Introduction to Machine Learning

Milos Hauskrecht
milos@cs.pitt.edu
5329 Sennott Square, x4-8845

people.cs.pitt.edu/~milos/courses/cs1675/

Administration

Instructor:
Prof. Milos Hauskrecht
milos@cs.pitt.edu
5329 Sennott Square, x4-8845

TA:
Amin Sobhani
ams543@pitt.edu
6804 Sennott Square

Office hours: TBA
Who am I?

- Milos Hauskrecht – Professor of Computer Science
- Secondary affiliations:
  - Intelligent Systems Program (ISP),
  - Department of Biomedical Informatics (DBMI)
- Research work:
  - Machine learning, Data mining, Outlier detection, Probabilistic modeling, Time-series models and analysis

Applications to healthcare:
- EHR data analysis, Patient monitoring and alerting, Patient safety

Administration

Study material
- Handouts, your notes and course readings
- Primary textbook:
Administration

Study material

- **Other books:**
  - J. Han, M. Kamber. Data Mining. Morgan Kauffman, 2011.

Homeworks:

- **Programming tool:** Matlab (free license, CSSD machines and labs)
  - **Matlab Tutorial:** next week

Exams:

- **Midterm + Final**
- **Midtem** – the week just before Spring break

Lectures:

- **Attendance and Activity**
Tentative topics

- Introduction to Machine Learning
- **Density estimation.**
- **Supervised Learning.**
  - Linear models for regression and classification.
- **Unsupervised Learning.**
  - Learning Bayesian networks.
  - Latent variable models. Expectation maximization.
  - Clustering

---

Tentative topics (cont)

- **Dimensionality reduction.**
  - Feature extraction.
  - Principal component analysis (PCA)
- **Ensemble methods.**
  - Mixture models.
  - Bagging and boosting.
- **Reinforcement learning**
Machine Learning

- The field of **machine learning** studies the design of computer programs (agents) capable of learning from past experience or adapting to changes in the environment.

- The need for building agents capable of learning is everywhere:
  - text, web page, image classification
  - web search
  - speech recognition
  - Image/video annotation and retrieval
  - adaptive interfaces
  - commercial software

---

Learning

**Learning process:**

Learner (a computer program) processes data $D$ representing past experiences and tries to either develop an appropriate response to future data, or describe in some meaningful way the data seen.

**Example:**

Learner sees a set of patient cases (patient records) with corresponding diagnoses. It can either try:

- to predict the occurrence of a disease for future patients
- describe the dependencies between diseases, symptoms
Types of learning problems

- **Supervised learning**
  - Takes data that consists of pairs \((x, y)\)
  - Learns mapping \(f: x \rightarrow y\) (output, response)

- **Unsupervised learning**
  - Takes data that consist of vectors \(x\)
  - Learns relations \(x\) among vector components
  - Groups/clusters data into the groups

- **Reinforcement learning**
  - Learns mapping \(f: x \rightarrow y\) (desired output)
  - From \((x, y, r)\) triplets where \(x\) is an input, \(y\) is a response chosen by the user/system, and \(r\) is a reinforcement signal
  - **Online**: see \(x\), choose \(y\) and observe \(r\)

- **Other types of learning**: Active learning, Transfer learning, Deep learning

---

**Supervised learning**

**Data:** \(D = \{d_1, d_2, ..., d_n\}\) a set of \(n\) examples

\[ d_i = \langle x_i, y_i \rangle \]

\(x_i\) is input vector, and \(y\) is desired output (given by a teacher)

**Objective:** learn the mapping \(f: X \rightarrow Y\)

\[ y_i \approx f(x_i) \quad \text{for all} \quad i = 1,..,n \]

**Two types of problems:**

- **Regression:** \(X\) discrete or continuous \(\rightarrow\)
  - \(Y\) is **continuous**
- **Classification:** \(X\) discrete or continuous \(\rightarrow\)
  - \(Y\) is **discrete**
Supervised learning examples

- **Regression:** Y is *continuous*

  Debt/equity
  Earnings
  Future product orders
  →
  company stock price

- **Classification:** Y is *discrete*

  Handwritten digit (array of 0,1s)
  →
  Label “3”

Unsupervised learning

- **Data:** \( D = \{d_1, d_2, \ldots, d_n\} \)

  \( d_i = \mathbf{x}_i \)  vector of values

  No target value (output) y

- **Objective:**
  - learn relations between samples, components of samples

Types of problems:

- **Clustering**
  Group together “similar” examples, e.g. patient cases

- **Density estimation**
  - Model probabilistically the population of samples
Unsupervised learning example

• **Clustering.** Group together similar examples \( d_i = x_i \)
Unsupervised learning example

- **Density estimation.** We want to build a probability model \( P(x) \) of a population from which we draw examples \( d_i = x_i \).

Unsupervised learning. Density estimation

- A probability density of a point in the two dimensional space
  - Model used here: **Mixture of Gaussians**
Reinforcement learning

We want to learn: \( f : X \rightarrow Y \)

- We see examples of inputs \( x \) but not \( y \)
- We select \( y \) for observed \( x \)
- We get a feedback (reinforcement) from a critic about how good our choice of \( y \) was

- The goal is to select outputs that lead to the best reinforcement

---

Learning: first look

- Assume we see examples of pairs \((x, y)\) in \( D \) and we want to learn the mapping \( f : X \rightarrow Y \) to predict \( y \) for some future \( x \)
- We get the data \( D \) - what should we do?
Learning: first look

• **Problem:** many possible functions \( f : X \rightarrow Y \) exists for representing the mapping between \( x \) and \( y \)
• Which one to choose? Many examples still unseen!

![Graph](image)

Learning: first look

• **Solution:** make an assumption about the model, say,

\[
\begin{align*}
  f(x) &= ax + b
\end{align*}
\]
Learning: first look

• Choosing a parametric model or a set of models is not enough
Still too many functions \( f(x) = ax + b \)
  – One for every pair of parameters \( a, b \)

Fitting the data to the model

• We want the best set of model parameters

Objective: Find parameters that:
• reduce the misfit between the model \( M \) and observed data \( D \)
• Or, (in other words) explain the data the best

Objective function:
• Error function: Measures the misfit between \( D \) and \( M \)
• Examples of error functions:
  – Average Square Error \( \frac{1}{n} \sum_{i=1}^{n} (y_i - f(x_i))^2 \)
  – Average misclassification error \( \frac{1}{n} \sum_{i=1}^{n} 1_{y_i \neq f(x_i)} \)

Average # of misclassified cases
Fitting the data to the model

- **Linear regression problem**
  - Minimizes the squared error function for the linear model
    \[ \frac{1}{n} \sum_{i=1}^{n} (y_i - f(x_i))^2 \]

Supervised learning: Regression

- **Application**: A new example \( x \) with unknown value \( y \) is checked against the model, and \( y \) is calculated
  \[ y = f(x) = ax + b \]
Supervised learning: Classification

• Data D: pairs \((x, y)\) where \(y\) is a class label:
  
y examples: patient will be readmitted or no, has disease (case) or no (control)

Supervised learning: Classification

• Find a model \(f: X \rightarrow \mathbb{R}\), say \(f(x) = ax_1 + bx_2 + c\) that defines a decision boundary \(f(x) = 0\) that separates well the two classes
  – Note that some examples are not correctly classified
Supervised learning: Classification

• A new example x with unknown class label is checked against the model, the class label is assigned

Learning: summary

Three basic steps:

• Select a model or a set of models (with parameters)
  E.g. \( f(x) = ax + b \)

• Select the error function to be optimized
  E.g. \( \frac{1}{n} \sum_{i=1}^{n} (y_i - f(x_i))^2 \)

• Find the set of parameters optimizing the error function
  – The model and parameters with the smallest error represent the best fit of the model to the data

But there are problems one must be careful about …