Arrays

MATH1900: Machine Learning

Location: http://people.sc.fsu.edu/~jburkardt/classes/ml_2019/arrays/arrays.pdf

---

How to tame a pack of numbers

What we call a single data item might, in fact, involve a set of several numbers, such as geographic longitude and latitude (2 values), tomorrow's hourly temperature forecast (24 values), or a student's calculus quiz scores (15 values). We might call such items a list, or in linear algebra a vector.

Some times data is in tabular form, as when a calculus class has 45 students, and so the quiz gradebook contains 45×15 values. Such data is often called a table, or in linear algebra, a matrix.

The Array Problem

How can we use Python arrays to:

- *create a space where we can store our data;*
- *easily locate, set, or change a specific value in our data;*
- *carry out mathematical operations on our data.*

When computing in Python, vectors and matrices and more complicated objects are all referred to as **Numpy arrays**. An array allows us to store and retrieve the values that constitute an item of data. it is also possible to carry out mathematical and linear algebra operations that are applied to the entire array, rather than having to consider each numeric entry separately. This means, for instance, that the calculus instructor could take compute the quiz average, or add 10 points to everyone's grade, using a single command, rather than 675 commands.

```
import numpy as np
quiz = np.zeros( [3,2] )
print ( quiz )
quiz[0,0] = 81
quiz[0,1] = 62
print ( quiz )
quiz[1,0] = 55
quiz[1,1] = 68
```

```
  quiz[2,0] = 100
  quiz[2,1] = 93
  print ( quiz )
```

We can enter all the data at once using the `np.array()` function:

```
  import numpy as np
  quiz = np.array(
  [ [  81, 62 ],
    [  55, 68 ],
    [ 100, 93 ] ] )
  print ( quiz )
```

Now let's subtract 10 points from the quiz grade of 55. Then let's add 5 points to all the second quiz grades. Finally, compute the overall average, the quiz average, and the student average:

```
import numpy as np
quiz = np.array(
[ [  81, 62 ],
  [  55, 68 ],
  [ 100, 93 ] ] )
print ( quiz )
quiz[1,0] = quiz[1,0] - 10
quiz[:,1] = quiz[:,1] + 5
print ( quiz )
overall_average = np.mean ( quiz )
print ( overall_average )
quiz_average = np.mean ( quiz, axis = 0 )
print ( quiz_average )
student_average = np.mean ( quiz, axis = 1 )
print ( student_average )
```

Other functions that might be useful for this quiz example include `np.min()`, `np.max()`, `np.sum()`, `np.std()` (standard deviation), and `np.var()` (variance). These commands also operate on all the numbers in the array, or can be told to operate by columns (`axis = 0`) or rows (`axis = 1`).

1 Vector Geometry

In linear algebra, a vector is a list of numbers, which often represent the coordinates of a point in a plane $\vec{v} = (x, y)$ or in space $\vec{v} = (x, y, z)$. By analogy, we can imagine a longer list of numbers as a point in a higher dimensional space.

In linear algebra, a vector has a length, the square root of the sum of the squares of its n entries.

$$||\vec{x}|| = \sum_{i=1}^{n} x_i^2$$

Numpy computes the norm by:

```
import numpy as np
x = np.array ( [3, 4] )
l = np.linalg.norm ( x )
y = np.array ( [ 1, 2, 3 ] )
l = np.linalg.norm ( y )
```

We say a vector $\vec{x}$ is *normalized* or has *unit norm* if $||\vec{x}|| = 1$.

Given any nonzero vector $\vec{x}$, we can create a normalized version $\vec{u}$ by dividing by the norm:

$$\vec{u} = \frac{\vec{x}}{||\vec{x}||}$$

which in Python would be

```python
import numpy as np
u = x / np.linalg.norm ( x )
```

In linear algebra, given two vectors $\vec{x}$ and $\vec{y}$ which both have n entries, the dot product is defined by

$$\vec{x} \cdot \vec{y} = \sum_{i=1}^{n} x_i y_i$$

In Python, we can compute the dot product easily:

```python
import numpy as np
x = np.array ( [ 1, 2, 3 ] )
y = np.array ( [ -3, 2, 1 ] )
d = np.dot ( x, y )
```

In linear algebra, given two vectors $\vec{x}$ and $\vec{y}$, the dot product has another representation involving α, the **angle** between the vectors:

$$\vec{x} \cdot \vec{y} = ||\vec{x}|| \, ||\vec{y}|| \cos(\alpha)$$

In particular, if $\vec{x} \cdot \vec{y} = 0$, the vectors are perpendicular or “orthogonal”.

Suppose $\vec{u}$ is a unit vector, and $\hat{x}$ is any vector. Then compute

$$\begin{aligned} \alpha &= \vec{u} \cdot \vec{x} \\ \vec{y} &= \vec{x} - \alpha * \vec{u} \\ \vec{x} &= \alpha * \vec{u} + \vec{y} \end{aligned}$$

Here, we have decomposed $\vec{x}$ into a part parallel to $\vec{u}$ and a part which is perpendicular. The component $\alpha * \vec{u}$ is called the **projection of $\vec{x}$ onto $\vec{u}$**.