STATISTICAL LEARNING Theory (SLT): CS6464

Statistical learning theory is a framework for machine learning, drawing from the fields of statistics and functional analysis.

Statistical learning theory deals with the problem of finding a predictive function based on data.

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

- **Supervised learning,**
- **Unsupervised learning,**
- **Semi-supervised learning**
- **Transfer Learning**
- **Online learning, and**
- **Reinforcement learning.**

From the perspective of statistical learning theory, supervised learning is best understood.

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.

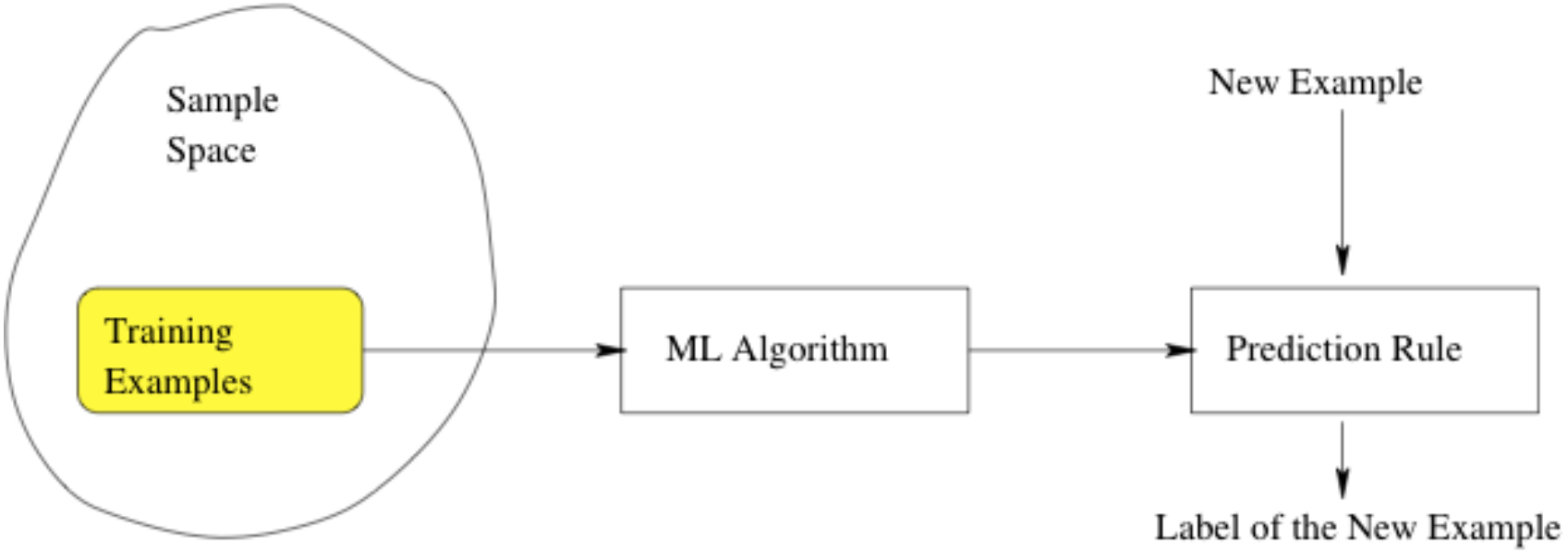
The algorithm takes these previously labeled samples and uses them to induce a classifier. This classifier is a function that assigns labels to samples including the samples that have never been previously seen by the algorithm.

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	Statistics
network, graphs	model
weights	parameters
learning	fitting
generalization	test set performance
supervised learning	regression/classification
unsupervised learning	density estimation, clustering

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

Statistical Modelling is ...
formalization of relationships between variables in the form of mathematical equations.



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.

Statistical Modelling is ...
a subfield of mathematics which deals with finding relationship between variables to predict an outcome

Statistical modeling usually work with a number of assumptions.

For instance a linear regression assumes :

1. **Linear relation between independent and dependent variable**
2. **Homoscedasticity**
3. **Mean of error at zero for every dependent value**
4. **Independence of observations**
5. **Error should be normally distributed for each value of dependent variable**

{ Homoscedasticity - all random variables in the sequence or vector have the same finite variance. This is also known as homogeneity of variance}.

Similarly Logistic regressions comes with its own set of assumptions. Even a non-linear model has to comply to a continuous segregation boundary.

Machine Learning algorithms do assume a few of these things but in general are spared from most of these assumptions. The biggest advantage of using a Machine Learning algorithm is that there might not be any continuity of boundary. Also, we need not specify the distribution of dependent or independent variable in a machine learning algorithm.

In addition to performance bounds, 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.

Contents	# of classes (approxm.)
Learning Problem, Risk functions, Statistical Decision Theory	4
Ill posed and well posed problems	2
Least Square Regression, Bias Variance tradeoff	6
Linear Models of Regression, Subset Selection methods, Shrinkage methods, Ridge regression	5
LASSO + LAR	1
Pattern Recognition – Statistical + ANN algos.	4
ERM (+Tikhonov Regularization), Iterative regularization by early stopping, SRM	2
BOW + Online Learning + TL & DA	2
ADMM, Proximal gradient, BP, AE	4
SVM, Kernel, VC dimension, RKHS	3

Textbooks:

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

Pattern Recognition and Machine Learning. Christopher Bishop.

Data Mining: Tools and Techniques, 3rd Edition. Jiawei Han and Michelline Kamber.

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

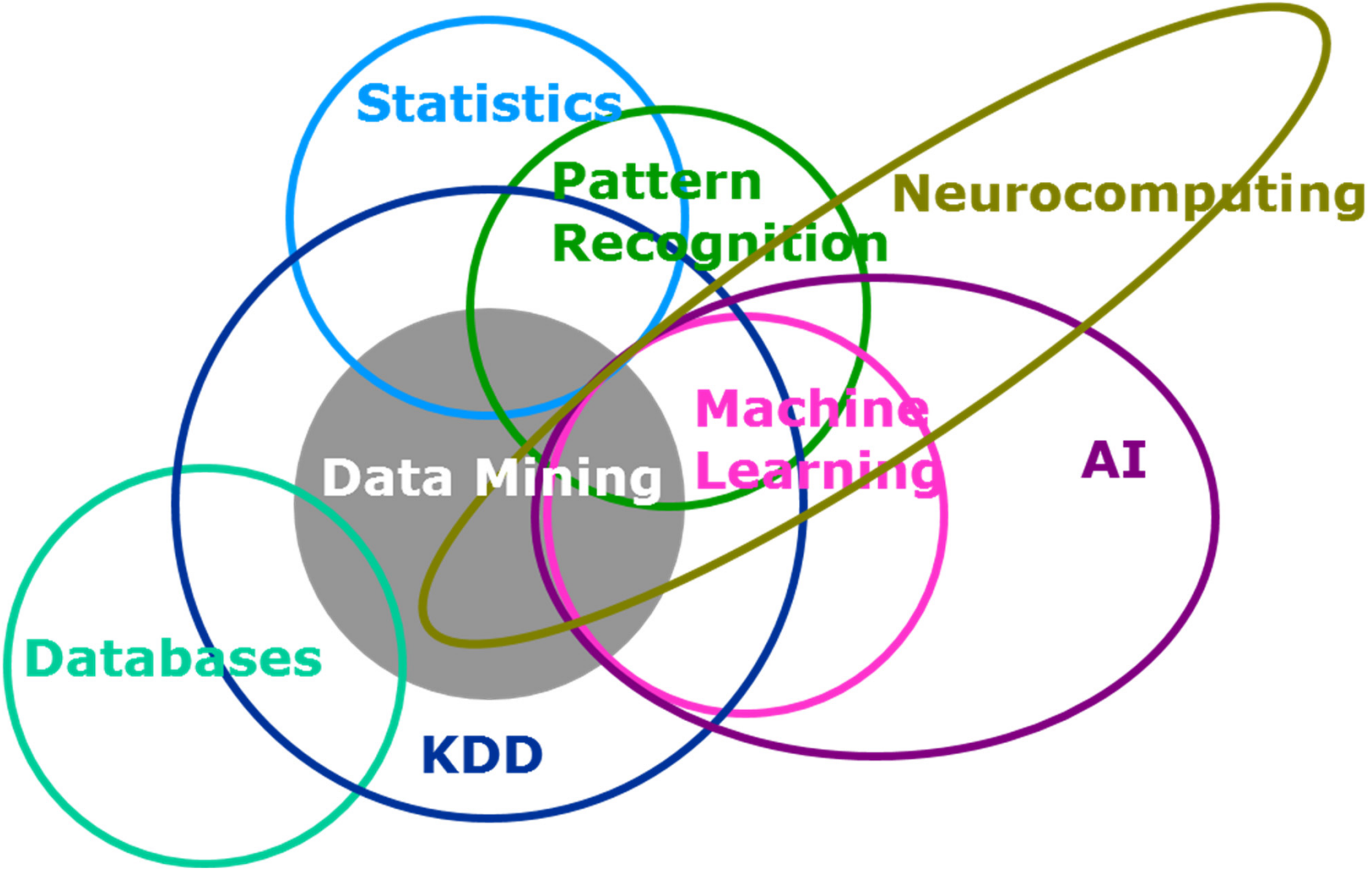
http://www.cse.iitm.ac.in/~vplab/statistical_learning_theory.html

- *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 $L: \mathcal{Y} \times \mathcal{Y}' \to \mathbb{R}_+$. 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
(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

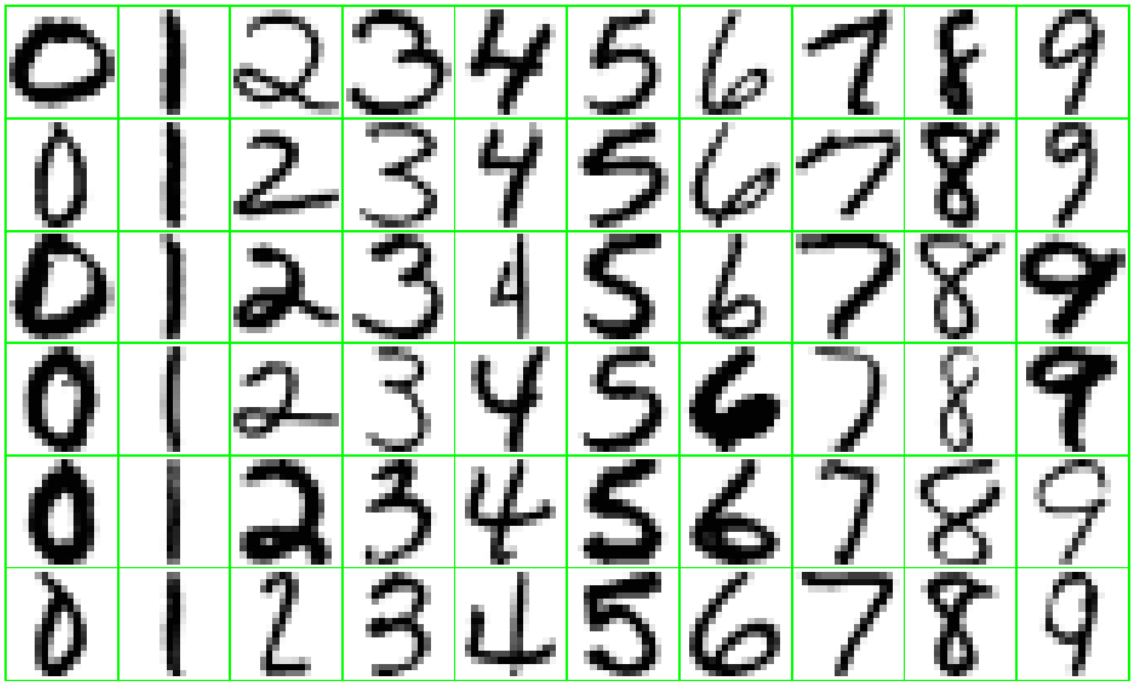
Types of Data

- **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

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



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**

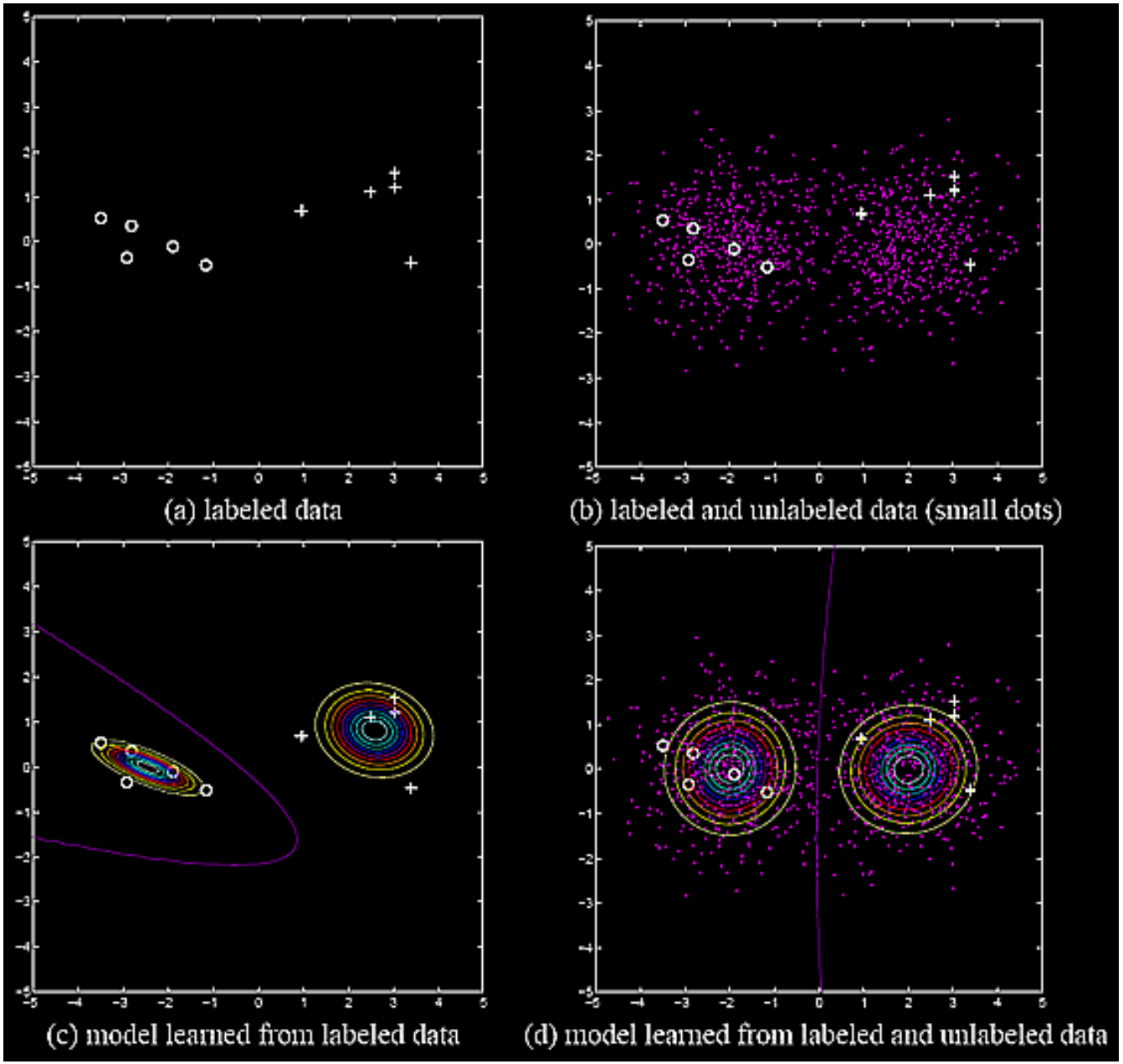
Unsupervised Learning

- 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

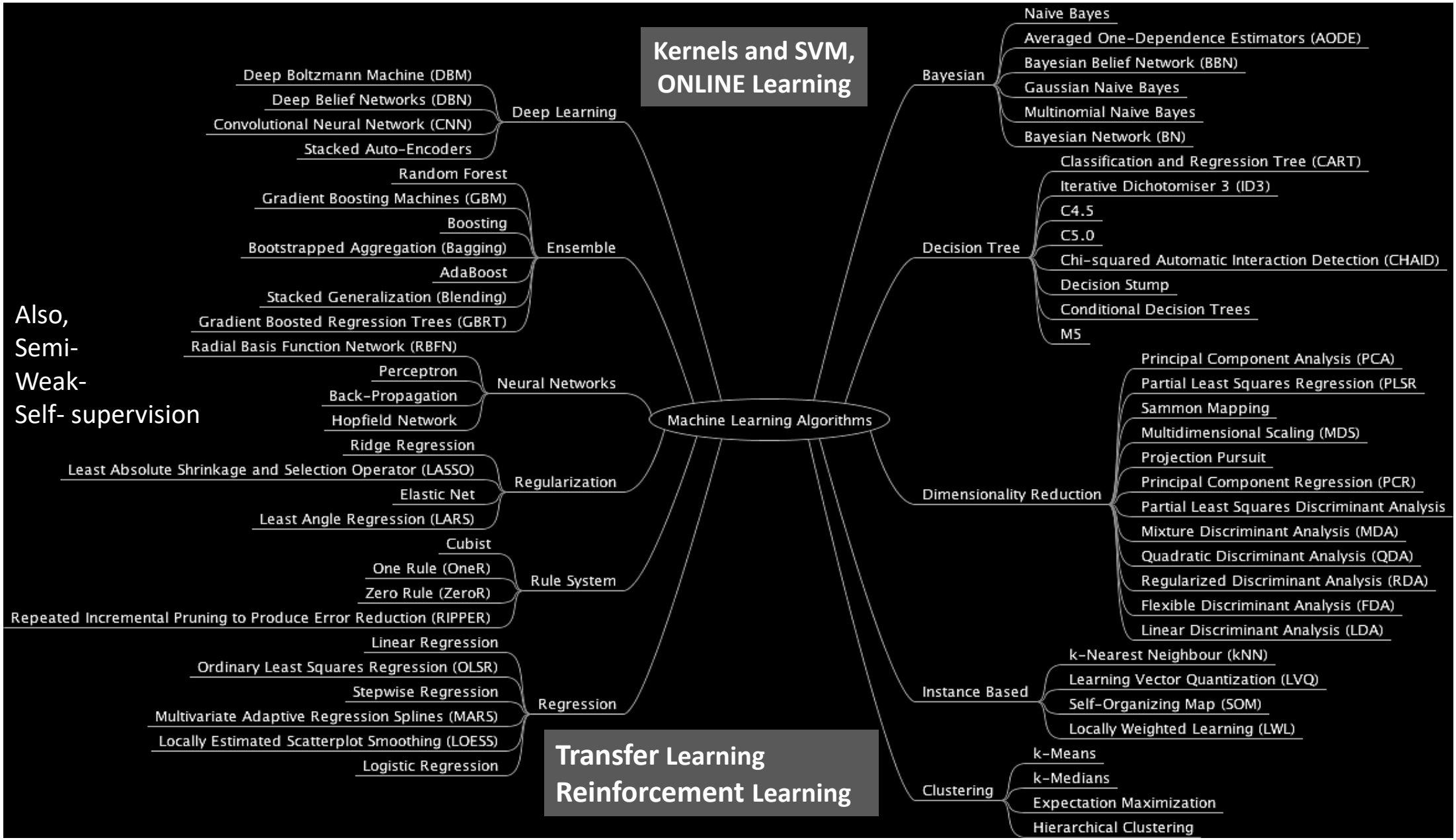
In the **self-supervised** learning technique, the model depends on the underlying structure of data to predict outcomes. It involves no labelled data.

However, in **semi-supervised learning**, we still provide a **small amount** of labelled data.

Weak supervision is a branch of machine learning where **noisy, limited, or imprecise sources** are used to provide supervision signal for labeling large amounts of training data in a supervised learning setting.

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!


Machine Learning Algorithms
Kernels and SVM, ONLINE Learning
Transfer Learning
Reinforcement Learning
Also, Semi- Weak- Self- supervision
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

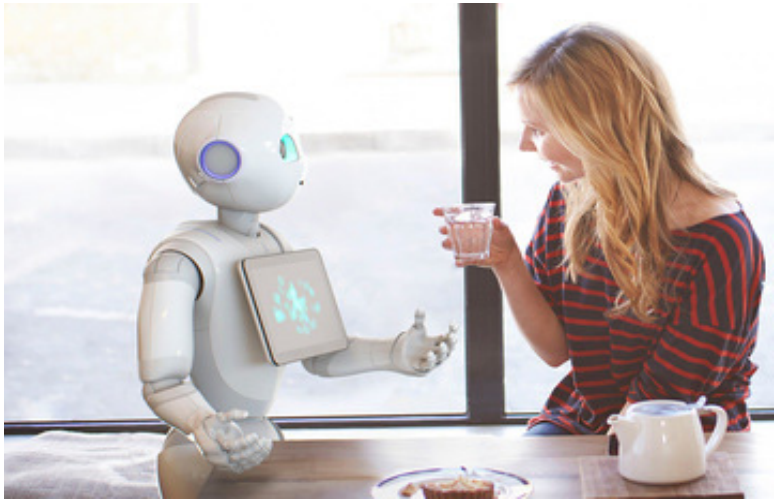
Computer Scientists' Contribution to Statistics: Kernel Methods

Vladimir Vapnik

Jerome H. Friedman

Applications:

- **Document Classification and email SPAM filtering;**
- **Object Recognition + face, fingerprint , hand-writing, printed text (OCR), inpainting**
- **Action Classification in videos; Video surveillance, Self-driving cars**
- **Exit polls, Stock Market, Weather, Social media**
- **Identifying patterns/clusters/structures in big data**
- **Search Engines, market analysis, Robotics**
- **Matrix completion**
- **Virtual Assistance – Alexa etc.**



- **Manufacturing; Quality Control, Customer support, product recommendations**
- **Health care, collaborative filtering, software/hardware design**
- **Agriculture**

Learning from Observations

- **Learning agents**
- **Inductive learning**
- **Decision tree learning**

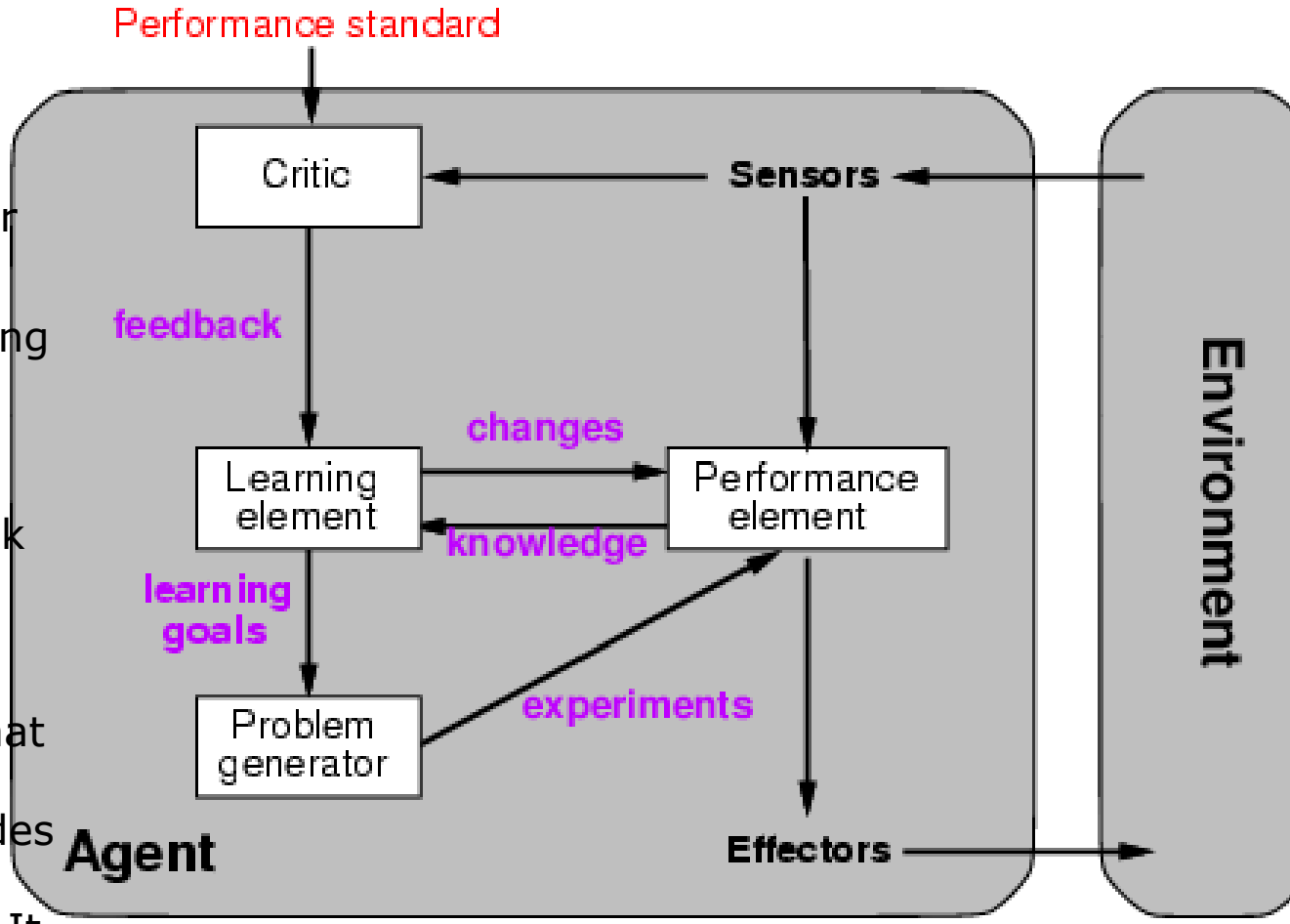
Learning agents

In artificial intelligence, an **intelligent agent (IA)** is an autonomous entity which observes through sensors and acts upon an environment using actuators (i.e. it is an agent) and directs its activity towards achieving goals.

Intelligent agents may also learn or use knowledge to achieve their goals.

"learning element", is responsible for making improvements, and the "performance element", is responsible for selecting external actions.

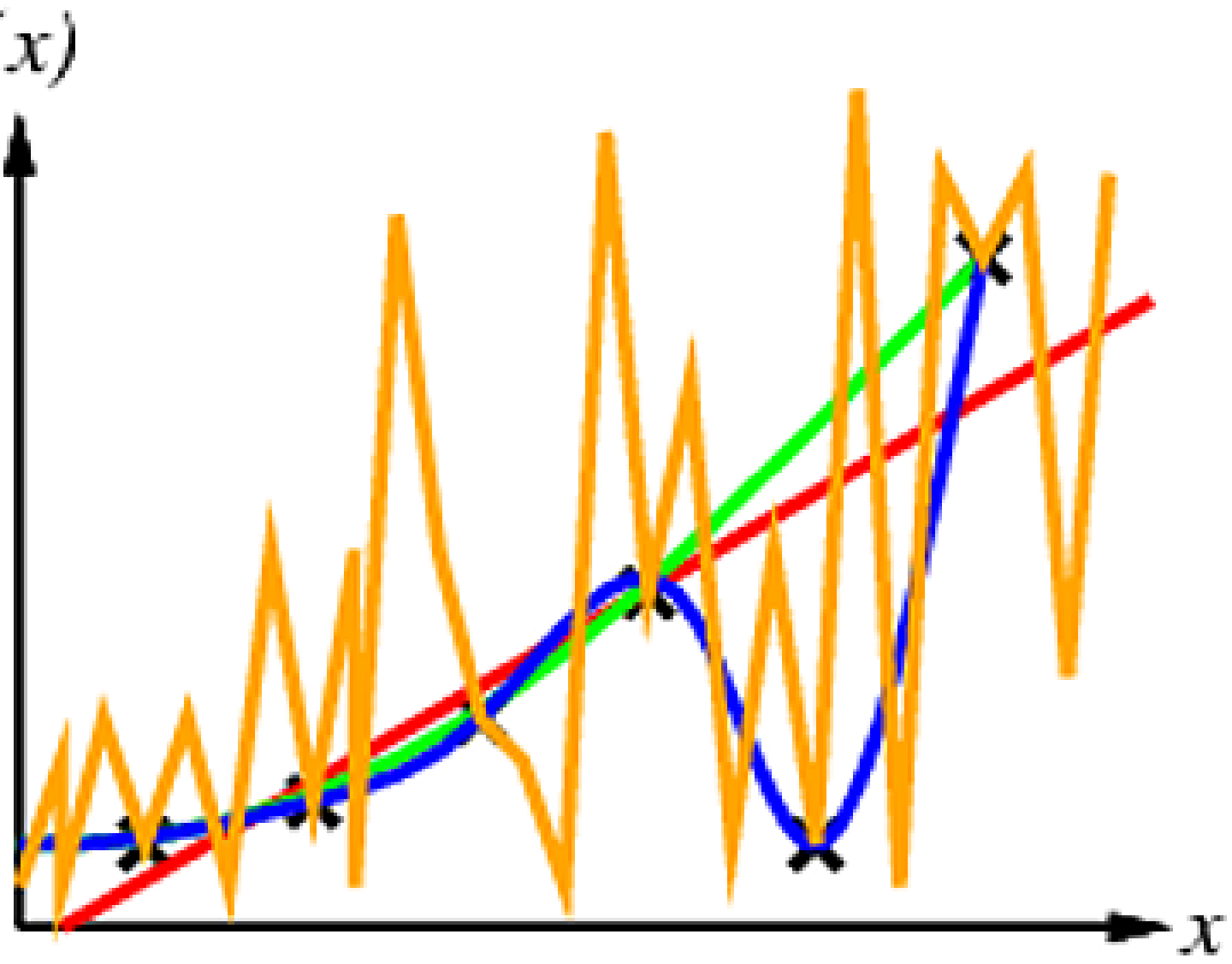
The learning element uses feedback from the "critic" on how the agent is doing and determines how the performance element should be modified to do better in the future. The performance element is what we have previously considered to be the entire agent: it takes in percepts and decides on actions. The last component of the learning agent is the "problem generator". It is responsible for suggesting actions that will lead to new and informative experiences.

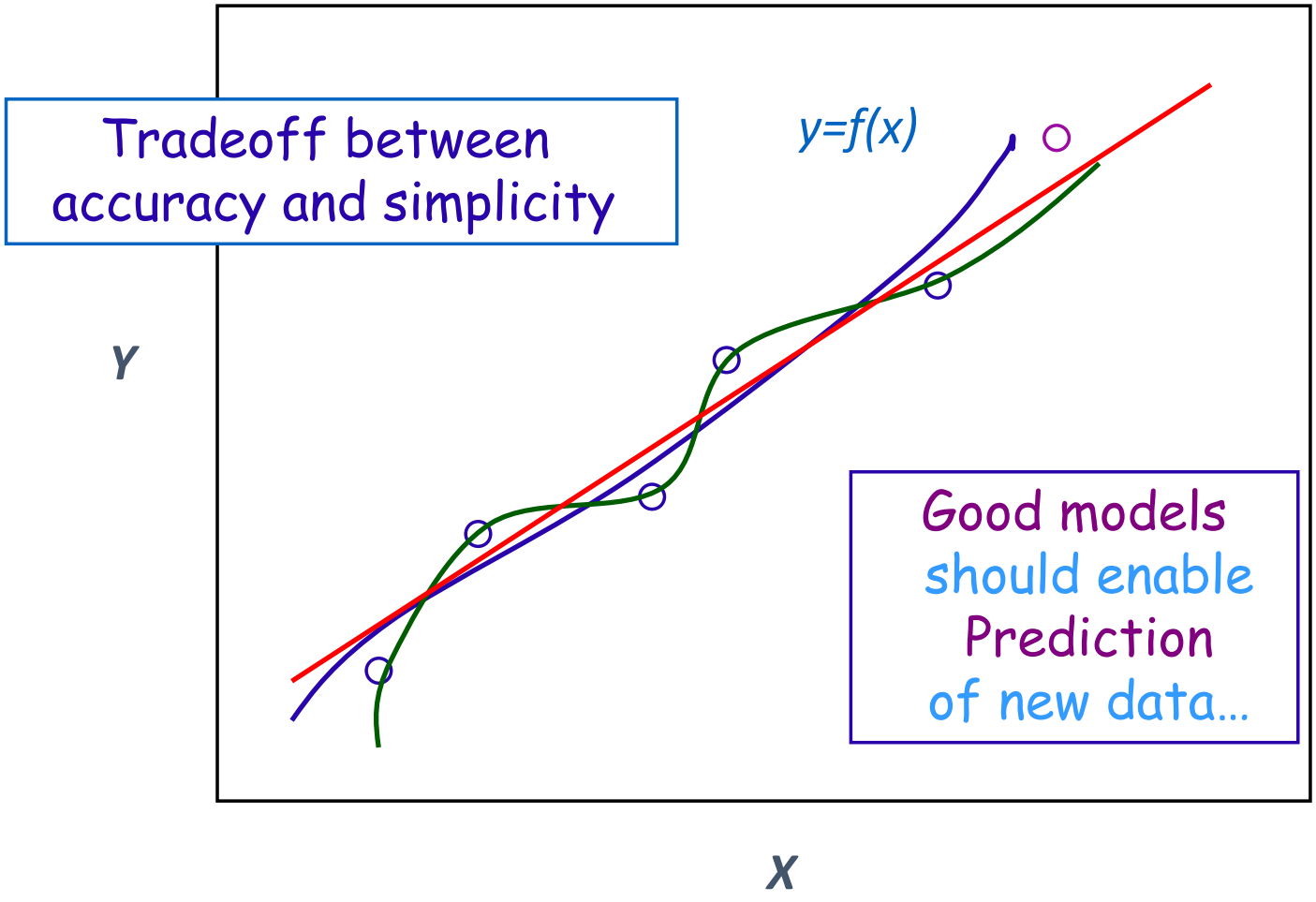


Ind

- Construct/adjust
- (*h* is consistent if
-
- E.g., curve fitting:

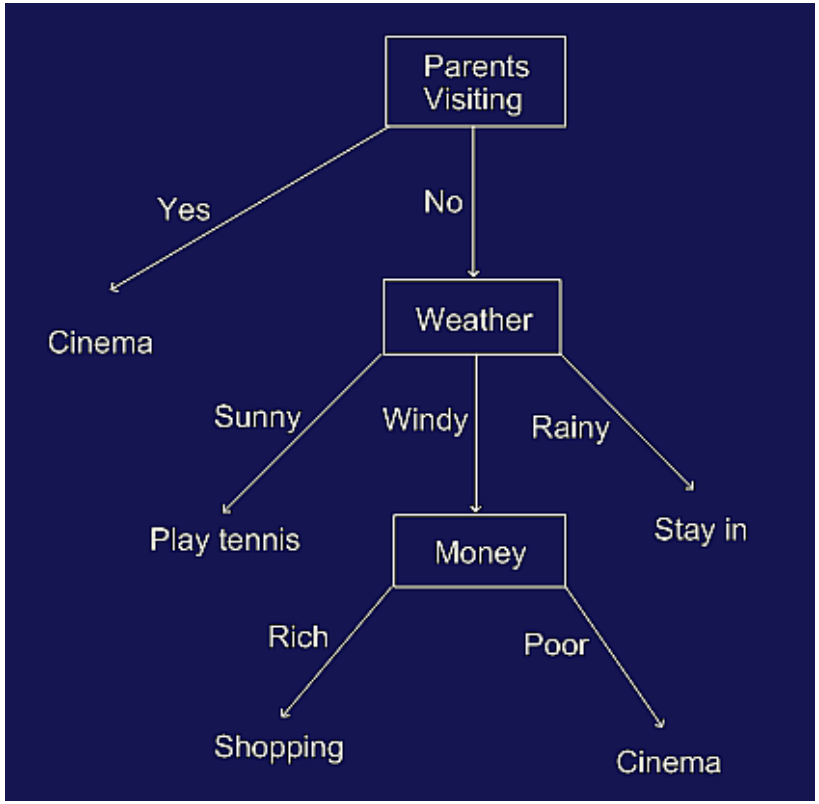
- Ockham's razor:
  data

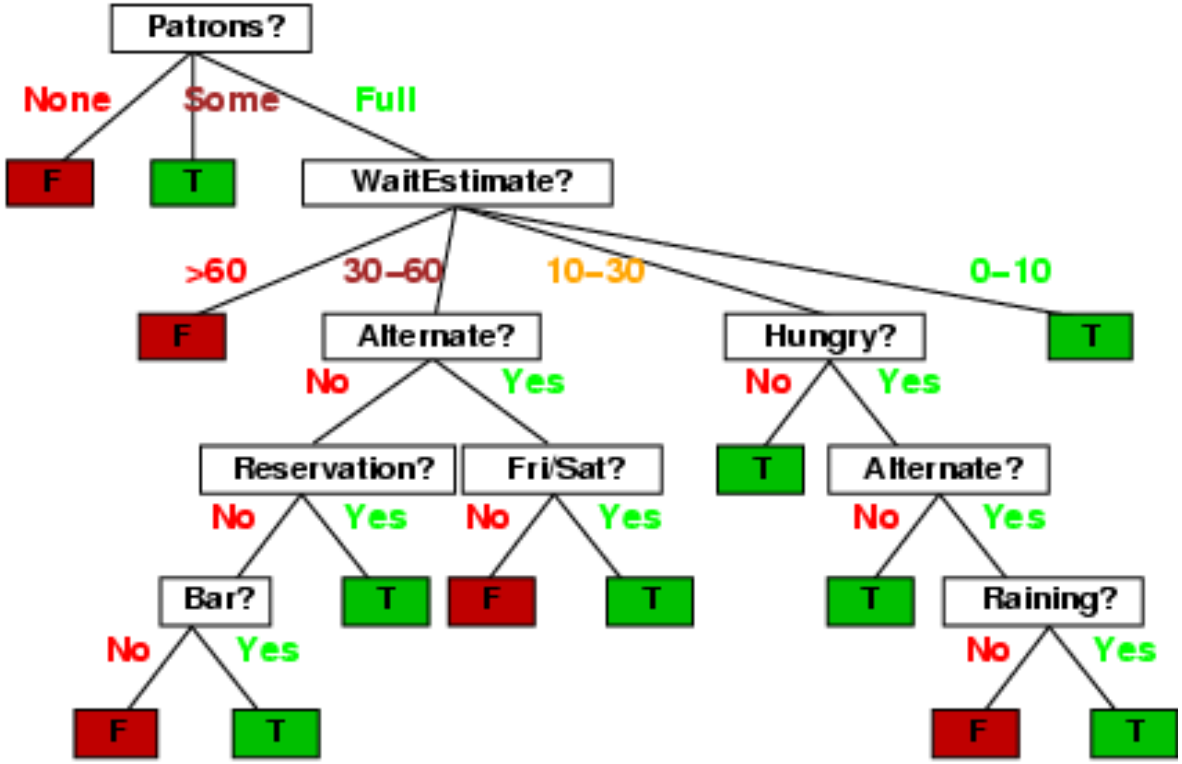



Tradeoff between
accuracy and simplicity
y=f(x)
Y
Good models
should enable
Prediction
of new data…
X

Decision trees

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





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.

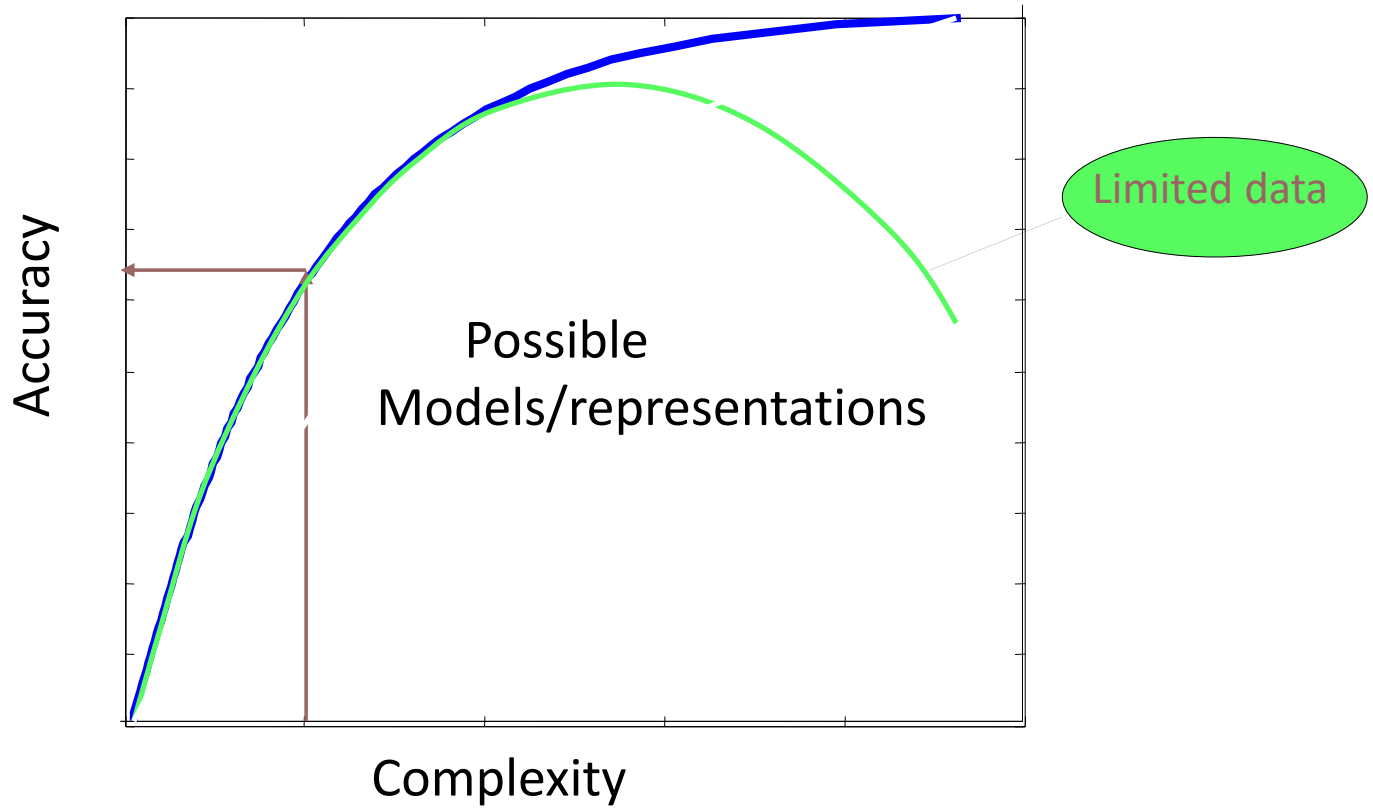
A purely online model would learn based on just the new input , the current best predictor and some extra stored information (which is usually expected to have storage requirements independent of training data size).

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.

- We are interested in hypothesis *generation* rather than hypothesis testing.
- We wish to make *no prior assumptions*

  about the structure of our data.
- We develop algorithms for *automated generation* of hypotheses.
- We are concerned with *computational efficiency.*

A Fundamental Dilemma of Science: Model Complexity vs Prediction Accuracy



In science, the term **model** can mean several different things (e.g., an idea about how something works or a physical model of a system that can be used for testing or demonstrative purposes). However, as a research method, modeling often means creating a **mathematical model** — a set of equations that indirectly represents a real system. These equations are based on relevant information about the system and on sets of hypotheses about how the system works.

Given a set of parameters, a **model can generate expectations** about how the system will behave in a particular situation. A model and the hypotheses it is based upon are supported when the model generates expectations that match the behavior of its real-world counterpart. Modeling often involves idealizing the system in some way — leaving some aspects of the real system out of the model in order to isolate particular factors or to make the model easier to work with computationally.

Hypothesis: A proposed explanation for a fairly narrow set of phenomena, usually based on **prior experience, scientific background knowledge, preliminary observations, and logic**. These reasoned explanations are not guesses — of the wild or educated variety. When scientists formulate new hypotheses, they are usually based on prior experience, scientific background knowledge, preliminary observations, and logic.

Theories, on the other hand, are **broad explanations for a wide range of phenomena. They are concise** (i.e., generally don't have a long list of exceptions and special rules), coherent, systematic, predictive, and **broadly applicable**. In fact, theories often integrate and generalize many hypotheses.

Hypotheses are proposed explanations for a narrow set of phenomena. They are not guesses.

Theories are powerful explanations for a wide range of phenomena. Accepted theories are not tenuous.

Forming hypotheses — scientific explanations — can be difficult for students. It is often easier for students to generate an expectation (what they think will happen or what they expect to observe) based on prior experience than to formulate a potential explanation for that phenomena. You can help students go beyond expectations to generate real, explanatory hypotheses by providing sentence stems for them to fill in: "I expect to observe A because B." Once students have filled in this sentence you can explain that B is a hypothesis and A is the expectation generated by that hypothesis.

< http://undsci.berkeley.edu/article/howscienceworks_19>

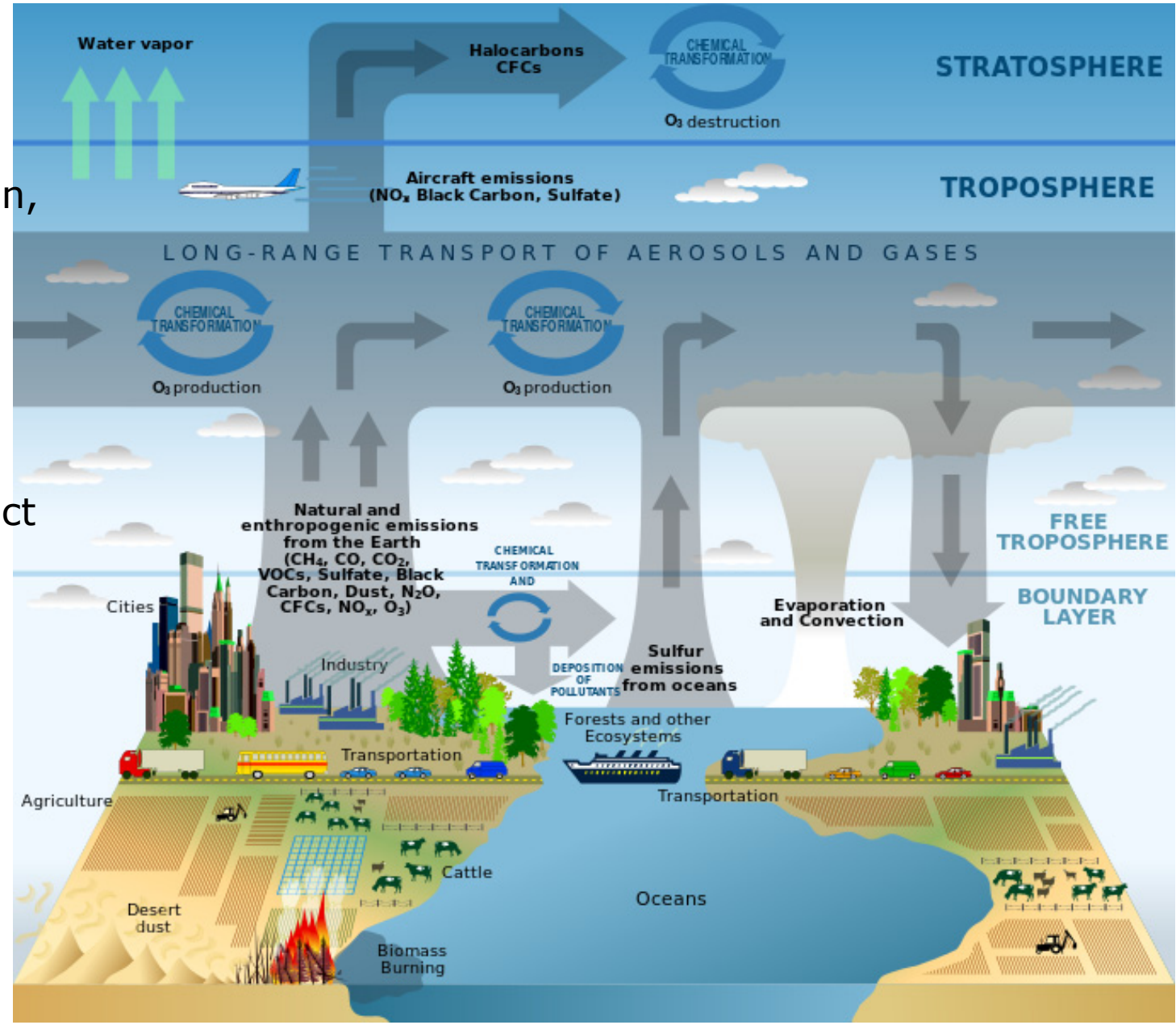
A scientific model represents objects, phenomena, and physical processes in a consistent and logical way.

Examples of SCIENTIFIC MODELS:

A model of the motions of the sun, moon and earth (which you participated in last year)

A model of predicting eclipses.

Models that explain weather phenomena can be used to predict weather.



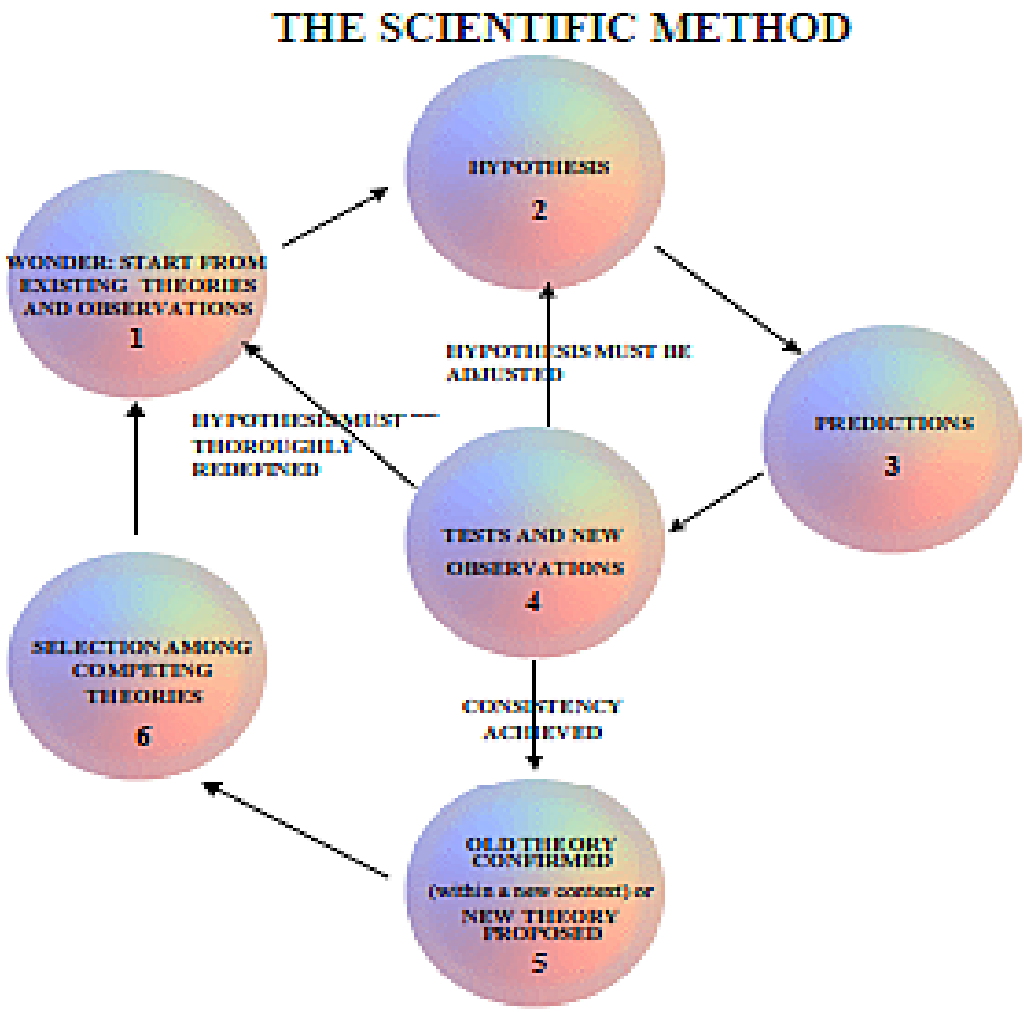
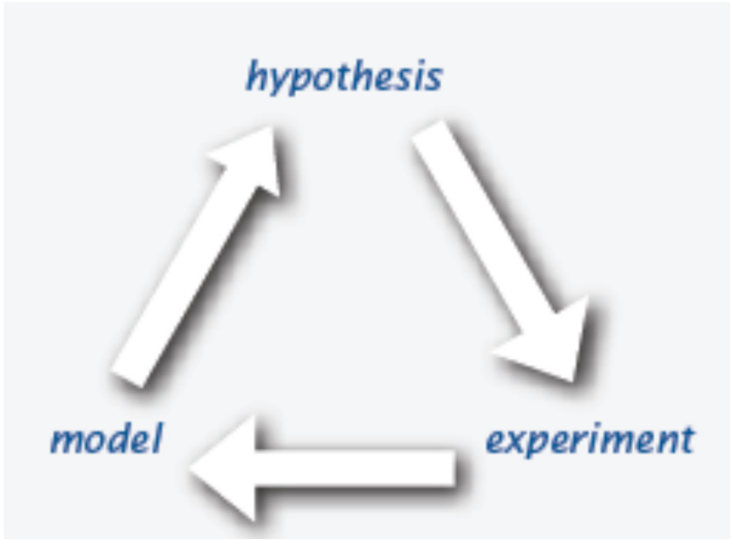


Figure 2 Diagram describing iterative nature of the scientific method (hypothetico-deductive)



Hypothesis: Running time is $\sim aN^b$

Theorem: Running time is $O(N^b)$

Problem Outline

- We are interested in
  (*automated*) Hypothesis Generation,
  rather than traditional *Hypothesis Testing*

- First obstacle: The danger of overfitting.

- First solution:
  Consider only a limited set of candidate hypotheses.

A **statistical hypothesis** is a hypothesis that is testable on the basis of observing a process that is modeled via a set of random variables. A **statistical hypothesis test** is a method of ***statistical inference***.

Commonly, two statistical data sets are compared, or a data set obtained by sampling is compared against a synthetic data set from an idealized model. A hypothesis is proposed for the statistical relationship between the two data sets, and this is compared as an *alternative to an idealized null hypothesis* that proposes no relationship between two data sets. The comparison is deemed *statistically significant* if the relationship between the data sets would be an unlikely realization of the null hypothesis according to a threshold probability—the significance level. Hypothesis tests are used in determining what outcomes of a study would lead to a rejection of the null hypothesis for a pre-specified level of significance.

Statistical inference is the process of deducing properties of an underlying distribution by analysis of data. Inferential statistical analysis infers properties about a population: this includes testing hypotheses and deriving estimates. The observed data is assumed to be sampled from a larger population.

Empirical Risk Minimization Paradigm

- Choose a *Hypothesis Class* ***H*** of subsets of ***X***.

- For an input sample ***S***, find some ***h*** in ***H*** that fits ***S*** well.

- For a new point ***x***, predict a label according to its membership in ***h***.

The Mathematical Justification

Assume both a training sample ***S*** and the test point ***(x, l)*** are generated i.i.d. by the same distribution over $X \varepsilon \{0,1\}$ then,

If ***H*** *is not too rich (in some formal sense) then,*

for every ***h*** *in* ***H****, the training error of* ***h*** *on the sample* ***S*** *is a good estimate of its probability of success on the new* ***x*** *.*

In other words – there is no overfitting

Statistical learning theory takes the perspective that there is some unknown probability distribution over the product space

$$Z = X \otimes Y$$

And unknown $p(z) = p(\vec{x}, y)$

The training set is made up of samples from this probability distribution, and is denoted by:

$$S = \{(\vec{x}_1, y_1), \ldots, (\vec{x}_n, y_n)\} = \{\vec{z}_1, \ldots, \vec{z}_n\}$$

Every x_i is an input vector from the training data, and y_i is the output that corresponds to it.

The inference problem consists of finding a function $f : X \mapsto Y$ **such that f(x) ~ y.**

Let $\mathscr{H}$ be a space of functions called the hypothesis space. The hypothesis space is the space of functions the algorithm will search through.

Let $V(f(\vec{x}), y)$ be the loss functional,
a metric for the difference between the predicted value *f(x)* and the actual value *y*.
The expected risk is defined to be

$$I[f] = \int_{X \otimes Y} V(f(\vec{x}), y) p(\vec{x}, y) d\vec{x} dy$$

The target function, the best possible function f that can be chosen, is given by the f that satisfies $\inf_{f \in \mathcal{H}} I[f]$

Because the probability distribution $p(x, y)$ is unknown, a proxy measure for the expected risk must be used. This measure is based on the training set, a sample from this unknown probability distribution. It is called the empirical risk:

$$I_S[f] = \frac{1}{n} \sum_{i=1}^{n} V(f(\vec{x}_i), y_i)$$

A learning algorithm that chooses the function f that minimizes the empirical risk is called empirical risk minimization.

The loss function also affects the convergence rate for an algorithm. It is important for the loss function to be convex. Different loss functions are used depending on whether the problem is one of regression or one of classification.

The most common loss function for *regression* is the square loss function. This familiar loss function is used in ordinary least squares regression. The form is:

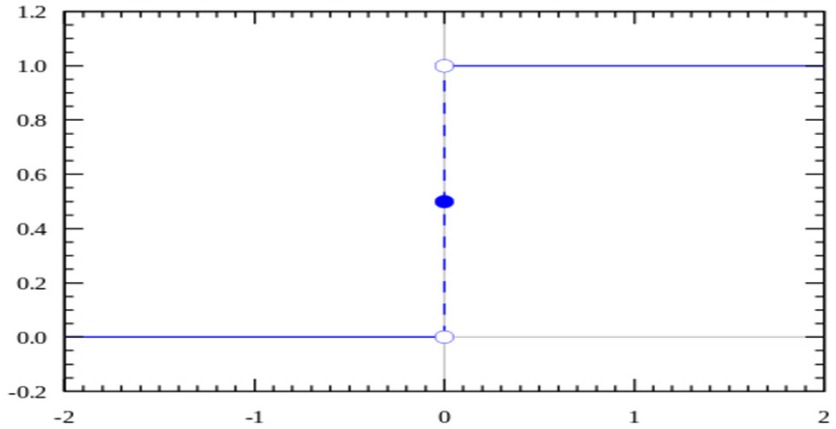
$$V(f(\vec{x}), y) = (y - f(\vec{x}))^2$$

$$V(f(\vec{x}), y) = |y - f(\vec{x})|$$

The 0-1 indicator function is the most natural loss function for *classification*. It takes the value 0 if the predicted output is the same as the actual output, and it takes the value 1 if the predicted output is different from the actual output. For binary classification with $Y = \{-1, 1\}$, this is:

$$V(f(\vec{x}), y) = \theta(-yf(\vec{x}))$$

where, θ is the Heaviside step function.

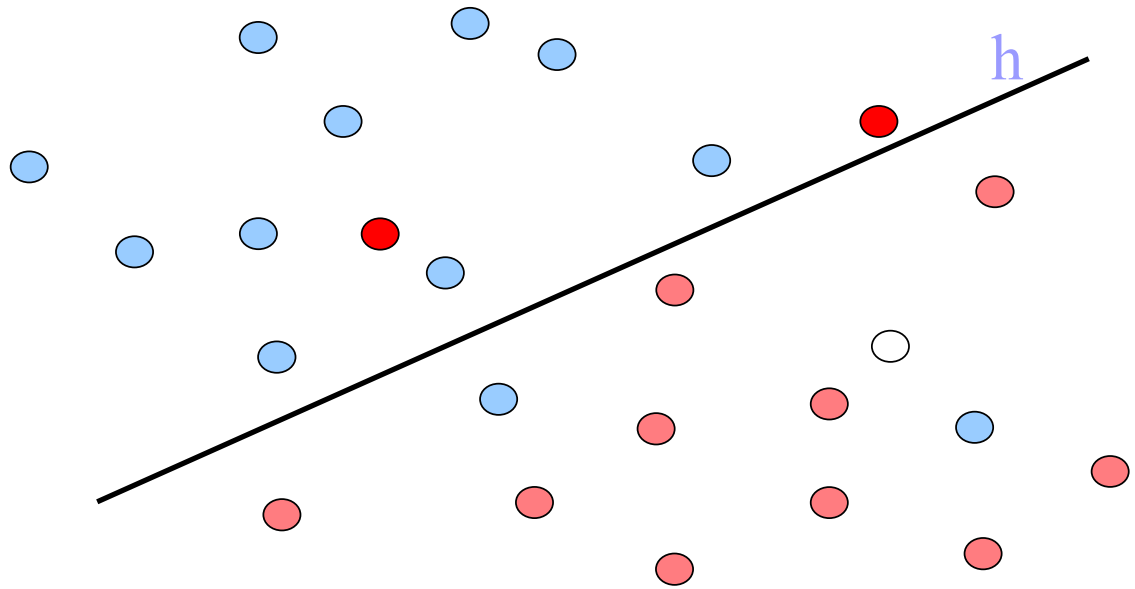


One form of *regularization* is Tikhonov regularization. This consists of minimizing

$$\frac{1}{n}\sum_{i=1}^{n} V(f(\vec{x}_i, y_i)) + \gamma \|f\|_{\mathcal{H}}^2$$

Where, γ is a fixed and positive parameter, the regularization parameter. Tikhonov regularization ensures existence, uniqueness, and stability of the solution. Hypothesis space $\mathscr{H}$ can be RKHS.

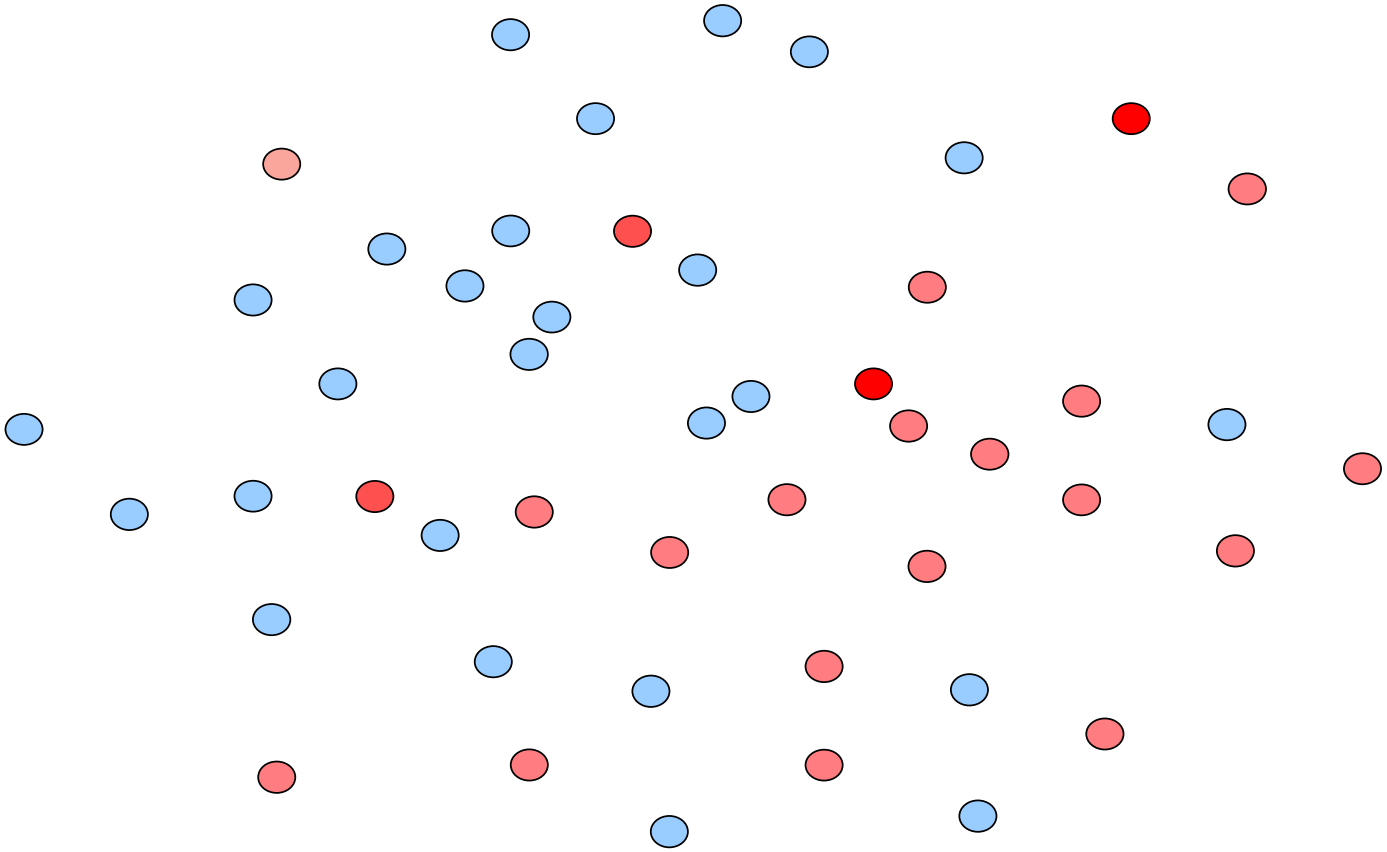
Concrete learning paradigm- linear separators



The predictor h: $\text{Sign}\ (\Sigma\ w_i\ x_i + b)$

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

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

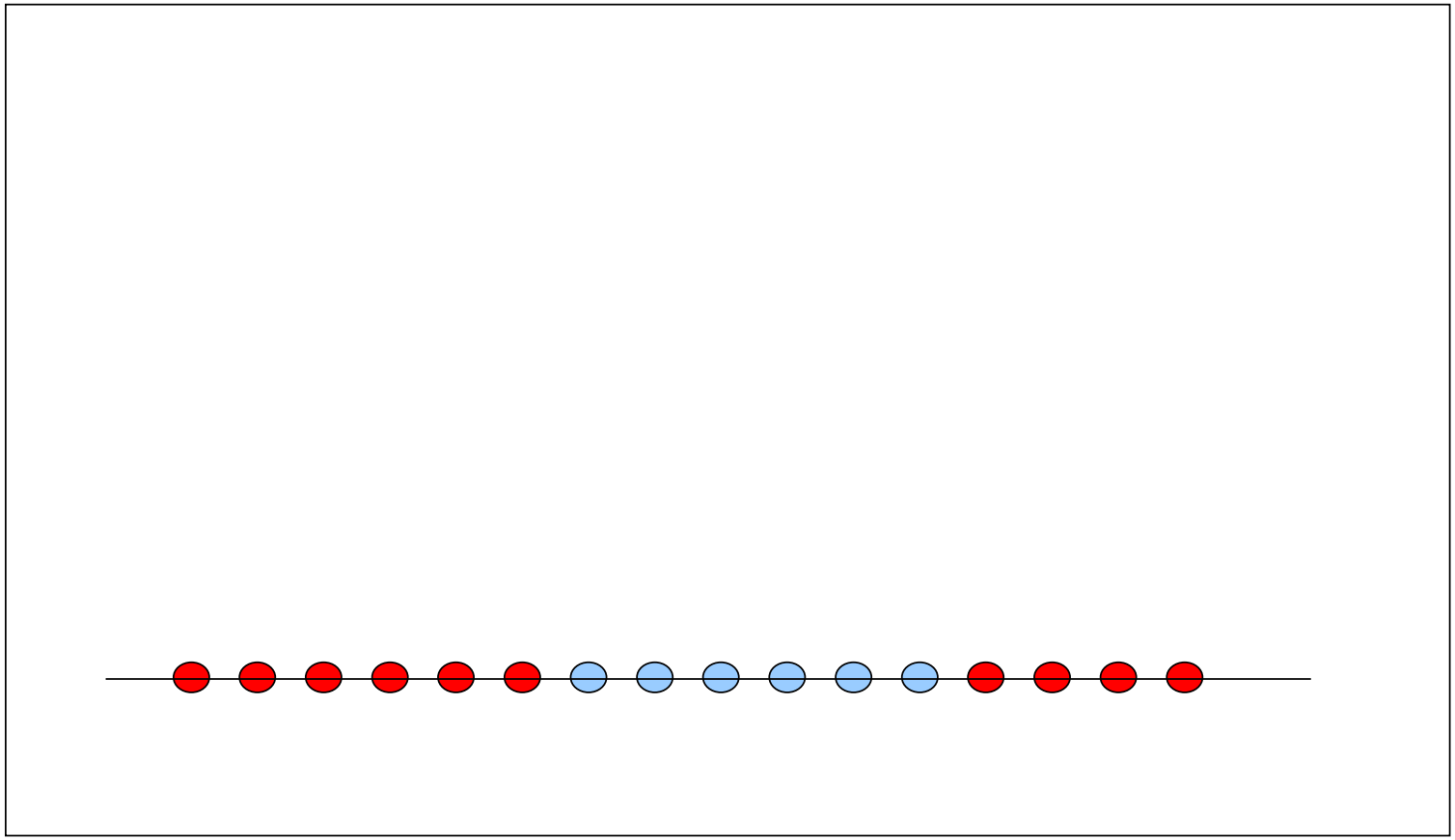


The SVM Paradigm

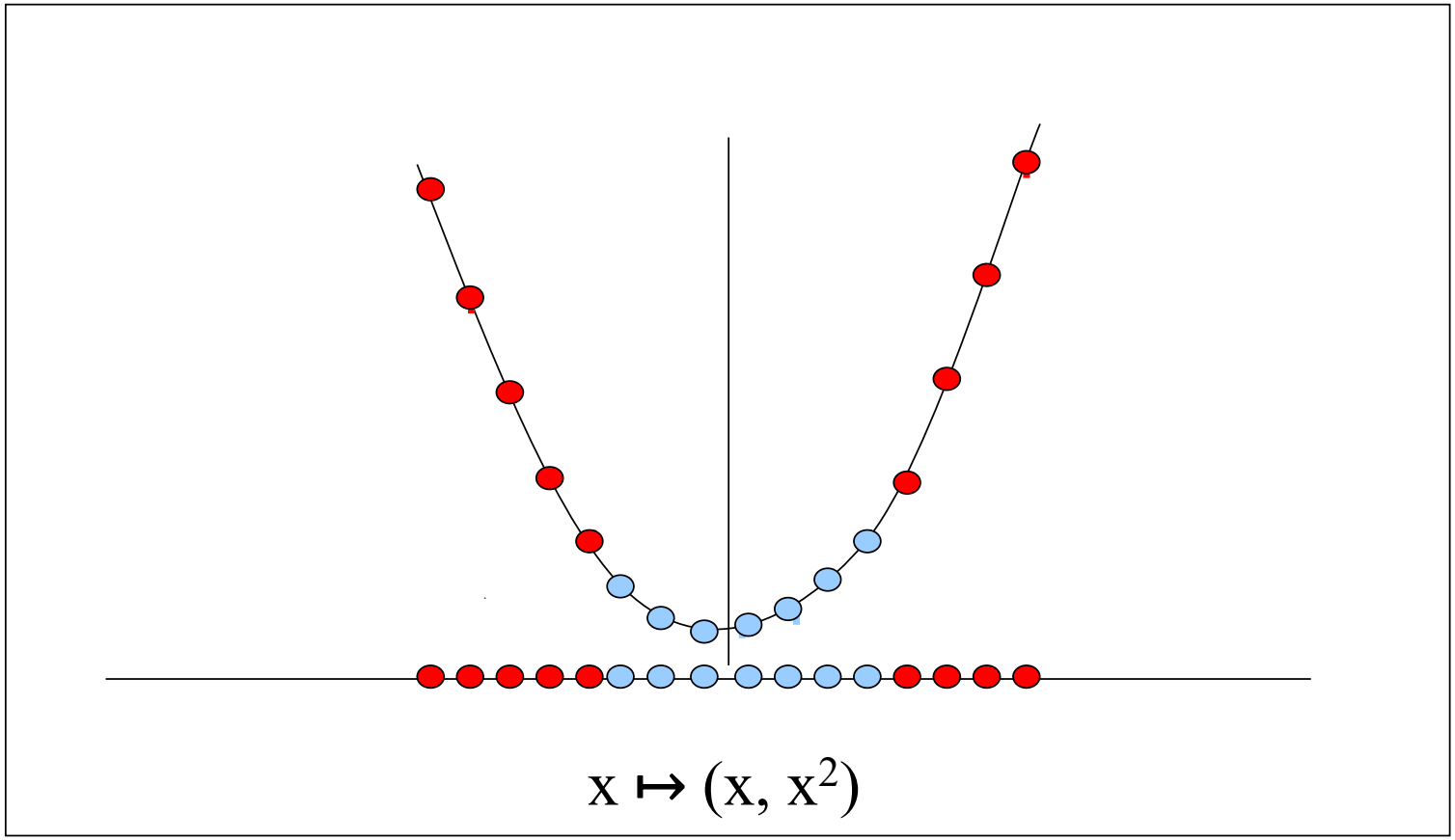
- Choose an *Embedding* of the domain X into some high dimensional Euclidean space, so that the data sample becomes (almost) linearly separable.
- Find a large-margin data-separating hyperplane in this image space, and use it for prediction.

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

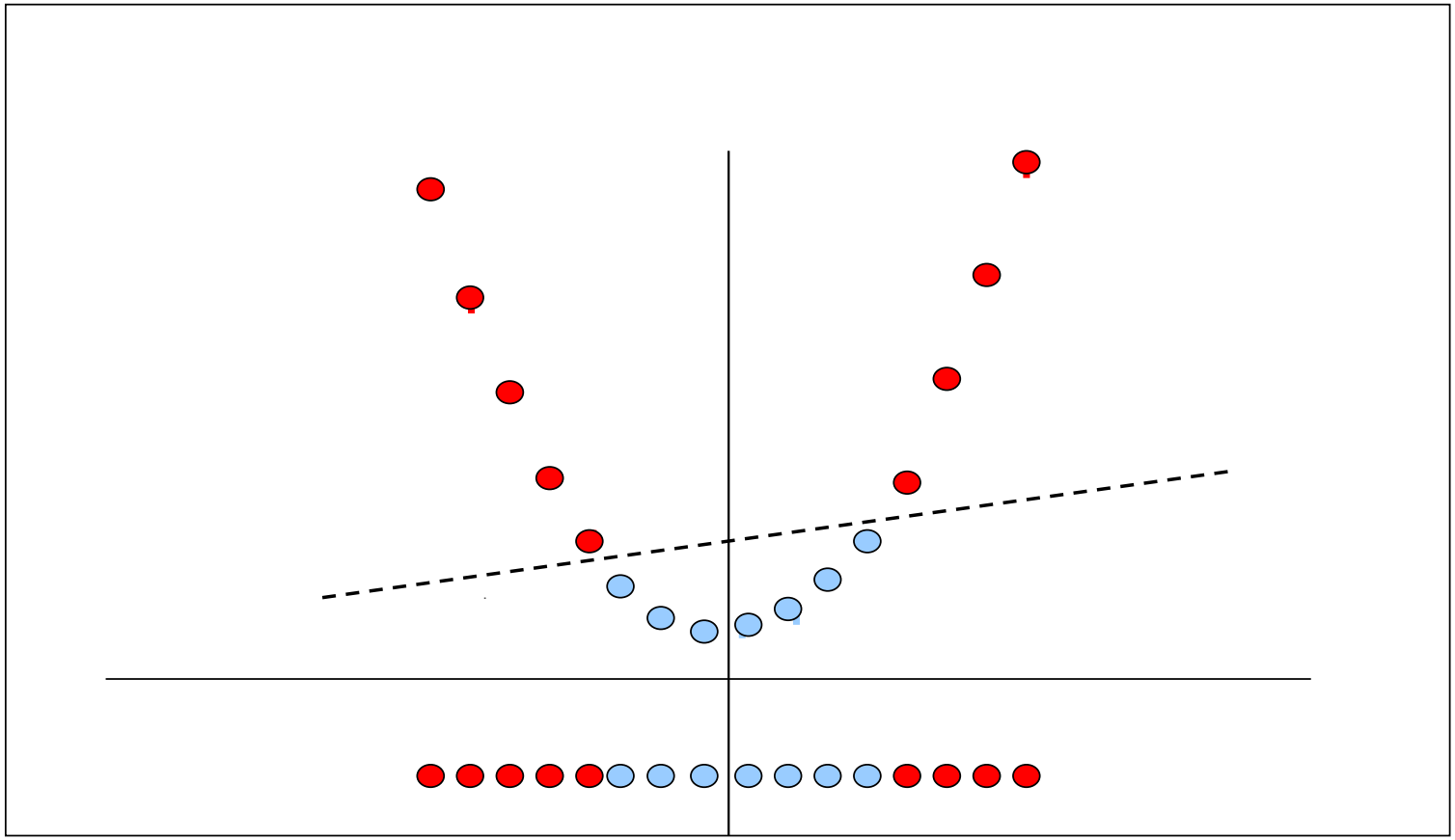
The SVM Idea: an Example



The SVM Idea: an Example



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

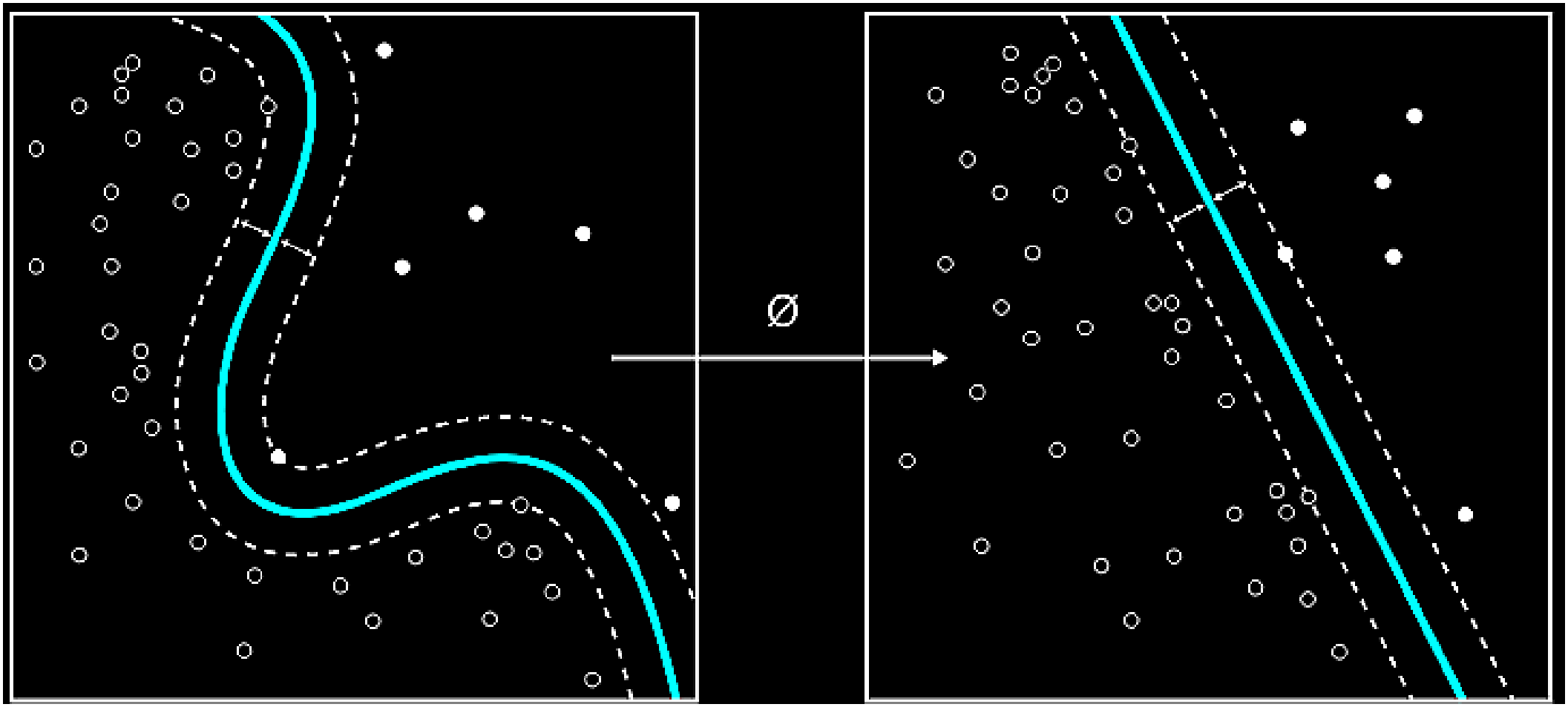
A Potential Problem: Generalization

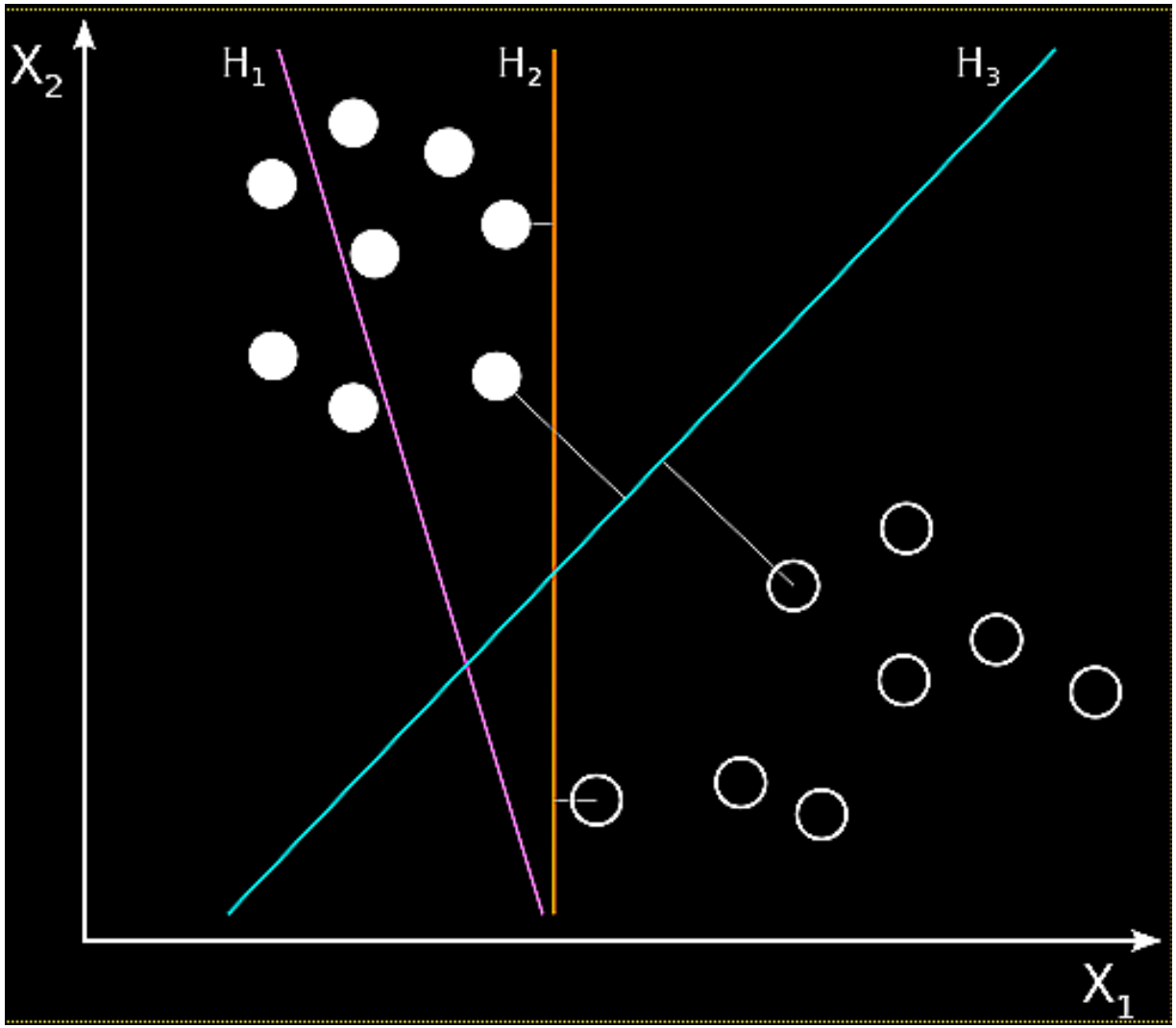
- **VC-dimension bounds**: The VC-dimension of the class of half-spaces in R^n is $n+1$.

  Can we guarantee low dimension of the embeddings range?

- **Margin bounds**: Regardless of the Euclidean dimension, generalization can bounded as a function of the margins of the hypothesis hyperplane.

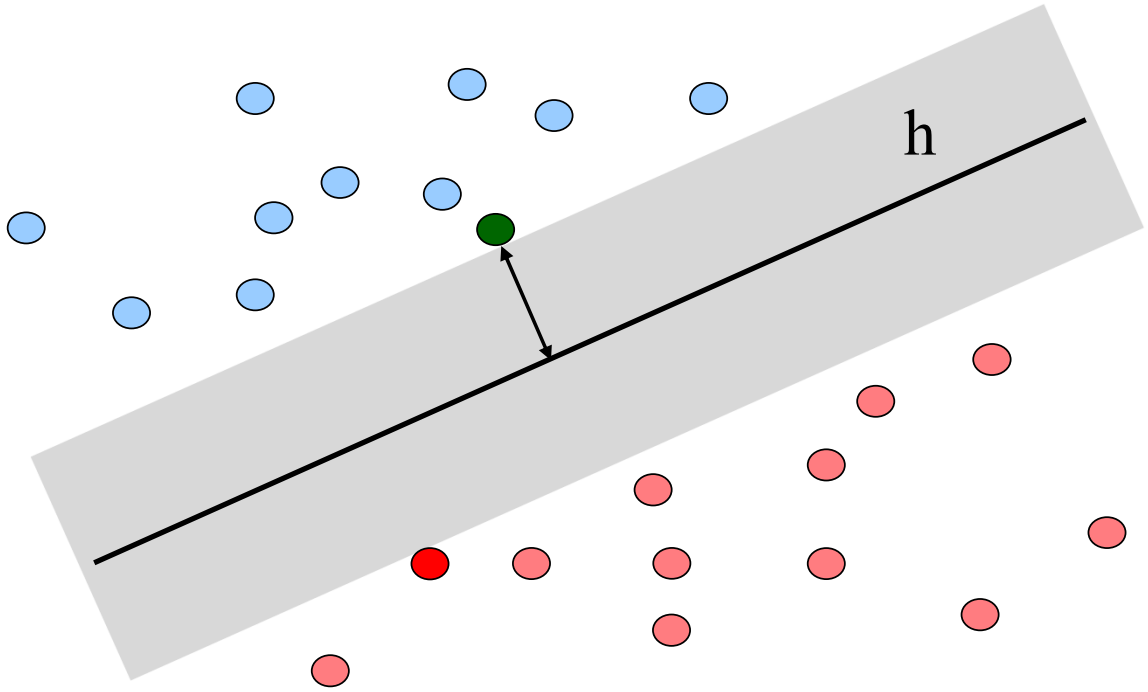
  Can one guarantee the existence of a large-margin separation?


Ø



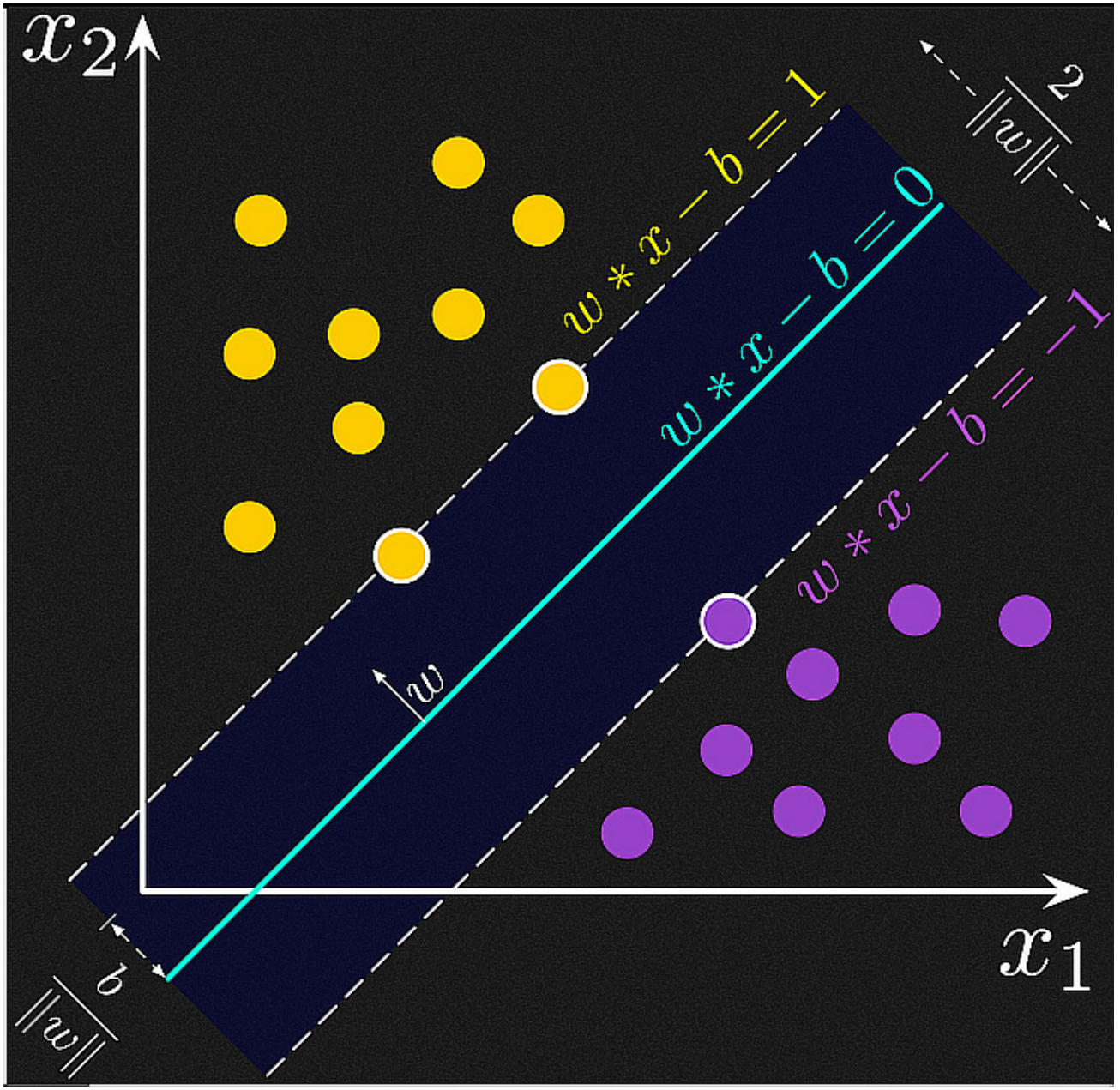
H_1 does not separate the classes.
H_2 does, but only with a small margin.
H_3 separates them with the maximal margin.

The Margins of a Sample



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

(where $\mathbf{w_n}$ is the weight vector of the hyperplane **h**)


x_2
x_1
$w * x - b = 1$
$w * x - b = 0$
$w * x - b = -1$
w
$\frac{2}{\|w\|}$
$\frac{b}{\|w\|}$

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”).

- Bayesian learning formulates learning as a form of probabilistic inference, using the observations to update a prior distribution over hypotheses.
- Maximum a posteriori (MAP) selects a single most likely hypothesis given the data.
- Maximum likelihood simply selects the hypothesis that maximizes the likelihood of the data (= MAP with a uniform prior).
- EM can find local maximum likelihood solutions for hidden variables.
- Instance-based models use the collection of data to represent a distribution.
  - Nearest-neighbor method

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