

# Machine Learning - Introduction & Overview

Md. Mohsin Uddin

East West University

*mmuddin@ewubd.edu*

April 29, 2019

- Slides borrowed from Pedro Domingos and Luke Zettlemoyer, University of Washington

# A Few Quotes

- “A breakthrough in machine learning would be worth ten Microsofts” (Bill Gates, Microsoft)
- “Machine learning is the next Internet” (Tony Tether, Director, DARPA)
- “Machine learning is going to result in a real revolution” (Greg Papadopoulos, Ex CTO, Sun)

# Machine Learning

- Automating automation
- Getting computers to program themselves
- Writing software is the bottleneck
- Let the data do the work instead!

# Sample Applications

- Web search
- Computational biology
- Finance
- E-commerce
- Space exploration
- Robotics
- Information extraction
- Social networks

# Growth of Machine Learning

- Improved machine learning algorithms
- Improved data capture, networking, faster computers, big data
- Software too complex to write by hand
- New sensors/IO devices
- Demand for self customization to user and environment

- Tens of thousands of machine learning algorithms
- Hundreds new every year
- Every machine learning algorithm has three components:
  - 1 Representation
  - 2 Evaluation
  - 3 Optimization

# Representation

- Decision trees
- Sets of rules / Logic programs
- Instances
- Graphical models (Bayes/Markov nets)
- Neural networks
- Support vector machines
- Model ensembles
- Etc.

# Evaluation

- Accuracy
- Precision and recall
- Squared error
- Likelihood
- Posterior probability
- Cost / Utility
- Margin
- Entropy
- K-L divergence
- Etc.



- Combinatorial optimization E.g.: Greedy search
- Convex optimization E.g.: Gradient descent
- Constrained optimization E.g.: Linear programming

# Types of Learning

- **Supervised (inductive) learning** - Training data includes desired outputs
- **Unsupervised learning** - Training data does not include desired outputs
- **Semi-supervised learning** - Training data includes a few desired outputs
- **Reinforcement learning** - Rewards from sequence of actions.

# Supervised Learning

- Given examples of a function  $(X, F(X))$
- Predict function  $F(X)$  for new examples  $X$ 
  - ① Discrete  $F(X)$ : Classification
  - ② Continuous  $F(X)$ : Regression
  - ③  $F(X) = \text{Probability}(X)$ : Probability estimation

# Classification

Assign object/event to one of a given finite set of categories (discrete or categorical values).

- Medical diagnosis
- Credit card applications or transactions
- Fraud detection in e-commerce
- Worm detection in network packets
- Spam filtering in email
- Recommended articles in a newspaper
- Recommended books, movies, music, or jokes
- Financial investments
- DNA sequences
- Spoken words
- Handwritten letters
- Astronomical images

Predicting a numeric value (continuous value)

- Predicting weather
- Predicting stock market
- Predicting monthly or yearly product sales

## Example: Spam Filter

- **Input:** email
- **Output:** spam/ham
- **Setup:**
  - Get a large collection of example emails, each labeled “spam” or “ham”
  - Note: someone has to hand label all this data!
  - Want to learn to predict labels of new, future emails
- **Features:** The attributes used to make the ham / spam decision
  - Words: FREE!
  - Text Patterns: \$dd, CAPS
  - Non-text: SenderInContacts
  - ...



Dear Sir.

First, I must solicit your confidence in this transaction, this is by virtue of its nature as being utterly confidential and top secret. ...



TO BE REMOVED FROM FUTURE MAILINGS, SIMPLY REPLY TO THIS MESSAGE AND PUT "REMOVE" IN THE SUBJECT.

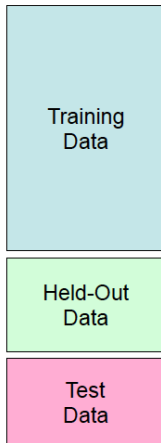
99 MILLION EMAIL ADDRESSES FOR ONLY \$99



Ok, I know this is blatantly OT but I'm beginning to go insane. Had an old Dell Dimension XPS sitting in the corner and decided to put it to use, I know it was working pre being stuck in the corner, but when I plugged it in, hit the power nothing happened.

# Important Concepts

- **Data:** labeled instances, e.g. emails marked spam/ham
  - Training set
  - Held out set (sometimes call Validation set)
  - Test set
- **Features:** attribute-value pairs which characterize each  $x$
- **Experimentation cycle**
  - Select a hypothesis  $f$  to best match training set
  - (Tune hyperparameters on held-out set)
  - Compute accuracy of test set
  - Very important: never “peek” at the test set!
- **Evaluation**
  - Accuracy: fraction of instances predicted correctly
- **Overfitting and generalization**
  - Want a classifier which does well on *test* data
  - Overfitting: fitting the training data very closely, but not generalizing well
  - We'll investigate overfitting and generalization formally in a few lectures



# Supervised and Unsupervised Learning Algorithms

- Supervised learning
  - ① Linear Regression
  - ② Logistic Regression
  - ③ Decision tree induction
  - ④ Bayesian learning
  - ⑤ Neural networks
  - ⑥ Support vector machines
  - ⑦ Model ensembles
- Unsupervised learning
  - ① Clustering (Grouping similar things)
  - ② Dimensionality reduction



# Unsupervised Learning

Predict structure:

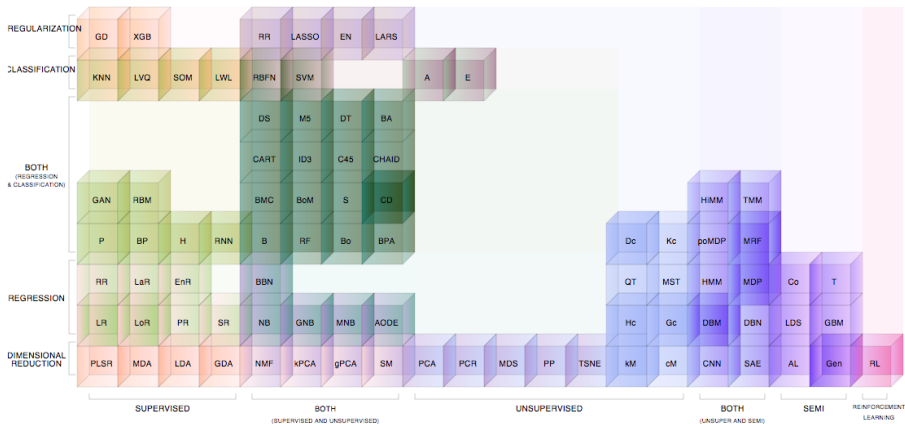
- Tree of life from DNA
- Find similar images
- Grouping similar things

- Understanding domain, prior knowledge, and goals
- Data integration, selection, cleaning
- Preprocessing
- Learning models
- Interpreting results
- Consolidating and deploying discovered knowledge
- Loop

# Learning Problem

- Definition: “A computer program is said to learn from experience  $E$  with respect to some task  $T$  and some performance measure  $P$ , if its performance on  $T$ , as measured by  $P$ , improves with experience  $E$ .”  
(Tom M. Mitchel)
- $T$ : Recognizing hand-written words  
 $P$ : Percentage of words correctly classified  
 $E$ : Database of human-labeled images of handwritten words
- $T$ : Driving on four-lane highways using vision sensors  
 $P$ : Average distance traveled before a human-judged error  
 $E$ : A sequence of images and steering commands recorded while observing a human driver.
- $T$ : Categorize email messages as spam or legitimate.  
 $P$ : Percentage of email messages correctly classified.  
 $E$ : Database of emails, some with human-given labels

# Periodic Table of ML Algorithms



# ML Algorithms : Part I

- PLSR: Partial Least Squares Regression
- MDA: MixtureDiscriminant Analysis
- LDA: Linear discriminant analysis
- GDA: Generalized discriminant analysis
- LR: Linear Regression
- LoR: Logistic Regression
- PR: Polynomial Regression
- SR: Stepwise Regression
- LaR: Lasso Regression
- EnR: ElasticNet Regression
- EnR: ElasticNet Regression
- P: Perceptron
- BP: Back-Propagation

# ML Algorithms : Part II

- H: Hopfield Network
- RBFN: Radial Basis Function Network
- KNN: K-Nearest Neighbour
- LVQ: Learning Vector Quantization
- SOM: Self-Organizing Map
- LWL: Locally Weighted Learning
- NMF: Non-negative Matrix Factorization
- kPCA: Kernel PCA
- gPCA: Graph-based kernel PCA
- SM: Sammon Mapping
- NB: Naive Bayes
- GNB: Gaussian Naive Bayes
- MNB: Multinomial Naive Bayes

# ML Algorithms : Part III

- AODE: Averaged One-Dependence Estimators
- BBN: Bayesian Belief Network
- CART: Classification and Regression Tree
- ID3: Iterative Dichotomiser 3
- C45: C45
- CHAID: Chi-squared Automatic Interaction Detection
- DS: Decision Stump
- M5: M5
- CD: Conditional Decision Trees
- B: Bayes optimal classifier
- RF: Random forests
- Bo: Boosting
- BPA: Bayesian parameter averaging

# ML Algorithms : Part IV

- BMC: Bayesian model combination
- BoM: Bucket of models
- S: Stacking
- RR: Ridge Regression
- LASSO: Least Absolute Shrinkage and Selection Operator
- EN: Elastic Net
- LARS: Least-Angle Regression
- PCA: Principal component analysis
- PCR: Principal Component Regression
- MDS: Multidimensional Scaling
- PP: Projection Pursuit
- A: Apriori algorithm
- E: Eclat algorithm



# ML Algorithms : Part V

- kM: K-means clustering algorithm
- cM: Fuzzy c-means clustering algorithm
- Hc: Hierarchical clustering algorithm
- Gc: Gaussian clustering algorithm
- QT: Quality Threshold clustering algorithm
- MST: MST based clustering algorithm
- Dc: Density based clustering algorithm
- Kc: Kernel k-means clustering algorithm
- HMM: Hidden Markov model
- MDP: Markov decision process
- poMDP: Partially observable Markov decision process
- MRF: Markov random field
- HiMM: Hierarchical Markov models

# ML Algorithms : Part VI

- TMM: Tolerant Markov model
- DBM: Deep Boltzman Machine
- DBN: Deep Belief Networks
- CNN: Convolution neural networks
- SAE: Stacked Auto-Encoders
- AL: Active learning
- Gen: Generative models
- LDS: Low-density separation
- GBM: Graph-based methods
- Co: Co-training
- T: Transduction
- RNN: Recurring Neural Network
- GAN: Generative Adversarial Network

# ML Algorithms : Part VII

- BA: Bootstrap Aggregating
- RBM: Restricted Boltzman Machine
- SVM: Support Vector Machines
- RL: Reinforcement Learning
- XGB: X-Gradient Boosting
- GD: Gradient Descent
- TSNE: t-Distributed Stochastic Neighbor Embedding
- DT: Decision Trees

# References



Christopher M. Bishop, Pattern recognition and Machine learning. Springer, 2006.



Tom Mitchell, Machine learning. McGraw-Hill, 1997



<https://refactored.ai/periodic-table>