Introduction to Machine Learning

Shan-Hung Wu
shwu@cs.nthu.edu.tw

Department of Computer Science,
National Tsing Hua University, Taiwan

NetDB-ML, Spring 2015

Outline

Outline

1. What is Machine Learning?
2. Learning Tasks
3. ML vs. AI vs. Data Mining
4. About This Course
5. FAQ

A Priori and Posteriori Knowledge

- To solve a problem, we need an algorithm
  - E.g., sorting (input: a set of numbers, output: their ordered list)
  - The algorithms we learn so far assume the ***a priori knowledge*** about the problem
- For some tasks, however, we do not have the a priori knowledge
  - E.g., to tell if an email is spam or not
  - The correct answer varies in time and from person to person
- This course introduces an another type of algorithms
  - Takes example data as the additional input
  - Observe the ***posteriori knowledge*** from example data
  - Use this knowledge to solve the problem

General Machine Learning Steps

1. Data collection and preprocessing (e.g., integration, cleaning, etc.)
2. Model development
    1. Assume a ***model*** that represents the posteriori knowledge we want to discover. The model has parameters
    2. Define an ***objective*** that measures "how good the model with a particular combination of parameters can explain the data"
3. ***Training***: employ an algorithm that optimizes the objective by finding the best (or good enough) parameters
4. ***Testing***: evaluate the model performance on hold-out data
5. Using the model

How to Use the Learned Models?

- ***Predictive*** models
  - Approximate the unknown a priori knowledge
  - For making predictions when seeing new data
- ***Descriptive*** models
  - Gain insights to the data

Outline

Example Learning Tasks

- Classification
- Regression
- Clustering
- Learning association rules
- And more...

Classification

Examples

Is email $\boldsymbol{x}$ a spam? Will customer $\boldsymbol{x}$ like this video? Does image $\boldsymbol{x}$ contain John's face or not?

- Experience: $\{\boldsymbol{x}^{(1)}, r^{(1)}\}, \{\boldsymbol{x}^{(2)}, r^{(2)}\}, \cdots$
  - $\boldsymbol{x}^{(t)} = [x_1^{(t)}, x_2^{(t)}, \cdots]^\top$ is a vector, whose each component $x_i^{(t)}$ denotes an attribute
  - $r^{(t)} \in \{true, false\}$ is the label
- A model (predictive): $f(\boldsymbol{x}; \theta) : \mathcal{I} \to \{true, false\}$
  - $\mathcal{I}$ and $\mathcal{R}$ are spaces of $\boldsymbol{x}^{(t)}$ and $r^{(t)}$ respectively
- Parameters: θ
- Objective: $\arg_\theta \min \sum_t l(f(\boldsymbol{x}^{(t)}; \theta), r^{(t)})$, where $l(a,b)$ equals 1 if $a \neq b$; 0 otherwise

Regression

Examples

Given the history of $\boldsymbol{x}$ in stock market, what is $\boldsymbol{x}$'s price tomorrow? How much chance the video $\boldsymbol{x}$ will be popular?

- A generalization of the classification problem where $\mathcal{R}$ is continuous
- Both classification and regression are called ***supervised learning*** problems where each example $\boldsymbol{x}^{(t)}$ has supervised output $r^{(t)}$
  - An ML algorithm learns the mapping

Clustering

- What if there is no supervised output $r^{(t)}$?
  - Well, we can still learn the regularities in $\mathcal{I}$

Example

What books are bought together frequently?

- Experience: $\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \cdots$
  - Each attribute $x_i^{(t)} \in \{true, false\}$ of $\boldsymbol{x}^{(t)}$ indicates whether $\boldsymbol{x}^{(t)}$ participated in the transaction i or not
- A model (predictive): a collection of groups, i.e., $\{G_k\}_{k=1}^{K}$
- Parameters: K, G_1, $\cdots$, and G_K
- Objective: $\arg_{K,G_1,\cdots,G_K} \max \prod_{t=1}^{N} \sum_{k=1}^{K} P(\boldsymbol{x}^{(t)}|G_k)P(G_k)$
- We ***cluster*** the books into groups without knowing what the groups are in advance
- Clustering is also useful to image compression
  - Identifies pixels with similar color and store them as a group

Learning Association Rules

Examples

Customers who buy books $\{A, B\}$ also buy E, or users clicking/watching video P also click/watch $\{Q, R\}$, etc.

- Input: transactions of items bought together, i.e., $\{A, E, G\}, \{B, G\}, \cdots$
- Model (descriptive):
  $\{(X_k, Y_k) : P(X_k, Y_k) \geqslant sup_{min}$ and $P(Y_k | X_k) \geqslant conf_{min}\}_{k=1}^{K}$
  - X_k and Y_k are sets of items
  - sup_{min} (minimum support) and $conf_{min}$ (minimum confidence) are user-specified constants
- Parameters: K, (X_1, Y_1), (X_2, Y_2), $\cdots$, and (X_K, Y_K)
- Objective: $\arg_{K,(X_1,Y_1),\cdots,(X_K,Y_K)} \max K$ such that $P(X_k, Y_k) \geqslant sup_{min}$ and $P(Y_k | X_k) \geqslant conf_{min}$ for any $1 \leqslant k \leqslant K$

Outline

ML vs. AI vs. Data Mining

- AI emulates human brains (e.g., playing chess)
    - ML is a branch of AI that emulates the "learning" tasks
- Data Mining (DM) overlaps ML a lot
    - Named differently by different groups of people
    - Arguably, ML pays more attention on theoretical aspects of models;
    - while DM cares more about practical aspects (e.g., data integration, scalability, etc.)

Outline

1. What is Machine Learning?
2. Learning Tasks
3. ML vs. AI vs. Data Mining
4. **About This Course**
5. FAQ

Target Audience

- Presents a consistent treatment of some selected machine learning problems and solutions
    - We go deep! So this is ***not*** a motivating course

Caution!

This course is highly theoretical

- You will ***not*** learn how to write a complete, intelligent system
- But you will have chances to try out individual ML algorithms using Matlab (or R)

Topics Covered (Incomplete)

Steps\Tasks	Classification	Regression	Latent Var.	Graph Ana.	Decision
Data Prep.	Imputation, Dim. Reduction (PCA)				
Model & Train.	SVM	Reg. LS	Spectral	Manifold Reg.	Tensor Dec.
	ML/MAP/Bayesian		EM/LDA	Markov Models	Reinforcement
Testing	Cross Validation, ROC/AUC, Entropy/Purity				

- We focus on probabilistic methods this year
    - Heavily rely on calculus and probability
- Geometric methods will NOT be covered

Grading

- Midterm exam: 30%
- Final exam: 30%
- Assignments (& presentations): 30%
  - You may be assigned extra readings in order to follow up the next lecture or to do your homework
- ***High-performance rewards: 10%***

Information

- Tue: 10:10am ~ ***12:00pm***
- Thu: 9:00am ~ ***12:00pm***
    - Extra time for TAs (if necessary)

The class may end late

Don't take this course if you are not satisfied with it!

- More information can be found in my web page: http://www.cs.nthu.edu.tw/~shwu/

Outline

FAQ

- Q: *Do I need to write programs in this course?*
  A: Yes, using Matlab or R. Prepare them yourself
- Q: *Do we need to come to the class?*
  A: No, as long as you can pass
- Q: *Is this a light-loading class or heavy-loading class?*
  A: Should be ***very heavy*** to most students. Our experience tells us that 2 to 6 hours per week is a must. ***Reserve your time, or you will have high chance to fail!***
- Q: *How often will the assignments be given?*
  A: Every 1 to 3 weeks (tentatively)
- Q: *I am not good in math. Can I take this course?*
  A: This course tries to be self-content in math. So, hard-working pays off. Evaluate yourself in the prerequisite exam

TODO

- Assigned Reading: Appendices A and B, except those sections marked **
- Register your seat in the next class