Machine Learning
CS 4900/5900

Lecture 01

Razvan C. Bunescu
School of Electrical Engineering and Computer Science

bunescu@ohio.edu
What is (Human) Learning?

• Merriam-Webster:
  – learn = to acquire knowledge, understanding, or skill … by study, instruction, or experience.

• Why do we learn?
  – to improve performance on a given task.

• What (tasks) do we learn:
  1. categorize email, recognize faces, diagnose diseases, translate, …
  2. clustering (fish, insects, birds, mice, humans), summarization, sound source separation, …
  3. walk, play backgammon, ride bikes, drive cars, fly helicopters, …
What is Machine Learning?

- **Machine Learning** = constructing computer programs that *automatically improve with experience*:
  - **Supervised Learning** i.e. learning from labeled examples.
What is Learning?
Occam’s Razor

William of Occam (1288 – 1348)
- English Franciscan friar, theologian and philosopher.

- “Entia non sunt multiplicanda praeter necessitatem”
  - Entities must not be multiplied beyond necessity.

i.e. Do not make things needlessly complicated.
i.e. Prefer the simplest hypothesis that fits the data.
What is Learning?

Class $C_1$

Class $C_2$
What is Learning?

Class $C_1$

Class $C_2$

Lecture 01
What is Learning?

Class $C_1$

Class $C_2$
What is Learning?
What is Learning?

Class $C_1$

Class $C_2$
ML Concepts & Notation

• A (labeled) example \((x, t)\) consists of:
  – Instance / observation / raw feature vector \(x\).
  – Label \(t\).

• Examples:
  1. Digit recognition:
     
     ![Digit Recognition Example](2)
     
     instance \(x = ?\)
     
     label \(t = ?\)

  2. Language modelling:
     
     ![Language Modelling Example](3)
     
     “machine .......... is a hot topic in AI”
     
     instance \(x = ?\)
     
     label \(t = ?\)
ML Concepts & Notation

• Often, a raw observation \( x \) is pre-processed and further transformed into a feature vector \( \varphi(x) = [\varphi_1(x), \varphi_1(x), \ldots, \varphi_K(x)]^T \).
  – Where do the features \( \varphi_k \) come from?
    • Feature engineering, e.g. in polynomial curve fitting:
      – manual, can be time consuming (e.g. SIFT).
    • (Unsupervised) feature learning, e.g. in modern computer vision
      – automatic, used in deep learning models.
ML Concepts & Notation

• A training dataset is a set of (training) examples $(x_1,t_1)$, $(x_2,t_2)$, … $(x_N,t_N)$:
  - The data matrix $X$ contains all instance vectors $x_1$, $x_2$, …, $x_N$ row-wise.
  - The label vector $t = [t_1, t_2, …, t_N]^T$.

• A test dataset is a set of (test) examples $(x_{N+1},t_{N+1})$, …, $(x_{N+M},t_{N+M})$:
  - Must be different from the training examples!
ML Concepts & Notation

• There is a function $f$ that maps an instance $x$ to its label $t = f(x)$.
  – $f$ is unknown / not given.
  – But we observe samples from $f$: $(x_1, t_1), (x_2, t_2), \ldots, (x_N, t_N)$.

• Learning means finding a model $h$ that maps an instance $x$ to a label $h(x) \approx f(x)$, i.e. close to the true label of $x$.
  – Machine learning = finding a model $h$ that approximates well the unknown function $f$.
  – Machine learning = function approximation!
ML Concepts & Notation

• Machine learning is **inductive**:
  – **Inductive hypothesis**: if a model performs well on training examples, it is expected to also perform well on unseen (test) examples.

• The **model** $y$ is often specified through a set of parameters $w$:
  – $x$ is mapped by the model to $h(x, w)$.

• The **objective function** $J(w)$ captures how poorly the model does on the training dataset:
  – Want to find $\hat{w} = \text{argmin}_w J(w)$
  – **Machine learning = optimization!**
Fitting vs. Generalization

• **Fitting** performance = how well the model performs on training examples.

• **Generalization** performance = how well the model performs on unseen (test) examples.

• We are interested in **Generalization**:
  – Prefer finding patterns to memorizing examples!
    • Overfitting:
    • Regularization:
Supervised Learning

Training

Training Examples \((x_k, t_k)\) \rightarrow \text{Learning Algorithm} \rightarrow \text{Model } h

Testing

Model \(h\) \rightarrow \text{Test Examples } (x, t) \rightarrow \text{Generalization Performance}
Machine Learning vs. Deep Learning

\[ x \xrightarrow{h} h(x,w) \]

\[ x \xrightarrow{\varphi} h(\varphi(x),w) \]

\[ x \xrightarrow{\varphi_1,\varphi_2,...,\varphi_K} h(\varphi_{1,K}(x),w) \]
What is Machine Learning?

• **Machine Learning** = constructing computer programs that *automatically improve with experience*:
  – **Supervised Learning** i.e. learning from labeled examples:
    • Classification
    • Regression
  – **Unsupervised Learning** i.e. learning from unlabeled examples:
    • Clustering.
    • Dimensionality reduction (visualization).
    • Density estimation.
  – **Reinforcement Learning** i.e. learning with delayed feedback.
Supervised Learning

• Task = learn a function $f : X \to T$ that maps input instances $x \in X$ to output targets $t \in T$:
  - **Classification**:
    • The output $t \in T$ is one of a finite set of discrete categories.
  - **Regression**:
    • The output $t \in T$ is continuous, or has a continuous component.

• Supervision = set of training examples:
  $$(x_1,t_1), (x_2,t_2), \ldots, (x_n,t_n)$$
Classification vs. Regression
Classification: Junk Email Filtering

[Sahami, Dumais & Heckerman, AAAI’98]

From: Tammy Jordan
jordant@oak.cats.ohiou.edu
Subject: Spring 2015 Course

CS690: Machine Learning
Instructor: Razvan Bunescu
Email: bunescu@ohio.edu
Time and Location: Tue, Thu 9:00 AM, ARC 101
Website: http://ace.cs.ohio.edu/~razvan/courses/ml6830

Course description:
Machine Learning is concerned with the design and analysis of algorithms that enable computers to automatically find patterns in the data. This introductory course will give an overview …

From: UK National Lottery
edreyes@uknational.co.uk
Subject: Award Winning Notice

---

UK NATIONAL LOTTERY. GOVERNMENT ACCREDITED LICENSED LOTTERY.
REGISTERED UNDER THE UNITED KINGDOM DATA PROTECTION ACT;

We happily announce to you the draws of ( UK NATIONAL LOTTERY PROMOTION ) International programs held in London, England Your email address attached to ticket number: 3456 with serial number: 7576/06 drew the lucky number 4-2-274, which subsequently won you the lottery in the first category …

• Email filtering:
  – Provide emails labeled as \(\{\text{Spam, Ham}\}\).
  – Train \textit{Naïve Bayes} model to discriminate between the two.

Lecture 01
• Link quality prediction:
  – Provide a set of training links:
    • received signal strength, send/forward buffer sizes
    • node depth from base station, forward/backward probability
      o LQI = Link Quality Indication, binarized as \{Good, Bad\}
  – Train \textit{Decision Trees} model to predict LQ using runtime features.
Classification: Handwritten Zip Code Recognition

[Le Cun et al., Neural Computation ‘89]

- Handwritten digit recognition:
  - Provide images of handwritten digits, labeled as \{0, 1, \ldots, 9\}.
  - Train *Neural Network* model to recognize digits from input images.
Classification: Medical Diagnosis

[Krishnapuram et al., GENSIPS’02]

• Cancer diagnosis from gene expression signatures:
  – Create database of gene expression profiles (X) from tissues of known cancer status (Y):
    • Human acute leukemia dataset:
      – http://www.broadinstitute.org/cgi-bin/cancer/datasets.cgi
    • Colon cancer microarray data:
      – http://microarray.princeton.edu/oncology
  – Train Logistic Regression / SVM / RVM model to classify the gene expression of a tissue of unknown cancer status.
ML for Software Verification / ATP

• Software verification requires theorem proving.

• Proving a mathematical theorem requires finding and using relevant previous theorems and definitions:
  – The space of existing theorems and definitions is huge.
  – Use machine learning to narrow the search space to relevant theorems and definitions:
    • “Premise Selection for Mathematics by Corpus Analysis and Kernel Methods”, Alama et al., JAR 2012.
Software Verification / ATP for ML

• An ML model is a program i.e. ML algorithms induce programs.

• ML models such as (Deep) Neural Networks may lack robustness:
  – Adversarial methods can fool the networks, e.g. classifying a school bus as an ostrich.
  – Software verification / ATP methods can be used to prove an ML model has (or not) desirable properties:
    • “Reluplex: An Efficient SMT Solver for Verifying Deep Neural Networks”, Katz et al., CAV 2017
      – Used to prove adversarial robustness for collision avoidance system for unmanned aircraft.
Classification: Other Examples

- Handwritten letter recognition
- Face recognition
- Credit card applications/transactions
- Recommender systems: books, music, …
- Fraud detection in e-commerce
- Worm detection in network packets
- Tone recognition
- Chord Recognition
- Named Entity Recognition
Regression: Examples

1. Stock market prediction:
   – Use the current stock market conditions \((x \in X)\) to predict tomorrow’s value of a particular stock \((t \in T)\).

2. Oil price, GDP, income prediction.

3. Chemical processes:
   – Predict the yield in a chemical process based on the concentrations of reactants, temperature and pressure.

• Algorithms:
   – *Linear Regression, Neural Networks, Support Vector Machines*, …
Unsupervised Learning: Hierarchical Clustering

Pan Troglodytes

Homo Sapiens
Unsupervised Learning: Clustering

- Partition unlabeled examples into disjoint clusters such that:
  - Examples in the same cluster are very similar.
  - Examples in different clusters are very different.
Unsupervised Learning: Clustering

• Partition unlabeled examples into disjoint clusters such that:
  – Examples in the same cluster are very similar.
  – Examples in different clusters are very different.

• Need to provide:
  – number of clusters \( k = 2 \)
  – similarity measure (Euclidean)
Unsupervised Learning: Dimensionality Reduction

- Manifold Learning:
  - Data lies on a low-dimensional manifold embedded in a high-dimensional space.
  - Useful for feature extraction and visualization.
Reinforcement Learning

- Interaction between agent and environment modeled as a sequence of actions & states:
  - Learn policy for mapping states to actions in order to maximize a reward.
  - Reward given at the end state => delayed reward.
  - States may be only partially observable.
  - Trade-off between exploration and exploitation.

- Examples:
  - Backgammon [Tesauro, CACM’95].
  - Aerobatic helicopter flight [Abbeel, NIPS’07].
  - 49 Atari games, using deep RL [Mnih et al., Nature’15].
Reinforcement Learning: TD-Gammon

• Learn to play Backgammon:
  – Immediate reward:
    • +100 if win
    • −100 if lose
    • 0 for all other states
  – *Temporal Difference Learning* with a *Multilayer Perceptron*.
  – Trained by playing 1.5 million games against itself.
  – Played competitively against top-ranked players in international tournaments.

[Tesauro, CACM‘95]
Relevant Disciplines

• Mathematics:
  – Probability & Statistics
  – Information Theory
  – Linear Algebra
  – Optimization

• Algorithms:
  – Computational Complexity

• Artificial Intelligence
  – Search

• Psychology

• Neurobiology
Readings

• PRML 1.2, 2.1 – 2.1.1, 2.2 – 2.2.1, 2.3 (2.3.4, 2.3.9).
• PRML Appendix B and C.