Solving Systems of Linear Equations with NumPy - Lab

Introduction

Now you've gathered all the required skills needed to solve systems of linear equations. You saw why there was a need to calculate inverses of matrices, followed by matrix multiplication to figure out the values of unknown variables.

The exercises in this lab present some problems that can be converted into a system of linear equations.

Objectives

You will be able to:

- Use matrix algebra and NumPy to solve a system of linear equations given a real-life example
- Use NumPy's linear algebra solver to solve for systems of linear equations

Exercise 1

A coffee shop is having a sale on coffee and tea.

On day 1, 29 bags of coffee and 41 bags of tea were sold, for a total of 490 dollars.

On day 2, they sold 23 bags of coffee and 41 bags of tea, for which customers paid a total of 448 dollars.

How much does each bag cost?

```
# Create and solve the relevant system of equations

# Let x be the price of a bag of coffee and y be the price of a bag of tea.

# 29x + 41y = 490

# 23x + 41y = 448

#  Create numpy matrices from above equations
import numpy as np
A = np.matrix([[29, 41], [23, 41]])
B = np.matrix([[490, 448]])

# Calculate inverse of A and take the dot product
A_inv = np.linalg.inv(A)
X = A_inv.dot(B.T)
print(X)

# Verify the answer linalg.solve()
np.linalg.solve(A, B.T)
```

```
[[7.]
 [7.]]
```

```
matrix([[7.],
        [7.]])
```

```
# Describe your result
# bag of coffee = $7 , bag of tea = $7
```

Exercise 2

The cost of admission to a popular music concert was 162 dollars for 12 children and 3 adults.

The admission was 122 dollars for 8 children and 3 adults in the same music concert.

How much was the admission for each child and adult?

```
# Create and solve the relevant system of equations

# Let x be the price per child and y be the price per adult

# 12x + 3y = 162
#
# 8x + 3y = 122

# Create matrices in numpy
A = np.matrix([[12, 3],[8, 3]])
B = np.matrix([162, 122])

# Calculate inverse of A and take the dot product
A_inv = np.linalg.inv(A)
X = A_inv.dot(B.T)
print (X)

# Verify the answer linalg.solve()
np.linalg.solve(A, B.T)
```

```
[[10.]
 [14.]]
```

```
matrix([[10.],
        [14.]])
```

```
# Describe your result
# price per child = $10, price per adult = $14
```

Exercise 3

You want to make a soup containing tomatoes, carrots, and onions.

Suppose you don't know the exact mix to put in, but you know there are 7 individual pieces of vegetables, and there are twice as many tomatoes as onions, and that the 7 pieces of vegetables cost 5.25 USD in total. You also know that onions cost 0.5 USD each, tomatoes cost 0.75 USD and carrots cost 1.25 USD each.

Create a system of equations to find out exactly how many of each of the vegetables are in your soup.

```
# Create and solve the relevant system of equations

# Let o represent onions, t - tomatoes and c - carrots.  p--> c .   b--> o, 0---> t

# t + c + o = 7

# .5o + .75t + 1.25c = 5.25

#  t  = 2o which is equal to: -2o + t + 0c = 0

# Create matrices in numpy
A = np.matrix([[1,1,1],[0.5, 0.75, 1.25], [-2,1,0]])
B = np.matrix([[7, 5.25, 0]])

# Calculate inverse of A and take the dot product
A_inv = np.linalg.inv(A)
X = A_inv.dot(B.T)
print (X)

# Verify the answer linalg.solve()
np.linalg.solve(A,B.T)
```

```
[[2.]
 [4.]
```

```
 [1.]]
```

```
matrix([[2.],
        [4.],
        [1.]])
```

```
# Describe your result
# onions = 2, tomatoes = 4, carrots = 1 , needed to make the soup
```

Exercise 4

A landlord owns 3 properties: a 1-bedroom, a 2-bedroom, and a 3-bedroom house.

The total rent he receives is 1240 USD.

He needs to make some repairs, where those repairs cost 10% of the 1-bedroom house's rent. The 2-bedroom repairs cost 20% of the 2-bedroom rental price and 30% of the 3-bedroom house's rent for its repairs. The total repair bill for all three houses was 276 USD.

The 3-bedroom house's rent is twice the 1-bedroom house's rent.

How much is the individual rent for three houses?

```
# Create and solve the relevant system of equations

# Let x,y,z represent rent value for house 1,2 and 3 respectively

# x + y + z = 1240

# .1x + .2y + .3z = 276

# 2x +0y -z = 0

# Create matrices in numpy
A = np.matrix([[1, 1, 1],[0.1, 0.2, 0.3], [2, 0, -1]])
B = np.matrix([[1240, 276, 0]])

# Calculate inverse of A and take the dot product
A_inv = np.linalg.inv(A)
X = A_inv.dot(B.T)
print (X)
```

```
# Verify the answer linalg.solve()
np.linalg.solve(A, B.T)
```

```
[[280.]
 [400.]
 [560.]]
```

```
matrix([[280.],
        [400.],
        [560.]])
```

```
# Describe your result
# Rent: house1 = 280, house2 = 400, house3 = 560
```

Summary

In this lab, you learned how to use NumPy to solve linear equations by taking inverses and matrix multiplication and also using numpy's `solve()` function. You'll now take these skills forward and see how you can define a simple regression problem using linear algebra and solve it with Numpy.

Releases

No releases published

Packages

No packages published

Contributors 3


ShakeelRaja Shakeel Raja


LoreDirick Lore Dirick

sumedh10 Sumedh Panchadhar

Languages

- **Jupyter Notebook** 100.0%