Statistics and Data Science 365 / 565

# Data Mining and Machine Learning

February 3

Yale

# Outline

- High level ideas
- Supervised learning: regression/classificatioon
- Unsupervised learning: finding structure/visualization
- How do we represent data?
- How do we assess if we've learned well?
- Notation
- Notebook

# Concepts: Learning examples

Given info about a house, predict its value

Given an image, predict the digit, or predict if it is offensive

Given some text, find underlying themes

Given some emails, automatically tag and group them

Given a word, find its translation

How do we approach this?

# Concepts

Supervised learning: given data in the form of observations and labels, learn

- predict house value
- predict image label
- word translation
- does studying more lead to better grades (causality)

# Concepts

Supervised learning: given data in the form of observations and labels, learn

- predict house value
- predict image label
- word translation
- does studying more lead to better grades (causality)

Unsupervised learning: given generic data, find some structure. **No labels to guide learning**

- find themes in text
- automatically group similar emails
- word translation

# Concepts

How do we represent our data?

Generally we will represent data as vectors, matrices, or tensors.

How do we know if we learned well?

Many ways: start with losses

# Representing data

Notation: We use vectors, matrices, or tensors

Vectors: $x \in \mathbb{R}^d$ is a vector in $d$-dimensions. $x_{(i)}$ is the $i^{th}$ coordinate. $x$ is taken as a column. $x^T$ as a transpose is a row.

Matrices: $M \in \mathbb{R}^{n \times d}$ is a $n \times d$ matrix. $M_{(ij)}$ is the entry of $M$ in the $i^{th}$ row and $j^{th}$ column. Column vector is $d \times 1$, row vector is $1 \times d$.

Tensors: $T \in \mathbb{R}^{n \times d \times k}$ is a three-dimensional tensor. Similar.

vector

$$\begin{pmatrix} x_{(1)} \\ x_{(2)} \\ \vdots \\ x_{(d)} \end{pmatrix} \in \mathbb{R}^d$$

Matrix

$$\begin{pmatrix} M_{(11)} & M_{(12)} & \cdots & M_{(1d)} \\ \vdots & & \ddots & \vdots \\ M_{(n1)} & \cdots & \cdots & M_{(nd)} \end{pmatrix}$$

$$M_{(1,2)} \equiv M_{(ij)} \quad i=1,\ j=2$$

# Representing data

The set of data will usually be represented in caligraphy.

Data: $\mathcal{X}$: information about homes, images, stock prices

Labels: $\mathcal{Y}$: price of home, label of image, future stock prices

Terminology:

Regression: $\mathcal{Y}$ is continuous
- price of home
- stock price

Classification: $\mathcal{Y}$ is discrete
(unordered discrete)
- image label
- stock goes up or down

**Home example:** Let $x \in \mathcal{X}$ be information about a home. Often take $\mathcal{X} = \mathbb{R}^d$. $x_{(i)}$ is the $i^{th}$ feature of the home.
(i)

$\mathcal{Y} = \mathbb{R}$, the price of the house.

**Images example:** $I \in \mathcal{X}$ be an image. Can take $\mathcal{X} = \mathbb{R}^{n \times d}$ or $\mathbb{R}^{n \times d \times 3}$.

$I_{(ab)}$ is the pixel value of an image at ab

$\mathcal{Y} = \{dog, cat, human, car\}$ or $[k] = \{1, 2, \ldots, k\}$. Associate each label with a unique number.

Can combine: $\mathcal{X}_1 = \mathbb{R}^d$, $\mathcal{X}_2 = \mathbb{R}^{m \times \ell}$

$\mathcal{X} = \mathcal{X}_1 \times \mathcal{X}_2$

$\Rightarrow x \in \mathcal{X}$, $(x_1, x_2)$ where $x_1 \in \mathcal{X}_1$, $x_2 \in \mathcal{X}_2$

(tuple)

**More data:** If have $n$ examples of data take all data to be $\mathcal{Z} = \{(x_i, y_i)\}_{i=1}^{n}$

**Homes:** $x_i$ is data representation of $i^{th}$ home. $y_i$ is value of $i^{th}$ home

**Compactify:** $X = \mathrm{matrix}[x_i^T] \in \mathbb{R}^{n \times d}$ and $y = \mathrm{vector}[y_i] \in \mathbb{R}^n$.

In this case $y_{(i)} = y_i$ and $X_{(ij)} = (x_i)_{(j)}$

$$X = \begin{pmatrix} \text{---} & x_1^T & \text{---} \\ \text{---} & x_2^T & \text{---} \\ & \vdots & \\ \text{---} & x_n^T & \text{---} \end{pmatrix} \qquad y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}$$

# Back to learning

Many goals with learning.

## Predict well

**Example:** Predict future stock prices.

## Reject outliers

**Example:** Identify malicious devices on a network.

## Identify if a feature actually matters

**Example:** Does adding a pool increase the value of my home?

Our focus to start on predicting well.

Last example is causal inference. Very important.

Nomenclature: Inference is over-used.

Statistics: Generally assess if a feature matters (not necessarily in a causal way)

Machine learning (Bayesian stats, neural nets): Fill in the unknowns (could mean predict too)

How do we learn and know we learned well?

How do we learn and know we learned well?

When have supervised learning, labels guide the way.

In supervised learning we want to find a function $f : \mathcal{X} \mapsto \mathcal{Y}$

Learn from examples yields an estimate $\widehat{f}$