Statistical Natural Language Processing

Python Refresher II

April 25, 2018

Outline

NumPy

```
>>> import numpy as np
>>> np.array([0, 1, 2])
array([0, 1, 2])
>>> l = [1, 2, 3]
>>> np.array(l)
array([1, 2, 3])
>>> np.array((1, 2, 3))
array([1, 2, 3])
>>> a = np.array([[0, 1, 2], [3, 4, 5]])
>>> a
array([[0, 1, 2],
       [3, 4, 5]])
>>> 'shape: {}, ndim: {}, size: {}, dtype: {}' \
... .format(a.shape, a.ndim, a.size, a.dtype)
'shape: (2, 3), ndim: 2, size: 6, dtype: int32'
>>> np.array([1.0]).dtype
dtype('float64')
>>> np.array([0, 1, 2], dtype=np.float32).dtype
dtype('float32')
```

NumPy

```
>>> import numpy as np
>>> np.arange(4)
array([0, 1, 2, 3])
>>> np.arange(1, 20, 3)
array([ 1,  4,  7, 10, 13, 16, 19])
>>> # 5 numbers from 0 to 2 (0 <= x <= 2)
... np.linspace(0, 2, 5)
array([0. , 0.5, 1. , 1.5, 2. ])
>>> a = np.arange(10)
>>> a[2]
2
>>> a[1:10:2]
array([1, 3, 5, 7, 9])
>>> np.append([1, 2, 3], [4, 5, 6])
array([1, 2, 3, 4, 5, 6])
```

NumPy

```
>>> import numpy as np
>>> np.zeros(5)
array([0., 0., 0., 0., 0.])
>>> np.zeros([2, 5], dtype=np.float16)
array([[0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0.]], dtype=float16)
>>> np.ones(5)
array([1., 1., 1., 1., 1.])
>>> np.eye(4)
array([[1., 0., 0., 0.],
       [0., 1., 0., 0.],
       [0., 0., 1., 0.],
       [0., 0., 0., 1.]])
>>> np.eye(1, M=6, k=4)
array([[0., 0., 0., 0., 1., 0.]])
```

NumPy

```
>>> import numpy as np
>>> # random initial content that depends
... # on the state of the memory
... np.empty(2)
array([2.12199579e-314, 6.36598737e-314])
>>> np.full((2, 5), 11)
array([[11, 11, 11, 11, 11],
       [11, 11, 11, 11, 11]])
>>> # floats randomly chosen from a uniform distribution
... # (0 <= x < 1)
... # np.random also contains other distributions
... np.random.random([2, 2])
array([[0.22566579, 0.24011842],
       [0.81133933, 0.49157668]])
```

NumPy

```
>>> import numpy as np
>>> np.arange(12)
array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11])
>>> np.arange(12).reshape(2, 6)
array([[ 0,  1,  2,  3,  4,  5],
       [ 6,  7,  8,  9, 10, 11]])
>>> a = np.arange(24).reshape(2, 3, 4)
array([[[ 0,  1,  2,  3],
        [ 4,  5,  6,  7],
        [ 8,  9, 10, 11]],

       [[12, 13, 14, 15],
        [16, 17, 18, 19],
        [20, 21, 22, 23]]])
```

NumPy

```
>>> import numpy as np
>>> a = np.arange(12).reshape(2, 6)
>>> a
array([[ 0,  1,  2,  3,  4,  5],
       [ 6,  7,  8,  9, 10, 11]])
>>> a.ravel()
array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11])
>>> a.T
array([[ 0,  6],
       [ 1,  7],
       [ 2,  8],
       [ 3,  9],
       [ 4, 10],
       [ 5, 11]])
```

NumPy

```
>>> import numpy as np
>>> a = np.arange(4)
>>> a
array([0, 1, 2, 3])
>>> a * 3
array([0, 3, 6, 9])
>>> a ** 2
array([0, 1, 4, 9], dtype=int32)
>>> a < 3
array([ True,  True,  True, False])
>>> np.sqrt(a)
array([0.        , 1.        , 1.41421356, 1.73205081])
```

NumPy

```
>>> import numpy as np
>>> # dot product
... np.dot(np.array([0, 1, 2]), np.array([3, 4, 5]))
14
>>> np.array([0, 1, 2]) @ np.array([3, 4, 5])
14
>>> # matrix multiplication
... a = np.arange(6).reshape((3, 2))
>>> b = np.arange(6).reshape((2, 3))
>>> a @ b
array([[ 3,  4,  5],
       [ 9, 14, 19],
       [15, 24, 33]])
>>> # L2-norm
... np.linalg.norm(np.arange(5), ord=2)
5.477225575051661
```

Matplotlib

```
import matplotlib.pyplot as plt
import numpy as np

x = np.linspace(-1, 1, 201)
y = x ** 3

fig, ax = plt.subplots()
ax.plot(x, y)
ax.set(xlabel='x', ylabel='y', title='x cubed')
ax.grid()

plt.show()
```

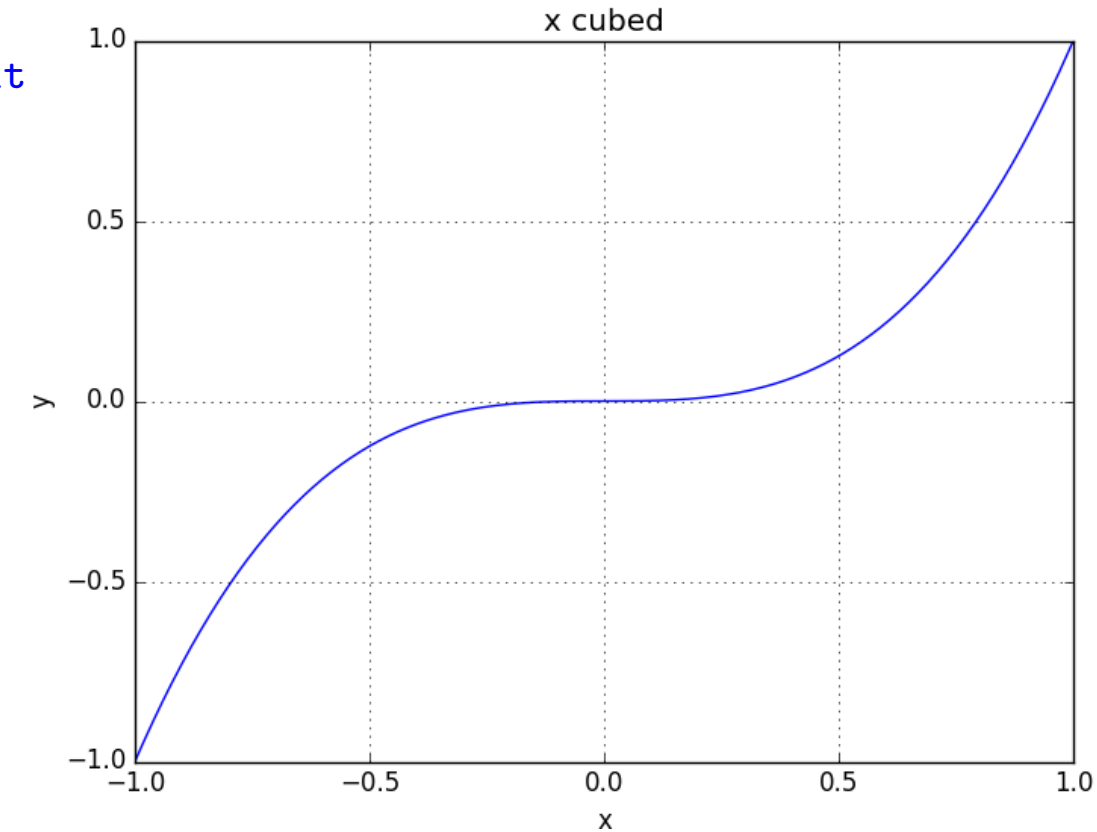
Matplotlib

```
import matplotlib.pyplot as plt
import numpy as np

x = np.linspace(-1, 1, 201)
y = x ** 3

fig, ax = plt.subplots()
ax.plot(x, y)
ax.set(xlabel='x',
       ylabel='y',
       title='x cubed')
ax.grid()

plt.show()
```



Matplotlib

```
import matplotlib.pyplot as plt
import scipy.stats

mu = 10
sigma = 5
n_samples = 5000
# Sample from a normal distribution
normal_distrib = scipy.stats.norm(mu, sigma)
random_sample = normal_distrib.rvs(size=n_samples)
x = range(mu - 4 * sigma, mu + 4 * sigma)
y = normal_distrib.pdf(x)  # and get its PDF.

# Plot the histogram belonging to the sample
plt.hist(random_sample, bins=50, normed=True)
plt.plot(x, y, color='red', linewidth=2)  # and the PDF
plt.axvline(x=mu, color='red')  # plus some lines.
plt.axvline(x=mu - sigma, color='green', linestyle='--')
plt.title('{} samples from N({}, {})'
          .format(n_samples, mu, sigma ** 2))
```

Matplotlib

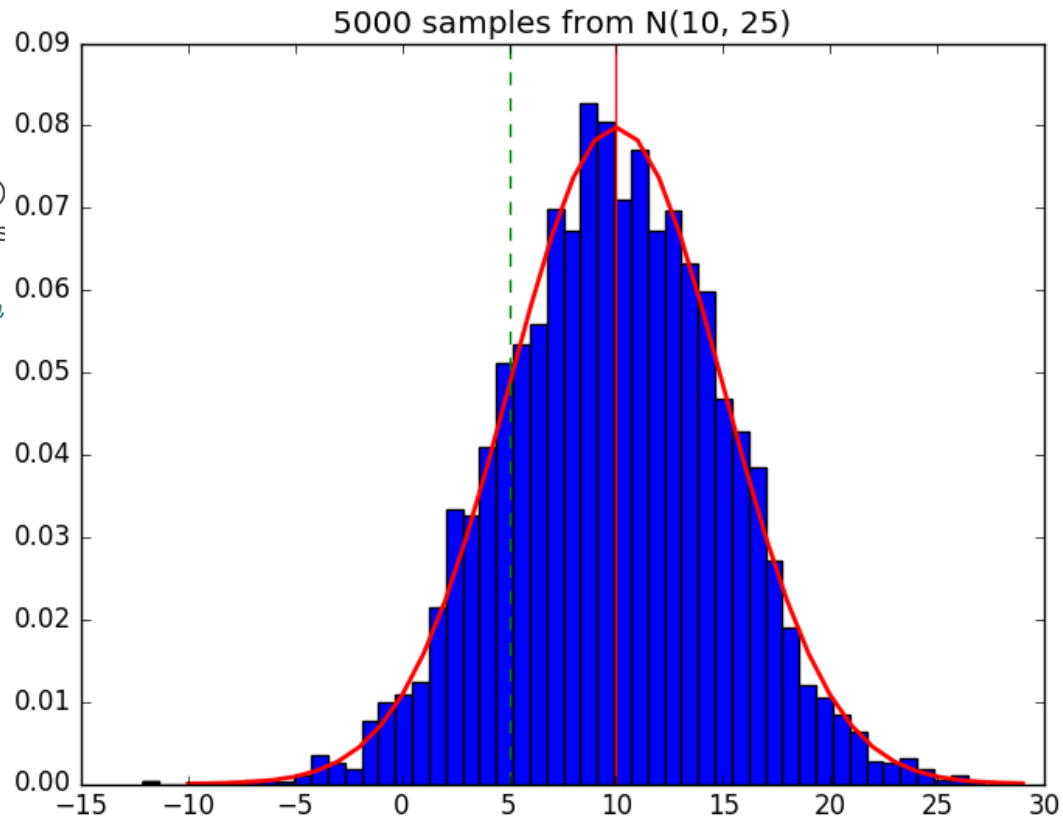
```
import matplotlib.pyplot as plt
import scipy.stats

mu = 10
sigma = 5
n_samples = 5000

norm_distrib = scipy.stats.norm(mu, sigma)
r_sample = norm_distrib.rvs(size=n_samples
x = range(mu - 4 * sigma, mu + 4 * sigma)
# and get its probability density function
y = norm_distrib.pdf(x)

# Plot the histogram
plt.hist(r_sample, bins=50, normed=True)
# and the PDF
plt.plot(x, y, color='red', linewidth=2)
# plus some lines.
plt.axvline(x=mu, color='red')
plt.axvline(x=mu - sigma, color='green',
            linestyle='--')
plt.title('{} samples from N({}, {})'
          .format(n_samples, mu,
                  sigma ** 2))

plt.show()
```



Links

NumPy documentation

https://docs.scipy.org/doc/numpy/user/quickstart.html

https://docs.scipy.org/doc/numpy/user/index.html

Matplotlib documentation

https://matplotlib.org/contents.html

https://matplotlib.org/gallery/index.html

https://matplotlib.org/tutorials/index.html