CS 6316

Exam Study Guide

Machine Learning

The material below is a summary of general ideas and topics that may be covered on the exam. The topics listed are the highlights, and a high-level overview (not an exhaustive list). The exam is cumulative and will cover material since the beginning of the semester, including probability (but mainly the last three probability sections.)

Resources

The following resources supplement the material presented in class: :

- ✓ Readings (*textbook*) – on course schedule, available on Home tab of Collab page
- ✓ In-class activities
- ✓ In-class discussions
- ✓ Homework assignments / solutions / explanation (in terms of familiarity with the material not necessarily the same kind of questions – no coding questions on this exam)
- ✓ Additional/miscellaneous resources (*posted on Collab Resources*)

Material for the exam is presented starting on page 2 (more details are provided in the first few sections.)

Summary:

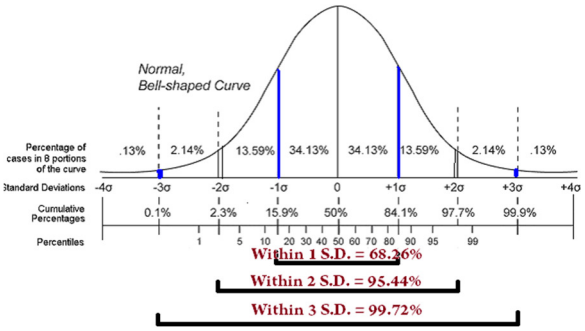
Material

❖ PREDICTIVE LEARNING [01]

- Uncertainty and learning
- Explanation and prediction
- Beliefs vs. true theories
  - Demarcation problem in philosophy – distinguishing between science and pseudoscience
- Explain the past and predict the future
  - Descriptive and Generalization
- Making decisions under uncertainly involves
  - Risk management
    - Apply decisions to known past events
    - Select one minimizing expected risk
  - Probabilistic approach
    - Estimate probabilities (of future events)
    - Assign costs and minimize expected risk
- Popular explanation (belief) vs. Classic first principle scientific explanation vs. Empirical knowledge
  - Beliefs: Non-empirical or *a priori* knowledge is possible independently of, or prior to, any experience, and requires only the use of reason
  - A first principle is a basic, foundational, self-evident proposition or assumption that cannot be deduced from any other proposition or assumption
  - Empirical knowledge: a belief that is learned by observing it using our *empirical knowledge*; e.g. sight, hearing, touch etc.
- General Experimental Procedure for Estimating Models from Data
- Types of prediction problems
  - Classification – assigning an object/item to a class
  - Regression – generalization of classification – output is real-valued
  - Clustering – organizing objects into meaningful groups
  - Description – representing an object in terms of a series of primitives – structural or linguistic description
- Features and patterns
  - Good/bad feature vectors
  - Separability (linear / non-linear), correlated features, multi-modal
- Review "Salmon vs Sea Bass" case study

❖ PROBABILITY [02]

- Abduction – incomplete set of observations → the likeliest possible explanation for the group of observations (making educated guess)
  - Doctors do this! Jurors do this! Sherlock Holmes too!
- Conditional Probability, Total probability, and Bayes Theorem
  - Conditional Probability $P(A|B) = \frac{P(A \cap B)}{P(B)}$, $P(B) > 0$
  - Total Probability // Bayes Theorem $P[B_j|A] = \frac{P[A \cap B_j]}{P[A]} = \frac{P[A|B_j]P[B_j]}{\sum_{k=1}^{N} P[A|B_k]P[B_k]}$
- Random Variables (CDF/PDF)
  - Cumulative Distribution Function
  - Probability Density Function
- Normal Distribution and Normal Curve
  - Properties of Normal Distribution



❖ LEARNING APPROACHES: CONCEPTS AND THEORY [03]

- Induction
- Deduction
- Data Pre-processing and Scaling, Outliers
- Learning System
  - Learning or estimation ("explanation of training data")
  - Test or prediction ("prediction of new (test) data")
- Supervised learning (regression / classification)
  - Quality of Prediction
    - Squared Loss (regression / classification)
    - Empirical Risk (quality of explanation / average training error / average fitting error)

$$R_{emp} = \frac{1}{n}\sum_{i=1}^{n} L(y_i, f(x_i))$$

- Prediction Risk (measure quality of prediction / average test error)

$$R = \frac{1}{T}\sum_{t=1}^{T} L(y_i, f(x_i))$$

T= number of test samples

- Does minimizing the training error improve prediction accuracy on new/testing data?
  - Generalization

Unsupervised learning

- Clustering
- Dimensionality Reduction
- Quality of Prediction
  - Loss function

$$L(y, f(x) = \|x - f(x)\|^2$$

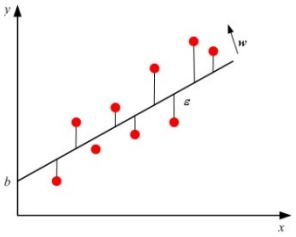
  - Goal: minimizing the squared distance between training points and their projections (mappings) onto a model space:

$$R_{emp} = \frac{1}{n}\sum_{i=1}^{n} L(x_i, f(x_i)) = \frac{1}{n}\sum_{i=1}^{n} \|x_i - f(x_i)\|^2$$

Basic Learning Approaches

- Parametric modeling
  - Specify parametric model
  - Estimate its parameters (via fitting to data)
  - Example: Linear regression $F(\boldsymbol{x}) = (\boldsymbol{w} . \boldsymbol{x}) + b$

$$\sum_{i=1}^{n} [y_i - (w\, x_i) - b]^2 \rightarrow min$$

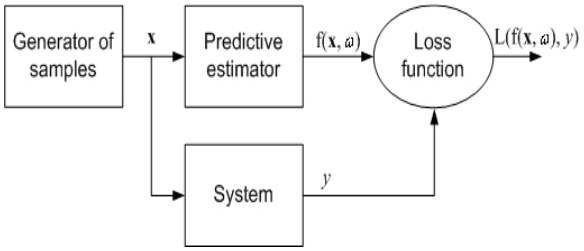


- Non-parametric modeling
  - Estimate the model for given $\boldsymbol{x}_0$ as "local average" of the data ("local estimation modeling") – example, k-NN

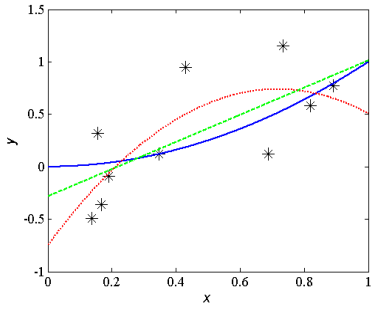
$$f(\boldsymbol{x}_0) = \frac{\sum_{j=1}^{k} y_j}{k}$$

- Data reduction
  - Compact encoding – example piece-wise linear regression

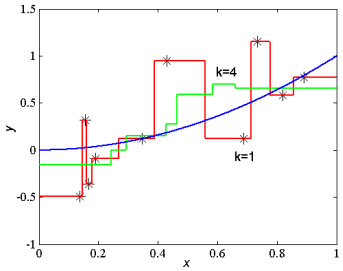
- Prediction Accuracy (generalization)
  - Explanation vs. Prediction (classification / regression)
  - Inductive learning setting



- Complexity Control
  - Parametric modeling / regression estimation



  - Local estimation / regression estimation



  - Complexity (of admissible models) affects generalization (for future data)
  - Complexity control = choosing good complexity (~ good generalization) for a given (training) data
  - Specific complexity indices for
    - Parametric models: ~ # of parameters
    - Local modeling: *size of local region*
    - Data reduction: # of clusters
  - Two ways to control complexity – one way is analytic approach the other is resampling approach
    - Analytic criteria estimate prediction error as a function of fitting error and model complexity
    - Representative analytic criteria for regression:

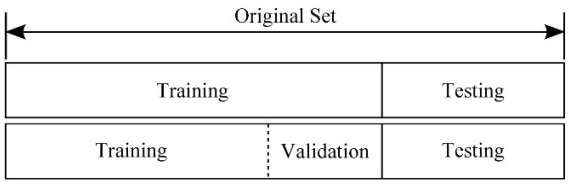
- Schwartz Criterion: $r(p,n) = 1 + p(1-p)^{-1} \ln n$
- Akaike's FPE: $r(p) = (1+p)(1-p)^{-1}$

  where p = $\frac{DoF}{n}$, n ~ sample size, DoF ~ degrees of freedom

- Predictive learning experiment
  - [TRAINING (+ TESTING)] Learning (estimating) unknown dependencies or rules from data samples (using "training data") and assessing the quality of these rules to obtain their "prediction accuracy" (using "testing data")
  - [PREDICTING] Using dependencies/rules learned in (1) to predict output(s) for future input values

➢ Resampling (K-fold Cross Validation)

- Split available data into 2 sets: Training + Testing
  (1) Use **training** set for model estimation (via data fitting)
  (2) Use **testing** data to estimate the prediction error of the model
- Always keep training data separate from testing data – why?
- Problems?
  - Might not have enough data
  - Happen across an unfortunate split – results are sensitive to data splitting
    - Nature of the data plays a part
    - Anomaly in the data
- Solution?
  - To mitigate the variance in the training data **k-fold cross validation** is employed
  - Estimate prediction error on left-out validation set $r_i = \frac{k}{n} \sum_{\mathbf{z}_i} \left(f_i(\mathbf{x}) - y \right)^2$
  - Estimate average prediction risk as $R_{cv} = \frac{1}{k} \sum_{i=1}^{k} r_i$
- Double Resampling method
  - For complexity control and for estimating prediction performance of a method
  - Estimation of prediction risk (test error) is critical for comparison of different learning methods



  - Stratified k-fold cross-validation (*balanced splits*) – Maintain the proportion of the classes in each fold

❖ PHILOSOPHICAL PERSPECTIVES [04]

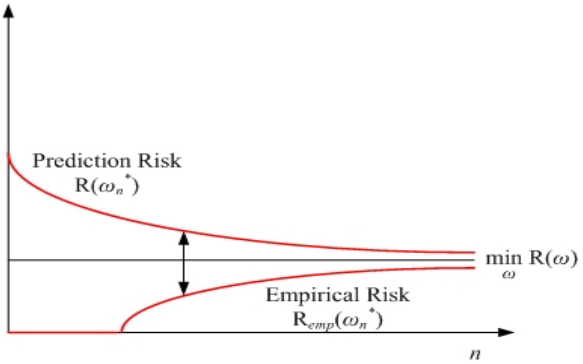
- Overview of Philosophy
  - Why philosophy is relevant?
    - Relationship between reality (facts) and mental constructs (ideas)
    - Epistemology and view of uncertainty
    - Inference: from facts to models
    - Truth vs utility
    - Role of human culture /social structure
  - Philosophy is concerned with the relationship between
    - Reality (Nature)
    - Sensory Perceptions
    - Mental Constructs
  - Three Philosophical Schools
    - Realism
      - Objective physical reality perceived via senses
      - Mental constructs reflect objective reality
    - Idealism
      - Primary role belongs to ideas (mental constructs)
      - Physical reality is a by-product of Mind
    - Instrumentalism
      - The goal of science is to produce useful theories
- Acquisition of Knowledge and Inference
  - Inference – the process of deriving a conclusion based on existing knowledge and/or facts (data)
  - Inference involves interaction between mental constructs (ideas) and facts
  - Predictive Learning needs empirical inference
  - Scientific discovery requires intelligent guessing (plausible reasoning) from observations ~ empirical inference
  - Empirical inference
  - Statistical inference
- Occam's Razor
- Popper's Falsifiability
- Bayesian Inference
  - See example – medical diagnosis

❖ STATISTICAL LEARNING THEORY AND LEARNING METHODS [05]

- Goal: introduce predictive learning as a scientific discipline
  - STL tied closely with VC-theory
  - Two factors responsible for generalization
    - Empirical risk
    - Complexity (capacity) of approximating functions
  - Connection to philosophy of science
    - VC-theory developed for binary classification (pattern recognition) ~ the simplest generalization problem
    - Natural sciences: from observations to scientific law
      → VC-theoretical results can be interpreted using general philosophical principles of induction, and vice versa
- Inductive learning problem setting
  - Inductive learning setting
  - Concepts and terminology
    - Approximating functions
    - Loss function
    - Expected risk
    - Goal: find an approximating function that minimizes the risk when join distribution is unknown
  - Empirical risk minimization (ERM)
    - Model parameterization: f(x, w)
    - Loss function: L(f(x, w),y)
    - Estimate risk from data: $$R_{emp}(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n} L(f(\mathbf{x}_i, \mathbf{w}), y_i)$$
    - Choose $\mathbf{w}^*$ that minimizes R_{emp}
  - Statistical Learning Theory developed from the theoretical analysis of ERM principle under finite sample settings
  - Curse of Dimensionality
    - In machine learning problems that involve learning a "state-of-nature" (maybe an *infinite distribution*) from a finite number of data samples in a high-dimensional feature space with each feature having a number of possible values, an **enormous** amount of training data is required to ensure that *there are several samples with each combination of values*
    - With a *fixed* number of training samples, *the predictive power reduces as the dimensionality increases*

- Keep it direct principle
  - The goal of learning is generalization rather than estimation of true function (system identification)
  - Keep-It-Direct Principle (Vapnik, 1995)
    - Do not solve an estimation problem of interest by solving a more general (harder) problem as an intermediate step
  - The goal of prediction (1) is different (less demanding) than the goal of estimating the true target function (2) everywhere in the input space.
  - The curse of dimensionality applies to system identification setting (2), but may not hold under predictive setting (1)
  - Philosophical interpretation of keep-it-direct
    - Interpretation of predictive models
      - Realism ~ objective truth (hidden in Nature)
      - Instrumentalism ~ creation of human mind (imposed on the data) – favored by KID
      - Objective Evaluation still possible (via prediction risk reflecting application needs)
        → Natural Science
- Empirical Risk Minimization (ERM)
  - VC-theory has 4 parts:
    - Analysis of consistency/convergence of ERM

$$R_{emp}(\omega) = \frac{1}{n}\sum_{i=1}^{n} L(y_i, f(\mathbf{x}_i, \omega)) \rightarrow \min$$

    - Generalization bounds
    - Inductive principles (for finite samples)
    - Constructive methods (learning algorithms) for implementing (3)
  - 

- VC-Dimension – Vapnik-Chervonekis Dimension
  - What does it mean to shatter points?
    - How many points can a linear boundary classify exactly? (1-D)
    - How many points can a linear boundary classify exactly? (2-D)
    - How many points can a linear boundary classify exactly? (d-D)
      - Can do **d+1** points
      - How many parameters in a linear classifier in d-dimensions?

$$w_0 + \sum_{i=1}^{d} w_i x_i$$

  - Shattering a set of points
    - Number of training points that can be classified exactly is VC dimension!
    - *Definition*: a **dichotomy** of a set *S* is a partition of S into two disjoint subsets
    - *Definition*: a set of instances *S* is **shattered** by hypothesis space *H* if and only if for every dichotomy of *S* there exists some hypothesis in *H* consistent with this dichotomy
      - If a set of n samples can be separated by a set of functions in all 2^n possible ways, the sample is said to be shattered (by the set of functions)
      - Shattering ~ a set of models can explain a given sample of size n (for all possible labelings)
    - *Definition*: The **Vapnik-Chervonenkis dimension**, *VC(H)*, of hypothesis space *H* defined over instance space *X* is the size of the largest finite subset of *X* **shattered** by *H*. If arbitrarily large finite sets of *X* can be shattered by *H*, then $VC(H) \equiv \infty$
  - Complexity of the classifier
    - Bias-Variance tradeoff in learning theory
    - **VC-dimension is *infinite*** *if a sample of size n can be split in all 2^n possible ways (in this case, no valid generalization is possible)*
    - Interpretation of the VC-dimension via **falsifiability**: functions with *small* VC-dim can be easily *falsified*
  - VC-dimension and falsifiability
    - A set of functions has VC-dimension h if
      - (a) It can explain (shatter) a set of *x* samples ~ there exists *x* samples that cannot falsify it; and
      - (b) It cannot shatter *x+1* samples ~ any *x+1* samples falsify this set
    - Finiteness of VC-dim is necessary and sufficient condition for **generalization**

❖ CLASSIFICATION AND REGRESSION [06-08]

➢ Decision Trees [06]

- The algorithm
  - Core algorithm for building decision trees: **ID3** (by J. R. Quinlan)
  - ID3 uses **Entropy** and **Information Gain** to construct a decision tree
- Entropy
  - One attribute:

$$\sum_{i=1}^{c} -p_i log_2 p_i$$

  - Two attributes:

$$E(T, X) = \sum_{c \in X} P(c)E(c)$$

- Information gain
  - Information Gain = Entropy(parent) – [average Entropy(children)]
  - $Gain(T, X) = Entropy(T) - Entropy(T, X)$
- Building a decision tree
- Decision tree to decision rules
- CART Trees

➢ Density Estimation (k-NN) [07]

- Definition
  - Instance based classifier – use observations directly (no models)
- Nearest neighbors and choosing 'k'
  - If 'k' is small:
    - ♦ Flexible
    - ♦ Varies a lot
  - If 'k' is large:
    - ♦ Smooth
    - ♦ Varies little
- K-NN algorithm
- Decision boundaries in global vs local models
  - Global
    - ♦ Stable, but can be inaccurate
  - Local
    - ♦ Accurate but can be unstable
- What ultimately matters? Generalization!

- Intro. To Bayesian Networks [08]
  - Definition
    - Generative classification approach – build a generative statistical model
  - Representation
    - Structured, graphical representations of probabilistic relationships between several random variables
    - Explicit representation of conditional independencies
    - Nodes and arcs
    - Directed graphical models
    - Conditional probability table (CPT)
  - Conditional independence
    - Global semantics defines the full joint distribution as the product of the local conditional distributions:

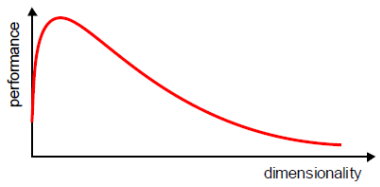
$$P(x_1, \ldots, x_n) = \prod_{i=1}^{n} P(x_i | parents(X_i))$$

(the product of probabilities)

  - Inference in Bayesian networks
  - Examples

❖ Feature Selection and Dimensionality Reduction [09]

- Principal Components Analysis (PCA)
  - Dimensionality reduction using principal components
  - High-dimensionality data sets
    - Datasets with a large number of features are called *high-dimensional datasets*
    - The curse of dimensionality
      - Refers to the problems associated with multivariate data analysis as the **dimensionality increases**
      - It relates to the fact that the *convergence of any estimator to the true value of a smooth function defined on a space of high dimension is very slow*
      - This means that, *a priori*, we need an "enormous" amount of observations to obtain a "*good*" estimate of a function



- PCA in a nutshell / Method
  - *Principal Component Analysis* (PCA) is a dimensionality reduction technique used to transform high-dimensional datasets into a dataset with fewer variables, where the set of resulting variables explains the maximum variance within the dataset
  - It extracts low dimensional set of features from a high dimensional data set with a motive to capture as much information as possible. With fewer variables, visualization also becomes much more meaningful.
  - Method: (high level)
    - Normalize variables [good idea; not always done] (Center / Scale)
    - Compute the covariance matrix of the transformed data
    - Compute the eigenvectors and eigenvalues (of the covariance matrix) in decreasing order
    - Choose the number of new dimensions and select the first 'd' eigenvectors
    - Project the transformed data points on the first 'd' eigenvectors
- Limitations of PCA
  - Does not consider class separability
  - No guarantee that the directions of maximum variance will contain good features for discrimination
- Relevant math background, including:
  - Covariance

$$cov(X,Y) = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{(n-1)}$$

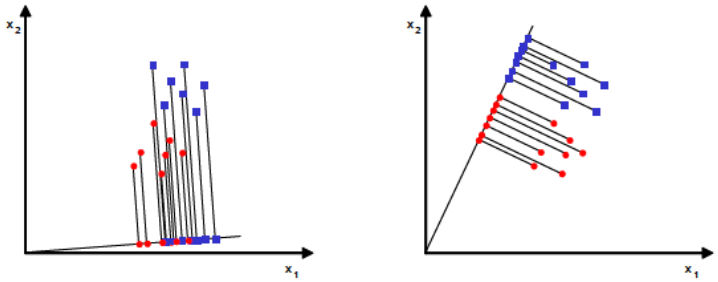
  - Properties of covariance matrix
  - Eigenvectors
  - Eigenvalues

➢ Linear Discriminant Analysis (LDA)

- Dimensionality reduction and Class discrimination
- Main limitations of PCA
  - Does not consider class separability
- Limitations of logistic regression
  - LDA addresses these limitations
- Linear Discriminant Analysis (LDA)
  - Linear Discriminant Analysis (LDA) is most commonly used as dimensionality reduction technique in the pre-processing step for pattern-classification and machine learning applications

- The goal is to project a dataset onto a lower-dimensional space with good class-separability in order avoid overfitting ("curse of dimensionality") and also reduce computational costs

- Dimensionality reduction does not only help reducing computational costs for a given classification task, but it can also be helpful to <u>avoid overfitting</u> by minimizing the error in parameter estimation ("curse of dimensionality")
- <u>LDA objective and method</u>
  - Simplifying assumptions about the data
    - Gaussian data
    - Each attribute has same variance
  - LDA seeks to reduce dimensionality while preserving as much of the class discriminatory information as possible
  - Of all the possible lines we would like to select the one that maximizes the <u>separability</u> of the scalars



  - Fisher suggested ("**Fisher's LDA**") maximizing the difference between the means, normalized by a measure of the within-class scatter
    - Calculating within-class scatter matrix, between-class scatter matrix
  - LDA method overview
- <u>Limitations of LDA</u>
- <u>PCA vs LDA</u>
  - The LDA approach is very similar to a Principal Components Analysis (PCA), but in addition to finding the component axes that maximize the variance of the data [this is PCA], it additionally is interested in the axes that maximizes the separation between multiple classes [LDA]
  - **PCA**: component axes that maximize the **variance**
  - **LDA**: maximizing the component axes for **class-separation**

❖ CLUSTERING AND ENSEMBLE METHODS [10]

➢ Motivation

- Many philosophical approaches (eastern philosophy / Bayesian averaging/etc) combine several theories (models) explaining the data
- Collective decision making prevalent in the real world – jury trial, multiple expert opinions (medicine, law, …)

➢ What is Ensemble Learning

- Standard inductive learning setting
- **Two combining strategies** (for improved generalization)
  - Apply different learning methods to the same data
    → Committee of Networks, Stacking, Bayesian averaging
  - Apply the same method to different (modified) realizations of training data
    → Bagging, Boosting
- Combining methods for classification:

$$F(\mathbf{x}) = sign\left(\sum_{k=1}^{N} w_k f_k(\mathbf{x})\right)$$

- Combining methods for regression
- Ensemble learning refers to a collection of methods that learn a target function by training a number of individual learners and **combining their predictions**
- Why ensemble learning?
  - **Accuracy:** a more reliable mapping can be obtained by combining the output of multiple "experts"
  - **Efficiency:** [divide-and conquer approach] a complex problem can be decomposed into multiple subproblems that are easier to understand and solve
  - There is not a single model that works for all problems!
  - The target function may not be implementable with **single classifiers**, but may be approximated by ensemble averaging
- When to use ensemble learning?
  - When you can build component classifiers that are more accurate than chance and, more importantly, that are independent from each other
- Why do ensemble methods work?
  - Because uncorrelated errors of individual classifiers can be eliminated through averaging

➢ Random Forests

- Tree based model
- Still suffers from bias and variance

- An optimal model should maintain a balance between these two types of errors. This is known as the **trade-off management** of bias-variance errors
- Ensemble learning is one way to execute this trade off analysis

▪ Bagging

- **Bagging** is a technique used to **reduce the variance** of our predictions by combining the result of multiple classifiers modeled on different sub-samples of the same data set

▪ Various implementations of bagging models, **Random Forest** is one of them

- Random Forest is a **versatile** machine learning method capable of performing both regression and classification tasks. It also undertakes dimensional reduction methods, handles outlier values and other essential steps of data exploration, and does a fairly good job
- It is a type of ensemble learning method, where a group of ***weak*** models combine to form a ***powerful*** model
- In Random Forest (RF), multiple trees are grown (hence "***forest***") as opposed to a single tree
- To classify a new object based on attributes, each tree gives a classification (the tree "votes" for that class)
- The forest chooses the **classification** having the most votes (over all the trees in the forest)
- In the case of **regression**, it takes the average of outputs by different trees
- RF involves sample of the input data with replacement
  - ♦ This is called bootstrap sampling
  - ♦ Out of bag error

▪ Advantages of Random Forest

▪ Disadvantages of Random Forest

➢ AdaBoost technique

▪ Bagging

- Subsampling the training set
- Bagging (for bootstrap aggregation) creates an ensemble by training individual classifiers on bootstrap samples of the training set
- As a result of the sampling-with-replacement procedure, each classifier is trained on the average of 63.2% of the training examples
- Bagging traditionally uses component classifiers of the same type (e.g., decision trees), and a simple combiner consisting of a majority vote across the ensemble

- Boosting
  - Boosting takes a different resampling approach than bagging, which maintains a constant probability of 1/N for selecting each example
  - In boosting, this probability is adapted over time based on performance
  - The component classifiers are built sequentially, and examples that are mislabeled by previous components are chosen more often than those that are correctly classified
  - Boosting is based on the concept of a "*weak learner*", an algorithm that performs slightly better than chance (e.g., 50% classification rate on binary tasks)
  - Schapire has shown that a weak learner can be converted into a strong learner by changing the distribution of training examples
  - A popular one is **AdaBoost** (Adaptive Boosting) – which allows the designer to continue adding components until an arbitrarily small error rate is obtained on the training set
- AdaBoost
  - AdaBoost (Adaptive Boosting) is a popular boosting technique which helps combine multiple "weak classifiers" into a single "strong classifier"
  - A weak classifier is simply a classifier that performs poorly, but performs better than random guessing
  - AdaBoost can be applied to any classification algorithm, so it's really a technique that builds on top of other classifiers as opposed to being a classifier itself
  - Benefits of AdaBoost:
    - It helps you choose the training set for each new classifier that you train based on the results of the previous classifier
    - It determines how much weight should be given to each classifier's proposed answer when combining the results
- AdaBoost steps and procedure
- AdaBoost advantages

➢ Self Organizing Maps (SOM) – Clustering (Unsupervised)

- Clustering
  - separating a data set into several groups (clusters) according to some measure of similarity
  - Partitioning a set of n objects (samples) into k disjoint groups, based on some similarity measure. ***Assumptions:***
    - Similarity: distance metric dist(i, j)
    - Usually k given a priori (but not always!)

- Intuitive Motivation:
  - Similar objects ~ into one cluster
  - Dissimilar objects ~ into different clusters
  - Distance needs to be defined for different types of input

- Goals of clustering:
  - interpretation (of resulting clusters)
  - exploratory data analysis
  - preprocessing for supervised learning
  - often the goal is not formally stated
- Self Organizing Map (SOM) – Kohonen Network
  - Biological motivation
  - A kind of unsupervised training
    - In which networks learn to form their own classifications of the training data without extra help
  - A SOM learns to classify the training data without any external supervision – thus requiring no target vector
  - Competitive aspect of SOM
    - Output neurons compete amongst themselves to be activated: only one is activated at any given time
    - Activated neuron: "winning neuron"
    - Such competition can be induced/implemented by having ***lateral inhibition connections*** (negative feedback paths) between the neurons
    - The result is that neurons are forced to organize themselves (hence the name)
  - SOMs are neural networks that employ unsupervised learning methods, mapping their weights to conform to the given input data with a goal of representing multidimensional data in an easier and understandable form for the human eye. (pragmatic value of representing complex data)
  - Goal of SOM: dimensionality reduction!
    - Project given (high-dim) data onto low-dim space (called a map)
    - Feature space (Z-space) is 1D or 2D
  - Goal of SOM: Transform incoming signal pattern of arbitrary dimension into a 1 or 2 dim discrete map
    - And to perform this transformation adaptively in a topologically ordered fashion
  - Topographic Maps; and overall understanding of the **architecture** and the **algorithm**
  - SOM applications

❖ SUPPORT VECTOR MACHINES [11]

➢ Definition

- Discriminative form of classification – directly estimate a decision rule/boundary

➢ Support Vector Machines (SVM)

- Motivation: Philosophical
  - **Classical view**: good model – explains the data + low complexity
    → Occam's razor (complexity ~ # parameters)
  - **VC theory**: good model – explains the data + low VC-dimension
    → VC-falsifiability (small VC-dim ~ large falsifiability), i.e. the goal is to find a model that: can explain training data / cannot explain other data
- Linear classifier – which is the best?
- Classifier margin – the width that the boundary could be increased by before hitting a data point
- Maximum margin – the maximum margin linear classifier is the linear classifier with the … maximum margin! This is the simplest kind of SVM (Called an **LSVM** ~ *Linear SVM*)
- SVMs maximize the margin around the separating hyperplane – a.k.a. large margin classifiers
- The decision function is fully specified by a subset of training samples, the **support vectors**
- The classifier is a **separating hyperplane**
- Most "important" training points are support vectors; they define the hyperplane
- Linearly non-separable cases?
  - If we use a hard margin:
    - Hard Margin: So far we require all data points be classified correctly – No training error
    - Overfitting!
  - Soft margin classification
    - Instead of minimizing the number of misclassified points we can minimize the distance between these points and their correct plane
- Types of Kernel functions
- Properties of SVM
  - Flexible
  - Sparseness of solution
  - Handle large feature spaces
  - Overfitting controlled by soft margin approach
- Why do SVMs work?
- Overfitting and Occam's Razor
- Weakness of SVM

❖ Connectionism and Neural Networks (incl. Perceptron) [12]

- ➢ Biological background
  - Neuron
  - Synapse
  - Model of the human brain
- ➢ Model of Neuron
  - Main processing unit
- ➢ Perceptron Learning "Delta" Rule
- ➢ Review perceptron example with sample values

❖ Metrics and Methods for Performance Evaluation [13]

- ➢ Metrics for Performance Evaluation
  - How to evaluate the performance of a model?
    - Focus on predictive capability of a model
    - Confusion Matrix
    - TP / TN / FP / FN
    - Accuracy
    - Error rate
    - Cost-sensitive measures
      - ♦ Precision
      - ♦ Recall (Sensitivity)
      - ♦ F-measure (F1 score)
    - Specificity
    - Matthews Correlation Coefficient (MCC)
- ➢ Methods for Performance Evaluation
  - How to obtain reliable estimates?
    - Holdout
    - Random subsampling
    - Stratified sampling
    - Bootstrap
    - Cross validation
      - ♦ K-fold
      - ♦ Leave-one-out
- ➢ Methods for Model Comparison
  - How to compare the relative performance among competing models?
    - ROC (Receiver Operating Characteristic)
      - ♦ Plots TP (on the y-axis) against FP (on the x-axis)

❖ INTRO. TO DEEP LEARNING ~ CONVOLUTIONAL NEURAL NETWORK (CNN) [14]

➢ Deep learning tasks prevalent

- Machine Learning systems are used to identify objects in images, transcribe speech into text, match news items, posts or products with users' interests, and select relevant results of search. Increasingly, these applications make use of a class of techniques called **deep learning.**

➢ Convolutional Neural Networks (CNN)

- Convolutional neural networks (CNN) are a type of artificial neural network (ANN) that includes both fully-connected layers and locally-connected layers known as convolutional layers
- In large ("*deep*") convolutional networks, it is common to see other types of layers such as pooling layers, activation layers, and batch normalization layers
- CNN architecture

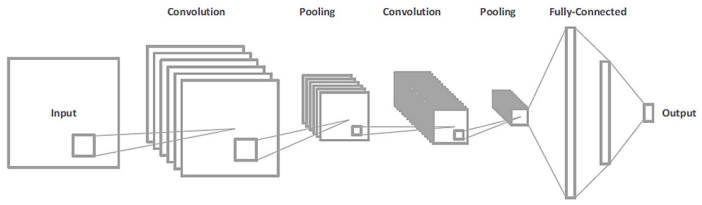


Figure 1: Convolutional neural network composed of convolution, pooling, and fully-connected layers

- The problem: semantic gap ~ why visual recognition is a hard task (challenge) for machines
- Challenges in visual recognition
- Convolutional layer
  - **Convolve** the filter with the image – i.e. *slide over the image spatially, computing dot products*
- Pooling Layer
  - Max Pooling
  - Hierarchies and Pooling
- Examples of CNN today and its uses
- Recurrent Neural Network (RNN) + Convolutional Neural Network (CNN)