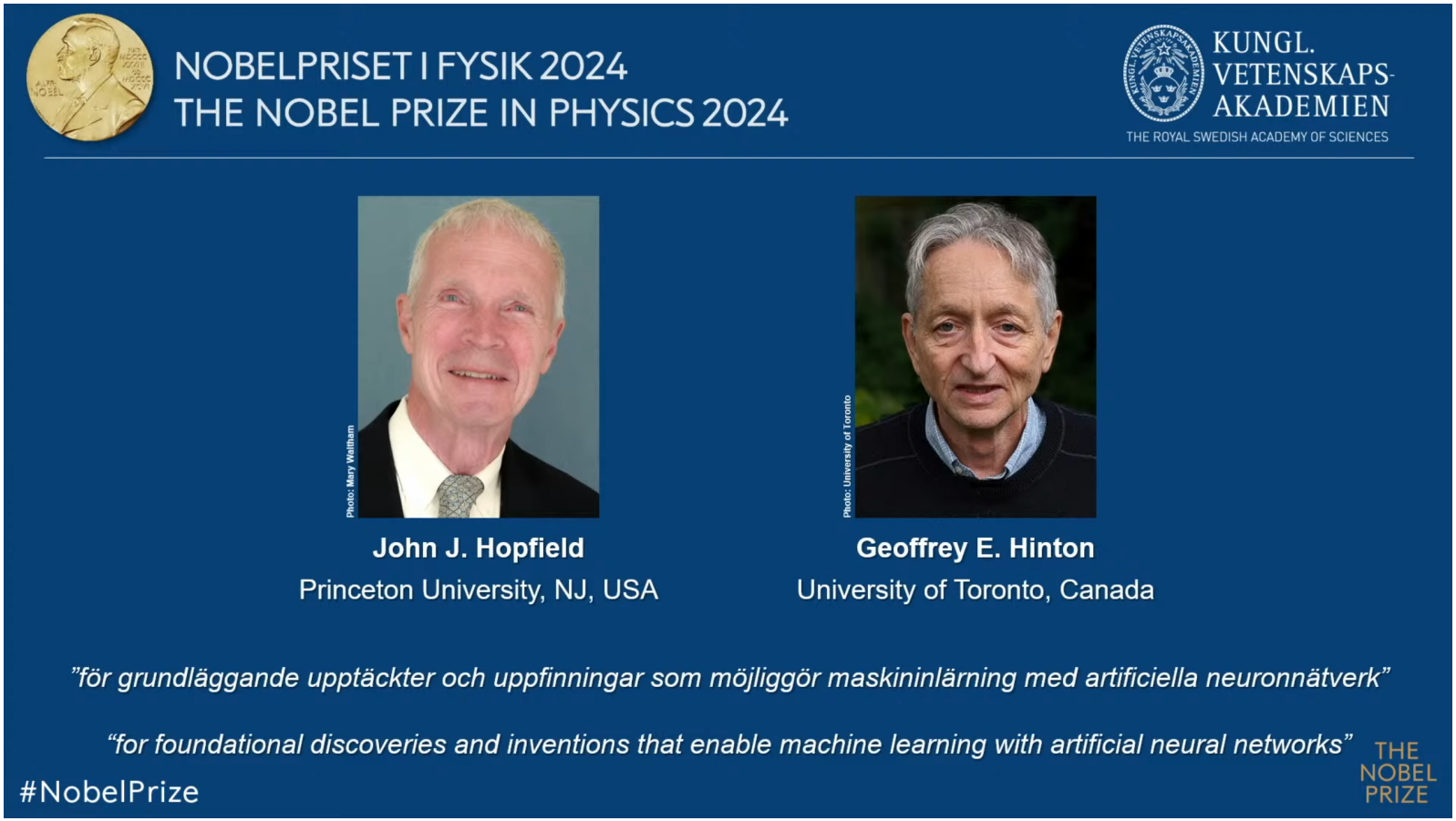
CS5489
Lecture 5.1: Regression

Kede Ma

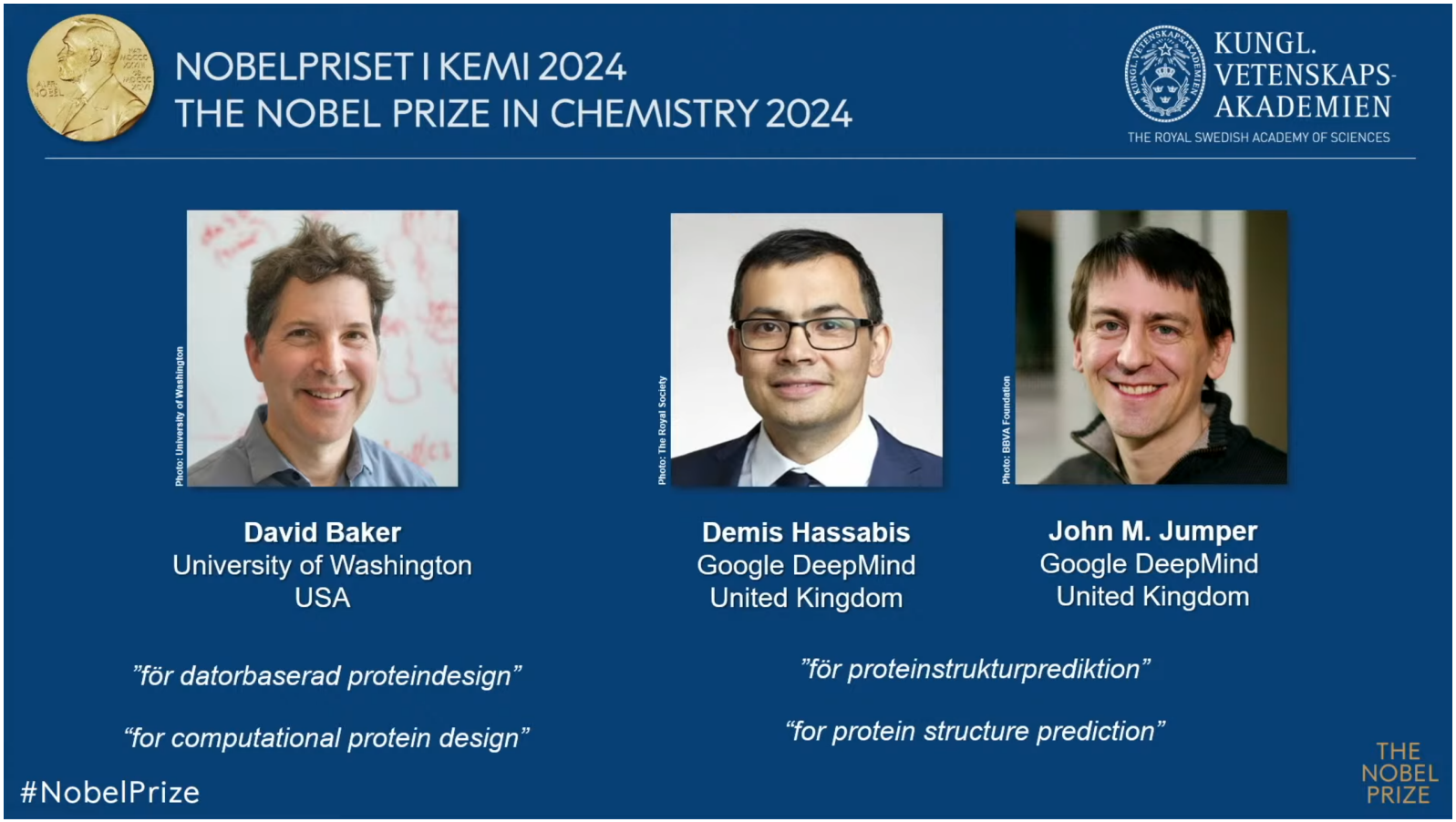
City University of Hong Kong (Dongguan)

Slide template by courtesy of Benjamin M. Marlin

The Nobel Prize in Physics Awarded to ML



The Nobel Prize in Chemistry Awarded to ML

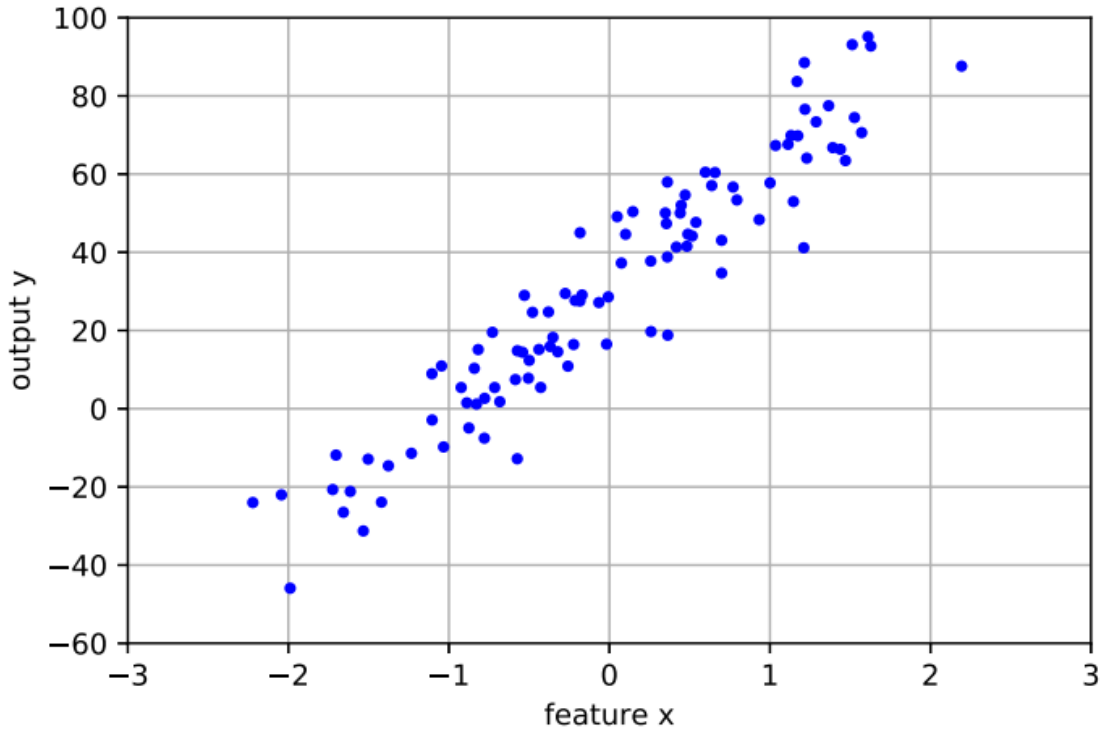


Outline

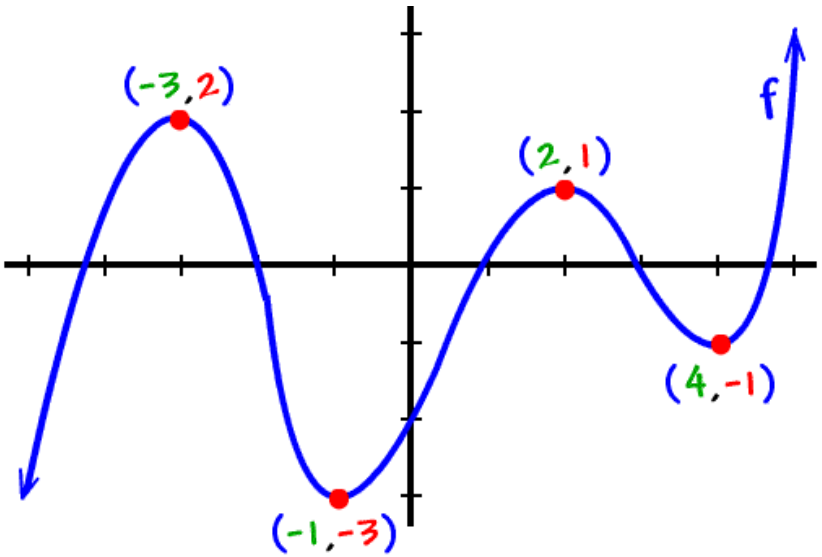
The Regression Task

Definition: The Regression Task

Given a feature vector $\mathbf{x} \in \mathbb{R}^N$, predict its corresponding output value $y \in \mathbb{R}$



Example: Curve Fitting



Given: M points sampled from some underlying curve (assuming no measurement noise)
Goal: Find that curve

- **Question 1**: Would a small handful of points work well, or the more the better?
- **Question 2**: Given the number of points fixed, do their locations matter?
- **Question 3**: Given a set of M points, is curve fitting unique?
- **Question 4**: Why you might be able to choose one "seemingly best" curve, from other possibilities?

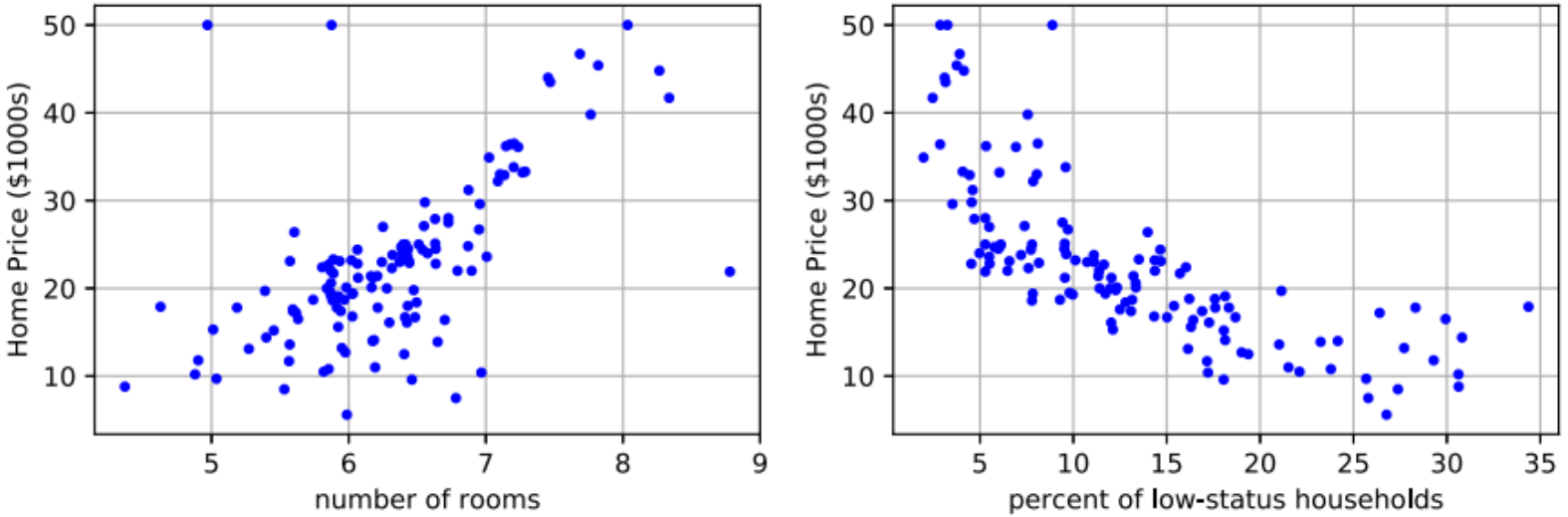
Example: Curve Fitting

- **Question 1**: Would a small handful of points work well, or the more the better?
- **Question 2**: Given the number of points fixed, do their locations matter?
- **Question 3**: Given a set of M points, is curve fitting unique?
- **Question 4**: Why you might be able to choose one "seemingly best" curve, from those infinite possibilities?

- We always like more samples, if possible
- Samples need be representative or informative
- Estimating continuous model from discrete data is an ill-posed problem and in most cases cannot be certain
- You need to have some criteria to choose your preferred model

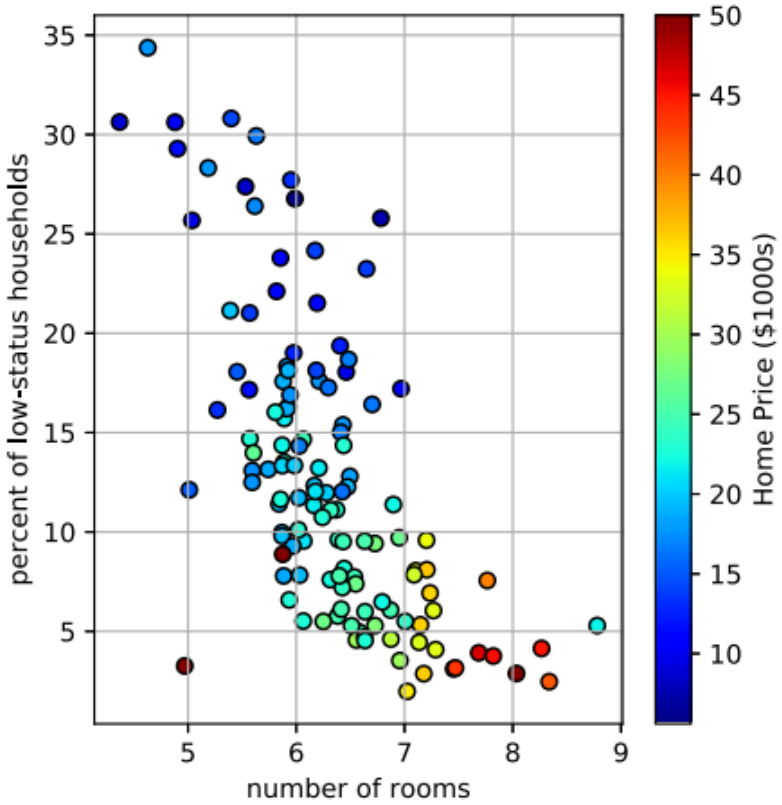
Example: House Price

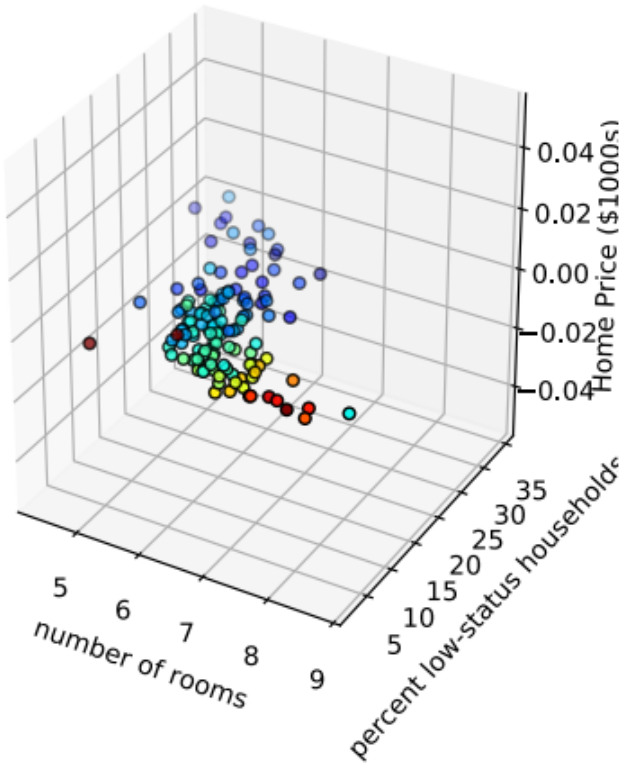
- Predict Boston house price from number of rooms, or percentage of low-status households in neighborhood



Example: House Price

- Predict from both features





The Regression Learning Problem

Definition: Regression Learning Problem

Given a data set of example pairs $\mathcal{D} = \{(\mathbf{x}^{(i)}, y^{(i)}), i = 1, \ldots, M\}$ where $\mathbf{x}^{(i)} \in \mathbb{R}^N$ is a feature vector and $y^{(i)} \in \mathbb{R}$ is the output, learn a function $f : \mathbb{R}^N \mapsto \mathbb{R}$ that accurately predicts y for any feature vector $\mathbf{x}$

Error Measure: Mean Squared Error (MSE)

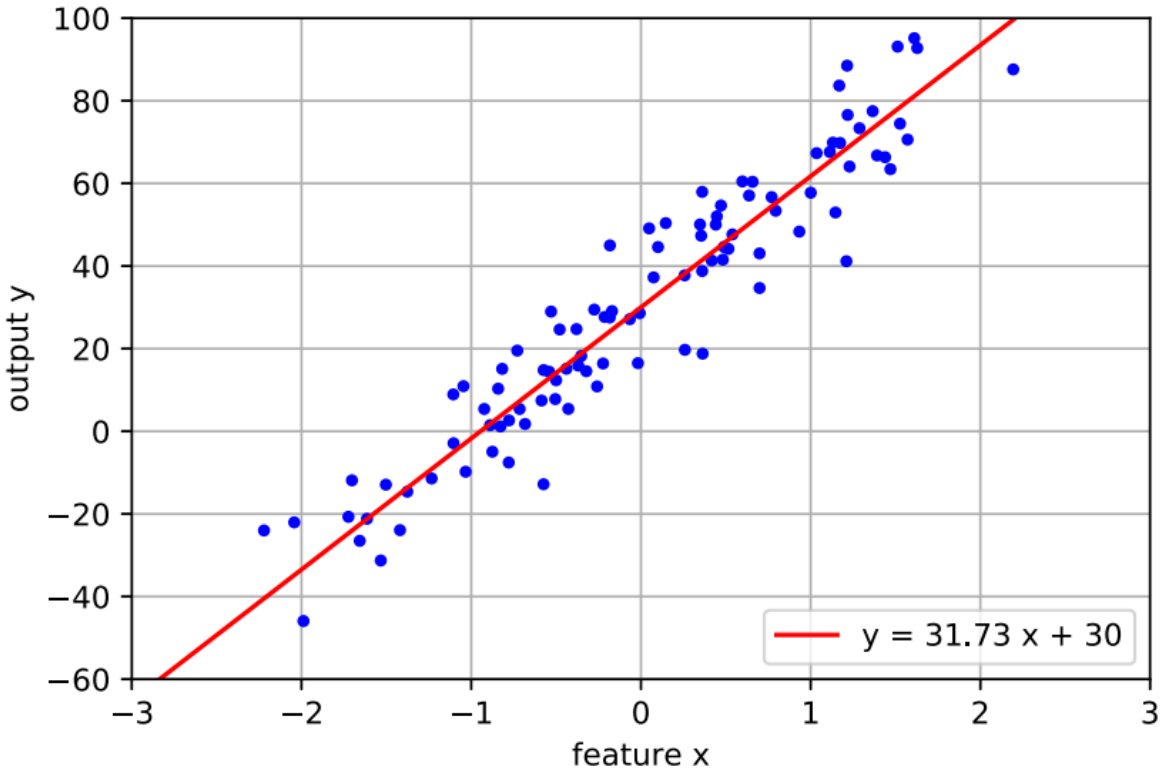
Given a data set of example pairs $\mathcal{D} = \{(\mathbf{x}^{(i)}, y^{(i)}), i = 1, \cdots, M\}$ and a function $f : \mathbb{R}^N \mapsto \mathbb{R}$, the MSE of f on $\mathcal{D}$ is

$$\text{MSE}(\mathcal{D}, f) = \frac{1}{M} \sum_{i=1}^{M} (y^{(i)} - f(\mathbf{x}^{(i)}))^2$$

Outline

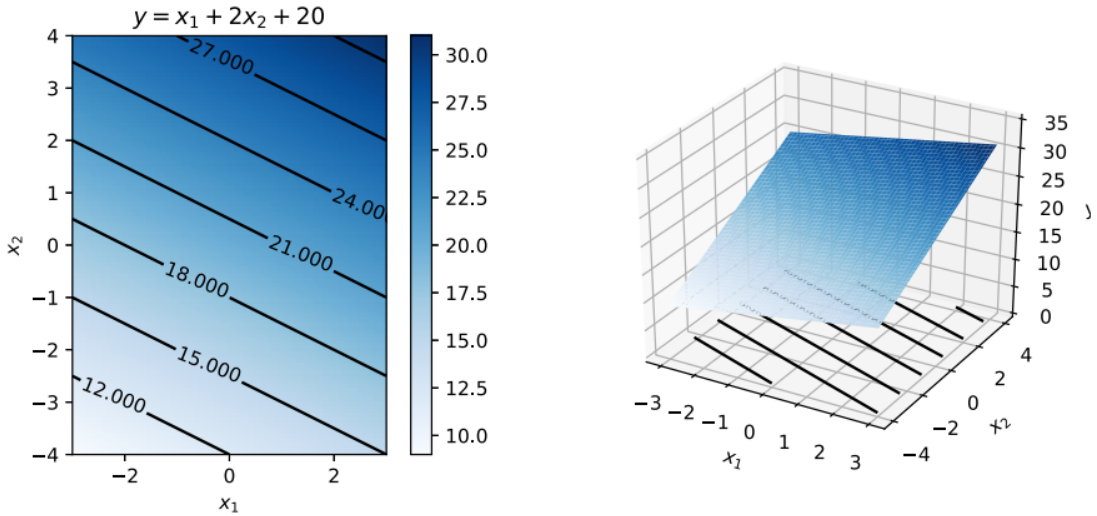
Linear Regression

- **1-D case**: the output y is a linear function of input feature x
  - $y = w \cdot x + b$
  - w is the slope, b is the intercept



Linear Regression

- **N-D case**: the output y is a linear combination of N input features $x_1, \ldots, x_N$:
  - $y = w_1 x_1 + w_2 x_2 + \cdots + w_N x_N + w_0$
- Equivalently,
  - $y = \mathbf{w}^T \mathbf{x} + w_0 = \sum_{j=1}^{N} w_j x_j + w_0 = \sum_{j=0}^{N} w_j x_j$, defining $x_0 = 1$
  - $\mathbf{x} \in \mathbb{R}^N$ is the vector of input values
  - $\mathbf{w} \in \mathbb{R}^N$ are the weights of the linear function and w_0 is the intercept (bias term)



Ordinary Least Squares (OLS)

- The linear function has form $f(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b$
- How to estimate the parameters $(\mathbf{w}, b)$ from the data?
- OLS selects the linear regression parameters to minimize the MSE on the training data set

$$\mathbf{w}^\star, b^\star = \arg\min_{\mathbf{w},b} \frac{1}{M}\sum_{i=1}^{M}(y^{(i)} - \mathbf{w}^T\mathbf{x}^{(i)} - b)^2$$

Solving OLS For One Feature

- Take the derivative of the objective w.r.t. w and set it to zero:

$$\frac{\partial}{\partial w}\frac{1}{M}\sum_{i=1}^{M}(y^{(i)} - wx^{(i)} - b)^2 = 0$$

$$2\frac{1}{M}\sum_{i=1}^{M}(y^{(i)} - wx^{(i)} - b)(-x^{(i)}) = 0$$

$$\left(\sum_{i=1}^{M}(x^{(i)})^2\right)w + \left(\sum_{i=1}^{M}x^{(i)}\right)b = \sum_{i=1}^{M}y^{(i)}x^{(i)}$$

Solving OLS For One Feature

- Take the derivative of the objective w.r.t. b and set it to zero:

$$\frac{\partial}{\partial b}\frac{1}{M}\sum_{i=1}^{M}(y^{(i)} - wx^{(i)} - b)^2 = 0$$

$$2\frac{1}{M}\sum_{i=1}^{M}(y^{(i)} - wx^{(i)} - b)(-1) = 0$$

$$\left(\sum_{i=1}^{M} x^{(i)}\right) w + Mb = \sum_{i=1}^{M} y^{(i)}$$

Solving OLS For One Feature

- Write the two equations in matrix form:

$$\begin{bmatrix} \sum_{i=1}^{M} (x^{(i)})^2 & \sum_{i=1}^{M} x^{(i)} \\ \sum_{i=1}^{M} x^{(i)} & M \end{bmatrix} \begin{bmatrix} w \\ b \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{M} y^{(i)} x^{(i)} \\ \sum_{i=1}^{M} y^{(i)} \end{bmatrix}$$

$$\begin{bmatrix} w \\ b \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{M} (x^{(i)})^2 & \sum_{i=1}^{M} x^{(i)} \\ \sum_{i=1}^{M} x^{(i)} & M \end{bmatrix}^{-1} \begin{bmatrix} \sum_{i=1}^{M} y^{(i)} x^{(i)} \\ \sum_{i=1}^{M} y^{(i)} \end{bmatrix}$$

General OLS Solution

- **Matrix notation**:
  - $\mathbf{x}^{(i)} \in \mathbb{R}^{N+1}$, where we have defined $x_0^{(i)} = 1$
  - $\mathbf{X} \in \mathbb{R}^{M \times (N+1)}$ with one data case $\mathbf{x}^{(i)} \in \mathbb{R}^{N+1}$ per row

$$\mathbf{X} = \begin{bmatrix} — & (\mathbf{x}^{(1)})^T & — \\ — & (\mathbf{x}^{(2)})^T & — \\ & \vdots & \\ — & (\mathbf{x}^{(M)})^T & — \end{bmatrix}$$

  - $\mathbf{y} \in \mathbb{R}^M$ is a column vector containing the corresponding outputs

$$\mathbf{y} = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(M)} \end{bmatrix}$$

  - $\mathbf{w} \in \mathbb{R}^{N+1}$, where $w_0 = b$

General OLS Solution

- The general OLS solution for $\mathbf{w} \in \mathbb{R}^{N+1}$ is

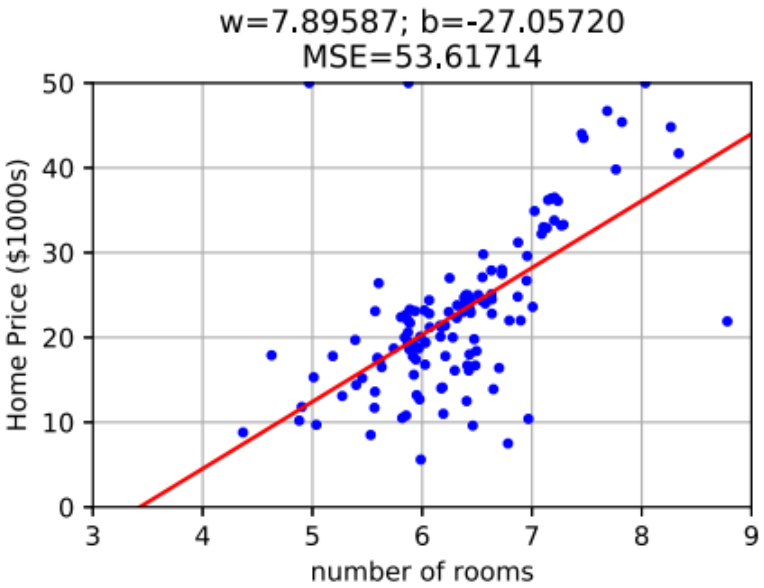
$$\begin{aligned}\mathbf{w}^\star &= \arg\min_{\mathbf{w}} \frac{1}{M} \sum_{i=1}^{M} (y^{(i)} - (\mathbf{x}^{(i)})^T \mathbf{w})^2 \\ &= \arg\min_{\mathbf{w}} \frac{1}{M} (\mathbf{y} - \mathbf{X}\mathbf{w})^T (\mathbf{y} - \mathbf{X}\mathbf{w})\end{aligned}$$

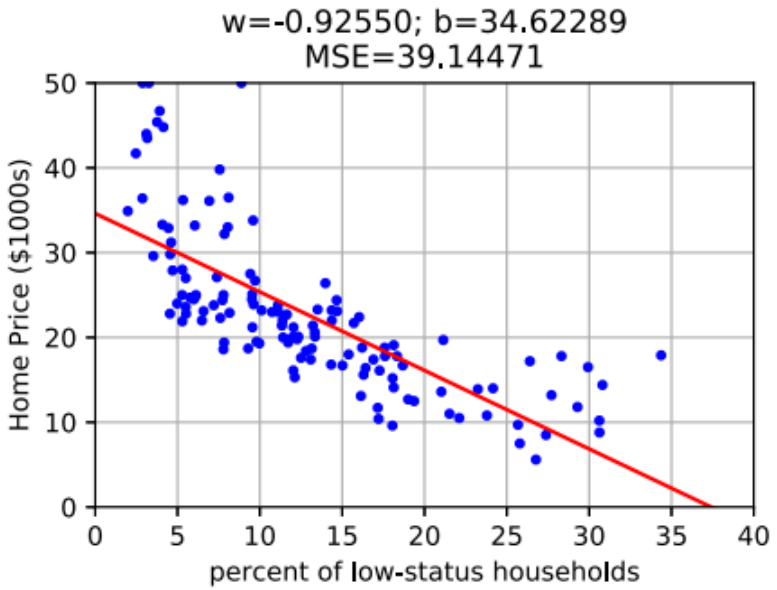
- Take the derivative w.r.t. $\mathbf{w}$ and set it to zero:

$$\begin{aligned}\frac{\partial}{\partial \mathbf{w}} \frac{1}{M} (\mathbf{y} - \mathbf{X}\mathbf{w})^T (\mathbf{y} - \mathbf{X}\mathbf{w}) &= 0 \\ \mathbf{X}^T (\mathbf{y} - \mathbf{X}\mathbf{w}) &= 0 \\ \mathbf{X}^T \mathbf{X}\mathbf{w} &= \mathbf{X}^T \mathbf{y} \\ \mathbf{w}^\star &= (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}\end{aligned}$$

Boston Housing price (1-D)

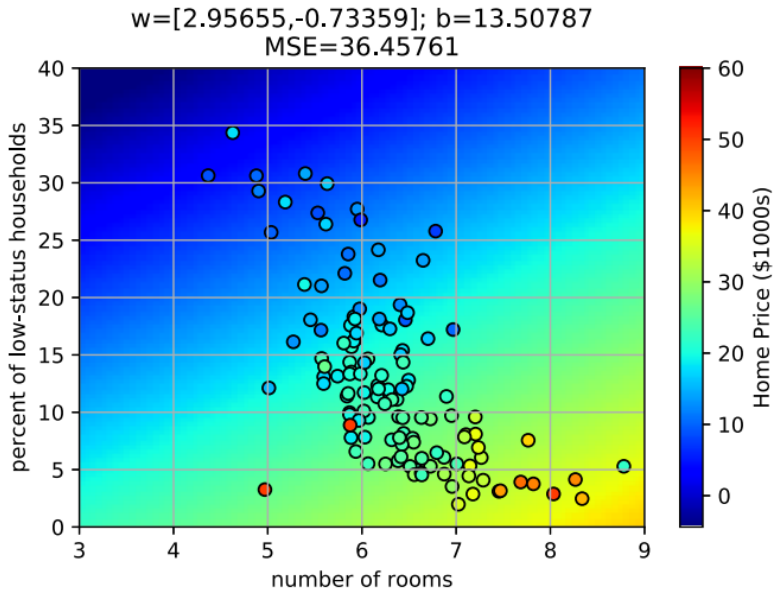
- Learn regression function for each feature separately





Boston Housing price (2-D)

- For both features together



- Interpretation from the linear model parameters
  - Each room increases home price by $\$2,956$ (w_1)
  - Each percentage of low-status households decreases home price by $\$733$ (w_2)
  - The "starting" price is $\$13,508$ (b)

Connection to Probabilistic Models

- The same solution can be derived as the MLE of the parameters under a conditional Gaussian model
- Assumption: the output (or the observation) $\mathbf{y}$ is from a deterministic function with additive Gaussian noise:

$$\mathbf{y} = \mathbf{X}\mathbf{w} + \boldsymbol{\epsilon}, \text{ where } p(\boldsymbol{\epsilon}; \sigma^2) = \mathcal{N}(\boldsymbol{\epsilon}; 0, \sigma^2\mathbf{I})$$

- This implies that

$$p(\mathbf{y}|\mathbf{X}; \mathbf{w}, \sigma^2) = p(\mathbf{y}-\mathbf{X}\mathbf{w}; \sigma^2) = \mathcal{N}(\mathbf{y}-\mathbf{X}\mathbf{w}; 0, \sigma^2\mathbf{I}) = \mathcal{N}(\mathbf{y}; \mathbf{X}\mathbf{w}, \sigma^2\mathbf{I})$$

- Equivalently,

$$p(\mathbf{y}|\mathbf{X}; \mathbf{w}, \sigma^2) = \frac{1}{\sqrt{(2\pi\sigma^2)^M}} \exp\left(-\frac{1}{2\sigma^2}(\mathbf{y} - \mathbf{X}\mathbf{w})^T(\mathbf{y} - \mathbf{X}\mathbf{w})\right)$$

Connection to Probabilistic Models

- Maximize the log likelihood:

$$\begin{aligned}
&\arg\max_{\mathbf{w},\sigma^2} \log p(\mathbf{y}|\mathbf{X};\mathbf{w},\sigma^2) \\
&= \arg\max_{\mathbf{w},\sigma^2} -\frac{M}{2}\log 2\pi - \frac{M}{2}\log\sigma^2 - \frac{1}{2\sigma^2}(\mathbf{y}-\mathbf{X}\mathbf{w})^T(\mathbf{y}-\mathbf{X}\mathbf{w}) \\
&= \arg\min_{\mathbf{w},\sigma^2} \frac{M}{2}\log\sigma^2 + \frac{1}{2\sigma^2}(\mathbf{y}-\mathbf{X}\mathbf{w})^T(\mathbf{y}-\mathbf{X}\mathbf{w})
\end{aligned}$$

- Note that solving for $\mathbf{w}$ is independent of σ^2:

$$\mathbf{w}^\star = \arg\min_{\mathbf{w}} \frac{1}{2}(\mathbf{y}-\mathbf{X}\mathbf{w})^T(\mathbf{y}-\mathbf{X}\mathbf{w})$$

More on Linear Regression

- Take a closer look at the closed-form solution

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

  - We call $\mathbf{X}^\dagger = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T$ pseudo-inverse (generalization of inverse to non-square matrix). See for a square invertible matrix $\mathbf{X}$, we have $\mathbf{X}^\dagger = \mathbf{X}^{-1}(\mathbf{X}^T)^{-1}\mathbf{X}^T = \mathbf{X}^{-1}$
  - What if $\mathbf{X}^T\mathbf{X} \in \mathbb{R}^{(N+1)\times(N+1)}$ is non-invertible?
    - If $M < N+1$, add more data or enforce regularization
    - If $M \geq N+1$, remove redundant features or enforce regularization
  - What if N is too large? Use gradient descent

$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \alpha\mathbf{X}^T(\mathbf{X}\mathbf{w}^{(t)} - \mathbf{y})$$

  - What if M is too large? Use mini-batch gradient descent

$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \alpha \sum_{(\mathbf{x},y)\in\mathcal{B}} \mathbf{x}(\mathbf{x}^T\mathbf{w}^{(t)} - y)$$