Asmt 5: Regression

David Winer
drwiner@cs.utah.edu

Overview

In this assignment you will explore regression techniques on high-dimensional data.

You will use a few data sets for this assignment:

- `http://www.cs.utah.edu/˜jeffp/teaching/cs5140/A5/A.dat`
- `http://www.cs.utah.edu/˜jeffp/teaching/cs5140/A5/X.dat`
- `http://www.cs.utah.edu/˜jeffp/teaching/cs5140/A5/Y.dat`
- `http://www.cs.utah.edu/˜jeffp/teaching/cs5140/A5/M.dat`
- `http://www.cs.utah.edu/˜jeffp/teaching/cs5140/A5/W.dat`

and a file stub:

- `http://www.cs.utah.edu/˜jeffp/teaching/cs5140/A5/FD.m`

These data sets are in matrix format and can be loaded into MATLAB or OCTAVE. By calling
`load filename` (for instance `load X.dat`)
it will put in memory the data in the file, for instance in the above example the matrix `X`. You can then
display this matrix by typing
`X`

*As usual, it is highly recommended that you use LaTeX for this assignment. If you do not, you may
lose points if your assignment is difficult to read or hard to follow. Find a sample form in this directory:*
`http://www.cs.utah.edu/˜jeffp/teaching/latex/`

1 Singular Value Decomposition (20 points)

First we will compute the SVD of the matrix `A` we have loaded
`[U,S,V] = svd(A)`

Then take the top k components of `A` for values of $k = 1$ through $k = 10$ using
```
Uk = U(:,1:k)
Sk = S(1:k,1:k)
Vk = V(:,1:k)
Ak = Uk*Sk*Vk'
```

A (10 points): Compute and report the L_2 norm of the difference between `A` and `Ak` for each value of k
using
`norm(A-Ak,2)`

```
n =

    [40.4833]
    [26.7169]
    [25.0000]
    [22.1915]
```

```
[17.6745]
[15.8129]
[13.3513]
[12.1880]
[ 9.1206]
[ 9.0000]
```

B (5 points): Find the smallest value k so that the L_2 norm of `A-Ak` is less than 10% that of `A`; k might or might not be larger than 10.

Answer

Norm(A,2) is 120.1943, and $.1 * 120.1943$ is 12.0194, so the k which is less than 10 percent is $k = 9$ ($A_9 = 9.1206$, $A_8 = 12.1880$).

C (5 points): Treat the matrix as 1125 points in 30 dimensions. Plot the points in 2 dimensions in the way that minimizes the sum of residuals squared.

Its $plot(A_2(:,1), A_2(:,2),"r.")$:
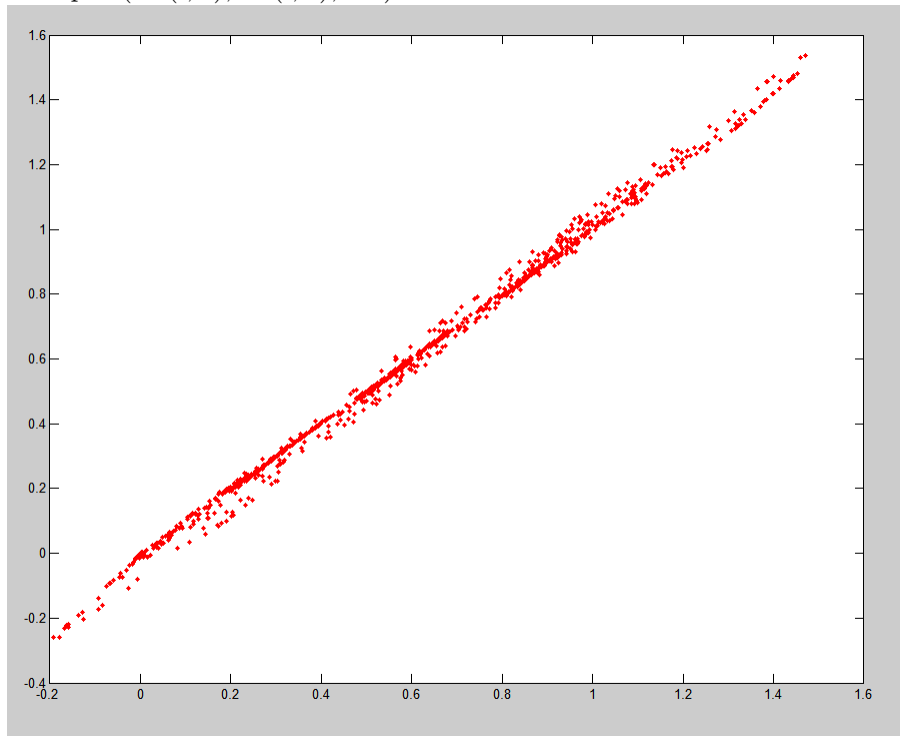
2   Frequent Directions and Random Projections (40 points)

Use the stub file `FD.m` to create a function for the Frequent Directions algorithm (**Algorithm 15.2.1**). We will consider running this code on matrix `A`.

A (20 points): We can measure the error $\max_{\|x\|=1} |\|Ax\|^2 - \|Bx\|^2|$ as `norm(A'*A - B'*B, 2)`.

- How large does `l` need to be for the above error to be at most $\|A\|_F^2/10$?

  Answer: $\|A\|_F^2/10 = 1903.7$. l must be at least 18, 1800.345

- How does this compare to the theoretical bound (e.g. for $k = 0$).

  Answer: I think the theoretical bound is $\|A\|_F^2/l$ when $k = 0$, and when $l = 36$, error is 1800.34451757 and the bound is 2115.3. Since the error is smaller than the bound, it's within the theoretical bound.

- How large does l need to be for the above error to be at most $\|A - A_k\|_F^2/10$ (for $k = 2$)?

  Answer: $l = 78$, $255.92 < \|A - A_2\|_F^2/10 = 295.1795$

Note: you can calculate $\|A\|_F^2$ as `norm(A, 'fro')^2`.

B (20 points): Create another l x d matrix B, but using random projections. You can do this by creating an l x n matrix S, and letting $B = SA$. Fill each entry of S by an independent normal random variable $S_{i,j} = \frac{1}{\sqrt{l}}N(0, 1)$ [was incorrectly $S_{i,j} = \sqrt{\frac{n}{l}}N(0, 1)$ before].

Estimate how large should l be in order to achieve $\max_{\|x\|=1} |\|Ax\|^2 - \|Bx\|^2| \le \|A\|_F^2/10$. To estimate the relationship between l and the error in this randomized algorithm, you will need to run multiple trials. Be sure to describe how you used these multiple trials, and discuss how many you ran and why you thought this was enough trials to run to get a good estimate.

  Answer:
  I ran each value of l 215 times to get an error bound of $\epsilon = .25$ with probability $\delta = .1$ where the number of times is $floor(1/\epsilon^2 * log_2(n/\delta))$. It's also a matter of practicality that 215 times per l wouldn't be that bad. I also checked that the averages decreased, in general, as I increased l.

  On the first pass, since I didn't know the range, I started in $l = 10$ and incremented by 2 until 1000 or until the average error of the 215 trials was less than the $1903.7 = \|A\|_F^2/10$.

  The average for when $l = 106$ is 1853.98, so the answer is that l should be at least 106 to acheive the error bound.

3   Linear Regression (40 points)

We will find coefficients C (was $a_1, \ldots, a_d$ in notes, but changed to avoid confusion with matrix A in **Q1**) to estimate $X*C \approx Y$, using the provided datasets X and Y. We will compare two approaches *least squares* and *ridge regression*.

Least Squares: Set `C = inverse(X' * X)*X'*Y`

Ridge Regression: Set `Cs = inverse(X'*X + s^2*eye(12))*X'*Y`

A (20 points):   Solve for the coefficients `C` (or `Cs`) using Least Squares and Ridge Regression with $s = \{0.1, 0.3, 0.5, 1.0, 2.0\}$ (i.e. s will take on one of those 5 values each time you try, say obtaining `C05` for $s = 0.5$). For each set of coefficients, report the error in the estimate $\hat{Y}$ of Y as `norm(Y - X*C,2)`.

  Answer:

  least-squares: Coefficients:

```
0.949378922318642
2.02147058307178
```

```
2.98822785238924
4.02561282572285
5.07415005254861
6.11066098334705
6.84572085868072
7.95396258817771
8.98882370006865
10.1458241261958
10.9280524526759
11.9861997785936
```

error is 25.8245

ridge regression: errors for coefficiens of s are $25.8247, 25.8261, 25.8289, 25.8421, 25.8944$ for $s \in \{0.1, 0.3, 0.5, 1.0, 2.0\}$

B (20 points):   Create three row-subsets of X and Y

- X1 = X(1:66,:) and Y1 = Y(1:66)
- X2 = X(34:100,:) and Y2 = Y(34:100)
- X3 = [X(1:33,:); X(67:100,:)] and Y3 = [Y(1:33); Y(67:100)]

Repeat the above procedure on these subsets and *cross-validate* the solution on the remainder of X and Y. Specifically, learn the coefficients C using, say, X1 and Y1 and then measure norm(Y(67:100) - X(67:100,:)*C,2).

Which approach works best (averaging the results from the three subsets): Least Squares, or for which value of s using Ridge Regression?
Answer: The best answer is Ridge regression with $s = 1.0$.

Linear regression:

coefficients learned from $X1, Y1$; *error* $C1 = 14.7833$
coefficients learned from $X2, Y2$; *error* $C2 = 14.5126$
coefficients learned from $X3, Y3$; *error* $C3 = 23.1624$
$avg = 17.4861$

Ridge Regression

$s = 0.1$
coefficients learned from $X1, Y2$; *error* $Cs1 = 14.7462$
coefficients learned from $X2, Y2$; *error* $Cs2 = 14.5237$
coefficients learned form $X3, Y3$; *error* $Cs3 = 23.1758$
$avg = 17.4819$
$s = 0.3$
coefficients learned from $X1, Y2$; *error* $Cs1 = 14.6734$
coefficients learned from $X2, Y2$; *error* $Cs2 = 14.5476$
coefficients learned form $X3, Y3$; *error* $Cs3 = 23.2038$

$avg = 17.4749$
$s = 0.5$
coefficients learned from $X1, Y2$; $error\ Cs1 = 14.6027$
coefficients learned from $X2, Y2$; $error\ Cs2 = 14.5737$
coefficients learned form $X3, Y3$; $error\ Cs3 = 23.2334$
$avg = 17.4699$
$s = 1.0$
coefficients learned from $X1, Y2$; $error\ Cs1 = 14.4353$
coefficients learned from $X2, Y2$; $error\ Cs2 = 14.6484$
coefficients learned form $X3, Y3$; $error\ Cs3 = 23.3147$
$avg = 17.4661$
$s = 2.0$
coefficients learned from $X1, Y2$; $error\ Cs1 = 14.1411$
coefficients learned from $X2, Y2$; $error\ Cs2 = 14.8373$
coefficients learned form $X3, Y3$; $error\ Cs3 = 23.5068$
$avg = 17.4951$

4 BONUS (2 points)

Consider a linear equation `W = M*S` where `M` is a measurement matrix filled with random values $\{-1, 0, +1\}$ (although now that they are there, they are no longer random), and `W` is the output of the sparse signal `S` when measured by `M`.

Use Orthogonal Matching Pursuit (as described in the notes) to recover the non-zero entries from `S`. Record the order in which you find each entry and the residual vector after each step.