

Optimization in Big Data Research (I) Introduction

Jianyong Sun
School of Mathematics and Statistics
Xi'an Jiaotong University

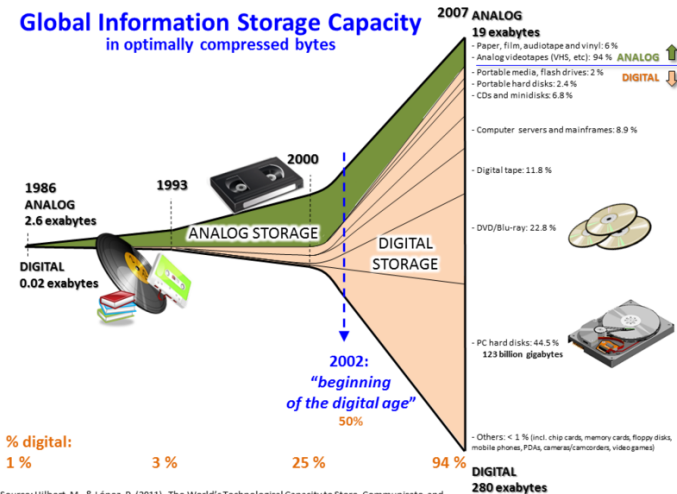
Xi'an, 2018

Table of Contents

- 1 What is Big Data?
- 2 Machine Learning
- 3 What is Learning?
- 4 The Cost Functions: Examples
- 5 Conclusions



In 2012, everyday 2.5 exabytes (2.5×10^{18}), grow exponentially 4.4 zettabytes (10^{21}) to 44 zettabytes between 2013 and 2020. By 2025, 163 zettabytes (by International Data Corporation)



Definition

- **Big data** is data sets that are so big and complex that traditional data-processing application software are inadequate to deal with them.
- **Big data challenges** include capturing data, data storage, data analysis, search, sharing, transfer, visualization, querying, updating, information privacy, data source and others.
- **Characteristics:** volume, variety, velocity. Other concepts later attributed with big data are veracity (i.e., how much noise is in the data) and value.
- **Trends:** the term "big data" tends to refer to the use of predictive analytics, user behavior analytics, or certain other advanced data analytics methods that extract value from data, and seldom to a particular size of data set.

Big Data Analysis

- Analysis of data sets can find new correlations to "spot business trends, prevent diseases, combat crime and so on."
- Scientists, business executives, practitioners of medicine, advertising and Governments alike regularly meet difficulties with large data-sets in areas including Internet search, financial technology, urban informatics, and business informatics.
- Scientists encounter limitations in e-Science work, including meteorology, genomics, astroinformatics, complex physics simulations, biology, environmental research and many others.

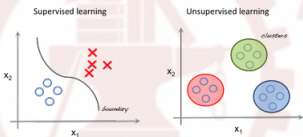
Table of Contents

- 1 What is Big Data?
- 2 **Machine Learning**
 - Machine Learning Tasks
 - Machine Learning Approaches
- 3 What is Learning?
- 4 The Cost Functions: Examples
- 5 Conclusions

- Within the field of **data analytics**, machine learning is a method used to devise complex models and algorithms that lend themselves to prediction; in commercial use, this is known as **predictive analytics**.
- These analytical models allow to **"produce reliable, repeatable decisions and results"** and uncover **"hidden insights"** through learning from historical relationships and trends in the data.
- Machine learning often uses statistical techniques to give computers the ability to **"learn"** (i.e., progressively improve performance on a specific task) from data.
- Machine learning grows out of the quest for artificial intelligence. It has strong ties to **mathematical optimization**, which delivers methods, theory and application domains to the field.

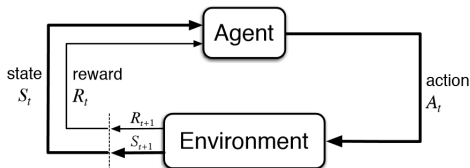
Machine Learning Tasks

- **Supervised learning:** data are (input, output) pairs, given by a "teacher", and the goal is to learn a general rule that maps inputs to outputs.



- **Unsupervised learning:** No labels are given to the learning algorithm, leaving it on its own to find structure in its input. Unsupervised learning can be a goal in itself (discovering hidden patterns in data) or a means towards an end (feature learning).

- As special cases, the input signal can be only partially available, or restricted to special feedback:
 - **Semi-supervised learning**: incomplete training, a training set with some (often many) of the target outputs missing.
 - **Active learning**: the computer can only obtain training labels for a limited set of instances (based on a budget), and also has to optimize its choice of objects to acquire labels for. When used interactively, these can be presented to the user for labeling.
- **Reinforcement learning**: training data (in form of rewards and punishments) is given only as feedback to the program's actions in a dynamic environment (unmanned car), such as driving a vehicle or playing a game (e.g. GO) against an opponent.



Applications of Machine Learning

- **Biomedical informatics:** health care, remote medical, etc.
- **Computer Vision:** gaining high-level understanding from digital images or videos
- **Natural language processing:** speech/handwritten recognition, natural language understanding, and natural language generation.
- **Pattern recognition:** facial recognition, handwriting recognition, image recognition, speech recognition, etc.
- **Recommendation system:** collaborative filtering
- **Search engine**
- ...

Decision Tree Learning

- Uses a **decision tree** as a predictive model to map observations about an item to conclusions about the item's target value.

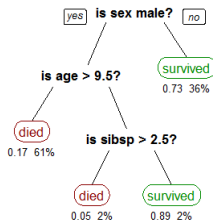
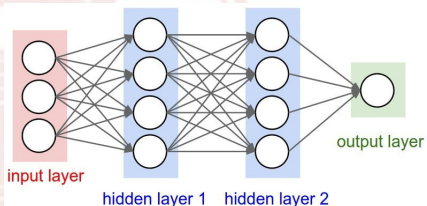
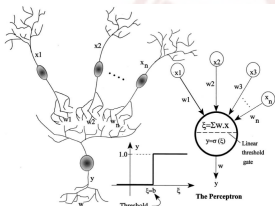


Figure: A tree showing survival of passengers on the Titanic.

- Different metrics: information gain, variance reduction, etc.
- Ensemble methods: Boosted trees (AdaBoost), Bootstrap aggregated or bagged decision trees (Random forest)
- Algorithms: ID3, C4.5, CART, MARS, etc.

Artificial Neural Networks

- inspired by biological neural networks. Modern neural networks are non-linear statistical data modeling tools.



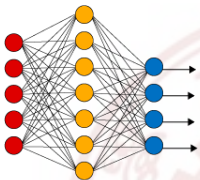
- used to model complex relationships between inputs and outputs, to find patterns in data, or to capture the statistical structure in an unknown joint probability distribution between observed variables.
- The multilayer perceptron is a **universal function approximator**, as proven by the universal approximation theorem.

Deep Learning

- based on **learning data representations**, as opposed to task-specific algorithms.
 - learn multiple levels of representations that correspond to different levels of abstraction; the levels form a hierarchy of concepts.
- to model the way the human brain processes light and sound into vision and hearing
- Deep learning architectures: deep neural networks, deep belief networks, recurrent neural networks, autoencoders
- particularly successful in computer vision, speech recognition, natural language processing, board game programs, and others.

Machine Learning Approaches

Simple Neural Network

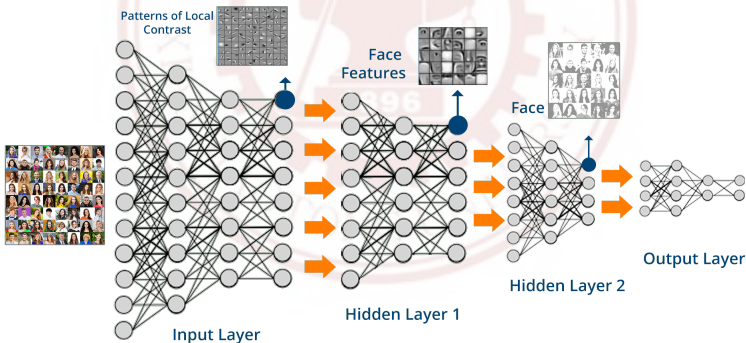
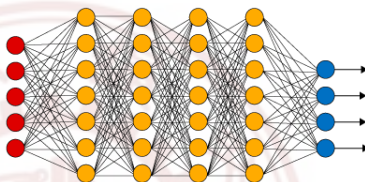


● Input Layer

● Hidden Layer

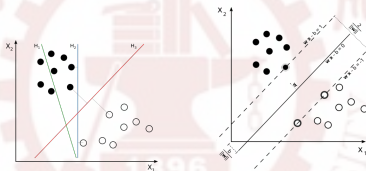
● Output Layer

Deep Learning Neural Network

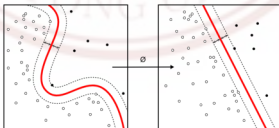


Support Vector Machine

- supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis.
- An SVM model is to divide separate classes by a clear gap that is as wide as possible.

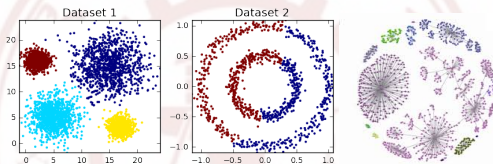


- SVMs efficiently perform a non-linear classification using the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces.



Clustering – Cluster Analysis

- to group a set of objects in such a way that objects in the same group are more similar (in some sense) to each other than to those in other groups



- Popular notions of clusters: groups with small distances between cluster members, dense areas of the data space, intervals or particular statistical distributions.
- Algorithms
 - Connectivity-based clustering (hierarchical clustering)
 - Centroid-based clustering (k-means)
 - Distribution-based clustering (Mixture of Gaussian)
 - Density-based clustering (DBSCAN)
 - ...

Representation Learning (feature learning)

- to automatically discover the representations needed for feature detection or classification/regression from raw data
- allows a machine to both learn the features and use them to perform a specific task
- attempt to preserve the information in their input but transform it in a way that makes it useful, often as a pre-processing step before performing classification or predictions
- Feature learning can be either supervised or unsupervised.
 - Supervised, examples: supervised neural networks, multilayer perceptron and (supervised) dictionary learning.
 - Unsupervised, examples: (unsupervised) dictionary learning, independent component analysis, PCA, local linear embedding, matrix factorization and clustering
 - Multilayer/deep architectures: Autoencoders, Restricted Boltzmann Machine, Deep Belief Network

Manifold Learning, Sparse coding, etc.

- **Manifold learning** algorithms attempt to learn low-dimensional representation (cf dimensionality reduction).
- **Sparse coding** algorithms attempt to learn sparse representation (has many zeros).
- **Multilinear subspace learning** algorithms aim to learn low-dimensional representations directly from tensor representations for multidimensional data, without reshaping them into (high-dimensional) vectors.
- **Deep learning** algorithms discover multiple levels of representation, or a hierarchy of features, with higher-level, more abstract features defined in terms of (or generating) lower-level features.

Sparse Dictionary Learning

- to find a sparse representation of the input data (sparse coding) in the form of a linear combination of basic elements (**atoms**) as well as those basic elements themselves.
- One of the most important applications of sparse dictionary learning is in the field of **compressed sensing** or signal recovery.
- Applications: data decomposition, compression and analysis and image denoising and classification, video and audio processing.
- Sparsity and overcomplete dictionaries have immense applications in image compression, image fusion and inpainting.



Feature Selection

- to select a subset of relevant features (variables, predictors) for use in model construction
 - simplification of models to make them easier to interpret by researchers/users,
 - shorter training times,
 - to avoid the curse of dimensionality,
 - enhanced generalization by reducing overfitting
- The central premise when using a feature selection technique is that the data contains some features that are either redundant or irrelevant, and can thus be removed without incurring much loss of information.

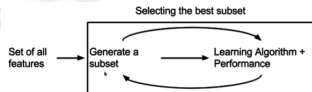
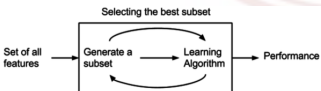
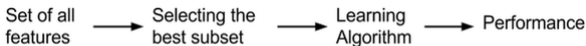


Table of Contents

- 1 What is Big Data?
- 2 Machine Learning
- 3 What is Learning?**
 - Choosing a Cost Function
- 4 The Cost Functions: Examples
- 5 Conclusions



Learning

- Given a specific task to solve, and a class of functions F , **learning** means using a set of observations (**data**) to find $f^* \in F$ which solves the task in some optimal sense.
- This entails defining a cost function $C : F \rightarrow \mathbb{R}$ such that, for the optimal solution f^* , $C(f^*) \leq C(f), \forall f \in F$ – i.e., no solution has a cost less than the cost of the optimal solution.
- The cost function C is a measure of how far away a particular solution is from an optimal solution to the problem to be solved. Learning algorithms search through the function space to find a function that has the smallest possible cost.
- For applications where the solution is data dependent, the cost *must necessarily be a function of the observations*, otherwise the model would not relate to the data.

The cost function depends on the learning task.

Learning Paradigms

- Supervised Learning: Empirical Risk Minimization
- Unsupervised Learning: Maximum Likelihood Estimation
- Reinforcement Learning

Supervised Learning

- Given a set of data $\mathcal{D} = \{(x_i, y_i), x_i \in X, y_i \in Y, 1 \leq i \leq m\}$, the aim is to find a function (hypothesis) $f : X \rightarrow Y$ in the allowed class of functions
- **joint probability distribution** $P(x, y)$ over X, Y exist but **unknown**, data $(x_i, y_i) \in \mathcal{D}$ are drawn **i.i.d.** from $P(x, y)$
- the assumption of a joint probability distribution allows us to model uncertainty in predictions (e.g. from noise in data) because y is not a deterministic function of x but rather a random variable with conditional distribution $P(y|x)$ for a fixed x .
- **Loss function** $L(\hat{y}, y)$: non-negative, $\hat{y} = f(x)$: the prediction of f , y is the true outcome. Examples: mean square error, hinge loss, etc.

Risk

The risk associated with hypothesis $f(x)$ is then defined as the expectation of the loss function:

$$R(f) = \mathbb{E}[L(f(x), y)] = \int L(f(x), y) dP(x, y)$$

The ultimate goal of a learning algorithm is to find a hypothesis f^* among a fixed class of functions \mathcal{F} for which the risk $R(f)$ is minimal:

$$f^* = \arg \min_{f \in \mathcal{F}} R(f)$$

$R(f)$ cannot be computed since $P(x, y)$ is unknown.

Empirical Risk:

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

The empirical risk minimization principle states that the learning algorithm should choose a hypothesis \hat{f} which minimizes the empirical risk:

$$\hat{f} = \arg \min_{f \in \mathcal{F}} R_{\text{emp}}(f).$$

Maximum Likelihood Estimation: used in computational statistics, to estimate parameters of a statistical model by maximizing a likelihood function, given the observations.


- **Likelihood function:** $\mathcal{L}(\theta; \mathcal{D})$ given a family of distributions $\{f(\cdot \dots \cdot; \theta) | \theta \in \Theta\}$ where θ denotes the parameter for the model.
- The method defines a maximum likelihood estimate (MLE)

$$\hat{\theta} \in \{\arg \max_{\theta \in \Theta} \mathcal{L}(\theta; \mathcal{D})\} \text{ or } \{\arg \max_{\theta \in \Theta} \prod_{i=1}^m f(\theta; x_i)\} \text{ if } \mathcal{D} \text{ is i.i.d.}$$

if a maximum exists; or equivalently the log-likelihood

$$\hat{\ell}(\theta; \mathcal{D}) = \frac{1}{m} \sum_{i=1}^m \log f(x_i; \theta)$$

Table of Contents

- 1 What is Big Data?
 - 2 Machine Learning
 - 3 What is Learning?
 - 4 The Cost Functions: Examples**
 - 5 Conclusions
- 
- The background of the slide features a large, faint watermark of the Tsinghua University logo. The logo is circular and contains a gear, a book, and a lamp, with the year '1896' at the bottom. The text 'TSINGHUA UNIVERSITY' is written around the perimeter of the circle.

Example 1: Computing the soft-margin **linear SVM** classifier amounts to minimizing

$$\left[\frac{1}{m} \sum_{i=1}^m \max(0, 1 - y_i(w \cdot x_i - b)) \right] + \lambda \|w\|^2$$

Here hinge loss

$$\ell(y, z) = \max(0, 1 - yz)$$

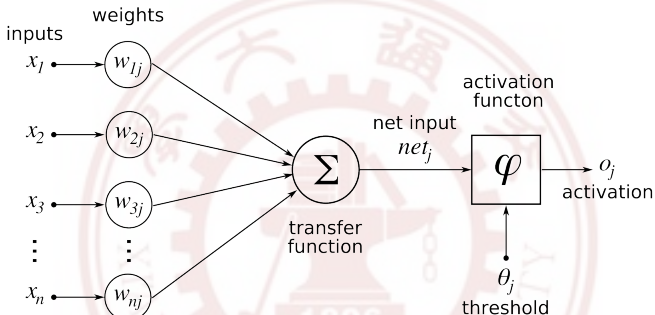
By introducing $\zeta_i = \max(0, 1 - y_i(w \cdot x_i - b))$, the problem becomes

$$\begin{cases} \text{minimize } \frac{1}{m} \sum_{i=1}^m \zeta_i + \lambda \|w\|^2 \\ \text{s.t. } y_i(w \cdot x_i - b) \geq 1 - \zeta_i \text{ and } \zeta_i \geq 0 \forall i \end{cases} \rightarrow \text{the primal problem}$$

By solving for the Lagrangian dual, we obtain the simplified problem

$$\begin{cases} \text{maximize } f(c_1, \dots, c_m) = \sum_{i=1}^m c_i - \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^m y_i c_i \langle x_i, x_j \rangle y_j c_j \\ \text{s.t. } \sum_{i=1}^m c_i y_i = 0, \text{ and } 0 \leq c_i \leq \frac{1}{2n\lambda}, \forall i \\ \rightarrow \text{the dual problem (quadratic programming)} \end{cases}$$

Example 2: Deep Neural Networks



- a single neuron will pass a message to another neuron across this interface if the sum of weighted input signals from one or more neurons (summation) into it is great enough (exceeds a threshold) to cause the message transmission. This is called **activation** when the threshold is exceeded and the message is passed along to the next neuron

A neuron's network function:

$$\text{net}_j^1 = \sum_i w_{ij}^1 x_i + b_j^1, o_j^1 = \varphi(\text{net}_j^1) \rightarrow \text{the first layer}$$

$$\text{net}_j^t = \sum_i w_{ij}^t o_i^{t-1} + b_j^t, o_j^t = \varphi(\text{net}_j^{t-1}) \rightarrow \text{the t-th layer}$$

where φ (the activation function) such as the

- hyperbolic tangent: $\tanh = \frac{e^x - e^{-x}}{e^x + e^{-x}} \in (-1, 1)$
- sigmoid function: $\sigma(x) = \frac{1}{1 + e^{-x}} \in (0, 1)$
- softmax function:

$$\sigma : \mathbb{R}^K \rightarrow \left\{ \sigma \in \mathbb{R}^K \mid \sigma_i > 0, \sum_{i=1}^K \sigma_i = 1 \right\}, \sigma(\mathbf{z})_j = \frac{e^{z_j}}{\sum_k e^{z_k}} \text{ for } j = 1, \dots, K$$
- rectifier function: $f(x) = x_+ = \max(0, x)$

Given data set $\mathcal{D} = \{(x_i, y_i), 1 \leq i \leq m\}$, the cost function

$$C(\theta) = \frac{1}{m} \sum (f(x_i) - y_i)^2 \rightarrow \text{mean square error}$$

To ensure certain features of θ , various regularization terms used

$$C(\theta) = \frac{1}{m} \sum (f(x_i) - y_i)^2 + \lambda \|w\|_p^p, p > 0$$

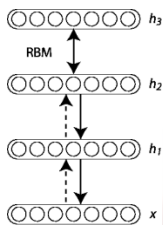
In case of linear regression $y_i = x_i \beta + e_i, i = 1, \dots, m, x_i \in \mathbb{R}^p$

$$C(\theta) = \frac{1}{m} \|\mathbf{y} - \mathbf{X}\beta\|^2 + \begin{cases} \lambda \|\beta\|_2^2 & \text{ridge regression} \\ \lambda \|\beta\|_1 & \text{LASSO} \\ \lambda \|\beta\|_2^2 + \lambda_2 \|\beta\|_1 & \text{elastic net} \\ \lambda \sqrt{\|\beta\|_{1/2}} & \text{bridge regression} \\ \dots & \end{cases}$$

where

$$\mathbf{y} = (y_1, \dots, y_m)^T, \mathbf{X} = [x_1, \dots, x_m], \|\beta\|_{1/2} = \left(\sum_j \sqrt{|\beta_j|} \right)^2.$$

Example 3: Deep Belief Network (DBN)

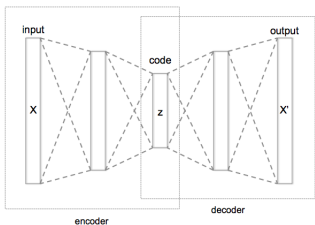


- A generative graphical model, composed of multiple layers of latent variables (hidden units) with connections between the layers but not between units within each layer.
- DBN acts as feature detectors when trained on data without supervision. After learning DBN, it can be further trained with supervision to perform classification.
- Maximum Likelihood Estimation:

$$p(x) = \frac{1}{Z} \sum_h e^{-E(x,h)}$$

where Z is the partition function, and $E(x, h)$ is the energy function assigned to the state of the network.

Example 4: Autoencoder: ANN for efficient data coding learning

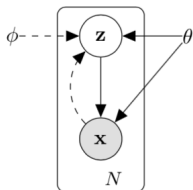


- It consists of two parts: the **encoder** and the **decoder**

$$\phi : \mathcal{X} \rightarrow \mathcal{F}$$

$$\psi : \mathcal{F} \rightarrow \mathcal{X}$$

$$\phi^*, \psi^* = \arg \min_{\phi, \psi} \|X - (\phi \circ \psi)X\|_2^2$$



- Denoising Autoencoder: corrupt x to \tilde{x} , use \tilde{x} for normal autoencoder, loss function $\mathcal{L}(x, \tilde{x}')$

- Contractive Autoencoder:

$$\mathcal{L}(x, x') + \lambda \sum_i \|\nabla_x h_i\|^2$$

- Variational Autoencoder: $p_\theta(x|z)$ (decoder), the encoder $q_\phi(z|x)$, the objective

$$\mathcal{L}(\phi, \theta, x) = D_{KL}(q_\phi(z|x) \| p_\theta(z)) - \mathbb{E}_{q_\phi(z|x)}(\log p_\theta(x|z))$$

Example 5: Sparse Dictionary Learning:

Given an input dataset $X = [x_1, \dots, x_K]$, $x_i \in \mathbb{R}^d$, we wish to find an **overcomplete dictionary** $\mathbf{D} \in \mathbb{R}^{d \times n}$ ($n > d$) and a representation $R = [r_1, \dots, r_K]$, $r_i \in \mathbb{R}^n$ such that both $\|X - \mathbf{D}R\|_F^2$ is minimized and the representations r_i are **sparse** enough.

$$\arg \min_{\mathbf{D} \in \mathcal{C}, r_i \in \mathbb{R}^n} \sum_{i=1}^K \|x_i - \mathbf{D}r_i\|_2^2 + \lambda \|r_i\|_0$$

where

$$\mathcal{C} = \{\mathbf{D} \in \mathbb{R}^{d \times n} : \|d_i\|_2 \leq 1, \forall i = 1, 2, \dots, n\}$$

Given dictionary \mathbf{D} ,

$$\arg \min_{r \in \mathbb{R}^n} \frac{1}{2} \|x - \mathbf{D}r\|_2^2 + \lambda \|r\|_0 \rightarrow \text{Sparse Approximation (Coding)}$$

Example 6: Sparse Approximation (Sparse Representation):

Consider a linear system of equations $x = \mathbf{D}\alpha$, where $x \in \mathbb{R}^m$, $\alpha \in \mathbb{R}^p$, $\mathbf{D} \in \mathbb{R}^{m \times p}$ ($m < p \rightarrow$ underdetermine linear system), to seek an α^* that has the fewest non-zeros

- Noiseless observations.

$$\min_{\alpha \in \mathbb{R}^p} \|\alpha\|_0 \text{ subject to } x = \mathbf{D}\alpha$$

- Noise observations

$$\min_{\alpha \in \mathbb{R}^p} \|\alpha\|_0 \text{ subject to } \|x - \mathbf{D}\alpha\|_2^2 \leq \epsilon^2$$

or in a Lagrangian form,

$$\min_{\alpha \in \mathbb{R}^p} \lambda \|\alpha\|_0 + \frac{1}{2} \|x - \mathbf{D}\alpha\|_2^2$$

Due to the ℓ_0 pseudo-norm, both problems are NP-hard. ℓ_1 -norm relaxation ensures sparsity, but convex optimization problems.

Example 7: Feature Selection: features to be selected based various metrics

- Minimum-redundancy-maximum-relevance (mRMR)

$$\max_S \left[\frac{1}{|S|} \sum_{f_i \in S} I(f_i; c) - \frac{1}{|S|^2} \sum_{f_i, f_j \in S} I(f_i; f_j) \right]$$

where $I(\cdot, \cdot)$ denotes mutual information; equ. to

$$\min_x \{ \alpha x^T H x - x^T F \}, \text{ s.t. } \sum x_i = 1, x_i \geq 0$$

where $F = [I(f_i; c)]$, $H = [I(f_i; f_j)]$ and x - feature weights

- Conditional Relevancy

$$\max_x \{ x^T Q x \}, \text{ s.t. } \|x\| = 1, x_i \geq 0$$

where $Q_{ii} = I(f_i; c)$, $Q_{ij} = I(f_i; c | f_j)$, $i \neq j$

- the Hilbert-Schmidt Independence Criterion Lasso (HSIC Lasso)

$$\min_x \frac{1}{2} \sum_{k,l} x_k x_l \text{HSIC}(f_k, f_l) - \sum_k x_k \text{HSIC}(f_k, c) + \lambda \|x\|_1$$

where $\text{HSIC}(f_k, c) = \text{tr}(\bar{\mathbf{K}}^{(k)} \bar{\mathbf{L}})$ or equ. to

$$\min_x \frac{1}{2} \left\| \bar{\mathbf{L}} - \sum_k x_k \bar{\mathbf{K}}^{(k)} \right\|_F^2 + \lambda \|x\|_1 \rightarrow \text{Lasso}$$

Example 8: Distribution-based Clustering: Mixture of Gaussian

- Assume a data generative model:

$$p(x) = \sum_{k=1}^K \pi_k \mathcal{N}(x | \mu_k, \sigma_k), x \in \mathbb{R}^d$$

- Given a set of data $\mathcal{D} = \{x_1, \dots, x_N\}$, parameters $\theta = \{\mu_k, \Sigma_k, 1 \leq k \leq K\}$ to be estimated through MLE

$$\mathcal{L}(\theta; \mathcal{D}) = \prod_{i=1}^N p(x_i) = \prod_{i=1}^N \left(\sum_k \pi_k \mathcal{N}(x_i | \mu_k, \sigma_k) \right)$$

- To estimate K ,
 - information criteria, such as AIC, BIC, MML, MDL, etc.
 - Variational Bayes

$$\mathcal{L}(\theta, \mathcal{M}) = p(\theta | \mathcal{M}) p(\mathcal{M})$$

Example 9: Low-rank Approximation

- to fit a given matrix (the data) and an approximating matrix (the optimization variable), subject to a constraint that the approximating matrix has reduced **rank** (maybe other constraints)
- Basic low-rank approximation

$$\min_{\hat{D}} \|D - \hat{D}\|_F \text{ subject to } \text{rank}(\hat{D}) \leq r$$

→ **Singular Value Decomposition (SVD)**

- Weighted low-rank approximation

$$\min_{\hat{D}} \text{vec}^T(D - \hat{D})W\text{vec}(D - \hat{D}) \text{ subject to } \text{rank}(\hat{D}) \leq r$$

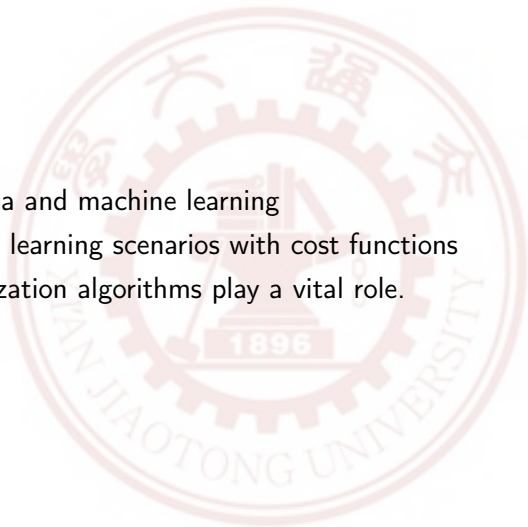
where **vec** vectorizes A column-wise and W is a given positive (semi)definite weight matrix.

Table of Contents

- 1 What is Big Data?
- 2 Machine Learning
- 3 What is Learning?
- 4 The Cost Functions: Examples
- 5 Conclusions**



- Big data and machine learning
- Various learning scenarios with cost functions
- Optimization algorithms play a vital role.



Questions?

