Review quizzes (1/2)

- Explain random forest algorithm
- Explain the goal of linear regression
- Let $\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix},\ \boldsymbol{f} = \begin{bmatrix} f_1(\boldsymbol{x}) \\ f_2(\boldsymbol{x}) \end{bmatrix}, \boldsymbol{a} = \begin{bmatrix} a_1 \\ a_2 \end{bmatrix}$
  - What is the shape of $\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}$?
  - If $f_1(\boldsymbol{x}) = 3\boldsymbol{x}^T\boldsymbol{x},\ f_2(\boldsymbol{x}) = \boldsymbol{a}^T\boldsymbol{x}$, what is $\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}$?
- $\frac{d\boldsymbol{a}^T\boldsymbol{x}}{d\boldsymbol{x}} = ?$
- $\frac{d\boldsymbol{x}^T\boldsymbol{x}}{d\boldsymbol{x}} = ?$

$\nabla_\theta L = 0$

Review quizzes (2/2)

$\hat{y}_i = \theta_0 \cdot 1 + \theta_1 x_1 + \theta_2 x_2 \ldots$

- Let
  - Features $\boldsymbol{X} = \begin{bmatrix} 1 & x_{1,1} & \cdots & x_{1,d} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n,1} & \cdots & x_{n,d} \end{bmatrix}$,

    $\begin{bmatrix} \hat{y}_1 \\ \vdots \\ \hat{y}_n \end{bmatrix} = \begin{bmatrix} 1 & x_1 & \\ \vdots & & \ddots \\ 1 & & \end{bmatrix} \begin{bmatrix} \theta_0 \\ \vdots \\ \theta_n \end{bmatrix}$
  - Targets $\boldsymbol{y} = \left[y_1, \ldots, y_n\right]^T$
- Why adding a column of 1s to $\boldsymbol{X}$
- If we use linear regression with parameters $\boldsymbol{\theta} = \left[\theta_0, \ldots, \theta_d\right]^T$, show the formula of prediction $\hat{\boldsymbol{y}}$
- Show the residual sum of square (RSS) objective function using $\boldsymbol{y}$ and $\hat{\boldsymbol{y}}$

  $(\vec{y} - \hat{y})^T(\vec{y} - \hat{y})$

  $= (\vec{y} - X\theta)^T(\vec{y} - X\theta)$
- Derive $\boldsymbol{\theta}$ to minimize RSS

Exercise 2

- Requirement
  - Coding (90%)
    - Implement a decision tree classifier using Python.
      - You **cannot** use existing decision tree libraries (e.g., `sklearn.tree.DecisionTreeClassifier`)
    - Use your classifier to predict the class based on the Balance Scale Data Set (http://archive.ics.uci.edu/ml/datasets/Balance+Scale).
      - Separate the data into training (70%) and test (30%) datasets. Please make sure the dataset is split in a stratified fashion, i.e., the class distributions in the training and the test datasets are the same as the class distribution in the entire dataset.
      - Report both the training and the test error
  - A brief discussion of the results. (10%)
- Please submit your code and report to new ee-class
- Due date: 10/19 23:59:59

Today’s schedule

- My lecture: 2:00 – 4:20
- Python introduction by TAs: 4:20 – 4:50