CZ4041/SC4000: Machine Learning

Lesson 8b: Linear Regression

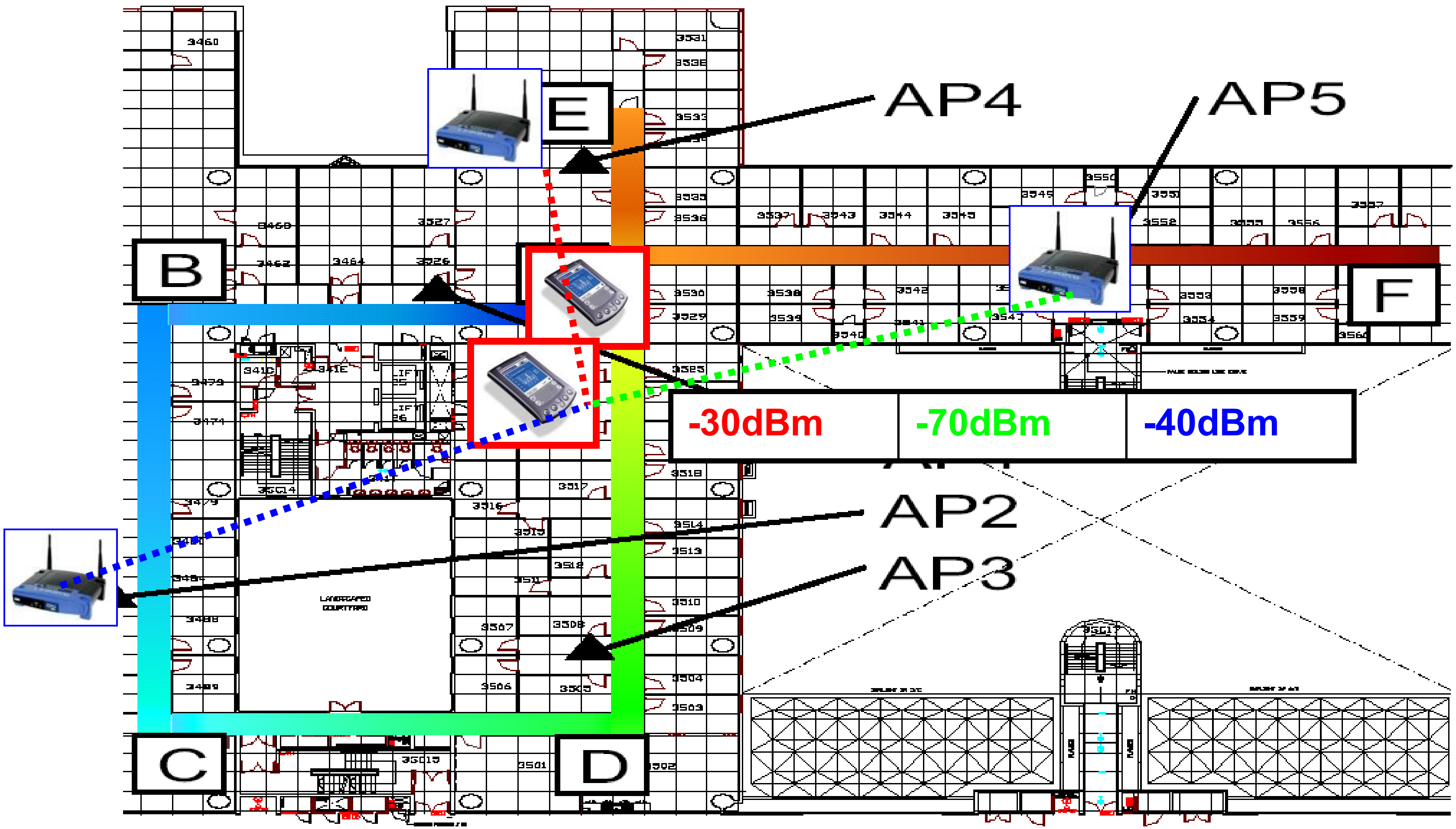
LI Boyang, Albert

School of Computer Science and Engineering, NTU, Singapore

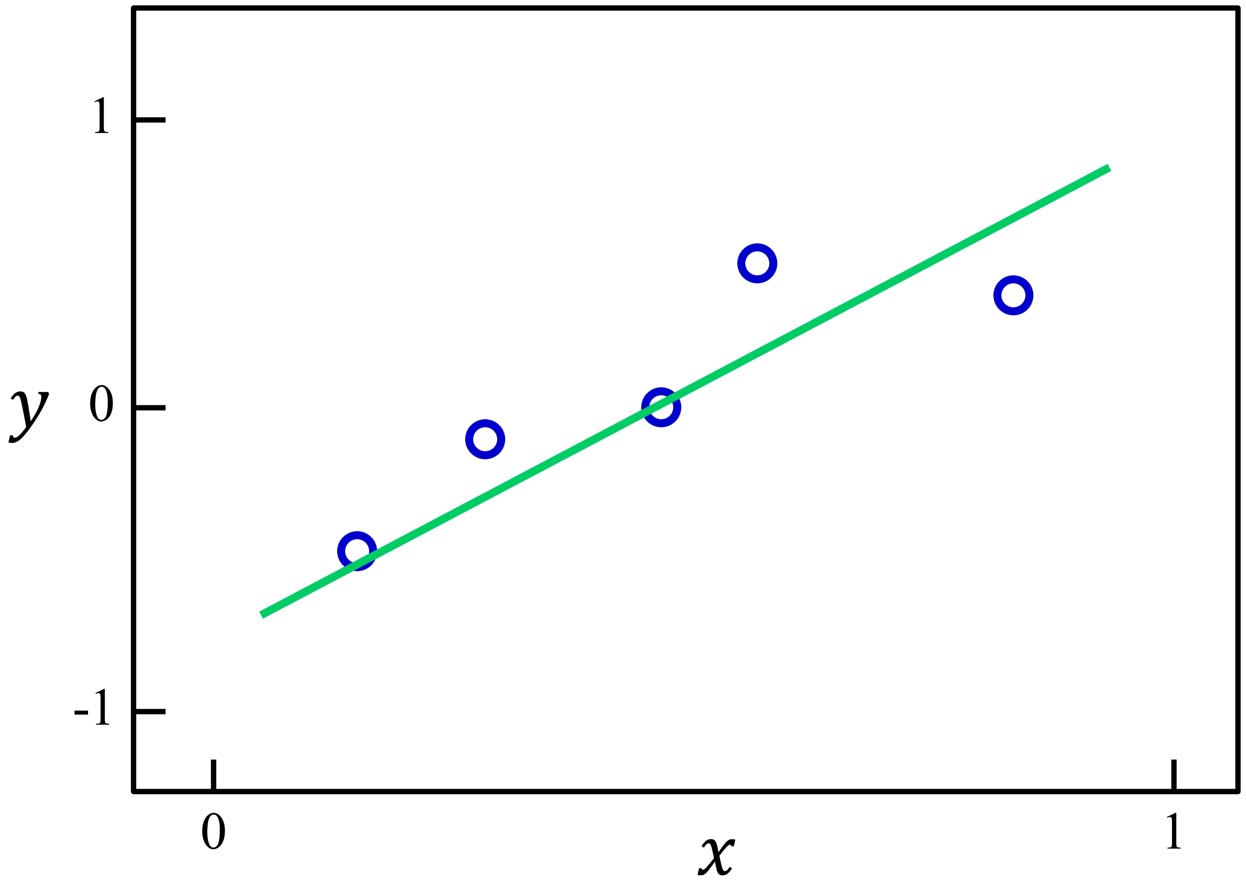
Acknowledgements: Slides are modified from the version prepared by Dr. Sinno Pan.

Regression: Example

Indoor WiFi localization



Linear Fitting



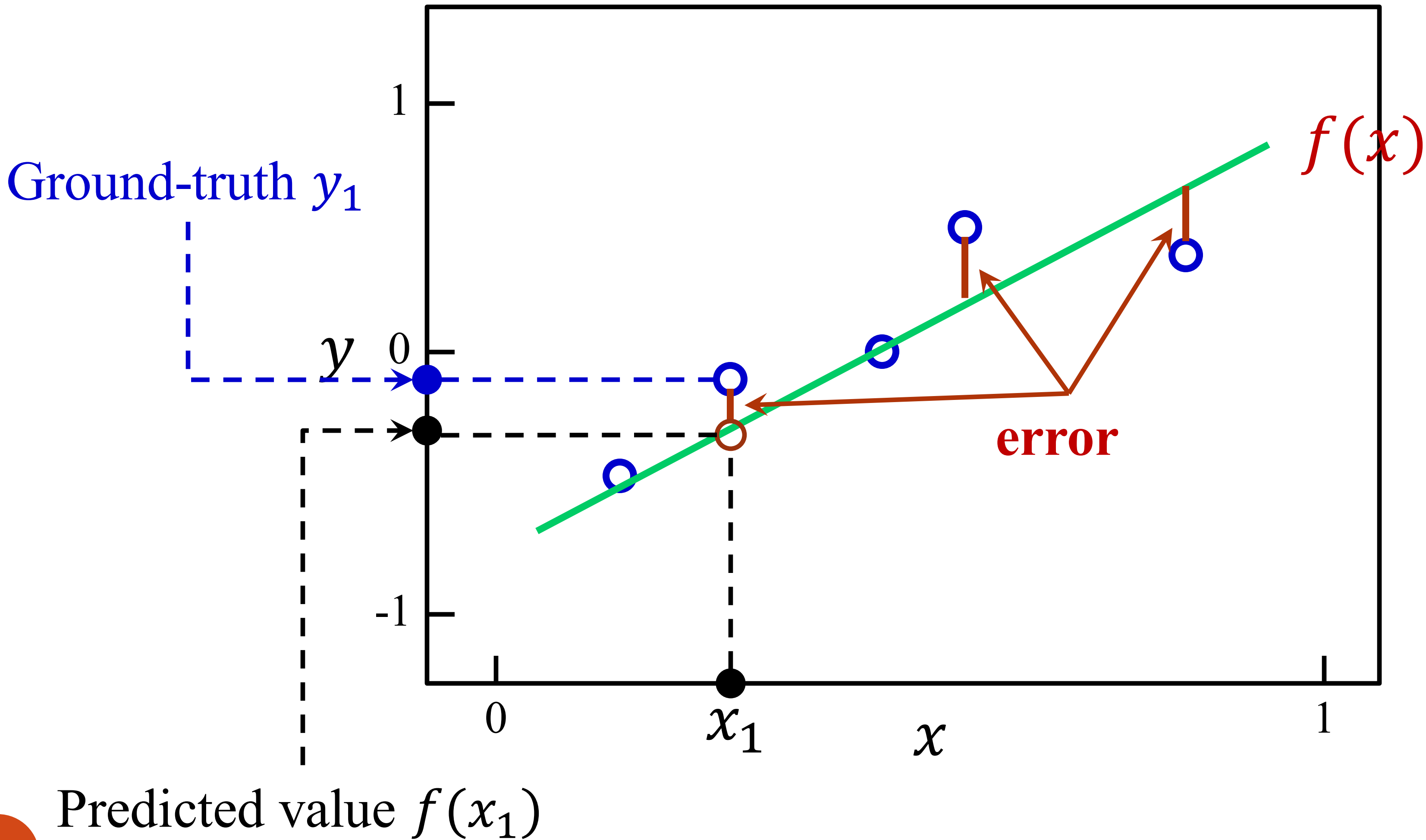
Linear Regression Model

- A special case, an instance is represented by one input feature
- To learn a linear function $f(x)$ in terms of w (drop bias term b for simplicity) from $\{x_i, y_i\}, i = 1, \ldots, N$

$$f(x) = w \cdot x$$

- Such that the difference (i.e., error) between the predicted values $f(x_i)'s$ and the ground-truth values $y_i's$ is as small as possible

Linear Regression Model (cont.)



Linear Regression Model (cont.)

- Suppose sum-of-squares error is used

$$\mathcal{L}(w) = \frac{1}{2}\sum_{i=1}^{N}(f(x_i) - y_i)^2 = \frac{1}{2}\sum_{i=1}^{N}(w \times x_i - y_i)^2$$

- Learn the linear model in terms of w by minimizing the error

$$w^* = \arg\min_{w} \mathcal{L}(w)$$

Linear Regression Model (cont.)

- To solve the unconstrained minimization problem, we can set the derivative of $E(w)$ w.r.t. w to zero

$$\frac{\partial \mathcal{L}(w)}{\partial w} = \frac{\partial \left(\frac{1}{2} \sum_{i=1}^{N} (w \times x_i - y_i)^2 \right)}{\partial w} = 0$$

$$\Downarrow$$

$$\sum_{i=1}^{N} (w \times x_i - y_i) \times x_i = 0$$

Linear Regression Model (cont.)

$$\sum_{i=1}^{N}(w \times x_i - y_i) \times x_i = 0$$

$$\Downarrow$$

$$w\sum_{i=1}^{N} x_i^2 - \sum_{i=1}^{N} y_i \times x_i = 0$$

$$\Downarrow$$

$$w = \frac{\sum_{i=1}^{N} y_i \times x_i}{\sum_{i=1}^{N} x_i^2}$$

Linear Regression Model (cont.)

- To learn a linear function $f(\boldsymbol{x})$ in more general form in terms of $\boldsymbol{w}$ and b,

$$f(\boldsymbol{x}) = \boldsymbol{w} \cdot \boldsymbol{x} + b$$

- By defining $w_0 = b$, and $X_0 = 1$, $\boldsymbol{w}$ and $\boldsymbol{x}$ are of $d + 1$ dimensions

$$f(\boldsymbol{x}) = \boldsymbol{w} \cdot \boldsymbol{x}$$

Linear Regression Model (cont.)

- Suppose sum-of-squares error is used

$$\mathcal{L}(\boldsymbol{w}) = \frac{1}{2}\sum_{i=1}^{N}(f(\boldsymbol{x}_i) - y_i)^2 = \frac{1}{2}\sum_{i=1}^{N}(\boldsymbol{w}\cdot\boldsymbol{x}_i - y_i)^2$$

- Learn the linear model in terms of $\boldsymbol{w}$ by minimizing the error

$$\|\boldsymbol{w}\|_2^2 = \boldsymbol{w}\cdot\boldsymbol{w}$$

$$\boldsymbol{w}^* = \arg\min_{\boldsymbol{w}} \mathcal{L}(\boldsymbol{w}) + \frac{\lambda}{2}\boxed{\|\boldsymbol{w}\|_2^2}$$

Positive tradeoff parameter

A regularization term to control the complexity of the model

Regularized Linear Regression

- To solve the unconstrained minimization problem, we can set the derivative of $\mathcal{L}(\boldsymbol{w}) + \frac{\lambda}{2}\|\boldsymbol{w}\|_2^2$ w.r.t. $\boldsymbol{w}$ to **zero**

$$\frac{\partial\left(\mathcal{L}(\boldsymbol{w}) + \frac{\lambda}{2}\|\boldsymbol{w}\|_2^2\right)}{\partial \boldsymbol{w}} = \mathbf{0}$$

- We can obtain a closed-form solution for $\boldsymbol{w}$ by the above equations

Closed-Form Solution

- Denote by $\mathbf{X} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_N)^T$

$$\mathbf{X} = \begin{array}{c} \begin{array}{ccc} \boldsymbol{x}_1 & & \end{array} \\ \begin{pmatrix} x_{10} & \cdots & x_{N0} \\ \vdots & \ddots & \vdots \\ x_{1d} & \cdots & x_{Nd} \end{pmatrix}^T \end{array} = \begin{array}{c} \boldsymbol{x}_1 \\ \begin{pmatrix} x_{10} & \cdots & x_{1d} \\ \vdots & \ddots & \vdots \\ x_{N0} & \cdots & x_{Nd} \end{pmatrix} \end{array}$$

- And by $\boldsymbol{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_N \end{pmatrix}$

- The closed-form solution for $\boldsymbol{w}$:

$$\boldsymbol{w} = \left(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I}\right)^{-1}\mathbf{X}^T\boldsymbol{y}$$

How to induce this closed-form solution?

Tutorial

Implementation Example

```
>>> from sklearn.linear_model import Ridge
...
>>> rlr = Ridge(alpha=0.1)
>>> rlr.fit(X, y)
>>> pred_train_rlr= rlr.predict(X)
```

Referred to as λ on our lecture notes

Evaluation

- Root Mean Square Error (RMSE)

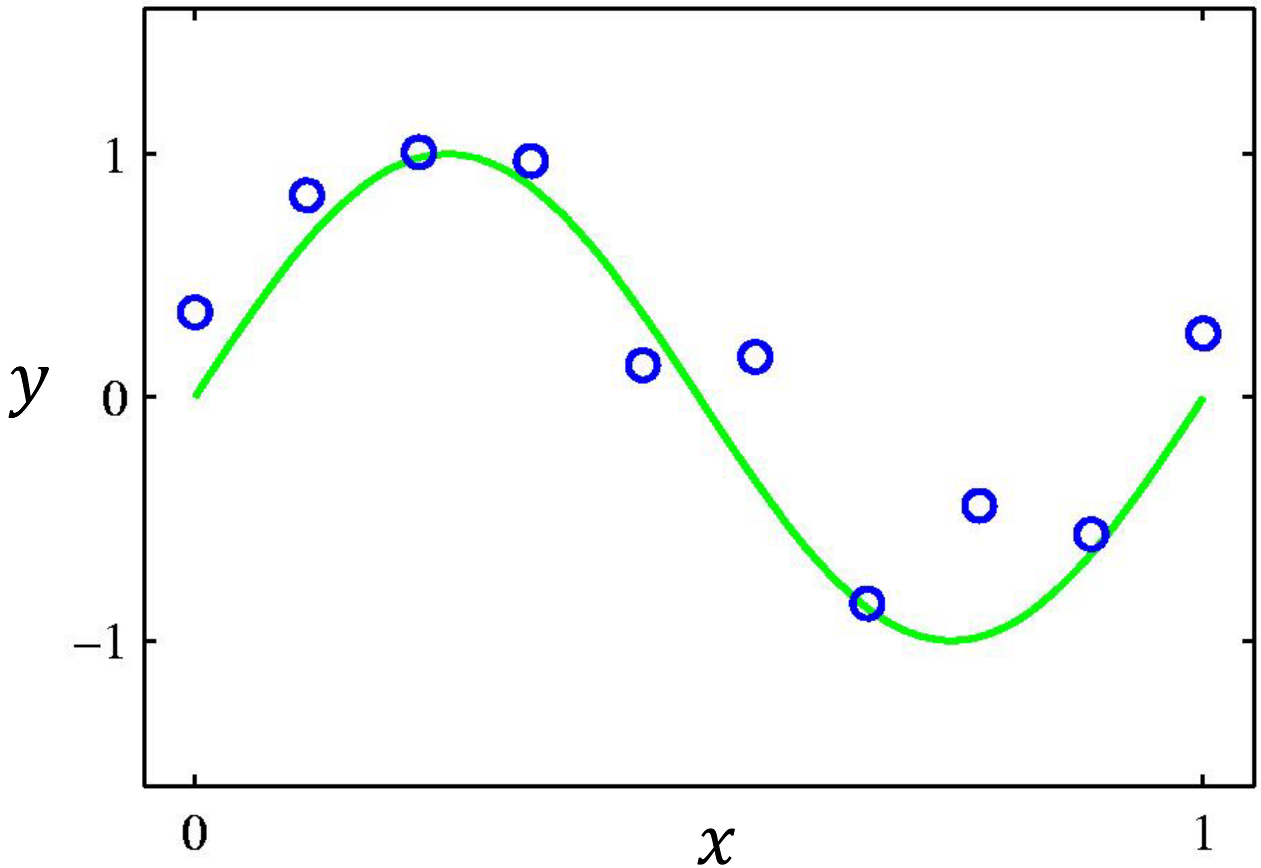
$$\sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(f(\boldsymbol{x}_i) - y_i\right)^2}$$

- Mean Absolute Error (MAE)

$$\frac{1}{N}\sum_{i=1}^{N}\left|f(\boldsymbol{x}_i) - y_i\right|$$

Additional Notes (Optional)

Nonlinear Fitting



Kernel trick

Thank you!