Project 2

Learning from Data

Outline

- Introduction

- Task Description

- Summary

Outline

- Introduction
- Task Description
- Summary

Introduction

Learning from data

- There is a training set which is the observed data
- The learning: An AI model is generated from the training set,
- The AI model is used in **unseen** new data, i.e., the test set

- **BASIC ASSUMPTION**
  - The data distribution (or “Pattern”) of the training set and the test set is the same
  - The AI model is an abstraction of the distribution.

Introduction

Typical Pipeline of Learning from Data

- Feature extraction: Make the raw input data (e.g., dialog text) be structured
- Feature selection: Discard some features for dimensionality reduction
- Training: Generate a model from the training set
- Deployment: Use the trained model to process the unseen data

Introduction

Typical Pipeline of Learning from Data

- Feature extraction: Make the raw input data (e.g., dialog text) be structured
- Feature selection: Discard some features for dimensionality reduction
- Training: Generate a model from the training set
- Deployment: Use the trained model to process the unseen data

Three sub-tasks related to feature selection and training are considered in this project.

Introduction

How to Make An Effective Training

Since unseen data (the test set) is not accessible during training, how to make sure the trained model does not **overfit** the training set?

- Split the training set into a validation set and a smaller training set
- Train the model on the smaller training set
- **Validate** the accuracy of the model on the validation set
- Tune the training parameters of the model to make sure the training is not overfit
- Use the training parameters to train the model on the whole training set from scratch

Introduction

Supervised Training

- The training data
  - **The input vectors**
  - **The corresponding target vectors**
- The unseen data, i.e., the test set
  - The input vectors
- The goal
  - Generate a model to predict the target vectors for the unseen data
  - Regression: continuous target vectors
  - Classification: discrete target vectors

Introduction

Unsupervised Training

- The training data
  - **The input vectors**
- The unseen data, i.e., the test set
  - The input vectors
- The goal
  - Clustering: discover groups by similarity metrics
  - Density estimation: determine the data distribution
  - Visualization: dimension reduction
  - ...

Introduction

Feature Selection

- Pros
  - Make the problem easier to solve
  - Speed up training
  - The training set and test set should use the same feature selection process
- Cons
  - Discarding features results in discarding input information
  - The accuracy of the whole system will suffer if important information is discarded
- The Goal
  - A **trade-off** between the accuracy (the higher the better) and the input dimension (the lower the better)
  - Density estimation: determine the data distribution
  - Visualization: dimension reduction
  - ...

Outline

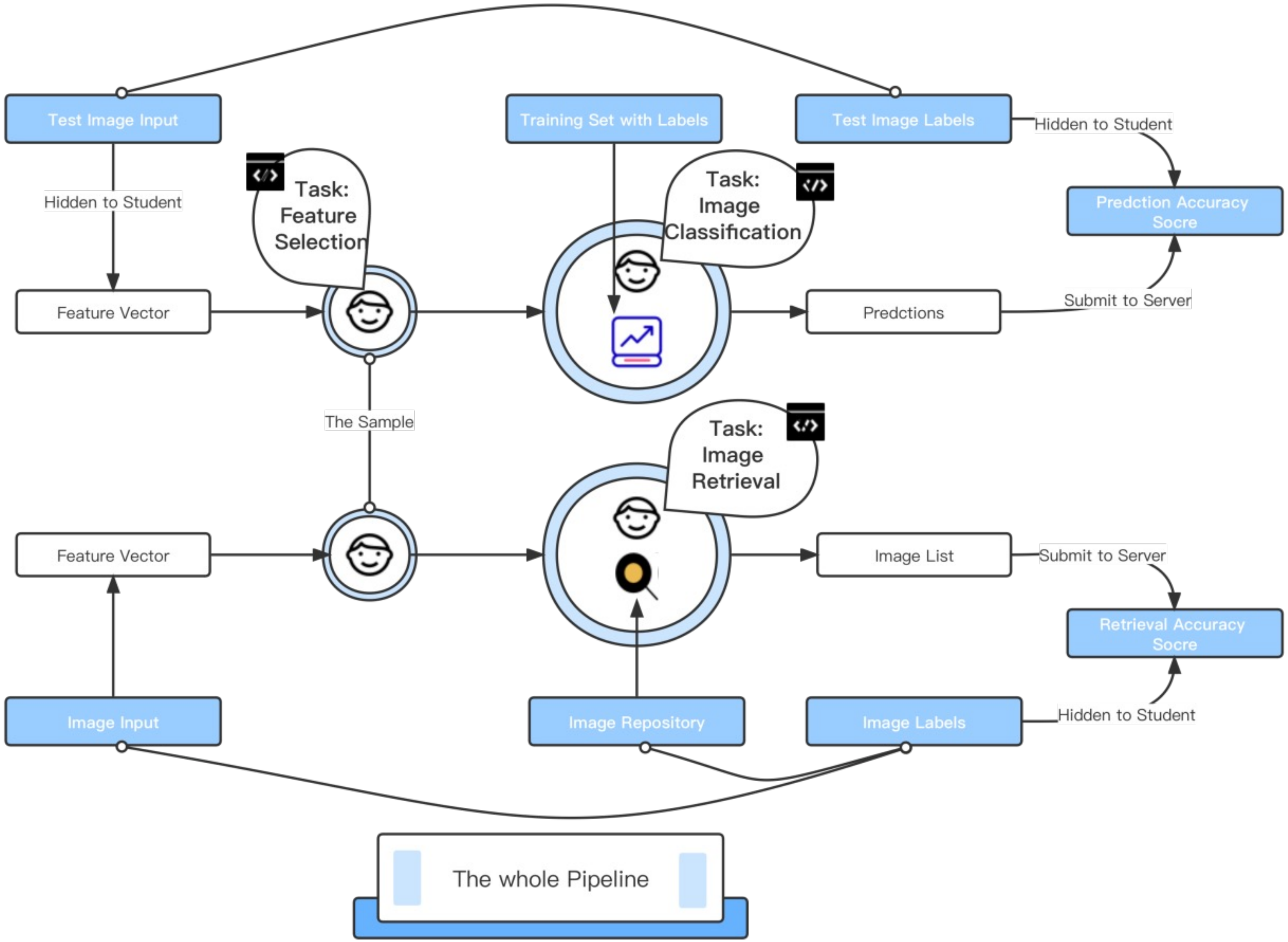
- Introduction

- Task Description

- Summary

Task Description

Overview



Task Description

Three sub-tasks

- Sub-task1: Supervised Learning
- Sub-task2: Unsupervised Learning
- Sub-task3: Feature Selection

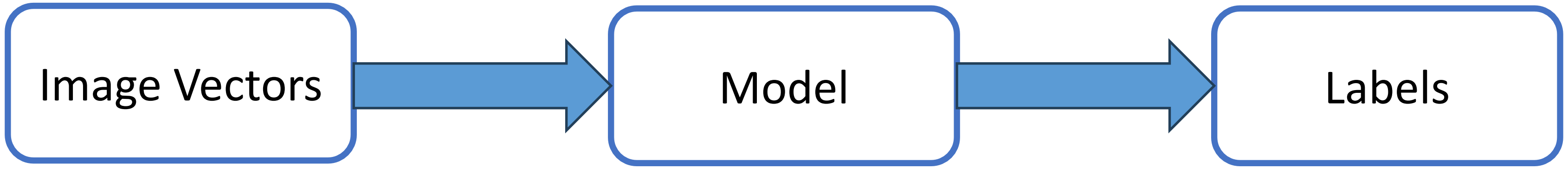
Task Description

Three sub-tasks

- **Sub-task1: Supervised Learning**
- Sub-task2: Unsupervised Learning
- Sub-task3: Feature Selection

Sub-task1: Supervised Learning

Supervised Learning for Image Classification

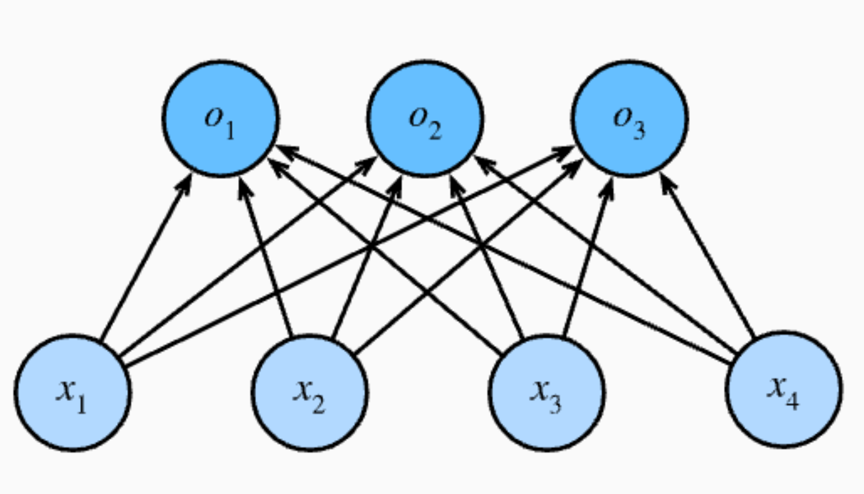


- This task is to **generate a model to predict the label of an input image**
- The raw image is not provided
- The preprocessed image vector is used in this task

Sub-task1: Supervised Learning

Baseline: SoftMax Regression

- Model
  - Linear model: $\mathbf{o} = \mathbf{W}\mathbf{x} + \mathbf{b}, \mathbf{x} \in \mathbb{R}^{\mathrm{d}}, \mathbf{W} \in \mathbb{R}^{\mathrm{q} \times \mathrm{d}}, \mathbf{b} \in \mathbb{R}^{\mathrm{q}}$
  - SoftMax: $\hat{\mathbf{y}} = \mathrm{softmax}(\mathbf{o})$, i.e., $\widehat{y}_i = \frac{\exp(\mathrm{o_i})}{\sum_{\mathrm{j}} \exp(\mathrm{o_j})}$
- Loss Function
  - $L(y, \hat{y}) = -\sum_{j=1}^{q} y_j \log \widehat{y}_i$



Sub-task1: Supervised Learning

Provided

- Training set
  - input vectors are given as **classification_train_data.pkl**
  - target vectors are given as **classification_train_label.pkl**
- Test set
  - input vectors are given as **classification_test_data.pkl**
- Python scripts
  - **image_load_demo.ipynb**: explain how to load data from classification_train_data.pkl and classification_train_label.pkl
  - **image_classification_demo.ipynb**: the baseline of this task

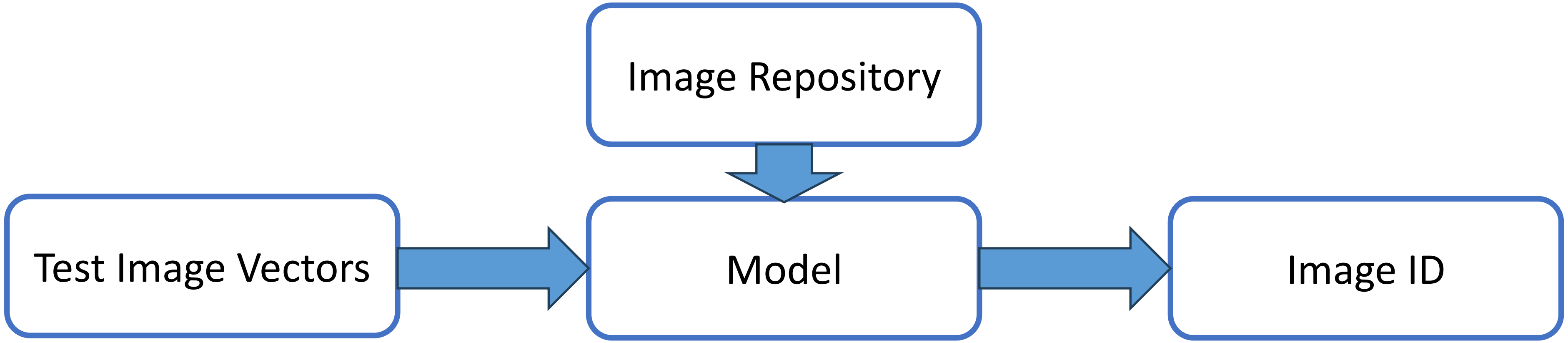
Task Description

Three sub-tasks

- Sub-task1: Supervised Learning
- **Sub-task2: Unsupervised Learning**
- Sub-task3: Feature Selection

Sub-task2: Unsupervised Learning

Unsupervised Learning for Image Retrieval



- This task is to **find similar images in the image repository**, given a test input image vector
- The raw image is not provided
- The training set is the image repository

Sub-task2: Unsupervised Learning

Baseline: Nearest Neighbor Search (NNS)

- Select the Euclidean distance as the similarity measure.
- For each “query” image, find K images that are most similar to it in the repository.

Sub-task2: Unsupervised Learning

Provided

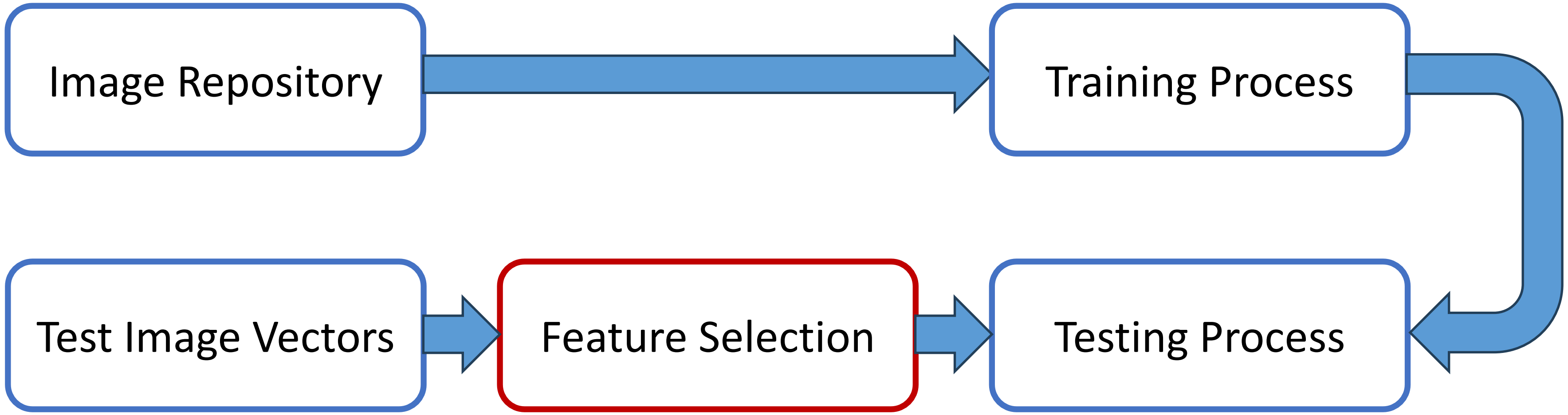
- Image Repository
  - preprocessed image vectors are given as **image_retrieval_repository_data.pkl**
- Test set
  - input vectors are given as **image_retrieval_test_data.pkl**
- Python scripts
  - **image_load_demo.ipynb**: explain how to load data from image_retrieval_repository_data.pkl
  - **image_retrieval_demo.ipynb**: the baseline of this task

Task Description

Three sub-tasks

- Sub-task1: Supervised Learning
- Sub-task2: Unsupervised Learning
- **Sub-task3: Feature Selection**

Sub-task3: Feature Selection



- This task is **to select 30 dimensions** from the original input dimension
- The training process and test process are fixed
- The classification accuracy indicates the quality of the selected features

Sub-task3: Feature Selection

Baseline: Random selection

- Select a fixed random seed.
- Randomly select 30 features.
- Generate mask for the selected 30 features.

Sub-task3: Feature Selection

Provided

- Validation set
  - input vectors are given as **classification_validation_data.pkl**
  - target vectors are given as **classification_validation_label.pkl**
- Python scripts
  - **feature_selection.ipynb**: demonstrate how to select features from the classification_validation_datal.pkl
  - **image_recognition.ipynb**: an image classification process for evaluating the quality of the selected features by evaluating the model on the test set (or the validation set for students off-line)

Outline

- Introduction
- Task Description
- Summary

Summary

Project 2: Learning from Data contains three sub-tasks

- Sub-task1: Supervised Learning
- Sub-task2: Unsupervised Learning
- Sub-task3: Feature Selection

[1] Christopher M. Bishop:Pattern recognition and machine learning, 5th Edition. Information science and statistics, Springer 2007, ISBN 9780387310732, pp. I-XX, 1-738

[2] Zhang, A., Lipton, Z. C., Li, M., & Smola, A. J. (2021). Dive into deep learning. arXiv preprint arXiv:2106.11342.