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

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

Olurluk (likelihood)

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

Log olurluk

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

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

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

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

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

Log bazli turev alinca log icindeki hersey oldugu gibi bolume gider, ve log icindekinin 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,

$$\frac{\partial}{\partial w_{ij}} \ln Z(W) = \frac{1}{Z(W)} \left[\sum_x \frac{\partial}{\partial w_{ij}} \exp\big(\frac{1}{2}x^T W x\big)\right]$$

$$\frac{\partial}{\partial w_{ij}} \exp\big(\frac{1}{2}x^T W x\big) = \frac{1}{2} \exp\big(\frac{1}{2}x^T W x\big) \frac{\partial}{\partial w_{ij}} x^T W x \quad (2)$$

(2)'in icindeki bolumu acalim,

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

Simdi (2)'ye geri koyalim,

$$= \frac{1}{2} \exp\big(\frac{1}{2} x^T W x\big) x_i x_j$$

$$\frac{\partial}{\partial w_{ij}} \ln Z(W) = \frac{1}{Z(W)} \bigg[\sum_x \frac{1}{2} \exp\big(\frac{1}{2} x^T W x\big) x_i x_j \bigg]$$

$$= \frac{1}{2} \sum_x \frac{1}{Z(W)} \exp\big(\frac{1}{2} x^T W x\big) x_i x_j$$

$$= \frac{1}{2} \sum_x P(x; W) x_i x_j$$

Ustteki son ifadede bir kisaltma kullanalim,

$$\sum_x P(x; W) x_i x_j =< x_i, x_j >_{P(x;W)}$$

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

$$\frac{\partial \mathcal{L}}{\partial w_{ij}} = \sum_{n=1}^{N} \bigg[\frac{\partial}{\partial w_{ij}} \frac{1}{2} x^{(n)^T} W x^{(n)} - \frac{\partial}{\partial w_{ij}} \ln Z(W) \bigg]$$

$$= \sum_{n=1}^{N} \bigg[\frac{1}{2} x_i^{(n)^T} x_j^{(n)} - \frac{\partial}{\partial w_{ij}} \ln Z(W) \bigg]$$

$$= \sum_{n=1}^{N} \bigg[\frac{1}{2} x_i^{(n)^T} x_j^{(n)} - \frac{1}{2} < x_i x_j >_{P(x;W)} \bigg]$$

1/2 sabitlerini atalim,

$$= \sum_{n=1}^{N} \bigg[x_i^{(n)^T} x_j^{(n)} - < x_i x_j >_{P(x;W)} \bigg]$$

Eger

$$< x_i x_j >_{Data} = \frac{1}{N} \sum_{n=1}^{N} x_i^{(n)^T} x_j^{(n)}$$

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

$$
N \cdot < x_i x_j >_{Data} = \sum_{n=1}^{N} {x_i^{(n)}}^T x_j^{(n)}
$$

simdi esitligin sag tarafi uc ustteki formule geri koyulabilir,

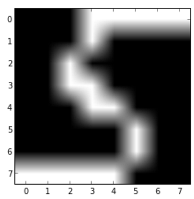
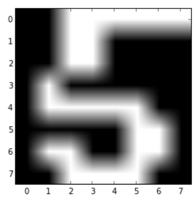
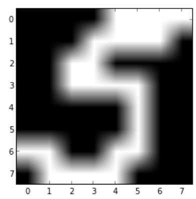
$$
\frac{\partial \mathcal{L}}{\partial w_{ij}} = N \Big[< x_i x_j >_{Data} - < x_i x_j >_{P(x;W)} \Big]
$$

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

```
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')
```

```
from pymc import *
X = Uniform('X', zeros(20), ones(20), value=0.5*ones(20))
@pm.potential
def SNLS(X=X):
    logp = -X[0]**2 / X[1]
    logp += -X[1]**2 / X[2]   # or whatever...
    return logp
m = pm.MCMC([X, SNLS])
m.use_step_method(pm.AdaptiveMetropolis, X)
m.sample(100)
print X.trace().sum()
print X.trace()
```

```
 [-----------------100%-----------------] 100 of 100 complete in 0.0 sec895.374930314
[[ 0.40334651  0.54660383  0.54176025 ...,  0.40196009  0.58059625
   0.44788672]
```

```
[ 0.39875173  0.47041323  0.56148785 ...,  0.30680384  0.59048635
  0.53342935]
[ 0.35293268  0.43296077  0.60987373 ...,  0.33573178  0.47394715
  0.5225162 ]
...,
[ 0.29668289  0.2283285   0.62517358 ...,  0.14847019  0.35648232
  0.00644193]
[ 0.29668289  0.2283285   0.62517358 ...,  0.14847019  0.35648232
  0.00644193]
[ 0.29668289  0.2283285   0.62517358 ...,  0.14847019  0.35648232
  0.00644193]]
```

Information Theory, Inference and Learning Algorithms

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