CS282 Topics in Machine Learning: Problem Set 1

Morris Yau

September 2016

1 Matrix Visualization

We visualize the matrices for data generated by the restaurant,stick breaking, and weak limit processes. We see that the restaurant process generates a left ordered matrix and as the observations increase to be greater than $\alpha = 10$ than no more dishes are sampled.

The stick breaking process imposes an ordering on the bernoulli flips, so that most of the features are observed for small column numbers as expected.

The weak limit approximation generates a matrix for which the rows are i.i.d, with some features flipped on more often than others.

2 Poisson Histograms

The restaurant, stick breaking, and weak limit processes all generate the same number of features per observation which is distributed as $poisson(\alpha)$ for $\alpha = 10$.

3 Total Features as Function of Observations

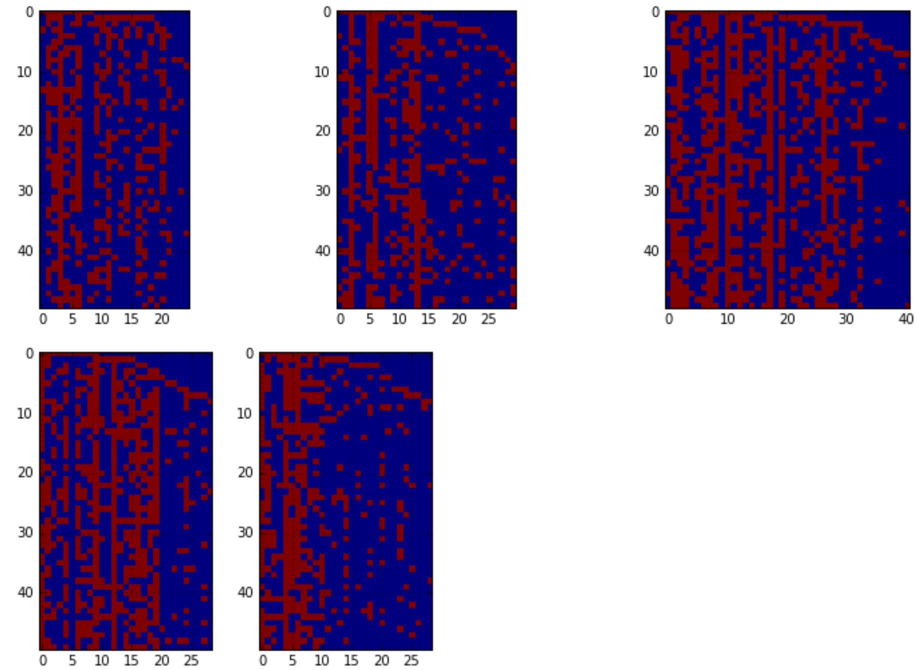
The total features grow harmonically as a function of the observations. We set $\alpha = 100$

4 Asymptotics Varying with K

As the number of features increase the weak limit and stick breaking processes exhibit the asymptotic poisson (per observation) and harmonic (over the entire matrix) behavior. We plot the number of features per observation and observe that the number converges to α as K goes to infinity. Note that the plot for the weak limit fluctuates because we're re-simulating the matrix for each K, which is necessary to plot the asymptotics.

Since the convergence of the weak limit and stick breaking processes is asymptotic, we need to determine when to truncate K. One method of determining a

Figure 1: Restaurant Process $\alpha = 10$, $N = 50$



good truncation value is to do what we have done, and to empirically simulate for various values of K and heuristically select one for which the number of features per observation is approximately α and the number of features for the whole matrix grows harmonically by the number of observations.

Figure 2: Stick Breaking $\alpha = 10$, $N = 50$, $K = 50$

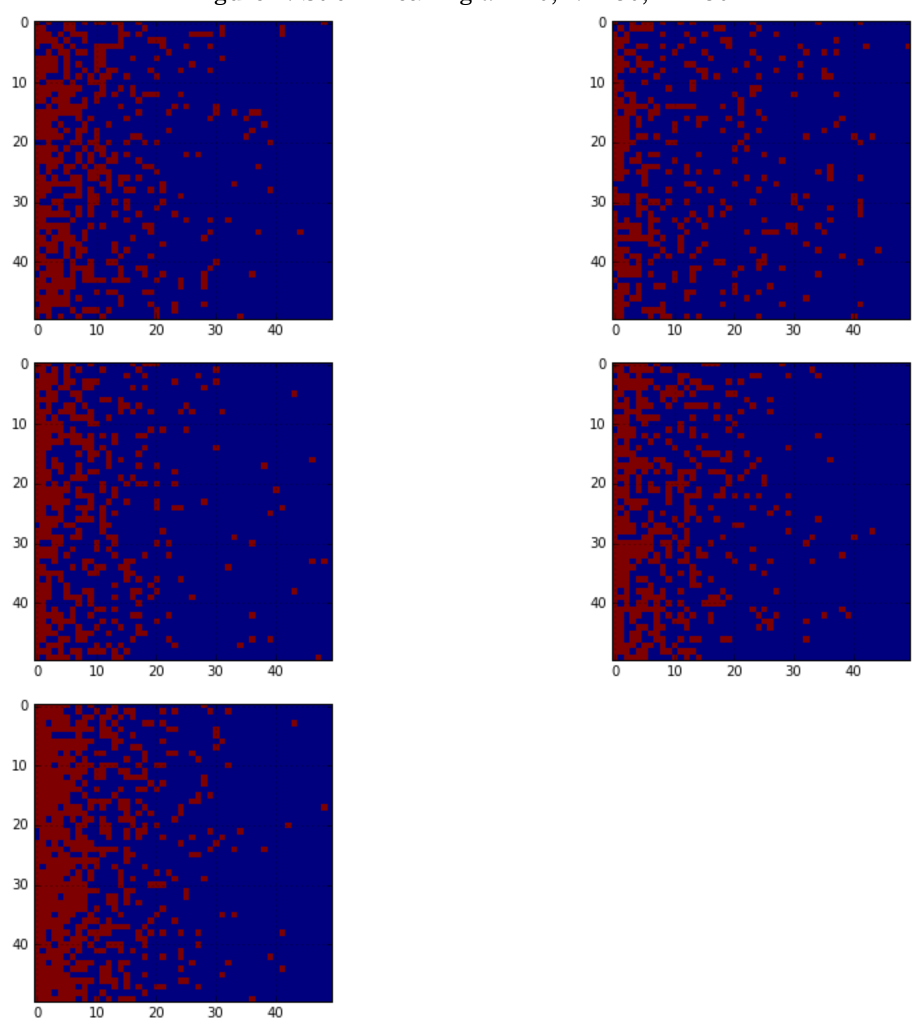


Figure 3: Weak Limit $\alpha = 10$, $N = 50$, $K = 50$

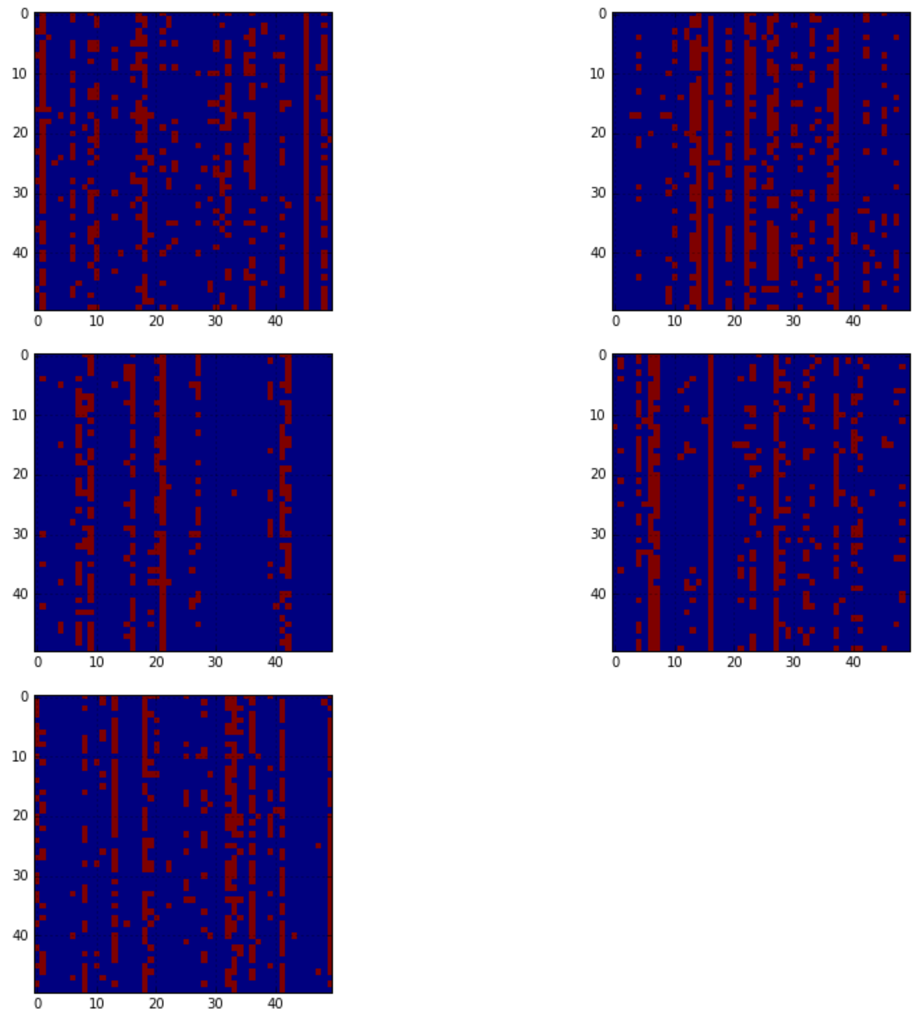


Figure 4: features per observation restaurant $\alpha = 10$

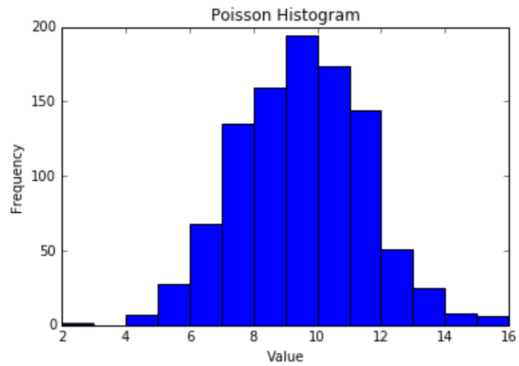


Figure 5: features per observation stick breaking $\alpha = 10$

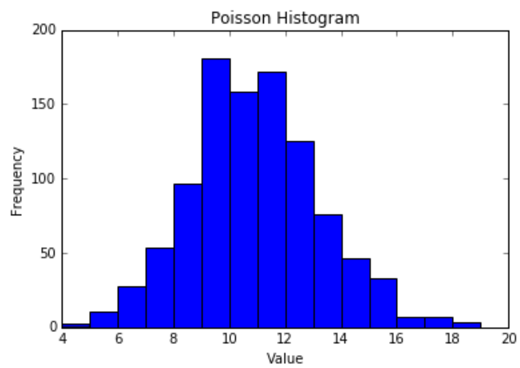


Figure 6: features per observation weak limit $\alpha = 10$

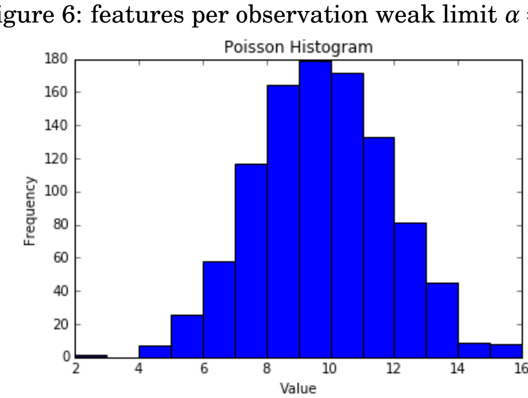


Figure 7: Harmonic Sums $\alpha = 100$

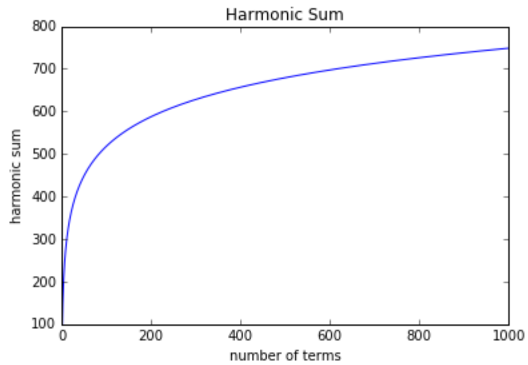


Figure 8: Total Features Restaurant $\alpha = 100$

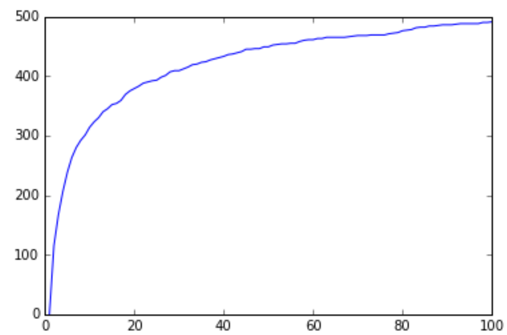


Figure 9: Total Features Weak Limit $\alpha = 100$

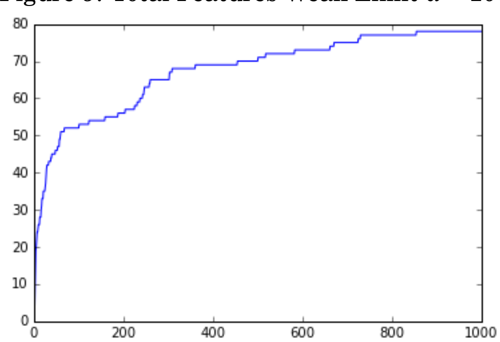


Figure 10: Total Features Stick Breaking $\alpha = 100$

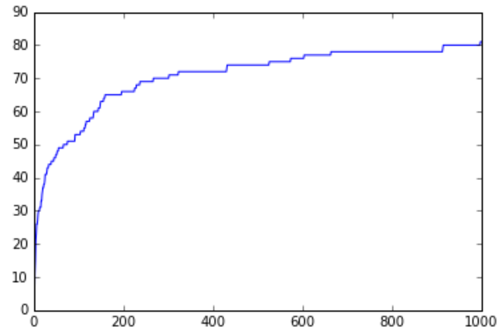


Figure 11: Asymptotics of Weak Limit $\alpha = 100$

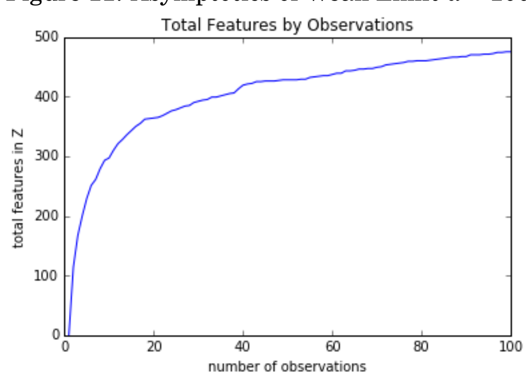
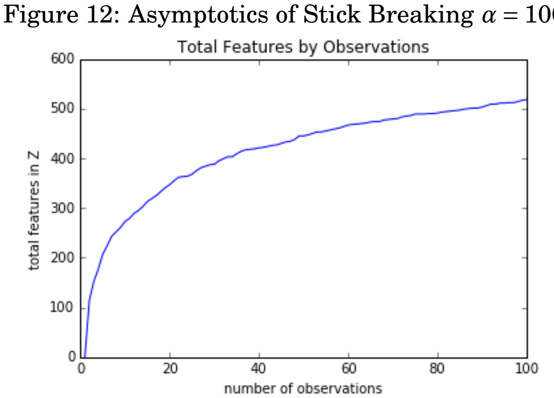


Figure 12: Asymptotics of Stick Breaking $\alpha = 100$



```python
import numpy as NP
import numpy.random as NR
import scipy.stats as SPST
import scipy.special as SPS
import matplotlib.pyplot as plt
from pylab import *

def weakLimit(alpha,N,K):
    Z = NP.zeros((N,K))
    u = SPST.beta.rvs(float(alpha)/K,1,size=K)
    for j in range(K):
        Z[:,j] = SPST.bernoulli.rvs(u[j],size=N)

    return Z

def stickBreak(alpha,N,K):
    Z = NP.zeros((N,K))
    u = [SPST.beta.rvs(alpha,1)]
    for i in range(1,K):
        u.append(u[i-1]*SPST.beta.rvs(alpha,1))

    for j in range(K):
         Z[:,j] = SPST.bernoulli.rvs(u[j],size=N)
    return Z

def restaurant(alpha, N):
    Z = NP.ones((0,0))
    for i in range(1,N+1):
        z_old = (NR.uniform(0,1,(1,Z.shape[1])) <
(Z.sum(axis=0).astype(NP.float)/i))
        num_new = NR.poisson(alpha/i)
        z_new =  NP.ones((1,num_new))
        zi = NP.hstack((z_old,z_new))
        Z = NP.hstack((Z,NP.zeros((Z.shape[0],num_new))))
        Z = NP.vstack((Z,zi))
    return Z

#distribution of nonzero features per observation
def plotRowHistogram(Z):
    data = Z.sum(axis=1)
    binwidth = 1
    plt.hist(data, bins=range(int(min(data)), int(max(data)) +
binwidth, binwidth))
    plt.title("Poisson Histogram")
    plt.xlabel("Value")
    plt.ylabel("Frequency")
    fig = plt.gcf()
    plt.show()

#growth of nonzero features as observations increase
```

```python
def plotMatrixHistogram(Z):
    data = []
    for i in range(Z.shape[0]):
        data.append(sum(x > 0 for x in Z[0:i,:].sum(axis=0)))
    plt.plot([y for y in range(1,Z.shape[0]+1)],data)
    plt.xlabel('number of observations')
    plt.ylabel('total features in Z')
    plt.title('Total Features by Observations')
    plt.show()

#asymptotics of features per observation varying with K
def Asymp_obs_K(Z):
    row,col = Z.shape
    data = []
    for i in range(1,col):
        data.append(Z[:,:i].sum()/row)
    plt.plot([y for y in range(1,col)],data)
    plt.xlabel('K')
    plt.ylabel('features per observation')
    plt.title('Features Per Observation Varying With K')
    plt.show()

def weak_obs_K(alpha,N,K):
    data = []
    for i in range(1,K):
        Z = weakLimit(alpha,N,i)
        row,col = Z.shape
        data.append(Z.sum()/row)
    plt.plot([y for y in range(1,K)],data)
    plt.xlabel('K')
    plt.ylabel('features per observation')
    plt.title('Features Per Observation Varying With K')
    plt.show()

def harmonic(alpha,N):
    cum = 0
    data = []
    for i in range(1,N):
        cum = cum + float(1)/i
        data.append(cum*alpha)
    plt.plot([y for y in range(1,N)],data)
    plt.xlabel('number of terms')
    plt.ylabel('harmonic sum')
    plt.title('Harmonic Sum with alpha = 100')
    plt.show()

# plotting
print('Restaurant Process')
for i in range(5):
    Z = restaurant(10,50)
```

```python
    figure(1)
    imshow(Z, interpolation='nearest')
    grid(True)
    plt.show()

#make note of alpha = 100 in your writeup
Z = restaurant(100,100)
plotRowHistogram(Z)
plotMatrixHistogram(Z)

print('Stick Breaking Process')
for i in range(5):
    Z = stickBreak(10,50,50)
    figure(1)
    imshow(Z, interpolation='nearest')
    grid(True)
    plt.show()

Z = stickBreak(100,100,1000)
plotRowHistogram(Z)
plotMatrixHistogram(Z)
Asymp_obs_K(Z)

print('Weak Limit Approximation')
for i in range(5):
    Z = weakLimit(10,50,50)
    figure(1)
    imshow(Z, interpolation='nearest')
    grid(True)
    plt.show()

Z = weakLimit(100,100,1000)
plotRowHistogram(Z)
plotMatrixHistogram(Z)
weak_obs_K(100,10,500)

harmonic(100,1000)
```