Introduction to Machine Learning

Benjamin Peherstorfer

Fall 2020

Slides adapted from David Sontag, Andrew Ng

Today

- Last time
  - What is machine learning
  - Three key questions of ML
    - Hypothesis space
    - What is a “good” hypothesis
    - Algorithm/computational methods to find good/best hypothesis

- Today
  - Least-squares regression (or recap on important math concepts)
  - (Stochastic) gradient descent
  - Recap concepts from probability theory (https://see.stanford.edu/materials/aimlcs229/cs229-prob.pdf)

- **Recommended reading:** Recap concepts from linear algebra: https://see.stanford.edu/materials/aimlcs229/cs229-linalg.pdf

Summary: Key questions in ML

- How do we choose a hypothesis space?
  – Often we use **prior knowledge (inductive bias)** to guide this choice

- How can we gauge the accuracy of a hypothesis on data?
  – Define a **loss (cost) function** and compute empirical loss on training data
  - ***Learning theory*** will help us quantify our ability to ***generalize*** as a function of the amount of training data and the hypothesis space **to unseen (test) data**

- How do we find the best hypothesis in the hypothesis space?
  – This is an **algorithmic** question, the main topic of computer science

- How to model applications as machine learning problems? (engineering challenge)

Supervised learning

Supervised learning

- Data
  - Inputs ("features") $\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(N)} \in \mathscr{X} \subset \mathbb{R}^n$
  - Outputs ("targets") $\boldsymbol{y}^{(1)}, \ldots, \boldsymbol{y}^{(N)} \in \mathscr{Y} \subset \mathbb{R}^{n'}$
- Training data set $\mathscr{D} = \{(\boldsymbol{x}^{(1)}, \boldsymbol{y}^{(1)}), \ldots, (\boldsymbol{x}^{(N)}, \boldsymbol{y}^{(N)})\}$
- Goal in supervised learning is to find hypothesis $h : \mathscr{X} \to \mathscr{Y}$ so that $h(x)$ is a good prediction of y
- Decisions we have to make
  - Choose a space of hypotheses $\mathscr{H}$
  - Decide what we mean with "good"
  - Develop a method to find the best hypothesis h^*
  - There are other decisions to be made that we ignore for now (e.g.?)

1. Hypothesis space

- As an initial choice, let us consider functions with linear dependence between $\boldsymbol{\theta}$ and $\boldsymbol{x}$

$$\mathscr{H} = \{h_\theta \,:\, h_\theta(\boldsymbol{x}) = \sum_{i=0}^{n} \theta_i x_i = \boldsymbol{\theta}^T \boldsymbol{x}\}$$

  - Parameter $\boldsymbol{\theta} = [\theta_0, \theta_1, \ldots, \theta_n]^T \in \Theta$
  - Convention $x_0 = 1$

- Other hypothesis spaces coming up...

2. Cost (loss) function

- We now need a way to compare functions in the space $\mathscr{H}$ w.r.t. training data
- A widely used choice is the least-squares cost function

$$L(\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)}) = \left(h_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)}) - \boldsymbol{y}^{(i)}\right)^2$$

- The **empirical cost** is the (scaled) cost function applied to train data

$$J(\boldsymbol{\theta}) = \frac{1}{2} \sum_{i=1}^{N} \left(h_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)}) - \boldsymbol{y}^{(i)}\right)^2$$

- It often is an engineering task to find the “right” cost function

3. Learn (=optimize)

- Finding h^* becomes an optimization problem

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$$

- Solve optimization problem with gradient descent
  - Select an initial $\boldsymbol{\theta}^0$
  - Repeatedly perform until convergence

$$\boldsymbol{\theta}^{j+1} = \boldsymbol{\theta}^j - \alpha \nabla J(\boldsymbol{\theta}^j)$$

  - Return $\boldsymbol{\theta}^*$ to define h^*

Gradient descent

board

3. Learn (cont'd)

- To perform the update, the gradient $\nabla J(\boldsymbol{\theta})$ is required
- Does gradient descend converge?
- Is solving least-squares regression with gradient descent a good idea?

3. Learn (cont'd)

- To perform the update, the gradient $\nabla J(\boldsymbol{\theta})$ is required
- Does gradient descend converge?
- Is solving least-squares regression with gradient descent a good idea?

Normal equations

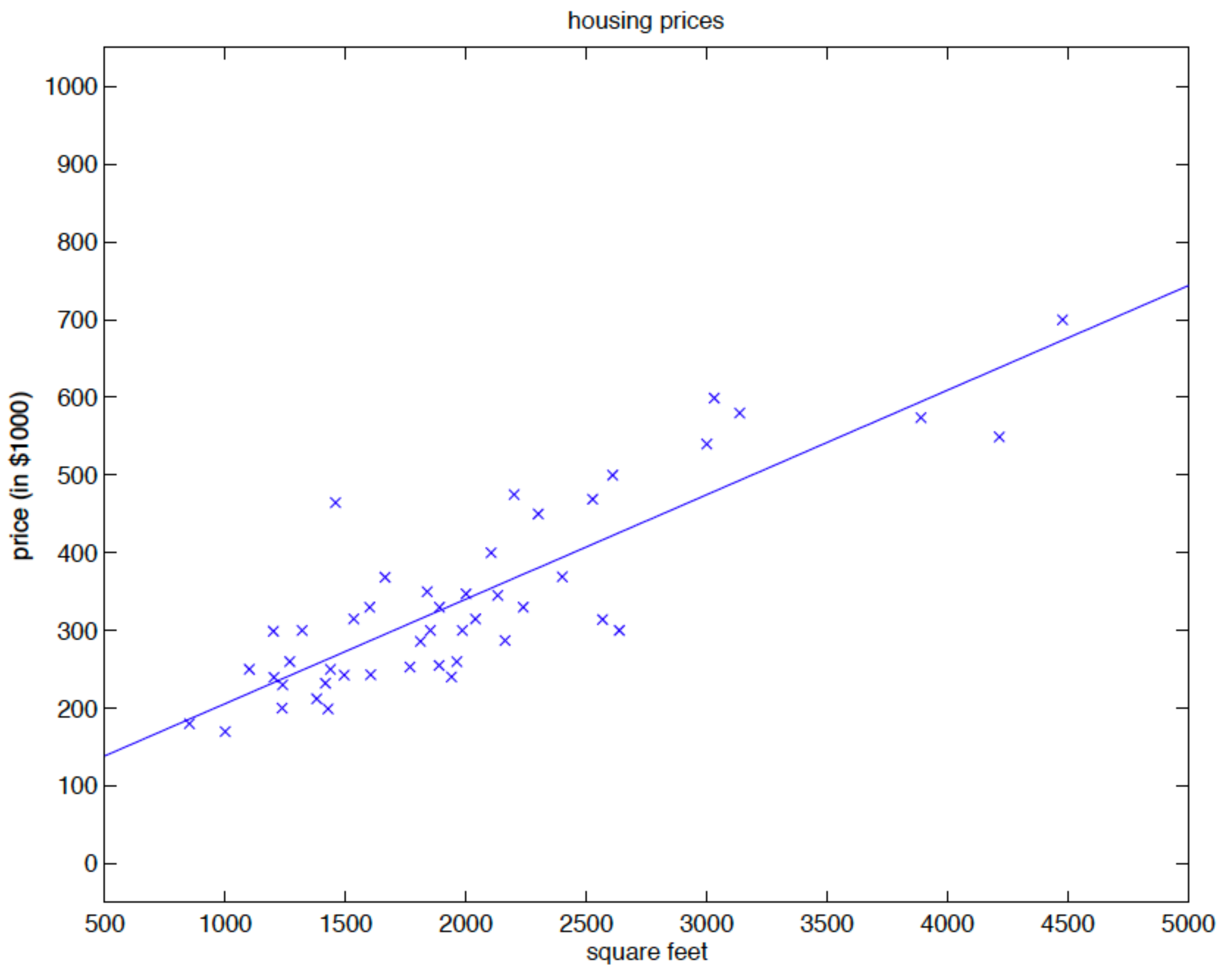
$$\boldsymbol{X} = \begin{bmatrix} -(\boldsymbol{x}^{(1)})^T- \\ \vdots \\ -(\boldsymbol{x}^{(N)})^T- \end{bmatrix}, \qquad \boldsymbol{Y} = \begin{bmatrix} \boldsymbol{y}^{(1)} \\ \vdots \\ \boldsymbol{y}^{(N)} \end{bmatrix}$$

$$\boldsymbol{\theta}^* = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{Y}$$

- Exploit structure of problem at hand, rather than apply black-box solver

Example

Living area (feet^2)	Price (1000$s)
2104	400
1600	330
2400	369
1416	232
3000	540
⋮	⋮



Predict price of houses as function of size of living area

[Andrew Ng]

Recap of probability theory and statistics