Georgia Tech
College of Computing

CX4240 Spring 2026

Logistic Regression

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

Project

Team Size

- Each project must be completed in a team of ~~3-5~~, 5-6.
- Once you have formed your group, please share it on the Team Signup sheet:

  CX4240 Project Team Signup

- If you have trouble forming a group, please send us an email and we will help you find project partners.

The team formation email will be due at 11:59 PM on Feb 16th.

Project

Project Topics:

- Reproduce classic papers, include but not limited to:
  - Deep Residual Learning for Image Recognition
  - Auto-Encoding Variational Bayes
  - A Simple Framework for Contrastive Learning of Visual Representations.
  - Sequence to Sequence Learning with Neural Networks
  - Efficient Estimation of Word Representations in Vector Space
  - etc
- You may also refer to the https://cs231n.stanford.edu/project.html.

Project

Deliverables:

- Presentation (15%): Every team will have a 10-mins presentation.
- Final Report (25%): *All write-ups should use the NeurIPS style.*

  Your final report is expected to be up to 6 pages excluding references. It should have roughly the following format:

  - *Introduction: problem definition and motivation*
  - *Background & Related Work: background info and literature survey*
  - *Methods – Overview of your proposed method – Intuition on why should it be better than the state of the art – Details of models and algorithms that you developed*
  - *Experiments – Description of your testbed and a list of questions your experiments are designed to answer – Details of the experiments and results*
  - *Conclusion: discussion and future work*

The project final report will be due at 11:59 PM on May 4th

Project

Criteria:

- 30% for proposed method (soundness and originality)
- 30% for correctness, completeness, and difficulty of experiments and figures
- 20% for empirical and theoretical analysis of results and methods
- 20% for quality of writing (clarity, organization, flow, etc.)

Project

Criteria:

- 30% for proposed method (soundness and originality)
- 30% for correctness, completeness, and difficulty of experiments and figures
- 20% for empirical and theoretical analysis of results and methods
- 20% for quality of writing (clarity, organization, flow, etc.)

- 10% for presentation completed in 10mins.

Computation Resources

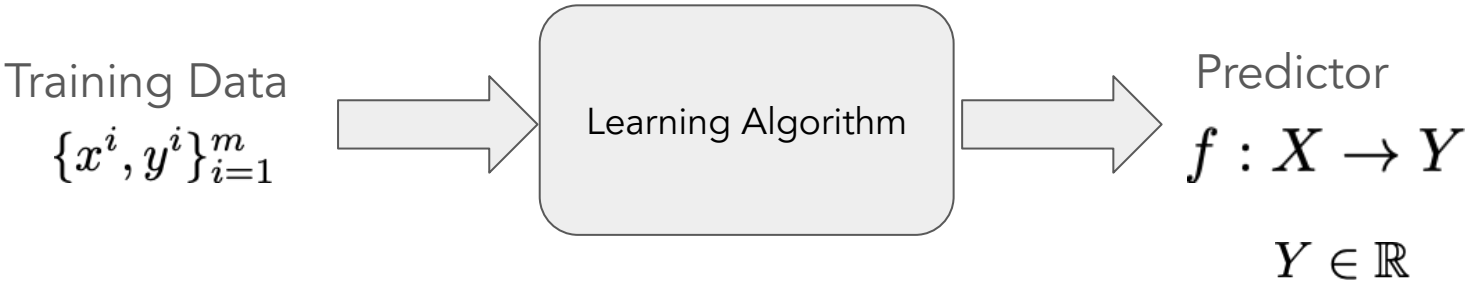
- Google Colaboratory allows free access to run Jupyter Notebooks using GPU resources and Gemini!.
- The Google Cloud Platform and AWS Educate are also good resources.
- The GitHub Student Developer Pack also offers free Microsoft Azure and Digital Ocean credits.
- This semester, we are also offering 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



Linear Regression Pipeline

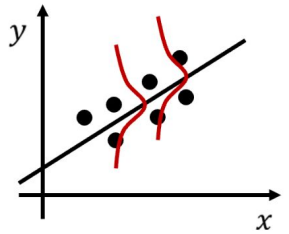
1. Build probabilistic models:
   Gaussian Distribution + Linear Model
2. Derive loss function: MLE and MAP
3. Select optimizer
   Necessary Condition vs. (Stochastic) GD

Probabilistic Model: Gaussian Likelihood

- Assume y is a linear in x plus noise ϵ

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

- Assume ϵ follows a Gaussian $N(0, \sigma)$



$$\epsilon \sim \mathcal{N}(0, \sigma)$$

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

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

Statistics

- Expectation: The mean value, center of mass, first moment:

$$E_X[g(X)] = \int_{-\infty}^{\infty} g(x)p_X(x)dx$$

- N-th moment: $g(x) = x^n$
- N-th central moment: $g(x) = (x - \mu)^n \quad \mu = E_X[x]$
- Mean: $E_X[X] = \int_{-\infty}^{\infty} xp_X(x)dx$
  - $E[\alpha X] = \alpha E[X]$
  - $E[\alpha + X] = \alpha + E[X]$
- Variance(Second central moment): $Var(x) = E_X[(X - E_X[X])^2] = E_X[X^2] - E_X[X]^2$
  - $Var(\alpha X) = \alpha^2 Var(X)$
  - $Var(\alpha + X) = Var(X)$

Probabilistic Model: Gaussian Likelihood

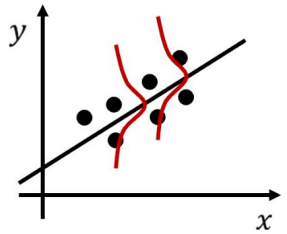
- Assume y is a linear in x plus noise ϵ

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

- Assume ϵ follows a Gaussian $N(0, \sigma)$

$$\epsilon \sim \mathcal{N}(0, \sigma)$$

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



Operations on Gaussian R.V.

- The linear transform of a Gaussian r.v. is a Gaussian. Remember that no matter how x is distributed

$$E(AX + b) = AE(X) + b$$
$$Cov(AX + b) = ACov(X)A^{\top}$$

this means that for Gaussian distributed quantities:

$$X \sim N(\mu, \Sigma) \rightarrow AX + b \sim N(A\mu + b, A\Sigma A^{\top})$$

- The sum of two independent Gaussian r.v. is a Gaussian

$$Y = X_1 + X_2, X_1 \perp X_2 \rightarrow \mu_y = \mu_1 + \mu_2, \Sigma_y = \Sigma_1 + \Sigma_2$$

- The multiplication of two Gaussian functions is another Gaussian function (although no longer normalized)

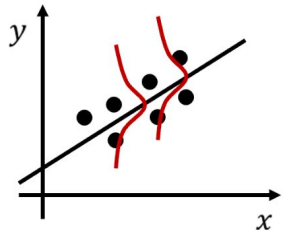
$$N(a, A)N(b, B) \propto N(c, C)$$
$$where\, C = (A^{-1} + B^{-1})^{-1}, c = CA^{-1}a + CB^{-1}b$$

Probabilistic Model: Gaussian Likelihood

- Assume y is a linear in x plus noise ϵ

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

- Assume ϵ follows a Gaussian $N(0, \sigma)$

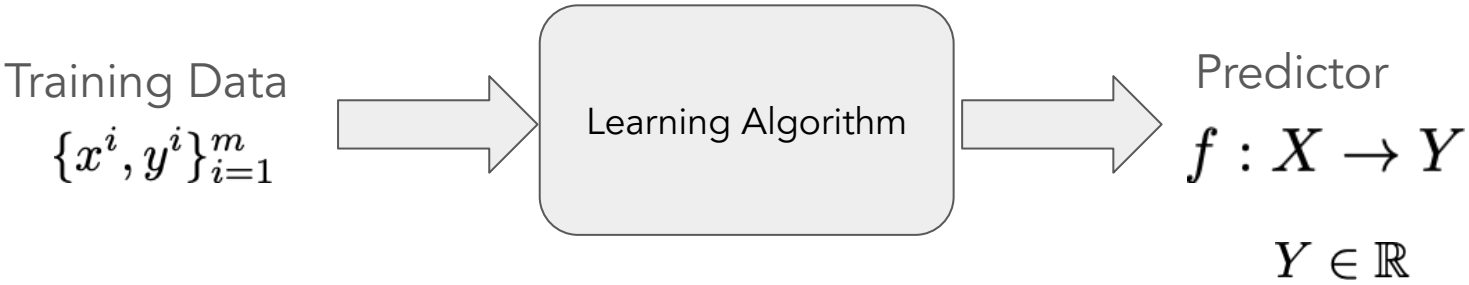


$$\epsilon \sim \mathcal{N}(0, \sigma)$$

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

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

Regression Algorithms



Linear Regression Pipeline

1. Build probabilistic models:
   Gaussian Distribution + Linear Model
2. Derive loss function: MLE and MAP
3. Select optimizer
   Necessary Condition vs. (Stochastic) GD

Maximum log-Likelihood Estimation (MLE)

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

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

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

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

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

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

MLE vs. MAP

MLE

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

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

MAP

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

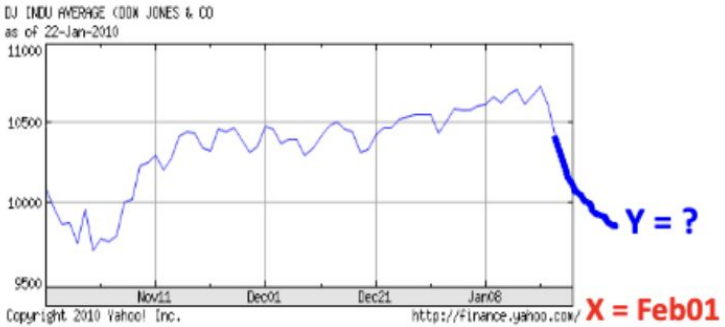
$$\hat{\theta} = (XX^\top + \lambda m I)^{-1}Xy$$

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

Supervised Learning

Goal: Construct a predictor f: X → Y

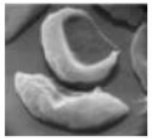
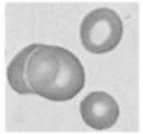
Sports
Science
News



Classification:
discrete categories

Regression:
Real-valued numbers

Classification Tasks

	Feature, X		Label, Y
Diagnosing sickle cell anemia			Anemic cell Healthy cell
Tax Fraud Detection			
Web Classification			Sports Science News

ML Algorithm Pipeline

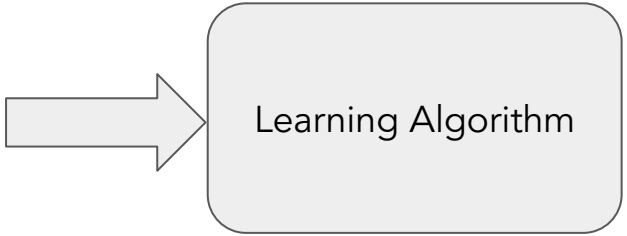
General ML Algorithm Pipeline

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

$f : X \to Y$

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

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

Tax Fraud Detection

Refund	Marital Status	Taxable Income
No	Married	80K

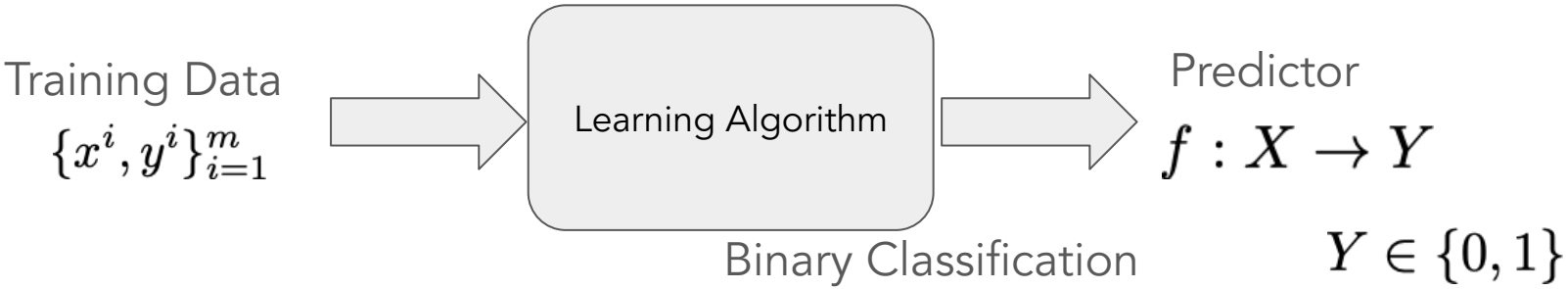
→ | Cheat |
|---|
| ? |

Web Classification

→ Sports Science News

Regression algorithms

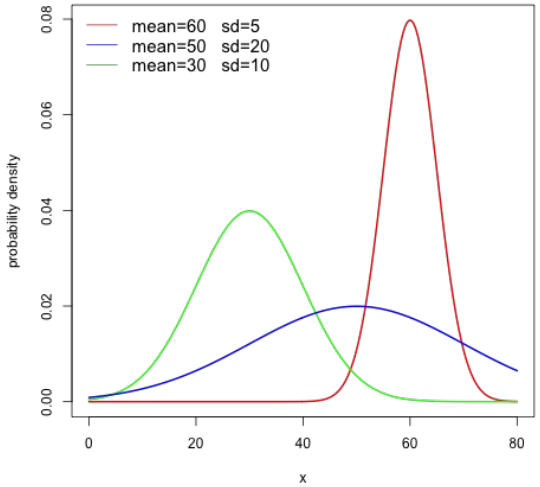
Binary Classification Algorithms



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

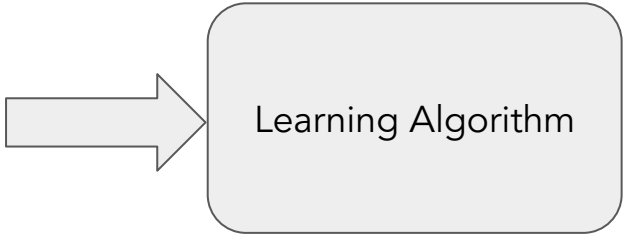


$$p(y^i \mid x^i; \theta) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y^i - \theta^\top x^i)^2}{2\sigma^2}\right)$$

Classification algorithms

Training Data

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



Predictor

$f : X \to Y$

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

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

Tax Fraud Detection

Refund	Marital Status	Taxable Income
No	Married	80K

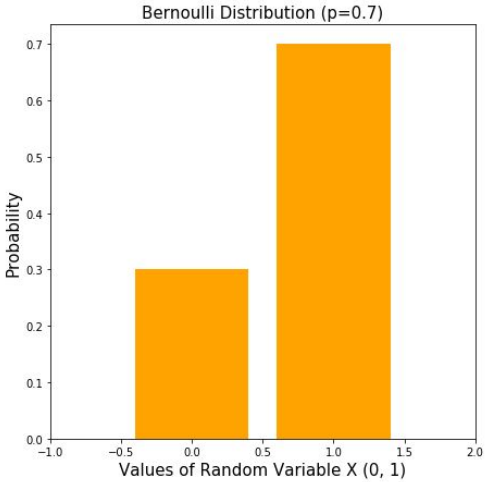
→ | Cheat |
|---|
| ? |

Web Classification

→ Sports Science News

Probabilistic Model in Classification: Bernoulli Likelihood

$$\begin{cases} p & \text{if } y = 1 \\ 1 - p & \text{if } y = 0 \end{cases} \qquad p \in [0, 1]$$



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

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

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

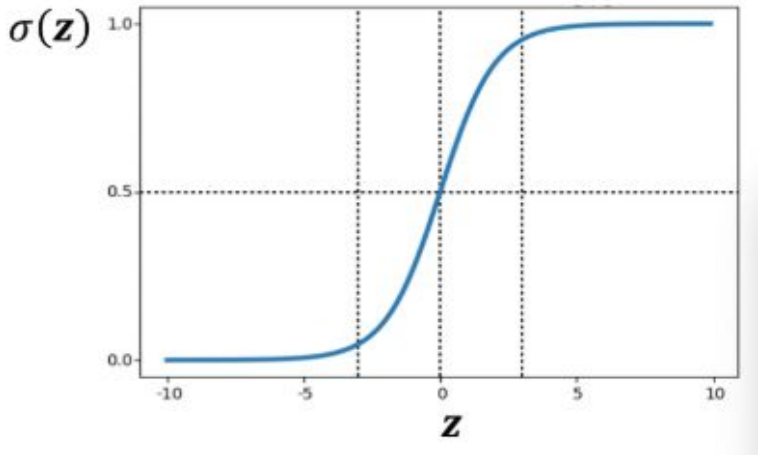
$$p(y|x;\theta) = p(y=1|\theta^\top x)^y\{1-p(y=1|\theta^\top x)\}^{(1-y)}$$

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

Probabilistic Model in Classification: Bernoulli Likelihood

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

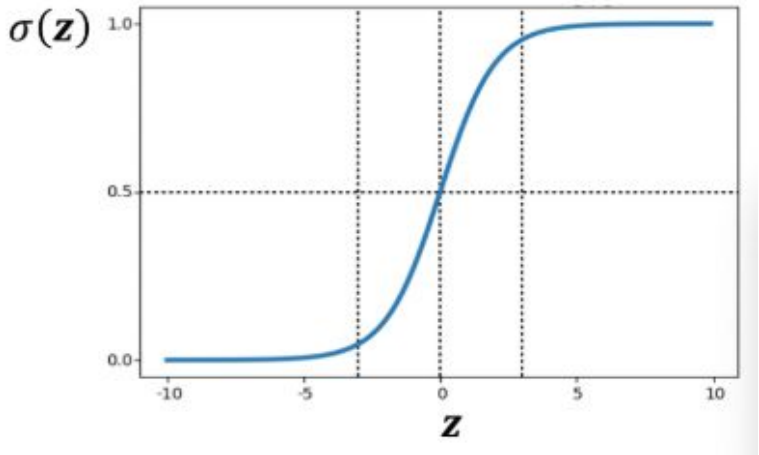
$$\sigma(z) = \frac{1}{1 + e^{-z}} = \frac{e^z}{1 + e^z}$$



Probabilistic Model in Classification: Bernoulli Likelihood

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

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



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

Probabilistic Model in Classification: Bernoulli Likelihood

- Logistic regression model

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

- Note that

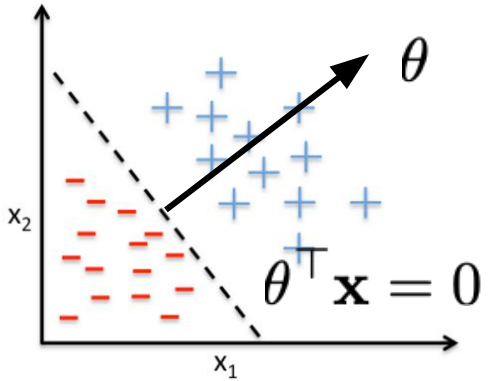
$$p(y = 0|x, \theta) = 1 - \frac{1}{1+\exp(-\theta^\top x)} = \frac{\exp(-\theta^\top x)}{1+\exp(-\theta^\top x)}$$

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 $\theta^\top \mathbf{x} = 0$.



Logistic Regression is a Linear Classifier

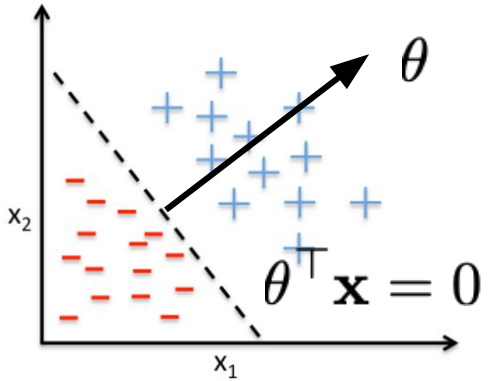
- Decision boundaries for Logistic Regression?
  - At the decision boundary, label 1/0 are equiprobable.

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

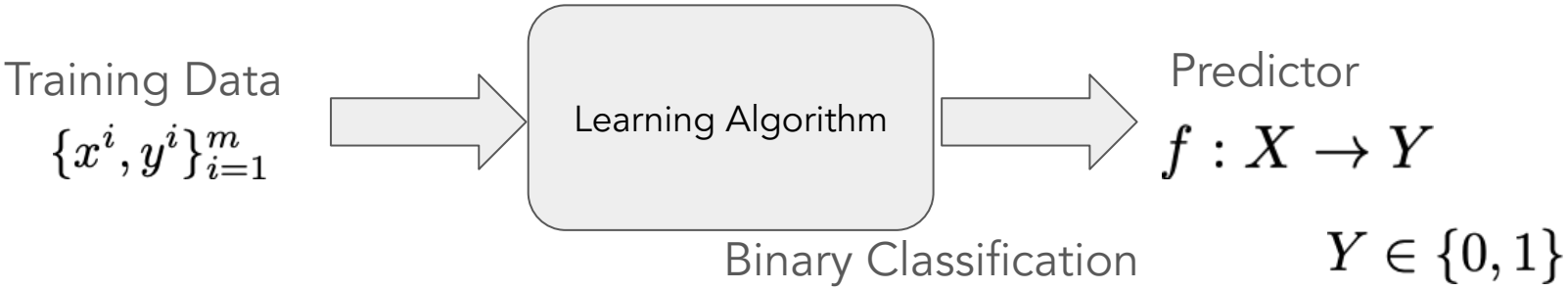
to be equal: $e^{-\theta^\top \mathbf{x}} = e^{\theta^\top \mathbf{x}}$, whose only solution is $\theta^\top \mathbf{x} = 0$.

✓ ⇒ Decision boundary is linear.

✓ ⇒ Logistic regression is a probabilistic linear classifier.



Binary Classification Algorithms



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

$$l(\theta) := \log \prod_{i=1}^{n} p(y^i|x^i,\theta) \qquad \text{(Bernoulli)}$$

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

$$l(\theta) := \log \prod_{i=1}^{n} p(y^i|x^i,\theta) \qquad \text{(Bernoulli)}$$

$$= \sum_{i=1}^{n} \log\left(\frac{\exp(-\theta^\top x^i)}{1+\exp(-\theta^\top x^i)}\right) \underbrace{I(y^i=0)}_{1-y^i} + \log\left(\frac{1}{1+\exp(-\theta^\top x^i)}\right) \underbrace{I(y^i=1)}_{y^i}$$

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

$$l(\theta):=\log\prod_{i=1}^{n}p(y^i|x^i,\theta) \qquad \text{(Bernoulli)}$$

$$=\sum_{i=1}^{n}\log\left(\frac{\exp(-\theta^\top x^i)}{1+\exp(-\theta^\top x^i)}\right)\underbrace{I(y^i=0)}_{1-y^i}+\log\left(\frac{1}{1+\exp(-\theta^\top x^i)}\right)\underbrace{I(y^i=1)}_{y^i}$$

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

MAP

- Logistic regression model

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

- Note that

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

- Prior

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

MAP

- Logistic regression model

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

- Note that

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

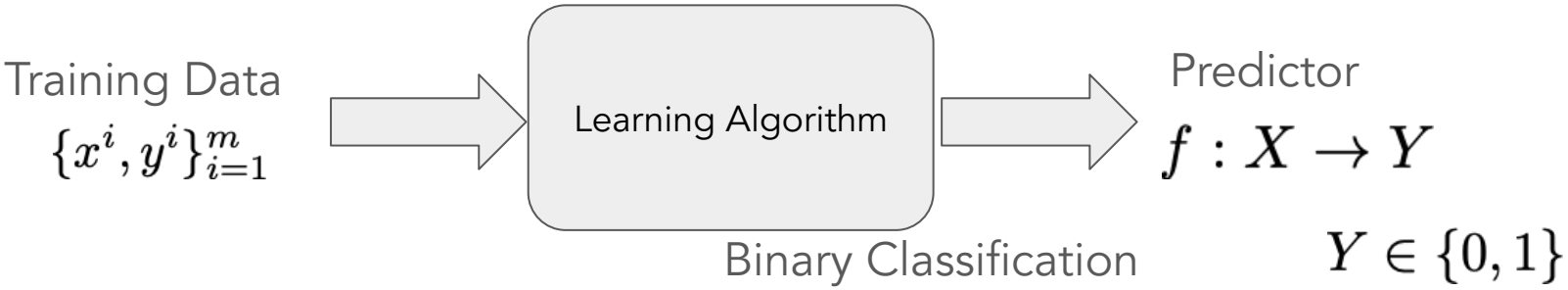
- Prior

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

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

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

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?

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

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

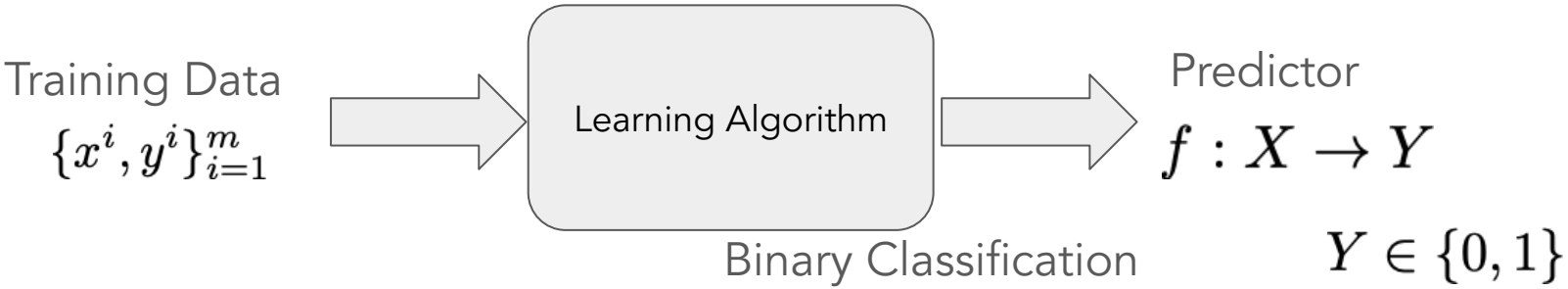
(Stochastic) Gradient Descent

- Initialize parameter θ^0

- Do

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

Binary Classification Algorithms



Logistic Regression Pipeline

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

Q&A