CS4641 Spring 2025

Linear Regression (Cont’)

Bo Dai
School of CSE, Georgia Tech
bodai@cc.gatech.edu

Computation Resources

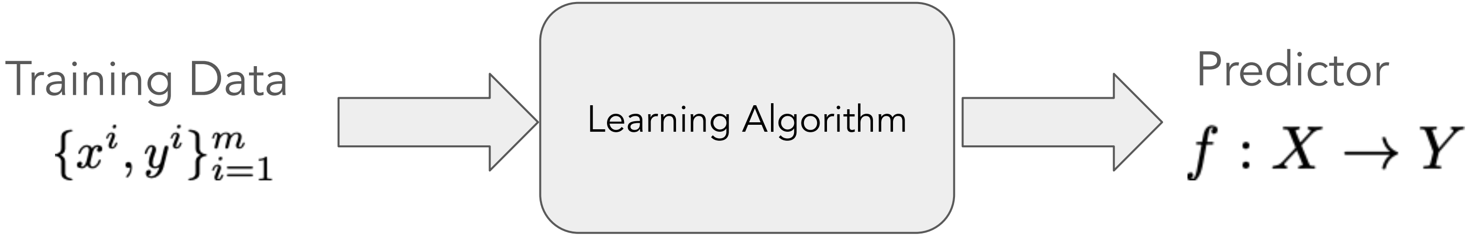
- Google Colaboratory allows free access to run Jupyter Notebooks using GPU resources.
- The Google Cloud Platform and AWS Educate are also good resources.
- The GitHub Student Developer Pack also offers free Microsoft Azure and Digital Ocean credits.
- This semester, we are also offering PACE ICE, Georgia Tech's in-home cluster to students.

ML Algorithm Pipeline

General ML Algorithm Pipeline

1. Build probabilistic models
2. Derive loss function (by MLE or MAP….)
3. Select optimizer

Regression algorithms



General ML Algorithm Pipeline

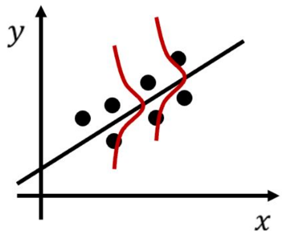
1. Build probabilistic models:
   Gaussian noise + linear model/polynomial model
2. Derive loss function:
   MLE vs. MAP
3. Select optimizer
   Necessary Condition vs. (Stochastic) GD

Probabilistic Model: Gaussian Likelihood

- Assume y is a linear in x plus noise ϵ

$$y = \theta^\top x + \epsilon$$

- Assume ϵ follows a Gaussian $N(0, \sigma)$ $\quad \epsilon \sim \mathcal{N}(0, \sigma)$



$$\mathbb{E}[y] = \theta^\top x + \mathbb{E}[\epsilon] = \theta^\top x$$

$$y = \theta^\top x + \epsilon \sim \mathcal{N}(\theta^\top x, \sigma)$$

$$p(y^i \mid x^i; \theta) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y^i - \theta^\top x^i)^2}{2\sigma^2}\right)$$

Maximum log-Likelihood Estimation (MLE)

$$L(\theta) = \prod_{i}^{m} p(y^i | x^i; \theta) = \left(\frac{1}{\sqrt{2\pi}\sigma}\right)^m \exp\left(-\frac{\sum_i^m (y^i - \theta^\top x^i)^2}{2\sigma^2}\right)$$

$$\max_{\theta} \; \log L(\theta) = -\frac{1}{2\sigma^2} \sum_{i=1}^{m} (y^i - \theta^\top x^i)^2 - m \log(\sqrt{2\pi}\sigma)$$

Select Optimizer

$$\min_{\theta} \; -\log L(\theta) \propto \frac{1}{m}\sum_{i=1}^{m}(y^i - \theta^\top x^i)^2$$

- Necessary Condition
- (Stochastic) Gradient Descent

Gradient Method Revisit

$$\min_{\theta} \underbrace{-\log L(\theta) \propto \frac{1}{m}\sum_{i=1}^{m}(y^i - \theta^\top x^i)^2}_{f(\theta)}$$

choose an initial $\theta^1 \in \mathbf{R}^d$ and $h^1 > 0$ (e.g., $\theta^1 = 0, h^1 = 1$)
for $k = 1,2,\ldots, k^{\max}$

1. compute $\nabla f(\theta^k)$; quit if $\|\nabla f(\theta^k)\|_2$ is small enough
2. form tentative update $\theta^{\text{tent}} = \theta^k - h^k \nabla f(\theta^k)$

Gradient Calculation

$$\min_{\theta} \; -\log L(\theta) \propto \frac{1}{m}\sum_{i=1}^{m}(y^i - \theta^\top x^i)^2$$

$$\frac{\partial \log L(\theta)}{\partial \theta} = -\frac{2}{m}\sum_{i=1}^{m}(y^i - \theta^\top x^i)x^{i\top}$$

Gradient Method Revisit

$$\min_{\theta} \; \underbrace{-\log L(\theta) \propto \frac{1}{m}\sum_{i=1}^{m}(y^i - \theta^\top x^i)^2}_{f(\theta)}$$

choose an initial $\theta^1 \in \mathbf{R}^d$ and $h^1 > 0$ (e.g., $\theta^1 = 0, h^1 = 1$)
for $k = 1,2,\ldots,k^{\max}$

1. compute $\nabla f(\theta^k)$; quit if $\left\|\nabla f(\theta^k)\right\|_2$ is small enough
2. form tentative update $\theta^{\text{tent}} = \theta^k - h^k \nabla f(\theta^k)$

$$\theta^k + \frac{h^k}{m}\sum_{i=1}^{m}(y^i - (\theta^k)^\top x^i)(x^i)^\top$$

Gradient Method Revisit

choose an initial $\theta^1 \in \mathbf{R}^d$ and $h^1 > 0$ (e.g., $\theta^1 = 0, h^1 = 1$)
for $k = 1,2, \dots, k^{\text{max}}$

Stochastic Approximation

1. compute ~~$\nabla f(\theta^k)$~~; quit if $\|\nabla f(\theta^k)\|_2$ is small enough
2. form tentative update $\theta^{\text{tent}} = \theta^k - h^k \nabla f(\theta^k)$

Stochastic Gradient Descent

$$\frac{\partial \log L(\theta)}{\partial \theta} = -\frac{2}{m}\sum_{i=1}^{m}(y^i - \theta^\top x^i)x^i \quad \approx \quad \left(y^i - \hat{\theta}^{t\top} x^i\right) x^i$$

Initialize $\theta^0 \in \mathbb{R}^d$ randomly Iterate until convergence:

1. Randomly sample a point (x_i, y_i) from the n data points
2. Compute noisy gradient $\tilde{g}^t = \left(y^i - (\theta^t)^\top x^i\right)\left(x^i\right)^\top$
3. Update (GD): $\theta^{t+1} = \theta^t - \eta \tilde{g}^t$

Optimizer Comparison

- Solve normal equations

$$(X^\top X)\hat{\theta} = X^\top y$$

  - Pros: a single-shot algorithm! Easiest to implement
  - Cons: need to compute inverse, expensive, numerical issue (e.g., matrix is singular …)
- Gradient descent

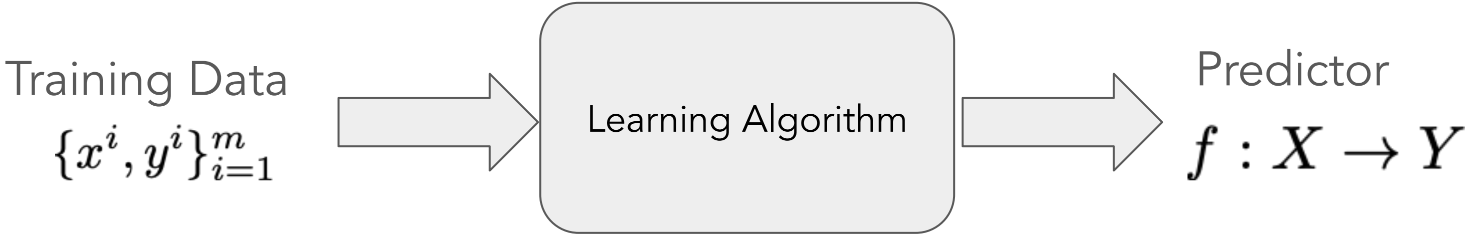
$$\hat{\theta}^{t+1} \leftarrow \hat{\theta}^t + \frac{\alpha}{m} \sum_{i=1}^m \left(y^i - (\hat{\theta}^t)^\top x^i\right) x^i$$

  - Pros: fast-converging, easy to implement
  - Cons: need to read all data
- Stochastic gradient update rule

$$\hat{\theta}^{t+1} \leftarrow \hat{\theta}^t + \beta \left(y^i - (\hat{\theta}^t)^\top x^i\right) x^i$$

  - Pros: low per-step cost
  - Cons: slow-converging

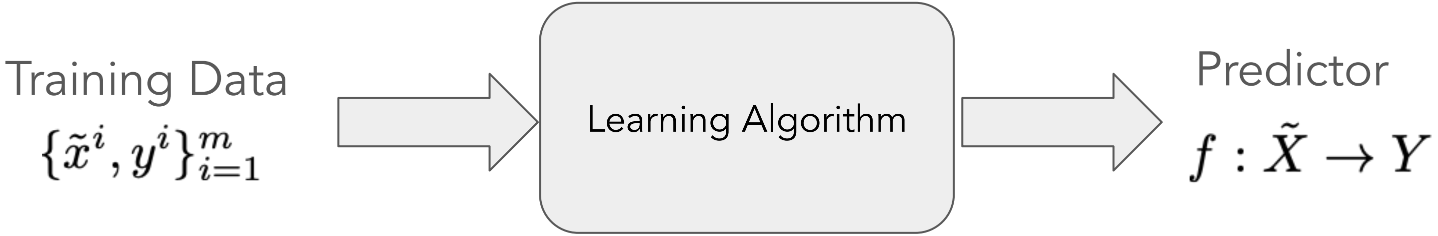
Regression algorithms



General ML Algorithm Pipeline

1. Build probabilistic models:
   Gaussian noise + linear model/ polynomial model
2. Derive loss function:
   MLE vs. MAP
3. Select optimizer
   Necessary Condition vs. (Stochastic) GD

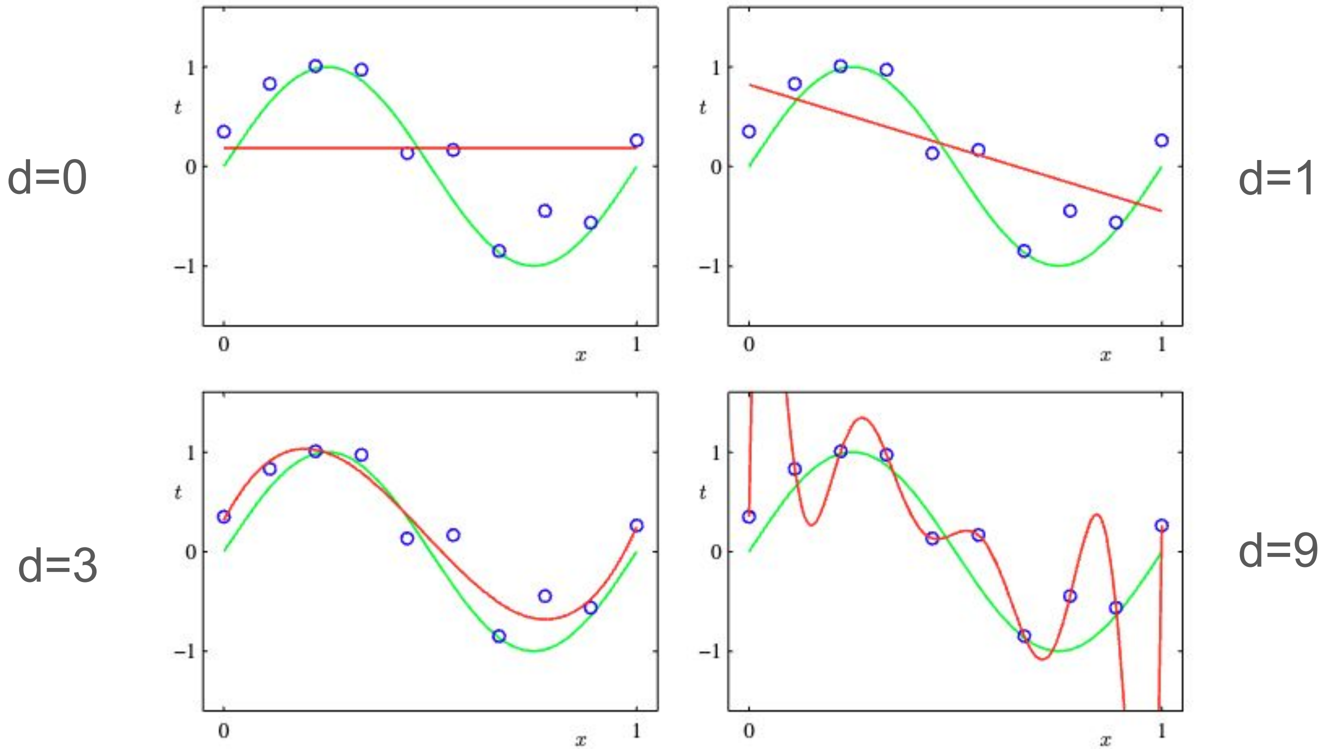
Polynomial Regression



- Features:
  - Living area, distance to campus, # of bedroom …
  - Denotes as $\tilde{x} = [1, x_1, (x_1)^2, \ldots, (x_1)^d, \ldots, x_n, \ldots, (x_n)^d]$
- Target
  - Rent
  - Denote as y
- Training set
  - $\tilde{X} = [\tilde{x}^1, \tilde{x}^2, \ldots, \tilde{x}^m]$
  - $y = (y^1, y^2, \ldots, y^m)^\top$

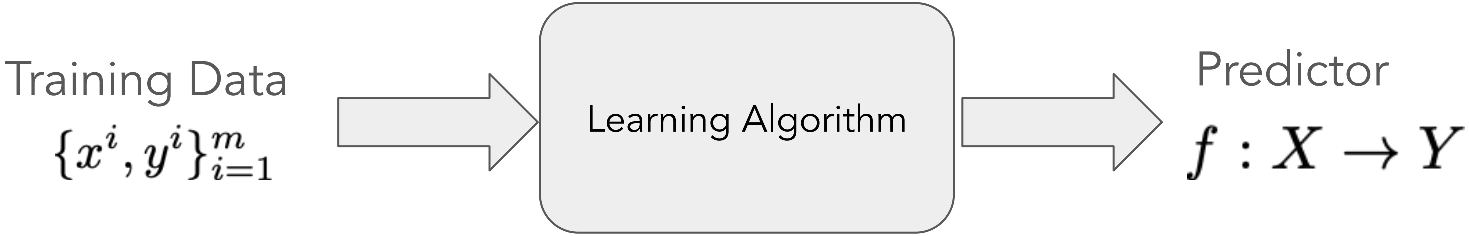
$$\begin{aligned} y = \theta_0 &+ \theta_1^1 x_1 + \theta_1^2 (x_1)^2 + \theta_1^3 (x_1)^3 + \ldots + \theta_1^d (x_1)^d \\ &+ \theta_2^1 x_2 + \theta_2^2 (x_2)^2 + \theta_2^3 (x_2)^3 + \ldots + \theta_2^d (x_2)^d \\ &+ \ldots \\ &+ \theta_n^1 x_n + \theta_n^2 (x_n)^2 + \theta_n^3 (x_n)^3 + \ldots + \theta_n^d (x_n)^d \end{aligned}$$

MLE Overfitting with Increased Degree



$$y = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + ...\theta_9 x^9$$

Regression algorithms



General ML Algorithm Pipeline

1. Build probabilistic models:
   Gaussian noise + linear model/ polynomial model
2. Derive loss function:
   MLE vs. MAP
3. Select optimizer
   Necessary Condition vs. (Stochastic) GD

Maximum a Posteriori (MAP)

$$L(\theta) = \prod_{i}^{m} p(y^i|x^i;\theta) = \left(\frac{1}{\sqrt{2\pi}\sigma}\right)^m \exp\left(-\frac{\sum_i^m (y^i - \theta^\top x^i)^2}{2\sigma^2}\right)$$

Likelihood

$$p(\theta) \propto \exp(-\lambda\|\theta\|_2^2)$$

Gaussian Prior

$$p(\theta|\{x^i, y^i\}_{i=1}^m) = \frac{\prod_{i=1}^m p(y^i|x^i, \theta)p(\theta)}{\int \prod_{i=1}^m p(y^i|x^i, \theta)p(\theta)d\theta}$$

Posterior:
Bayes' Rule

$$\max_\theta \ \log p(\theta|\{x^i, y^i\}_{i=1}^m) = \log L(\theta) + \log p(\theta)$$

$$\propto -\frac{1}{m}\sum_{i=1}^m (y^i - \theta^\top x^i)^2 - \boxed{\lambda\|\theta\|_2^2}$$

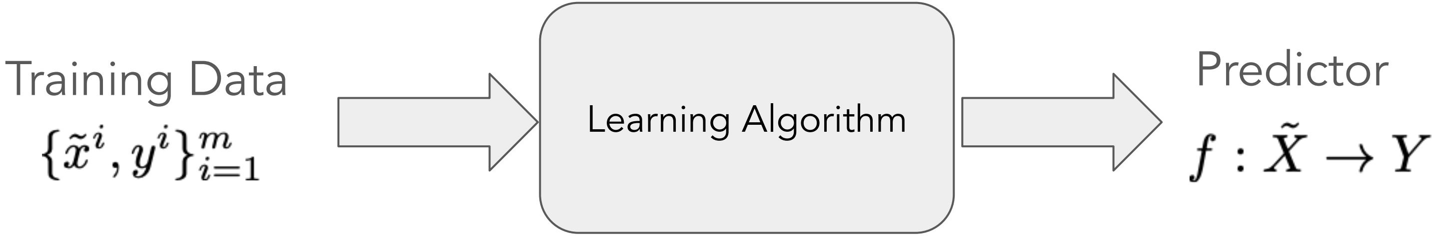
Ridge Regression

Select Optimizer

$$\min_{\theta} \; -\log p(\theta | \{x^i, y^i\}_{i=1}^m) \propto \frac{1}{m} \sum_{i=1}^{m} (y^i - \theta^\top x^i)^2 + \boxed{\lambda \|\theta\|_2^2}$$

- Necessary Condition
- (Stochastic) Gradient Descent

Polynomial Regression



- Features:
  - Living area, distance to campus, # of bedroom …
  - Denotes as $\tilde{x} = [1, x_1, (x_1)^2, \ldots, (x_1)^d, \ldots, x_n, \ldots, (x_n)^d]$
- Target
  - Rent
  - Denote as y
- Training set
  - $\tilde{X} = [\tilde{x}^1, \tilde{x}^2, \ldots, \tilde{x}^m]$
  - $y = (y^1, y^2, \ldots, y^m)^\top$

$$
\begin{aligned}
y = \theta_0 &+ \theta_1^1 x_1 + \theta_1^2 (x_1)^2 + \theta_1^3 (x_1)^3 + \ldots + \theta_1^d (x_1)^d \\
&+ \theta_2^1 x_2 + \theta_2^2 (x_2)^2 + \theta_2^3 (x_2)^3 + \ldots + \theta_2^d (x_2)^d \\
&+ \ldots \\
&+ \theta_n^1 x_n + \theta_n^2 (x_n)^2 + \theta_n^3 (x_n)^3 + \ldots + \theta_n^d (x_n)^d
\end{aligned}
$$

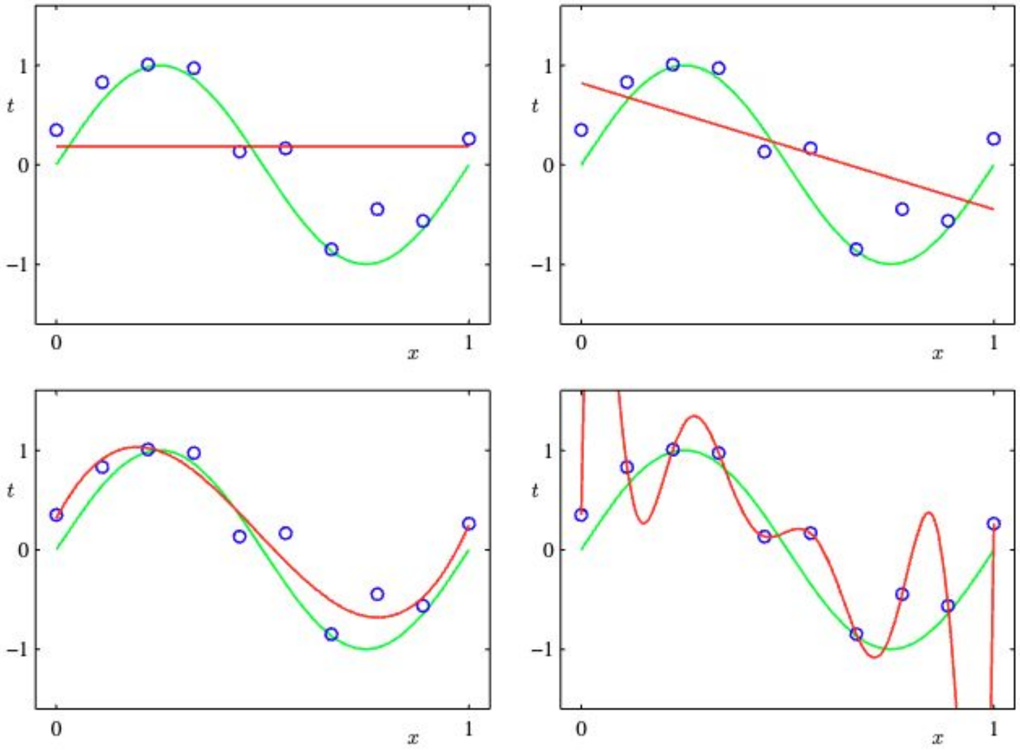
Necessary Condition

$$\frac{2}{m}\sum_{i=1}^{m} y^i x^i - \frac{2}{m}\sum_{i=1}^{m} x^i (x^i)^\top \theta + 2\lambda\theta = 0$$

$$\frac{2}{m} X y - \frac{2}{m} X X^\top \theta + 2\lambda\theta = 0$$

$$\Rightarrow \hat{\theta} = (X X^\top + \lambda m I)^{-1} X y$$

MLE Overfitting with Increased Degree



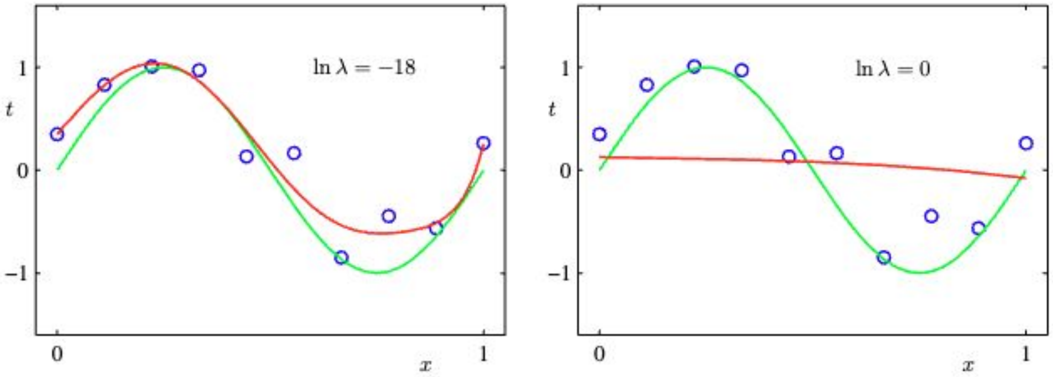
d=0

d=1

d=3

d=9

Best Degree?



	$\ln\lambda = -\infty$	$\ln\lambda = -18$	$\ln\lambda = 0$
θ_0	0.35	0.35	0.13
θ_1	232.37	4.74	-0.05
θ_2	-5321.83	-0.77	-0.06
θ_3	48568.31	-31.97	-0.05
θ_4	-231639.30	-3.89	-0.03
θ_5	640042.26	55.28	-0.02
θ_6	-1061800.52	41.32	-0.01
θ_7	1042400.18	-45.95	-0.00
θ_8	-557682.99	-91.53	0.00
θ_9	125201.43	72.68	0.01

$$y = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \ldots \theta_9 x^9$$

- MLE with appropriate d
- MAP with large d, regularization will automatically select the appropriate model

MLE vs. MAP

MLE

- We chose the "best" θ that maximized the likelihood given data
- No prior

$$\hat{\theta} = (XX^\top)^{-1}Xy$$

- Numerical issue
- Overfitting

MAP

- We chose the "best" θ that maximized the posterior given data
- Prior matters

$$\hat{\theta} = (XX^\top + \lambda m I)^{-1}Xy$$

- No numerical issue
- Mitigate overfitting

CS4641 Spring 2025

Logistic Regression

Bo Dai
School of CSE, Georgia Tech
bodai@cc.gatech.edu

Georgia Tech
College of Computing

ML Algorithm Pipeline

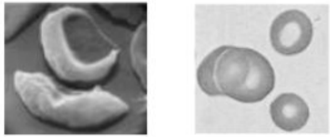
General ML Algorithm Pipeline

1. Build probabilistic models
2. Derive loss function (by MLE or MAP….)
3. Select optimizer

Classification Tasks

	Feature, X		Label, Y

Diagnosing sickle cell anemia



Anemic cell
Healthy cell

Tax Fraud Detection

Refund	Marital Status	Taxable Income
No	Married	80K

Cheat
?

Web Classification

Sports
Science
News

ML Algorithm Pipeline

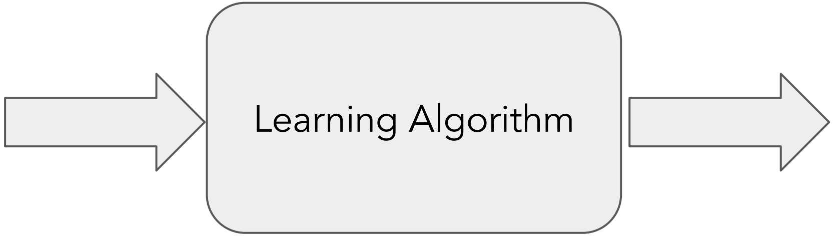
General ML Algorithm Pipeline

1. Build probabilistic models
2. Derive loss function (by MLE or MAP….)
3. Select optimizer

Classification algorithms

Training Data

$\{x^i, y^i\}_{i=1}^{m}$



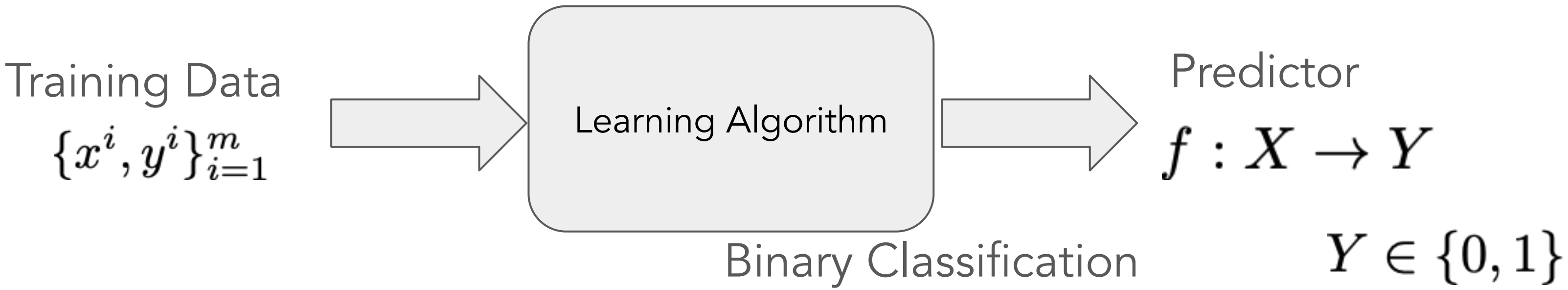
Predictor

$f : X \rightarrow Y$

Binary Classification $Y \in \{0, 1\}$

Multiclass Classification $Y \in \{0, 1, \ldots, k\}$

Binary Classification Algorithms



General ML Algorithm Pipeline

1. Build probabilistic models
2. Derive loss function (by MLE or MAP)
3. Select optimizer

Probabilistic Model in Regression: Gaussian Likelihood

$$p(y^i \mid x^i; \theta) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{\left(y^i - \theta^\top x^i\right)^2}{2\sigma^2}\right)$$

Probabilistic Model in Classification: Bernoulli Likelihood

$$\begin{cases} p & \text{if } y = 1 \\ 1 - p & \text{if } y = 0 \end{cases}$$

$$p \in [0, 1]$$

$$p(y) = p^y (1 - p)^{(1-y)}$$

Probabilistic Model in Classification: Bernoulli Likelihood

$$p(y) = p^y (1-p)^{(1-y)} \qquad p \in [0,1]$$

$$p(y|x;\theta) = p(y=1|\theta^\top x)^y \{1 - p(y=1|\theta^\top x)\}^{(1-y)}$$

Probabilistic Model in Classification: Bernoulli Likelihood

$$p(y) = p^y(1-p)^{(1-y)} \qquad p \in [0,1]$$

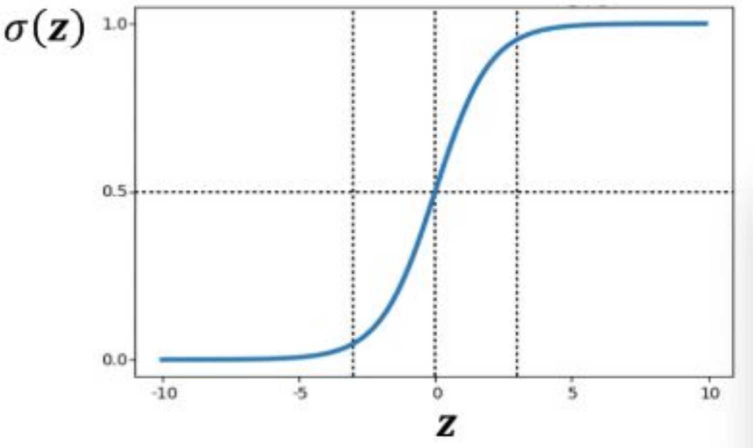
$$p(y|x;\theta) = p(y=1|\theta^\top x)^y\{1 - p(y=1|\theta^\top x)\}^{(1-y)}$$

$$p(y=1|\theta^\top x) \in [0,1]$$

Probabilistic Model in Classification: Bernoulli Likelihood

$$p(y = 1|\theta^\top x) \in [0, 1] \qquad \theta^\top x \in \mathbb{R}$$

$$\sigma(\mathbf{z}) = \frac{1}{1 + e^{-\mathbf{z}}} = \frac{e^{\mathbf{z}}}{1 + e^{\mathbf{z}}}$$



$$p(y = 1|\theta^\top x) = \sigma(\theta^\top x) \in [0, 1]$$

Probabilistic Model in Classification: Bernoulli Likelihood

- Logistic regression model

$$p(y=1|x,\theta) = \frac{1}{1+\exp(-\theta^\top x)}$$

- Note that

$$p(y=0|x,\theta) = 1 - \frac{1}{1+\exp(-\theta^\top x)} = \frac{\exp(-\theta^\top x)}{1+\exp(-\theta^\top x)}$$

Logistic Regression is a Linear Classifier

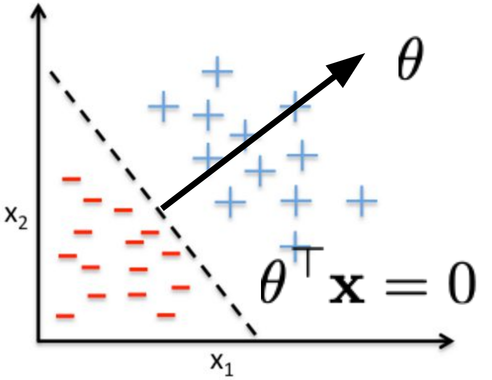
- Decision boundaries for Logistic Regression?
  - At the decision boundary, label 1/0 are equiprobable.

$$P(y=1|\mathbf{x},\theta) = \frac{1}{1+e^{-\theta^\top \mathbf{x}}}, \qquad P(y=0|\mathbf{x},\theta) = \frac{1}{1+e^{\theta^\top \mathbf{x}}}$$

to be equal: $e^{-\theta^\top \mathbf{x}} = e^{\theta^\top \mathbf{x}}$, whose only solution is $\theta^\top \mathbf{x} = 0$.

✓ ⇒ Decision boundary is linear.

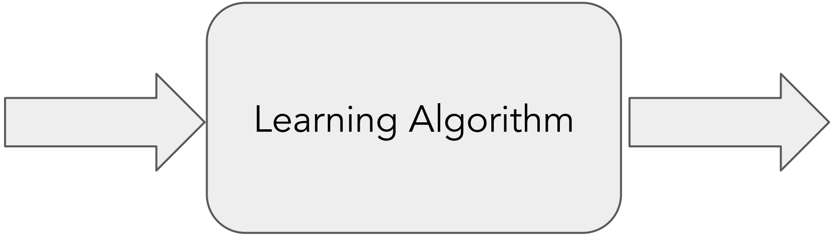
✓ ⇒ Logistic regression is a probabilistic linear classifier.



Binary Classification Algorithms

Training Data

$\{x^i, y^i\}_{i=1}^m$



Predictor

$f : X \to Y$

Binary Classification

$Y \in \{0, 1\}$

General ML Algorithm Pipeline

1. Build probabilistic models
2. Derive loss function (by MLE or MAP)
3. Select optimizer

MLE

- Logistic regression model

$$p(y=1|x,\theta) = \frac{1}{1+\exp(-\theta^\top x)}$$

- Note that

$$p(y=0|x,\theta) = 1 - \frac{1}{1+\exp(-\theta^\top x)} = \frac{\exp(-\theta^\top x)}{1+\exp(-\theta^\top x)}$$

- Plug in

$$\begin{aligned} l(\theta) :&= \log \prod_{i=1}^{n} p(y^i|x^i,\theta) \qquad \text{(Bernoulli)} \\ &= \sum_{i=1}^{n} \log\left(\frac{\exp(-\theta^\top x^i)}{1+\exp(-\theta^\top x^i)}\right) \underbrace{I(y^i=0)}_{1-y^i} + \log\left(\frac{1}{1+\exp(-\theta^\top x^i)}\right) \underbrace{I(y^i=1)}_{y^i} \\ &= \sum_{i=1}^{n} (y^i-1)\theta^\top x^i - \log(1+\exp(-\theta^\top x^i)) \end{aligned}$$

MAP

- Logistic regression model

$$p(y=1|x,\theta) = \frac{1}{1+\exp(-\theta^\top x)}$$

- Note that

$$p(y=0|x,\theta) = 1 - \frac{1}{1+\exp(-\theta^\top x)} = \frac{\exp(-\theta^\top x)}{1+\exp(-\theta^\top x)}$$

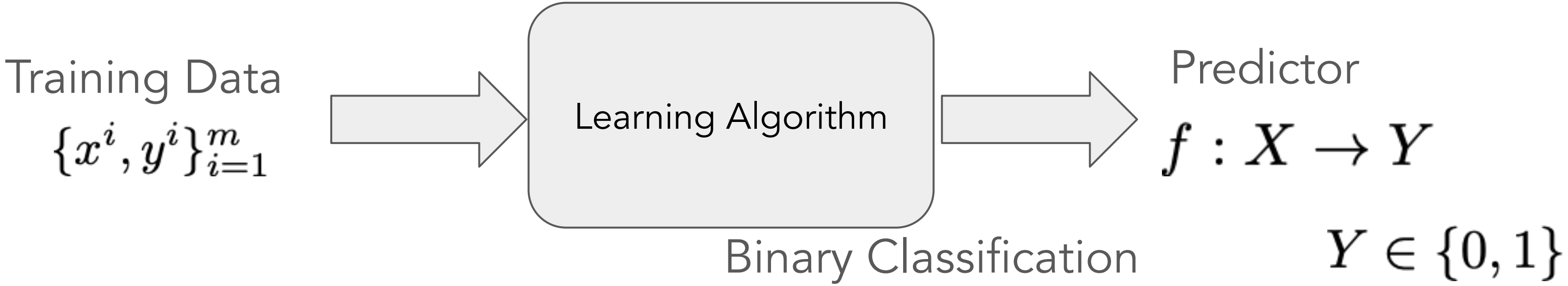
-

$$p(\theta) \propto \exp(-\lambda\|\theta\|_2^2)$$

$$\max_\theta \; \log p(\theta|\{x^i, y^i\}_{i=1}^m) = \log L(\theta) + \log p(\theta)$$

$$= \sum_i (y^i - 1)\,\theta^\top x^i - \log(1 + \exp(-\theta^\top x^i)) \; - \lambda\|\theta\|_2^2$$

Binary Classification Algorithms



General ML Algorithm Pipeline

1. Build probabilistic models
2. Derive loss function (by MLE or MAP)
3. Select optimizer

Select Optimizer

$$\max_{\theta} \ \log L(\theta) = \sum_{i} (y^i - 1)\, \theta^\top x^i - \log(1 + \exp(-\theta^\top x^i))$$

- Necessary Condition
- (Stochastic) Gradient Descent

Gradient Calculation of MLE

$$\max_{\theta} \; \log L(\theta) = \sum_i (y^i - 1)\, \theta^\top x^i - \log(1 + \exp(-\theta^\top x^i))$$

$$\frac{\partial \log L(\theta)}{\partial \theta} = \sum_i (y^i - 1) x^i + \frac{\exp(-\theta^\top x^i) x^i}{1 + \exp(-\theta^\top x^i)}$$

Gradient Calculation of MAP

$$\max_{\theta} \log L(\theta) = \sum_i (y^i - 1)\,\theta^\top x^i - \log(1 + \exp(-\theta^\top x^i)) - \lambda \|\theta\|_2^2$$

$$\frac{\partial \log L(\theta)}{\partial \theta} = \sum_i (y^i - 1)x^i + \frac{\exp(-\theta^\top x^i)x^i}{1 + \exp(-\theta^\top x^i)} - 2\lambda\theta$$

Necessary Condition?

$$\frac{\partial \log L(\theta)}{\partial \theta} = \sum_i (y^i - 1)x^i + \frac{\exp(-\theta^\top x^i)x^i}{1 + \exp(-\theta^\top x^i)} = 0$$

Nonlinear Equation!
Does NOT admit a closed-form solution

(Stochastic) Gradient Descent

- Initialize parameter θ^0

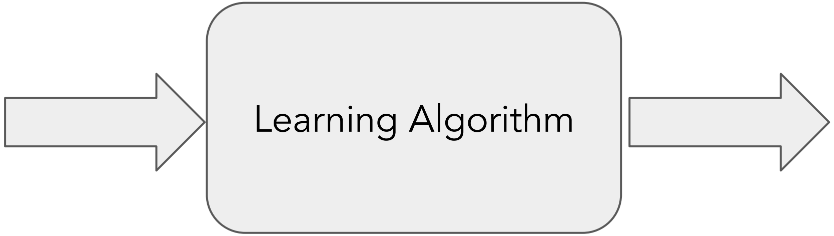
- Do

$$\theta^{t+1} \leftarrow \theta^t + \eta \sum_i (y^i - 1)x^i + \frac{\exp(-\theta^\top x^i)x^i}{1 + \exp(-\theta^\top x)} \left[-2\lambda\theta \right]$$

Binary Classification Algorithms

Training Data

$\{x^i, y^i\}_{i=1}^m$



Predictor

$f : X \to Y$

Binary Classification

$Y \in \{0, 1\}$

Logistic Regression Pipeline

1. Build probabilistic models: Bernoulli Distribution
2. Derive loss function: MLE and MAP
3. Select optimizer: (Stochastic) Gradient Descent

Q&A