Prostate data modeling

July 3, 2019

import modules

```
In [1]: import numpy as np
        import pandas as pd
```

reading data and variable setting

```
In [2]: dat = pd.read_csv('prostate.csv',encoding='UTF8').drop(columns = 'Unnamed: 0')
```

```
In [3]: train = dat[dat['train']].drop(columns = 'train')
```

```
In [4]: y = train['lpsa']
        X = train.drop(columns = 'lpsa')
        X = X.values
```

```
In [5]: N,p = X.shape
```

Make the function wich return XX'

```
In [6]: def product(a):
            n = len(a)
            out = np.zeros([n,n])
            for i in range(n):
                for j in range(n):
                    out[i,j] = a[i]*a[j]
            return(out)
```

Make expit function which is inverse function of *logit*

```
In [7]: def expit(x):
            if x < 100:
                return(np.exp(x)/(1+np.exp(x)))
            else:
                return(1)
```

Start Variational inference which return w_j which is the parameter of the $q_4^*(\gamma_j) \sim Bernoulli(w_j)$

```
In [8]: def rhotest(rho):
            sigmab = 1
            A = 10**(-7)
            B = 10**(-7)
            tau = 1
            #rho = 0.8
            w = np.repeat(0.5,p)
            lamb= np.log(rho/(1-rho))
            t=0
            for iteration in range(1000):
                test= False
                W = np.diag(w)
                omega = product(w) + W.dot(np.eye(p)-W)
                sigma = np.linalg.inv(tau*np.multiply(X.T.dot(X),omega)+ (1/sigmab) * np.eye(p)
                mu = tau*sigma.dot(W.dot(X.T.dot(y)))
                s = B + 0.5*(np.linalg.norm(y)**2 -2*y.T.dot(X).dot(W).dot(mu)
                             + np.trace(np.multiply(X.T.dot(X),omega).dot(product(mu) +sigma)))
                tau = (A+N/2)/s
                eta =np.zeros(len(w))
                for j in range(p):
                    eta[j] = lamb - 0.5*tau *(mu[j]**2 + sigma[j,j])*np.linalg.norm(X[:,j])**2
                    + tau*X[:,j].T.dot(y*mu[j]-np.delete(X,j,1).dot(np.diag(np.delete(w,j))).do
                    w[j] = expit(eta[j])
                t= t+1
            return(w)

In [9]: rhopost = []
        rholst = [0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9]
        for i in rholst:
            rhopost.append(rhotest(i))
```

Plotting the change of w_j

```
In [10]: df = pd.DataFrame(rhopost)
         df.columns = dat.columns[:-2]
         df.index = rholst
         df.round(3)
```

Out[10]:

	lcavol	lweight	age	lbph	svi	lcp	gleason	pgg45
0.1	0.0	0.0	0.0	0.000	0.039	0.000	0.0	0.0
0.2	0.0	0.0	0.0	0.000	0.075	0.000	0.0	0.0
0.3	0.0	0.0	0.0	0.000	0.107	0.000	0.0	0.0
0.4	0.0	0.0	0.0	0.000	0.134	0.000	0.0	0.0
0.5	0.0	0.0	0.0	0.000	0.161	0.000	0.0	0.0
0.6	0.0	0.0	0.0	0.000	0.334	0.000	0.0	0.0
0.7	0.0	0.0	0.0	0.000	0.464	0.442	0.0	0.0
0.8	0.0	0.0	0.0	0.328	0.601	0.642	0.0	0.0
0.9	0.0	0.0	0.0	0.669	0.768	0.822	0.0	0.0