Portfolio Milestone 2: Analysis of Algorithms and Data Structures

Donghwan Kim

Colorado State University Global

CSC 506: Design and Analysis of Algorithms

Dr. Lori Farr

May 30, 2022

Portfolio Milestone 2: Analysis of Algorithms and Data Structures

This Portfolio Milestone 2 assignment implements the five different types of data structures for multi-dimensional lists based on some of Python basic data structures like list, nested list, dictionary, and so on. The multi-dimensional list is important in modern Artificial Intelligence and Machine Learning. Python does not provide the multi-dimensional list but this assignment shows various ways how to construct it with Python built-in data structures.

Data Structures For Multi-dimensional List

In general, a 2-dimensional list is denoted X having $m \times n$ elements where m is the number of rows and n is the number of columns. It is declared with $DIMENSION(m,n)$ in Fortran 90/95 (Wagener 1998). And it is usually expressed as X[i,j] for $i = 1, 2, \cdots, m$ and $j = 1, 2, \cdots, n$. The elements can be accessed with the index [i, j]. Python does not provide the multi-dimensional list explicitly (Van Rossum and Drake Jr 2009; Python Documentation 2022). However, it is possible in several ways. The multi-dimensional list can be constructed with n separate Python lists each of which has m elements. The first data structure named DS1 is as follows:

$$\textbf{DS1}.\ \ X[:,0], X[:,1], X[:,2], \cdots, X[:,n-1]$$

where $X[:,j] = [X[0,j], X[1,j], \cdots, X[m-1,j]]$ for $j = 0, 1, \cdots, n-1$. It is commonly used in Fortran 90/95. Another simple alternative is to contain all the elements in a list

DS2.

$$\begin{aligned}
&[X(0,0), X(1,0), X(2,0), \cdots, X(m-1,0),\\
&\ X(0,1), X(1,1), X(2,1), \cdots, X(m-1,1),\\
&\ X(0,2), X(1,2), X(2,2), \cdots, X(m-1,2),\\
&\ \vdots,\\
&\ X(0,n-1), X(1,n-1), \cdots, X(m-1,n-1)]
\end{aligned}$$

Python nested list can also be used to implement the data structure in two different ways which is shown below. The data structures DS3 and DS4 are possible.

DS3. $[X[:,0], X[:,1], X[:,2], \cdots, X[:,n-1]]$

DS4. $[X[0,:], X[1,:], X[2,:], \cdots, X[m-1,:]]$

where $X[i,:] = X[i,0], X[i,1], \cdots, X[i,n-1]$ for $i = 0, 1, \cdots, m-1$. The data structure DS5 uses dictionaries in a list. Python dictionary is a good data structure for data lookup (Lysecky and Vahid 2019).

DS5.

$$\begin{aligned}
&[\{\text{‘}col0' : X[0,0], \text{‘}col1' : X[0,1], \cdots, \text{‘}colLast' : X[0,n-1]\},\\
&\ \{\text{‘}col0' : X[1,0], \text{‘}col1' : X[1,1], \cdots, \text{‘}colLast' : X[1,n-1]\},\\
&\ \{\text{‘}col0' : X[2,0], \text{‘}col1' : X[2,1], \cdots, \text{‘}colLast' : X[2,n-1]\},\\
&\ \vdots\\
&\ \{\text{‘}col0' : X[m-1,0], \text{‘}col1' : X[m-1,1], \cdots, \text{‘}colLast' : X[m-1,n-1]\}
\end{aligned}$$

Implementing Data Structures

The data structures from DS1 to DS5 discussed above are implemented with Python. The data structure named DS1 can be implemented with five Python list Algorithm (DS2) shows how a multi-dimensional list can be implemented in a list. The data structure DS2 has the length of NObs * NVar in column-major order where NObs is the number of rows (i.e. records or observations) and NVar is the number of columns. Data is created with the Python random module. The first column X[:,0] is stored in the first DS[0:NObs-1]. The second column X[:,1] is stored in the DS[NObs:(NObs*2) - 1].

```
Algorithm (DS2).
DS2 = [None] * NObs * NVar
for i in range(NObs):
    DS2[i] = float(random.randint(0,4))
    DS2[i + NObs] = random.random()*100
    DS2[i + NObs*2] = float(random.randint(0,1))
    DS2[i + NObs*3] = random.random()*50
    DS2[i + NObs*4] = 1 + 3 * DS2[i] + 5 * DS2[i + NObs] -
                      6 * DS2[i + NObs*2] +
                      9 * DS2[i + NObs*3] + random.random()
```

The data structure DS3 is implemented in Python using Algorithm (DS3) which uses a Python nested list. Data can be accessed with DS3[j][i]. That is, the first nested list in DS3 contains the first column of X. Unlike the DS3, the data structure DS4 stores data row-by-row (Lutz 2013). The data structure DS5 uses dictionaries to store records.

```
Algorithm (DS3).
DS3 = [[None]*NObs for _ in range(NVar)]
for i in range(NObs):
     DS3[0][i] = float(random.randint(0,4))
```

```
    DS3[1][i] = random.random()*100
    DS3[2][i] = float(random.randint(0,1))
    DS3[3][i] = random.random()*50
    DS3[4][i] = 1 + 3 * DS3[0][i] + 5 * DS3[1][i] -
                6 * DS3[2][i] +
                9 * DS3[3][i] + random.random()
```

The five data structures from DS1 to DS5 are implemented and tested with Python. Since it uses random number generators, the values in the data of each of the structures are different. But all of the data structures are two-dimensional lists where the first column contains a string variable, and the third one contains a binary variable. The final project tests the performance of the five data structures with some of the operations.

Conclusions

Multi-dimensional lists are essential in data-centered Artificial Intelligence and Machine Learning where time efficiency matters. In this Portfolio Milestone 2, five data structures for the multi-dimensional list are implemented with Python basic data structures list, nest list, and a mix of list and dictionary. The data structures are discussed and tested. Further performance test will be followed in the final project with some operations to the data structures.

References

Lysecky, R. and Vahid, F. (2019). Design and analysis of algorithms. In Lysecky, R. and Vahid, F., editors, *Data structures essential: Pseudocode with python examples.* Zybooks.

Lutz, M. (2013). *Learning Python.* O'Reilly Media.

Van Rossum, G. and Drake Jr, F. L. (2009). *Python tutorial.* Scotts Valley, CA: CreateSpace.

Wagener, J. (1998). *Fortran 90 concise reference.* Absoft Corporation.

Python Documentation (2022). Python 3.10.4 documentation: The tutorial. `https://docs.python.org/3/`.