Some slides from Machine Learning book by E. Alpaydin; or from the slides of Machine Learning book by T. Mitchell; or Gutierrez-Osuna.
Outline

- Course overview
- What is machine learning, why learn...?
- Concepts:
  - train, test sets, learning, estimation,
  - error measures,
  - generalization, overfitting,
  - ...
- Introduction to regression and classification
  - Linear regression
  - Decision tree classification
DA514-Machine Learning

Please refer to SUCourse for:

- Announcements
  - all important ones will be also emailed to your registered addresses
- Assignments
- Discussion boards
  - Please use the board to discuss anything about the course, except for sharing main parts of your homework
- Resources: lectures, data, homeworks, ...
- Schedule
  - will contain hw etc deadlines
Tentative Syllabus

1. Introduction - Machine Learning concepts (3hrs)
2. Inductive Learning Fundamentals (1hr)
3. Decision Trees (3hrs), overfitting
   - Hw1
4. Probability Basics, Bayesian Decision Criteria & Naive Bayes (4-5 hrs)
5. Multivariate Normal Distribution & Parameter Estimation (3-4 hrs)
   - Hw2
6. Nearest Neighbour Classifiers & Distance Metrics (3hrs)
7. Preprocessing & Dimensionality Reduction (3hrs)
8. Neural Networks & Gradient Descent (4hrs)
   - Hw3
9. Clustering (3hrs)
10. Support Vector Machines (3hrs)
11. Classifier Combination (4 hrs) incl. Bias and Variance
12. Final exam (3hrs – in class)

Total: 42 hrs
Scheduled 36-38 hours, rest for flexibility.
There should be little changes from this schedule.
Grading

- %50 in-class final
  - covering the fundamentals
  - must have 30/100 to pass the course

- %30 homeworks (3 homeworks)
  - randomly/partly graded

- %20 project
  - last 2-3 weeks
    - Feb 6, Feb 8
    - Feb 13, Feb 17
What is Learning...
What is learning?

It is very difficult to pin down what learning is. As our programs/agents/robots start doing more and more things, we do not find them intelligent anymore.

Some definitions from famous people in CS/AI history:

- “Learning denotes changes in a system that ... enable a system to do the same task more efficiently the next time.” – Herbert Simon

- “Learning is any process by which a system improves performance from experience.” – Herbert Simon

- “Learning is constructing or modifying representations of what is being experienced.” – Ryszard Michalski

- “Learning is making useful changes in our minds.” – Marvin Minsky
Why learn?

- Build software **agents that can adapt** to their users or to other software agents or to changing environments
  - Mars robot

- Develop **systems that are too difficult/expensive to construct manually** because they require specific detailed skills or knowledge tuned to a specific task
  - Large, complex AI systems cannot be completely derived by hand and require dynamic updating to incorporate new information.

- **Discover new things** that were previously unknown to humans
  - Examples: data mining, scientific discovery
Tasks too hard to program

ALVINN [Pomerleau] drives 70 MPH on highways
Related Disciplines

The following are close disciplines:

- **Artificial Intelligence**
  - Machine learning deals with the learning part of AI
- **Pattern Recognition**
  - Concentrates more on “tools” rather than theory
- **Data Mining**
  - More specific about discovery

The following are useful in machine learning techniques or may give insights:

- **Probability and Statistics**
- **Information theory**
- **Psychology (developmental, cognitive)**
- **Neurobiology**
- **Linguistics**
- **Philosophy**
History of Machine Learning

- **1950s**
  - Samuel’s checker player
  - Selfridge’s Pandemonium

- **1960s:**
  - Neural networks: Perceptron
  - Minsky and Papert prove limitations of Perceptron

- **1970s:**
  - Expert systems and the knowledge acquisition bottleneck
  - Mathematical discovery with AM
  - Symbolic concept induction
  - Winston’s arch learner
  - Quinlan’s ID3
  - Michalski’s AQ and soybean diagnosis
  - Scientific discovery with BACON
1980s:
- Resurgence of neural networks (connectionism, backpropagation)
- Advanced decision tree and rule learning
- Explanation-based Learning (EBL)
- Learning, planning and problem solving
- Utility theory
- Analogy
- Cognitive architectures
- Valiant’s PAC Learning Theory

1990s
- Data mining
- Reinforcement learning (RL)
- Inductive Logic Programming (ILP)
- Ensembles: Bagging, Boosting, and Stacking
History of Machine Learning (cont.)

- 2000s
  - **Kernel methods**
    - Support vector machines
  - **Graphical models**
  - Statistical relational learning
  - Transfer learning

- Applications
  - Adaptive software agents and web applications
  - Learning in robotics and vision
  - E-mail management (spam detection)
  - …
Different Learning Paradigms
Major paradigms of machine learning

- **Rote learning** – “Learning by memorization.”
  - Employed by first machine learning systems, in 1950s
    - Samuel’s Checkers program

- **Supervised learning** – Use specific examples to reach general conclusions or extract general rules
  - Classification
  - Regression

- **Unsupervised learning (Clustering)** – Unsupervised identification of natural groups in data

- **Reinforcement learning** – Feedback (positive or negative reward) given at the end of a sequence of steps

- **Analogy** – Determine correspondence between two different representations

- **Discovery** – Unsupervised, specific goal not given

...
Rote Learning is Limited

- Memorize I/O pairs and perform exact matching with new inputs

- If a computer has not seen the precise case before, it cannot apply its experience

- We want computers to “generalize” from prior experience
  - Generalization is the most important factor in learning
The inductive learning problem

- Extrapolate from a given set of examples to make accurate predictions about future examples

- Supervised versus unsupervised learning
  - Learn an unknown function $f(X) = Y$, where $X$ is an input example and $Y$ is the desired output.
  - **Supervised learning** implies we are given a training set of $(X, Y)$ pairs by a “teacher”
  - **Unsupervised learning** means we are only given the Xs.
  - **Semi-supervised learning**: mostly unlabelled data
Classification, Regression
Types of supervised learning: Classification

a) Classification:
- We are given the label of the training objects: \{(x_1,x_2,y=T/O)\}
- We are interested in classifying future objects: \((x_1',x_2')\) with the correct label.
  I.e. Find \(y'\) for given \((x_1',x_2')\).

b) Concept Learning:
- We are given positive and negative samples for the concept we want to learn (e.g. Tangerine): \{(x_1,x_2,y=+/\})
- We are interested in classifying future objects as member of the class (or positive example for the concept) or not.
  I.e. Answer +/- for given \((x_1',x_2')\).
Classification

- Assign object/event to one of a given finite set of categories.
  - Medical diagnosis
  - Credit card applications or transactions
  - Fraud detection in e-commerce
  - Spam filtering in email
  - Recommended books, movies, music
  - Financial investments
  - Spoken words
  - Handwritten letters
The task of a classifier is to partition feature space into class-labeled decision regions

- Borders between decision regions are called **decision boundaries**
- The classification of feature vector $x$ consists of determining which decision region it belongs to, and assign $x$ to this class
Types of Supervised Learning: Regression

- Target function is **continuous** rather than class membership
  - E.g. price of a house as a “function” of its size
- Linear or non-linear mappings may be considered.
- Multiple attributes would create a more precise mapping.

![Graph showing regression relationship between size and price](image-url)
Learning: Key Steps

- data and assumptions
  - what data is available for the learning task?
  - what can we assume about the problem?

- representation
  - how should we represent the examples to be classified

- method and estimation
  - what are the possible hypotheses?
  - what learning algorithm to use to infer the most likely hypothesis?

- evaluation
  - how well are we doing?
Feature

- Feature is any **distinctive** aspect, quality or characteristic
  - Features may be symbolic (i.e., color) or numeric (i.e., height)

- Definitions
  - The combination of $d$ features is represented as a $d$-dimensional column vector called a **feature vector**
  - The $d$-dimensional space defined by the feature vector is called the **feature space**
  - Objects are represented as points in feature space. This representation is called a **scatter plot**

\[
X = \begin{bmatrix}
X_1 \\
X_2 \\
X_d
\end{bmatrix}
\]
What makes a “good” feature vector?

- The quality of a feature vector is related to its ability to discriminate examples from different classes
  - Examples from the same class should have similar feature values
  - Examples from different classes have different feature values

More feature properties

- Linear separability
- Non-linear separability
- Highly correlated features
- Multi-modal
EVALUATION OF LEARNING ALGORITHMS
Evaluation of Learning Systems

- **Experimental**
  - Conduct controlled cross-validation experiments to compare various methods on a variety of benchmark datasets.
  - Gather data on their performance, e.g. test accuracy, training-time, testing-time…
  - Maybe even analyze differences for statistical significance.

- **Theoretical**
  - Analyze algorithms mathematically and prove theorems about their:
    - Ability to fit training data
    - Computational complexity
    - Sample complexity (number of training examples needed to learn an accurate function)
Measuring Performance

Performance of the learner can be measured in one of the following ways, as suitable for the application:

- **Accuracy**
  - Number of mistakes (in *classification* problems)
  
  - Mean Squared Error (in *regression* problems)
    \[
    \text{MSE} = \frac{1}{n} \sum_{i=1}^{n} (\hat{Y}_i - Y_i)^2
    \]
  
  - Loss functions (more general, taking into account different costs for different mistakes)

- **Solution quality** (length, efficiency)
- **Speed of performance**
- ...
Introduction to Overfitting
Overfitting and Model Complexity

- Imagine that we have some training data (blue dots in the next slides), and we want to learn the underlying function between the independent variable $x$ and the target values $t$.

- We can fit polynomials in varying degrees: lines to higher degree polynomials.

- Higher degrees make the polynomial very capable to bend/flex to match the data as it has many parameters to change/adapt.

- However having zero train error does not mean the model (high order polyn.) will also have high generalization performance.

- In fact, a simpler model that has a similar performance compared to a more complex model, is often preferred (more on these later and more formally).
Polynomial Curve Fitting

\[ y(x, \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + \ldots + w_M x^M = \sum_{j=0}^{M} w_j x^j \]
Sum-of-Squares Error Function

\[ E(w) = \frac{1}{2} \sum_{n=1}^{N} \{y(x_n, w) - t_n\}^2 \]
$0^{th}$ Order Polynomial

$M = 0$
1st Order Polynomial

\[ M = 1 \]
3rd Order Polynomial
9\textsuperscript{th} Order Polynomial

\begin{align*}
  t & \quad M = 9 \\
  x & \quad 0 \quad 1
\end{align*}
We do not know yet which is the best model, maybe the 9\textsuperscript{th} degree polynomial after all.
This was an introduction of the concepts, we will cover each issue more in depth in the coming weeks.

So you will get to ask more questions.
What you should know

- why machine learning
- classification and regression
- supervised learning, inductive learning frameworks
- features
- train, test, validation sets, unbiased testing
- generalization, overfitting
- error measures (MSE, misclassification...)
- cross-validation: k-fold, leave-one-out
- ...

...