PR & ML: CS5691

The goal of learning is prediction. Learning falls into many categories, including:

- **Supervised learning,**
- **Unsupervised learning,**
- **Semi-supervised, self-supervised, weakly-supervised learning**
- **Transfer Learning**
- **Reinforcement learning**
- **Incremental/Online Learning; Data Drift**
- **Few Shot Learning, Co-Training etc.**
- **Deep Learning.**

Supervised learning is best understood and studied.

Machine Learning is ...

an algorithm that can learn from data without relying on rules-based programming.

learning problems. Some major classes of learning problems are:

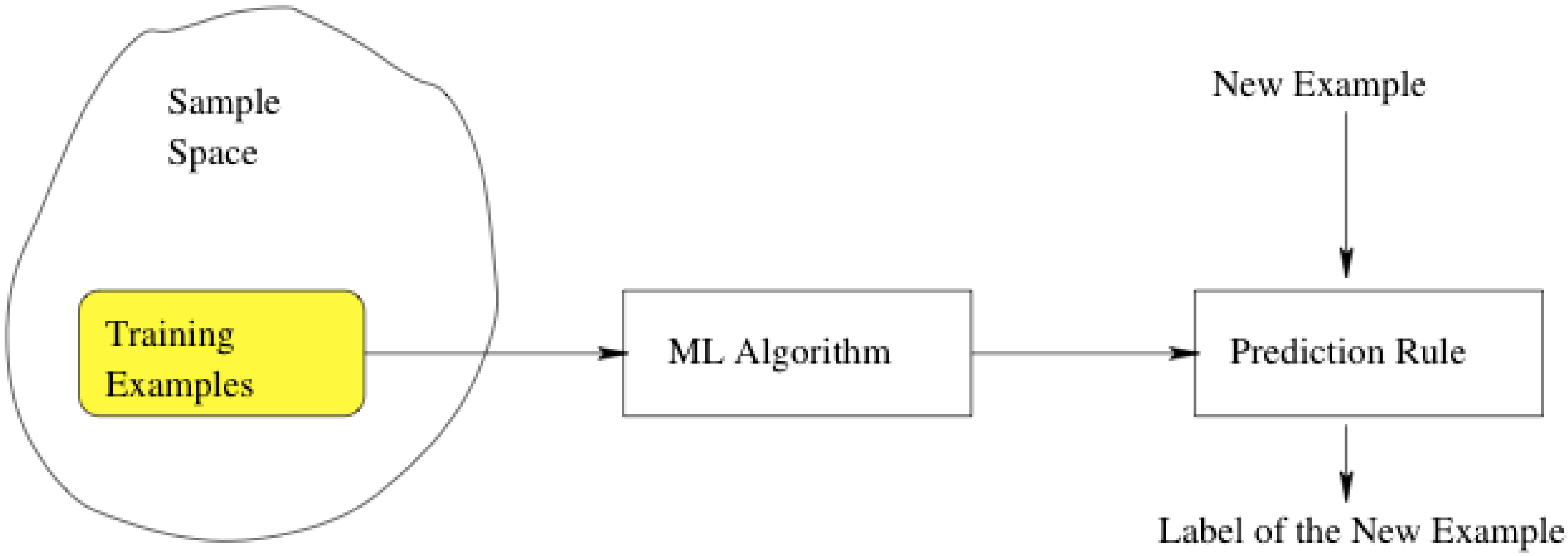
- *Classification*: Assign a category to each item. For example, document classification may assign items with categories such as *politics, business, sports*, or *weather* while image classification may assign items with categories such as *landscape, portrait*, or *animal*. The number of categories in such tasks is often relatively small, but can be large in some difficult tasks and even unbounded as in OCR, text classification, or speech recognition.
- *Regression*: Predict a real value for each item. Examples of regression include prediction of stock values or variations of economic variables. In this problem, the penalty for an incorrect prediction depends on the magnitude of the difference between the true and predicted values, in contrast with the classification problem, where there is typically no notion of closeness between various categories.
- *Ranking*: Order items according to some criterion. Web search, e.g., returning web pages relevant to a search query, is the canonical ranking example. Many other similar ranking problems arise in the context of the design of information extraction or natural language processing systems.
- *Clustering*: Partition items into homogeneous regions. Clustering is often performed to analyze very large data sets. For example, in the context of social network analysis, clustering algorithms attempt to identify "communities" within large groups of people.
- *Dimensionality reduction* or *manifold learning*: Transform an initial representation of items into a lower-dimensional representation of these items while preserving some properties of the initial representation. A common example involves preprocessing digital images in computer vision tasks.

In supervised learning, an algorithm is given samples that are labeled in some useful way. For example, the samples might be descriptions of apples, and the labels could be whether or not the apples are edible.

Supervised learning involves learning from a training set of data. Every point in the training is an input-output pair, where the input maps to an output. The learning problem consists of inferring the function that maps between the input and the output in a predictive fashion, such that the learned function can be used to predict output from future input.

The algorithm takes these previously *labeled samples* and uses them to induce a classifier. This *classifier is a function* that assigns labels to samples including the samples that have never been previously seen by the algorithm.

The goal of the supervised learning algorithm is to *optimize some measure of performance* such as minimizing the number of mistakes made on new samples.



Machine Learning is ...
a subfield of computer science and artificial intelligence which deals with building systems that can learn from data, instead of explicitly programmed instructions.

CS5691: Pattern Recognition and Machine Learning

Jan. - May 2023

Course Contents | References | Lecture Slides | Schedule

Handout

- Basics of Linear Algebra, Probability Theory and Optimization
  Vectors, Inner product, Outer product, Inverse of a matrix, Eigenanalysis, Singular value decomposition, Probability distributions - Discrete distributions and Continuous distributions; Independence of events, Conditional probability distribution and Joint probability distribution, Bayes theorem, Unconstrained optimization, Constrained optimization - Lagrangian multiplier method.

- Methods for Function Approximation:
  Linear models for regression, Parameter estimation methods - Maximum likelihood method and Maximum a posteriori method; Regularization, Ridge regression, Lasso, Bias-Variance decomposition, Bayesian linear regression.

- Probabilistic Models for Classification
  Bayesian decision theory, Bayes classifier, Minimum error-rate classification, Normal (Gaussian) density - Discriminant functions, Decision surfaces, Maximum-Likelihood estimation, Maximum a posteriori estimation; Gaussian mixture models -- Expectation-Maximization method for parameter estimation; Naive Bayes classifier, Non-parametric techniques for density estimation -- Parzen-window method, K-nearest neighbors method, Hidden Markov models (HMMs) for sequential pattern classification -- Discrete HMMs and Continuous density HMMs.

- Discriminative Learning based Models for Classification
  Logistic regression, Perceptron, Multilayer feedforward neural network - Gradient descent method, Error backpropagation method; Support vector machine.

- Dimensionality Reduction Techniques
  Principal component analysis, Fisher discriminant analysis, Multiple discriminant analysis.

- Non-Metric Methods for Classification
  Decision trees, CART.

- Ensemble Methods for Classification
  Bagging, Boosting, Gradient boosting.

- Pattern Clustering
  Criterion functions for clustering, Techniques for clustering -- K-means clustering, Hierarchical clustering, Density based clustering and Spectral clustering; Cluster validation.

Textbooks

Elements of Statistical Learning. Hastie, Tibshirani, and Friedman. Springer, 2009+.

Pattern Recognition and Machine Learning. Christopher Bishop, Springer, 2006+.

Kevin R Murphy, "Machine Learning - A Probabilistic Perspective", The MIT Press, 2012.

http://www.cse.iitm.ac.in/~vplab/prml.html

EXAM PATTERN
(tentative range only, to be finalized before ES):

END SEM -	**40-50;**
Mid-Sem –	**15-20;**
Tuts -	**10-15**
Software Assignments (2-3) –	**30-40**

MID-SEM - (Duration: 60 mins)

End Semester - (Duration: 150-180 mins)

Visit:
http://www.cse.iitm.ac.in/~vplab/prml.html

Types of Data

- Two basically different types of data
  - Quantitative (numerical): e.g. stock price
  - Categorical (discrete, often binary): cancer/no cancer
- Data are predicted
  - on the basis of a set of features (e.g. diet or clinical measurements)
  - from a set of (observed) training data on these features
  - For a set of objects (e.g. people).
  - Inputs for the problems are also called predictors or independent variables
  - Outputs are also called responses or dependent variables
- The prediction model is called a learner or estimator (Schätzer).
  - Supervised learning: learn on outcomes for observed features
  - Unsupervised learning: no feature values available

Computer Scientists' Contribution to Statistics: Kernel Methods

Vladimir Vapnik

Jerome H. Friedman

CENTER FOR BIOINFORMATICS

INFORMATIK

- *Examples*: Items or instances of data used for learning or evaluation. In our spam problem, these examples correspond to the collection of email messages we will use for learning and testing.
- *Features*: The set of attributes, often represented as a vector, associated to an example. In the case of email messages, some relevant features may include the length of the message, the name of the sender, various characteristics of the header, the presence of certain keywords in the body of the message, and so on.
- *Labels*: Values or categories assigned to examples. In classification problems, examples are assigned specific categories, for instance, the SPAM and non-SPAM categories in our binary classification problem. In regression, items are assigned real-valued labels.
- *Training sample*: Examples used to train a learning algorithm. In our spam problem, the training sample consists of a set of email examples along with their associated labels. The training sample varies for different learning scenarios, as described in section 1.4.

- *Validation sample*: Examples used to tune the parameters of a learning algorithm when working with labeled data. Learning algorithms typically have one or more free parameters, and the validation sample is used to select appropriate values for these model parameters.
- *Test sample*: Examples used to evaluate the performance of a learning algorithm. The test sample is separate from the training and validation data and is not made available in the learning stage. In the spam problem, the test sample consists of a collection of email examples for which the learning algorithm must predict labels based on features. These predictions are then compared with the labels of the test sample to measure the performance of the algorithm.
- *Loss function*: A function that measures the difference, or loss, between a predicted label and a true label. Denoting the set of all labels as $\mathcal{Y}$ and the set of possible predictions as $\mathcal{Y}'$, a loss function L is a mapping $L\colon \mathcal{Y} \times \mathcal{Y}' \to \mathbb{R}_+$. In most cases, $\mathcal{Y}' = \mathcal{Y}$ and the loss function is bounded, but these conditions do not always hold. Common examples of loss functions include the zero-one (or misclassification) loss defined over $\{-1, +1\} \times \{-1, +1\}$ by $L(y, y') = 1_{y' \neq y}$ and the squared loss defined over $I \times I$ by $L(y, y') = (y' - y)^2$, where $I \subseteq \mathbb{R}$ is typically a bounded interval.
- *Hypothesis set*: A set of functions mapping features (feature vectors) to the set of labels $\mathcal{Y}$. In our example, these may be a set of functions mapping email features to $\mathcal{Y} = \{\text{SPAM}, \text{non-SPAM}\}$. More generally, hypotheses may be functions mapping features to a different set $\mathcal{Y}'$. They could be linear functions mapping email feature vectors to real numbers interpreted as *scores* ($\mathcal{Y}' = \mathbb{R}$), with higher score values more indicative of SPAM than lower ones.

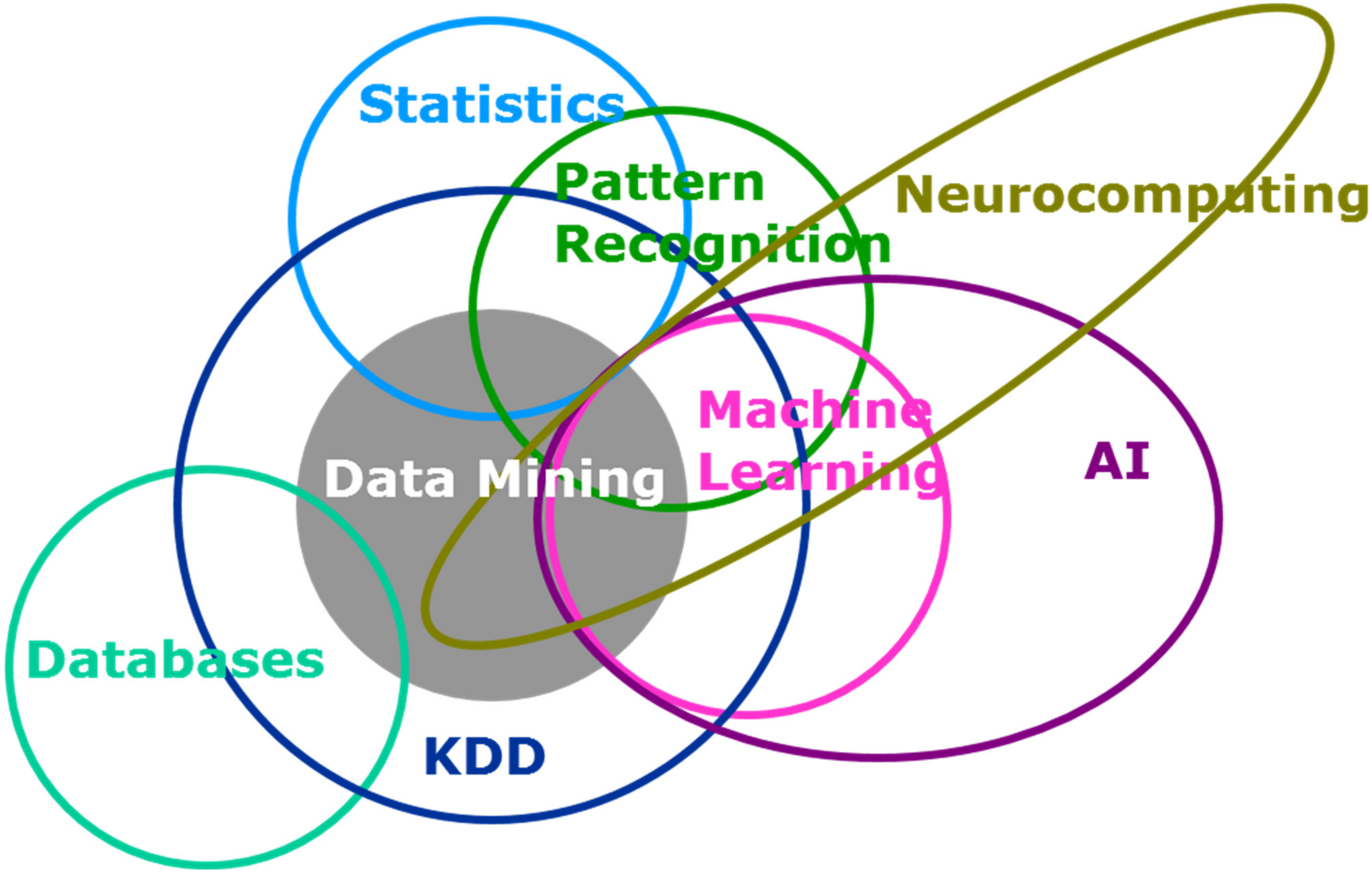
Computational learning theory studies the time complexity and feasibility of learning. In computational learning theory, a computation is considered feasible if it can be done in polynomial time.

Classification problems are those for which the output will be an element from a discrete set of labels. Classification is very common for machine learning applications. The input would be represented by a large multidimensional vector whose elements represent pixels in the picture, say CV applications.

After learning a function based on the training set data, that function is validated on a test set of data, data that did not appear in the training set.

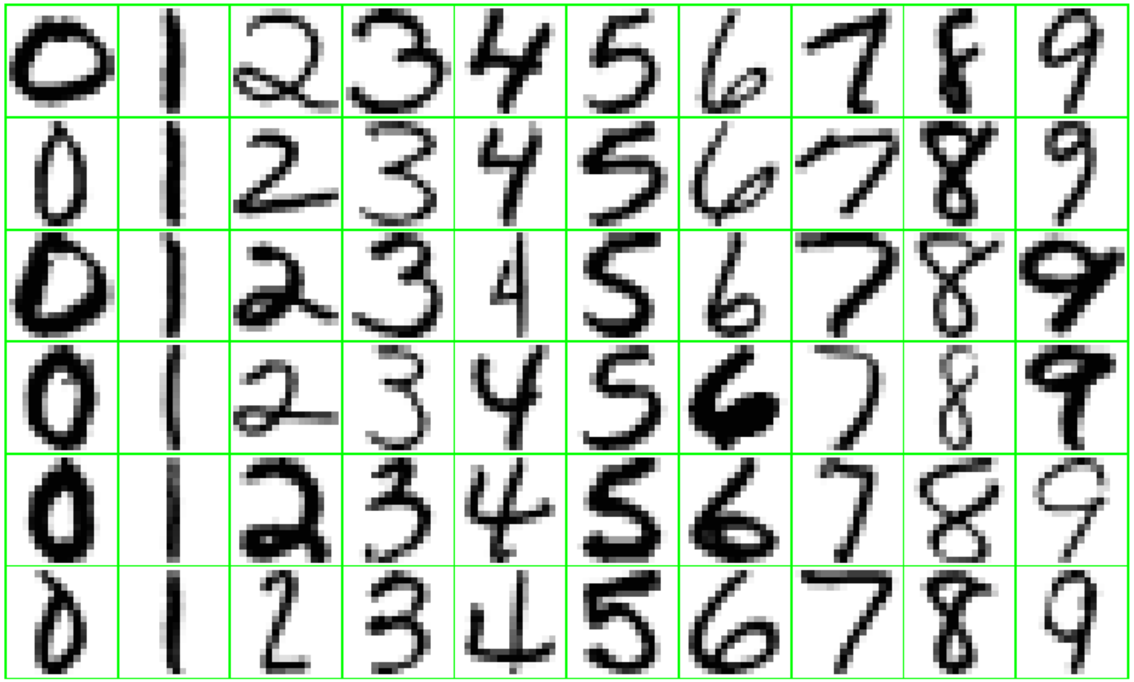
Computational learning theory
(Wikipedia)

- **Probably approximately correct learning** (PAC learning) --Leslie Valiant
  - inspired boosting
- **VC theory** --Vladimir Vapnik
  - led to SVMs
- **Bayesian inference** --Thomas Bayes
- **Algorithmic learning theory** --E. M. Gold
- **Online machine learning** --Nick Littlestone
- SRM (**Structural risk minimization)**
  - model estimation


Statistics
Pattern
Recognition
Neurocomputing
Machine
Learning
AI
Data Mining
Databases
KDD

Example: Recognition of Handwritten Digits

- Data: images are single digits 16x16 8-bit gray-scale, normalized for size and orientation
- Classify: newly written digits
- Non-binary classification problem
- Low tolerance to misclassifications



Applications:

- **Document Classificati**
- **Object Recognition +**

npainting

- **Action Classification i**
- **Exit polls, Stock Mark**

rt, product recommendations

rdware design

ion

Categories of **Supervised Learning**:

- Linear Regression – Prediction using Least Squares
- Function Approximation – Linear basis expansion, cross entropy
- Bayes
- Regularization
- Kernel methods & SVM;
- Basis and Dictionary methods;
- Model selection
- Perceptron, ANN
- Bagging, Boosting, Additive Trees
- Logistic Regression, LDA
- Inductive Learning
- Decision Trees
- Deep Learning

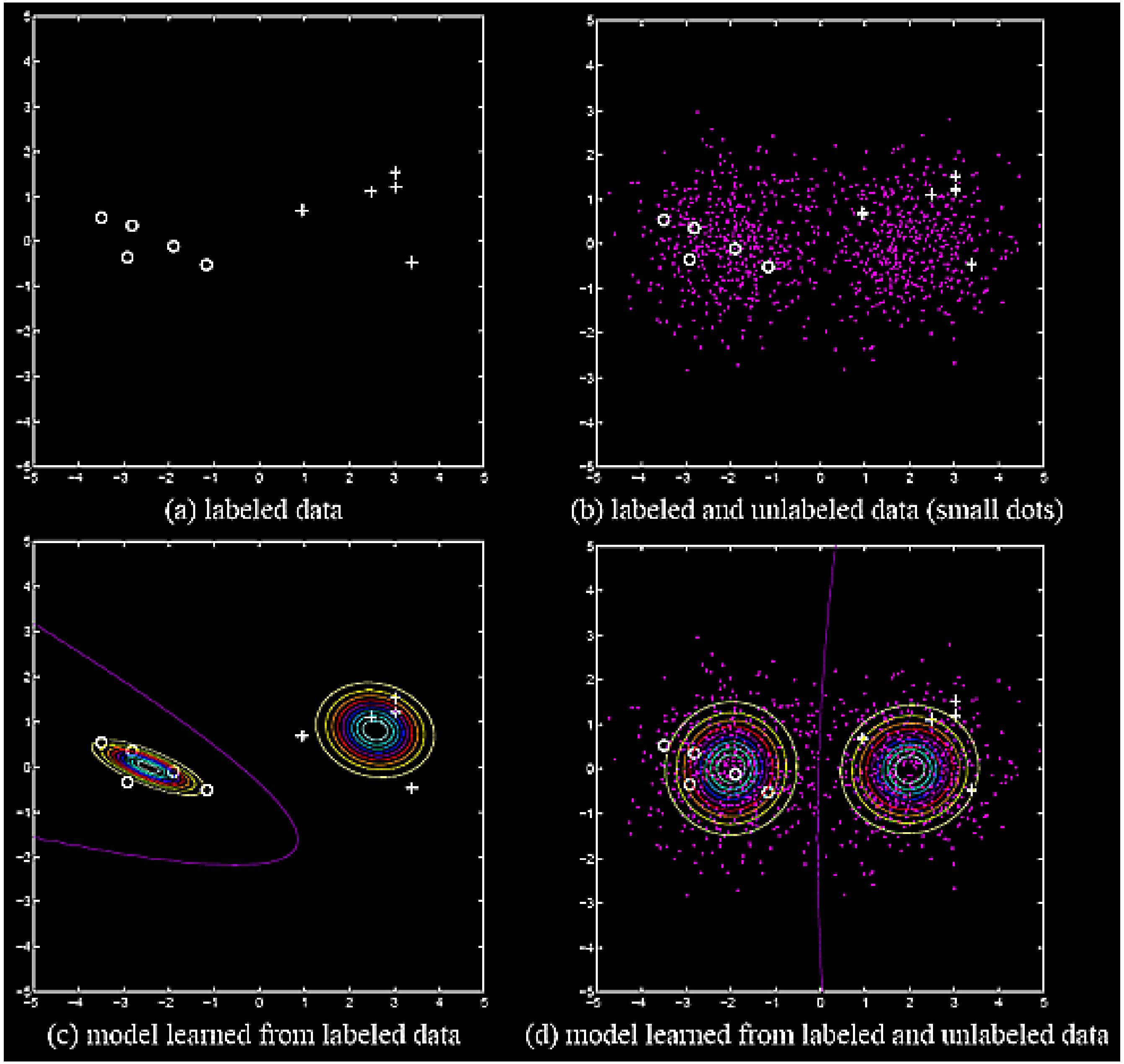
Unsupervised Learning

- No training data in the form of (input, output) pair is available
- Applications:
  - Dimensionality reduction
  - Data compression
  - Outlier detection
  - Classification
  - Segmentation/clustering
  - Probability density estimation
  - …

Semi-supervised Learning

- Uses both labeled data (in the form (input, output) pairs) and unlabelled data for learning
- When labeling of data is a costly affair semi-supervised techniques could be very useful
- Examples: Generative models, self-training, co-training

Example: Semi-supervised Learning



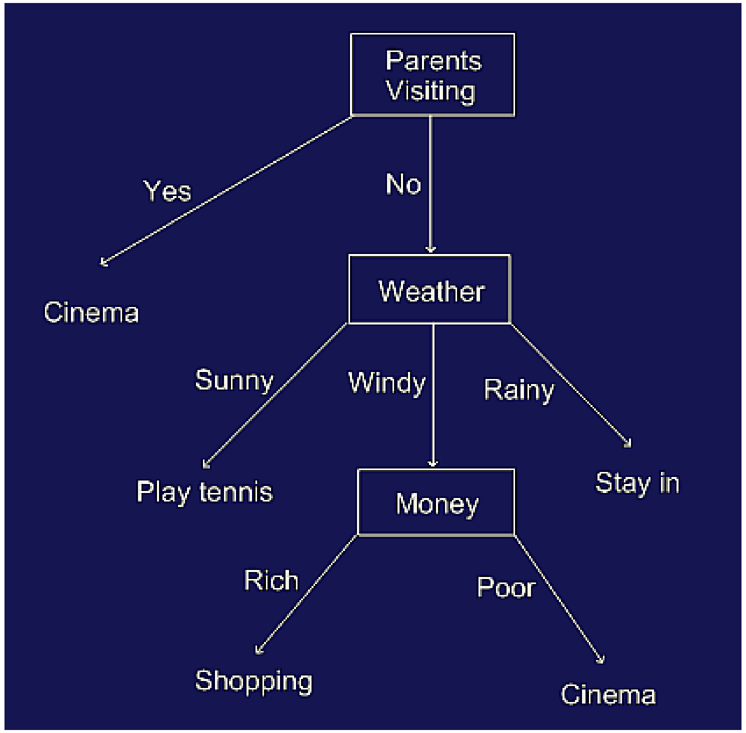
Source: Semi-supervised literature survey by X. Zhu, Technical Report

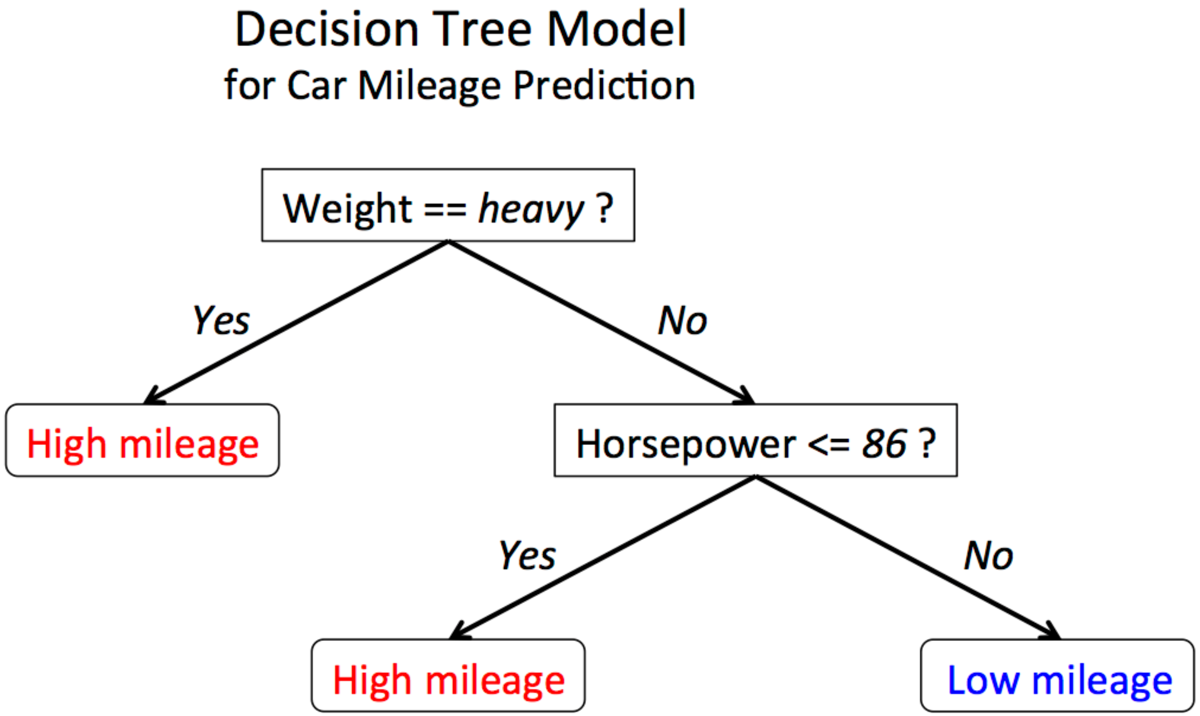
Reinforcement Learning

- Reinforcement learning is the problem faced by an agent that must learn behavior through trial-and-error interactions with a dynamic environment.
- There is no teacher telling the agent wrong or right
- There is critic that gives a reward / penalty for the agent’s action
- Applications:
  - Robotics
  - Combinatorial search problems, such as games
  - Industrial manufacturing
  - Many others!

Decision trees

- One possible representation for hypotheses
- E.g., here is the "true" tree for deciding whether to wait:





https://www.crondose.com/2016/07/easy-way-understand-decision-trees/

http://www.doc.ic.ac.uk/~sgc/teaching/pre2012/v231/lecture11.html

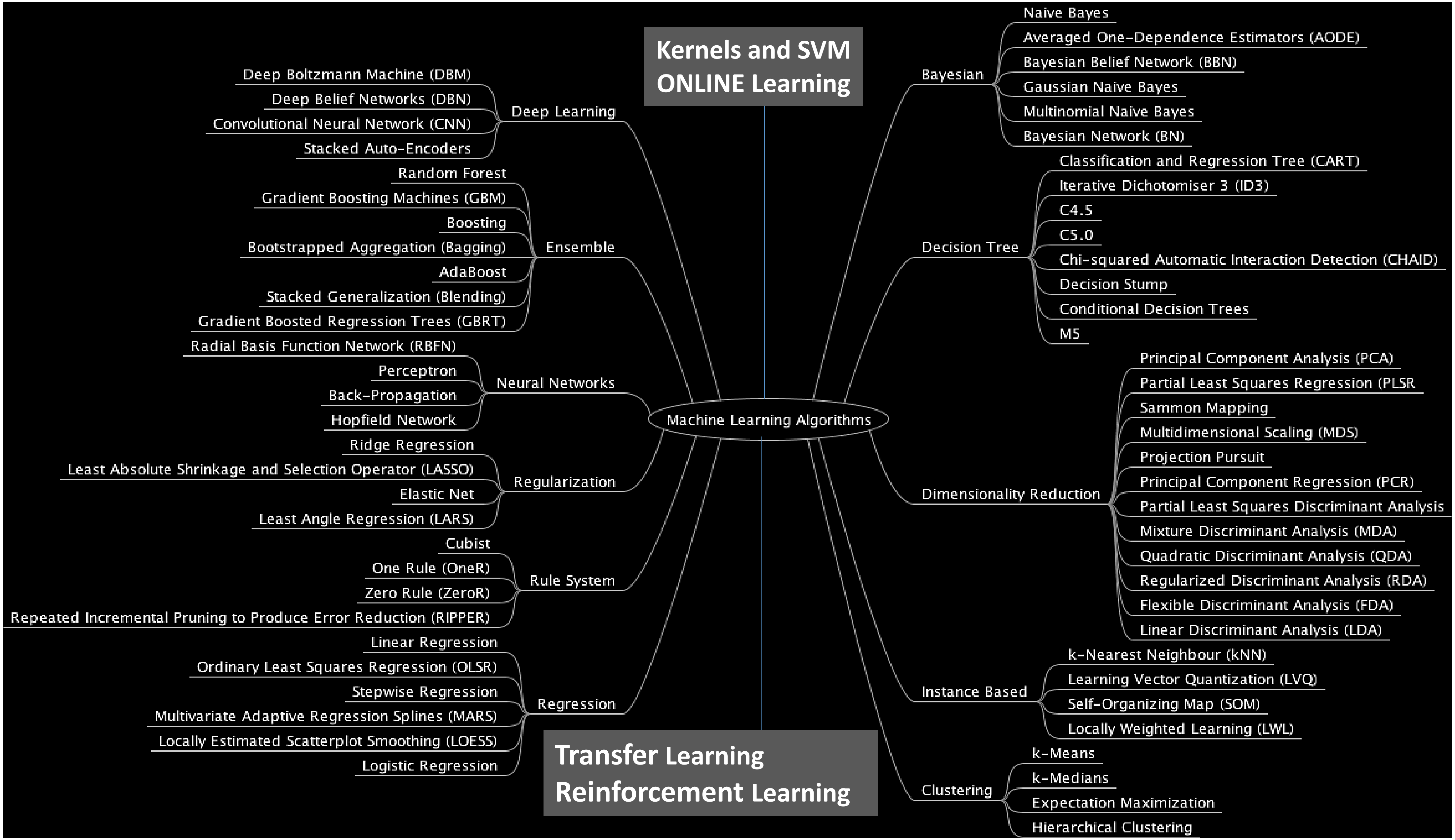
ONLINE LEARNING (src: Wiki)

In Online machine learning data becomes available in a sequential order and is used to update our best predictor for future data at each step, as opposed to batch learning techniques which generate the best predictor by learning on the entire training data set at once.

In this case, it is necessary for the algorithm to dynamically adapt to new patterns in the data, or when the data itself is generated as a function of time, e.g. stock price prediction. Online learning algorithms may be prone to catastrophic interference. This problem is tackled by incremental learning approaches.

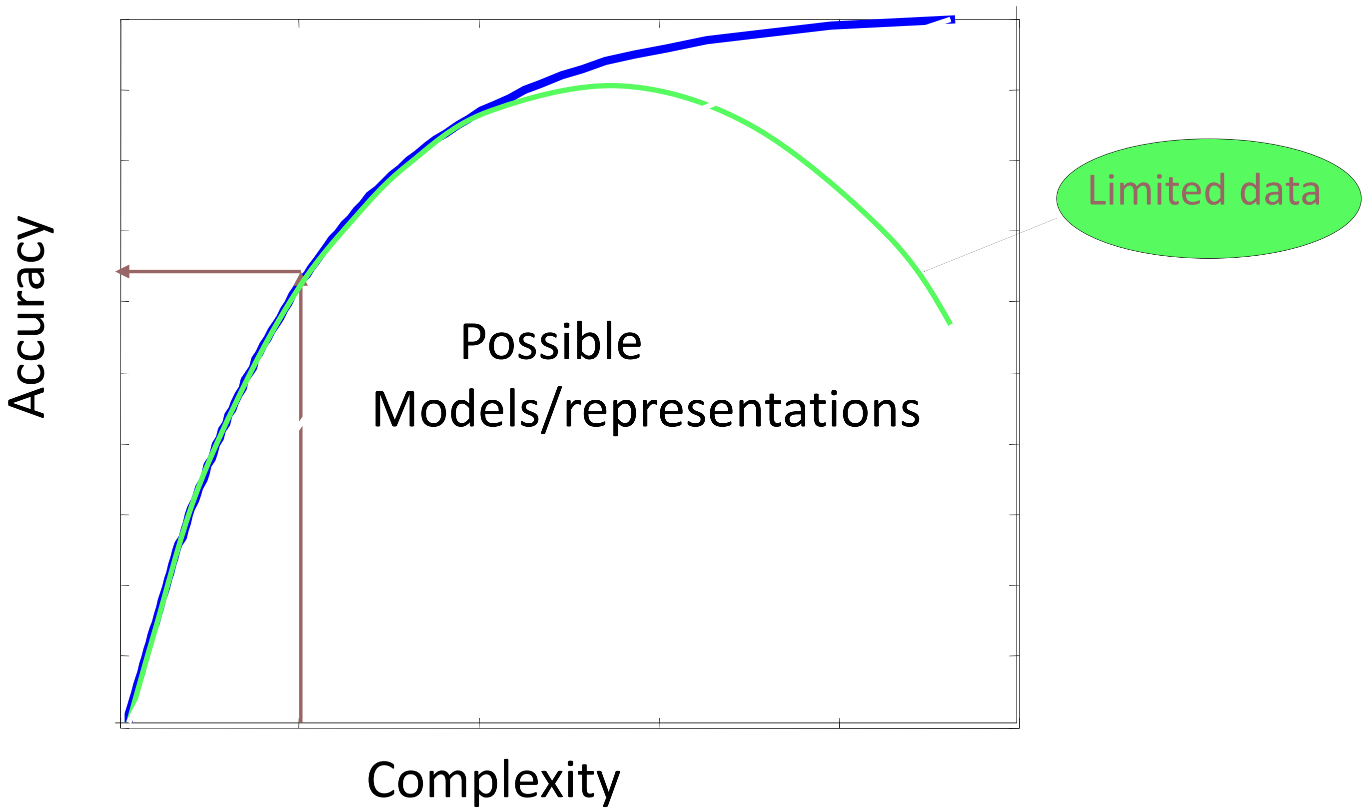
A purely online model would learn based on just the new input , the current best predictor and some extra stored information (which is usually expected to have storage requirements independent of training data size).

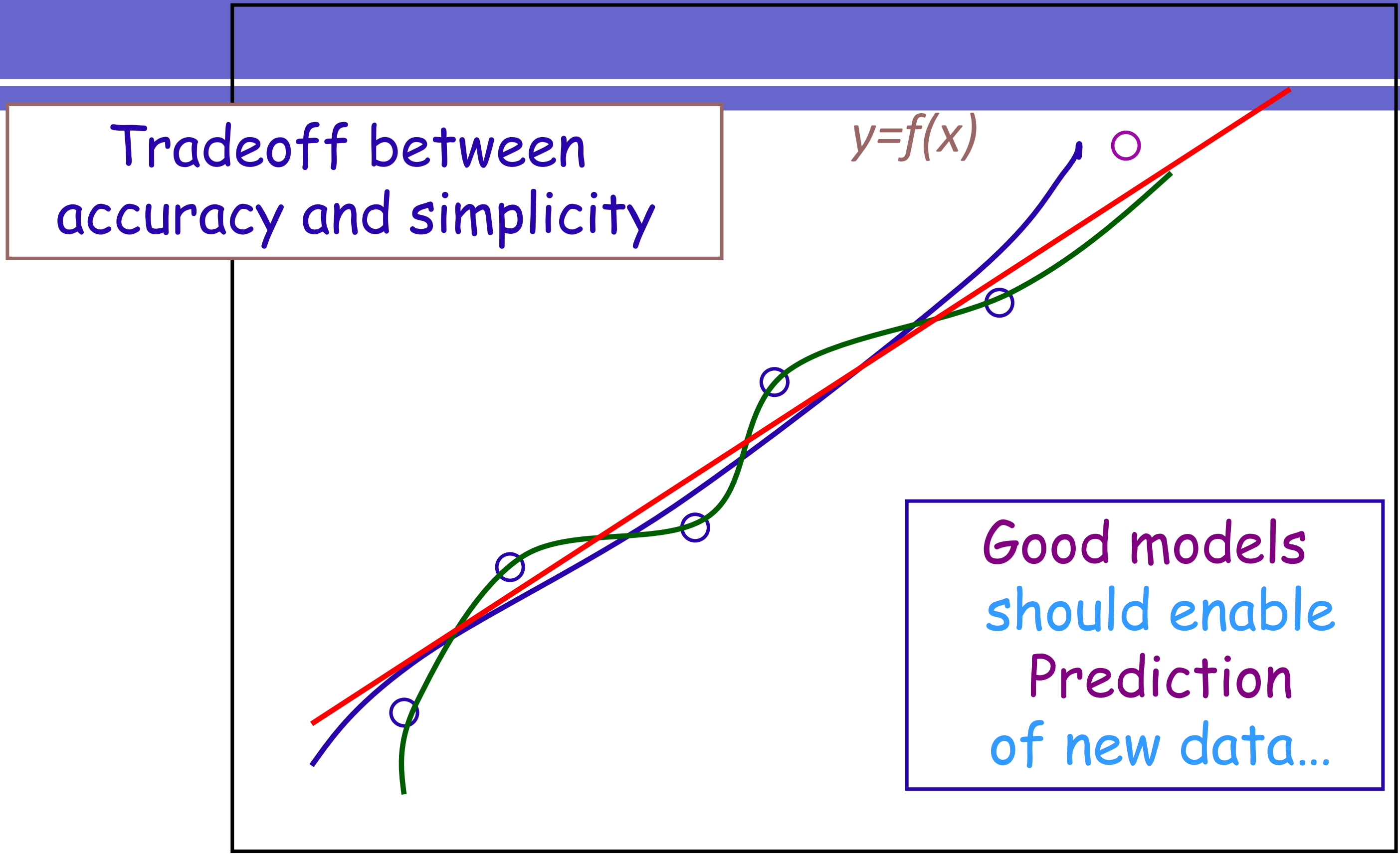
A common strategy to overcome the issue of storage, is to learn using mini-batches, which process a small batch of data points at a time, this can be considered as pseudo-online learning for much smaller than the total number of training points.



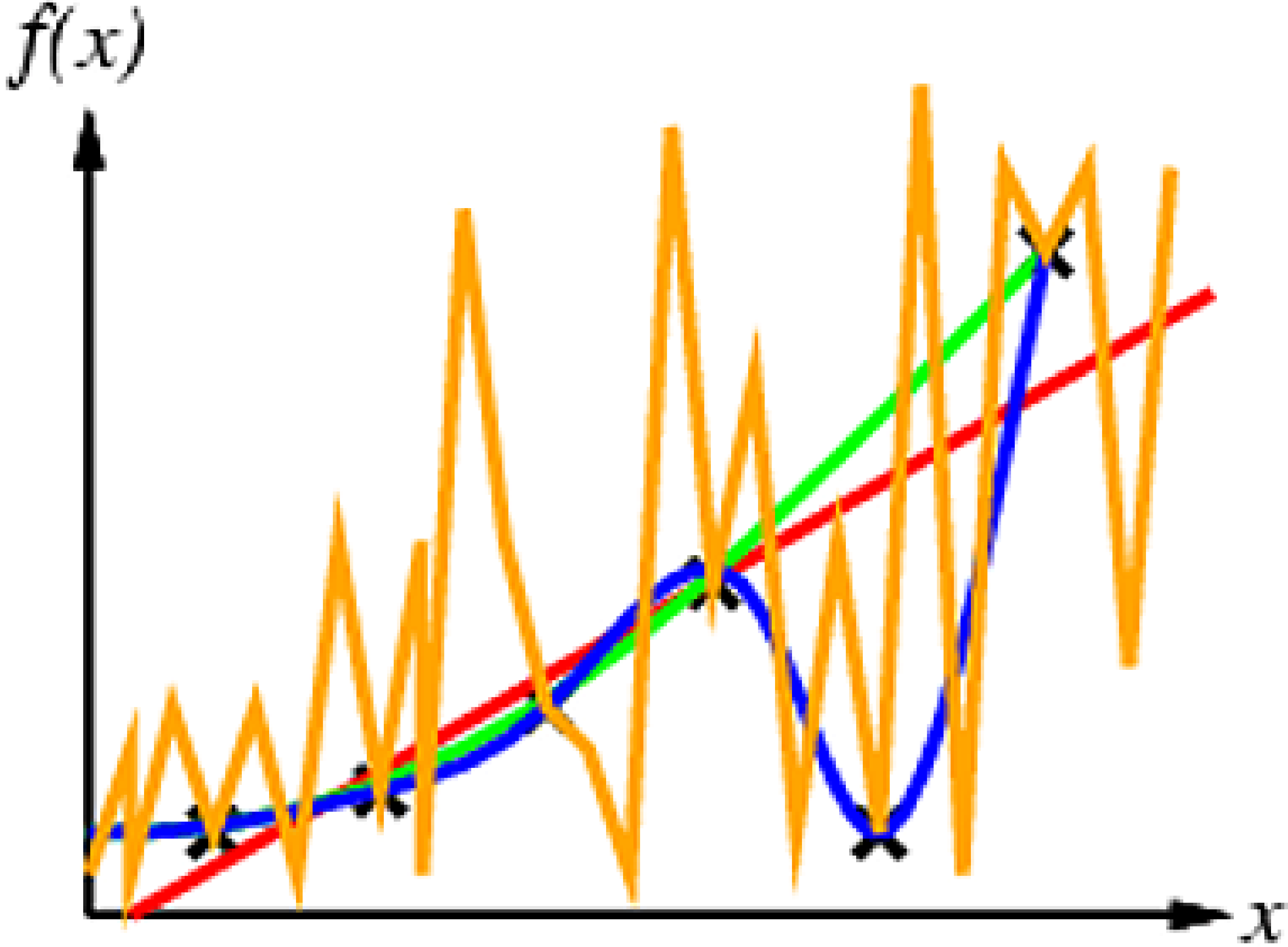
Kernels and SVM
ONLINE Learning
Machine Learning Algorithms
Deep Learning
Deep Boltzmann Machine (DBM)
Deep Belief Networks (DBN)
Convolutional Neural Network (CNN)
Stacked Auto-Encoders
Ensemble
Random Forest
Gradient Boosting Machines (GBM)
Boosting
Bootstrapped Aggregation (Bagging)
AdaBoost
Stacked Generalization (Blending)
Gradient Boosted Regression Trees (GBRT)
Neural Networks
Radial Basis Function Network (RBFN)
Perceptron
Back-Propagation
Hopfield Network
Regularization
Ridge Regression
Least Absolute Shrinkage and Selection Operator (LASSO)
Elastic Net
Least Angle Regression (LARS)
Rule System
Cubist
One Rule (OneR)
Zero Rule (ZeroR)
Repeated Incremental Pruning to Produce Error Reduction (RIPPER)
Regression
Linear Regression
Ordinary Least Squares Regression (OLSR)
Stepwise Regression
Multivariate Adaptive Regression Splines (MARS)
Locally Estimated Scatterplot Smoothing (LOESS)
Logistic Regression
Bayesian
Naive Bayes
Averaged One-Dependence Estimators (AODE)
Bayesian Belief Network (BBN)
Gaussian Naive Bayes
Multinomial Naive Bayes
Bayesian Network (BN)
Decision Tree
Classification and Regression Tree (CART)
Iterative Dichotomiser 3 (ID3)
C4.5
C5.0
Chi-squared Automatic Interaction Detection (CHAID)
Decision Stump
Conditional Decision Trees
M5
Dimensionality Reduction
Principal Component Analysis (PCA)
Partial Least Squares Regression (PLSR
Sammon Mapping
Multidimensional Scaling (MDS)
Projection Pursuit
Principal Component Regression (PCR)
Partial Least Squares Discriminant Analysis
Mixture Discriminant Analysis (MDA)
Quadratic Discriminant Analysis (QDA)
Regularized Discriminant Analysis (RDA)
Flexible Discriminant Analysis (FDA)
Linear Discriminant Analysis (LDA)
Instance Based
k-Nearest Neighbour (kNN)
Learning Vector Quantization (LVQ)
Self-Organizing Map (SOM)
Locally Weighted Learning (LWL)
Clustering
k-Means
k-Medians
Expectation Maximization
Hierarchical Clustering
Transfer Learning
Reinforcement Learning

A Fundamental Dilemma of Science: Model Complexity vs Prediction Accuracy

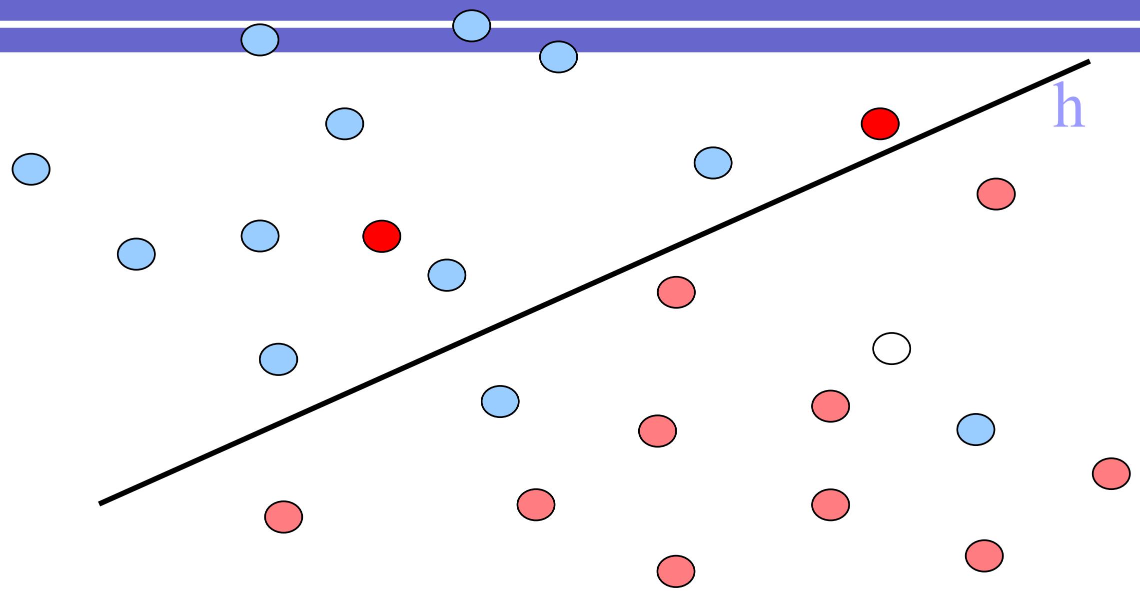




Ockham’s razor: prefer the simplest hypothesis consistent with data


$f(x)$
x

Concrete learning paradigm- linear separators

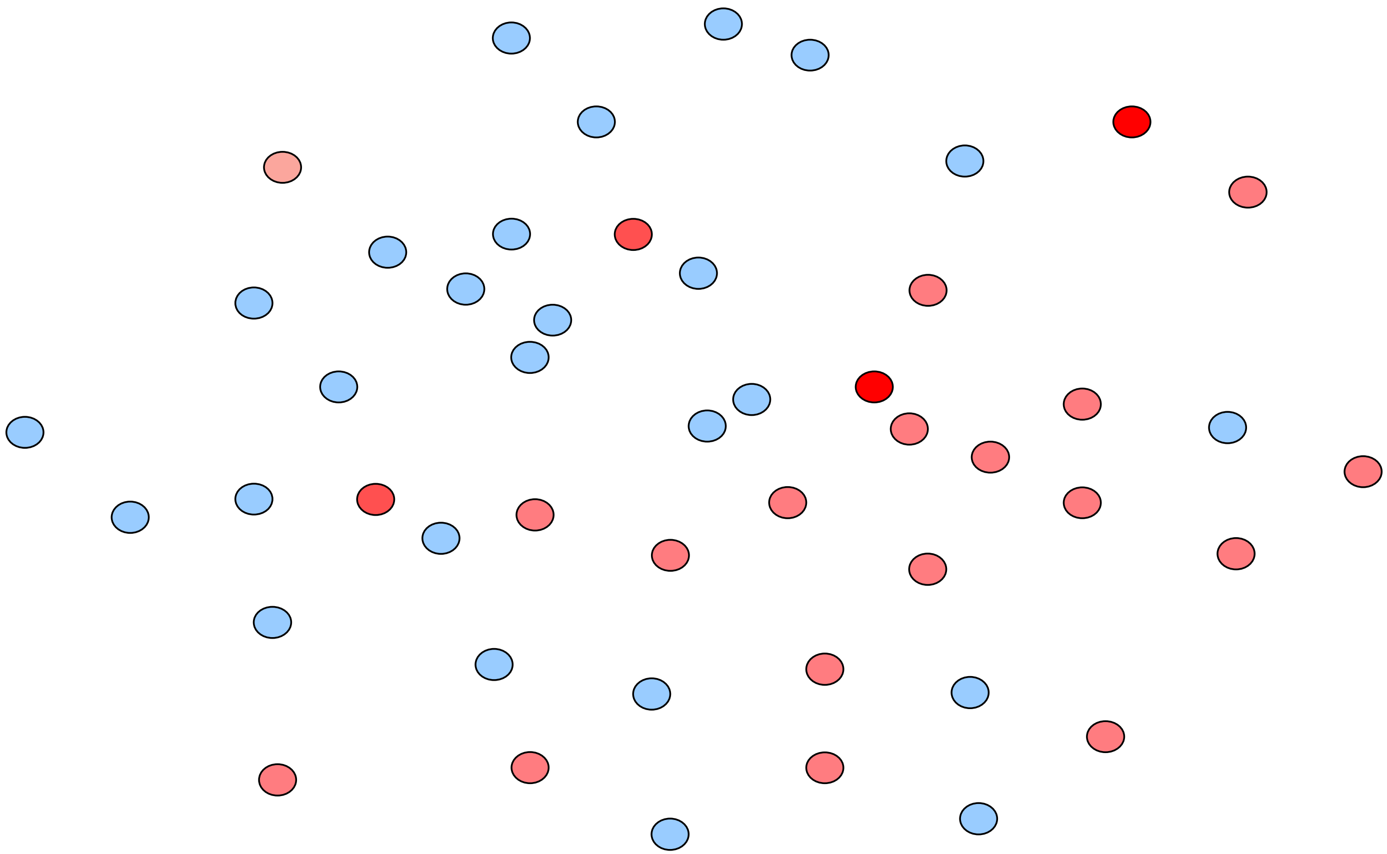


The predictor h: Sign $(\Sigma\, w_i\, x_i + b)$

(where **w** is the weight vector of the hyperplane **h**,

and $\mathbf{x=(x_1, \ldots x_i, \ldots x_n)}$ is the example to classify)

Potential problem – data may not be *linearly separable*

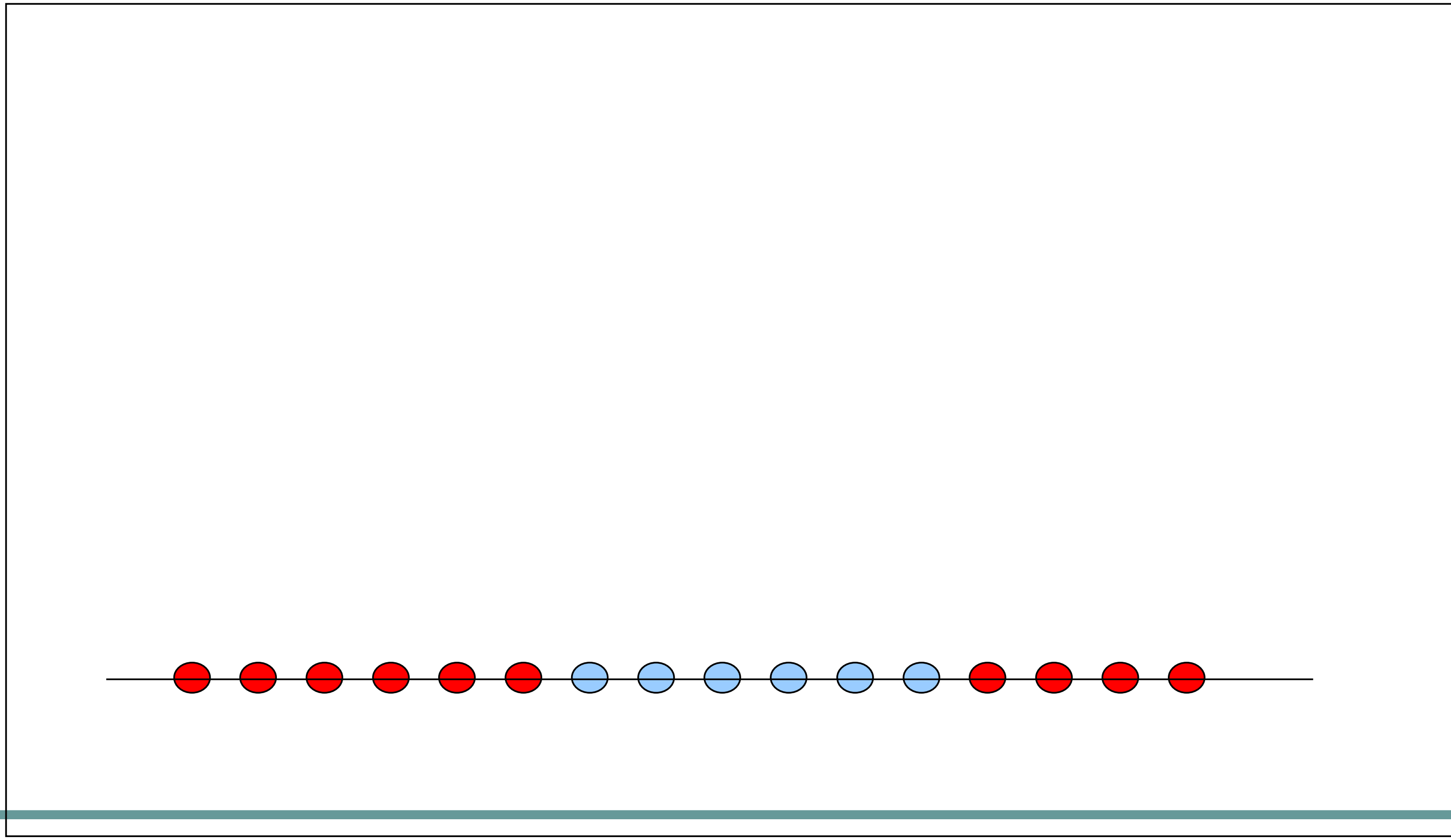


The SVM Paradigm

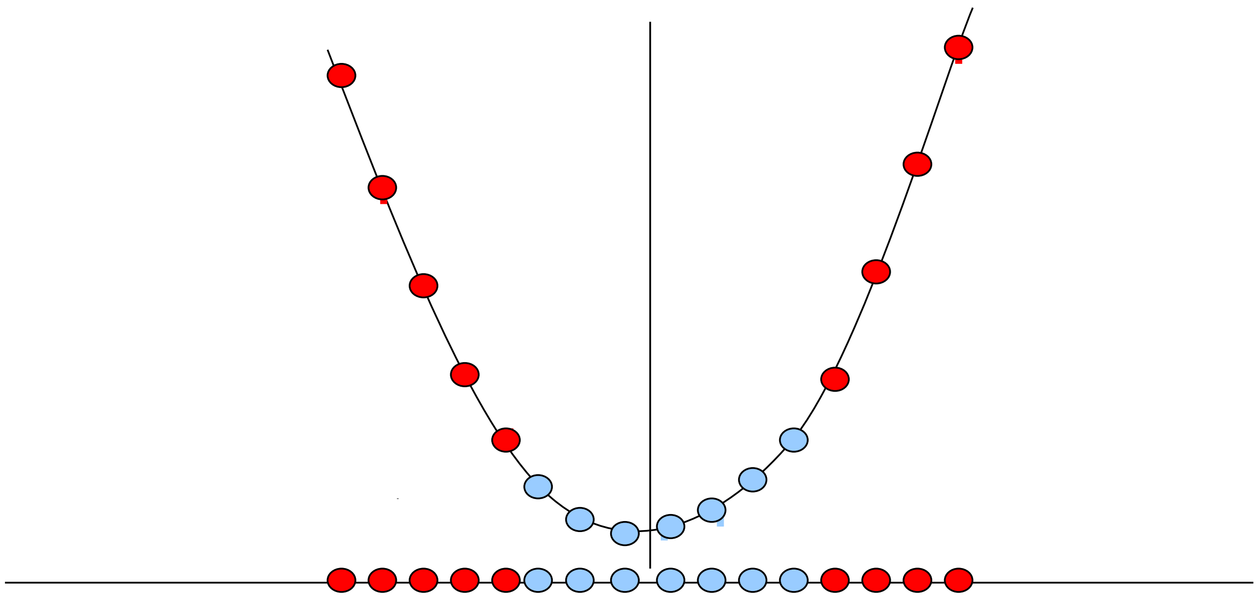
- Choose an *Embedding* of the domain X into some high dimensional Euclidean space, so that the data sample becomes (almost) linearly separable.
- Find a large-margin data-separating hyperplane in this image space, and use it for prediction.

⇨ **Important gain:** *When the data is separable, finding such a hyperplane is computationally feasible.*

The SVM Idea: an Example

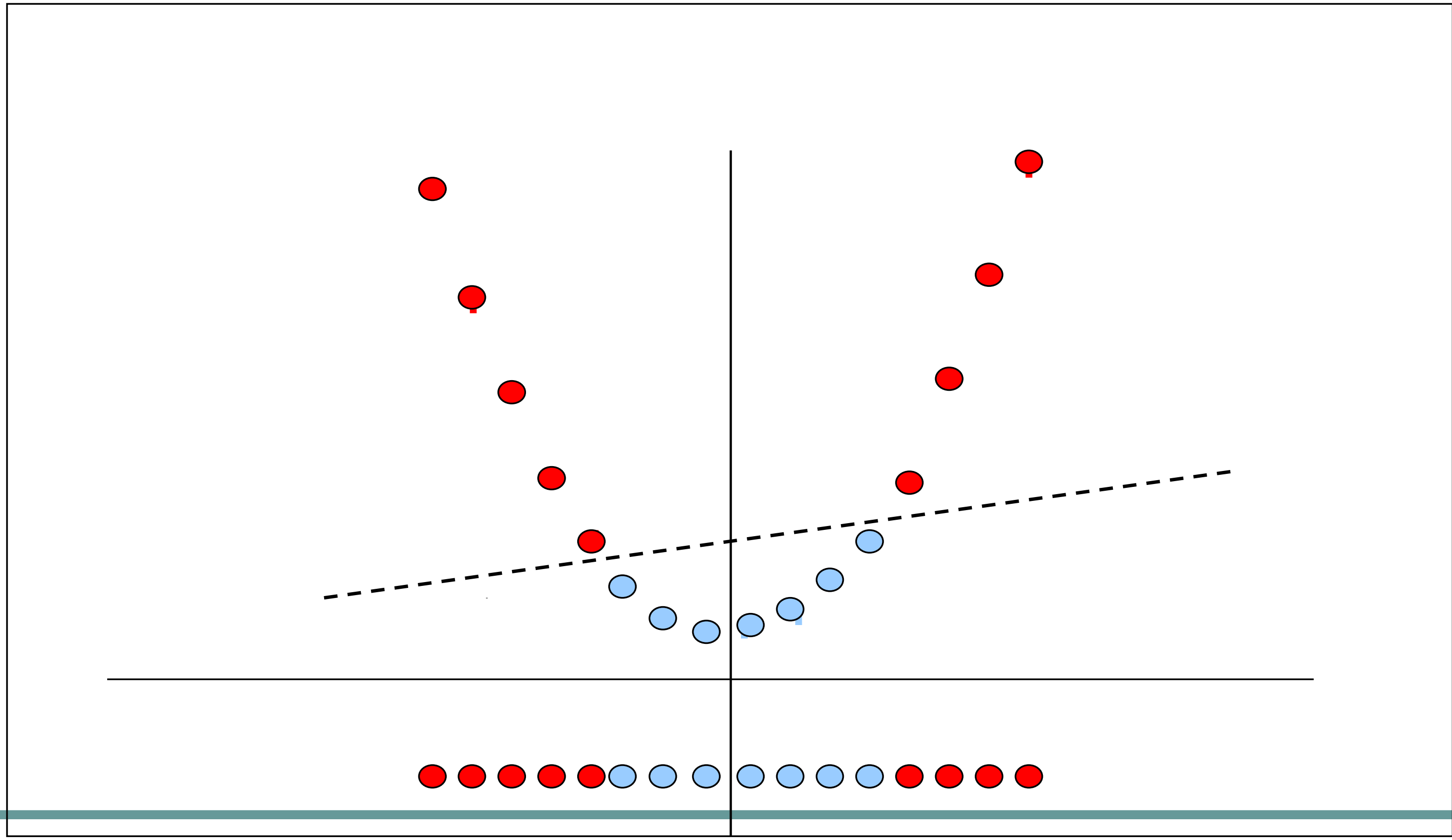


The SVM Idea: an Example



$x \mapsto (x, x^2)$

The SVM Idea: an Example



Controlling Computational Complexity

Potentially the embeddings may require very high Euclidean dimension.

How can we search for hyperplanes efficiently?

The Kernel Trick: Use algorithms that depend only on the inner product of sample points.

Kernel-Based Algorithms

Rather than define the embedding explicitly, define just the matrix of the inner products in the range space.

$$\begin{pmatrix} K(x_1x_1) & K(x_1x_2) & \cdots\cdots & K(x_1x_m) \\ \vdots & & & \vdots \\ & & K(x_ix_j) & \\ \vdots & & & \vdots \\ K(x_mx_1) & & \cdots\cdots\cdots & K(x_mx_m) \end{pmatrix}$$

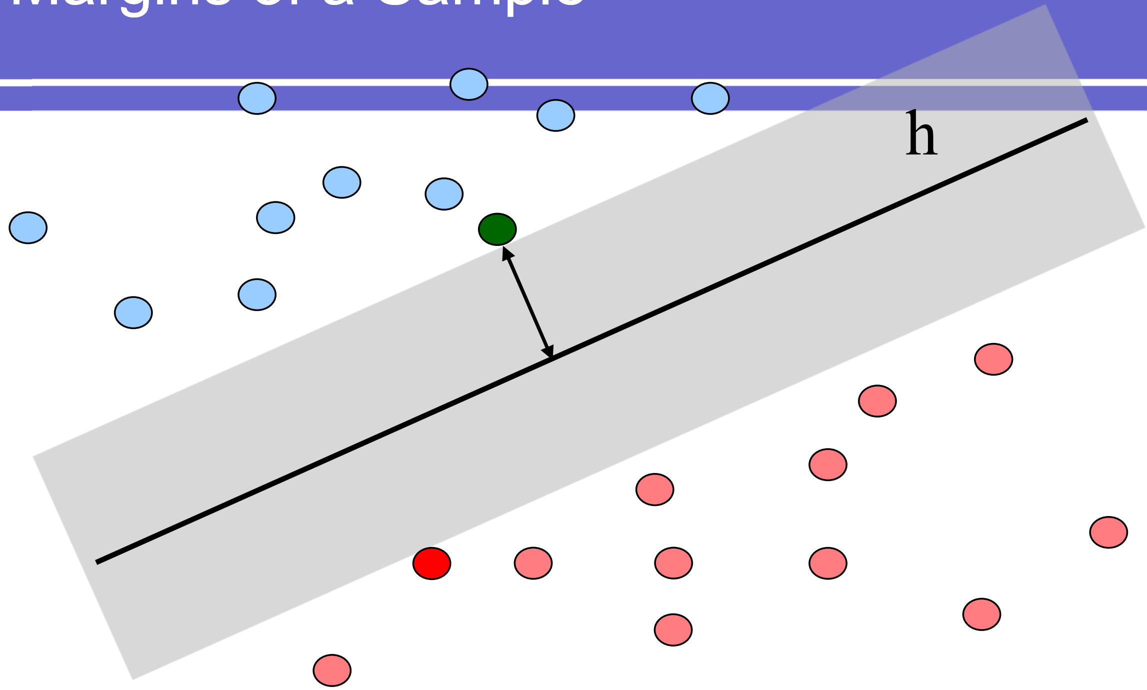
Mercer Theorem: If the matrix is symmetric and positive semi-definite, then it is the inner product matrix with respect to some embedding

Support Vector Machines (SVMs)

On input: Sample $(x_1\, y_1) \ldots (x_m y_m)$ and a kernel matrix K

Output: A "good" separating hyperplane

The Margins of a Sample



$$\max_{\text{separating } h} \; \min_{x_i} \; w_n \cdot x_i$$

(where w_n is the weight vector of the hyperplane h)

Summary of SVM learning

1. The user chooses a "Kernel Matrix"

   - a measure of similarity between input points.

2. Upon viewing the training data, the algorithm finds a linear separator the maximizes the margins (in the high dimensional "Feature Space").

- **Model Selection;**
- **Online Learning**
- **Curse of Dimensionality**
- **Bias-Variance Tradeoff**
- **Transfer Learning – Domain Adaptation**
- **BOW, Sparse Coding**
- **Incremental Learning**

References and Journals

- Text: *The Elements of Statistical Learning* by Hastie, Tibshirani, and Friedman (book website: http://www-stat.stanford.edu/~tibs/ElemStatLearn/)
- Reference books:
  - *Pattern Classification* by Duda, Hart and Stork
  - ***Pattern Recognition and Machine Learning* by C.M. Bishop**
  - *Machine Learning* by T. Mitchell
  - *Introduction to Machine Learning* by E. Alpaydin
- Some related journals / associations:
  - Machine Learning (Kluwer).
  - Journal of Machine Learning Research.
  - Journal of AI Research (JAIR).
  - Data Mining and Knowledge Discovery - An International Journal.
  - Journal of Experimental and Theoretical Artificial Intelligence (JETAI).
  - Evolutionary Computation.
  - Artificial Life.
  - Fuzzy Sets and Systems
  - IEEE Intelligent Systems (Formerly IEEE Expert)
  - **IEEE Transactions on Knowledge and Data Engineering**
  - **IEEE Transactions on Pattern Analysis and Machine Intelligence**
  - **IEEE Transactions on Systems, Man and Cybernetics**
  - Journal of AI Research
  - Journal of Intelligent Information Systems
  - Journal of the American Statistical Association
  - Journal of the Royal Statistical Society

References and Journals…

- **Pattern Recognition**
- Pattern Recognition Letters
- Pattern Analysis and Applications.
- Computational Intelligence .
- Journal of Intelligent Systems .
- Annals of Mathematics and Artificial Intelligence.
- IDEAL, the online scientific journal library by Academic Press.
-
- ACM (Association for Computing Machinery).
- Association for Uncertainty in Artificial Intelligence.
- ACM SIGAR
- **ACM SIGMOD**
- American Statistical Association.
- Artificial Intelligence
- Artificial Intelligence in Engineering
- Artificial Intelligence in Medicine
- Artificial Intelligence Review
- Bioinformatics
- Data and Knowledge Engineering
- Evolutionary Computation

Some Conferences & Workshops

- Congress on Evolutionary Computation
- European Conference on Machine Learning and Principles and Practice of Knowledge Discovery
- **The ACM SIGKDD International Conference on Knowledge Discovery and Data Mining**
- National Conference on Artificial Intelligence
- Genetic and Evolutionary Computation Conference
- **International Conference on Machine Learning (ICML, ECML, ICLR**)
- Conference on Autonomous Agents and Multiagent Systems
- European Symposium on Artificial Neural Networks Advances in Computational Intelligence and Learning
- Artificial and Ambient Intelligence
- Computational Intelligence in Biomedical Engineering
- IEEE International Symposium on Approximate Dynamic Programming and Reinforcement Learning
- International Joint Conference on Artificial Intelligence (**IJCAI**)

ECCAI (European Coordinating Committee on Artificial Intelligence).

AAAI (American Association for Artificial Intelligence).

NIPS, CVPR