CS 446/ECE 449: Machine Learning

Shenlong Wang

University of Illinois at Urbana-Champaign, 2024

Logistics:

- **Signup:** Campuswire and Gradescope. (https://campuswire.com/p/G47CE41F1, Code: 0662)
- **Tutorial:** Amnon will host a Probability / Numpy / PyTorch tutorial soon. Stay tuned for more information.
- **HW1 Update:** We will release it on Wednesday (January 24), and the due date is February 6.
- **Slides:** Slides will be uploaded before the class begins. Handwritten notes (if any) will be uploaded after the class ends.

L3: Linear Regression

Goals of this lecture

- Math Intro
- Getting to know linear regression
- Understanding how linear regression works
- Examples for linear regression

Reading Material

- K. Murphy; Machine Learning: A Probabilistic Perspective; Chapter 7

Math Intro:

- Vector: $\boldsymbol{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} \in \mathbb{R}^n$
- Matrix: $\mathbf{X} = \begin{bmatrix} x_{1,1} & \cdots & x_{1,m} \\ \vdots & \ddots & \vdots \\ x_{n,1} & \cdots & x_{n,m} \end{bmatrix} \in \mathbb{R}^{n \times m}$
- Norm: $\|x^{(1)} - x^{(2)}\|_2^2 = \sum_{i=1}^n (x_i^{(1)} - x_i^{(2)})^2$ distance between two points in n dimensions
- Transpose: $\mathbf{X}^T = \begin{bmatrix} x_{1,1} & \cdots & x_{n,1} \\ \vdots & \ddots & \vdots \\ x_{1,m} & \cdots & x_{m,n} \end{bmatrix} \in \mathbb{R}^{m \times n}$

$$\boldsymbol{x}^T = \begin{bmatrix} x_1 & \cdots & x_n \end{bmatrix} \in \mathbb{R}^{1 \times n}$$

- Matrix multiplication: $\mathbf{X}^T\boldsymbol{x}$ or $\mathbf{X}\boldsymbol{x}$?

Discrete Probability: $y \in \{1, \ldots, 6\}$

- Discrete probability distribution: $p(Y = y) \in [0, 1]$ with $\sum_{y \in \{1,\ldots,6\}} p(Y = y) = 1$
- Abbreviation: $p(Y = y) = p(y) \in [0, 1]$
- Expectation: $\mathbb{E}_{p(y)}[f(y)] = \sum_{y \in \{1,\ldots,6\}} p(y)f(y)$

Continuous probability: $y \in \mathbb{R}$

- $p(Y = 1) = 0$
- Probability density function: $p(y) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2\sigma^2}(y-\mu)^2\right)$
- Mean: $\mathbb{E}_{p(y)}[y] = \int_{-\infty}^{\infty} yp(y)dy = \mu$
- Variance: $\mathbb{E}_{p(y)}[(y-\mu)^2] = \sigma^2$

Multivariate continuous probability: $\boldsymbol{y} \in \mathbb{R}^n$ $\mu \in \mathbb{R}^n$

- n-dimensional density:
  $p(\boldsymbol{y}) = p(y_1, \ldots, y_n) = \frac{1}{\sqrt{(2\pi)^n|\Sigma|}} \exp\left(-\frac{1}{2}(\boldsymbol{y}-\mu)^T\Sigma^{-1}(\boldsymbol{y}-\mu)\right)$
- Covariance matrix: Σ

Multivariate calculus: $\boldsymbol{x} \in \mathbb{R}^n$, $\boldsymbol{w} \in \mathbb{R}^n$, $\mathbf{A} \in \mathbb{R}^{n \times n}$

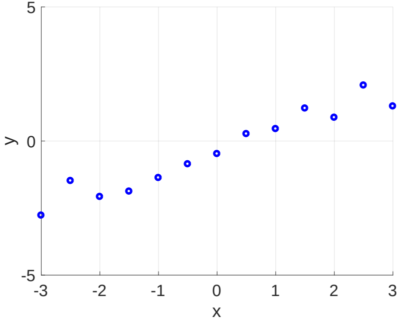
- Multivariate function: $f(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{x}$
- Derivative: $\frac{\partial f}{\partial \boldsymbol{x}} = \boldsymbol{w}$
- Multivariate function: $f(\boldsymbol{x}) = \boldsymbol{x}^T \mathbf{A} \boldsymbol{x}$
- Derivative: $\frac{\partial f}{\partial \boldsymbol{x}} = (\mathbf{A} + \mathbf{A}^T)\boldsymbol{x}$

Recap: What we have learned so far?

- Lecture 1: KNN
- Lecture 2: Naive Bayes
- Parameteric or Non-parametric?
- What are the key underlying assumptions?
- Linear or non-linear decision boundaries?
- Key pros and cons?

Classification problem (output is category)

Regression - The Problem:



Given continuous-valued outcomes $y^{(i)} \in \mathbb{R}$ for covariates $x^{(i)} \in \mathbb{R}$ (e.g. predict housing price based on area's annual family income),

- How do we parametrize the model?
- What loss / objective function should we use to judge the fit?
- How do we optimize fit to unseen test data (generalization)?

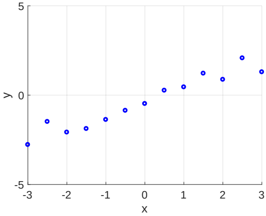
Linear Regression:

Let's assume a linear model with parameters $w_1 \in \mathbb{R}$ and $w_2 \in \mathbb{R}$

$$y = w_1 \cdot x + w_2$$

Given a dataset of N pairs (x, y):

$$\mathcal{D} = \{(x^{(i)}, y^{(i)})\}_{i=1}^{N}$$



How do we find the parameters w_1, w_2?

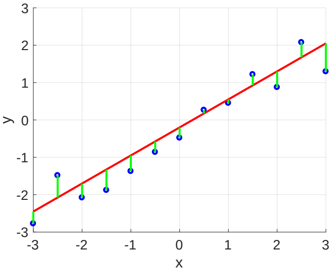
Assuming model

$$y = w_1 \cdot x + w_2$$

Find parameters w_1, w_2 such that the squared error is small

$$\arg\min_{w_1, w_2} \frac{1}{2} \sum_{i=1}^{N} \left(y^{(i)} - w_1 \cdot x^{(i)} - w_2 \right)^2$$

What exactly is the error?



Program:

$$\color{blue}{\arg\min_{w_1,w_2}} \frac{1}{2} \sum_{i=1}^{N} \left(y^{(i)} - w_1 \cdot x^{(i)} - w_2 \right)^2$$

Vector notation:

$$\arg\min_{w_1,w_2} \frac{1}{2} \left\| \underbrace{\begin{bmatrix} y^{(1)} \\ \vdots \\ y^{(N)} \end{bmatrix}}_{\mathbf{Y} \in \mathbb{R}^N} - \underbrace{\begin{bmatrix} x^{(1)} & 1 \\ \vdots & \vdots \\ x^{(N)} & 1 \end{bmatrix}}_{\mathbf{X}^\top \in \mathbb{R}^{N\times 2}} \cdot \underbrace{\begin{bmatrix} w_1 \\ w_2 \end{bmatrix}}_{\mathbf{w} \in \mathbb{R}^2} \right\|_2^2$$

Program:

$$\arg\min_{\boldsymbol{w}} \underbrace{\frac{1}{2}\|\boldsymbol{Y} - \boldsymbol{X}^\top \boldsymbol{w}\|_2^2}_{\text{loss function}}$$

How to solve the program:

- Take derivative w.r.t. $\boldsymbol{w}$ of loss function
- Set derivative w.r.t. $\boldsymbol{w}$ to zero
- Solve for $\boldsymbol{w}$

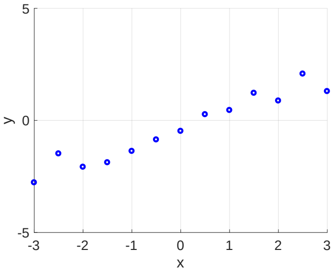
Derivative:

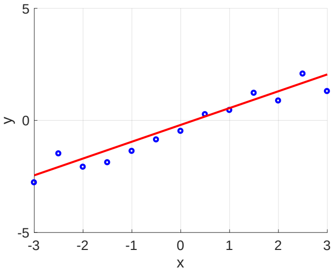
$$\boldsymbol{X}\boldsymbol{X}^\top \boldsymbol{w}^* - \boldsymbol{X}\boldsymbol{Y} = 0$$

Solution:

$$\boldsymbol{w}^* = \left(\boldsymbol{X}\boldsymbol{X}^\top\right)^{-1} \boldsymbol{X}\boldsymbol{Y}$$

Linear regression:





Extensions:

- Higher dimensional problems ($\boldsymbol{x}^{(i)} \in \mathbb{R}^d$)
- Regularization
- Higher order polynomials

Higher dimensional problems ($\boldsymbol{x}^{(i)} \in \mathbb{R}^d$, $y^{(i)} \in \mathbb{R}$)
Model:

$$y^{(i)} = w_0 + \sum_{k=1}^{d} \boldsymbol{x}_k^{(i)} w_k$$

Program:

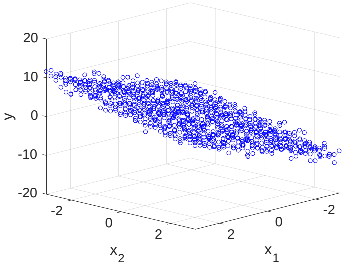
$$\arg\min_{\boldsymbol{w}} \frac{1}{2} \| \underbrace{\boldsymbol{Y}}_{\in \mathbb{R}^N} - \underbrace{\boldsymbol{X}^\top}_{\in \mathbb{R}^{N \times (d+1)}} \underbrace{\boldsymbol{w}}_{\in \mathbb{R}^{d+1}} \|_2^2$$

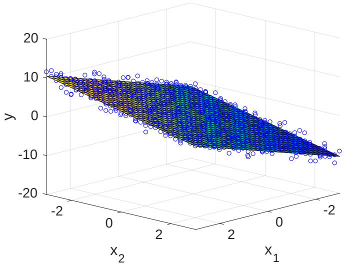
Solution: (obviously the same as before)

$$\boldsymbol{w}^* = \left(\boldsymbol{X}\boldsymbol{X}^\top \right)^{-1} \boldsymbol{X}\boldsymbol{Y}$$

Example:

$$\boldsymbol{w}^* = \left(\boldsymbol{X}\boldsymbol{X}^\top\right)^{-1}\boldsymbol{X}\boldsymbol{Y}$$





What if $N < d + 1$?

Regularization:

we want to make sure that the parameters are not too large

we want to make sure we can invert the matrix

Program:

$$\arg\min_{\boldsymbol{w}} \underbrace{\frac{1}{2}\|\boldsymbol{Y} - \boldsymbol{X}^\top \boldsymbol{w}\|_2^2 + \frac{C}{2}\|\boldsymbol{w}\|_2^2}_{\text{cost function}}$$

Solution:

$$\boldsymbol{w}^* = \left(\boldsymbol{X}\boldsymbol{X}^\top + C\boldsymbol{I}\right)^{-1} \boldsymbol{X}\boldsymbol{Y}$$

Higher order polynomials ($x^{(i)} \in \mathbb{R}$, $y^{(i)} \in \mathbb{R}$)

Model:

$$y^{(i)} = w_2 \cdot \left(x^{(i)}\right)^2 + w_1 \cdot x^{(i)} + w_0$$

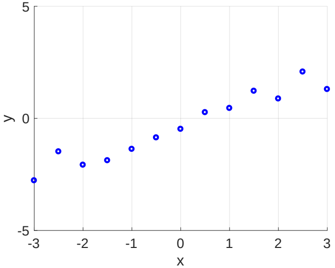
Program:

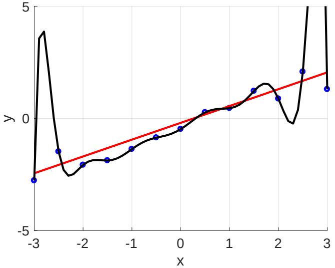
$$\arg\min_{w_0,w_1,w_2} \frac{1}{2} \left\| \underbrace{\begin{bmatrix} y^{(1)} \\ \vdots \\ y^{(N)} \end{bmatrix}}_{\mathbf{Y} \in \mathbb{R}^N} - \underbrace{\begin{bmatrix} \left(x^{(1)}\right)^2 & x^{(1)} & 1 \\ \vdots & \vdots & \vdots \\ \left(x^{(N)}\right)^2 & x^{(N)} & 1 \end{bmatrix}}_{\Phi^\top \in \mathbb{R}^{N \times M}} \cdot \underbrace{\begin{bmatrix} w_2 \\ w_1 \\ w_0 \end{bmatrix}}_{\mathbf{w} \in \mathbb{R}^M} \right\|_2^2$$

Solution:

$$\boldsymbol{w}^* = \left(\Phi\Phi^\top\right)^{-1} \Phi \boldsymbol{Y}$$

Example:





Which model is more reasonable?

Generalizing all aforementioned cases:

- $x^{(i)}$ is some data (e.g., images)
- $\phi(x^{(i)}) \in \mathbb{R}^M$ is a transformation into a feature vector

Model:

$$y^{(i)} = \phi(x^{(i)})^\top \boldsymbol{w}$$

Program:

$$\arg\min_{\boldsymbol{w}} \frac{1}{2} \sum_{i=1}^{N} \left(y^{(i)} - \phi(x^{(i)})^\top \boldsymbol{w} \right)^2$$

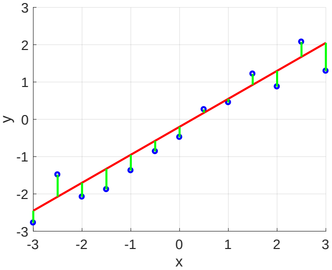
Solution:

$$\boldsymbol{w}^* = \left(\Phi \Phi^\top \right)^{-1} \Phi \boldsymbol{Y} \quad \text{where} \quad \Phi = \left[\phi(x^{(1)}), \cdots, \phi(x^{(N)}) \right] \in \mathbb{R}^{M \times N}$$

Linear regression:

- So far: Error view

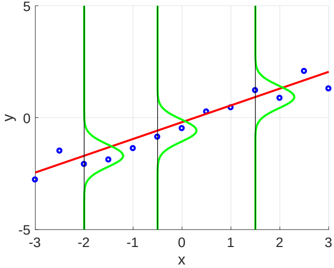
$$\left(y^{(i)} - \phi(x^{(i)})^\top \boldsymbol{w}\right)^2$$



- Alternatively: Probabilistic view

A probabilistic interpretation of linear regression:
Model: Gaussian distribution

$$p(y^{(i)}|x^{(i)}) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(y^{(i)} - \boldsymbol{w}^\top \phi(x^{(i)}))^2\right)$$



How to find $\boldsymbol{w}$?

Maximize likelihood of dataset $\mathcal{D} = \{(x^{(i)}, y^{(i)})\}_{i=1}^N$ assuming samples to be drawn independently from an identical distribution (i.i.d.).

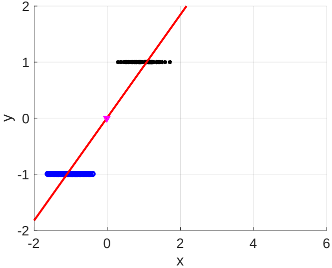
$$
\begin{aligned}
\arg\max_{\boldsymbol{w}} p(\mathcal{D}) &= \arg\max_{\boldsymbol{w}} \prod_i^N p(y^{(i)}|x^{(i)}) \quad \text{(i.i.d.)} \\
&= \arg\max_{\boldsymbol{w}} \sum_i^N \log p(y^{(i)}|x^{(i)}) \quad \text{(log of prod)} \\
&= \arg\max_{\boldsymbol{w}} \sum_i^N \log \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(y^{(i)} - \boldsymbol{w}^\top \phi(x^{(i)}))^2\right) \\
&= \arg\max_{\boldsymbol{w}} \sum_i^N \left(-\frac{1}{2\sigma^2}(y^{(i)} - \boldsymbol{w}^\top \phi(x^{(i)}))^2\right) + C \quad \text{(log-exp)} \\
&= \arg\min_{\boldsymbol{w}} \sum_{i=1}^N \left(y^{(i)} - \phi(x^{(i)})^\top \boldsymbol{w}\right)^2 \quad \text{(take out minus sign)}
\end{aligned}
$$

Linear regression for classification?

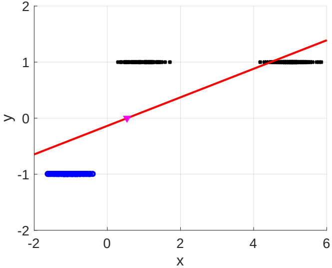
$$y^{(i)} \in \{-1, 1\}$$

Model:

$$y = w_1 x + w_0 \qquad \text{threshold at } y = 0$$



perfect classification



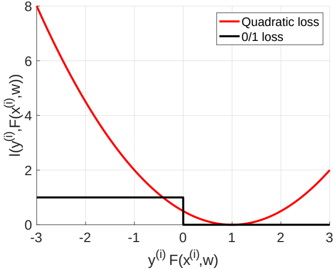
decision boundary shifted

Why is this?

Linear regression: Quadratic loss (recall $y^{(i)} \in \{-1, 1\}$)

$$\ell(y_i, \phi(x^{(i)})^\top \boldsymbol{w}) = \frac{1}{2}(y^{(i)} - \phi(x^{(i)})^\top \boldsymbol{w})^2$$

$$\overset{(y^{(i)})^2=1}{=} \frac{1}{2}(\underbrace{1 - y^{(i)} \underbrace{\phi(x^{(i)})^\top \boldsymbol{w}}_{F(x^{(i)},\boldsymbol{w})}}_{F(x^{(i)},\boldsymbol{w},y^{(i)})})^2$$



We penalize samples that are 'very easy to classify.'

How to fix this?
Next lecture...

Quiz

- Linear regression optimizes what loss function?
- How can we optimize this loss function?
- What are the assumptions?
- What are issues of linear regression applied to classification?

Important topics of this lecture

- We learned about linear regression
- We saw how to solve linear regression problems
- We got to know examples of where to use linear regression
- We understood some shortcomings

What's next:

- Understanding shortcomings of linear classification
- Fixing those shortcomings (logistic regression)

Temporary page!

LaTeX was unable to guess the total number of pages correctl
there was some unprocessed data that should have been ad
final page this extra page has been added to receive it.
If you rerun the document (without altering it) this surplus pag
away, because LaTeX now knows how many pages to expect f
document.