DDA3020: Machine Learning - Tutorial 1

- Python, Scikit-Learn Basis & Gradient Descent

Xudong Wang

SDS, CUHK(SZ)

September 13, 2022

香港中文大學(深圳)
The Chinese University of Hong Kong, Shenzhen

Course Information

Tutorial:
Venue: TB102 (126 seats), Tuesday 18:00 - 21:00pm from Week 2 -Week 13. S1:18:00-18:50pm(for T02,05), S2: 19:00-19:50pm(for T03,04), S3: 20:00-20:50pm(for T01,06).
All the 3 sessions will give the same tutorial content, and you can select a convenient time to attend. The Zoom link will be given on BB before each tutorial and the tutorial recordings will also be provided.
TAs:

- **Xudong Wang:** xudongwang@link.cuhk.edu.cn
  OH: Tue 4:30-5:30pm, Seat 70, SDS Lab, 4F ZX
- **Dan Qiao:** danqiao@link.cuhk.edu.cn
  OH: Fri 4:00-5:00pm, Seat 1, SDS Lab, 4F, ZX
- **Fei Yu:** feiyu1@link.cuhk.edu.cn
  OH: Thu 3:30-4:30pm, Seat 14, SDS Lab, 4F, ZX
- **Sho Inoue:** shoinoue@link.cuhk.edu.cn
  OH: Tue 5:30-6:30pm, ZOOM: 759 394 8941 (pw123456)
- **UStaff TAs: TBA**

Add/Drop: September 5th (00:00 Monday) to September 16th (23:59 Friday)

Python Checklist

- **Install:** Python Version >= 3.7, Choose a IDE as you like: VSCode, PyCharm, Spyder, **Jupyter Notebook/Lab** (Web-based, interactive)...
- **Basis Knowledge:**
  - Values Types: int, float, str, byte ...
  - Values Assignment and Operations
  - Input/output Operations
  - Control Flow in Python Conditional statements : if ... else ... Loop statements: for ... / while ...
  - Function call: def function:
  - Data Strucure(list, tuple, dict...)
  - Python Classes and Objects: OOD
  - Others: import, comments

Recap the CSC1001 materials or Ref the Python3 Cheat Sheet.

Python Packages for ML

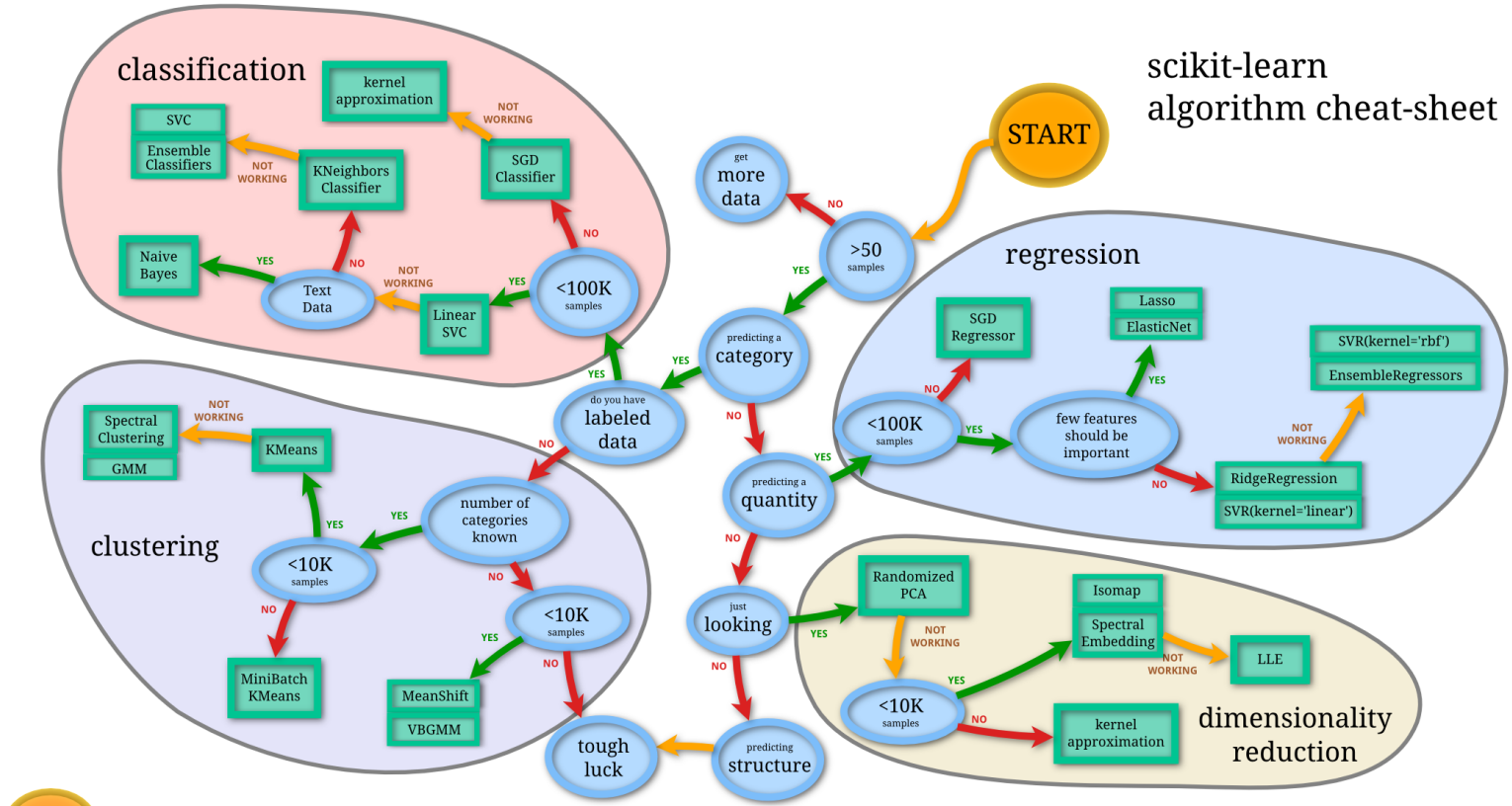
- Numpy + Scipy:
  i) Useful data structure: ndarray, matrix, array
  ii)Algorithms: statistics, scientific computation
- Pandas:
  Processing table–like dataset (such as Excel), DataFrame, Series
- Scikit-learn: Various data mining/machine learning functions
- NLTK: Natural Language Toolkit
- Visualization: Matplotlib, Seaborn...
- Deep Learning: TensorFlow (Google), PyTorch (Facebook), Caffe, Keras, MXNeT (Amazon)

Recap the lecture content, we can express the data in vector/matrix formula: one data sample with d features $\boldsymbol{x_i} \in R^d$, total n sample we can stack a matrix $X = [\boldsymbol{x_1}, \boldsymbol{x_2}, ..., \boldsymbol{x_n}] \in R^{d \times n}$

In this tutorial, we will briefly introduce the **Numpy: ndarray** and **Scikit-learn**.

Scikit-Learn: Various data mining/machine learning functions

Strongly recommend to ref the Scikit-Learn official website:
`https://scikit-Learn.org/stable/user_guide.html`



Numpy & Scikit-Learn

Move to Jupyter Notebook

Gradient

In vector calculus, the gradient of a **scalar-valued differentiable function** f of several variables is the vector field (or vector-valued function) ∇ f whose value at a point $\boldsymbol{x}$ is the vector whose components are the partial derivatives of f at $\boldsymbol{x}$. That is, for:

$$f : R^d \to R$$

its gradient:

$$\nabla f : R^d \to R^d$$

is defined at the point:

$$\boldsymbol{x} = (x_1, x_2, ..., x_d)^T$$

in n-dimensional space as the vector

$$\nabla f(\boldsymbol{x}) = \begin{bmatrix} \frac{\partial f}{\partial x_1}(\boldsymbol{x}) \\ \vdots \\ \frac{\partial f}{\partial x_d}(\boldsymbol{x}) \end{bmatrix}$$

We can find that the $\nabla f(\boldsymbol{x})$ have the same shape of $\boldsymbol{x}$. Actually, it gives a direction.

Gradient Direction: "Steepest" direction to increase function value

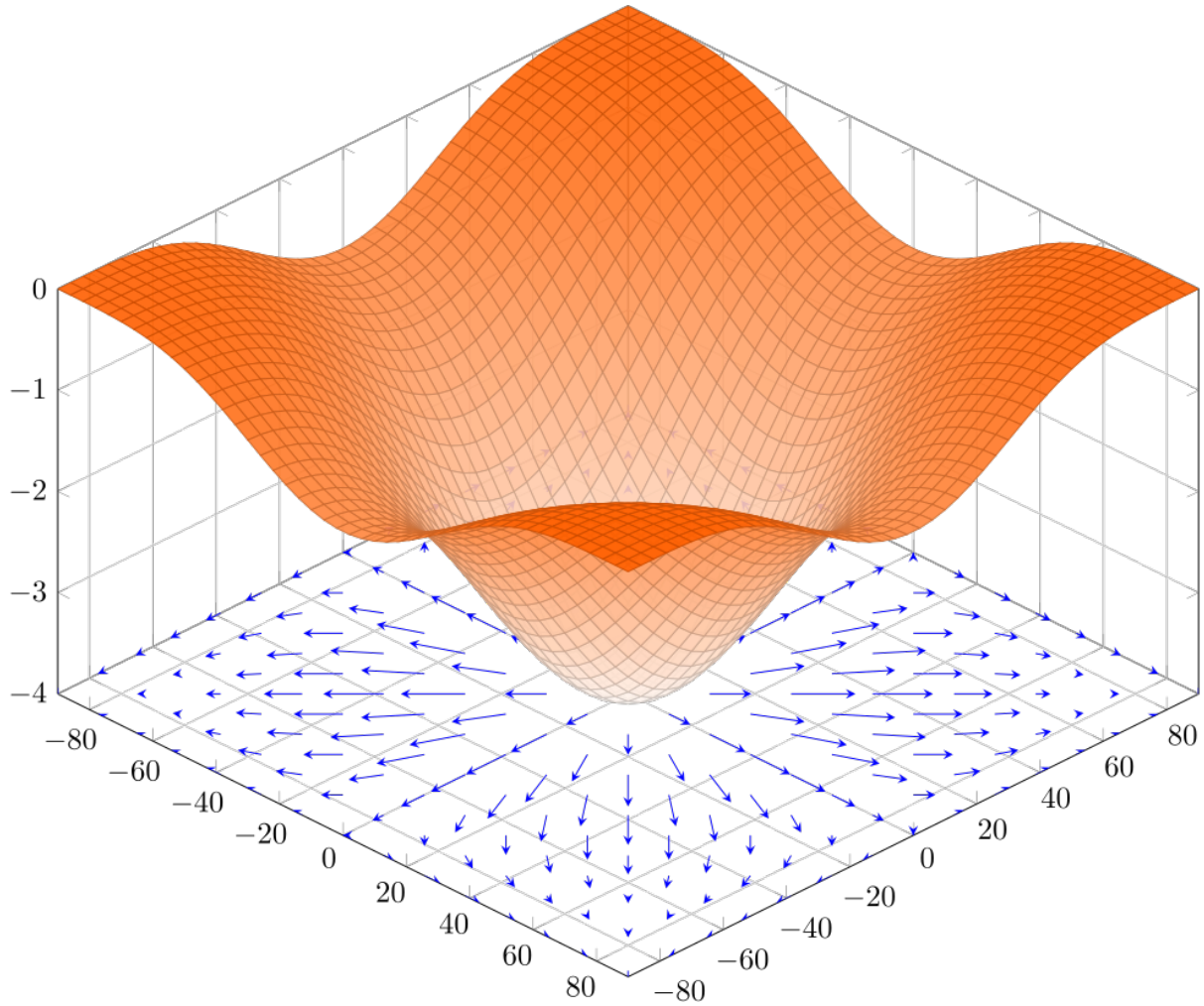


Figure 1: The gradient of the function $f(x, y) = -(cos^2 x + cos^2 y)2$ depicted as a projected vector field on the bottom plane.

Gradient Descent Algorithm

GD is a "fundamental" algorithm in **continuous optimization**, both historically and mathematically. Consider the unconstrained problem:

$$\min_{\theta} f(\theta)$$

where f: $R^d \rightarrow R$ A basic form of the gradient descent algorithm (constant-stepsize) is:

$$\theta_{k+1} = \theta_k - \eta \nabla f\left(\theta_k\right), k = 0, 1, 2, \ldots$$

where η is a constant called "stepsize" (a.k.a. learning rate in machine learning community), and $\nabla f(\cdot)$ denotes the gradient of f.

It means that at each iteration, the iterate will move along the negative gradient direction and how far it moves is controlled by the stepsize η.

GD is an **iterative** method. Using **heuristic** metrics: function value(loss), gradient... to determine the stop criteria.

Summary

- Python Basis
- Numpy ndarray
- Scikit-learn Package Usage
- Gradient Descent (Solve, optimize model parameters)

Thank you for listening!

香港中文大學(深圳)
The Chinese University of Hong Kong, Shenzhen

Xudong Wang
xudongwang@link.cuhk.edu.cn

September 13, 2022