50.007 Machine Learning

2026 Spring

4. Regression

Na Zhao
Assistant Professor, ISTD

SINGAPORE UNIVERSITY OF
TECHNOLOGY AND DESIGN

Recap - Linear Classifier (separable case)

1. Training Set (Linearly Separable)

$$\left(x^{(1)}, y^{(1)}\right), \left(x^{(2)}, y^{(2)}\right), \ldots, \left(x^{(n)}, y^{(n)}\right)$$

2. Model (Linear Classifier)

$$h(x; \theta) = \operatorname{sign}(\theta_1 x_1 + \cdots + \theta_d x_d)$$

3. Training Error (Zero-one Loss)

$$\boldsymbol{\varepsilon}_n(\theta) = \frac{1}{n} \sum_{(x,y) \in \mathcal{S}_n} [\![\, y(\theta^\top x) \leq 0 \,]\!]$$

4. Algorithm (The Perceptron Algorithm)

$$\text{if } y^{(t)} \neq h(x^{(t)}; \theta^{(k)}) \text{ then}$$

$$\theta^{(k+1)} = \theta^{(k)} + y^{(t)} x^{(t)}$$

Recap - Linear Classifier (Non-separable case)

1. Training Set (Not Necessarily Linearly Separable)

$$\left(x^{(1)}, y^{(1)}\right), \left(x^{(2)}, y^{(2)}\right), \ldots, \left(x^{(n)}, y^{(n)}\right)$$

2. Model (Linear Classifier)

$$h(x; \theta) = \operatorname{sign}(\theta_1 x_1 + \cdots + \theta_d x_d)$$

3. Training Error (Hinge Loss)

$$R_n(\theta) = \frac{1}{n} \sum_{(x,y) \in \mathcal{S}_n} \max\{1 - y(\theta^\top x), 0\}$$

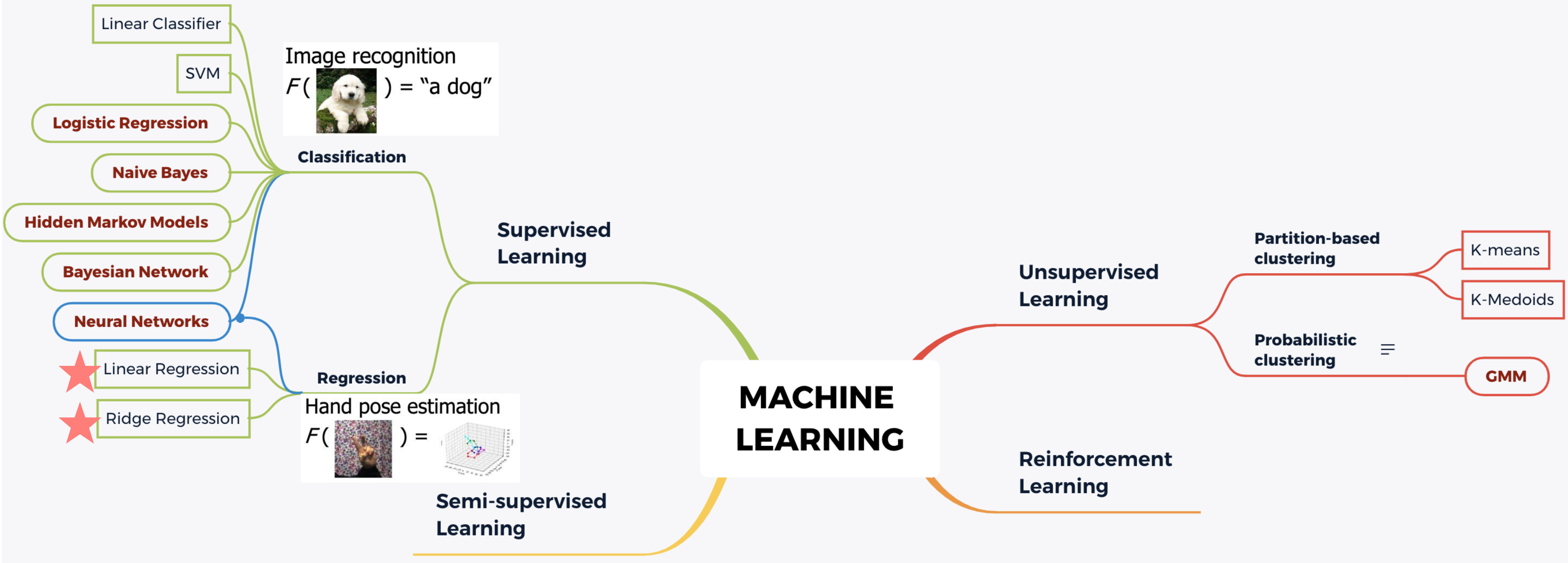
4. Algorithm (Stochastic Sub-Gradient Descent)

$$\text{select } t \in \{1, \ldots, n\} \text{ at random,}$$

$$\text{if } y^{(t)}(\theta^{(k)} \cdot x^{(t)}) \leq 1, \text{ then}$$

$$\theta^{(k+1)} = \theta^{(k)} + \eta_k y^{(t)} x^{(t)}$$

Roadmap



Linear Classifier
SVM
Logistic Regression
Naive Bayes
Hidden Markov Models
Bayesian Network
Neural Networks
Linear Regression
Ridge Regression
Image recognition
F( ) = "a dog"
Classification
Regression
Hand pose estimation
F( ) =
Supervised Learning
Semi-supervised Learning
MACHINE LEARNING
Unsupervised Learning
Partition-based clustering
K-means
K-Medoids
Probabilistic clustering
GMM
Reinforcement Learning

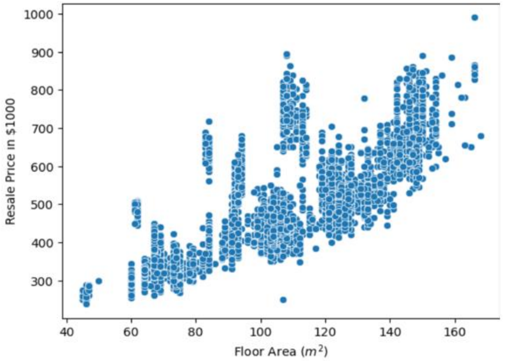
Learning Objectives

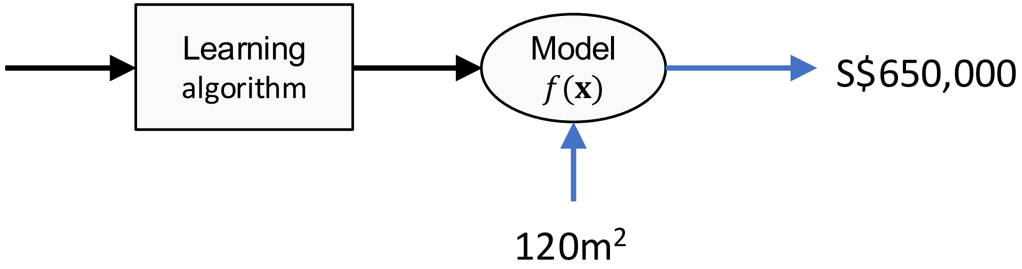
You should be able to know:

1. What is linear regression?
2. How to learn a linear regression model?
3. What is ridge regression and how is ridge regression different from linear regression?
4. What is regularization and why do we need to do regularization?

Regression Examples

- **House Price Prediction**





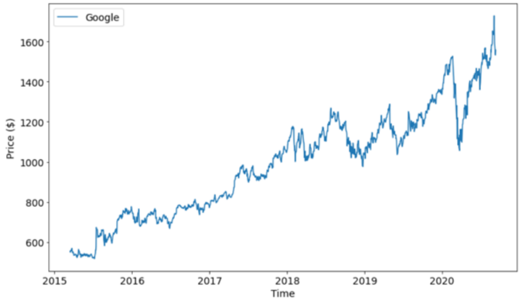
Learning algorithm

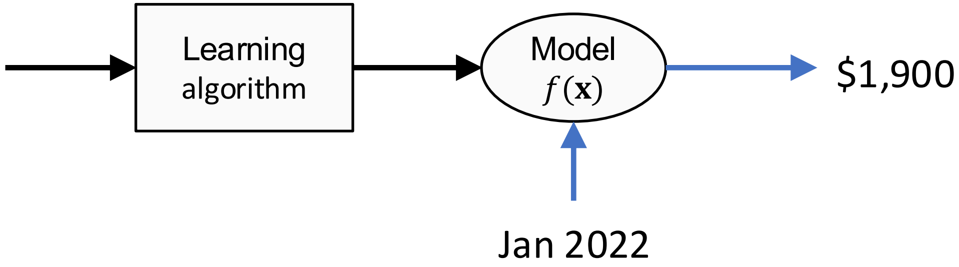
Model $f(\mathbf{x})$

S$650,000

120m^2

- **Stock Price Prediction**





Learning algorithm

Model $f(\mathbf{x})$

$1,900

Jan 2022

Learning a function

$$y = f(x)$$

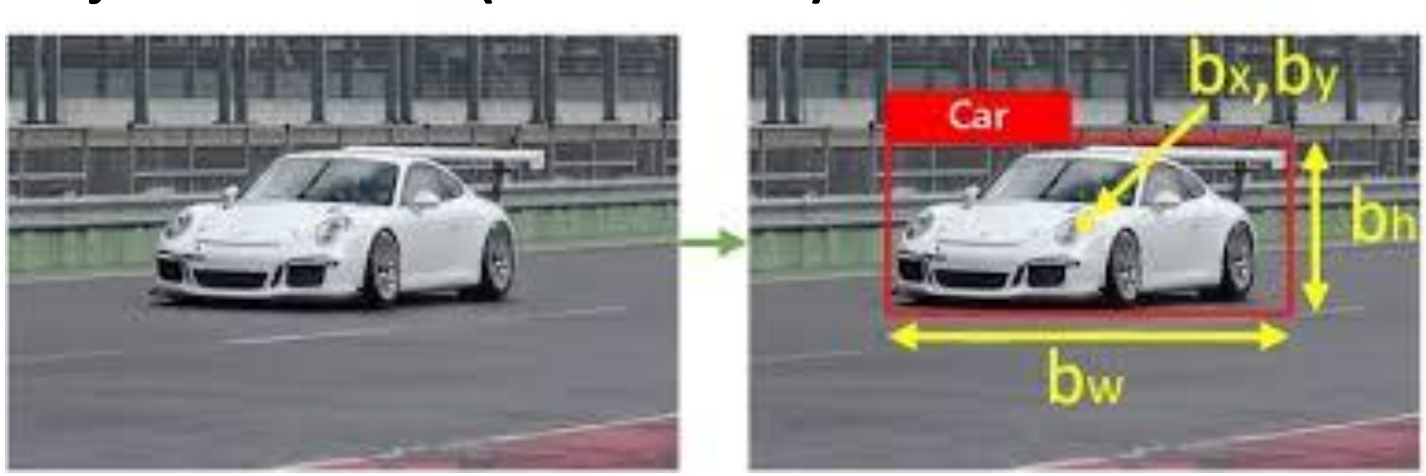
$$x \in \mathbb{R}^d$$

$$y \in \mathbb{R}^m$$

m=1 in these two examples

Regression Examples

- **Object Detection (Localization)**

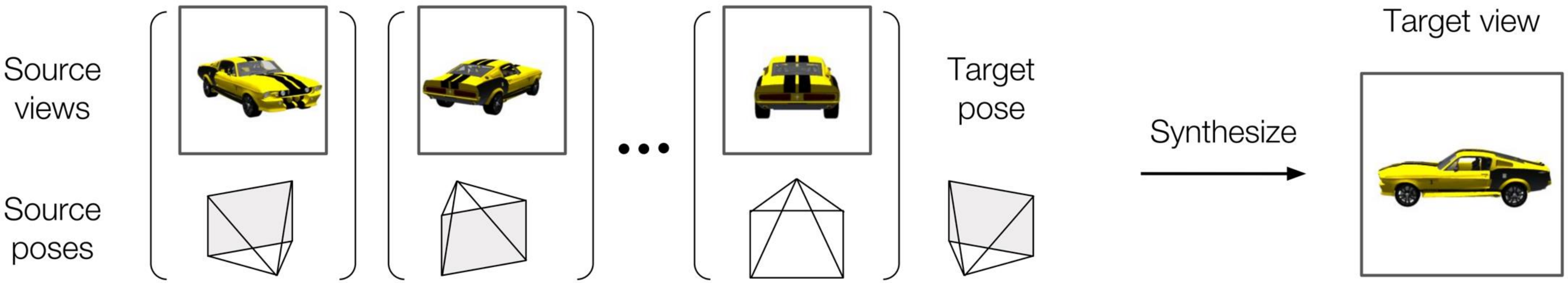


Learning a function

$$y = f(x)$$
$$x \in \mathbb{R}^d$$
$$y \in \mathbb{R}^m$$

m>1 in these two examples

- **Novel View Synthesis**



Linear Regression

- A **linear regression** function is simply a linear function of the feature vectors:

$$f(x;\theta,\theta_0)=\theta\cdot x+\theta_0=\sum_{i=1}^{d}\theta_i x_i+\theta_0$$

training sample (fixed)

- Similar as in linear classification, different parameter choices $\theta\in\mathbb{R}^d$, $\theta_0\in\mathbb{R}$, give rise to the **hypothesis set** $\mathcal{F}$.

determined by x & θ

- **Training Objective**: find $f(x;\hat{\theta},\hat{\theta}_0)$ that would yield accurate predictions on yet unseen examples based on $\mathcal{S}_n$.

Training data

$$\mathcal{S}_n=\{\,(x^{(t)},y^{(t)})\,|\,t=1,\ldots,n\,\}$$

- Features/Inputs $x^{(t)}=\left(x_1^{(t)},\ldots,x_d^{(t)}\right)^{\mathsf{T}}\in\mathbb{R}^d$
- Response/Output $y^{(t)}\in\mathbb{R}$

How does it differ from the linear classification function?

shows difference btw true label & predicted label

Linear Regression – Cost Function

- Similar to the classification setting, we will measure training error in terms of **empirical risk**:

$$\theta' = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_n \end{bmatrix}, \quad x' = \begin{bmatrix} x'_1 \\ \vdots \\ x_n \end{bmatrix}$$

$$R_n(\theta) = \frac{1}{n}\sum_{t=1}^{\overset{\text{n training sample}}{n}} \text{Loss}(\overset{\text{true label}}{y^{(t)}} - \underbrace{\overset{\text{predicted label}}{\theta \cdot x^{(t)}}}_{\theta' \; x^{(t)},})$$

Note θ_0 is dropped here for simplicity, remember you can always add it back.

- Cost/loss function: "**Least Squares**" criterion

why putting square here?

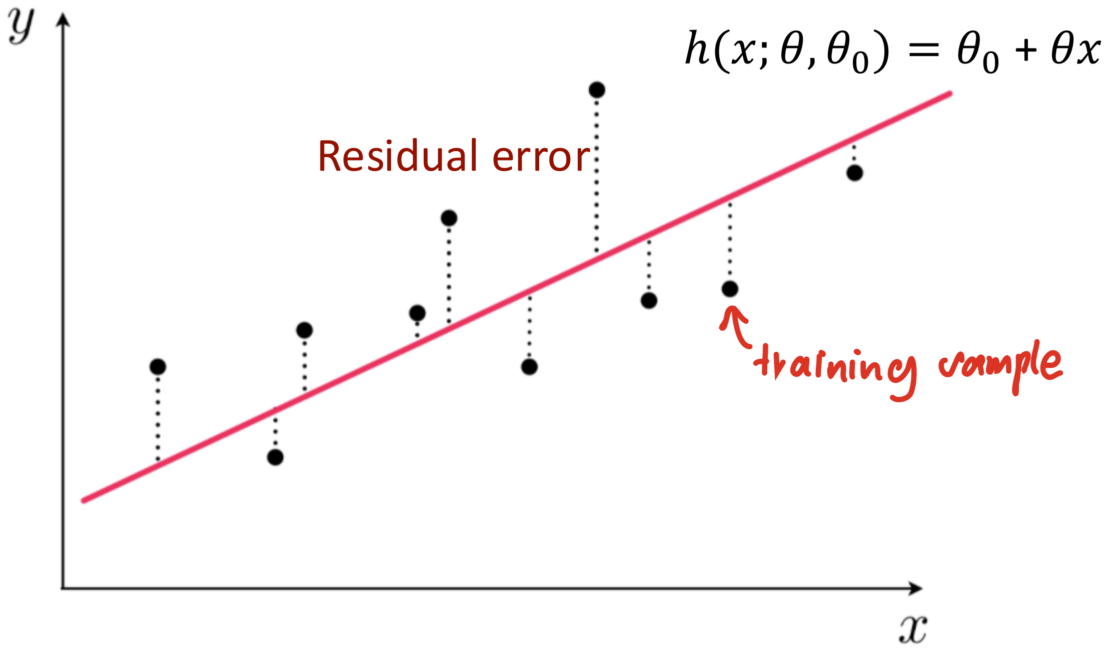
$$R_n(\theta) = \frac{1}{n}\sum_{t=1}^{n} \text{Loss}(y^{(t)} - \theta \cdot x^{(t)}) = \frac{1}{n}\sum_{t=1}^{n} (y^{(t)} - \theta \cdot x^{(t)})^{\boxed{2}} / \boxed{2}$$

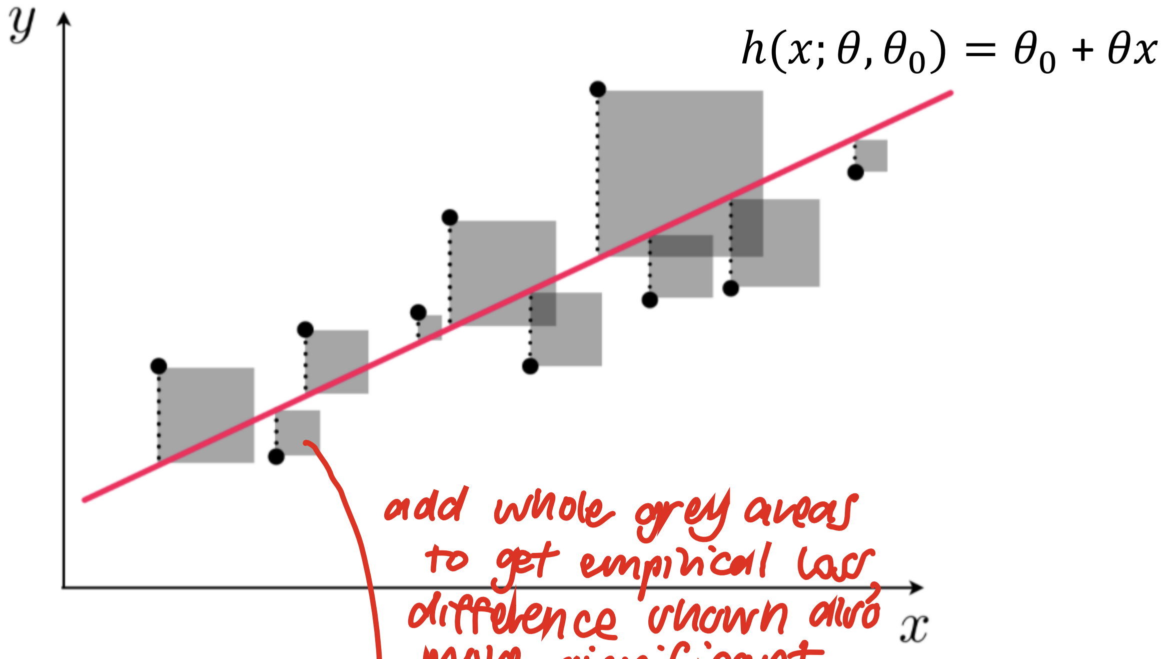
why is the result divided by 2?

- The square is to avoid cancellation due to positive and negative differences
- And it also can permit small discrepancies but heavily penalize large deviations

difference x be >1

Cost Function - Graphical Interpretation





add whole grey areas to get empirical loss, difference shown also more significant

$$R_n(\theta) = \frac{1}{n}\sum_{t=1}^{n}\Big(\overset{\text{true label}}{y^{(t)}} - \overset{\text{prediction score}}{h\big(x^{(t)};\theta,\theta_0\big)}\Big)^2/2$$

Linear Regression – Optimization

iterative way to optimise linear regression

- **Stochastic Gradient Descent**

$\theta^{(0)} = 0$ (vector) *# Random initialization*

select $t \in \{1, \ldots, n\}$ at random,

difference

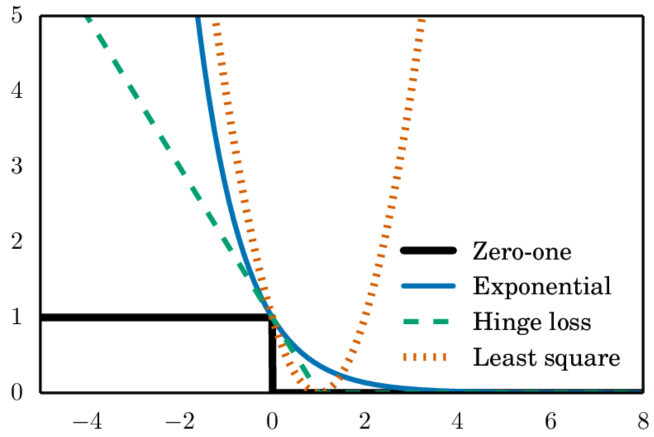
$$\theta^{(k+1)} = \theta^{(k)} - \eta_k \nabla_\theta (y^{(t)} - \theta \cdot x^{(t)})^2/2 \,|_{\theta=\theta^{(k)}}$$ *# Update*

Repeat the update step until stopping criterion is met.

$$\nabla_\theta (y^{(t)} - \theta \cdot x^{(t)})^2/2 = (y^{(t)} - \theta \cdot x^{(t)}) \nabla_\theta (y^{(t)} - \theta \cdot x^{(t)})$$

$$= -(y^{(t)} - \theta \cdot x^{(t)}) x^{(t)}$$

$$\boxed{\theta^{(k+1)} = \theta^{(k)} + \eta_k (y^{(t)} - \theta \cdot x^{(t)}) x^{(t)}}$$



$$R_n(\theta) = \frac{1}{n} \sum_{t=1}^{n} (y^{(t)} - \theta \cdot x^{(t)})^2/2$$

- *Convex!*
- *Everywhere differentiable!*

Linear Regression – Optimization

- **Closed Form Solution**
  - **Find $\hat{\theta}$**: minimizing empirical risk directly by **setting gradient to zero**.

$$\nabla R_n(\theta)_{\theta=\hat{\theta}} = \frac{1}{n}\sum_{t=1}^{n} \nabla_\theta \left\{(y^{(t)} - \theta \cdot x^{(t)})^2/2\right\}_{|\theta=\hat{\theta}}$$

gradient

$$= \frac{1}{n}\sum_{t=1}^{n} \{-(y^{(t)} - \hat{\theta} \cdot x^{(t)})x^{(t)}\}$$

$$= -\frac{1}{n}\sum_{t=1}^{n} y^{(t)}x^{(t)} + \frac{1}{n}\sum_{t=1}^{n} (\hat{\theta} \cdot x^{(t)})x^{(t)} \qquad \boxed{\hat{\theta} \cdot x^{(t)} = (x^{(t)})^T\hat{\theta}}$$

optimized

$$= \underbrace{-\frac{1}{n}\sum_{t=1}^{n} y^{(t)}x^{(t)}}_{=b} + \underbrace{\frac{1}{n}\sum_{t=1}^{n} x^{(t)}(x^{(t)})^T}_{=A}\,\hat{\theta} \;=\; -b + A\hat{\theta}$$

Linear Regression – Optimization

- **Closed Form Solution**
  - **Find** $\hat{\theta}$: minimizing empirical risk directly by **setting gradient to zero**.

$$\nabla R_n(\theta)_{\theta=\hat{\theta}} = -\underbrace{\frac{1}{n}\sum_{t=1}^{n} y^{(t)} x^{(t)}}_{=b} + \underbrace{\frac{1}{n}\sum_{t=1}^{n} x^{(t)}(x^{(t)})^T}_{=A}\hat{\theta} \quad = -b + A\hat{\theta} = 0$$

$$X = \begin{bmatrix} x^1 \\ x^2 \\ \vdots \\ x^n \end{bmatrix}$$

  - If A is **invertible** ($n \geq d$), we can find the *optimal* parameter $\hat{\theta}$ as:

$$\overset{\in \mathbb{R}^d}{\hat{\theta}} = A^{-1}b.$$

where, $\underset{\in \mathbb{R}^d}{b = \frac{1}{n}X^T\vec{y}}$, $\quad \underset{\in \mathbb{R}^{d\times d}}{A = \frac{1}{n}X^T X}$

$$X = [x^{(1)}, \ldots, x^{(n)}]^T \in \mathbb{R}^{n\times d}$$
$$\vec{y} = [y^{(1)}, \ldots, y^{(n)}]^T \in \mathbb{R}^{n}$$

  - **Note**: The time complexity of inverting A is $\mathcal{O}(d^3)$. It would be costly if d is large.

What can go wrong in Linear Regression?

- $n < d$ (A is *not* invertible)
  - It is not unusual to see the number of input variables greatly exceed the number of observations, *e.g.* micro-array data analysis, environmental pollution studies.

- Overfitting
  - The training objective is just about fitting the data as well as possible, but we might also want to keep the hypothesis from getting *too* attached to the training data, as this may hinder its generalization to unseen data.

☞ Regularizing parameters*!*

Ridge Regression

- The cost/loss function of **ridge regression**:

$$J_{n,\lambda}(\theta) = \frac{\lambda}{2}\left\|\theta\right\|^2 + R_n(\theta) = \boxed{\frac{\lambda}{2}\left\|\theta\right\|^2} + \boxed{\frac{1}{n}\sum_{t=1}^{n}(y^{(t)} - \theta \cdot x^{(t)})^2/2}$$

Pressure to simplify model

Pressure to fit data

  - λ > 0 is the regularization parameter
  - It quantifies the *trade-off* between keeping the parameters small and fitting to the training data.

  - $||\theta||^2$ is also known as ***L*₂ regularization**, it can be used in many contexts aside from linear regression, *e.g.,* logistic regression and support vector machines.

Ridge Regression – Optimization

$$\boxed{\nabla_\theta \left\{ \frac{\lambda}{2} \|\theta\|^2 + (y^{(t)} - \theta \cdot x^{(t)})^2/2 \right\}_{|\theta=\theta^{(k)}} = \lambda\theta^{(k)} - (y^{(t)} - \theta^{(k)} \cdot x^{(t)})x^{(t)}}$$

Linear Regression

$$\boxed{\nabla_\theta (y^{(t)} - \theta \cdot x^{(t)})^2/2 = -(y^{(t)} - \theta \cdot x^{(t)})x^{(t)}}$$

- **Stochastic Gradient Descent**

```
$\theta^{(0)} = 0$ (vector)          # Random initialization
select $t \in \{1, \ldots, n\}$ at random,
    $\theta^{(k+1)} = \boxed{(1 - \lambda\eta_k)}\theta^{(k)} + \eta_k(y^{(t)} - \theta \cdot x^{(t)})x^{(t)}$     # Update
Repeat the update step until stopping criterion is met.
```

$$\theta^{k+1} = \theta^k - \eta_k \nabla_\theta L_r$$

$$\eta_k (\lambda\theta^k + \nabla_\theta L_i)$$

- **Closed Form Solution**

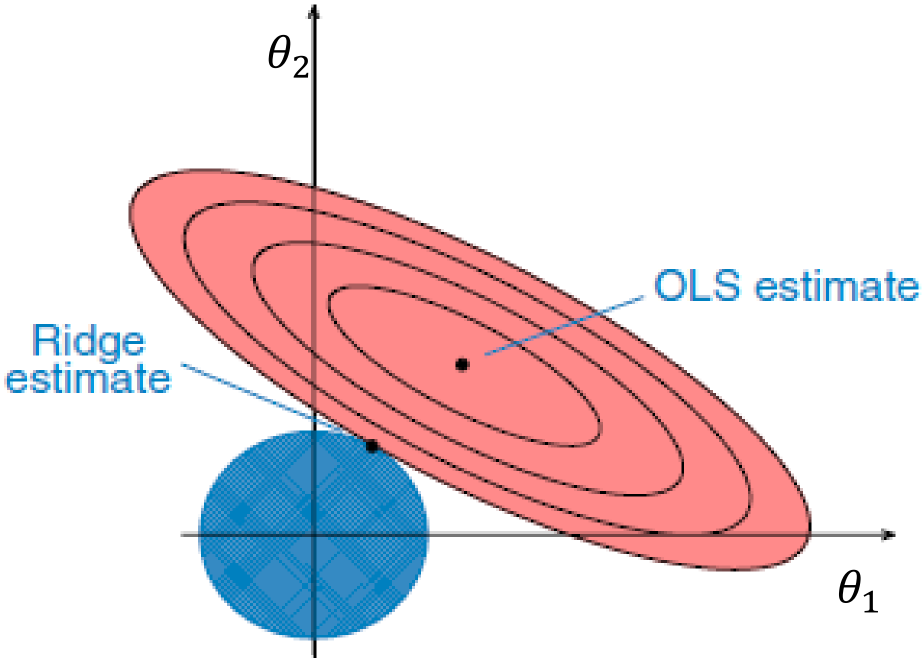
$$\nabla \underbrace{\frac{\lambda}{2} \|\theta\|^2}_{\text{regularisation term}} + R_n(\theta) = -\underbrace{\frac{1}{n}\sum_{t=1}^{n} y^{(t)}x^{(t)}}_{=b} + \underbrace{\frac{1}{n}\sum_{t=1}^{n} (\lambda + x^{(t)}(x^{(t)})^T)}_{=A}\hat{\theta} = -b + A\hat{\theta} = 0$$

$\boxed{A = \lambda I + (1/n)X^T X}$ In ridge regression (<u>$\lambda > 0$</u>), A is *always* invertible.

Ridge Regression - Geometric Interpretation

- The ellipses correspond to the contours of residual sum of squares (RSS)
  - The inner ellipse has smaller RSS
  - RSS is minimized at ordinal least square (OLS) estimates.
- The circle corresponds to the L2 regularization.
- In **ridge regression**, we are trying to minimize the ellipse size and circle *simultaneously*.
  - The ridge estimate ($\hat{\theta}$) is given by the point at which the ellipse and the circle touch.
- The **trade-off** between the penalty (L2) term and RSS:
  - A large θ would give you a better RSS but then it will push the penalty term higher.
  - While a smaller θ would a worse RSS.

$$\boxed{J_{n,\lambda}(\theta) = \frac{\lambda}{2}\|\theta\|^2 + \frac{1}{n}\sum_{t=1}^{n}(y^{(t)} - \theta \cdot x^{(t)})^2/2}$$

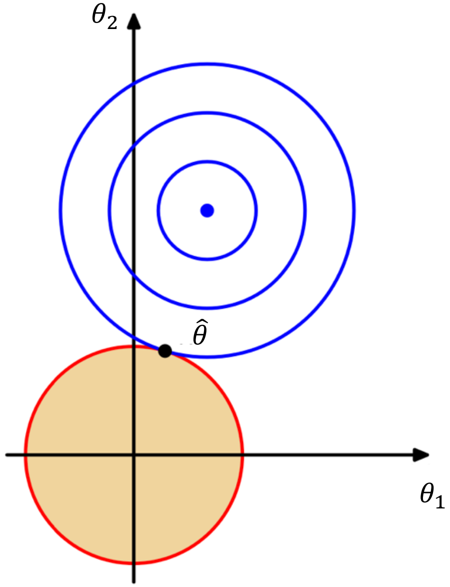


More General Regularizer

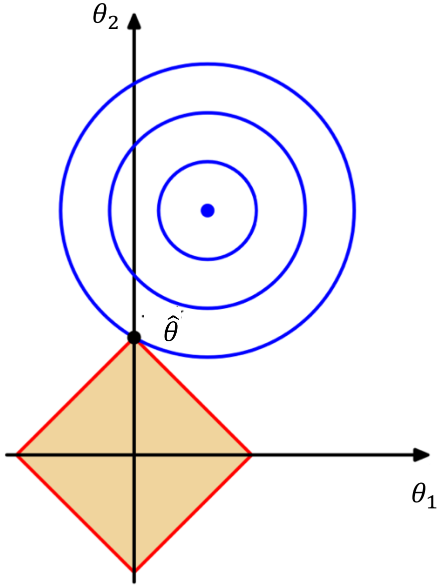
- Allow the regularization term to be raised to different powers p:

$$J_{n,\lambda}(\theta) = \frac{\lambda}{2}\,|\theta|^p + R_n(\theta)$$

- $p = 2$ → **Ridge Regression**
- $p = 1$ → **Lasso Regression**
  - *a.k.a.* **L1 Regularization**
  - if λ is sufficiently large, some of the weights θ_i are driven to zero, leading to a *sparse* model.
  - can be used for **feature selection**.

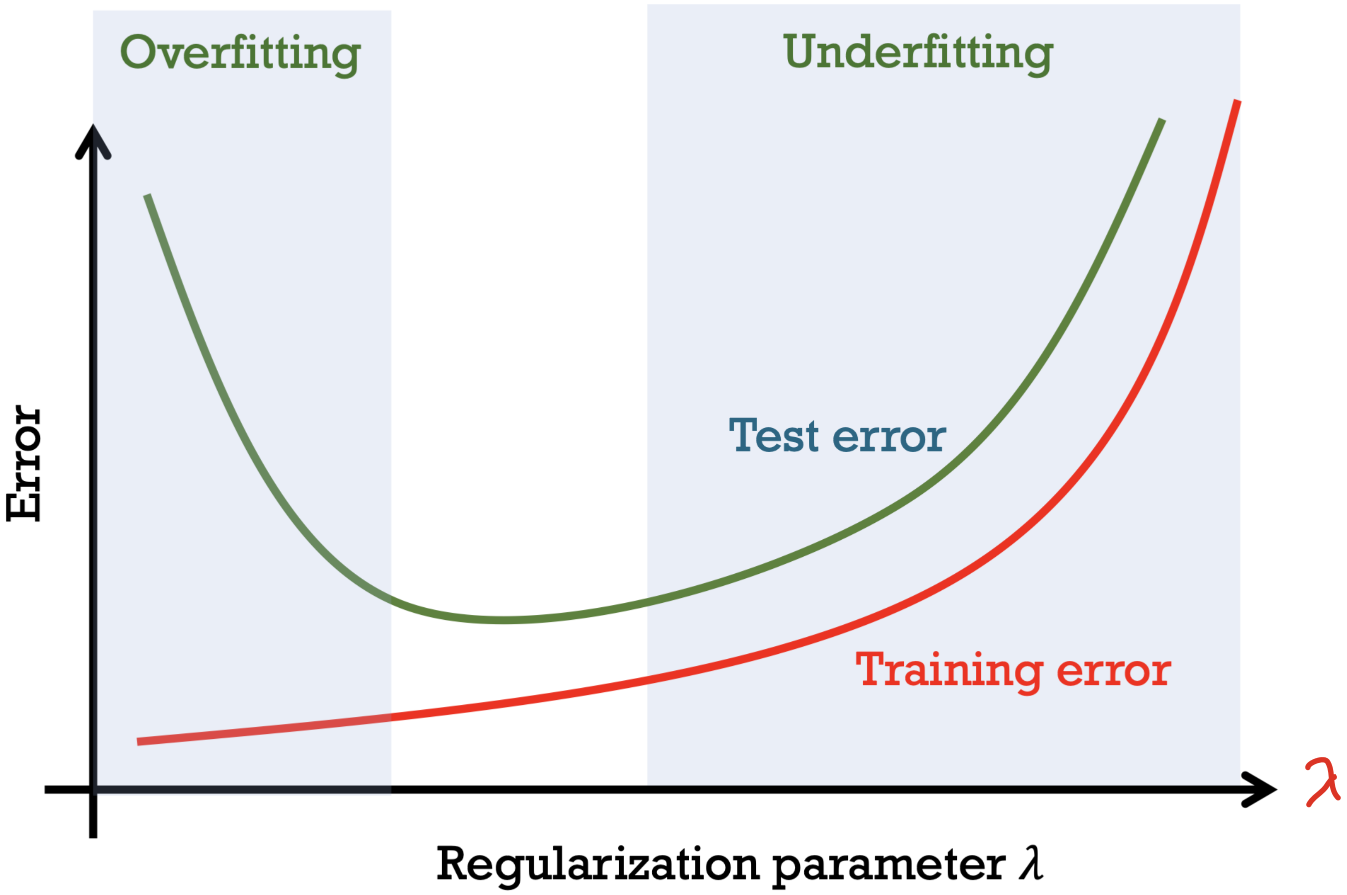


Ridge Regression



Lasso Regression

Effect of Regularization

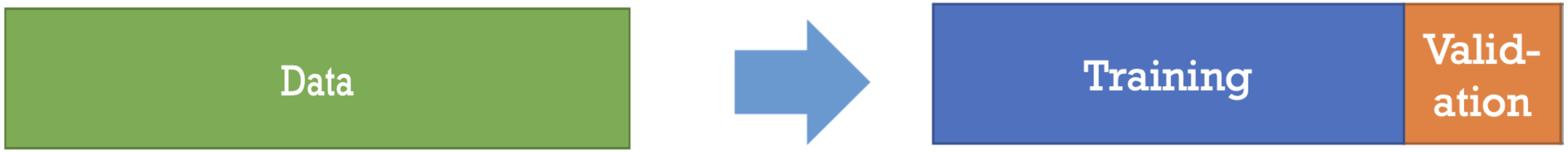


How to pick λ?

- The regularization parameter λ is an example of a *hyperparameter*, which affects the model complexity.

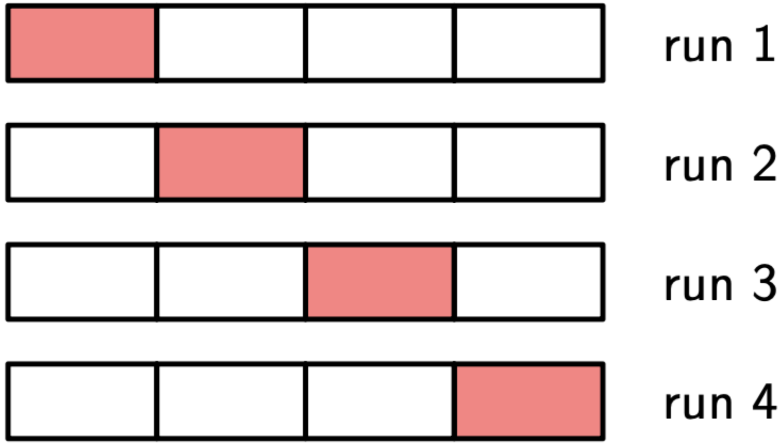
- We don't usually have access to the test data. How to pick optimal λ that minimizes the test error?

- The solution is to create a ***validation*** data set, as a proxy to the test data, and to compute the *validation error*.



Cross-Validation

- Dividing the dataset into a fixed training and validation sets can be *problematic* if the amount of training data is **limited**:
  - We need as much of the available data as possible for training to build good models.
  - The small validation set will give a relatively noisy estimate of predictive performance.
- **Cross-Validation** is one solution for this dilemma.
  - **k-fold cross-validation**
  - Use a proportion $(k - 1)/k$ of the available data for training
  - Leave-one-out for validation (assess performance)
  - Repeated for all k possible choices for the held-out group, then average the performance scores from the k runs.



4-fold cross-validation

k = no. of folds

Summary

1. What is linear regression?

$$\boxed{f(x;\theta,\theta_0) = \theta \cdot x + \theta_0 = \sum_{i=1}^{d} \theta_i x_i + \theta_0}$$

2. How to learn a linear regression model?

$$\boxed{R_n(\theta) = \frac{1}{n}\sum_{t=1}^{n} \text{Loss}(y^{(t)} - \theta \cdot x^{(t)}) = \frac{1}{n}\sum_{t=1}^{n}(y^{(t)} - \theta \cdot x^{(t)})^2/2}$$

Can optimize the cost function using either SGD or closed form solution.

3. What is ridge regression and how is ridge regression different from linear regression?

Add a L2 regularization term to the linear regression cost function.

4. What is regularization and why do we need to do regularization?

Regularization adds constraint on the parameter values to prevent the model overfitting to the training data.

The main aim of regularization is to reduce the over-complexity of the models and help the model learn a simpler function to promote generalization.

Acknowledgements

- Some slides and content of this lecture are adopted from:
  - MIT 6.036 Introduction to Machine Learning
  - SUTD 50.007 Machine Learning, Spring 2023 (Asst Prof. Malika Meghjani)
  - PennState STAT 897D Applied Data Mining and Statistical Learning
  - Bishop, C. M. (2006). Pattern recognition and machine learning. Springer. (Chapter 3)