STA 250

Final Exam (Project)

(Plots and Code)
Problem 1. Robust PCA

1. I decide to use ADMM to develop this RPCA problem.

Objective function is $\quad \min \quad \frac{1}{2}(\|W\|_F^2 + \|H\|_F^2) + \lambda\|S\|_1$

The constrain is $WH^T + S - M = 0$.

We can see it as a three-block problem, Lagrangian equation is:

$$L_\beta = \frac{1}{2}(\|W\|_F^2 + \|H\|_F^2) + \lambda\|S\|_1 + u^T(WH^T + S - M) + \frac{\beta}{2}\|WH^T + S - M\|_F^2$$

To update W, H and S, we need to derive the partial gradient of Lagrangian of S using soft-thresholding,

W : $W + \beta(WH^T + S - M)H + uH$

H : $\mathrm{H} + \beta(WH^T + S - M)^T W + u^T W$

$$\mathrm{S}: \begin{cases} M - WH^T - \frac{u}{\beta} - \frac{\lambda}{\beta} & if \;\; M - WH^T - \frac{u}{\beta} > \frac{\lambda}{\beta} \\ M - WH^T - \frac{u}{\beta} + \frac{\lambda}{\beta} & if \;\; M - WH^T - \frac{u}{\beta} < -\frac{\lambda}{\beta} \\ 0 & o.w. \end{cases}$$

We can directly derive the minimal value of W and H:

$$\frac{\partial L_\beta}{\partial W} = W + \beta\left(WH^T + S - M\right)H + uH = 0$$

$$W = (\beta(M - S) - U)H(I_K + \beta H^T H)^{-1}$$

$$\frac{\partial L_\beta}{\partial H} = \mathrm{H} + \beta(WH^T + S - M)^T W + u^T W = 0$$

$$\mathrm{H} = (\beta M^T - \beta S^T - U^T)\mathrm{W}(I_K + \beta W^T W)^{-1}$$

So the algorithm is:

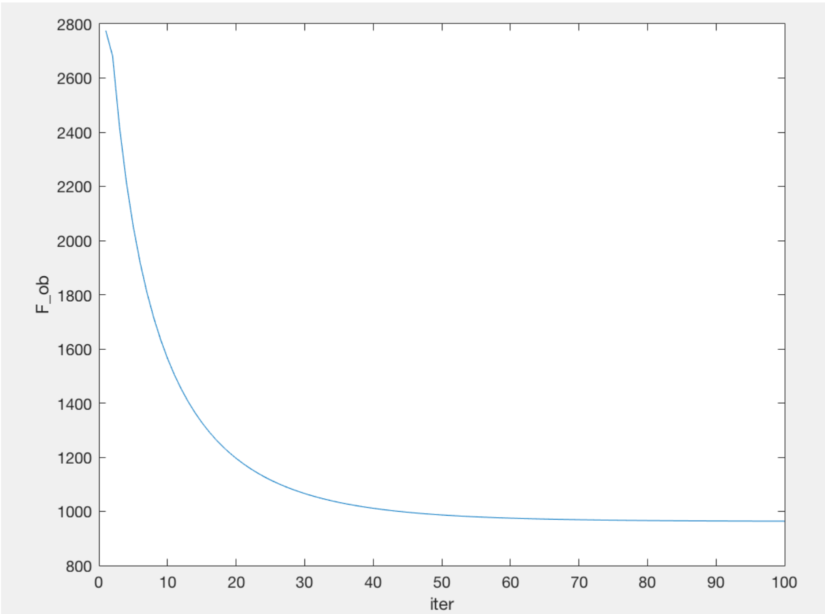
For k = 0,1,....

$$W = (\beta(M - S) - U)H(I_K + \beta H^T H)^{-1}$$
$$\mathrm{H} = (\beta M^T - \beta S^T - U^T)\mathrm{W}(I_K + \beta W^T W)^{-1}$$
$$S^{k+1} = \underset{S}{argmin} L_\beta(W^K, H^K, S, U^K)$$
$$U^{K+1} = U^K + \eta(WH^T + S - M)$$

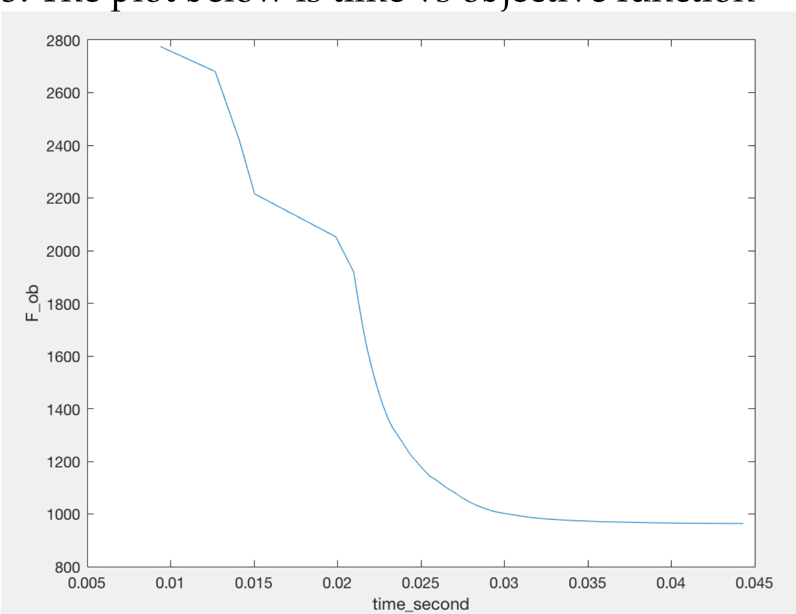
```
end
```

$W \in \mathbb{R}^{m*k}$, H$\in \mathbb{R}^{n*k}$, S$\in \mathbb{R}^{m*n}$
So the time complexity equals to O(mnk).

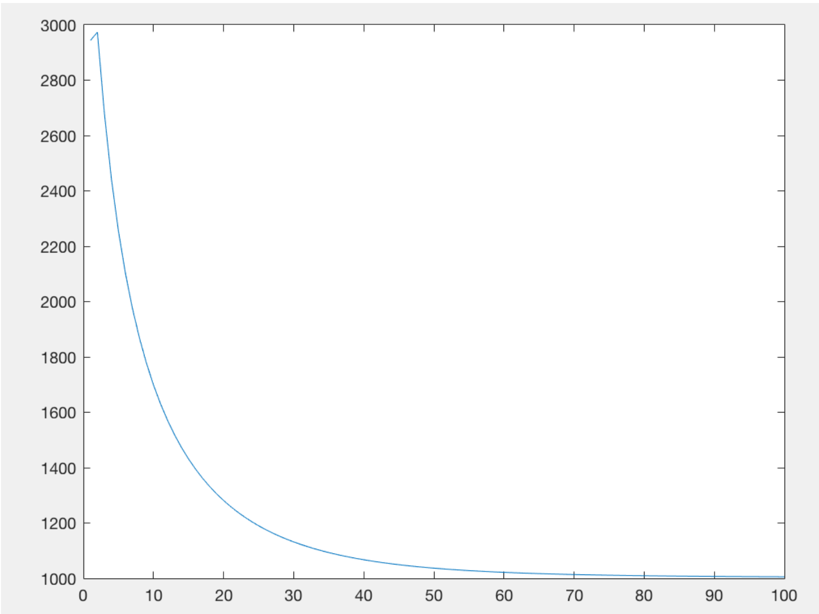
2. $\lambda = 2.5$. I derived the value of WH' at the end and compare it with real WH', the f norm of the difference between them is 16.7316 and I think it's small to cover the matrix.



3. The plot below is time vs objective function



4. $\lambda = 2.5$ dataset = mnist



4.242424727403974e+03

code

```
%problem 1
S_star = sprand(100, 100, 0.05);
%disp(S_star);
W_star = randn(100,10);
%disp(W_star);
H_star = randn(100,10);
%disp(H_star);
M = S_star + W_star * (H_star.');
%disp(M);


W = randn(100,10);
H = randn(100,10);
S = randn(100,100);
U = randn(100,100);
beta = 1;
yita = 0.02;
lamda = 2.5;
F_ob = zeros(100,1);
time = 0;
tic;


for i = 1:100
    W = (beta*(M - S)- U)*H*inv(eye(10) + beta*H'*H);
```

```
    H = (beta*M' - beta*S' - U')*W*inv(eye(10) + beta*W'*W);
    S = ST_S(W, H,  S,M, U, beta, lamda);
    U = U + yita*(W*H.' + S - M);
    f_ob = calc_ob(W, H, S,lamda);
    F_ob(i) = f_ob;
end
toc
plot(F_ob);


function S = ST_S(W, H, S, M, U, beta, lamda)
    if M - W*H.' - U/beta > lamda/beta
    S = M - W*H.' - U/beta - lamda/beta;
    elseif M - W*H.' - U/beta < -lamda/beta
            S = M - W*H.' - U/beta + lamda/beta;
    else
        S = 0;
    end
end


function f_ob = calc_ob(W, H,S, lamda)


f_ob = 0.5*(norm(W, 'fro')^2 + norm(H, 'fro')^2) +
lamda*(sum(abs(S(:))));
end
```

Problem 2. Extreme classification

2. I decide to use CD to develop this problem.

Objective function is min $\frac{1}{2}(\|Y - XWH^T\|_F^2) + \lambda \|W\|_F^2 + \lambda\|H\|_F^2$

We can generate W and H:

$$\frac{\partial L_\beta}{\partial W} = X^T \left(XWH^T - Y\right) + 2\lambda W$$

$$\frac{\partial L_\beta}{\partial H} = (XWH\text{^}T - Y)^T XW + 2\lambda H$$

Use GD to update

So the algorithm is:

For k = 0,1,....

$W^{k+1} = \underset{w}{argmin} L_\beta(W, H^K)$

$H^{k+1} = \underset{H}{argmin} L_\beta(W^K, H)$

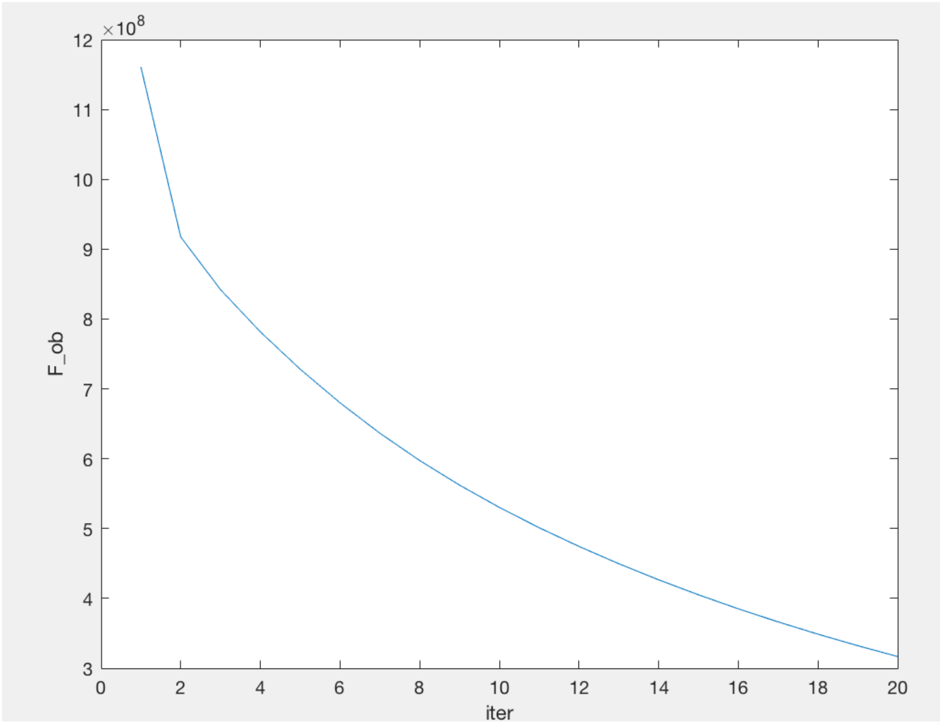
end

Time complexity equals to O(nd^2).

3.

If $\lambda = 2$,	If $\lambda = 5$.
When k = 5 P_k = 0.0154;	When k = 5 P_k = 0.0148;
When k = 1 P_k = 0.0123;	When k = 1 P_k = 0.0155;

With different λ , P_k is very close to each other.



×10^8
12
11
10
9
8
7
6
5
4
3
F_ob
0
2
4
6
8
10
12
14
16
18
20
iter

code

```
%Problem 2
filePath =
'/Users/zhuguanghua/Downloads/Bibtex/Bibtex_data.txt';
[ft_mat, lbl_mat] = read_data(filePath);

A = bibtextrSplit(:,1);
B = bibtextstSplit(:,1);
ft_mat = ft_mat.';
X_train = ft_mat(A,:);
lbl_mat = lbl_mat.';
Y_train = lbl_mat(A,:);
X_test = ft_mat(B,:);
Y_test = lbl_mat(B,:);

X = X_train;
Y = Y_train;
lamda = 2;
yita = 0.0000001;
W = randn(1836, 50);
H = randn(159, 50);
F_ob = zeros(20,1);
for i=1:20
    W = GD_W(W, H, X, Y, lamda, yita);
    H = GD_H(W, H, X, Y, lamda, yita);
    F_ob(i) = 1/2*(norm((Y - X*W*H.'),'fro')^2) + lamda*(norm(W,
'fro')^2) +lamda*(norm(H, 'fro')^2);
end
plot(F_ob);

yt = X_train*W*H.';
[a,b] = size(Y_test);
[c,d] = sort(yt, 2, 'descend');
k = 5;%(or k = 1)
p = zeros(a, k);
index = d(:, 1:k);

for i = 1:a
    p(i,1:k)=Y_test(i, index(i,:));
end
pk=sum(p,2)/k;
p_k=sum(pk)/a;
disp(p_k);

function W = GD_W(W, H, X, Y, lamda, yita)
    W = W - yita*(X.'*(X*W*H.' - Y)*H + 2*lamda*W);
```

```
end

function H = GD_H(W, H, X, Y, lamda, yita)
    H = H - yita*((X*W*H.' - Y).'*X*W + 2*lamda*H);
end
```