SOI1010
Machine Learning II

Lecture 3: Linear Regression

Department of Data Science
Hanyang University

What is Machine Learning?

- "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T as measured by P, improves with experience E."

------- Machine Learning, Tom Mitchell, 1997

Example 1: Image classification

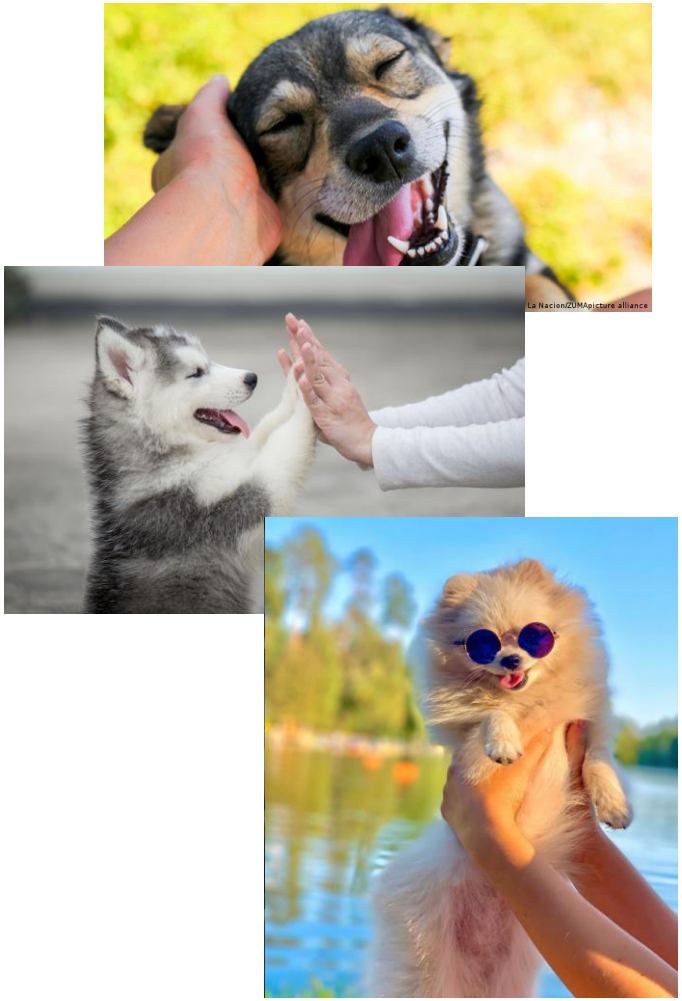
cat

- Task: determine if the image is cat or not
- Performance measure: probability of misclassification (error rate)

Example 1: Image classification

cats

Experience/Data:
Images with labels



dogs

Example 1: image classification

- A few terminologies
  - Training data: the images given for learning
  - Validation data: for model selection
  - Test data: the images to be classified (for evaluation)
  - Binary classification: classify into two classes
    - E.g., dogs vs cats
  - Multiclass classification: classify into three or more classes

Performance measure

- Specific to task T
- Evaluated using datasets
  - Training/validation/test sets
- Often challenging to choose
  1. Difficult to decide what to measure
  2. Know ideal measure but measurement could be impractical

Central challenge in ML

- Generalization
  - ➢ Ability to perform well on previously unseen examples
- Generalization error
  - ➢ Expected error on new examples ⇒ implausible to calculate
- Training error
  - ➢ Measured on a training set ⇒ bad proxy for generalization error
- Test error
  - ➢ Measured on a test set (not used in training)

  ⇒ better proxy for generalization error

Example 2: clustering images

- Task: partition the images into 2 groups
- Performance: similarities within groups
- Data: a set of images

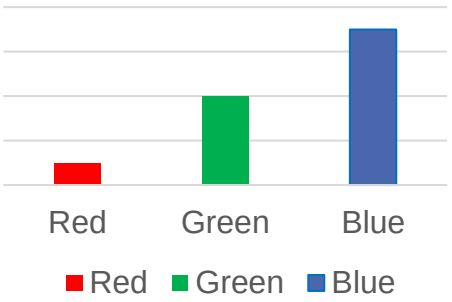
Example 2: clustering images

- A few terminologies
  - Unlabeled data vs labeled data
  - Unsupervised learning vs supervised learning
- Examples of unsupervised learning
  - Principal component analysis / dimensionality reduction (e.g., PCA)
  - Cluster analysis (e.g., k-means)
- Examples of supervised learning
  - SVM, Linear regression, Logistic regression, k-NN, neural networks
  - Naïve Bayes, Linear Discriminant Analysis, Decision trees

Math formulation

Extract features

Feature vector: $\boldsymbol{x}^{(i)}$



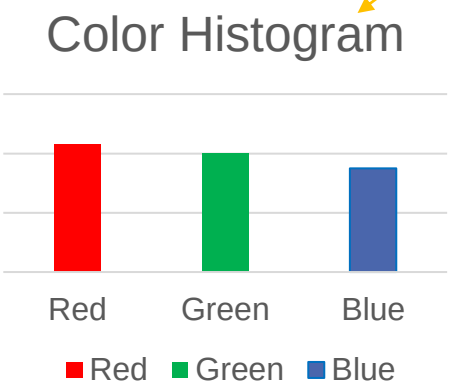
Outdoor

Label: $y^{(i)}$

1

Math formulation

Extract features



Feature vector: $\boldsymbol{x}^{(j)}$

Indoor

0

Label: $y^{(j)}$

Math formulation

- Given training data $\{(\boldsymbol{x}^{(i)}, y^{(i)}) : 1 \leq i \leq n\}$
- Find $y = f(\boldsymbol{x})$ using training data
- s.t. f produces correct results on new/unseen test data

But.. What kind of functions?

Math formulation

- Given training data $\{(\boldsymbol{x}^{(i)}, y^{(i)}) : 1 \leq i \leq n\}$
- Find $y = f(\boldsymbol{x}) \in \mathcal{H}$ using training data
- s.t. f produces correct results on new/unseen test data

Hypothesis class

Math formulation

- Given training data $\{(\boldsymbol{x}^{(i)}, y^{(i)}) : 1 \leq i \leq n\}$
- Find $y = f(\boldsymbol{x}) \in \mathcal{H}$ using training data
- s.t. f produces correct results on new/unseen test data

Connection between training and test data?

Math formulation

- Given training data $\{(\boldsymbol{x}^{(i)}, y^{(i)}) : 1 \leq i \leq n\}$ i.i.d. from distribution $\mathcal{D}$
- Find $y = f(\boldsymbol{x}) \in \mathcal{H}$ using training data
- s.t. f produces correct results on new/unseen test data i.i.d. from distribution $\mathcal{D}$

Connection between training and test data?

i.i.d.: independently identically distributed

Math formulation

- Given training data $\{(\boldsymbol{x}^{(i)}, y^{(i)}) : 1 \leq i \leq n\}$ i.i.d. from distribution $\mathcal{D}$
- Find $y = f(\boldsymbol{x}) \in \mathcal{H}$ using training data
- s.t. f produces correct results on new/unseen test data i.i.d. from distribution $\mathcal{D}$

What kind of performance measure?

Math formulation

- Given training data $\{(\boldsymbol{x}^{(i)}, y^{(i)}) : 1 \leq i \leq n\}$ i.i.d. from distribution $\mathcal{D}$
- Find $y = f(\boldsymbol{x}) \in \mathcal{H}$ using training data
- s.t. the expected loss is small

$$\mathcal{L}(f) = \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}[l(f, \boldsymbol{x}, y)]$$

Various loss functions

Math formulation

- Given training data $\{(\boldsymbol{x}^{(i)}, y^{(i)}) : 1 \leq i \leq n\}$ i.i.d. from distribution $\mathcal{D}$
- Find $y = f(\boldsymbol{x}) \in \mathcal{H}$ using training data
- s.t. the expected loss is small

$$\mathcal{L}(f) = \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}[l(f, \boldsymbol{x}, y)]$$

- Examples of loss functions:
  - 0-1 loss
    - $l(f, \boldsymbol{x}, y) = \mathbb{I}[f(\boldsymbol{x}) \neq y]$
    - $\mathcal{L}(f) = P[f(\boldsymbol{x}) \neq y]$
  - l_2 loss
    - $l(f, \boldsymbol{x}, y) = [f(\boldsymbol{x}) - y]^2$
    - $\mathcal{L}(f) = \mathbb{E}[f(\boldsymbol{x}) - y]^2$

Math formulation

- Given training data $\{(\boldsymbol{x}^{(i)}, y^{(i)}) : 1 \leq i \leq n\}$ i.i.d. from distribution $\mathcal{D}$
- Find $y = f(\boldsymbol{x}) \in \mathcal{H}$ using training data
- s.t. the expected loss is small

$$\mathcal{L}(f) = \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}[l(f, \boldsymbol{x}, y)]$$

How to use?

Math formulation

- Given training data $\{(\boldsymbol{x}^{(i)}, y^{(i)}) : 1 \leq i \leq n\}$ i.i.d. from distribution $\mathcal{D}$
- Find $y = f(\boldsymbol{x}) \in \mathcal{H}$ that minimizes $\hat{\mathcal{L}}(f) = \frac{1}{n} \sum_{i=1}^{n} l(f, \boldsymbol{x}_i, y_i)$
- s.t. the expected loss is small

$$\mathcal{L}(f) = \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}[l(f, \boldsymbol{x}, y)]$$

Empricial loss

Machine Learning 1-2-3

- Collect data and extract features
- Build model: choose hypothesis class $\mathcal{H}$ and loss function l
- Optimization: minimize the empirical loss

Wait…

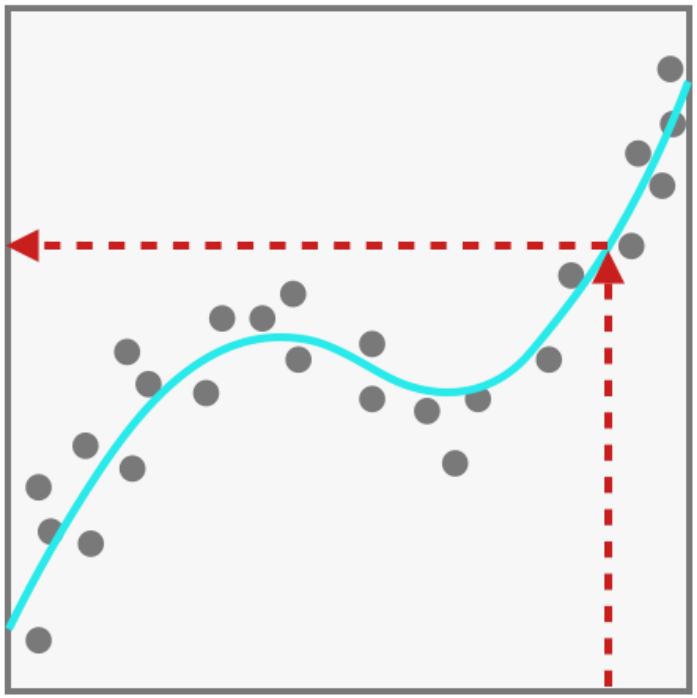
- Does Machine Learning-1-2-3 include all approaches?
  - Include many but not all
  - For now, we will focus on Machine Learning-1-2-3

Wait…

- We can use prior knowledge to design suitable features
- But.. why handcraft the feature vectors?
- Can computers learn the features on the raw images?
  - Learn features directly on the raw images: Representation Learning
  - Deep Learning ⊆ Representation Learning ⊆ Machine Learning ⊆ Artificial Intelligence
- We could use prior knowledge to facilitate representation learning

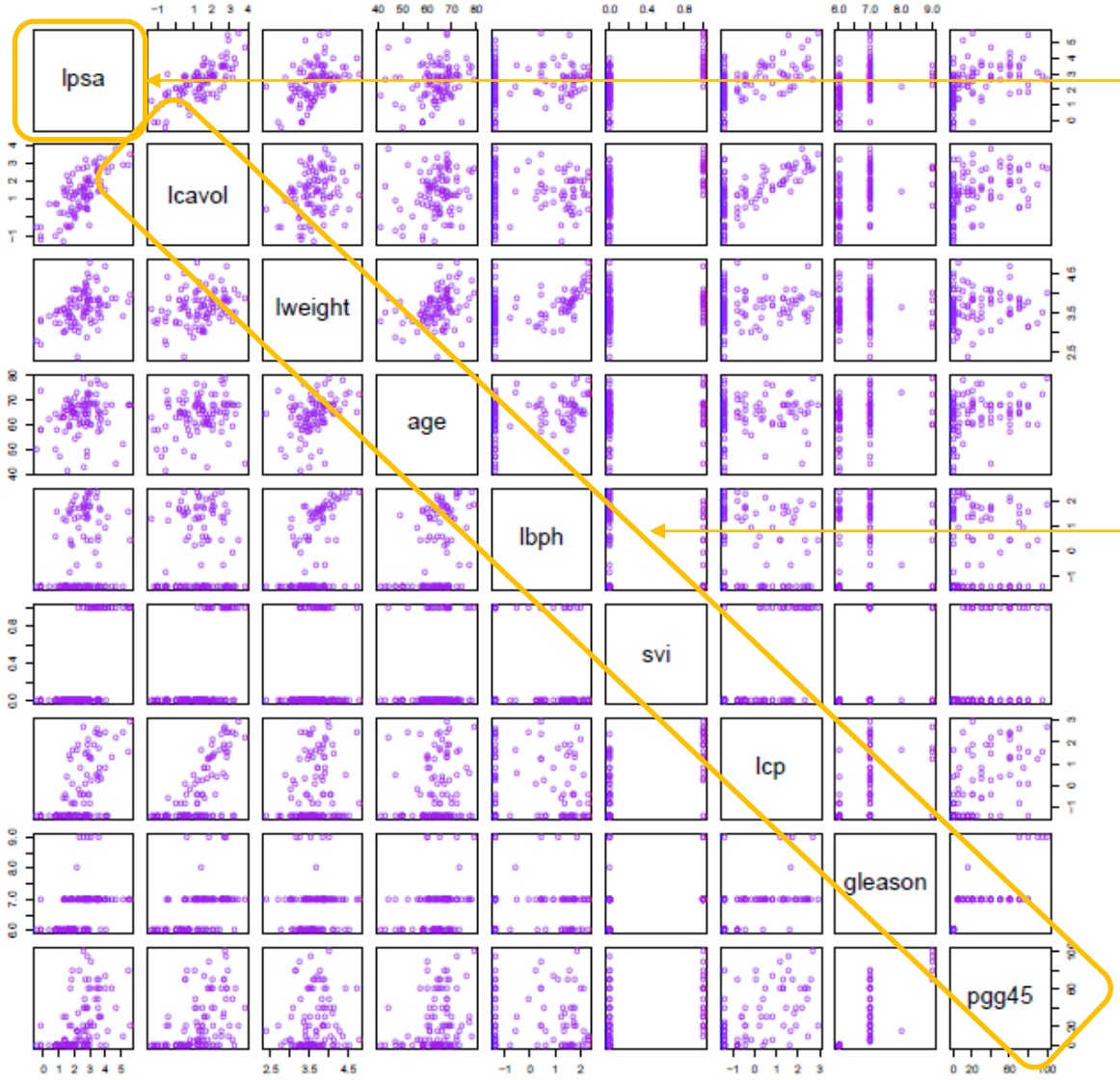
Classification vs Regression

Classification



Regression

Linear regression



y: size/extent of cancer

$\boldsymbol{x} = (x_1, \ldots, x_8)$: clinical measures

Linear regression: Math formulation

- Given training data $\{(\boldsymbol{x}^{(i)}, y^{(i)}) : 1 \leq i \leq n\}$ i.i.d. from distribution $\mathcal{D}$
- Find $f_{\boldsymbol{w}}(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{x}$ that minimizes $\hat{\mathcal{L}}(f_{\boldsymbol{w}}) = \frac{1}{n} \sum_{i=1}^{n} (\boldsymbol{w}^T \boldsymbol{x}^{(i)} - y^{(i)})^2$

Hypothesis class $\mathcal{H}$

l_2 loss; a.k.a.
mean squared error (MSE)

Linear regression: Math formulation

- Given training data $\{(\boldsymbol{x}^{(i)}, y^{(i)}) : 1 \leq i \leq n\}$ i.i.d. from distribution $\mathcal{D}$
- Find $f_{\boldsymbol{w}}(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{x}$ that minimizes $\hat{\mathcal{L}}(f_{\boldsymbol{w}}) = \frac{1}{n} \sum_{i=1}^{n} (\boldsymbol{w}^T \boldsymbol{x}^{(i)} - y^{(i)})^2$

- Let $\boldsymbol{X}$ be a matrix whose i-th row is $\boldsymbol{x}_i^T$.
- And let $\boldsymbol{y}$ be the vector $(y_1, \ldots, y_n)^T$

$$\hat{\mathcal{L}}(f_{\boldsymbol{w}}) = \frac{1}{n} \sum_{i=1}^{n} (\boldsymbol{w}^T \boldsymbol{x}^{(i)} - y^{(i)})^2 = \frac{1}{n} \|\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y}\|_2^2$$

Linear regression: Math formulation & optimization

- Set the gradient to 0 to get the minimizer

$$\nabla_{\boldsymbol{w}} \hat{\mathcal{L}}(f_{\boldsymbol{w}}) = 0$$

$$\nabla_{\boldsymbol{w}} \frac{1}{n} \|\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y}\|_2^2 = 0$$

$$\nabla_{\boldsymbol{w}} [(\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y})^T (\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y})] = 0$$

$$\nabla_{\boldsymbol{w}} [\boldsymbol{w}^T \boldsymbol{X}^T \boldsymbol{X}\boldsymbol{w} - 2\boldsymbol{w}^T \boldsymbol{X}^T \boldsymbol{y} + \boldsymbol{y}^T \boldsymbol{y}] = 0$$

$$2\boldsymbol{X}^T \boldsymbol{X}\boldsymbol{w} - 2\boldsymbol{X}^T \boldsymbol{y} = 0$$

$$\boldsymbol{w} = (\boldsymbol{X}^T \boldsymbol{X})^{-1} \boldsymbol{X}^T \boldsymbol{y}$$

$$\frac{\partial \boldsymbol{x}^T \boldsymbol{A}}{\partial \boldsymbol{x}} = \boldsymbol{A}$$

$$\frac{\partial \boldsymbol{x}^T \boldsymbol{A}\boldsymbol{x}}{\partial \boldsymbol{x}} = (\boldsymbol{A} + \boldsymbol{A}^T)\boldsymbol{x}$$

$\boldsymbol{X}^T \boldsymbol{X}$ is symmetric

Linear regression: Math formulation & optimization

- Algebraic view of the minimizer
  - If $\boldsymbol{X}$ is invertible, just solve $\boldsymbol{X}\boldsymbol{w} = \boldsymbol{y}$ and get $\boldsymbol{w} = \boldsymbol{X}^{-1}\boldsymbol{y}$
  - But typically $\boldsymbol{X}$ is a tall matrix

$$\begin{bmatrix} \\ \boldsymbol{X} \\ \\ \end{bmatrix} \begin{bmatrix} \\ \boldsymbol{w} \\ \\ \end{bmatrix} = \begin{bmatrix} \\ \boldsymbol{y} \\ \\ \end{bmatrix} \Rightarrow \begin{bmatrix} \\ \boldsymbol{X}^T\boldsymbol{X} \\ \\ \end{bmatrix} \begin{bmatrix} \\ \boldsymbol{w} \\ \\ \end{bmatrix} = \begin{bmatrix} \\ \boldsymbol{X}^T\boldsymbol{y} \\ \\ \end{bmatrix}$$

$$\Downarrow$$

$$\boldsymbol{w} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}$$

Linear regression

- Given training data $\{(\boldsymbol{x}^{(i)}, y^{(i)}) : 1 \leq i \leq n\}$ i.i.d. from distribution $\mathcal{D}$
- Find $f_{\boldsymbol{w}}(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{x}$ that minimizes the loss $\hat{\mathcal{L}}(f_{\boldsymbol{w}}) = \frac{1}{n} \sum_{i=1}^{n} (\boldsymbol{w}^T \boldsymbol{x}^{(i)} - y^{(i)})^2$

Linear regression with bias

- Given training data $\{(\boldsymbol{x}^{(i)}, y^{(i)}) : 1 \leq i \leq n\}$ i.i.d. from distribution $\mathcal{D}$
- Find $f_{\boldsymbol{w}}(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x} + b$ that minimizes the loss

Bias term

- Reduce to the case without bias:
  - Let $\boldsymbol{w}' = [\boldsymbol{w}; b], \boldsymbol{x}' = [\boldsymbol{x}, 1]$
  - Then, $f_{\boldsymbol{w},b}(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x} + b = (\boldsymbol{w}')^T(\boldsymbol{x}')$
- Thus, whatever we did in previous slides still holds