Introduction to machine learning and AI

Silèye Ba

Lecturer, Neoma Business School, Paris
Machine Learning Scientist, L'Oreal Research & Innovation, Paris

08.01.2023

Overview

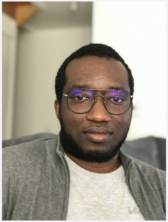
1. Course organization
   - Lecturers
   - Course organization
2. Introduction to machine learning and AI
   - Glossary : meaning of all these concepts
   - Machine learning approaches
   - Machine learning applications
   - Machine learning projects in industry
3. Some problems faced in ML project
   - Model training cost
   - Bias in datasets
   - Other issues
4. Next session

Overview

1. Course organization
   - Lecturers
   - Course organization
2. Introduction to machine learning and AI
   - Glossary : meaning of all these concepts
   - Machine learning approaches
   - Machine learning applications
   - Machine learning projects in industry
3. Some problems faced in ML project
   - Model training cost
   - Bias in datasets
   - Other issues
4. Next session

Lecturers

Silèye BA



- PhD, Ecole Polytechnique de Lausanne, 2007
- Machine Learning Scientist, L'Oreal Research & Innovation, Paris

Anindya ROY

- PhD, Ecole Polytechnique de Lausanne, 2011
- Data Scientists Manager at PMP - Performance Management Partner, Paris

Course program

Topic	Session
Introduction to machine learning	Lecture
Gradient descent in machine learning	Labs
Supervised learning	Lecture
Supervised learning	Labs
Unsupervised learning	Lecture
Unsupervised learning	Labs
Reinforcement learning	Lecture
Reinforcement learning	Labs

Session	**Introduction to machine learning**	**Supervised learning**	**Unsupervised learning**	**Reinforcement learning**
	Course organisation	K nearest neighbors	Clustering (kmean, kernel kmeans)	Multi arm bandits
	Machine learning quick survey	Decision trees & random forest	Embeddings (PCA, Kernel PCA, autoencoder)	Markov decision processes
Lectures	Applications in industry	Kernel methods (SVM)	Data distribution modelling (mixture models, GANs)	TD learning
		Deep neural networks		Q learning
	Intro to Python/Jupyter/Scikit/Keras	House price prediction	Text clustering & topic modelling	Multi arm bandit based portfolio manager
		Text classification	Deep portfolio	RL based portfolio manager
(Possible) Labs		Asset price movement prediction	Assets prices sampling with GANs	
		Index tracker with ML		

Course evaluation

- Two-three persons' group projects : notebook + report
- Individual projects : notebook + report
- Final grades : average marks

Bibliography : books, blogs, video lectures

Books

- **Deep learning**, I. Goodfellow, Y. Bengio, and A. Courville, Adaptive Computation and Machine Learning Series
- **An introduction to reinforcement learning**, R.S. Sutton, and A.G. Barto, MIT Press,
- **Machine learning in finance : from theory to practise**, M. F. Dixon, I. Halperin, P. Bilokon, Springer
- **Advances in financial machine learning**, M. lopez de Prado, Wiley,
- **Machine learning for algorithmic trading**, S. Jansen, Packt.

Blogs & online lectures

- DS Medium blogs : Towards Data Science, KDnuggets, ODSC, ...
- Coursera, KDD, NEURIPS, ...

Prerequisites for modern machine learning

Mathematics

- Calculus and optimization
- Linear algebra
- Probabilities and statistics

Programming

- Python : core programming
- Jupyter notebooks : rapid prototyping
- Scikit learn : classical machine learning
- Tensorflow/Keras, Pytorch : deep learning

Students background

Whats your background, major

- Economy
- Finance
- Mathematics
- Telecom
- Computer science
- Law
- Others

Overview

1. Course organization
   - Lecturers
   - Course organization
2. Introduction to machine learning and AI
   - Glossary : meaning of all these concepts
   - Machine learning approaches
   - Machine learning applications
   - Machine learning projects in industry
3. Some problems faced in ML project
   - Model training cost
   - Bias in datasets
   - Other issues
4. Next session

Important concepts (1/3)

Statistics

The discipline that concerns the collection, organization, displaying, analysis, interpretation and presentation of data.

Machine learning (ML)

The scientific study of algorithms and statistical models that computer systems use to perform a specific task without using explicit instructions, relying on patterns and inference instead.

Data science

A multi-disciplinary field that uses scientific methods, processes, algorithms and systems to extract knowledge and insights from structured and unstructured data.

Important concepts (2/3)

Deep learning

Also known as deep structured learning or hierarchical learning, is a family of machine learning methods based on artificial neural networks.

Intelligence

The ability to perceive or infer information, and to retain it as knowledge to be applied towards adaptive behaviors within an environment or context.

Artificial Intelligence

The study of *intelligent agents* : any device that perceives it's environment and takes actions that maximize it's chance of successfully achieving it's goals. The term *artificial intelligence* is often used to describe machines (or computers) that mimic "cognitive" functions that humans associate with the human mind, such as *learning* and *problem solving*.

Important concepts (3/3)

Data (Merriam-Webster)

- Factual information (such as measurements or statistics) used as a basis for reasoning, discussion, or calculation.
- Information in digital form that can be transmitted or processed.
- Information output by a sensing device or organ that includes both useful and irrelevant or redundant information and must be processed to be meaningful.

Model (Merriam-Webster)

A system of postulates, data, and inferences presented as a mathematical description of an entity.

Algorithm (Merriam-Webster)

A procedure for solving a mathematical problem in a finite number of steps that frequently involves repetition of an operation.

Machine learning approaches

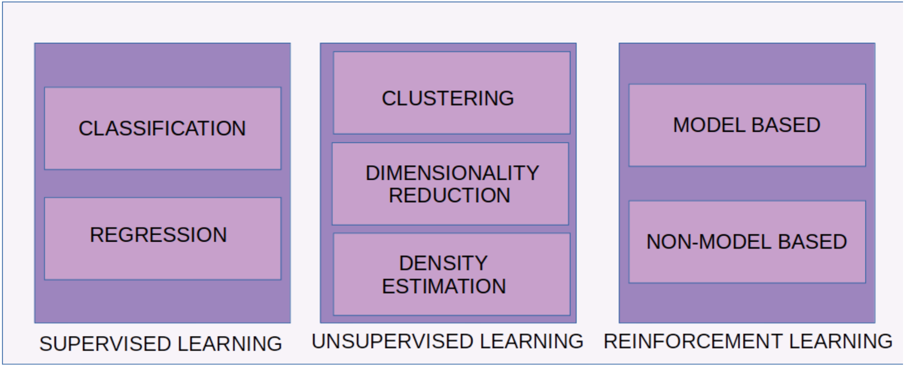
Course objectives

- This course is mostly about using models and data to build systems to solve specific tasks without manually providing explicit instructions
- We will be making intensive use of
  - Data
  - Models
  - Algorithms

Approaches

- Supervised learning
- Unsupervised learning
- Reinforcement learning

Machine learning approaches



MACHINE LEARNING

Supervised learning : classification

What is classification

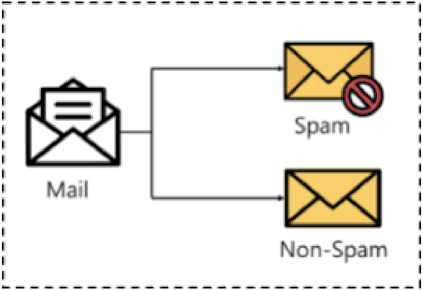
We are given data samples $(\mathbf{x}_n)$ with corresponding labels (y_n) where $y_n \in \{0, 1, ..., C - 1\}$.

- C is called the number of classes
- Binary classification : $C = 2$
- Multi-class problem : $C > 2$
- Multi-label problem : non-mutually exclusive labels

Classification examples

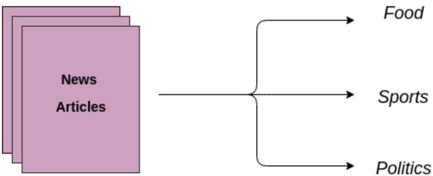
Email classification

- Sample $\mathbf{x}_n$: email words
- Label y_n : spam or not-spam



News article classification

- $\mathbf{x}_n$: article words
- y_n : topic (politics, sport, food)



Supervised learning : classification model learning

Learning formalized as function approximation

- Given samples $(\mathbf{x}_n)$ with corresponding discrete labels (y_n).
- Hypothesis : assume $y = f_{\mathbf{w}}(\mathbf{x})$ for some unknown function $f_{\mathbf{w}}$ parametrized by $\mathbf{w}$
- Learning problem : estimate $\mathbf{w}$ given labeled dataset $(\mathbf{x}_n, y_n)$
- Make reliable predictions $y = f_{\mathbf{w}}(\mathbf{x})$ for new samples $\mathbf{x}$
  - Generalization : ability to reliably predict on new samples

Gradient of $f : \mathbb{R}^D \rightarrow \mathbb{R}$

If $\mathbf{x} = (x_1, x_2, ..., x_D)^t$, the gradient of $f(\mathbf{x}) \in \mathbb{R}$ is the vector constituted of it's partial derivative

$$\nabla_{\mathbf{x}} f(\mathbf{x}) = \left(\frac{\partial f}{\partial x_1}(\mathbf{x}), \frac{\partial f}{\partial x_2}(\mathbf{x}), ..., \frac{\partial f}{\partial x_D}(\mathbf{x}) \right)^t$$

If $\mathbf{x} = (x_1, x_2)^t$, and $f(x_1, x_2) = x_1^2 + x_2^2$ the gradient of $f(x_1, x_2)$ is :

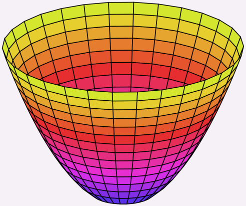
$$\nabla_{\mathbf{x}} f(x_1, x_2) = (2x_1, 2x_2)^t$$

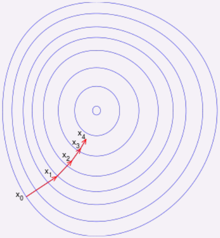
Mathematical reminders : gradient descent

$f : \mathbb{R}^D \rightarrow \mathbb{R}$ a convex function, gradient descent is an algorithm to iteratively estimate the minimum $\hat{\mathbf{x}}$ of f according to the steps :

1. Select an initial value $\hat{\mathbf{x}}_0$, set $t = 0$
2. Update $\hat{\mathbf{x}}_{t+1} = \hat{\mathbf{x}}_t - \lambda \nabla_{\mathbf{x}} f(\hat{\mathbf{x}}_t)$
3. Update $t = t + 1$, and loop back to step 2 until convergence

$f(x_1, x_2) = x_1^2 + x_2^2$





Mathematical reminders : maximum likelihood

Maximum likelihood

Let $(\mathbf{x}_1, ..., \mathbf{x}_N)$ samples independantly drawn according to a probability distribution with density $g_{\mathbf{w}}(\mathbf{x})$ parametrized by $\mathbf{w}$. Samples likelihood is

$$L(\mathbf{w}) = \prod_{n=1}^{N} g_{\mathbf{w}}(\mathbf{x}_n)$$

$\mathbf{w}$ can be estimated as $\hat{\mathbf{w}} = \arg\max_{\mathbf{w}} L(\mathbf{w})$ or $\hat{\mathbf{w}} = \arg\min_{\mathbf{w}} -\log L(\mathbf{w})$.

Normal density $g(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp -\frac{1}{2\sigma^2}(x-\mu)^2$

Parameters are $\mathbf{w} = (\mu, \sigma)$ and samples negative log likelihood is :

$$-\log L(\mu, \sigma) = N\left(\log\sigma + \frac{\log\pi}{2}\right) + \sum_{n=1}^{N} \frac{1}{2\sigma^2}(x_n - \mu)^2$$

A simple solution to classication : logistic regression

Reminder

- Bernouilli : $Y \sim Ber(\alpha)$ if $p(Y = y) = Ber(y;\alpha) = \alpha^y(1-\alpha)^{1-y}$
- Sigmoid function : $\sigma(u) = 1/(1 + \exp(-u))$.

Given samples $(\mathbf{x}_n, y_n)$ with $y_n \in \{0, 1\}$, the logistic regression solution to the classification problem is based on the assumption :

$$p(y_n|\mathbf{x_n}, \mathbf{w}) = Ber(y_n; \sigma(\mathbf{w}^t\mathbf{x}_n))$$

With iid assumptions, $\mathbf{w}$ can be found using negative log-likelihood (loss function) minimization :

$$\hat{\mathbf{w}} = \arg\min_{\mathbf{w}} -\log\left[\prod_{n=1}^{N} \sigma(\mathbf{w}^t\mathbf{x_n})^{y_n}(1-\sigma(\mathbf{w}^t\mathbf{x_n}))^{1-y_n}\right]$$

which is solved using gradient descent.

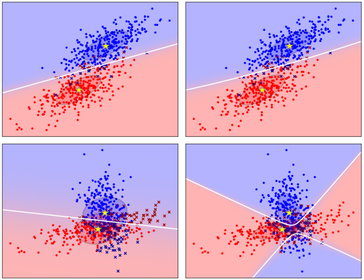
Logistic regression's decision boundary

Linear decision boundaries

- Logistic regression produces linear decision boundaries using $\mathbf{w}^t\mathbf{x_n}$
- Can not addresss problems with non-linear decision boundaries

Non linear decision boundaries

- Build logistic regression on tensorized inputs : $\mathbf{x}_n \otimes \mathbf{x}_n \otimes \ldots \otimes \mathbf{x}_n$
  - Not scalable if $\mathbf{x}_n$ is high dimensional
- More powerful models : random forest, kernel methods, neural nets
  - Will be addressed in the next sessions



Supervised learning : regression

What is regression

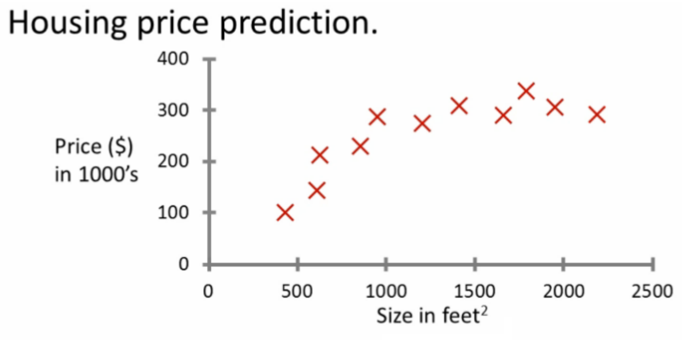
We are given data samples $(\mathbf{x}_n)$ with associated values $(\mathbf{y}_n)$ where $\mathbf{y}_n$ are continuous variable $(\in \mathbb{R}^d)$.

- Similar to classification problem but with continuous outputs

Learning formalized as function approximation

- Hypothesis : assume $\mathbf{y} = f_{\mathbf{w}}(\mathbf{x})$ for some unknown function $f_{\mathbf{w}}$ parametrized by $\mathbf{w}$
- Learning problem : estimate $\mathbf{w}$ given training dataset $(\mathbf{x}_n, \mathbf{y}_n)$
- Make reliable prediction $\mathbf{y} = f_{\mathbf{w}}(\mathbf{x})$ for new samples $\mathbf{x}$

Regression problem example : house price prediction



Linear solution to regression problem

Given samples $(\mathbf{x}_n, \mathbf{y}_n)$ with $y_n \in \mathbb{R}$, the linear regression solution to the regression problem is based on choosing a solution of the form :

$$\mathbf{y} = f_{\mathbf{w}}(\mathbf{y}) = \mathbf{w}^t\mathbf{x} = w_0 + w_1 x_1 + ... + w_D x_D$$

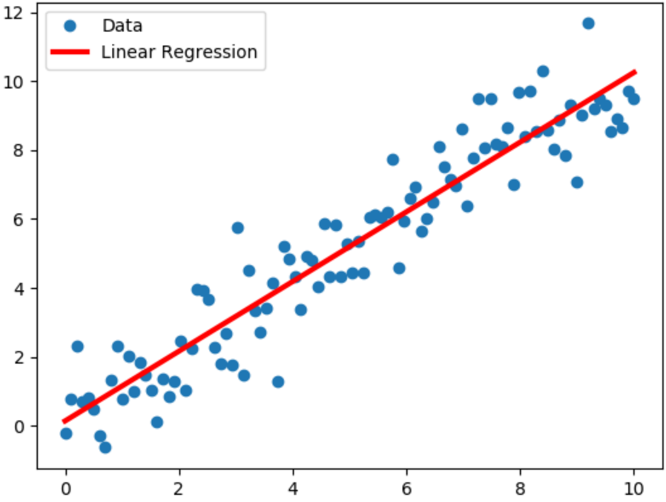
where sample $\mathbf{x}_n$ are augmented with 1 so the first component of $\mathbf{w}$ is the regression bias.

Parameter $\mathbf{w}$ can be estimated by minimizing the sum of square Euclidean distance (loss function) :

$$\hat{\mathbf{w}} = \arg\min_{\mathbf{w}} \sum_{n=1}^{N} (\mathbf{w}^t\mathbf{x}_n - y_n)^2$$

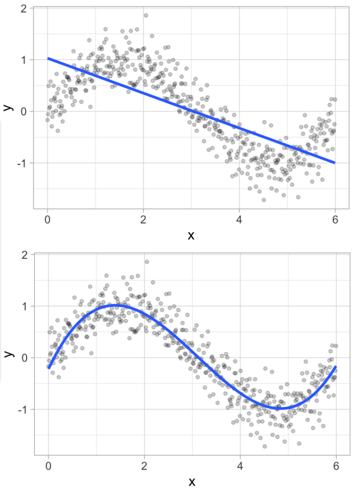
which can be solved by computing the gradient.

Linear solution to regression problem



Addressing non-linear regression problems

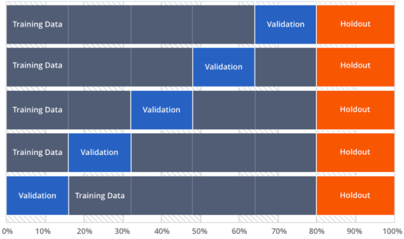
- Build linear regression on tensorized inputs : $\mathbf{x}_n \otimes \mathbf{x}_n \otimes \ldots \otimes \mathbf{x}_n$
    - Not scalable if $\mathbf{x}_n$ is high dimensional
- Use more powerful models :
    - Decision trees/random forest
    - Kernel methods/SVR
    - Neural networks
    - Will be addressed in the next sessions



Supervised learning evaluation : cross validation

- Assessing models generalization abilities :
  - How well models performs on samples outside the training set
- Evaluation protocol : cross-validation
  - Split available data into three subset : training, validation, test
  - Use the training set to fit model parameters :
    - Many parameter configurations are considered
  - Use the validation dataset to select the best parameters
  - Use the test set to assess the best model generalization abilities

Six fold cross-validation



Unsupervised learning

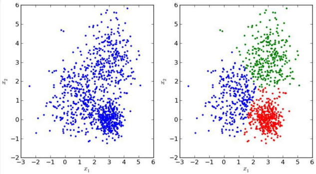
Definition

We are just given sample data $(\mathbf{x}_n)$, without any labels. The goal is to discover latent data structure. Unsupervised learning is also called knowledge discovery

Canonical examples

Given N data samples $\mathbf{x}_1, ...\mathbf{x}_n, ..., \mathbf{x}_N$

- Clustering : cluster samples into K groups
- Density estimation : estimate data density $f_{\mathbf{w}}(\mathbf{x}_n) = p(\mathbf{x}_n|\mathbf{w})$
- Find a lower dimension embedding of the samples $\mathbf{x}_n$



Unsupervised clustering with K-mean

Principle

The goal of the K-means algorithm is to partition a sample set $\{\mathbf{x}_n, n = 1, ..., N\}$ into K sub-sets $\mathbf{S} = \{S_1, S_2, ..., S_K\}$ minimizing the within-class distance (loss) :

$$\hat{\mathbf{S}} = \arg\min_{\mathbf{S}} \sum_{k=1}^{K} \sum_{\mathbf{x}_n \in S_k} ||\mathbf{x}_n - \mu_k||_2$$

where $\mu_k = \frac{1}{|S_k|} \sum_{\mathbf{x}_n \in S_k} \mathbf{x}_n$

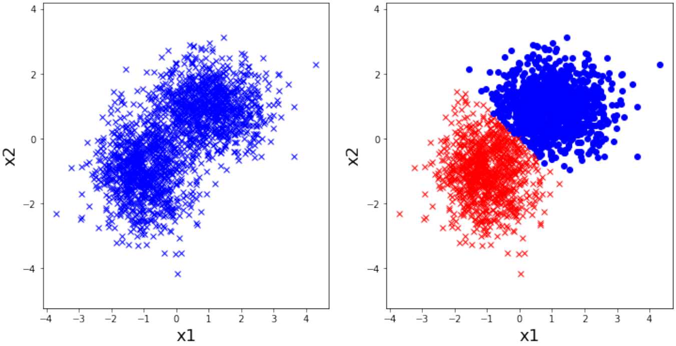
Algorithm

1. Select among the training samples K initial cluster centroid μ_k
2. Select a sample $\mathbf{x}_n$, add it to the cluster of the closest centroid
3. Update the cluster centroid using the new sample $\mathbf{x}_n$
4. loop back to step 2 until convergence

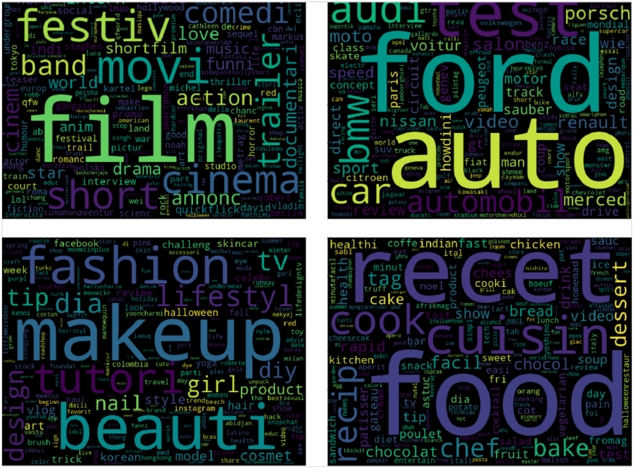
K-means clustering illustration

Experiment protocol

- Generate a cloud of points as a mixture of two Gaussian distribution
- Cluster generated cloud into two classes using k-means algorithm



Clustering channel using counts of words occuring in video metadata.



4 cluster centroids

Unsupervised learning with the E.M. algorithm

Expectation Maximization (E.M.)

- Data is represented as a mixture of a known distribution (e.g. Gaussian, multinomial, Dirichlet)

$$p(\mathbf{x}) = \sum_{k=1}^{K} w_k p_k(\mathbf{x})$$

- E.M. produces a density distribution of the given dataset
- For each sample, E.M. produces an assignment score to each mixture
- E.M. is a generalization of K-means algorithm
  - K-means : hard samples assignment to clusters
  - E.M. : soft samples assignment to clusters

Unsupervised learning of a G.M.M. with E.M.

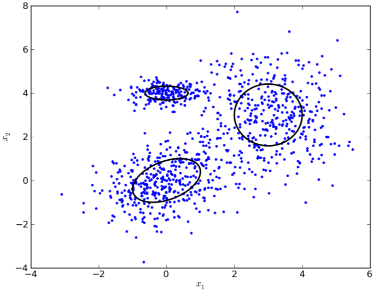
D-dimensional Gaussian density

$$g(\mathbf{x}_n; \mu_k, \Sigma_k) = \frac{1}{\sqrt{(2\pi)^D |\Sigma_k|}} exp - \frac{1}{2}(\mathbf{x}_n - \mu_k)^t \Sigma_k^{-1} (\mathbf{x}_n - \mu_k)$$

Problem

Given a dataset $(\mathbf{x}_n)$, learn using EM a gaussian mixture model (GMM)

$$p(\mathbf{x}_n) = \sum_{k=1}^{K} \alpha_k g(\mathbf{x}_n; \mu_k, \Sigma_k)$$



Learning a GMM with EM algorithm : principle

Latent variables

Introduce assignment variable Z_n such that $p(Z_n = k) = \alpha_k$ is the prior probability sample $\mathbf{x}_n$ being generated by the mixture component k.

Maximum marginal likelihood

Estimate the GMM parameters $\Theta = (\alpha_k, \mu_k, \Sigma_k)$ maximizing likelihood :

$$
\begin{aligned}
p(\mathbf{x}_{1:N}, Z_{1:N}|\Theta) &= \prod_{n=1}^{N} p(\mathbf{x}_n|Z_n)p(Z_n) \\
&= \prod_{n=1}^{N}\prod_{k=1}^{K} [p(\mathbf{x}_n|Z_n = k)p(Z_n = k)]^{\delta_{Z_n}(k)} \\
&= \prod_{n=1}^{N}\prod_{k=1}^{K} [g(\mathbf{x}_n; \mu_k, \Sigma_k)\alpha_k]^{\delta_{Z_n}(k)}
\end{aligned}
$$

Learning a GMM with EM algorithm : the algorithm

Expectation step : workout derivations

Compute expectation of log-marginal likelihood wrt observations :

$$Q(\Theta|\Theta_{t-1}) = \mathbb{E}_{Z_{1:N}|\mathbf{X}_{1:N},\Theta_{t-1}}[\log p(\mathbf{X}_{1:N}, Z_{1:N}|\Theta)]$$

Maximization step : workout derivations

Maximize expectation wrt to parameters :

$$\Theta_t = \arg\max_{\Theta} Q(\Theta|\Theta_{t-1})$$

EM Algorithm :

1. Initialize the GMM parameters Θ_0, set $t = 1$
2. Compute log-marginal likelihood expectation $Q(\Theta|\Theta_{t-1})$
3. Compute Θ_t as the arg-maximum of $Q(\Theta|\Theta_{t-1})$
4. Increment t, and loop back to set 2 until convergence

Unsupervised learning evaluation

- Contrarily to supervised learning, unsupervised learning evaluation is not straightforward
  - There are no associated labels
- Evaluation possible for simulated data
- Evaluation possible on downstream classification tasks where features extracted using unsupervised learning methods are used as input features

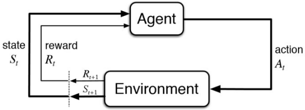
Sutton & Barto's definition

Reinforcement learning for an agent is learning how to map situations to actions so as to maximize a numerical reward signal. The agent must discover by trial and error which actions generate the most rewards by trying them, while rewards may be delayed from actions.

Problem variables

- S_t : environment's state (or observation)
- A_t : agent's action modifying S_t
- R_t : reward resulting from action A_t

R.L. system



Roomba cleaning agent

Reinforcement learning : key concepts

Policy : $A_{t+1} \sim \pi(A|S_t)$

Policy defines agent's way of behaving at a given time.

Expected return : $\mathcal{R} = \sum_{t=0} \gamma^t R_{t+1}$

Agent's goal is to act according to actions' sequence maximizing expected return.

Value function : $V^\pi(s) = \mathbb{E}\left[\sum_{t=0} \gamma^t R_{t+1} | S_0 = s\right]$

Value function for state s is the expected return when starting at state s and following policy π.

Model : $p(S_{t+1}, R_{t+1}|S_t, a_t)$

Represents the environment behaviour given agents actions. When known, model can be used for planning, to gain insights about future actions.

Issues in reinforcement learning

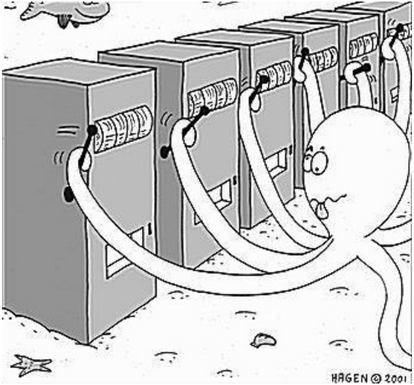
Exploitation vs exploration

- Exploitation : agent always choose the action with the highest immediate reward
  - High risk never to discover actions with high long term rewards
- Exploration : agent choose actions that have not tested to have knowledge about their rewards
  - In large action spaces, exploration is very costly as large number of trials are needed to have information actions rewards

Delayed rewards

- Reward may be received long time after corresponding action occurs
  - Chess : agent receives a reward only for winning or loosing, after a very long sequence of actions

Simple reinforcement learning problem : multi-arm bandits



Simple reinforcement learning problem : multi-arm bandits

- Agent pulls one arm out of K possible : action A_t
- Every time agent pulls an arm, he receives a reward R_{t+1}
- Simplification wrt R.L. : no environment state S_t
- Goal : maximize undiscounted reward after T actions
- Return for an action a_k :

$$Q_t(a_k) = \frac{\sum_{t=1}^{t} R_t \delta_{A_t(a_k)}}{\sum_{t=1}^{t} \delta_{A_t(a_k)}}$$

Greedy policy : agent always choose best arm $A_t = \arg\max_{a_k} Q_t(a_k)$.

- Only exploitation : agent may be stucked on a non optimal choice

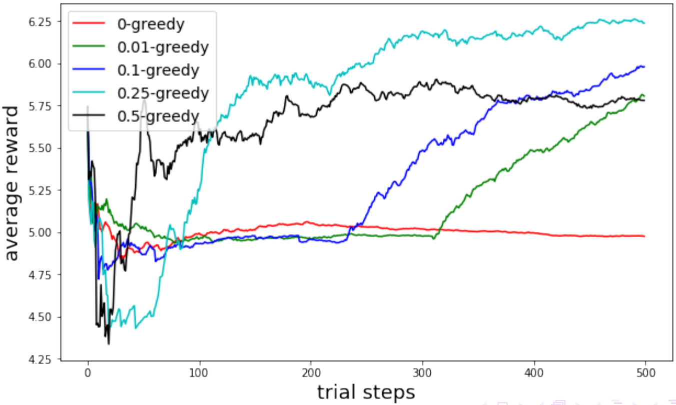
ϵ-greedy policy : agent choose greedily with probability $1 - \epsilon$, and with probability ϵ, uniformly among the K arms.

- Exploration & exploitation : better specially in non stationary cases

Multi-arm bandits : illustration

Experimental setup

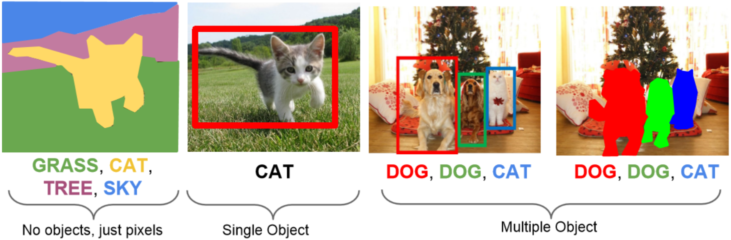
- Agent has the choice among 4 arms during 500 trials
- Expected return for behaviour following epsilon-greedy policies
  - 0-greedy : greedy policy



Applications in :

- Computer vision : related to images and videos
- Audio processing : inference from audio signal
- Natural language processing : related to inference using speech and text
- Recommendation systems : related recommending new items to users
- Robotics : related to physical automatic systems design
- Finance : related to financial information processing
- and many others application domains

Computer vision : detection, recognition,segmentation, etc



- One of the most fruitfuil application domain for ML and AI
- Sensors : any kind of sensors producing images (cameras, depth sensors, radar, x-rays, scanner, ...)
- Applications :
  - Recognition : recognizing image content (people, car, cats, ...)
  - Localization : localizing with a bounding box specific objects in images
  - Segmentation : find all pixels of predifined objects
- Companies : all major companies

Principle

Detect and recognize uttered words

- Sensors : microphones
- Models : beam forming, sound localization, spotting, ASR

Companies

- Google Home : OK Google
- Amazon Echo : Alexa.
- Apple Siri : Hey Siri
- Microsoft Cortana : Cortana

Natural language processing : machine translation

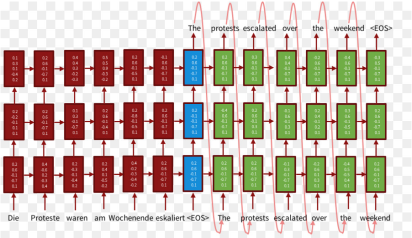
Principle

Translate sequence of words in a language into corresponding sequence in another language

- Inputs : aligned text corpora (e.g. wikipedia)
- Models : statistical machine learning, neural networks

Companies

- Google, Facebook, Microsoft, Baidu, Amazon, Alibaba, etc



Principle

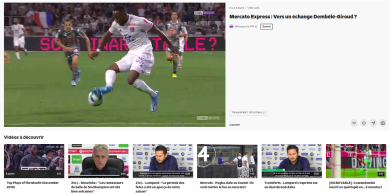
Recommend user items based on what they interact with previously

- Data : Views and cliks
- Models : Matrix factorization, neural networks

Companies : every company selling something online

- Video platforms : Dailymotion, Youtube, etc
- Online retails : Fnac, Darty, Amazon, etc

Dailymotion recommendations



Amazon recommendations

Robotics applications : manufacturing robots

Principle

Design autonomous robot that are able to complete tasks usually completed by humans

- Sensors : camera, lidar, haptics
- Models : computer vision, planning, reinforcement learning

Actors

Amazon, Alibaba, Softbank, Sony, etc

Robotics applications : self-driving cars

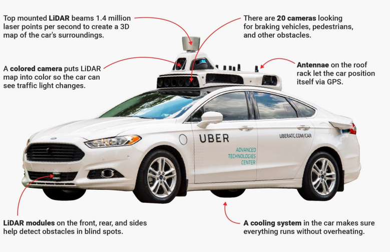
Principle

car exploits data from sensors and machine learning to drive itself

- Sensors : GPS, 360 camera, lidar, accelerometers
- Models : computer vision, reinforcement learning

Companies

Navya-Valeo, Tesla, Uber, Lyft, Waymo (ex Google), Yandex, Ford, Pony-Toyota, General Motor



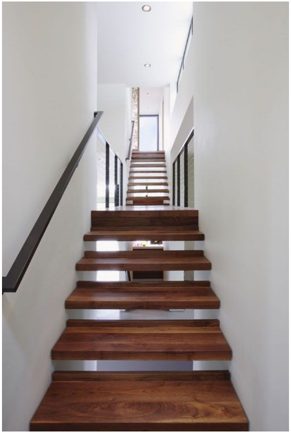
Machine learning in finance

Companies intensively using machine learning in 2018

- Jane Street : trading company
- Jump Trading : trading company
- Hudson River Trading (HRT) : trading comapny
- Ant Financial : Alibaba's Finetech
- JP Morgan-Chase AI : optimize financial services
- Bloomberg AI : financial information network

Machine learning projects in industry : project steps

- Problem statement
- Data processing
  - Annotation, analysis, vizualization, preprocessing, feature extraction
- Model
  - Selection, training, validation
- Deployment
  - Platform, monitoring, alerting
- Output analysis
  - Business, monitoring, ML



ML projects in industry : project statement

- Business problem
  - What to predict, why
- Business value
  - What is the added value for the company
- Business KPIs
  - How to assess business success

ML projects in industry : from POC to industrialization

Proof-of-concept

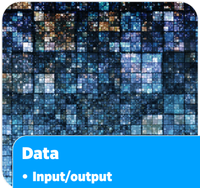
- ML Research problem
- Unknown ML method, uncertain performances
- Run exploratory investigation
- Expected results : certainty on performances

Industrialization

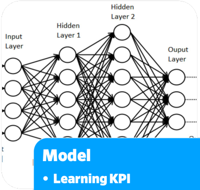
- ML solution known
- Implement engineering solution (data collection, model training, validation)
- Expected results : industrialized system

ML projects in industry : POC lifecycle

Business Problem
- Task elicitation
- Business values
- Business KPI

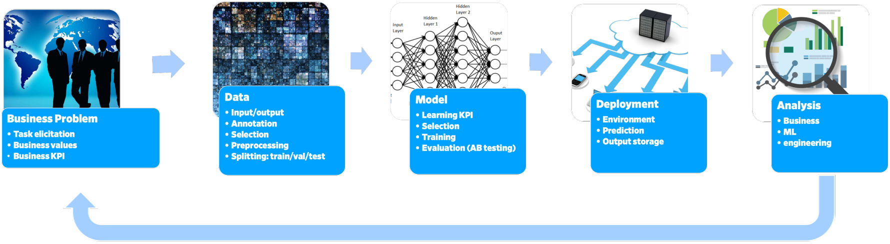


Data
- Input/output
- Annotation
- Exploration
- Selection
- Preprocessing
- Splitting: train/val/test



Model
- Learning KPI
- Selection
- Training
- Evaluation
- Risk assessment

ML projects in industry : Industrialization lifecycle



7

Overview

1. Course organization
   - Lecturers
   - Course organization
2. Introduction to machine learning and AI
   - Glossary : meaning of all these concepts
   - Machine learning approaches
   - Machine learning applications
   - Machine learning projects in industry
3. Some problems faced in ML project
   - Model training cost
   - Bias in datasets
   - Other issues
4. Next session

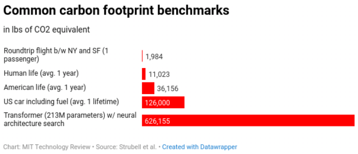
Problems in ML : GPU cost

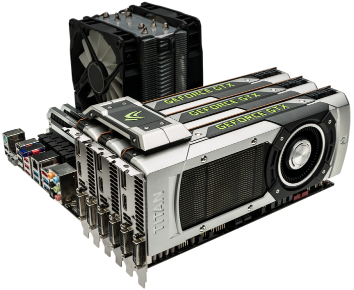
Problem

- XLNet training cost : $245,000
  - 512 TPU v3 chips * 2.5 days * $8/TPU
- High carbon emission foot print
  - DL model : five cars carbon footprint

Solution

- Create/use model you need not the biggest available
- Use pretrained model





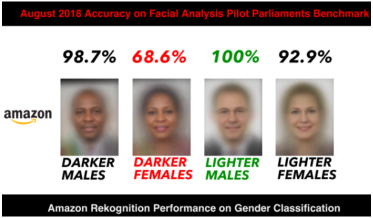
Problems in ML : biased dataset

Problem

- Face detection and dark skin people
- Gender recognition and dark skin women
- Hiring AI and women applications
- Low income aeras and loans
- Problem :
  - Biased datasets
  - Models learn human biases

Solution

- Consciousness about bias
- Design unbiased dataset
- design models robust to bias



Costly data labelling

Supervised learning requires data manually annotated by multiple people using system such as Amazon Mechanical Turk. This is a time consuming and costly.

Recommendation bubble

One views the same type of content because recommendation engine propose to view content similar to what one already viewed, or what people alike have viewed.

Next session

Labs about gradient descent in machine learning

- Loss function minimization with analytical gradient
- Loss function minimization with numeric gradient in Tensorflow/Keras
- Applications to linear and logistic regressions parameters fitting