$$

$$\Longleftrightarrow \sum_i Y_i x_{ik} = \sum_i \sum_j x_{ij} \hat{\beta}_j x_{ik} \Longleftrightarrow \boxed{\mathbf{X}^T \mathbf{Y} = \mathbf{X}^T \mathbf{X} \hat{\boldsymbol{\beta}}}$$

Normal Equation (N.E.)

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{Y}$$

(when $\mathbf{X}$ is full rank)

where $\mathbf{X}^T \mathbf{Y} = \left[\sum_{i=1}^{n} x_{ik} Y_i \right]_{p \times 1}$, $\mathbf{X}^T \mathbf{X} = \left[\sum_{i=1}^{n} x_{ij} x_{ik} \right]_{p \times p}$

$$\mathbf{X}^T \mathbf{X} \hat{\boldsymbol{\beta}} = \left[\sum_{j=1}^{p} \left(\sum_{i=1}^{n} x_{ij} x_{ik} \right) \hat{\beta}_j \right]_{p \times 1}$$

Ex: Linear Model for Learning and Prediction

- Training data (3 data points / a random sample of size 3):
  - Feature/predictor 1: (2, 1, 1). Feature/predictor 2: (1, 2, 1).
  - Labels: (1, 1, 1).
- Test data (2 data points / a random sample of size 2):
  - Feature 1: (1.2, 1.8). Feature 2: (0.9, 1.3).
  - Labels: (0.9, 0.8).
- Tasks:
  - a) Learn a linear model without intercept.
  - b) Evaluate the mean squared errors (MSEs) of training and testing.

a) $$\mathbf{X} = \begin{array}{c} \left[\begin{array}{cc} 2 & 1 \\ 1 & 2 \\ 1 & 1 \end{array}\right] \\ \begin{array}{cc} \text{feat.1} & \text{feat. 2} \end{array} \end{array} \qquad \mathbf{Y} = \left[\begin{array}{c} 1 \\ 1 \\ 1 \end{array}\right] \begin{array}{l} \text{data point \#1} \\ \text{data point \#2} \\ \end{array} \qquad \mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{e} \qquad (\mathbf{X}, \mathbf{Y}) :$$ training data

Estimated/ trained model parameters:

$$\hat{\boldsymbol{\beta}} = \left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{Y}$$

$$= \left(\begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \end{bmatrix}\begin{bmatrix} 2 & 1 \\ 1 & 2 \\ 1 & 1 \end{bmatrix}\right)^{-1}\left(\begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \end{bmatrix}\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}\right)$$

$$= \begin{bmatrix} 6 & 5 \\ 5 & 6 \end{bmatrix}^{-1}\begin{bmatrix} 4 \\ 4 \end{bmatrix} = \begin{bmatrix} 6 & -5 \\ -5 & 6 \end{bmatrix} \cdot \frac{1}{11} \cdot \begin{bmatrix} 4 \\ 4 \end{bmatrix}$$

$$= \frac{4}{11}\begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

Predicted output based on training data:

$$\hat{\mathbf{Y}} = \mathbf{X}\hat{\boldsymbol{\beta}} = \begin{bmatrix} 2 & 1 \\ 1 & 2 \\ 1 & 1 \end{bmatrix}\frac{4}{11}\begin{bmatrix} 1 \\ 1 \end{bmatrix} = \frac{1}{11}\begin{bmatrix} 12 \\ 12 \\ 8 \end{bmatrix} \neq \mathbf{Y}\text{, or}$$

$$\mathbf{H} = \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T = \frac{1}{11}\begin{bmatrix} 2 & 1 \\ 1 & 2 \\ 1 & 1 \end{bmatrix}\begin{bmatrix} 6 & -5 \\ -5 & 6 \end{bmatrix}\begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \end{bmatrix} = \frac{1}{11}\begin{bmatrix} 10 & -1 & 3 \\ -1 & 10 & 3 \\ 3 & 3 & 2 \end{bmatrix}$$

$$\hat{\mathbf{Y}} = \mathbf{H}\mathbf{Y} = \frac{1}{11}\begin{bmatrix} 12 \\ 12 \\ 8 \end{bmatrix}$$

b) Training error (in MSE):

$$\frac{1}{3}\sum_{i=1}^{3}\left(y_i - \mathbf{x}_i^T\hat{\boldsymbol{\beta}}\right)^2 = \frac{1}{3}\|\mathbf{Y}-\hat{\mathbf{Y}}\|^2 = \frac{1}{3}\|\mathbf{Y}-\mathbf{X}\hat{\boldsymbol{\beta}}\|^2$$

$$= \frac{1}{3}\cdot\frac{1}{11^2}\left\|\begin{bmatrix} 12-11 \\ 12-11 \\ 8-11 \end{bmatrix}\right\|^2 = \frac{1}{3}\cdot\frac{1}{11^2}(1+1+9) = \frac{1}{3}\cdot\frac{1}{11} = 0.03$$

Testing error (in MSE):

$$\mathbf{X}_{\text{test}} = \begin{bmatrix} 1.2 & 0.9 \\ 1.8 & 0.3 \end{bmatrix} \quad \mathbf{Y}_{\text{test}} = \begin{bmatrix} 0.9 \\ 0.8 \end{bmatrix} \quad (\mathbf{X}_{\text{test}}, \mathbf{Y}_{\text{test}}):$$

testing data

$$\frac{1}{2}\sum_{i=4}^{5}\left(y_i - \mathbf{x}_i^T\hat{\boldsymbol{\beta}}\right)^2 = \frac{1}{2}\|\mathbf{Y}_{\text{test}}-\hat{\mathbf{Y}}_{\text{test}}\|^2 = \frac{1}{2}\|\mathbf{Y}_{\text{test}}-\mathbf{X}_{\text{test}}\hat{\boldsymbol{\beta}}\|^2$$

$$= \frac{1}{2}\left\|\begin{bmatrix} 0.9 \\ 0.8 \end{bmatrix} - \begin{bmatrix} 1.2 & 0.9 \\ 1.8 & 0.3 \end{bmatrix}\left(\frac{4}{11}\begin{bmatrix} 1 \\ 1 \end{bmatrix}\right)\right\|^2 = \frac{1}{2}\left\|\begin{bmatrix} 0.14 \\ 0.04 \end{bmatrix}\right\|^2 = 0.01$$

Geometric Interpretation of Linear Models

Wait a minute … more on Linear Algebra

- Linear independence
- Vector space
- Dimension of vector space
- Rank of a matrix

(A comprehensive treatment of linear algebra can be found in Scheffe's appendices. You may also consult your favorite linear algebra textbook.)

Linear Independence of a Set of Vectors

- Given $\{\mathbf{v}_1, \ldots, \mathbf{v}_n\}$. Defs:

$$\alpha_1\mathbf{v}_1 + \cdots + \alpha_n\mathbf{v}_n = \mathbf{0} \Rightarrow \alpha_i = 0, \forall i$$ (linearly independent)

$$\alpha_1\mathbf{v}_1 + \cdots + \alpha_n\mathbf{v}_n = \mathbf{0} \Rightarrow \text{not all } \alpha_i = 0$$ (linearly dependent)

- For "linearly dependent" case (when $\alpha_1 \neq 0$), we may write:

$$\mathbf{v}_1 = \beta_2\mathbf{v}_2 + \cdots + \beta_n\mathbf{v}_n$$

Why?

- Ex: $\mathbf{v}_1 = \begin{bmatrix}1 & 2 & 1\end{bmatrix}^T, \mathbf{v}_2 = \begin{bmatrix}1 & 0 & 1\end{bmatrix}^T.$

$$\alpha_1\begin{bmatrix}1\\2\\1\end{bmatrix} + \alpha_2\begin{bmatrix}1\\0\\1\end{bmatrix} = \mathbf{0} \quad \Rightarrow \begin{cases}\alpha_1+\alpha_2=0\\2\alpha_1+0=0\\\alpha_1+\alpha_2=0\end{cases} \Rightarrow \begin{cases}\alpha_1=0\\\alpha_2=0\end{cases} \Rightarrow \text{linearly independent}$$

Linear Independence of a Set of Vectors (cont'd)

- Ex: $\mathbf{v}_1 = \begin{bmatrix} 1 & 2 & 1 \end{bmatrix}^T, \mathbf{v}_4 = \begin{bmatrix} -2 & -4 & -2 \end{bmatrix}^T.$

$$\mathbf{v}_4 = -2\mathbf{v}_1 \Rightarrow \text{linearly dependent}$$

- Ex: $\mathbf{v}_1 = \begin{bmatrix} 1 & 2 & 1 \end{bmatrix}^T, \mathbf{v}_2 = \begin{bmatrix} 1 & 0 & 1 \end{bmatrix}^T, \mathbf{v}_3 = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}^T.$

$$\mathbf{v}_1 = \mathbf{v}_2 + 2\mathbf{v}_3 \Rightarrow \text{linearly dependent}$$

Vector Space

- Def: Vector space: A set, V, of all vectors that are linear combination of $\{\mathbf{v}_i\}_{i=1}^{n}$, i.e.,

$$\boxed{V = \left\{ \mathbf{v} = \sum_{i=1}^{n} \alpha_i \mathbf{v}_i ,\ \alpha_i \in \mathbb{R} \right\}.}$$

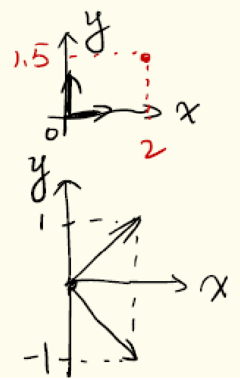
, can be replaced by : or |

$\mathbf{v}_i$'s are said to span the vector space, i.e., $V = \text{span}\{\mathbf{v}_1, \ldots, \mathbf{v}_n\}.$

- Ex:

$$V^{(1)} = \left\{ \alpha_1 \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \alpha_2 \begin{bmatrix} 0 \\ 1 \end{bmatrix} ,\ \alpha_i \in \mathbb{R} \right\} = \mathbb{R}^2$$

$$V^{(2)} = \left\{ r_1 \begin{bmatrix} 1 \\ 1 \end{bmatrix} + r_2 \begin{bmatrix} 1 \\ -1 \end{bmatrix} ,\ r_i \in \mathbb{R} \right\} = \mathbb{R}^2$$



Vector Space (cont'd)

- Def: Vector space: A set, V, of all vectors that are linear combination of $\{\mathbf{v}_i\}_{i=1}^{n}$, i.e.,

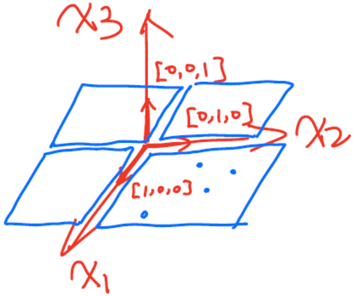
$$\boxed{V = \left\{\mathbf{v} = \sum_{i=1}^{n} \alpha_i \mathbf{v}_i ,\ \alpha_i \in \mathbb{R}\right\}.}$$

, can be replaced by : or |

$\mathbf{v}_i$'s are said to span the vector space, i.e., $V = \mathrm{span}\{\mathbf{v}_1, \ldots, \mathbf{v}_n\}.$

- Ex: $$V^{(3)} = \left\{\alpha_1 \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + \alpha_2 \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix},\ \alpha_i \in \mathbb{R}\right\}$$

$$= \left\{\begin{bmatrix} \alpha_1 \\ \alpha_2 \\ 0 \end{bmatrix}\right\} = \text{Hori. plane of 3-D space} \subset \mathbb{R}^3$$



Basis for Vector Space

- Def: A basis for V is a set of linearly independent vectors that span V.
- Ex: Q1. What is V? Q2. Are vectors linearly independent?

$$\left\{ \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ -1 \end{bmatrix} \right\} \qquad \left\{ \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right\}$$

$$\left\{ \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right\} \qquad \left\{ \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 2 \end{bmatrix} \right\}$$

Basis for Vector Space

- Def: A basis for V is a set of linearly independent vectors that span V.
- Ex: Q1. What is V? Q2. Are vectors linearly independent?

$$\left\{ \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ -1 \end{bmatrix} \right\}$$ yes

$$\left\{ \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right\}$$ yes

$$\left\{ \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right\}$$ yes

$$\left\{ \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 2 \end{bmatrix} \right\}$$ no

Dimension of Vector Space

- Def: The dimension of vector space V is the number of vectors in any/a basis for V (or the # of independent vectors in V).
- Column/row rank: The dimension of column/row vector space, respectively.
- Ex: What's the column rank of matrix

$$\mathbf{X} = \begin{bmatrix} 1 & 1 & 0 \\ 2 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}?$$

It's another way to ask: what's the dimension of column vector space

$$V = \left\{ \mathbf{v} = \alpha_1 \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} + \alpha_2 \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} + \alpha_3 \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \ \alpha_i \in \mathbb{R} \right\}?$$

Dimension of Vector Space (cont'd)

◆ Approach 1: By observation, we notice that any (and only) two pairs of vectors spanned V are linearly independent. Hence, we can immediately write out at least three bases:

$$\left\{\begin{bmatrix}1\\2\\1\end{bmatrix}, \begin{bmatrix}1\\0\\1\end{bmatrix}\right\} \quad \text{or} \quad \left\{\begin{bmatrix}1\\2\\1\end{bmatrix}, \begin{bmatrix}0\\1\\0\end{bmatrix}\right\} \quad \text{or} \quad \left\{\begin{bmatrix}1\\0\\1\end{bmatrix}, \begin{bmatrix}0\\1\\0\end{bmatrix}\right\}$$

Hence, the column rank of $\mathbf{X}$ or dimension of vector space V is 2.

◆ Approach 2: Define the three vectors to be $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$, respectively.

$$\begin{aligned} V &= \left\{\mathbf{v} = \alpha_1(\mathbf{v}_2 + 2\mathbf{v}_3) + \alpha_2\mathbf{v}_2 + \alpha_3\mathbf{v}_3\right\} \\ &= \left\{\mathbf{v} = (\alpha_1 + \alpha_2)\mathbf{v}_2 + (2\alpha_1 + \alpha_3)\mathbf{v}_3\right\}. \end{aligned}$$

$\mathbf{v}_2 \perp \mathbf{v}_3 \Rightarrow$ they are linearly independent. So the dim/rank is 2.

Geometric Interpretation of Linear Models (for real)

Least-Squares for Parameter Estimation

Problem Setup: $\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{e}$, where $\mathbf{X} = [x_{ij}]_{n\times p} \triangleq \left[\boldsymbol{\xi}_1 \middle| \cdots \middle| \boldsymbol{\xi}_p\right]$.

Estimate $\boldsymbol{\beta}$ such that $J(\boldsymbol{b}) = \|\mathbf{Y} - \mathbf{X}\boldsymbol{b}\|^2$ is minimized.

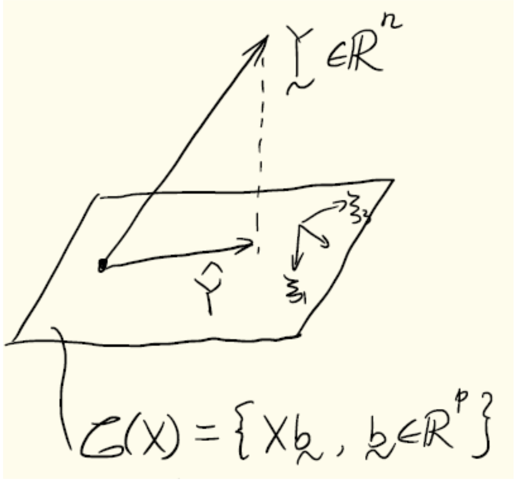
$$\text{or } J(\boldsymbol{b}) = \sum_{i=1}^{n}(Y_i - \sum_{j=1}^{p} x_{ij}b_j)^2$$

The solution of Least-Squares is given by the Normal Equation:

$$\boxed{\mathbf{X}^T(\mathbf{Y} - \mathbf{X}\hat{\boldsymbol{\beta}}) = \mathbf{0}}$$

Geometric Interpretation of Least-Squares (LS)

- Remark: The LS procedure finds a vector $\widehat{\boldsymbol{\beta}}$, which results $\widehat{\mathbf{Y}} = \mathbf{X}\widehat{\boldsymbol{\beta}}$ in the column (vector) space of $\mathbf{X}$, i.e., $\mathcal{C}(\mathbf{X}) = \{\mathbf{Xb}, \mathbf{b} \in \mathbb{R}^p\}$ such that
  - $\widehat{\mathbf{Y}}$ is as close as possible to $\mathbf{y}$, or
  - $\left(\mathbf{Y} - \widehat{\mathbf{Y}}\right) \perp \mathcal{C}(\mathbf{X})$.



$\mathbf{Xb} =$

$$
\begin{aligned}
& (\mathbf{Y} - \hat{\mathbf{Y}}) \perp \mathcal{C}(\mathbf{X}) \\
\Longleftrightarrow\ & (\mathbf{Y} - \hat{\mathbf{Y}}) \perp \mathbf{Xb}, \quad \forall \mathbf{b} \in \mathbb{R}^p \\
\Longleftrightarrow\ & \boldsymbol{\xi}_j^T (\mathbf{Y} - \hat{\mathbf{Y}}) = 0, \quad j = 1, \cdots, p \\
\Longleftrightarrow\ & [\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_p]^T (\mathbf{Y} - \mathbf{X}\hat{\beta}) = \mathbf{0} \\
\Longleftrightarrow\ & \mathbf{X}^T\mathbf{Y} = \mathbf{X}^T\mathbf{X}\hat{\beta}
\end{aligned}
$$

Properties of Least-Square Estimate

If $\text{rank}(\mathbf{X}) \triangleq r = p$ ① $\boxed{\hat{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}}$ is unique soluntion.

$$\mathbb{E}[\hat{\beta}] = \mathbb{E}[(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}] = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T(\mathbf{X}\beta) = \beta \text{ (unbiased)}$$

② $\hat{\mathbf{Y}} = \mathbf{X}\hat{\beta} = \underbrace{\mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T}\mathbf{Y} = \mathbf{HY}$

$\mathbf{H}$: "hat" matrix, or "orthogonal projector." $\mathbf{H}^n = \mathbf{H}$. Why?

Ex: Linear Model for Learning and Prediction

- Training data (3 data points / a random sample of size 3):
  - Feature/predictor 1: (2, 1, 1). Feature/predictor 2: (1, 2, 1).
  - Labels: (1, 1, 1).
- Test data (2 data points / a random sample of size 2):
  - Feature 1: (1.2, 1.8). Feature 2: (0.9, 1.3).
  - Labels: (0.9, 0.8).
- Recall parameter estimation results:
  - Estimated weights: $\hat{\beta} = \left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{Y} = \frac{4}{11}[1, 1]^T$
  - Predicted outcome: $\hat{\mathbf{Y}} = \mathbf{X}\hat{\beta} = \frac{1}{11}[12, 12, 8]^T$
  - Sum of squared error/residue, or training error: $\|\mathbf{Y} - \hat{\mathbf{Y}}\|^2 = \frac{1}{11}$

Geometric Illustration of Data and Learned Model