CSE 847 (Spring 2016): Machine Learning— Homework 6

Ding Wang

1 Sparse Learning

1. **Answer:**

   For MAP estimation, we need

$$w_{MAP}(X) = \arg max_w p(y|X, w, \lambda)p(w|\lambda)$$

Since

$$p(X|w, \lambda) = N(Xw, I_n)$$

$$p(w_i|\lambda) = \frac{\lambda}{2}e^{-\lambda|w_i|}$$

Then the log likelihood function is

$$\begin{aligned} L =& \log p(y|X, w, \lambda)p(w|\lambda) \\ =& \log \prod_{i=1}^{n} \frac{1}{(2\pi)^{d/2}} \exp\left(-\frac{1}{2}(y_i - x_i w)^T (y_i - x_i w)\right) \prod_{i=1}^{d} \frac{\lambda}{2} e^{-\lambda|w_i|} \\ =& nd \log \lambda - \frac{nd}{2} log(2\pi) - \frac{1}{2}\|Y - Xw\|^2 - \lambda \sum_{i=1}^{d} |w_i| \end{aligned}$$

When maximum the log likelihood function, it is the same as minimize the function

$$\arg min_w \frac{1}{2}\|Y - Xw\|^2 + \lambda \sum_{i=1}^{d} |w_i|$$

It is a Lasso optimization problem.
If $X^T X = 1$, then

$$\begin{aligned} & \frac{1}{2}\|Y - Xw\|^2 + \lambda \sum_{i=1}^{d} |w_i| \\ =& \frac{1}{2}(Y - Xw)^T (Y - Xw) + \lambda \sum_{i=1}^{d} |w_i| \\ =& \frac{1}{2}(Y^T Y - w^T X^T Y - Y^T Xw + w^T w) + \lambda \sum_{i=1}^{d} |w_i| \end{aligned}$$

Minimizing the previous equation by doing a derivative with w_i, we have

$$\begin{aligned} & - Y^T X^i + w_i + \lambda sign(w_i) = 0 \\ \Rightarrow & w_i = \begin{cases} Y^T X^i - \lambda, \text{if } sign(w_i) > 0 \\ Y^T X^i + \lambda, \text{if } sign(w_i) < 0 \\ 0, \text{other case} \end{cases} \end{aligned}$$

where X^i is the ith feature of all instance X.

2. **Answer:**

   For minimizing the project gradient, do a derivative with x, we have

$$\begin{aligned}
&\frac{\partial x_{i+1}}{\partial x} = 0 \\
\Rightarrow &\nabla \mathcal{L}(x_i)^T - \frac{1}{\gamma_i}(x - x_i) = 0 \\
\Rightarrow &x - (x_i - \gamma_i \nabla \mathcal{L}(x_i)^T) = 0
\end{aligned}$$

   For minimizing the Euclidean projection, do a derivative with x, we have

$$\begin{aligned}
&\frac{\partial x_{i+1}}{\partial x} = 0 \\
\Rightarrow &2\left(x - (x_i - \gamma_i \nabla \mathcal{L}(x_i)^T)\right) = 0 \\
\Rightarrow &x - (x_i - \gamma_i \nabla \mathcal{L}(x_i)^T) = 0
\end{aligned}$$

   Therefore, these two problems are equivalent.

3. I implement stability selection on top of sparse logistic regression used in Homework 4. The L1 parameter was set to 0.1. The Top 20 features are shown in the following table.

Table 1: Top 20 Features

Name	Score
Cingulum_Post_L	1.000000
Cerebelum_3_L	0.865000
Caudate_L	0.704000
Rolandic_Oper_R	0.637000
ParaHippocampal_R	0.570000
Vermis_8	0.472000
Angular_L	0.439000
Angular_R	0.417000
Vermis_9	0.408000
Temporal_Sup_L	0.398000
Lingual_L	0.392000
Cingulum_Ant_L	0.332000
Temporal_Inf_L	0.327000
Cingulum_Mid_L	0.277000
ParaHippocampal_L	0.262000
Caudate_R	0.256000
Cerebelum_8_R	0.245000
Cerebelum_4_5_L	0.233000
Frontal_Mid_Orb_L	0.232000
Hippocampus_L	0.225000

2 Matrix Completion

Answer:

I implement a simple version of hard impute base on SVD function. The images can be recovered very well with only 20-30 ranks.

The plot of Recovery grey images are shown in Figure 1, the recovery error of grey image is shown in Figure 2.

The plot of Recovery color images are shown in Figure 3, the recovery error of color image is shown in Figure 4.

Matlab code is uploaded to GitHub `https://github.com/cqwangding/CSE847`.

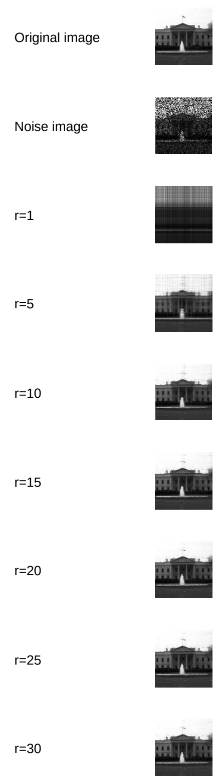

Figure 1: Recovery grey images with r = 1, 5, 10, 15, 20, 25, 30.

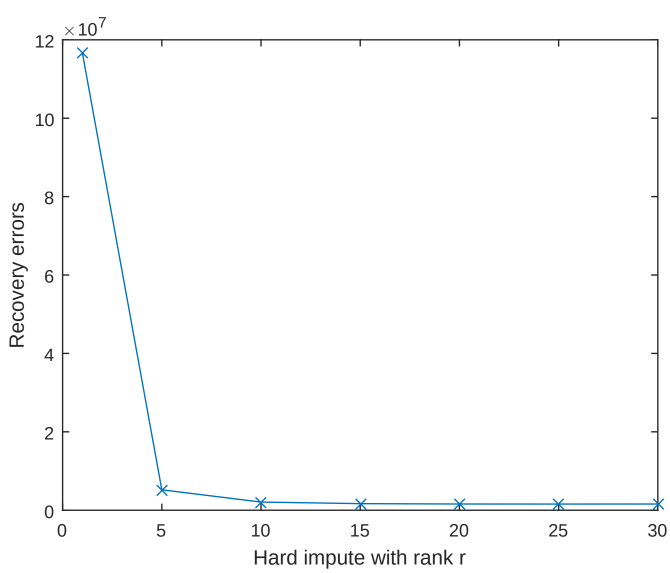

Figure 2: Recovery grey images error with r = 1, 5, 10, 15, 20, 25, 30.

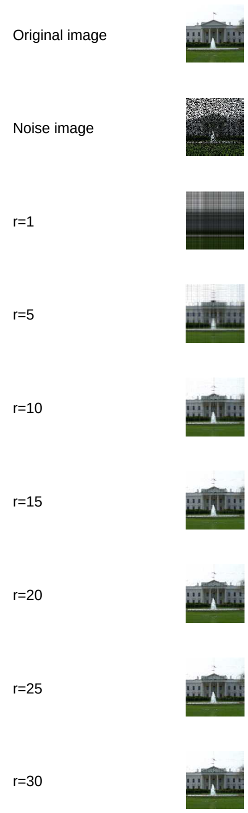

Figure 3: Recovery color images with r = 1, 5, 10, 15, 20, 25, 30.

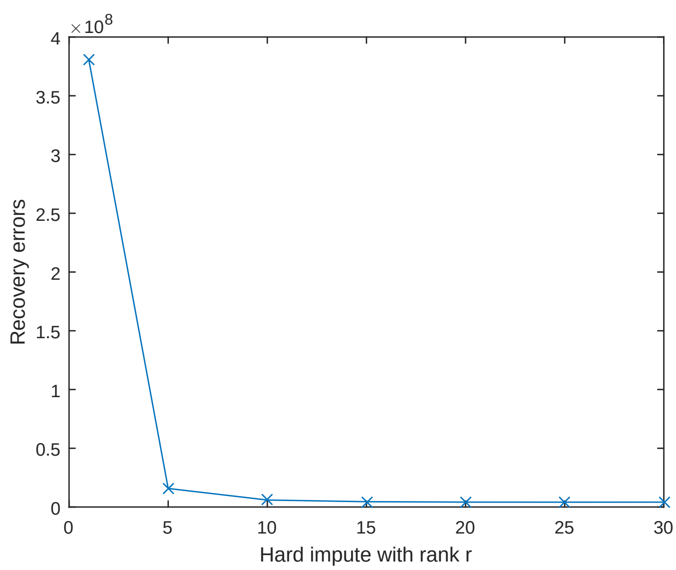

Figure 4: Recovery color images error with r = 1, 5, 10, 15, 20, 25, 30.