Uzakliklar, Norm, Benzerlik

Literaturdeki anlatim norm ve uzaklik konusu etrafinda biraz kafa karisikligi yaratabiliyor, bu yazida biraz aciklik getirmeye calisalim. Norm bir buyukluk olcusudur. Vektor uzaylari ile olan alakasini gormek icin *Fonksiyonel Analiz* notlarina bakilabilir. Buyukluk derken bir x vektorunun buyuklugunden bahsediyoruz, ki bu cogunlukla $||x||$ gibi bir kullanimda gorulur, eger altsimge yok ise, o zaman 2 kabul edilir, yani $||x||_2$. Bu ifade bir L2 norm'unu ifade eder. $||x||_1$ varsa L1 norm'u olurdu.

L1,L2 normalari, ya da genel olarak p uzerinden L_p normlari soyle gosterilir

$$||x||_p = (\sum_i |x_i|^p)^{1/p}$$

ki x_i, x vektoru icindeki ogelerdir. Eger $p = 2$ ise, L2 norm

$$||x||_2 = \left(\sum_i |x_i|^2 \right)^{1/2}$$

Ustel olarak 1/2'nin karekok demek oldugunu hatirlayalim, yani

$$||x||_2 = \sqrt{\sum |x_i|^2}$$

Bu norm ayrica Oklitsel (Euclidian) norm olarak ta bilinir, tabii ki bunun Oklitsel uzaklik ile yakin baglantisi var (iki vektoru birbirinden cikartip Oklit normunu alirsak Oklit uzakligini hesaplamis oluruz).

Eger $p = 1$ olsaydi, yani L1 norm, o zaman ustel olarak 1/1 olur, yani hicbir ustel / koksel islem yapilmasina gerek yoktur, iptal olurlar,

$$||x||_1 = \sum |x_i|^2$$

Ornek

$$a = \begin{bmatrix} 3 \\ -2 \\ 1 \end{bmatrix}$$

$$||a|| = \sqrt{3^2 + (-2)^2 + 1^2} = 3.742$$

Ornekte altsimge yok, demek ki L2 norm.

Ek Notasyon, Islemler

L1 normu icin yapilan islemi dusunelim, vektor ogeleri kendileri ile carpiliyor ve sonuclar toplaniyor. Bu islem

$||x||_1 = x^T x$

olarak ta gosterilemez mi? Ya da $x \cdot x$ olarak ki bu noktasal carpimdir.

Bazen de yapay ogrenim literaturunde $||x||^2$ sekilde bir kullanim gorebiliyorsunuz. Burada neler oluyor? Altsimge yok, demek ki L2 norm. Sonra L2 normun karesi alinmis, fakat L2 normu tanimina gore bir karekok almiyor muydu? Evet, fakat o zaman kare islemi karekoku iptal eder, demek ki L2 normunun karesini almak bizi L1 normuna dondurur! Eh bu normu da $x^T x$ olarak hesaplayabildigimize gore hemen o notasyona gecebiliriz, demek ki $||x||^2 = x^T x = x \cdot x$.

Ikisel Vektorlerde Benzerlik

Diger ilginc bir kullanim ikisel degerler iceren iki vektor arasinda cakisan 1 degerlerinin toplamini bulmak. Mesela

```
a = np.array([1,0,0,1,0,0,1,1])
b = np.array([0,0,1,1,0,1,1,0])
```

Bu iki vektor arasindaki 1 uyusumunu bulmak icin noktasal carpim yeterli, cunku 1 ve 0, 0 ve 1, 0 ve 0 carpimi sifir verir, ama 1 carpi 1 = 1 sonucunu verir. O zaman L1 norm bize ikisel iki vektor arasinda kabaca bir benzerlik fikri verebilir.

```
print np.dot(a,b)
```

```
2
```

Ortalama Cikartmak

Scipy seyrek matrislerde ortalamayi almak kulfetli olabiliyor, Scipy ortalamayi cikartmayi izin vermez (olmayan degerler ortalama alirken sifir mi kabul edilecektir? bu tam bilinmedigi icin izin verilmemis). Fakat bu ozellik gerekiyorsa, soyle yapilir,

```
import scipy.sparse as sps

def center(mat):
    mat = mat.T
    vec = sps.csc_matrix(mat.mean(axis=1))
    mat_row = mat.tocsr()
    vec_row = vec.T
    mat_row.data -= np.repeat(vec_row.toarray()[0],np.diff(mat_row.indptr))
    return mat_row.T

mat = sps.csc_matrix([[1, 2, 3, 5.],
                      [2, 3, 4, 5.],
                      [3, 4, 5, 5.]])

print center(mat).todense()
```

```
[[-1. -1. -1.  0.]
 [ 0.  0.  0.  0.]
 [ 1.  1.  1.  0.]]
```

Normalize Etmek

Standardize etmek hem ortalamayi cikartmak (demean), sonra normalize etmek demektir. Bu iki islem birbirinden bagimsiz yapilabilir, bazen biri bazen digeri kullanilabilir. Normalize etmek icin scikit-learn paketinin fonksiyonlari vardir,

```
from sklearn.preprocessing import normalize
print normalize(mat, norm='l1', axis=0).todense()
```

```
[[-0.5 -0.5 -0.5  0. ]
 [ 0.   0.   0.   0. ]
 [ 0.5  0.5  0.5  0. ]]
```

Ustteki cagri matrisin kolonlarini (cunku `axis=0` secildi, bu kolon demek) L1 normu kullanarak normalize etti; yani her kolonun L1 buyuklugu hesaplandi ve o kolonun her hucresi bu buyukluk ile bolundu. L2 norm kullabilirdik,

```
print normalize(mat, norm='l2', axis=0).todense()
```

```
[[-0.70710678 -0.70710678 -0.70710678  0.        ]
 [ 0.          0.          0.          0.        ]
 [ 0.70710678  0.70710678  0.70710678  0.        ]]
```

Satirlari normalize edebilirdik,

```
print normalize(mat, norm='l1', axis=1).todense()
```

```
[[-0.33333333 -0.33333333 -0.33333333  0.        ]
 [ 0.          0.          0.          0.        ]
 [ 0.33333333  0.33333333  0.33333333  0.        ]]
```