Machine Learning Bullet Points

Data

- data modality: numbers, images, videos, text, graph, audios, point cloud
- data cleaning

  As an example, given user-commodity matrix in recommendation system, we may have to solve the following

  $$A = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix} \approx U\Sigma_2 V^T$$
  $$U \text{ is the user-property matrix} \in \mathbb{R}^{3\times 2},$$
  $$V \text{ is the commodity-property matrix} \in \mathbb{R}^{4\times 2}$$

  But when there is missing entries in A, we may try something else:

  $$A = \begin{bmatrix} 1 & 0 & ? & 0 \\ 1 & ? & 1 & ? \\ 0 & 1 & ? & 0 \end{bmatrix}$$
  $$M = \begin{bmatrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \end{bmatrix}$$
  $$\min_{U,V} ||M \otimes (UV^T - A)||_F$$
  $$\text{s.t. } \operatorname{rank} U, \operatorname{rank} V \leq r$$

- data augmentation
- feature extraction / feature engineering
  - domain expertise
  - statistical knowledge
    - principal component analysis
    - kernel methods

$$\mathrm{x} = [x_1, x_2, x_3]$$
$$\mathrm{y} = [y_1, y_2, y_3]$$
$$feature(\mathrm{x}) = x_1 x_2 + x_3^3$$
$$feature(\mathrm{y}) = y_1 y_2 + y_3^3$$
$$d(\mathrm{x}, \mathrm{y}) = feature(\mathrm{x})^T feature(\mathrm{y})$$
$$d(\mathrm{x}, \mathrm{y}) = e^{\frac{\mathrm{x}^T \mathrm{y} - \mu}{\sigma^2}}$$

    - neural network
- feature normalization/standardization

  $x_1, x_2, x_3 \rightarrow \frac{x_1 - \mu}{\sigma}, \frac{x_2 - \mu}{\sigma}, \frac{x_3 - \mu}{\sigma}$
- imbalanced data
- data splitting

  training set + validation set + test set

Algorithm / Model

- supervised model (x,y) / unsupervised model (x)
- regression model / classification model
- generative model / discriminative model
- linear model / non-linear model

$$f(x + y) = f(x) + f(y)$$
$$f(cx) = cf(x)$$

  Linear model is simple so we usually like it. But linear model is not powerful enough to handle all kinds of data.

  So we usually use the combination of nonlinear feature + linear model. For example, neural network = original data + nonlinear part (for extracting nonlinear feature) + linear part.

Evaluation / Metrics

Training Metrics

- mean squared error

$$\text{prediction: } [0.6, 0.4], \text{truth: } [1, 0]$$
$$\text{error : } [(0.6 - 1)^2 + (0.4 - 0)^2]/2$$

- cross-entropy loss

$$\text{prediction: } [0.6, 0.4], \text{truth: } [1, 0]$$
$$\text{error} : 1 \log 0.6 + 0 \log 0.4$$

- l1-norm

$$\text{prediction: } [0.6, 0.4], \text{truth: } [1, 0]$$
$$\text{error} : [|(0.6 - 1)| + |(0.4 - 0)|]$$

Test Metrics

- accuracy, recall, precision

Truth/Prediction	Positive	Negative
Positive	TP	FN
Negative	FP	TN

  accuracy: $\frac{TP+TN}{TP+FN+FP+TN}$
  recall: $\frac{TP}{TP+FN}$
  precision: $\frac{TP}{TP+FP}$
- mean squared error

Visualization

Python + Machine Learning

- NumPy
  - matrix computation
  - function
  - random sampling
- scikit-learn
  - decision tree
  - SVM
  - logistic regression
  - KNN
  - KMeans
  - multi-layer perceptron
- Pandas

- PyTorch / TensorFlow
- Matplotlib