EE6550 Machine Learning

Lecture One – Part I
Introduction

Chung-Chin Lu

Department of Electrical Engineering

National Tsing Hua University

February 13, 2017

The Contents of This Lecture - Part I

- Basic definitions and concepts.
- Introduction to the problem of learning.

Machine Learning

- Defined as computational methods using experience to improve performance or to make accurate predictions.
  - Experience: the past information available to the learner, typically taking the form of electronic data.
- Consisting of designing efficient and accurate prediction algorithms.
  - Efficiency: time and space complexity.
  - Accuracy: sample complexity.
- Inherently related to data analysis and statistics.
  - Data-driven: combining fundamental concepts in computer science with ideas from statistics, probability and optimization.

Examples of Learning Tasks

- Text or document classification, e.g., spam detection;
- Speech recognition, speaker verification;
- Optical character recognition (OCR);
- Image recognition, face detection;
- Fraud detection (credit card, telephone) and network intrusion;
- Games, e.g., chess;
- Unassisted vehicle control (robots, navigation);
- Bioinformatics, e.g., protein function or structured prediction;
- Medical diagnosis;
- Recommendation systems, search engines, information extraction systems.

Major Areas of Machine Learning

- Classification: assign a category to each item.
  - Document classification, image classification;
  - OCR, text classification, speech recognition.
- Regression: predict a real value for each item.
  - Prediction of stock values and economic variables.
- Ranking: order items according to some criterion.
  - Relevant web pages returned by a search engine.
- Clustering: partition data into *homogenous* regions.
  - Analysis of very large data sets, e.g., social network analysis.
- Dimensionality reduction: find lower-dimensional manifold preserving some properties of the data.
  - Preprocessing digital images in computer vision tasks.

Objectives of Machine Learning

- Theoretical questions:
  - What can actually be learned, under what conditions?
  - Are there learning guarantees?
  - Analysis of learning algorithms.
- Algorithms:
  - Design efficient and robust algorithms to generate accurate predictions for unseen items.
  - Deal with large-scale problems.
  - Handle a variety of different learning problems.

Definitions and Terminology

- Item: an instance or point or example or member in the population to be studied.
  - email messages in spam detection.
- Features: the set of visible attributes associated to an item, often represented as a vector.
  - length of an email message, name of the sender, various characteristics of the header, the presence of certain keywords in the body of the message in spam detection.
- Label: the hidden attribute, either category (classification) or real value (regression) associated to an item.
  - spam and non-spam categories in spam detection, a binary classification problem.

- Data sample: the data associated with a set of items (a sample) collected from the population and used for learning and evaluation.
  - training data (sub)sample (typically labeled).
  - validation data (sub)sample (labeled, for tuning parameters).
  - test data (sub)sample (labeled but labels not seen).

Feature Extraction

- User's prior knowledge about the learning task.
  - It is usually difficult to know what features should be extracted in a general learning problem.
- Representation learning : a machine learning approach - deep learning.
  - Deep learning introduces representations that are expressed in terms of other, simpler representations.

Deep Learning

- The depth of deep learning allows the machine to build complex representations out of simpler ones.
  - A nested hierarchy of representations.
- The depth of deep learning allows the machine to learn a multi-step learning algorithm.
  - A nested hierarchy of concepts.

Standard Machine Learning Scenarios

- Unsupervised learning: no labeled data.
  - Without any label, it can be difficult to quantitatively evaluate the performance of a learner.
  - Density estimation: aims to find the (statistical) structure of the feature space such that certain features occur more often than others.
  - Examples: clustering and dimensionality reduction.
- Supervised learning: uses labeled data for prediction on unseen items.
  - Examples: classification, regression, and ranking problems.

- Semi-supervised learning: uses labeled and unlabeled data for prediction on unseen items.
  - Unlabeled data is easily accessible but labels are expensive to obtain.
  - Hoping that the distribution of unlabeled data can help to achieve a better performance than in the supervised setting using only labeled data.
  - Under what conditions a better performance can be achieved in this setting ?
  - Examples: classification, regression, and ranking tasks.

- Transductive inference: uses labeled and unlabeled data for prediction on seen items.
  - A (small) labeled training sample along with a (large) set of unlabeled test items.
  - To predict labels only for these particular test items.
  - Advantage: able to consider all of the items, not just the labeled items, while performing the labeling task and to make better predictions with fewer labeled items, from the natural breaks found in the unlabeled items.
  - Disadvantage: builds no predictive model and can be computationally expensive.
  - Under what conditions a better performance can be achieved in this setting ?
  - Examples: classification, regression, and ranking tasks.

The Contents of This Lecture - Part I

- Basic definitions and concepts.
- Introduction to the problem of learning.

Definitions

- Input space $\mathscr{I}$: the population of all possible items.
- Feature space $\mathscr{X}$: the set of all possible feature vectors.
  - $\boldsymbol{X}(\omega)$: the feature vector associated with an item ω in the population $\mathscr{I}$.
  - $\mathscr{X} = \{\boldsymbol{X}(\omega) \mid \omega \in \mathscr{I}\}$.
- Label space $\mathscr{Y}$: the set of all possible labels.
  - $Y(\omega)$: the label associated with an item ω in the population $\mathscr{I}$.
  - $\mathscr{Y} = \{Y(\omega) \mid \omega \in \mathscr{I}\}$.
- Output space $\mathscr{Y}'$: the set of all possible predictions, usually $\mathscr{Y}' = \mathscr{Y}$.

- Loss function: $L : \mathscr{Y}' \times \mathscr{Y} \to \mathbb{R}$.
  - $L(y', y)$: cost of predicting y' instead of label y.
  - binary classification: 0-1 loss with $\mathscr{Y}' = \mathscr{Y}$, $L(y', y) = 1_{y' \neq y}$.
  - regression: $\mathscr{Y}' = \mathscr{Y} \subseteq \mathbb{R}$, $L(y', y) = (y' - y)^2$.
- Hypothesis set $\mathcal{H} \subseteq {\mathscr{Y}'}^{\mathscr{I}}$: A subset of (measurable) functions from the input space $\mathscr{I}$ to the output space $\mathscr{Y}'$, out of which the learner select her/his hypothesis.

Learning Stages

- Randomly partition a given data sample into a training (sub)sample, a validation (sub)sample, and a test (sub)sample.
- Associate relevant features to the items.
  - A critical step in the design of machine learning solutions.
  - The choice of the features is either left to the user based on the user's prior knowledge about the learning tasks or by a deep representation learning.
- Use the selected features to train our learning algorithm (by using the training sample) in different settings of the values of its free parameters.
  - For each set of values of these parameters, the algorithm selects a different hypothesis out of the hypothesis set $\mathcal{H}$.

- Choose among them the hypothesis resulting in the best performance on the validation sample.
- Use that hypothesis to predict the labels of the items in the test sample.
- Evaluate the performance of the algorithm by using the loss function associated to the task.

Cross Validation

- Used when the amount of labeled data available is too small to set aside a validation sample since that would leave an insufficient amount of training data.
- $\boldsymbol{\theta}$: the vector of free parameters of the learning algorithm.
- Randomly partition a given training sample S of m labeled items into n subsamples or folds.
- $((\omega_{i1}, y_{i1}), ..., (\omega_{im_i}, y_{im_i}))$: the ith fold of size m_i, $1 \leq i \leq n$.
- For any $i \in [1, n]$, the learning algorithm is trained on all but the ith fold to generate a hypothesis h_i, and the performance of h_i is tested on the ith fold.

- $\hat{R}_{CV}(\boldsymbol{\theta})$: the cross-validation error.

$$\hat{R}_{CV}(\boldsymbol{\theta}) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{m_i}\sum_{j=1}^{m_i} L(h_i(\omega_{ij}), y_{ij}).$$

- Choose a parameter vector $\boldsymbol{\theta}^*$ which minimize the cross-validation error $\hat{R}_{CV}(\boldsymbol{\theta})$. That is

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \hat{R}_{CV}(\boldsymbol{\theta}).$$

- Usually $m_i = \frac{m}{n}$ for all i.
- How to choose n ? Typically $n = 5$ or 10.

Model Selection

- $\boldsymbol{\theta}$: the free parameter vector to be selected or determined.
- Partition the full labeled data into a training sample and a test sample.
- Use the training sample of size m to compute the n-fold cross-validation error $\hat{R}_{CV}(\boldsymbol{\theta})$ for a finite set $\boldsymbol{\Theta}$ of possible values of $\boldsymbol{\theta}$.
- $\boldsymbol{\theta}_0$: the parameter vector which minimizes $\hat{R}_{CV}(\boldsymbol{\theta})$ for all $\boldsymbol{\theta}$ in $\boldsymbol{\Theta}$. That is

$$\boldsymbol{\theta}_0 = \arg\min_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} \hat{R}_{CV}(\boldsymbol{\theta}).$$

- Train the algorithm with the parameter setting $\boldsymbol{\theta}_0$ over the full training sample of size m.
- Evaluate the performance on the test sample.