Name: John Min, jcm2199

COMS 4772 Homework Set 4

(1) You may use the fact that *expectation is a linear operator*.

(a) For a random variable X, let EX denote its expected value. Show that

$$E\left((X-EX)(X-EX)^T\right) = E(XX^T) - EX(EX)^T.$$

The quantity on the left hand side is the variance-covariance matrix for X, which we will call $V(X)$.

Solution:

$$\begin{aligned}
V(X) &= E\Big[\Big(X - E(X)\Big)\Big(X - E(X)\Big)^\top\Big] \\
&= E\Big[XX^\top - E(X)X^\top - X\big[E(X)\big]^\top + E(X)\big[E(X)\big]^\top\Big] \\
&= E\big[XX^\top\big] - E\big[E(X)X^\top\big] - E\big[X[E(X)]^\top\big] + E\big[E(X)E(X)^\top\big] \\
&= E\big[XX^\top\big] - E(X)\big[E(X)\big]^\top
\end{aligned}$$

(b) Show that, for any (appropriately sized) matrix A we have

$$V(AX) = A(V(X))A^T.$$

Solution:

$$\begin{aligned}
V(AX) &= E\Big[\Big(AX - E(AX)\Big)\Big(AX - E(AX)\Big)^\top\Big] \\
&= E\Big[AXX^\top A^\top - E\Big(AX\Big)X^\top A^\top - AXE\Big(X^\top A^\top\Big) + E\Big(AX\Big)E\Big(X^\top A^\top\Big) \\
&= E\Big(AXX^\top A^\top\Big) + E\Big(AX\Big)E\Big(X^\top A^\top\Big) \\
&= AE\Big(XX^\top\Big)A^\top + AE\Big(X\Big)E\Big(X^\top\Big)A^\top \\
&= A\big[V(X)\big]A^\top
\end{aligned}$$

(c) Show that

$$E(\|X\|^2) = \text{trace}(V(X)) + \|EX\|^2.$$

Solution:

$$
\begin{aligned}
E\Big(||X^2||\Big) &= E\Big(X^\top X\Big) \\
||E(X)||^2 &= \big[E(X)\big]^\top E(X) \\
\operatorname{tr}\Big(V(X)\Big) &= tr\Big(E(XX^\top) - E(X)E(X)^\top\Big)
\end{aligned}
$$

(d) Solve the stochastic optimization problem

$$\min_y E\|X - y\|_2^2,$$

where X is a random vector, and the expectation is taken with respect to X. What is the minimizer? What's the minimum value?

(2) Frobenius norm estimation. Suppose we want to estimate

$$\|A\|_F^2 = \operatorname{trace}(A^T A)$$

of a large matrix A. One way to do this is to hit A by random vectors w, and then measure the resulting norm.

(a) Find a sufficient conditions on a random vector w that ensures

$$E\|Aw\|^2 = \|A\|_F^2.$$

Prove that your condition works.

(b) What's a simple example of a distribution that satisfies the condition you derived above?

(c) Explain how you can put the relationship you found to practical use to estimate $\|A\|_F^2$ for a large A. In particular, you must explain how to estimate $\|A\|_F^2$ more or less accurately, depending on the need.

(d) Test out the idea in Matlab. Generate a random matrix A, maybe 500 x 1000. Compute its frobenius norm using `norm(A, 'fro')` command. Compare this to the result of your approach. Are they close? Is your approach faster?

(3) Consider again the logistic regression problem. Included with this homework is the covtype dataset (500K examples, 54 features).

Consider again the logistic regression formulation:

$$\min_\theta \frac{1}{N}\sum_{i=1}^{N} \log(1 + \exp(\tilde{x}_i^T\theta)) + \lambda\|\theta\|_2$$

where $\tilde{x}_i = -y_i x_i$ and you can take $\lambda = 0.01$ (small regularization).

Implement a stochastic gradient method for this problem.

Use the following options for step length:

(a) Pre-specified constant

(b) Decreasing with the rule $\alpha(k) \propto \frac{1}{k}$ (with some initialization)
(c) Decreasing with rule $\alpha(k) \propto \frac{1}{k^{0.6}}$ (with some initialization)

Divide covtype into two datasets, 90% training and 10% testing. Tune each of the three previous step size routines (i.e. adjust the constant or the constant initialization) until you are happy each one performs reasonably well. Make a graph showing the value of the *test likelihood* as a function of the iterates for each of the three strategies.

(4) (BONUS)

(a) Change the counting in the previous problem to be as a function of *effective passes through the data*, rather than iterations. For example, five iterations with batch size 1 should be no different than one iteration with batch size 5 in this metric.

(b) For the pre-specified constant step length strategy, compare test likelihood as a function of effective passes through the data for different random batch sizes, e.g. 1, 10, and 100.

(c) Again for pre-specified constant step length strategy, implement a growing batch size strategy, where the size of the batch increases with iterations. Can this strategy beat the fixed batch size strategy, with respect to effective passes through the data?

```
'''
Logistic regression using stochastic gradient descent with options for step
pre-specified constant
proportional to 1/k
proportional to k^(0.6)
'''

import numpy as np
import pandas as pd
import random
from numpy import exp
from numpy import dot

#class LogisticRegression:



N, K  = X.shape

def data_split(X, p, N, K):
    # check to see that 0 < p < 1
    n_train = int(float(p) * N)
    row_idx = random.shuffle([i for i in xrange(n_train)])
    X_train = X.ix[row_idx[:n_train], :]
```

```
    X_test = X.ix[row_idx[n_train:], :]
    return X_train, X_test

def X_tilde(X, y):
    return X.dot(y)

def SGD(X_tilde, theta, lamb, alpha, N):
    grad = 0
    for i in xrange(N):
    grad += sigmoid(X_tilde[i,:].dot(theta)).dot(-X_tilde[i,:].dot(theta))
    return alpha*grad/float(N)

def main():
# randomly initialize thetas
```