Using Machine Learning to Optimize Storage Systems

Dr. Kiran Gunnam

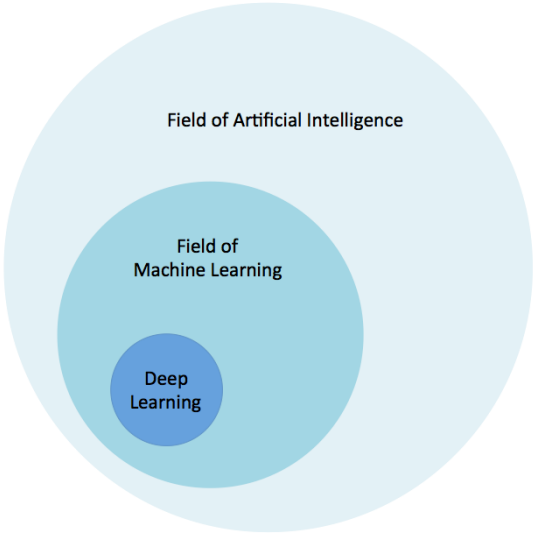
Outline

1. Overview
2. Building Flash Models using Logistic Regression.
3. Storage Object classification
4. Storage Allocation recommendation engines for elastic storage systems.
5. Building an end-to-end model

Overview

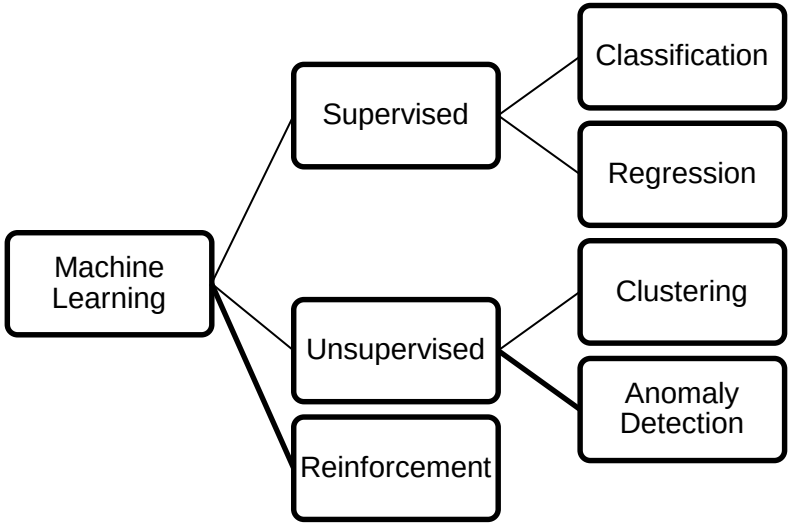
What is Machine Learning?

- **AI**: capability of a machine to imitate intelligent human behavior. (Artificial general intelligence (AGI)-not yet there, Artificial narrow intelligence (ANI)- specific tasks such as playing GO, playing chess, self-driving cars)
- **Machine learning**: subset of AI. Algorithms to analyze data to make and improve the decision making process.
- **Formal Definition**: A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E.
- **Deep Learning** : Subset of Machine Learning. Analyze data using multiple layers of abstraction. Examples are Convolutional Neural Networks, Recurrent Neural Networks etc.



Machine learning techniques

- Supervised Learning involved constructing trained models with known input data and output data.
- Unsupervised Learning finds patterns within data about which nothing is known prior.
- Reinforcement learning is a reward based learning where the algorithm learns to make specific decisions based on broader guideline.



Supervised vs Unsupervised

SUPERVISED

- Evidence Based Prediction
- Takes a known set of input data and responses to predict responses to new input data.
- Classification - discrete responses
- Regression - continuous responses

UNSUPERVISED

- Draw inferences from data sets consisting of input data without strictly labelled responses.
- Clustering- analyze data to find patterns and groupings

Algorithms

Supervised		Unsupervised	Reinforcement
Classification	Regression	Clustering	
Logistic Regression k Nearest Neighbor Neural Networks Support Vector Machines Bayesian Network Decision Tree	Linear Regression Logistic Regression Support Vector Regression	k Means Mixture Models Expectation–Maximization (EM) algorithm Belief Propagation (pre-processing techniques: Principal component analysis, Independent component analysis, Non-negative matrix factorization, Singular value decomposition)	Value-based (Q-learning, real-time dynamic programming) Policy-based (Actor-Critic Algorithm) Model-based (applied for games such as Chess and GO where rules of game are known and model is perfect)

Flash Model and Parameter Optimization

Flash Parameter Optimization

- Device Manufacturers tune the flash parameters to achieve reasonable specification for broad
- However it is possible to optimize the flash parameters based on the operating state of the flash memory.
- Advantage of flash parameter optimization: Improve the flash endurance.
- How ever the challenge is that many registers has dependencies. It is critical to determine a set of register settings that satisfy all the criterion.

Logistic Regression to build Flash Models

- Why use logistic regression?
- Estimation by maximum likelihood
- Interpreting coefficients
- Hypothesis testing
- Evaluating the performance of the model

Model Building

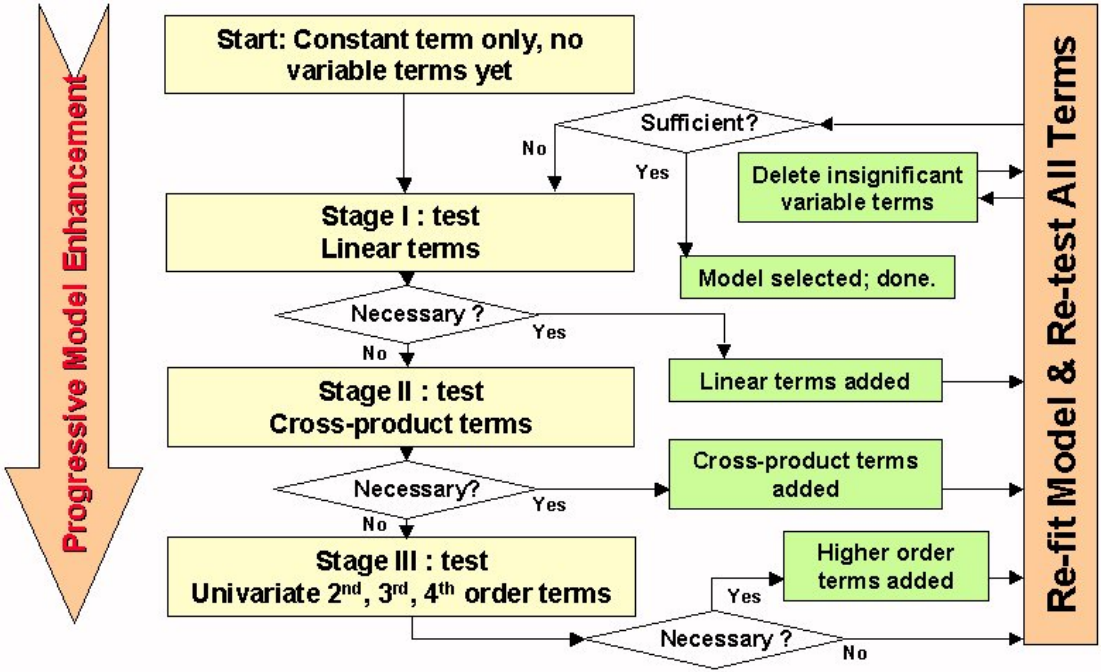
The main approaches are:

•**Forward selection**, which involves starting with no variables in the model, testing the addition of each variable using a chosen model fit criterion, adding the variable (if any) whose inclusion gives the most statistically significant improvement of the fit, and repeating this process until none improves the model to a statistically significant extent.

•**Backward elimination**, which involves starting with all candidate variables, testing the deletion of each variable using a chosen model fit criterion, deleting the variable (if any) whose loss gives the most statistically insignificant deterioration of the model fit, and repeating this process until no further variables can be deleted without a statistically significant loss of fit.

•**Bidirectional elimination**, a combination of the above, testing at each step for variables to be included or excluded.

Model Building



Why use logistic regression for Flash Models?

- There are many important research topics for which the dependent variable is "limited."
- There are several independent variables and combinations of independent variables (flash device registers) affect the dependent variable (flash page error)
- Binary logistic regression is a type of regression analysis where the dependent variable is a dummy variable: coded 0 (error) or 1(no error)

The Linear Probability Model

In the OLS (ordinary least squares (OLS) or linear least squares regression):

$Y = \gamma + \varphi X + e$; where Y = (0, 1)

- The error terms are heteroskedastic
- e is not normally distributed because Y takes on only two values
- The predicted probabilities can be greater than 1 or less than 0

More:

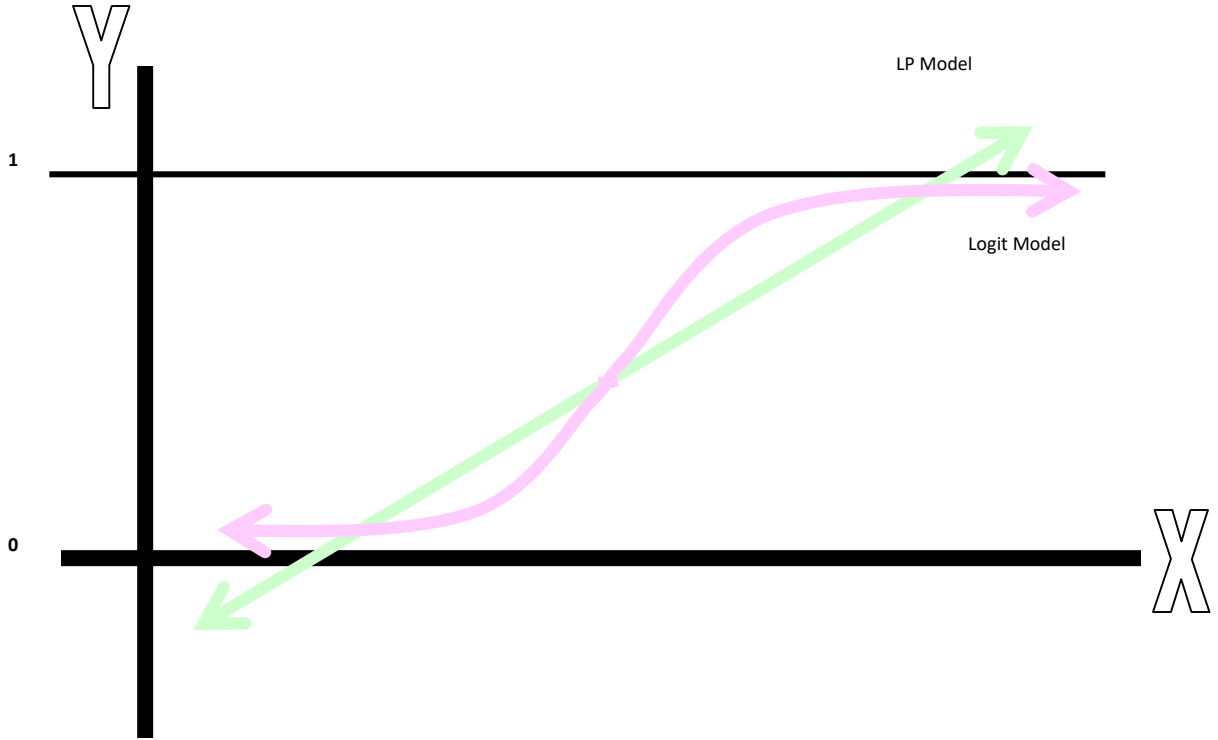
- The logistic distribution constrains the estimated probabilities to lie between 0 and 1.
- The estimated probability is:

$$p = 1/[1 + \exp(-\alpha - \beta X)]$$

- if you let $\alpha + \beta X = 0$, then p = .50
- as $\alpha + \beta X$ gets really big, p approaches 1
- as $\alpha + \beta X$ gets really small, p approaches 0

Flash Memory Summit

Comparing LP and Logit Models



Maximum Likelihood Estimation (MLE)

- MLE is a statistical method for estimating the coefficients of a model.
- The likelihood function (L) measures the probability of observing the particular set of dependent variable values (p_1, p_2, ..., p_n) that occur in the sample:

  $L = \text{Prob}\ (p_1 * p_2 * * * p_n)$

- The higher the L, the higher the probability of observing the ps in the sample.

MLE

- MLE involves finding the coefficients (α, β) that makes the log of the likelihood function (LL < 0) as large as possible
- Or, finds the coefficients that make -2 times the log of the likelihood function (-2LL) as small as possible
- The maximum likelihood estimates solve the following condition:

$$\{Y - p(Y=1)\}X_i = 0$$

summed over all observations, i = 1,…,n

Interpretation

- An interpretation of the logit coefficient which is usually more intuitive is the "odds ratio"
- Since:

$$[p/(1-p)] = \exp(\alpha + \beta X)$$

$\exp(\beta)$ is the effect of the independent variable on the "odds ratio"

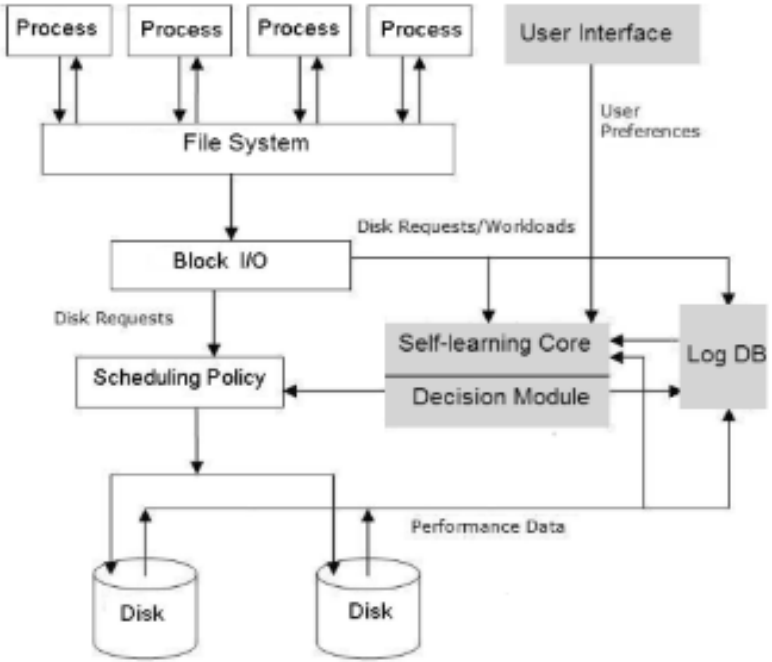
Storage Object Classification

Storage Object Classes

File class	Example policy
File size is zero	Allocate with directory
0 <size≤16KB	Use RAID1 for availability
File lifespan ≤ 1sec	Store in NVRAM
File is write-only	Store in an LFS partition
File is read-only	Aggressively replicate

Table 1. Classes we want to predict.

Architecture overview of self-learning



Source: http://repository.cmu.edu/pdl/69/

Features

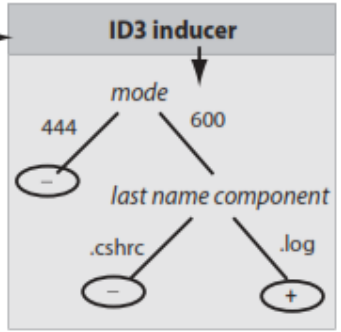
Logged Features for Requests and Workloads

Request Features	Workload Features
Types of current and x previous requests	Number of reads, number of writes, and read/write ratio
Individual request size	Average request size
Sequential or random	Sequential/random ratio
Arrival times of current and x previous requests	Average request arrival rate
Number of processes (issuing requests)	Average number of processes
Think time for each request	Average think time
Inter-request block number distances between current request and x previous requests	
Logical block number of each request	

Example Data and Tree

extension	gid	mode	uid	wronly
.cshrc	18aa0	600	18b72	–
.cshrc	18b11	600	18b7e	–
.cshrc	18aac	600	18b28	–
.cshrc	18b58	600	18c6a	–
.cshrc	18abe	600	18b7f	–
.log	18aad	600	18b2f	+
.log	18aad	600	18b2f	+
.log	18aab	600	18ab4	+
.login	18abe	444	18b7f	–
.html	18abe	444	18b7c	–
.pl	18a90	444	18aa1	–
.txt	18abe	444	18b7c	–

training data



decision tree

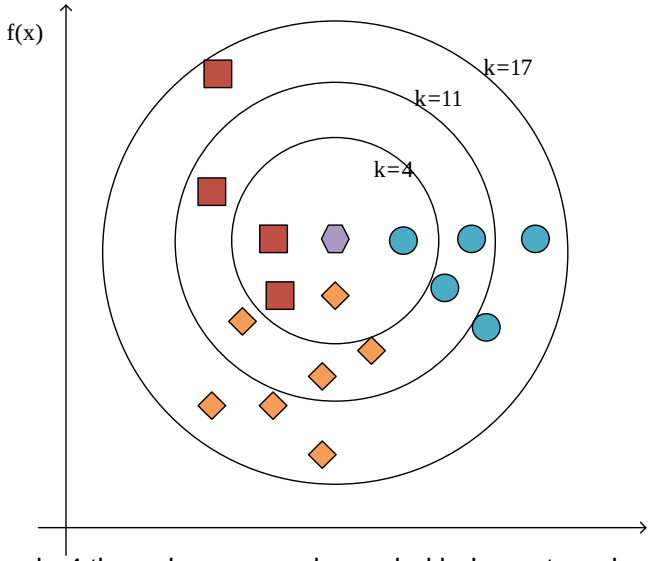
ID3 Algorithm

1. Select attribute A as node to split on (based on relative ranking).
2. Split samples according to values they take on attribute A.
3. If leaf nodes are "pure", done.
4. Else, if attributes remaining, goto 1.

Figure 1. Example data and tree.

k Nearest Neighbor

- Three things are required
  - Feature Space (Training Data)
  - Distance metric - to compute distance between records.
    - Euclidean Distance
    - $d(x_i x_j) = \sqrt{\sum_k (x_{i,k} - x_{j,k})^2}$
    - Manhattan Distance
    - $d(x_i x_j) = \sum_k |x_{i,k} - x_{j,k}|$
  - The value of k – the number of nearest neighbors to retrieve from which to get majority class.
- Choosing the value of k:
  - If k is too small, sensitive to noise points
  - If k is too large, neighborhood may include points from other classes
  - Choose an odd value for k, to eliminate ties



k=4 the unknown purple symbol belongs to red square class
k=11 the unknown symbol belongs to yellow diamond class
K = 17 it belongs to the yellow diamond class still.

More Distances: Quantitative Variables

Identity (absolute) error

$$d_j(x_{ij}, x_{i'j}) = I(x_{ij} \neq x_{i'j})$$

Squared distance

$$d_j(x_{ij}, x_{i'j}) = (x_{ij} - x_{i'j})^2$$

L_q norms

$$L_{qii'} = \left[\sum_j |x_{ij} - x_{i'j}|^q\right]^{1/q}$$

Canberra distance

$$d_{ii'} = \sum_j \frac{|x_{ij} - x_{i'j}|}{|x_{ij} + x_{i'j}|}$$

Data point:

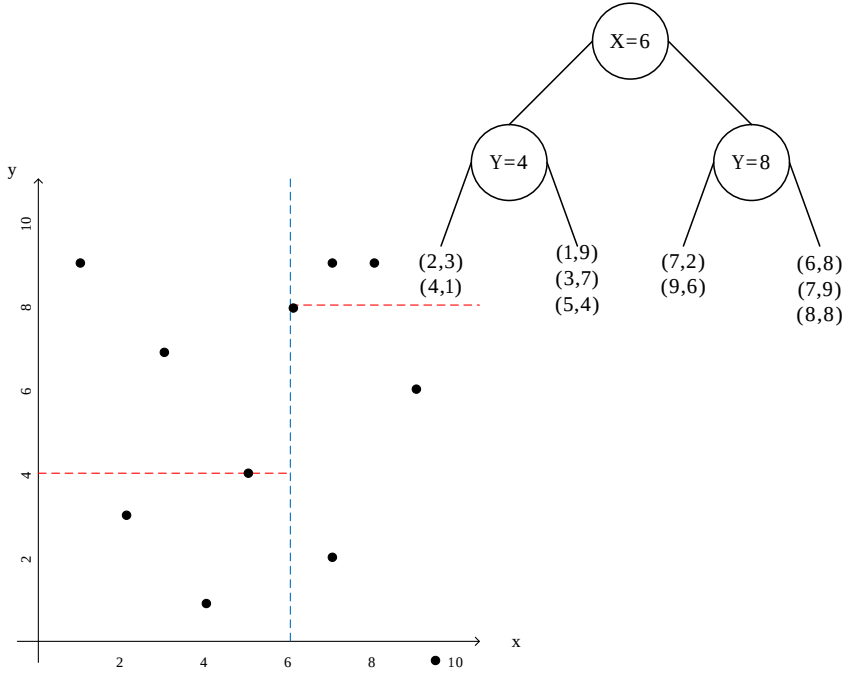
$$x_i = [x_{i1} \ldots x_{ip}]^T$$

Correlation

$$\rho(x_i, x_{i'}) = \frac{\sum_j (x_{ij} - \bar{x}_i)(x_{i'j} - \bar{x}_{i'})}{\sqrt{\sum_j (x_{ij} - \bar{x}_i)^2 \sum_j (x_{i'j} - \bar{x}_{i'})^2}}$$

k Nearest Neighbor

- The straightforward algorithm has a cost of O(nlog(k)), not good if the dataset is large.
- We can use indexing with k-d trees as follows
- For eg. We have the following training data
- {(1,9),(2,3),(4,1),(3,7), (5,4), (6,8), (7,2), (8,8), (7,9), (9,6)}
- Build a K-D tree
  - Pick a dimension, find median, split data, repeat.
- Find nearest neighbors for new point (7,4)
  - Find region contacting (7,4)
  - Compare to all point in region



Decision Trees

- Approximation of discrete functions by a decision tree. In the nodes of trees are attributes and in the leaves are values of discrete function.
- Deriving a tree
- Until each leaf node is populated by as homogeneous a sample set as possible: Select a leaf node with an inhomogeneous sample set. Replace that leaf node by a test node that divides the inhomogeneous sample set into minimally inhomogeneous subsets, according to an entropy calculation.

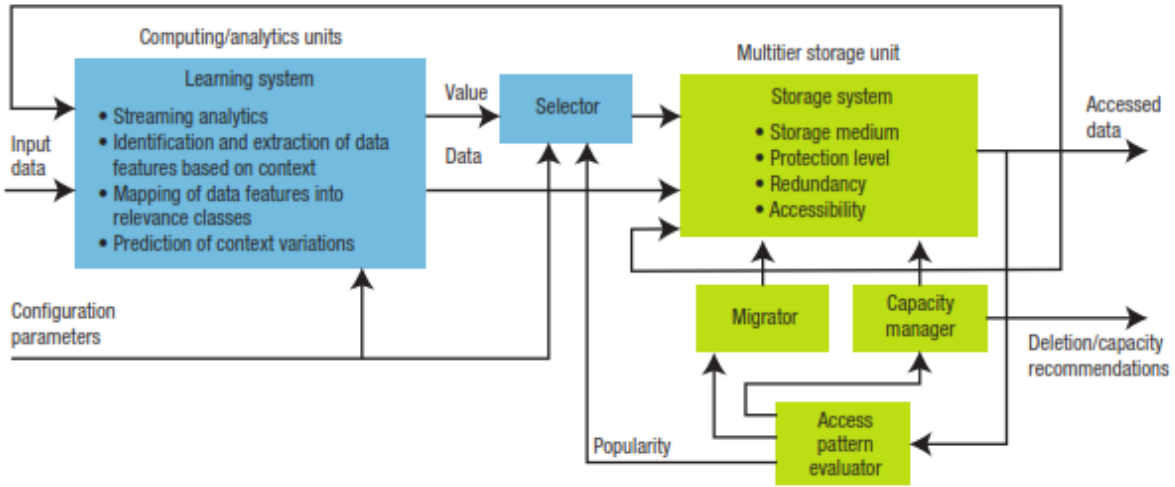
Using Boosted Decision Trees

- Boosting means that each tree is dependent on prior trees, and learns by fitting the residual of the trees that preceded it.
- Flexible: can deal with both continuous and categorical variables
- How to control bias/variance trade-off
  - Size of trees
  - Number of trees
- Boosting trees often works best with a small number of well-designed features
- Boosting “stubs” can give a fast classifier
- supervised learning method and needs labeled dataset

Allocation recommendation engines in Elastic Storage Systems

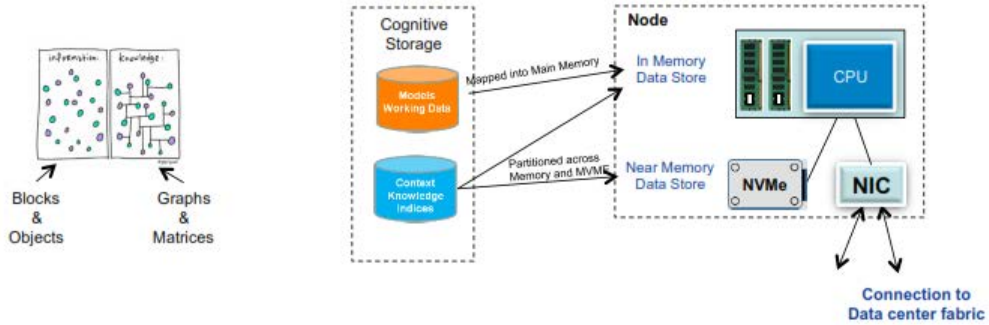
Flash Memory Summit

Block diagram of a storage allocation system



Source: IBM

More on storage allocation



Partitioned into Node vs Cluster shared
Application Optimized (semantics consistent with data types)
Application Direct Accessed (use space IO)
Intelligent Data placement for scalability

K-means Clustering for Storage

An unsupervised clustering algorithm

"*K*" stands for number of clusters, it is typically a user input to the algorithm; some criteria can be used to automatically estimate *K*

It is an approximation to an NP-hard combinatorial optimization problem

K-means algorithm is iterative in nature

It converges, however only a local minimum is obtained

Works only for numerical data

Easy to implement

Source: Nilanjan Ray

K-means: Setup

$x_1, \ldots, x_N$ are data points or vectors of observations

Each observation (vector x_i) will be assigned to one and only one cluster

$C(i)$ denotes cluster number for the i^{th} observation

Dissimilarity measure: Euclidean distance metric

K-means minimizes within-cluster point scatter:

$$W(C) = \frac{1}{2}\sum_{k=1}^{K}\sum_{C(i)=k}\sum_{C(j)=k}\left\|x_i - x_j\right\|^2 = \sum_{k=1}^{K} N_k \sum_{C(i)=k}\left\|x_i - m_k\right\|^2$$

where

m_k is the mean vector of the k^{th} cluster

N_k is the number of observations in k^{th} cluster

K-means Algorithm

For a given cluster assignment *C* of the data points, compute the cluster means m_k:

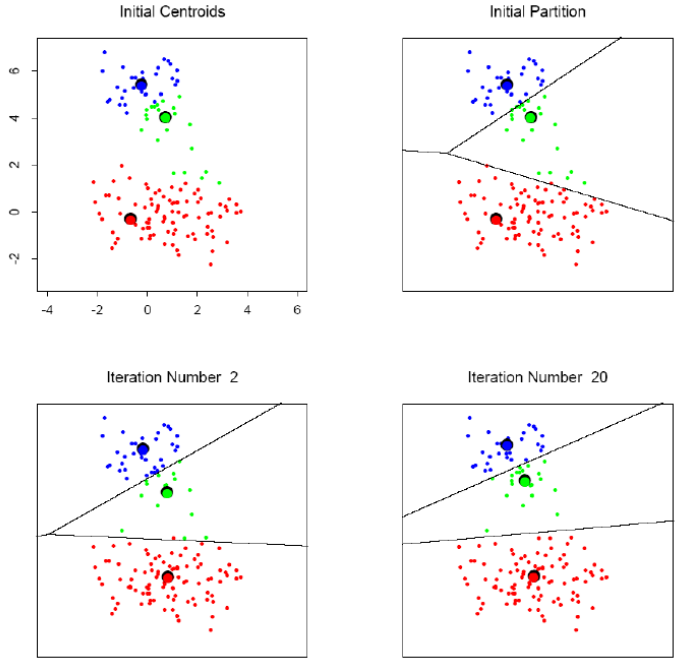
$$m_k = \frac{\sum_{i:C(i)=k} x_i}{N_k},\ k = 1,\ldots,K.$$

For a current set of cluster means, assign each observation as:

$$C(i) = \arg\min_{1 \le k \le K} \left\| x_i - m_k \right\|^2,\ i = 1,\ldots,N$$

Iterate above two steps until convergence

K-means clustering example



How to build an end to end application

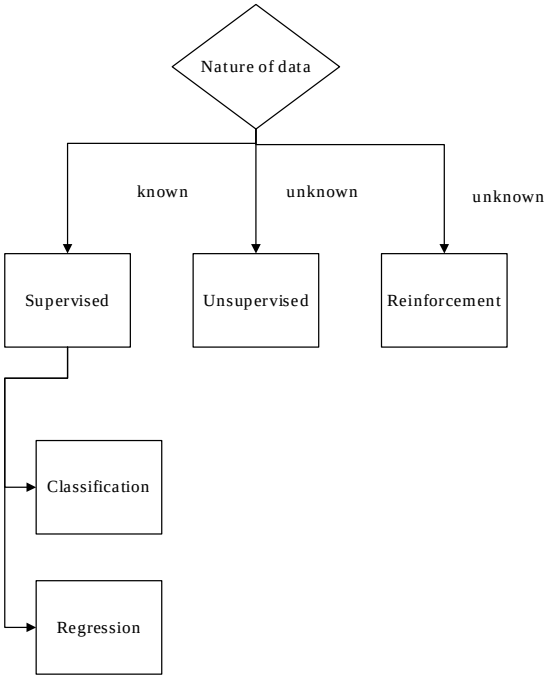
UNDERSTANDING CHALLENGES AND SELECTING RIGHT MACHINE LEARNING ALGORITHM, DATA PREPROCESSING, EVALUATING MODEL

CHALLENGES IN MACHINE LEARNING

- Heterogeneous Data
  - Dataset could have images, text and sensor signals as composite data.
- Preprocessing of data
  - Numeric datasets could have data lying in different data ranges and would require to be normalized before they are used.
- Picking the best suitable model
  - Based on the kind of application we are dealing with we need to pick a suitable algorithm. This decision may not be very intuitive.

Selecting an algorithm

- Selecting a machine learning algorithm deals with having insight about the kind of data with which we are dealing.
- If we know the nature of the data and need to make a future prediction then we should select supervised machine learning algorithms.
- If we do not know the nature of the data would like to discover the patterns within it we should select an unsupervised machine learning algorithms.



The perfect classification algorithm

- Objective function: encodes the right loss for the problem
- Parameterization: makes assumptions that fit the problem
- Regularization: right level of regularization for amount of training data
- Training algorithm: can find parameters that maximize objective on training set
- Inference algorithm: can solve for objective function in evaluation

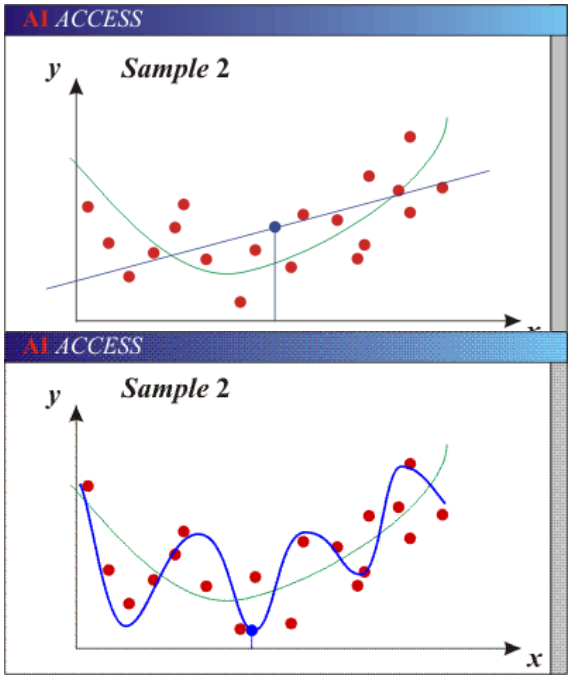
Selecting Model

Try simple classifiers first

Better to have smart features and simple classifiers than simple features and smart classifiers

Use powerful classifiers if more training data is available (bias-variance tradeoff)

Bias-Variance Trade-off



- Models with too few parameters are inaccurate because of a large bias (not enough flexibility).
- Models with too many parameters are inaccurate because of a large variance (too much sensitivity to the sample).

Source: D. Hoiem

Bias-Variance Trade-off

$$E(MSE) = noise^2 + bias^2 + variance$$

Unavoidable error

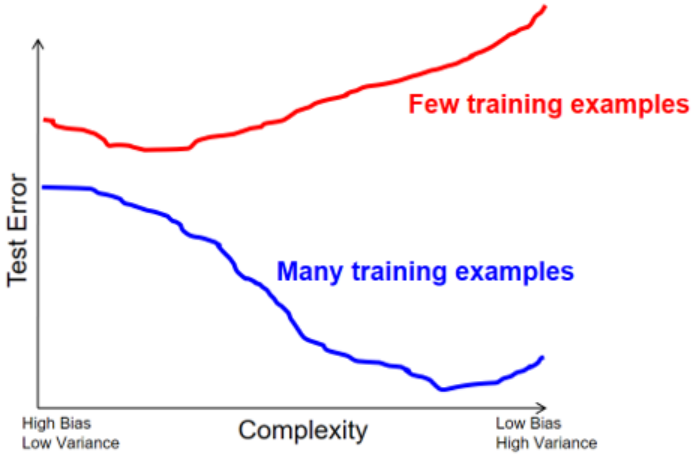
Error due to incorrect assumptions

Error due to variance of training samples

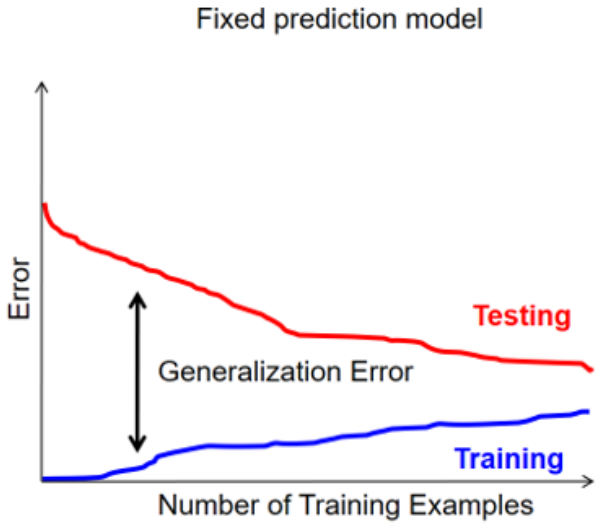
See the following for explanations of bias-variance (also Bishop's "Neural Networks" book):

- http://www.inf.ed.ac.uk/teaching/courses/mlsc/Notes/Lecture4/BiasVariance.pdf

Bias-Variance Trade-off

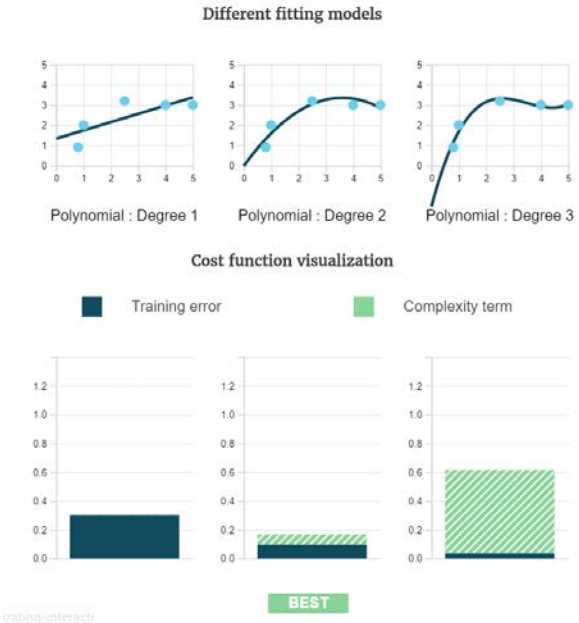


Effect of Training Size



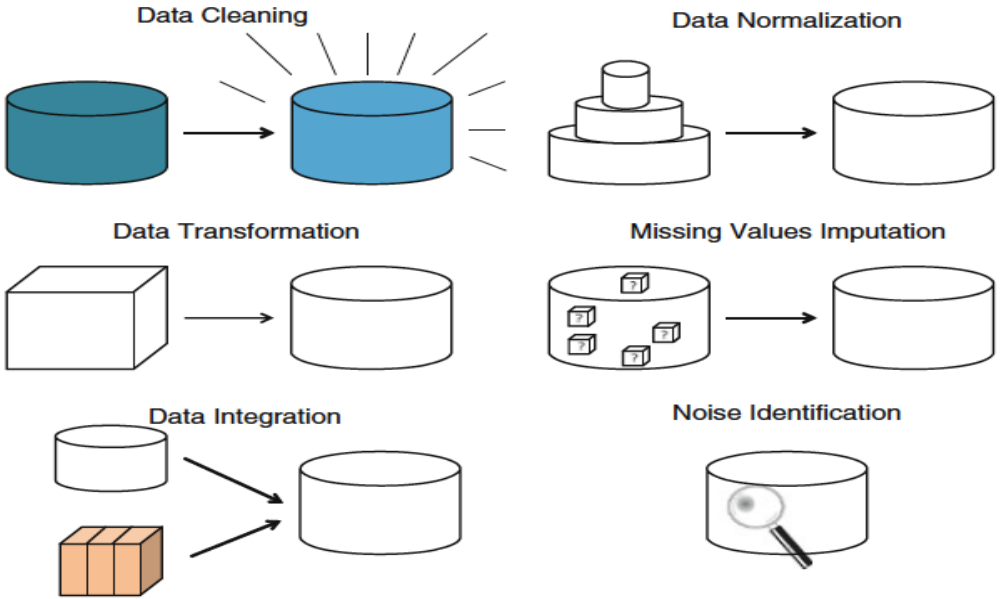
Regularization

- When the model fits the training data but does not have a good predicting performance and generalization power, we have an overfitting problem.
- Regularization is a technique used to avoid this overfitting problem.
- The idea behind regularization is that models that overfit the data are complex models that have for example too many parameters.



Source: datanice

Data Preprocessing



Data Pre-processing

- Data Cleaning
  - Correct bad data, filter some incorrect data out of the data set and reduce the unnecessary detal of data.
- Data Transformation/normalization/Feature scaling
  - Feature scaling is a method used to standardize the

$$x' = \frac{x - \bar{x}}{\sigma}$$

Where x is the original feature vector, $\bar{x}$ is the mean of that feature vector, and σ is its standard deviation.

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)}$$

Model Assessment

If we are in a data-rich situation, the best approach for both *model selection* and *model assessment* is to randomly divide the dataset into three parts: training set, validation set, and test set.

The *training set* is used to fit the models. The *validation set* is used to estimate prediction error for model selection. The *test set* is used for assessment of the prediction error of the final chosen model.

A typical split might by 50% for training, and 25% each for validation and testing.

Source: Nathan Bastian

Resampling methods –a simple list

- The validation set approach,
- Leave-one-out cross-validation, and *K*-fold cross-validation.
- The bootstrap method for assessing statistical accuracy.

- Please see **http://tinyurl.com/ybojkt9f**
- **And http://tinyurl.com/y8oxwq9s**
- **For more details.**

Thanks!