CS 4824/ECE 4424: Linear Regression

Acknowledgement:
Many of these slides are derived from Tom Mitchell, Pascal Poupart, Pieter Abbeel, Eric Eaton, Carlos Guestrin, William Cohen, and Andrew Moore.

Regression

- So far, we've been interested in learning $P(Y|X)$ where Y has discrete values (called 'classification')
- What if Y is continuous? (called 'regression')
  - predict snow/rainfall from current and past weather features
  - predict stock price from current and past market conditions
  - predict weight from gender, height, age, ...

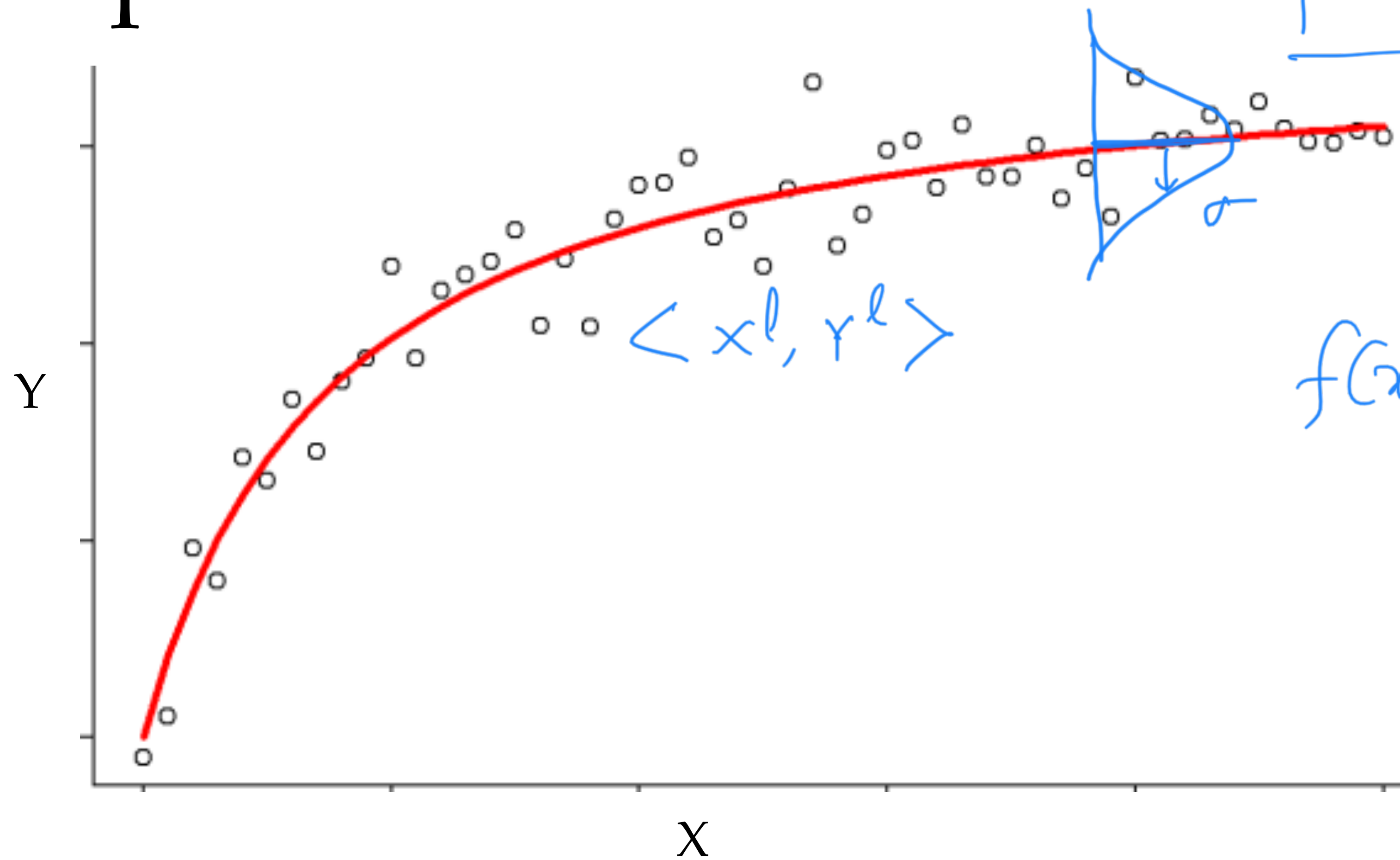
Regression: problem setting

approximate

- Wish to learn $\boxed{f: X \rightarrow Y}$ where Y is real-valued, given training data $\{\langle X^1, Y^1 \rangle \ldots \langle X^n, Y^n \rangle\}$

- Approach:
  - Choose some parameterized form for $\boxed{P(Y \mid X, \theta)}$ where θ is the vector of parameters
  - Estimate θ using MLE or MAP estimation

Choose parameterized form for $\boxed{P(Y \mid X, \theta)}$



$\langle x^l, y^l \rangle$

σ

$f(x) = x^2 + 3x + 5$

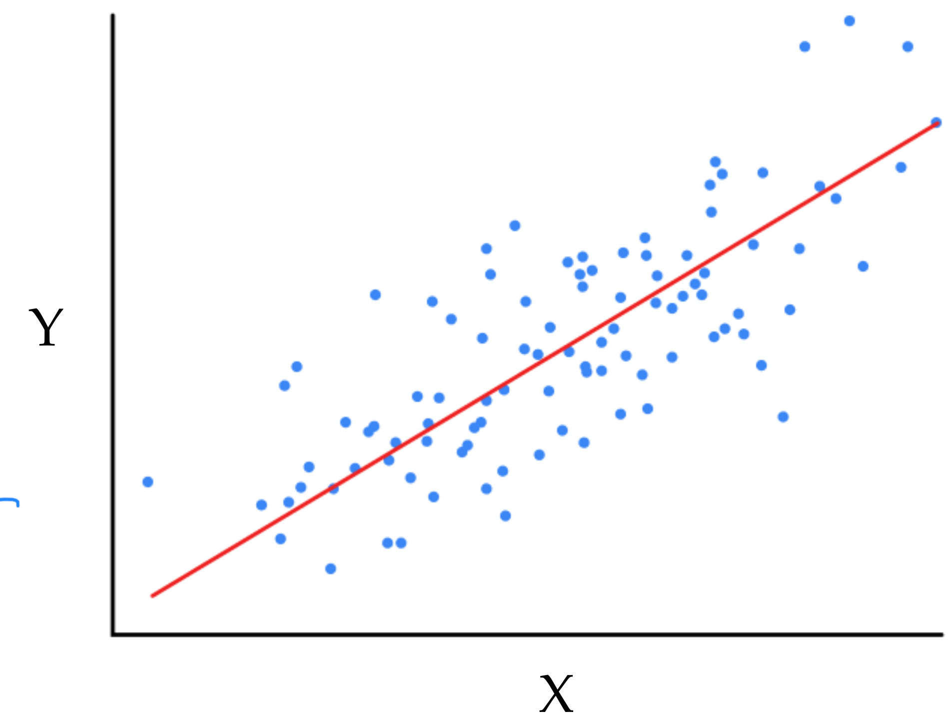
- Assume Y is some deterministic f(X), plus random noise
  - $\boxed{y = f(x) + \varepsilon}$ $\boxed{\text{where } \varepsilon \sim \mathcal{N}(0,\sigma)}$
- Therefore, Y is a random variable that follows the distribution
  - $\boxed{p(y \mid x)} = \boxed{\mathcal{N}(f(x), \sigma)}$
- And the expected value of y for any given x is f(x)

Consider linear regression

- $p(y \mid x) = \mathcal{N}(f(x), \sigma)$

  $f(x) = w_0 + w_1 x$

- Assume $\boxed{\text{f(x) is a linear function of } x}$
  i.e.,



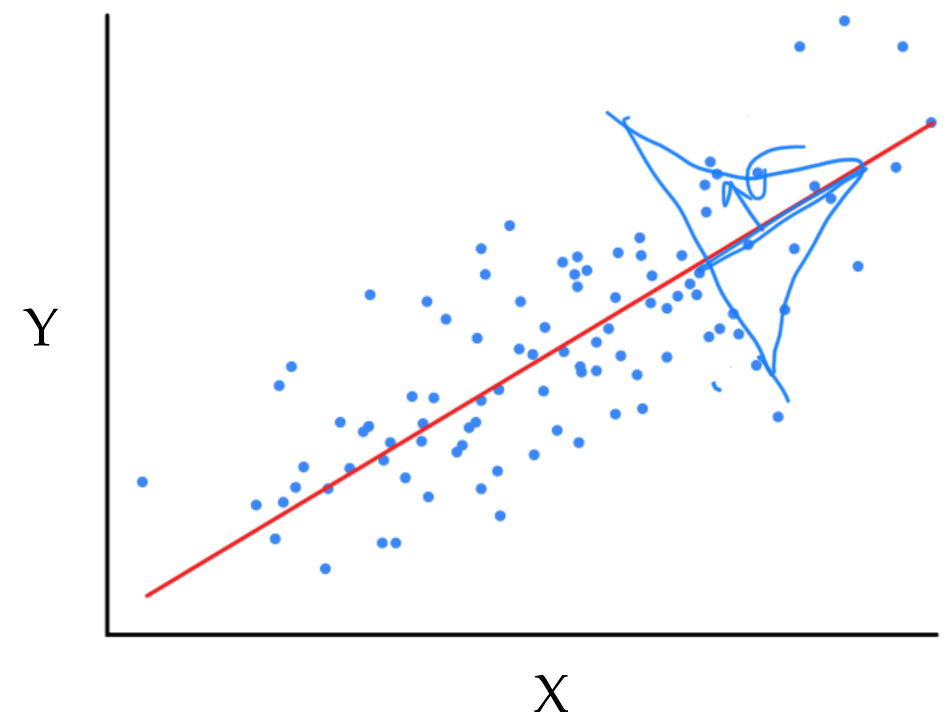
  - $p(y \mid x) = \mathcal{N}(w_0 + w_1 x, \sigma)$
  - $\mathbb{E}(y \mid x) = w_0 + w_1 x$

Consider linear regression

- $p(y|x) = \mathcal{N}(f(x), \sigma)$

- Assume f(x) is a linear function of x

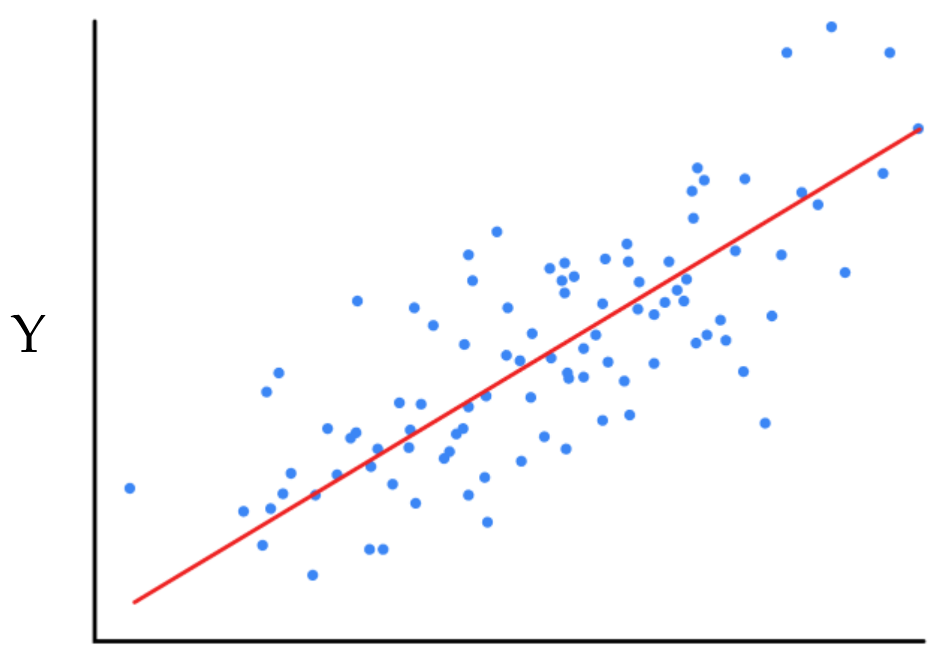
  i.e., $y = w_0 + w_1 x$
  - $p(y|x) = \mathcal{N}(w_0 + w_1 x, \sigma)$
  - $\mathbb{E}(y|x) = w_0 + w_1 x$



- Note: to make our parameters explicit, let's write
  - $W = < w_0, w_1 >$
  - $\boxed{p(y|x, W) = \mathcal{N}(w_0 + w_1 x, \sigma)}$

Training linear regression

- $p(y \mid x) = \mathcal{N}(f(x), \sigma)$



- How can we learn $\boxed{\text{W}}$ from data?

- Learn W using Maximum Conditional Likelihood Estimation!
  - $$W_{MCLE} = \arg\max_{W} \prod_{l} \boxed{P(Y^l \mid X^l, W)}$$
  - $$W_{MCLE} = \arg\max_{W} \sum_{l} \underline{\ln p(y^l \mid x^l, W)}$$
  - Where $\boxed{P(y \mid x, W)} = \dfrac{1}{\sigma\sqrt{2\pi}} e^{\left(-\frac{1}{2}\left(\frac{y^l - f(x, W)}{\sigma}\right)^2\right)}$

MCLE derivation

$f(x) = w_0 + w_1 x$

- $$\boxed{W_{MCLE} = \arg\max_W \sum_l \ln p(y^l \mid x^l, W)}$$
- $$P(y \mid x, W) = \frac{1}{\sigma\sqrt{2\pi}} e^{\left(-\frac{1}{2}\left(\frac{y^l - f(x,W)}{\sigma}\right)^2\right)} \quad \text{or} \quad \boxed{P(y \mid x, W) = \frac{1}{\sigma\sqrt{2\pi}} e^{\left(-\frac{1}{2}\left(\frac{y^l - (w_0 + w_1 x^l)}{\sigma}\right)^2\right)}}$$

$$W_{MCLE} = \arg\max_W \sum_l \boxed{\ln \frac{1}{\sigma\sqrt{2\pi}}} + \left(-\frac{1}{2}\left(\frac{y^l - (w_0 + w_1 x^l)}{\sigma}\right)^2\right)$$

maximization

$$= \arg\max_W \sum_l \boxed{-}\frac{1}{2\sigma^2}\left(y^l - (w_0 + w_1 x)\right)^2$$

minimization

$$= \arg\min_W \sum_l \boxed{\frac{1}{2\sigma^2}}\left(y^l - (w_0 + w_1 x)\right)^2$$

$$= \arg\min_W \sum_l \left(y^l - (w_0 + w_1 x)\right)^2$$

Training linear regression

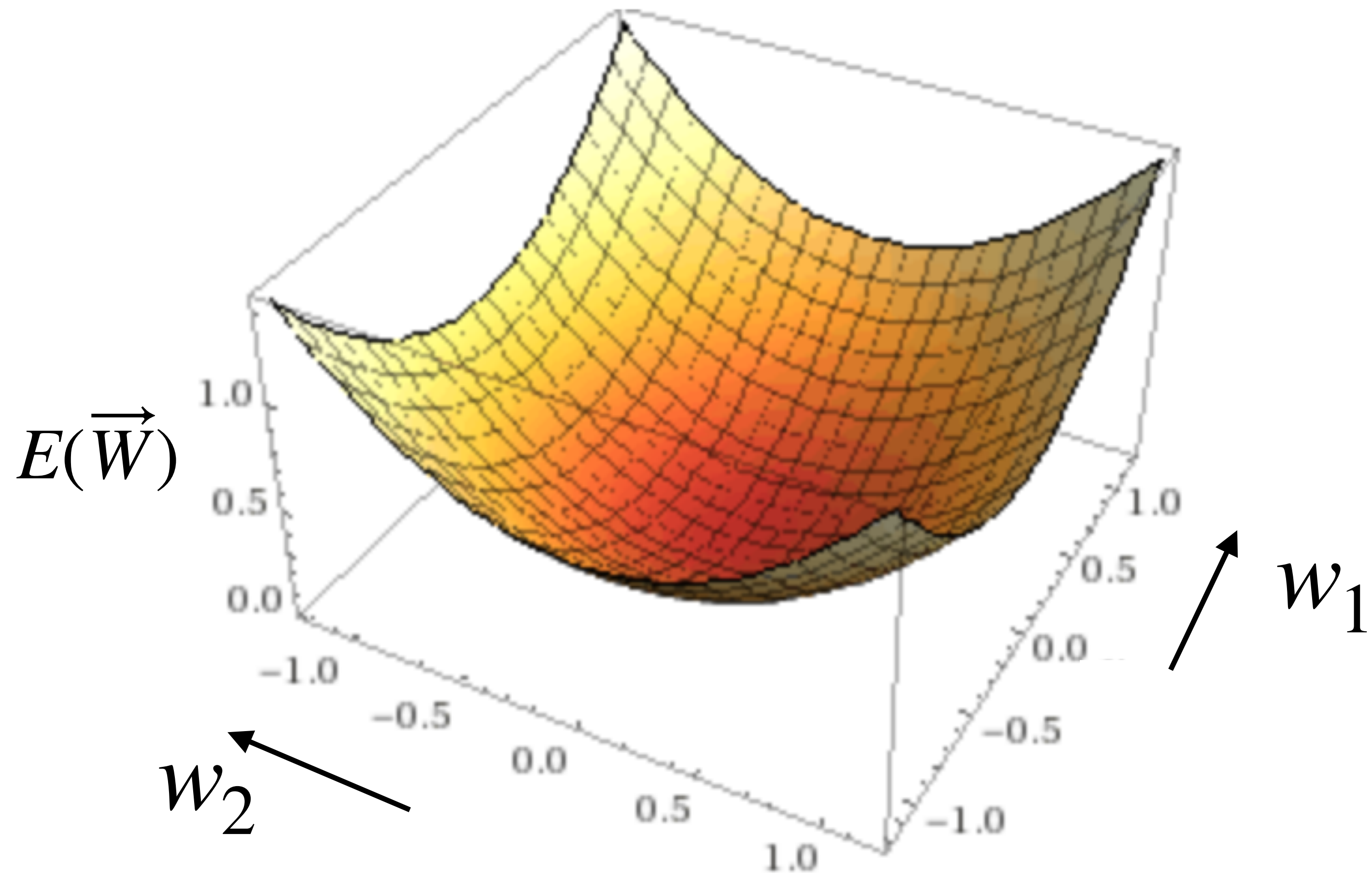
- Learn W using Maximum Conditional Likelihood Estimation:

$$\boxed{W_{MCLE} = \arg\min_{W} \sum_{l} (y - f(x, W))^2}$$

- This corresponds to minimizing sum of squared errors (often used in "curve fitting")

- How to perform the minimization in order to choose optimal $W = < w_0, w_1 > $?

Gradient descent



- **Gradient** $\nabla E(\vec{w}) = [\frac{\partial E}{\partial w_0}, \frac{\partial E}{\partial w_1}, \ldots, \frac{\partial E}{\partial w_n}]$

- Training Rule: $\vec{w}^{(i+1)} \leftarrow \vec{w}^{i} \boxed{-}\, \eta \boxed{\nabla E(\vec{w})}$

- i.e. $\Delta w_i = -\eta \frac{\partial E}{\partial w_i}$

© Debswapna Bhattacharya

Minimizing squared error: gradient descent

- $$\frac{\partial E}{\partial w_1} = \sum_l 2(y^l - (w_o + w_1 x^l))(-x^l)$$
- $$= \boxed{-2} \sum_l (y^l - (w_o + w_1 x^l))(x^l)$$

- $$\frac{\partial E}{\partial w_0} = -2 \sum_l (y^l - (w_o + w_1 x^l))$$

Minimizing squared error: gradient descent

- $$\frac{\partial E}{\partial w_1} = -2\sum_l (y^l - (w_o + w_1 x^l))(x^l) \qquad \frac{\partial E}{\partial w_0} = -2\sum_l (y^l - (w_o + w_1 x^l))$$

- Update rule:

$$w_1 \leftarrow w_1 - \eta\left[-2\sum_l x^l\left(y^l - (w_0 + w_1 x^l)\right)\right]$$

$$w_1 \leftarrow w_1 + 2\eta\sum_l x^l\left(y^l - (w_0 + w_1 x^l)\right)$$

$$w_0 \leftarrow w_0 + 2\eta\sum_l\left(y^l - (w_0 + w_1 x^l)\right)$$

Linear regression more generally

- $p(y|x) = \mathcal{N}(f(x), \sigma I)$ $\quad \overrightarrow{X} = < x_1, x_2, \ldots, x_n >$
- $f(x) = w_0 + \sum_{i=1}^{n} w_i x_i$
- $p(y|x) = \mathcal{N}(w_0 + \sum_{i=1}^{n} w_i x_i, \sigma I)$
- $\mathbb{E}(y|x) = w_0 + \sum_{i=1}^{n} w_i x_i$
- $p(y|x, W) = \mathcal{N}(w_0 + \sum_{i=1}^{n} w_i x_i, \sigma I)$ $\quad \overrightarrow{W} = < w_0, w_1, \ldots, w_n >$

Minimizing squared error more generally: gradient descent

$$f(x) = w_0 + \sum_{j=1}^{n} w_j x_j$$

- **Gradient descent algorithm**: iterate until change < ε
  - $\forall\, i$ repeat $$w_i \leftarrow w_i + 2\eta \sum_l X_i^l \Big(Y^l - \underbrace{\boxed{\Big(w_0 + \sum_{j=1}^{n} w_j x_j\Big)}}_{}\Big)$$
    $f(x)$
  - assume $X_0 = 1$ for w_0

How about MAP estimation?

- $$W_{MAP} = \arg\max_W \left(\boxed{-c \sum_l w_i^2} \right) + \sum_l \ln P(Y^l \mid X^l, W)$$

- Called a “**regularization**” term
- Helps reduce overfitting, especially for sparse data situations
- Keeps weights near zero with prior W ~ $\mathcal{N}(0,\sigma I)$, or whatever the prior suggests

© Debswapna Bhattacharya

Demo Time 🙂

https://lukaszkujawa.github.io/gradient-descent.html

Summary of regression

- Assuming $p(y \,|\, x, W) = \mathcal{N}(w_0 + w_1 x, \sigma)$
  - MLE corresponds to minimizing sum of squared errors
  - MAP estimate minimizes SSE plus sum of squared weights
  - Again, learning is an optimization problem once we choose our objective function
    - maximize data likelihood
    - maximize posterior prob of W
- Again, we can use gradient descent as a general learning algorithm
- Be careful about outliers while performing regression
- **NOTE**: Almost nothing here required that f(x) be linear in x