

Bir Coklu Ikisel Dagilim (Multivar. Binary Distribution) ve Boltzmann Dagilimi

$$P(\mathbf{x}; W) = \frac{1}{Z(W)} \exp \left[ \frac{1}{2} \mathbf{x}^T W \mathbf{x} \right] \quad (3)$$

ki  $W$  simetrik ve caprazinda (diagonal) sifir iceren bir matristir.

Olurluk (likelihood)

$$\prod_{n=1}^N P(\mathbf{x}^{(n)}; W) = \frac{1}{Z(W)} \exp \left[ \frac{1}{2} \sum_{n=1}^N \mathbf{x}^{(n)T} W \mathbf{x}^{(n)} \right]$$

Log olurluk

$$\mathcal{L} = \ln \left( \prod_{n=1}^N P(\mathbf{x}^{(n)}; W) \right) = \sum_{n=1}^N \left[ \frac{1}{2} \mathbf{x}^{(n)T} W \mathbf{x}^{(n)} - \ln Z(W) \right] \quad (1)$$

Birazdan  $\frac{\partial \mathcal{L}}{\partial w_{ij}}$  turevini alacagiz, o sirada  $\ln Z(W)$ 'nin turevi lazim, daha dogrusu  $Z(W)$ 'yi nasıl turevi alinir hale getiririz?

$Z(W)$  normalizasyon sabiti olduguna gore, dagilimin geri kalaninin sonsuzlar uzerinden entegrali (ya da toplami) normalizasyon sabitine esittir,

$$Z(W) = \sum_{\mathbf{x}} \exp \left[ \frac{1}{2} \mathbf{x}^T W \mathbf{x} \right]$$

$$\ln Z(W) = \ln \left[ \sum_{\mathbf{x}} \exp \left( \frac{1}{2} \mathbf{x}^T W \mathbf{x} \right) \right]$$

Log bazli turev alinca log icindeki hersey oldugu gibi bolume gider, ve log icindeki turevi alinirak bolume koyulur. Fakat log icine dikkatli bakarsak bu zaten  $Z(W)$ 'nin tanimidir, boylece denklemi temizleme sansi dogdu, bolume hemen  $Z(W)$  deriz, ve turevi log'un icine uygulariz,

$$\begin{aligned} \frac{\partial}{\partial w_{ij}} \ln Z(W) &= \frac{1}{Z(W)} \left[ \sum_{\mathbf{x}} \frac{\partial}{\partial w_{ij}} \exp \left( \frac{1}{2} \mathbf{x}^T W \mathbf{x} \right) \right] \\ \frac{\partial}{\partial w_{ij}} \exp \left( \frac{1}{2} \mathbf{x}^T W \mathbf{x} \right) &= \frac{1}{2} \exp \left( \frac{1}{2} \mathbf{x}^T W \mathbf{x} \right) \frac{\partial}{\partial w_{ij}} \mathbf{x}^T W \mathbf{x} \end{aligned} \quad (2)$$

(2)'in icindeki bolumu acalim,

$$\frac{\partial}{\partial w_{ij}} \mathbf{x}^T W \mathbf{x} = x_i x_j$$

Simdi (2)'ye geri koyalim,

$$\begin{aligned}
&= \frac{1}{2} \exp \left( \frac{1}{2} \mathbf{x}^T \mathbf{W} \mathbf{x} \right) x_i x_j \\
\frac{\partial}{\partial w_{ij}} \ln Z(\mathbf{W}) &= \frac{1}{Z(\mathbf{W})} \left[ \sum_{\mathbf{x}} \frac{1}{2} \exp \left( \frac{1}{2} \mathbf{x}^T \mathbf{W} \mathbf{x} \right) x_i x_j \right] \\
&= \frac{1}{2} \sum_{\mathbf{x}} \frac{1}{Z(\mathbf{W})} \exp \left( \frac{1}{2} \mathbf{x}^T \mathbf{W} \mathbf{x} \right) x_i x_j \\
&= \frac{1}{2} \sum_{\mathbf{x}} P(\mathbf{x}; \mathbf{W}) x_i x_j
\end{aligned}$$

Ustteki son ifadede bir kisaltma kullanalim,

$$\sum_{\mathbf{x}} P(\mathbf{x}; \mathbf{W}) x_i x_j = \langle x_i, x_j \rangle_{P(\mathbf{x}; \mathbf{W})}$$

Artik  $\ln Z(\mathbf{W})$ 'nin turevini biliyoruz. O zaman tum log olurlugun turevine (1) donebiliriz,

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial w_{ij}} &= \sum_{n=1}^N \left[ \frac{\partial}{\partial w_{ij}} \frac{1}{2} \mathbf{x}^{(n)T} \mathbf{W} \mathbf{x}^{(n)} - \frac{\partial}{\partial w_{ij}} \ln Z(\mathbf{W}) \right] \\
&= \sum_{n=1}^N \left[ \frac{1}{2} x_i^{(n)} x_j^{(n)} - \frac{\partial}{\partial w_{ij}} \ln Z(\mathbf{W}) \right] \\
&= \sum_{n=1}^N \left[ \frac{1}{2} x_i^{(n)} x_j^{(n)} - \frac{1}{2} \langle x_i x_j \rangle_{P(\mathbf{x}; \mathbf{W})} \right]
\end{aligned}$$

1/2 sabitlerini atalim,

$$= \sum_{n=1}^N \left[ x_i^{(n)} x_j^{(n)} - \langle x_i x_j \rangle_{P(\mathbf{x}; \mathbf{W})} \right]$$

Eger

$$\langle x_i x_j \rangle_{\text{Data}} = \frac{1}{N} \sum_{n=1}^N x_i^{(n)} x_j^{(n)}$$

olarak alırsak, esitligin sag tarafi verisel kovaryansi (empirical covariance) temsil eder. Duzenleyince,

$$N \cdot \langle x_i x_j \rangle_{\text{Data}} = \sum_{n=1}^N x_i^{(n)T} x_j^{(n)}$$

simdi esitligin sag tarafi uc ustteki formule geri koyulabilir,

$$\frac{\partial \mathcal{L}}{\partial w_{ij}} = N \left[ \langle x_i x_j \rangle_{\text{Data}} - \langle x_i x_j \rangle_{P(x;W)} \right]$$

Bu bir gradyan guncelleme formulu olarak gorulebilir, ve N yerine bir guncelleme sabiti alınabilir.

Ana dagilim fonksiyonu baz alınarak, yeni veri  $x$  uzerinde o  $x$  uzerinde biri haric tum ogelerin bilindigi durumda bilinmeyen tek hucre  $i$  icin 1 olma olasilik degeri,

$$P(x_i = 1 | x_j, j \neq i) = \frac{1}{1 + e^{-a_i}}$$

ve,

$$a_i = \sum_j w_{ij} x_j$$

Bu kosulsal olasiligin ne kadar temiz oldugu onemli, ustteki gorulen bir sigmoid fonksiyonu. Bu fonksiyonlar hakkında daha fazla bilgi *Lojistik Regresyon* yazisinda bulunabilir. Ana formül (3)'ten bu noktaya nasil eristik?

$x$  vektörü icinde sadece  $x_i$  ogesinin 1 olmasini  $x^b$  olarak alalım. Once kosulsal dagilimda “verili” olan kısmi elde etmek lazim. O zaman

$$P(x_j, j \neq i) = P(x^0) + P(x^1)$$

Bu bir marjinalizasyon ifadesi, tum olasi  $i$  degerleri uzerinde bir toplam alinca geri kalan  $j$  degerlerinin dagilimini elde etmis oluruz.

$$P(x_i = 1 | x_j, j \neq i) = \frac{P(x^1)}{P(x^0) + P(x^1)}$$

cunku  $P(A|B) = P(A, B)/P(B)$  bilindigi gibi, ve  $P(x^1)$  icinde  $x_1 = 1$  setini iceren tum veriler uzerinden.

Esitligin sag tarafinda  $P(x^1)$ 'i bolen olarak gormek daha iyi, ayrica ulasmak istedigimiz  $1/(1 + e^{-a_i})$  ifadesinde  $+1$ 'den kurtulmak iyi olur, boylece sadece  $e^{-a_i}$  olan esitligi ispatlariz. Bunun her iki denklemde ters ceviri 1 cikartabiliriz,

$$\begin{aligned} 1/P(x_i = 1|x_j, j \neq i) &= \frac{P(x^0) + P(x^1)}{P(x^1)} \\ &= 1 + \frac{P(x^0)}{P(x^1)} \end{aligned}$$

Bir cikartirsak,  $\frac{P(x^0)}{P(x^1)}$  kalir. Bu bize ulasmak istedigimiz denklemde  $e^{-\alpha_i}$  ibaresini birakir. Artik sadece  $\frac{P(x^0)}{P(x^1)}$ 'in  $e^{-\alpha_i}$ 'e esit oldugunu gostermek yeterli.

$$\frac{P(x^0)}{P(x^1)} = \exp(x^{0T} W x^0 - x^{1T} W x^1)$$

Simdi  $x^T W x$  gibi bir ifadeyi indisler bazinda acmak icin sunlari yapalim,

$$x^T W x = \sum_{k,j} x_k x_j w_{kj}$$

Ustteki cok iyi bilinen bir acilim. Eger

$$\sum_{k,j} \underbrace{x_k x_j w_{ij}}_{Y_{kj}} = \sum_{k,j} Y_{kj}$$

alirsak birazdan yapacagimiz islemler daha iyi gorulebilir. Mesela  $k = i$  olan durumu dis toplamdan disari cekebiliriz

$$= \sum_{k \neq i} \sum_j Y_{kj} + \sum_j Y_{ij}$$

Daha sonra  $j = i$  olan durumu ic toplamdan disari cekebiliriz,

$$= \sum_{k \neq i} (\sum_{j \neq i} Y_{kj} + Y_{ki}) + \sum_j Y_{ij}$$

Ic dis toplamlari birlestirelim,

$$\begin{aligned} &= \sum_{k \neq i, j \neq i} Y_{kj} + \sum_{k \neq i} Y_{ki} + \sum_j Y_{ij} \\ &= \sum_{k \neq i, j \neq i} Y_{kj} + \sum_k Y_{ki} + \sum_j Y_{ij} + Y_{ii} \end{aligned}$$

Ustteki ifadeyi  $\exp(x^{0T} W x^0 - x^{1T} W x^1)$  icin kullanirsak,

$$\exp \left( \sum_k Y_{ki}^0 + \sum_j Y_{ij}^0 + Y_{ii}^0 - \left( \sum_k Y_{ki}^1 + \sum_j Y_{ij}^1 + Y_{ii}^1 \right) \right)$$

$\sum_{k \neq i, j \neq i} Y_{kj}$  teriminin nereye gittiği merak edilirse, bu ifade i'ye dayanmadığı için bir eksi bir artı olarak iki defa dahil edilip iptal olacaktı.

$$= \exp \left( 0 - \left( \sum_k Y_{ki}^1 + \sum_j Y_{ij}^1 + Y_{ii}^1 \right) \right)$$

W'nin simetrik matris olduğunu düşünürsek,  $\sum_k Y_{ki}^1$  ile  $\sum_j Y_{ij}^1$  aynı ifadedir,

$$= \exp \left( - \left( 2 \sum_j Y_{ij}^1 + Y_{ii}^1 \right) \right)$$

W sıfır caprazlı bir matristir, o zaman  $Y_{ii}^1 = 0$ ,

$$= \exp \left( 2 \sum_j Y_{ij}^1 \right) = \exp(-2a_i)$$

Orijinal dağılım denkleminde 1/2 ifadesi vardı, onu basta işlemlere dahil etmemistik, edilseydi sonuç  $\exp(-a_i)$  olacaktı.

```
class Boltzmann:
```

```

def __init__(self, n_iter=100, eta=0.01, sample_size=100, init_sample_size=10):
    self.n_iter = n_iter
    self.eta = eta
    self.sample_size = sample_size
    self.init_sample_size = init_sample_size

def sigmoid(self, u):
    return 1. / (1. + np.exp(-u));

def draw(self, Sin, T):
    """
    draw - perform single Gibbs sweep to draw a sample from distribution
    """
    N=Sin.shape[0]
    S=Sin.copy()
    rand = np.random.rand(N, 1)
    for i in xrange(N):
        h=np.dot(T[i, :], S)
        S[i]=rand[i]<self.sigmoid(h);
    return S

def sample(self, T):
    N=T.shape[0]
    s=np.random.rand(N)<self.sigmoid(0)
    for k in xrange(self.init_sample_size):
        s=self.draw(s, T)
```

```

S=np.zeros((N,self.sample_size))
S[:,0]=s
for i in xrange(1,self.sample_size):
    S[:,i]=self.draw(S[:,i-1],T)
return S.T

def fit(self, X):
    W_data=np.dot(X.T,X)/X.shape[1];
    print W_data.shape
    W_init = np.random.uniform(size=W_data.shape)
    W_init = np.tril(W_init) + np.tril(W_init, -1).T
    np.fill_diagonal(W_init, 0)
    print W_init
    W = W_init.copy()
    for i in range(self.n_iter):
        print 'Iteration', i
        S = self.sample(W)
        W_guess=np.dot(S.T,S)/S.shape[1];
        W += self.eta * (W_data - W_guess)
        np.fill_diagonal(W, 0)

    return W

import numpy as np

clf = Boltzmann()

A = np.array([\
[0.,1.,1.,0],\
[1.,1.,0, 0],\
[1.,1.,1.,0],\
[0, 1.,1.,1.],\
[0, 0, 1.,0]\
])

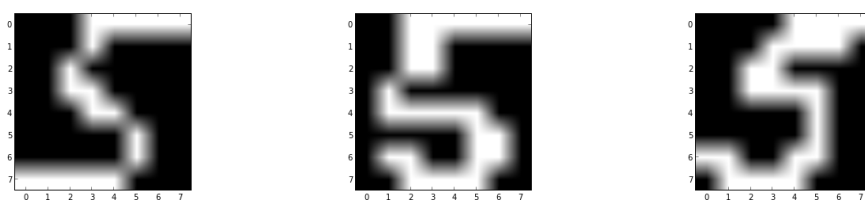
print clf.fit(A)

Y = np.loadtxt('../..//stat/stat_mixbern/binarydigits.txt')
label = np.ravel(np.loadtxt('../..//stat/stat_mixbern/bindigitlabels.txt'))
Y5 = Y[label==5]
plt.imshow(Y5[0,:].reshape((8,8),order='C'), cmap=plt.cm.gray)
plt.savefig('boltzmann_01.png')

plt.imshow(Y5[1,:].reshape((8,8),order='C'), cmap=plt.cm.gray)
plt.savefig('boltzmann_02.png')

plt.imshow(Y5[2,:].reshape((8,8),order='C'), cmap=plt.cm.gray)
plt.savefig('boltzmann_03.png')

```



Information Theory, Inference and Learning Algorithms, D. MacKay

<http://nbviewer.ipython.org/gist/aflaxman/7d946762ee99daf739f1>

<http://math.stackexchange.com/questions/1095491/from-pxw-fractional-zw-exp-bigl-fractional-2-xt-w-x-bigr-to-sigmoid/>