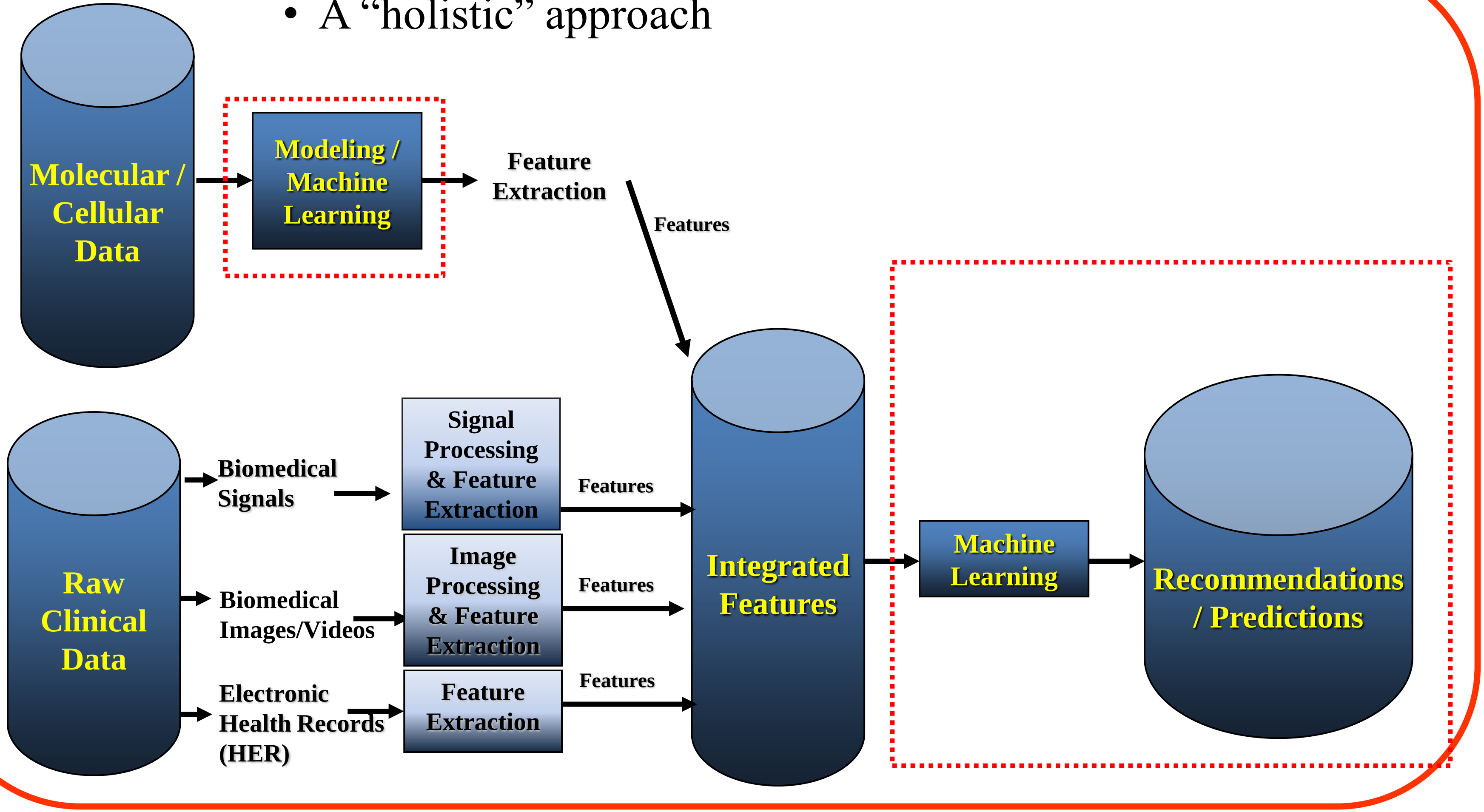
Systems Biology for Clinical Decision Support Systems

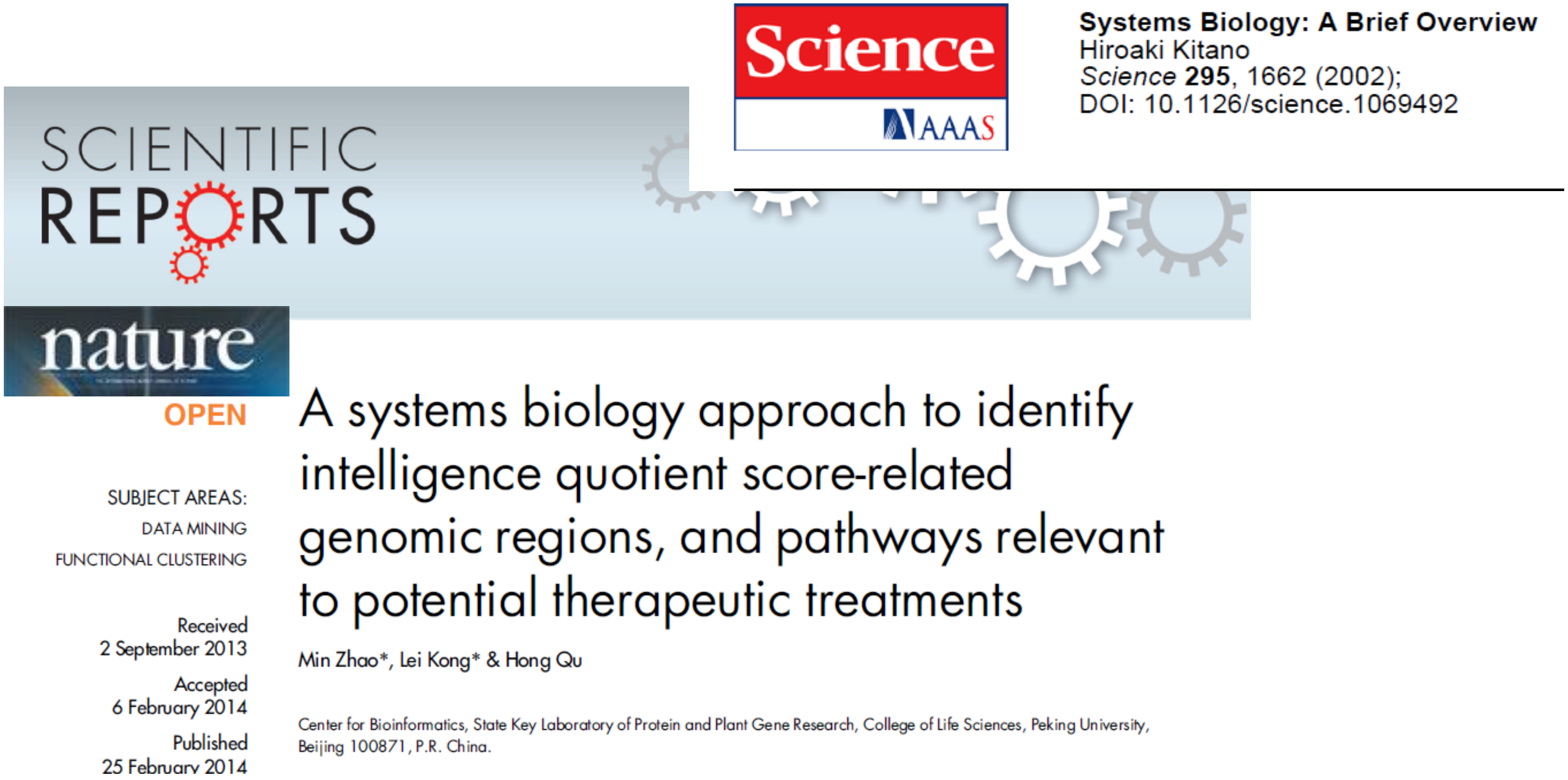
- Outlines:
  - – Machine learning; bridge between systems biology / medicine, and clinical decision support systems
  - – Brief introduction to fundamental techniques in machine learning
    - Clustering/unsupervised learning
    - Classification/supervised learning

Big Picture

- A “holistic” approach


Molecular / Cellular Data
Modeling / Machine Learning
Feature Extraction
Features
Raw Clinical Data
Biomedical Signals
Signal Processing & Feature Extraction
Features
Biomedical Images/Videos
Image Processing & Feature Extraction
Features
Electronic Health Records (HER)
Feature Extraction
Features
Integrated Features
Machine Learning
Recommendations / Predictions

Big Picture (cont'd)



Science AAAS

Systems Biology: A Brief Overview
Hiroaki Kitano
Science **295**, 1662 (2002);
DOI: 10.1126/science.1069492

SCIENTIFIC REPORTS

nature

OPEN

SUBJECT AREAS:
DATA MINING
FUNCTIONAL CLUSTERING

Received
2 September 2013

Accepted
6 February 2014

Published
25 February 2014

A systems biology approach to identify intelligence quotient score-related genomic regions, and pathways relevant to potential therapeutic treatments

Min Zhao*, Lei Kong* & Hong Qu

Center for Bioinformatics, State Key Laboratory of Protein and Plant Gene Research, College of Life Sciences, Peking University, Beijing 100871, P.R. China.

Machine Learning

- Learning / extracting patterns from data
- Applied when data are analyzed for:
  - Pattern recognition
  - Classification and clustering
  - Prediction
  - Forming recommendation
- Inspired by biology
- Unlike modeling methods that conform to the physical laws of the systems being model, machine learning relies on pattern in training data

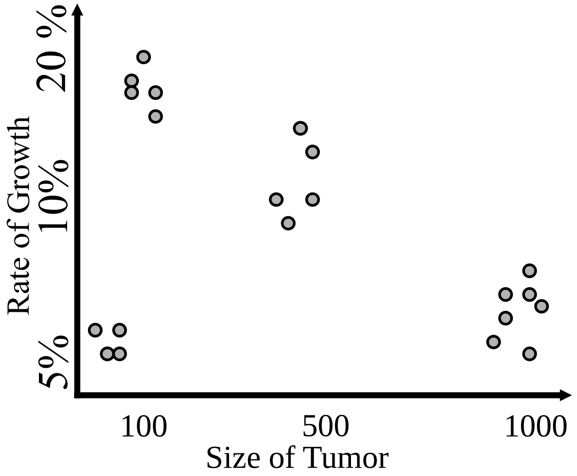
Supervised vs. Unsupervised Learning *

- Supervised Learning / Classification
  - Training data are labeled, i.e. the output class of all training data are given
  - Example: predicting if a sequence is an *a-helix* or a *β-sheet*
    - Training set: $\{Seq1 \rightarrow \alpha - helix,\ Seq2 \rightarrow \beta - sheet,\ Seq3 \rightarrow \alpha - helix,\ Seq4 \rightarrow \beta - sheet\}$
    - Classification: $Seq5 \rightarrow ?$
- Unsupervised Learning / Clustering
  - Training data are not labeled
  - Output classes must be generated during training
  - Similarity across features of training examples creates different classes

* Reinforcement learning is not discussed here

Supervised vs Unsupervised Learning (cont'd)

- *Example* (*visual insight via scattered plot*): Identification of tumor types
  - Input features: Size of Tumor & Rate of Growth
  - Training data create natural clusters
    - No specific labels are available
    - Assume all tumors are cancerous
  - From graph:



Class #1: small size of tumor with small rate of growth

Class #2: small size of tumor with large rate of growth

Class #3: medium size of tumor with medium rate of growth

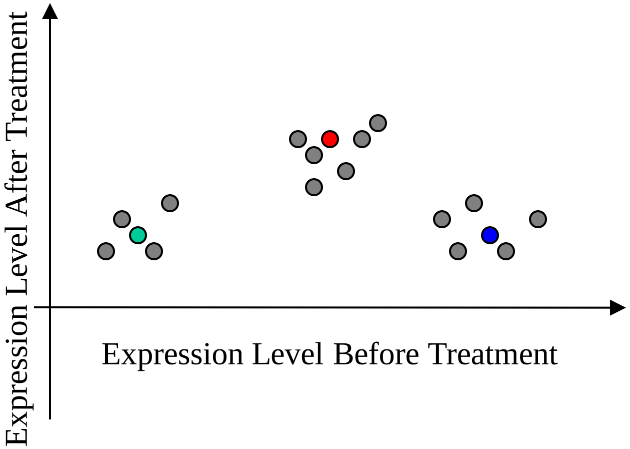
Class #4: large size of tumor with small rate of growth

  - Classification: a tumor with SOT=600 & ROG=12% is mapped to Class #3

- Classification is performed based on clustering results
  - Each clustering algorithm results to a classification technique

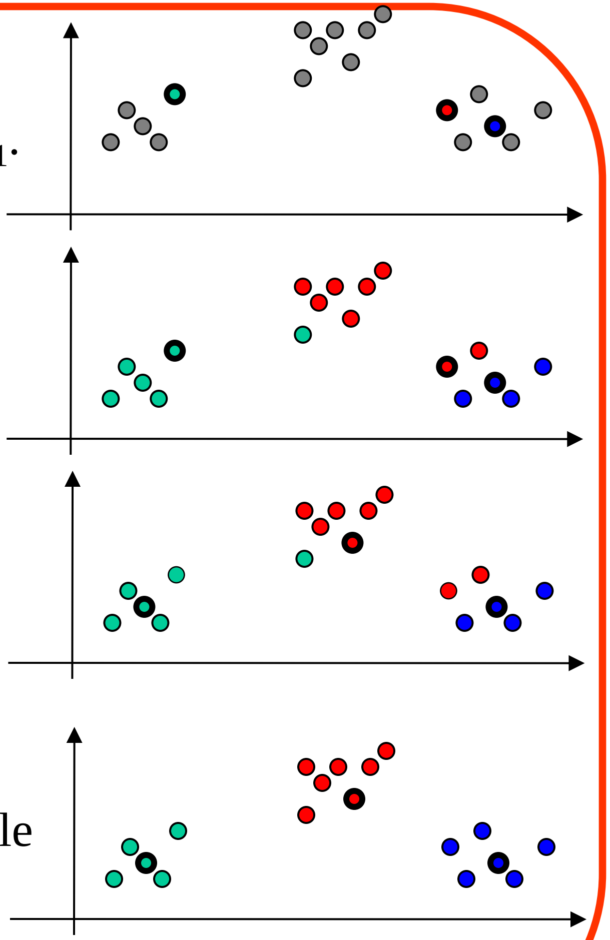
Clustering

- The simplest clustering algorithm: K-means
- K-means clustering algorithm
  - General concept of clustering:
    - Assume there are K classes
    - Represent each class with its center
      - Start with some random initial centers
    - Find the best centers for classes, i.e. optimize the centers



  - In classification phase:
    - Find the distance of a new pattern from all centers
    - Find the center whose distance from the the new pattern is minimal
    - Classify the new pattern to the class whose center has minimal distance from the pattern

Clustering (cont'd)



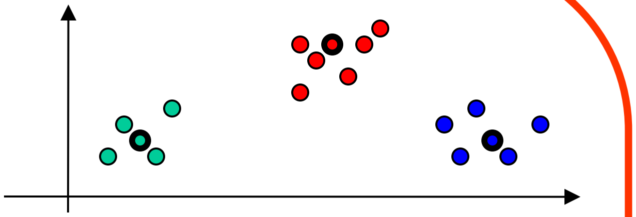
- Mathematical formulation:
  - Randomly initialize the centers $m_0, m_1, \ldots, m_{K\text{-}1}$.
  - Find the distance of all samples $x_0, x_1, \ldots, x_{n\text{-}1}$ from all centers $m_0, m_1, \ldots, m_{K\text{-}1}$, i.e. for all $0 \le i \le n\text{-}1$ and $0 \le j \le K\text{-}1$ find:

$$d(x_i, m_j) = \left\| x_i - m_j \right\|$$

  - Form classes $j = 1,2,\ldots, K$-1, by assigning each sample to the closet center, i.e. put together all examples whose distance to center j is minimal to form class j.
  - Then find the new centers by finding the sample that is the closet sample to the average of all samples in the class, i.e., new m_j = average(all examples in class j)

Clustering (cont'd)

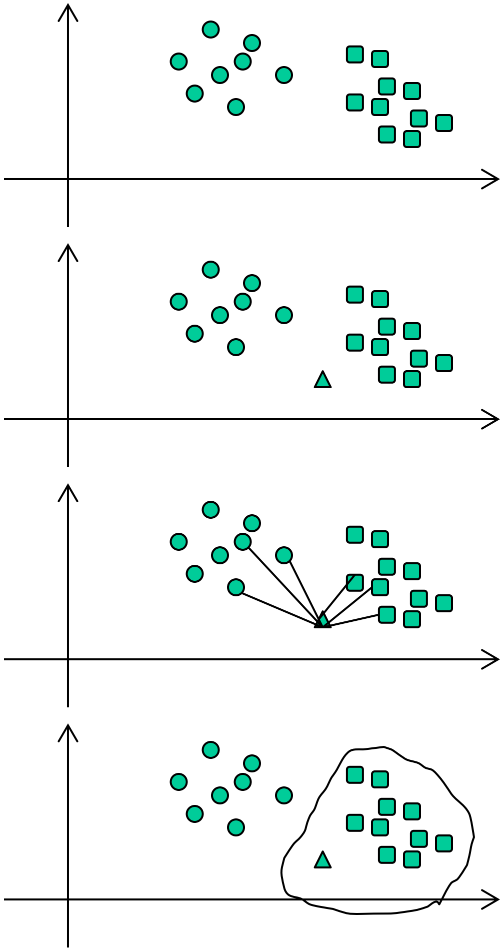
- Repeat the previous two steps, unless during the last iteration, no example has changed its class



- Advantages of K-means
  - Simple and fast
  - Works on a wide range of simple data
- Disadvantages:
  - Assumes that the number of classes is available
  - Depends on initial centers
  - Unable to cluster complicated data
- Solutions:
  - ISODATA
  - Kohonen self-organizing map

Classification with KNN

- Simplest type of classification algorithm: K Nearest Neighbors
  - General concept:
    - Calculate the distance of the new example to all known examples (and not just centers)
    - For each class, find the K known examples that belong to the class and have the minimal distance from the new pattern (among all examples belonging to this class)
    - Find the total distance of all the examples of each class and do this for all of the classes
    - The class with minimum total distance wins the new pattern



Classification with KNN (cont'd)

- The main advantage of KNN is that, just like neural networks, it does not require the knowledge of probability distribution of the classes
- KNN is much less complex and computationally intensive than connectionist methods such as neural networks and support vector machines
- More computationally-intensive classifiers exist that do require the exact knowledge or at least an accurate estimation of the distributions

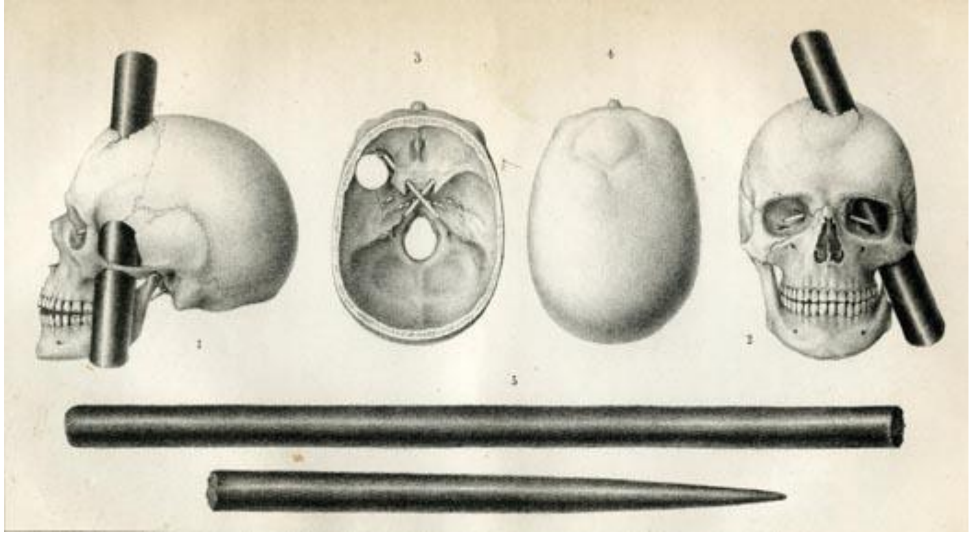
Introduction to Connectionist Models

- Why connectionist models?
  - Algorithms developed over centuries do not fit the complexity of real world problem
  - The human brain: most sophisticated computer suitable for solving extremely complex problems
- Historical knowledge on human brain
  - Phineas Gage's Story
    - In a rail accident, a metal bar was shot through the head of Mr. Phineas P. Gage at Cavendish, Vermont, Sept 14, 1848
      - Iron bar was 3 feet 7 inches long and weighed 13 1/2 pounds. It was 1 1/4 inches in diameter at one end

http://www.boston.com/news/local/massachusetts/articles/2009/07/22/newly_discovered_image_offers_fresh_insights_about_1848_medical_miracle/ (From the collection of Jack and Beverly Wilgus)

Introduction to Connectionist Models (cont'd)

- He survived the accident!
  - Originally he seemed to have [almost] fully recovered



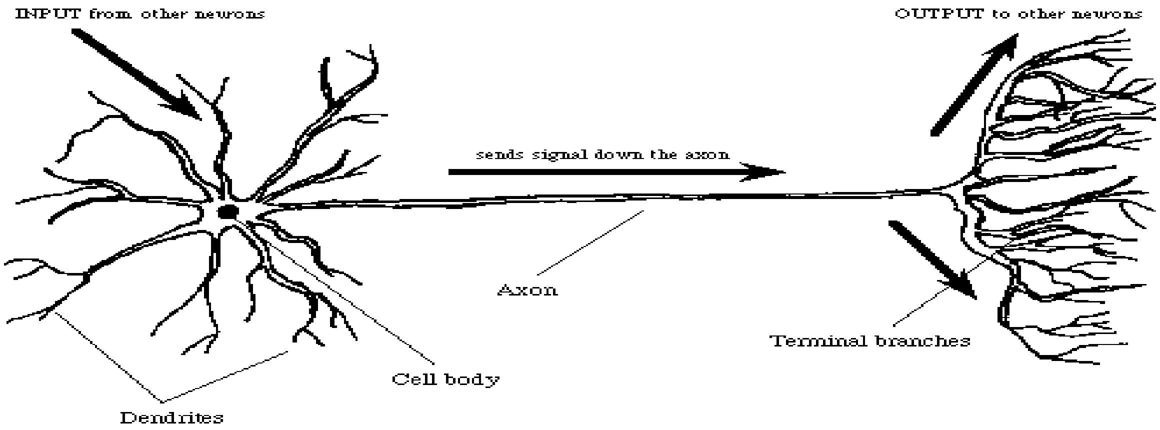
Bigelow, Henry J. "Dr. Harlow's case of recovery from the passage of an iron bar through the head." American Journal of the Medical Sciences, n.s. v.20 (July 1850): 13-22. Available at: https://cms.www.countway.harvard.edu/wp/?tag=phineas-gage

- After a few weeks, Phineas exhibited profound personality changes

– This is the first time, researchers have a clear evidence that the brain is not a continuum of cell mass and rather each region has relatively independent task

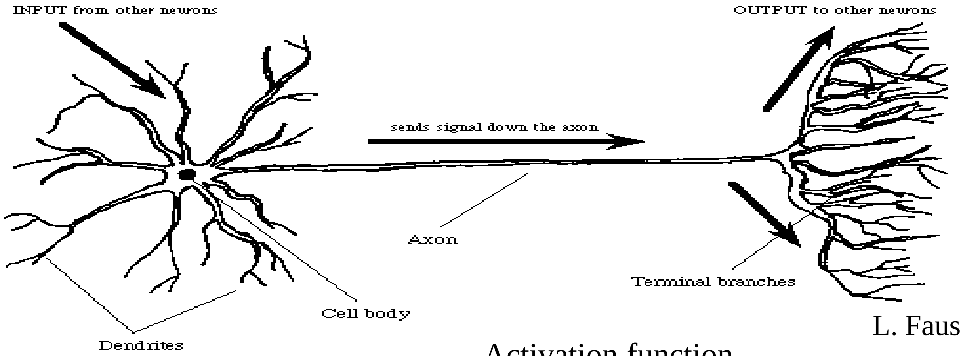
Introduction to Connectionist Models (Continued)

- learning and generalization through examples
- simple building block: neuron
  - Dendrites: collecting signals from other neurons
  - Soma (cell body): spatial summation and processing
  - Axon: transmitting signals to dendrites of other cells



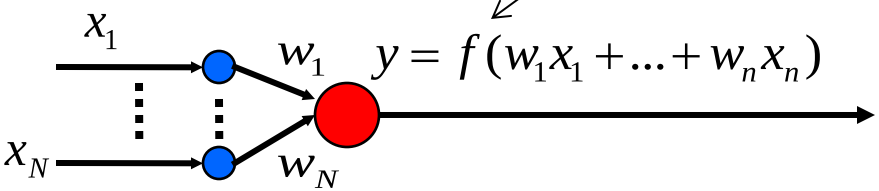
From biological to artificial neural nets

- Reference book: "Fundamentals of Neural Networks; Architecture, Algorithms, And Applications", by: L. Fausett.
  - Highly recommended for this course to learn practical applications of neural nets
- Biological vs. artificial neurons
  - From biological neuron to schematic structure of artificial neuron
    - biological:
      - Inputs
      - Summation of inputs
      - Processing unit
      - Output
    - artificial:



L. Fausett

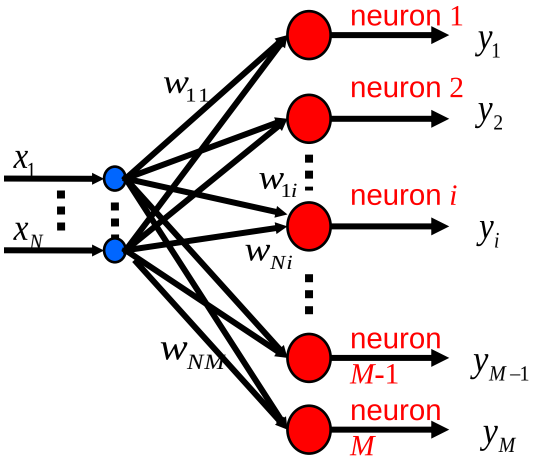
Activation function



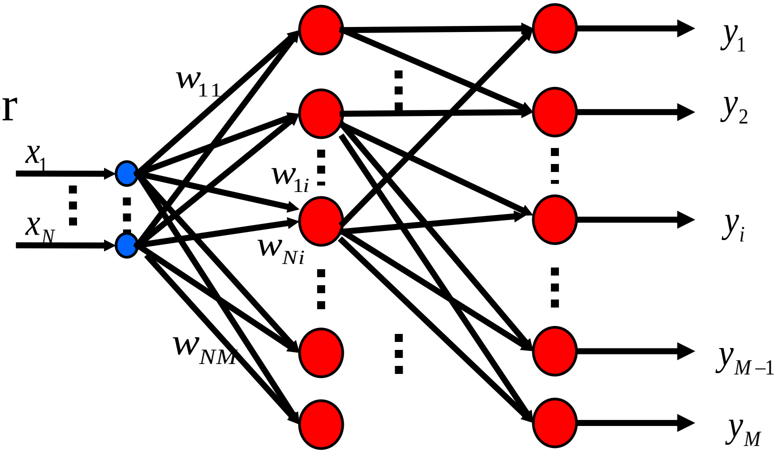
$$y = f(w_1 x_1 + \ldots + w_n x_n)$$

From biological to artificial neural net (continued)

- Artificial neural nets:
  - Single-layer:



  - Multi-layer



A Supervised NN: Backpropagation Neural Network

- Training: using a set of known examples to estimate best values of weights (e.g. backpropagation algorithm)
- Architecture:
  - Number of layers and number of neurons in each layer: variable
- It was proved that three layers (one input, one hidden and one output) with sufficient number of neurons in the hidden layer, and learn practically any function
  - As such there is no need to try more than three layers
  - A rule of thumb suggests that the number of hidden layers must be less than 1/10 of the number of training examples
- Majority of algorithms designed for neural networks are gradient based

Backpropagation Neural Networks (cont'd)

- Another issue that all types of machine learning methods, including neural networks can suffer from is overfitting
  - They can be "over-trained" with the training data and perform poorly on the data they have not seen before
  - A common approach to address this problem is to ensure that the size of the parameter search space (weight space in case of neural nets) is reduced
  - For instance, to address this issue neural nets, often a term containing the following value to the cost function to be minimized:

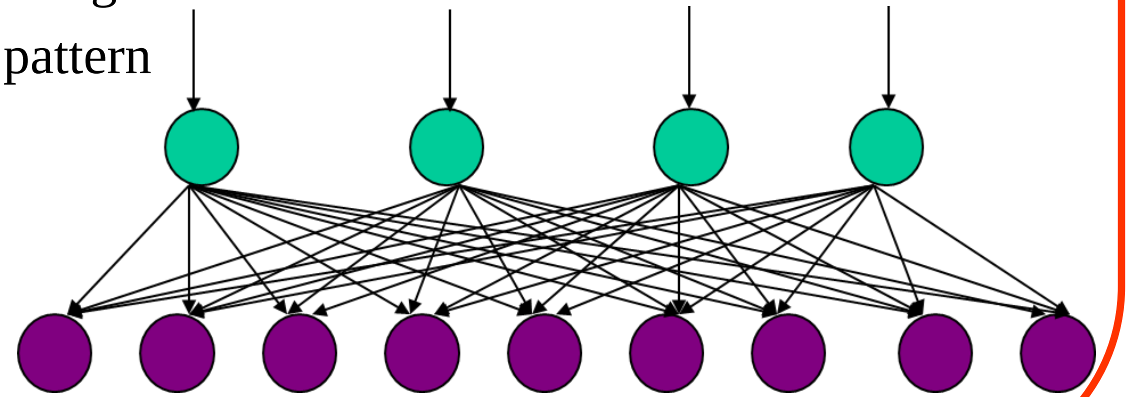
$$\sum \|w\|^2$$

Neural Networks for Clustering

- Artificial competitive neural networks:
  - Each artificial neuron (or group of neurons) stores a pattern
  - Networks are trained such that neurons in the same neighborhood store similar patterns
  - During classification, neurons compete to relate to the new pattern
  - The neurons with highest similarity to the new pattern are the winners
  - The most popular type of unsupervised neural network is Kohonen Self-Organizing Map (K-map)
  - K-maps are heavily used in systems biology and bioinformatics, e.g. clustering of gene expression data

Kohonen Self-Organizing Maps

- Kohonen Self-Organizing Map (Kohonen SOM): most popular unsupervised learning machine
- Architecture (one-dimensional): (Fausett)
  - Input neurons map *n*-dimensional patterns to output (competitive) neurons with *p*-dimensional output patterns
  - Output neurons with similar patterns are arranged to be neighbors
  - Some output neurons might store more than one pattern



Kohonen Self-Organizing Maps (continued)

- *Example: object classification using one-dimensional array*

Inputs

pencil

airplane

chair

book

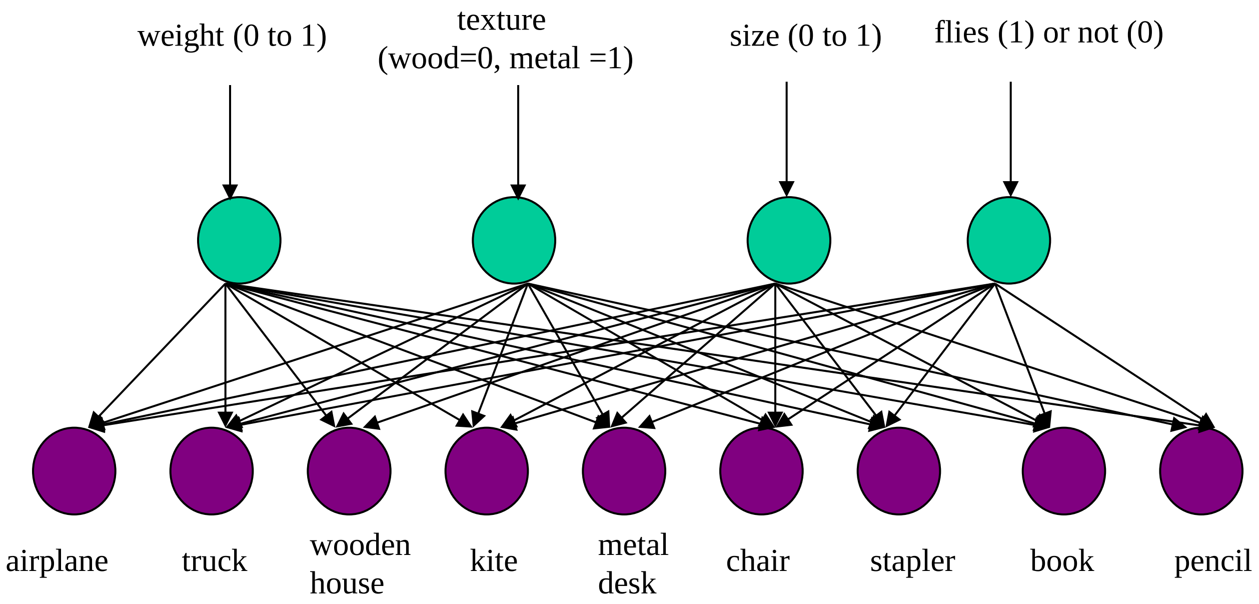
wooden house

stapler

metal desk

truck

kite



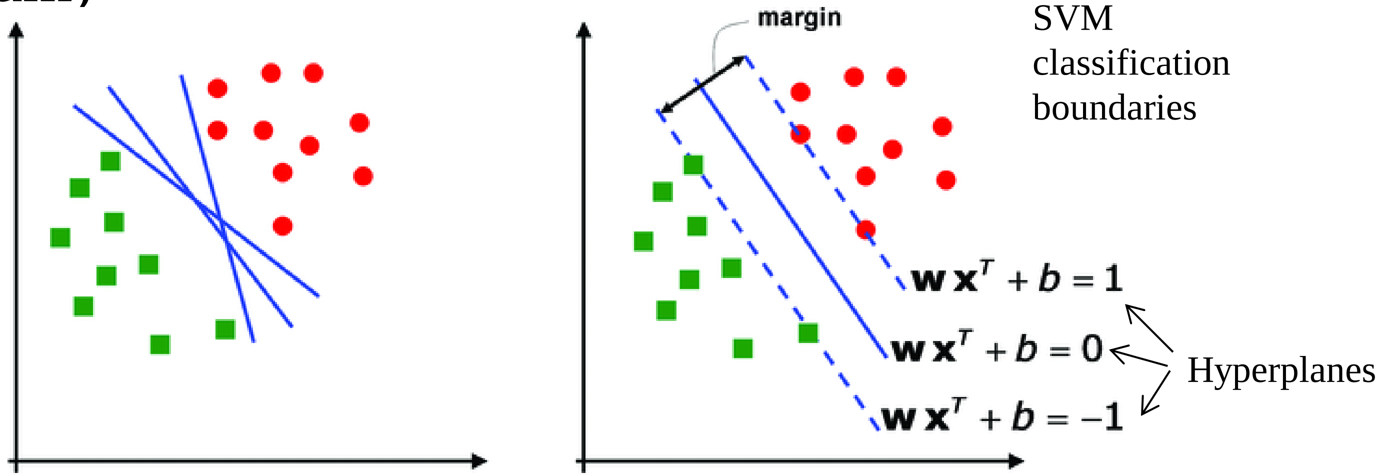
– *New pattern: hovercraft*
  - *is mapped to the nationhood of truck or airplane*

– *Disadvantage: one-dimensional array positions some objects with some similarities far from each other*

Support Vector Machines (SVMs)

- Idea: design classifier such that the margin between boundaries and instances of the classes is optimal (maximum)

Different options of classification boundaries


margin
$\mathbf{w}\,\mathbf{x}^T + b = 1$
$\mathbf{w}\,\mathbf{x}^T + b = 0$
$\mathbf{w}\,\mathbf{x}^T + b = -1$

SVM classification boundaries

Hyperplanes

Tarca AL, Carey VJ, Chen X-w, Romero R, Drăghici S (2007) Machine Learning and Its Applications to Biology. PLoS Comput Biol 3(6): e116. doi:10.1371/journal.pcbi.0030116

  - Having a good margin makes classifier less susceptible to uncertainty and more likely to learn/generalize the data.

SVMs (cont'd)

- Formulation (linear kernel):

$$(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_N, y_N)$$

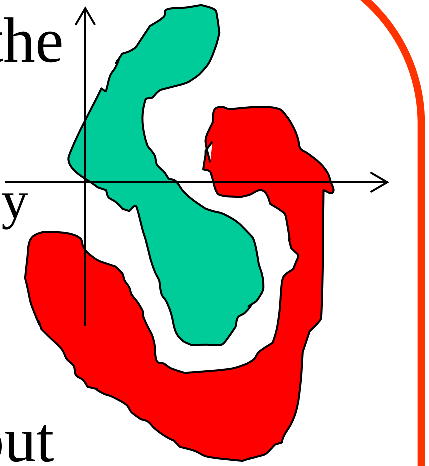
$$y_i \in \{-1,1\}$$

  - Objective: to find **w** and *b* such that the hyperplane $\mathbf{wx}^T+b=0$ that:
    1. Separates all/most of samples in the two classes
    2. Maximizes the margin between the classes labeled -1 and 1.
  - Without discussing the details of the solutions to this problem:
    - This optimization problem can be formulated in at least two different methods and solved rather effectively
    - Decision function:

$$f(\mathbf{x}) = sign(\mathbf{wx}^{\mathbf{T}} + b) = sign\left(\sum_i \alpha_i y_i (\mathbf{x_i x^T}) + b\right)$$

    - The problem can be easily extended into multi-class classification scenarios

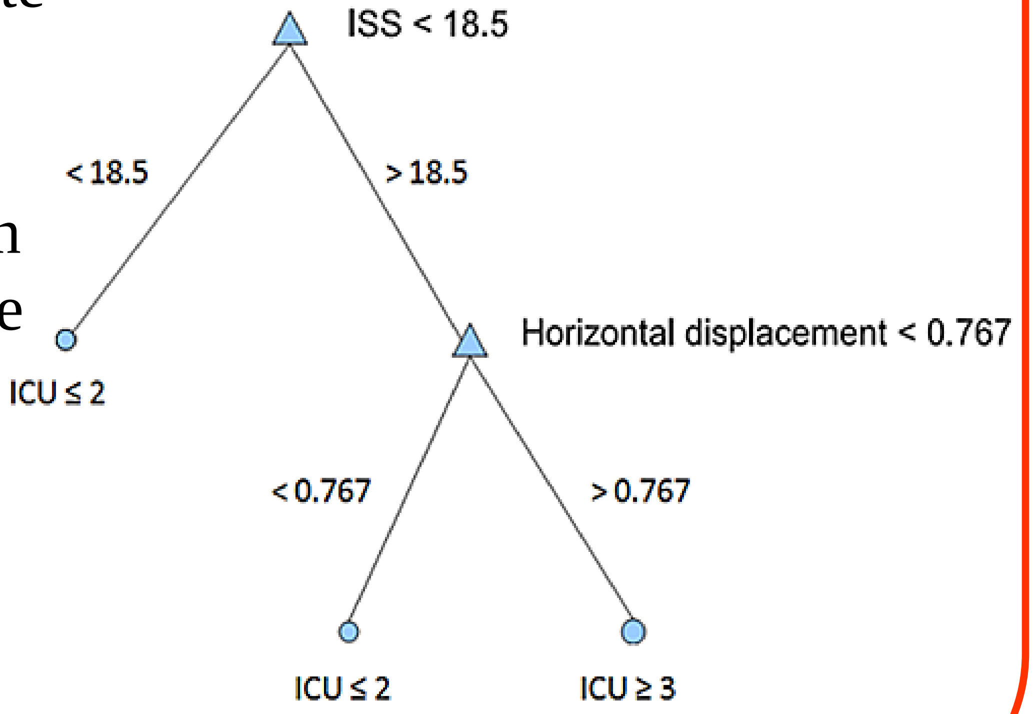
SVMs (cont'd)



- What if the inputs are not linearly separable in the original feature/input space?
  - Inputs in the two regions shown here are separable, by by a good margin, but lines/hyperplanes are not the best elements to separate them
- In such cases, we use a kernel that maps the input space into space that might create better separation
  - These kernels are often nonlinear functions with symmetric properties and with [much] higher dimension that the original space
  - Popular kernels: radial basis functions (RBF) and Gaussian

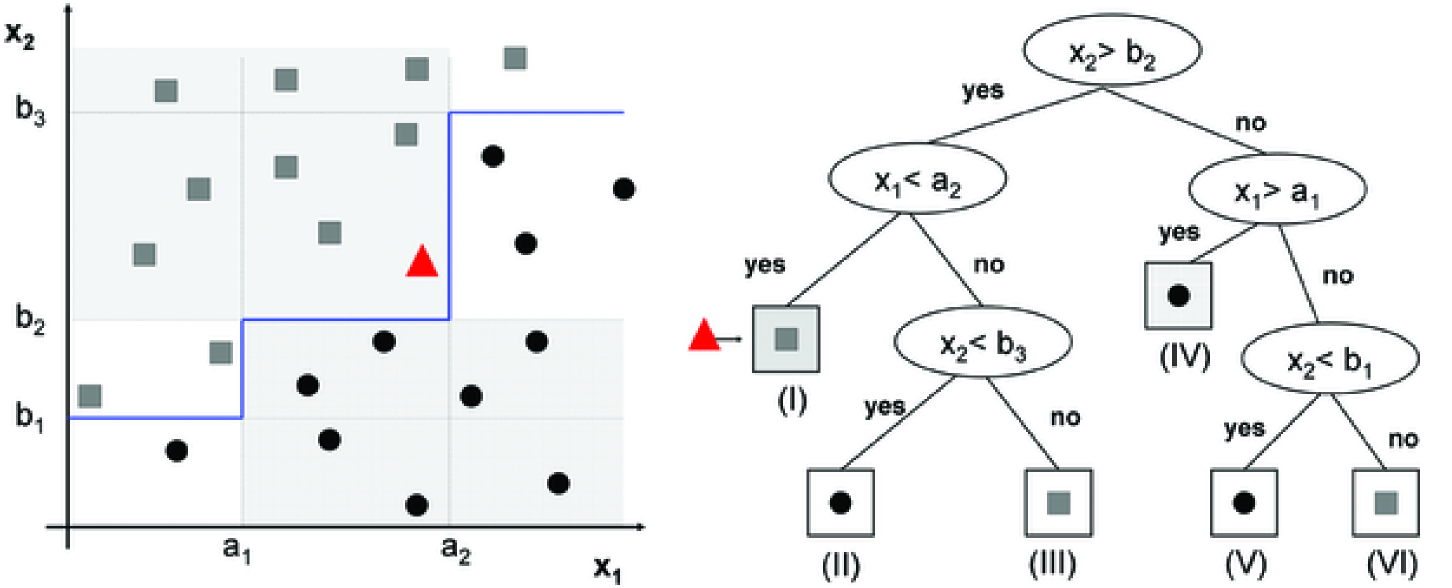
Decision/Regression Trees

- Decision and regression trees hierarchically create rules that classify / analyze data
- They are very popular in medical applications due to their transparency



Decision/Regression Trees (cont'd)

- A visual example:



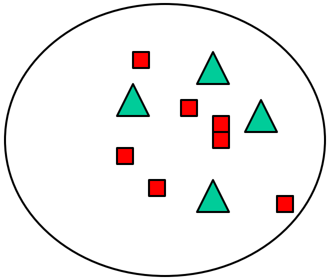
Tarca AL, Carey VJ, Chen X-w, Romero R, et al. (2007) Machine Learning and Its Applications to Biology. PLoS Comput Biol 3(6): e116. doi:10.1371/journal.pcbi.0030116
http://www.ploscompbiol.org/article/info:doi/10.1371/journal.pcbi.0030116

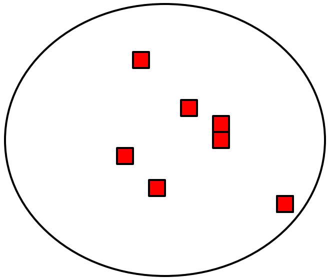
Decision/Regression Trees (cont'd)

- What measure can tell us:
  - What variable to branch on next?
  - What value for this variable is the threshold for branching
- Although different measures are used for this purpose, resulting to variety of decision trees, almost all of them are based on entropy, which is a measure of uncertainty/impurity

$$E = -\sum_i p_i \log p_i$$

High entropy, high impurity





Low entropy, low impurity

- Entropy quantifies lack of "purity" among the examples left in a given node

Decision/Regression Trees (cont'd)

- One typical measure used for splitting nodes; information gain:

  Information Gain = Entropy(parent) – [Average Entropy(children)]

- Unlike connectionist methods, dealing with categorical variables in the nature of decision trees
- There are many tree-based methods such as ID3, CART (Classification and Regression Tree) and C4.5
- Decision/regression trees can overfit the data, in particular when depth of the tree becomes too large
  - Pruning is used to partially address this issue; trees are first expanded to large depth and then pruned/collapsed according to some quality measure

Random Forest

- Another solution to address the issue of overfitting when dealing with trees using bootstrapping / bagging
- Data are randomly split into subsets
- Random subsets of input variables are formed
  - This allows a level of variable/feature selection not available in many other machine learning models
- Using subsets of data and subsets of input variables, trees are forms
- Performance of these trees are tested using the data not seen by those trees
  - This is a level of validation not available in many other tree-based methods

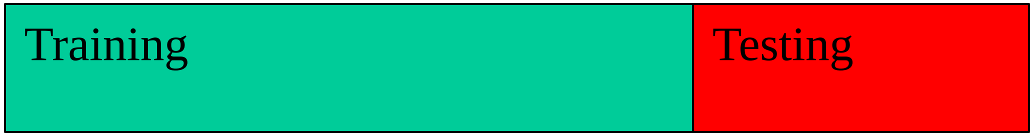
Random Forest (cont'd)

- Finally, the output of random forest to a new input will be the average of the output of all trees to that input
- Random forest is known to have good accuracy without serious overfitting
- Also, it is one of the fastest machine learning methods given its desirable performance
- Furthermore, since a level of feature selection is conducted by Random Forest, it may be applied to some datasets large number of input features without feature selection
- It was shown, at least in some microarray based studies, that on average Random Forest might produce the best results for bioinformatics applications

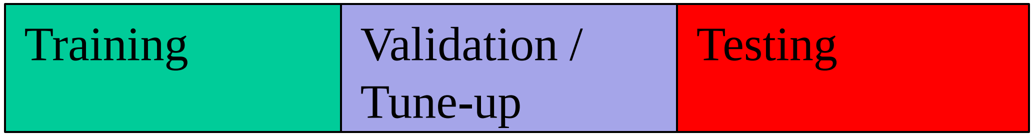
Testing and Validation of Machine Learning Models

- There are two main approaches:
  1. Training-testing: Randomly divide data into training and testing sets, and test the performance on the testing data.

One approach

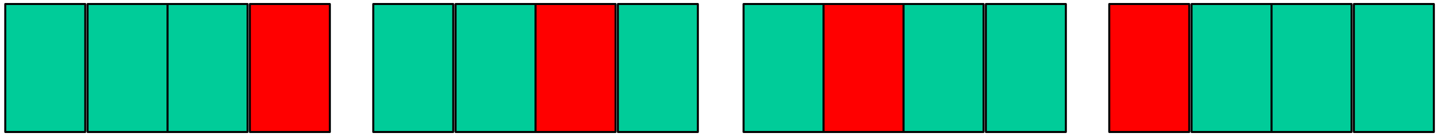


Another approach



  2. *n*-fold cross validation: Randomly divide data into *n* folds (often *n*=10) and in a “round-robin” like approach, train the model using *n*-1 folds and test on the *n*th. The average the performance over all n trained models.

n=4



Testing and Validation of Machine Learning Models (cont'd)

- An extreme but special case of *n*-fold cross validation is called "leave-one-out" in which for each of the round-robin attempts, all but one example is training and that example is used for testing
- If the number of examples is small, leave-one-out is a reasonable option; if not, *n*-fold cross validation is preferred because it is less likely to overfit the data

- Performance measures
  - Regardless of specific approach used for testing and validation, some popular measures are used for evaluating the goodness of classification
  - These measures are evaluated based on "confusion matrix"
  - When diagonal elements are large, and others are small, life is beautiful!

	Class Positive	Class Negative
Predicted as Positive	True Positive	False Positive
Predicted as Negative	False Negative	True Negative

Testing and Validation of Machine Learning Models (cont'd)

– Some important measures based on confusion matrix:

$$Accuracy = \frac{True\,Positive + True\,Negative}{True\,Positive + True\,Negative + False\,Positive + False\,Negative} = \frac{True\,Predictions}{All\ Predictions}$$

$$Sensitivity = Recall = True\,Positive\ \ Rate = \frac{True\,Positive}{True\,Positive + False\,Negative} = \frac{True\,Positive}{Positive\ \ Class}$$

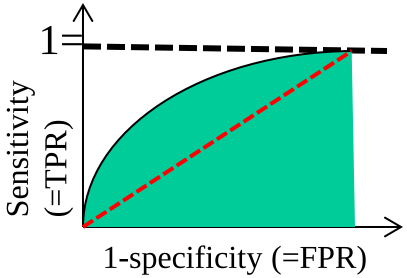
$$Specificit\,y = \frac{True\,Negative}{True\,Negative + False\,Positive} = \frac{True\,Negative}{Negative\ \ Class}$$

$$Precision = Positive\ Predictive\ \ Value = \frac{True\,Positive}{True\,Positive + False\,Positive} = \frac{True\,Positive}{Predicted\ \ as\ Positive}$$

$$False\ \ Positive\ \ Rate = 1 - Specificit\,y = \frac{False\,Positive}{Faslse\,Positive + True\,Negative} = \frac{False\,Positive}{Negative\ \ Class}$$

Testing and Validation of Machine Learning Models (cont'd)

- For about the same accuracy, machine learning classes can be trained to have a different balance between true sensitivity and specificity; or equivalently the balance between false positive rate (FPR) and true positive rate (TPR)
  - Remember: FPR=1-Specificity & TPR = Sensitivity
- Receiver operating characteristic (ROC) curve is measure to show the trade-off between FPR and TPR
  - Different threshold values or weighting on cost functions are used to generate points in ROC
  - A good ROC is the one that rises early so that high sensitivity and specificity are achieved at the same time



- Area Under Curve (AUC) of ROC
  - A measure of how fast ROC rises and how soon a good balance point between specificity and sensitivity is achieved

Feature Reduction / Selection

- Fewer features reduces the computational complexity and likelihood of overfitting
- There are three main approaches to reduce the number of features:
  1. Using domain knowledge to select some of the most relevant features
  2. Computationally mixing all features to generate fewer combined features (such as PCA)
  3. Computationally selecting a smaller set using a criterion / ranking system
     - These methods in turn divide into filter-based and wrapper-based methods
     - Typical measures/criteria used for this process include information gain

Summary

- Machine learning applied on molecular / cellular data, along with other types of data, would allow system biologic study of the system
- Clustering (unsupervised learning) and classification (supervised learning) are the two main tasks in machine learning
- Multilayer perceptron neural networks, support vector machines and regression tress are the main techniques used for classification
  - Many techniques are based on these methods, e.g. Random Forest is a method based on collective decision of a number of tress

Summary

- Machine learning models are susceptible to overfitting issue
- Methods such as n-fold cross validation are used for testing and validation of machine learning models
- There is often need to reduce dimensionality of data
- A number of measures such as sensitivity and specificity are used to assess the quality of a model