Final Project of MATP6610

Shuting Yang RIN: 661967441

May-07-2021

Topic I: Support Vector Machine

Part 2 The problem of soft-margin SVM, defined as

$$\text{minimize}_{w,b,t} \sum_{i=1}^{N} t_i + \frac{\lambda}{2}||w||^2, \tag{1}$$

subject to

$$y_i(w^T x_i + b) \geq 1 - t_i,\ t_i > 0,\ i = 1, 2, \cdots, N$$

is solved using the augmented Lagrangian Method (ALM).

We solve this inequality constraint problem by introducing slack variables $s_i, i = 1, 2, \cdots, N$, and convert it into an equality constraint. The soft-margin SVM problem then becomes

$$\text{minimize}_{w,b,t}(f(w,b,t) = \sum_{i=1}^{N} t_i + \frac{\lambda}{2}||w||^2),\ \text{s.t. } g_i(w,b,t) + s_i = 0,\ \text{and } s_i \geq 0, i = 1, 2, \cdots, N, \tag{2}$$

where

$$g_i(w,b,t) = 1 - t_i - y_i(w^T x_i + b).$$

Let $u \geq 0$ be the multiplier to $g_i + s_i$. The augmented Lagrangian function of (2) is

$$\mathcal{L}_\beta(w,b,t,s,u) = f(w,b,t) + \sum_{i=1}^{N} \left[u_i(g_i(w,b,t) + s_i) + \frac{\beta}{2}(g_i(w,b,t) + s_i)^2 \right],$$

i.e.,

$$\mathcal{L}_\beta(w,b,t,s,u) = \sum_{i=1}^{N} t_i + \frac{\lambda}{2}||w||^2 + \sum_{i=1}^{N} \left[u_i(1 - t_i - y_i(w^T x_i + b) + s_i) + \frac{\beta}{2}(1 - t_i - y_i(w^T x_i + b) + s_i)^2 \right],$$

by Quadratic Penalty.

An iterative approach of the ALM requires us to solve the following subproblems at the k-th iteration:

$$w^{(k+1)} = \text{argmin}_w \mathcal{L}_\beta(w, b^{(k)}, t^{(k)}, s^{(k)}, u^{(k)}) \tag{3}$$

$$b^{(k+1)} = \text{argmin}_b \mathcal{L}_\beta(w^{(k)}, b, t^{(k)}, s^{(k)}, u^{(k)}) \tag{4}$$

$$t^{(k+1)} = \text{argmin}_t \mathcal{L}_\beta(w^{(k)}, b^{(k)}, t, s^{(k)}, u^{(k)}) \tag{5}$$

$$s^{(k+1)} = \text{argmin}_s \mathcal{L}_\beta(w^{(k)}, b^{(k)}, t^{(k)}, s, u^{(k)}) \tag{6}$$

And the multiplier u is updated as

$$u_i^{(k+1)} = u_i^{(k)} + \beta(g_i(w^{(k)}, b^{(k)}, t^{(k)}) + s_i^{(k)}),\ i = 1, 2, \cdots, N.$$

Applying the Projected Gradient Method, we need to solve the following problems at the k-th iteration:

$$w^{(k+1)} = \text{Proj}_{w \in \mathbb{R}^p}(w^{(k,\beta)} - \alpha_k \nabla_w \mathcal{L}_\beta) \tag{7}$$

$$b^{(k+1)} = \text{Proj}_{b \in \mathbb{R}}(b^{(k,\beta)} - \alpha_k \nabla_b \mathcal{L}_\beta) \tag{8}$$

$$t^{(k+1)} = \text{Proj}_{t \in \mathbb{R}^N}(t^{(k,\beta)} - \alpha_k \nabla_t \mathcal{L}_\beta) \tag{9}$$

$$s^{(k+1)} = \text{Proj}_{s \in \mathbb{R}^N}(s^{(k,\beta)} - \alpha_k \nabla_s \mathcal{L}_\beta). \tag{10}$$

To choose an appropriate step size α in each of the subproblems (7), (8), (9), and (10), a Backtracking Line Search method (Algorithm 3.1 of textbook) is applied.

Use the update of w as an example.

Treating all other variables as constants.

Choose

$$\bar{\alpha} > 0,\ \rho \in (0, 1),\ c \in (0, 1);$$

Set

$$\alpha \leftarrow \bar{\alpha};$$

Repeat until

$$\mathcal{L}_\beta(w + \alpha p_k) \leq \mathcal{L}_\beta(w) + c\alpha \nabla \mathcal{L}_\beta^T \cdot p_k.$$

$$\alpha_k \leftarrow \bar{\alpha}$$

End repeat.
Terminate $\alpha_k = \alpha$.

Part 3. Using this Backtracking Line Search Method, the step size α for each of the subproblems (7), (8), (9), and (10) can be found. Apart from the above stopping condition

used in Line Search, the other two stopping conditions used in solving the subproblems in (7), (8), (9), and (10) are gradient error (grad-err) and maximum iterations. The project gradient method for solving w, b, t, s will stop if grad-err is less than the tolerance (sub-tol) of 10^{-3}. The stopping conditions of the outer loop are max(dual violation, primal violation) $\leq$ tolerance and maximum iteration time.

Specifically, when solving for w, b, t, and s, we set the gradients in the following (11)-(14) to zero to meet the optimality condition.

$$\nabla_w \mathcal{L}_\beta = \lambda w - \sum_{i=1}^{N} u_i y_i x_i - \beta \sum_{i=1}^{N} (1 - t_i - y_i(w^T x_i + b) + s_i) y_i x_i \tag{11}$$

$$\nabla_b \mathcal{L}_\beta = -\sum_{i=1}^{N} u_i y_i - \beta \sum_{i=1}^{N} y_i (1 - t_i - y_i(w^T x_i + b) + s_i) \tag{12}$$

$$\nabla_{t_i} \mathcal{L}_\beta = 1 - u_i - \beta(1 - t_i - y_i(w^T x_i + b) + s_i),\ i = 1, 2, \cdots, N \tag{13}$$

$$\nabla_{s_i} \mathcal{L}_\beta = u_i + \beta(1 - t_i - y_i(w^T x_i + b) + s_i),\ s_i \geq 0,\ i = 1, 2, \cdots, N. \tag{14}$$

To derive the dual feasibility and primal feasibility, we need to consider the KKT conditions.

$$\begin{cases} \nabla f + \sum_{i=1}^{N} u_i g_i = \overrightarrow{0} \ \text{ (dual feasibility)} \\ 1 - t_i - y_i(w^T x_i + b) + s_i = 0 \\ s_i \geq 0 \\ g_i \leq 0 \\ u_i g_i = 0,\ i = 1, 2, \cdots, N. \end{cases}$$

The violation of primary feasibility is

$$\varphi_i^{(k)} = 1 - t_i - y_i b + s_i - y_i x_i^T w^{(k)},\ i = 1, 2, \cdots, N.$$

By update rules of u_i,

$$u_i^{(k+1)} = u_i^{(k)} + \beta(1 - t_i - y_i(w^T x_i + b)).$$

The optimality conditions for each of (3)-(6) holds at

$$w^{(k+1)}:\ \nabla_w \mathcal{L}_\beta(w^{(k+1)}, b^{(k)}, t^{(k)}, s^{(k)}, u^{(k)}) = \overrightarrow{0}$$

$$b^{(k+1)}:\ \nabla_b \mathcal{L}_\beta(w^{(k)}, b^{(k+1)}, t^{(k)}, s^{(k)}, u^{(k)}) = \overrightarrow{0}$$

$$t^{(k+1)}:\ \nabla_t \mathcal{L}_\beta(w^{(k)}, b^{(k)}, t^{(k+1)}, s^{(k)}, u^{(k)}) = \overrightarrow{0}$$

$$s^{(k+1)}:\ \nabla_s \mathcal{L}_\beta(w^{(k)}, b^{(k)}, t^{(k)}, s^{(k+1)}, u^{(k)}) = \overrightarrow{0}.$$

Replacing the u_i in the above equations, we get

$$\lambda w^{(k+1)} - \sum_{i=1}^{N} u_i^{(k+1)} y_i x_i - 2\beta \sum_{i=1}^{N} (1 - t_i - y_i(w^T x_i + b) + s_i) y_i x_i = \overrightarrow{0} \tag{15}$$

$$-\sum_{i=1}^{N} u_i^{(k+1)} y_i = 0 \tag{16}$$

$$1 - u_i^{(k+1)} = 0 \tag{17}$$

$$u_i^{(k+1)} = 0 \tag{18}$$

Comparing (15)-(18) with (11)-(14), we derive the violation of dual feasibility as follows:

$$||\beta \sum_{i=1}^{N} (1 - t_i - y_i(w^T x_i + b) + s_i) y_i x_i|| + || \sum_{i=1}^{N} (1 - t_i - y_i(w^T x_i + b) + s_i)|| + 2||\beta \sum_{i=1}^{N} (1 - t_i - y_i(w^T x_i + b) + s_i)||.$$

The testing results are in the pages following.

Part 4-5: SVM-spam Test Result and Figures

Using a $\beta = 0.5$ and $\lambda = 1$ for ALM; c = 0.3 and dec-ratio = 0.5 for parameters in line search, the testing result for SVM-spam is as follows. Considering the length of print, the following printing result only included 20 outer iterations. Note that the classification result is satisfactory, but the computational time is long, which implies that the solver efficiency can be improved.

```
>> test_model_SVM_spam
Testing by student code

out iter = 1, pres = 1.5913e+01, dres = 7.7029e+03, subit = 20
out iter = 2, pres = 1.9653e+01, dres = 7.2852e+03, subit = 20
out iter = 3, pres = 2.4659e+01, dres = 6.9207e+03, subit = 20
out iter = 4, pres = 2.9430e+01, dres = 6.6608e+03, subit = 20
out iter = 5, pres = 3.3975e+01, dres = 6.4147e+03, subit = 20
out iter = 6, pres = 3.8469e+01, dres = 6.1799e+03, subit = 20
out iter = 7, pres = 4.2963e+01, dres = 5.9699e+03, subit = 20
out iter = 8, pres = 4.7484e+01, dres = 5.8101e+03, subit = 20
out iter = 9, pres = 5.2063e+01, dres = 5.6763e+03, subit = 20
out iter = 10, pres = 5.6713e+01, dres = 5.5477e+03, subit = 20
out iter = 11, pres = 6.1445e+01, dres = 5.4226e+03, subit = 20
out iter = 12, pres = 6.6201e+01, dres = 5.2982e+03, subit = 20
out iter = 13, pres = 7.0988e+01, dres = 5.1773e+03, subit = 20
out iter = 14, pres = 7.5842e+01, dres = 5.0672e+03, subit = 20
out iter = 15, pres = 8.0834e+01, dres = 4.9793e+03, subit = 20
out iter = 16, pres = 8.5810e+01, dres = 4.8936e+03, subit = 20
out iter = 17, pres = 9.0873e+01, dres = 4.8193e+03, subit = 20
out iter = 18, pres = 9.6033e+01, dres = 4.7560e+03, subit = 20
out iter = 19, pres = 1.0120e+02, dres = 4.6913e+03, subit = 20
out iter = 20, pres = 1.0641e+02, dres = 4.6273e+03, subit = 20
Running time is 7.5966
classification accuracy on testing data: 80.87%

Testing by instructor code

out iter = 1, pres = 5.8762e+01, dres = 1.2285e+02, subit = 20
```

```
out iter = 2, pres = 5.6312e+01, dres = 1.7191e+02, subit = 20
out iter = 3, pres = 5.2984e+01, dres = 2.9018e+02, subit = 20
out iter = 4, pres = 4.9673e+01, dres = 4.2253e+02, subit = 20
out iter = 5, pres = 4.5763e+01, dres = 6.1716e+02, subit = 20
out iter = 6, pres = 4.1862e+01, dres = 4.0074e+02, subit = 20
out iter = 7, pres = 3.6981e+01, dres = 5.7551e+02, subit = 20
out iter = 8, pres = 3.2761e+01, dres = 7.5702e+02, subit = 20
out iter = 9, pres = 2.7349e+01, dres = 6.3044e+02, subit = 20
out iter = 10, pres = 2.1775e+01, dres = 5.7407e+02, subit = 20
out iter = 11, pres = 1.6240e+01, dres = 6.6476e+02, subit = 20
out iter = 12, pres = 1.1068e+01, dres = 6.8514e+02, subit = 20
out iter = 13, pres = 8.7292e+00, dres = 7.2352e+02, subit = 20
out iter = 14, pres = 3.9052e+00, dres = 5.4746e+02, subit = 20
out iter = 15, pres = 3.0713e+00, dres = 6.9758e+02, subit = 20
out iter = 16, pres = 2.2853e+00, dres = 7.1635e+02, subit = 20
out iter = 17, pres = 1.9643e+00, dres = 6.0396e+02, subit = 20
out iter = 18, pres = 1.4222e+00, dres = 4.6273e+02, subit = 20
out iter = 19, pres = 8.2589e-01, dres = 4.2048e+02, subit = 20
out iter = 20, pres = 5.1938e-01, dres = 3.7364e+02, subit = 20
Running time is 1.2367
classification accuracy on testing data: 78.87%
```

Part 4: SVM-rho02 Test Result

Using a $\beta = 0.05$ and $\lambda = 0.1$ for ALM; c $= 0.2$ and dec-ratio $= 0.5$ for parameters in line search, the testing result for SVM-rho02 is as follows. Considering the length of print, the following printing result only included 20 outer iterations. Note that the classification result is satisfactory. But the computational time is also long, which implies that the solver efficiency can be improved.

```
>> test_model_SVM_rho02
Testing by student code

out iter = 1, pres = 6.0296e+02, dres = 3.7561e+04, subit = 20
out iter = 2, pres = 6.2264e+02, dres = 3.5346e+04, subit = 20
```

```
out iter = 3, pres = 6.4119e+02, dres = 3.3164e+04, subit = 20
out iter = 4, pres = 6.5899e+02, dres = 3.1088e+04, subit = 20
out iter = 5, pres = 6.7609e+02, dres = 2.9114e+04, subit = 20
out iter = 6, pres = 6.9252e+02, dres = 2.7238e+04, subit = 20
out iter = 7, pres = 7.0830e+02, dres = 2.5453e+04, subit = 20
out iter = 8, pres = 7.2348e+02, dres = 2.3756e+04, subit = 20
out iter = 9, pres = 7.3808e+02, dres = 2.2144e+04, subit = 20
out iter = 10, pres = 7.5212e+02, dres = 2.0612e+04, subit = 20
out iter = 11, pres = 7.6563e+02, dres = 1.9158e+04, subit = 20
out iter = 12, pres = 7.7864e+02, dres = 1.7781e+04, subit = 20
out iter = 13, pres = 7.9117e+02, dres = 1.6475e+04, subit = 20
out iter = 14, pres = 8.0323e+02, dres = 1.5238e+04, subit = 20
out iter = 15, pres = 8.1484e+02, dres = 1.4068e+04, subit = 20
out iter = 16, pres = 8.2603e+02, dres = 1.2958e+04, subit = 20
out iter = 17, pres = 8.3680e+02, dres = 1.1909e+04, subit = 20
out iter = 18, pres = 8.4716e+02, dres = 1.0920e+04, subit = 20
out iter = 19, pres = 8.5711e+02, dres = 9.9894e+03, subit = 20
out iter = 20, pres = 8.6667e+02, dres = 9.1170e+03, subit = 20
Running time is 2.3713
classification accuracy on testing data: 91.00%

Testing by instructor code

out iter = 1, pres = 0.0000e+00, dres = 2.1968e+00, subit = 20
out iter = 2, pres = 0.0000e+00, dres = 1.1382e+00, subit = 20
out iter = 3, pres = 0.0000e+00, dres = 2.3946e-01, subit = 20
out iter = 4, pres = 0.0000e+00, dres = 2.1082e-01, subit = 20
out iter = 5, pres = 0.0000e+00, dres = 1.9195e-01, subit = 20
out iter = 6, pres = 0.0000e+00, dres = 1.7443e-01, subit = 20
out iter = 7, pres = 1.8128e-03, dres = 1.5715e-01, subit = 20
out iter = 8, pres = 0.0000e+00, dres = 1.4038e-01, subit = 20
out iter = 9, pres = 0.0000e+00, dres = 1.2686e-01, subit = 20
out iter = 10, pres = 5.7565e-04, dres = 1.1427e-01, subit = 20
out iter = 11, pres = 6.7911e-04, dres = 1.0713e-01, subit = 20
out iter = 12, pres = 0.0000e+00, dres = 9.6717e-02, subit = 20
out iter = 13, pres = 3.6957e-04, dres = 9.1574e-02, subit = 20
out iter = 14, pres = 5.3285e-04, dres = 8.7507e-02, subit = 20
```

```
out iter = 15, pres = 9.0454e-04, dres = 8.2353e-02, subit = 20
out iter = 16, pres = 4.6719e-04, dres = 7.7391e-02, subit = 20
out iter = 17, pres = 0.0000e+00, dres = 7.3662e-02, subit = 20
out iter = 18, pres = 9.7905e-04, dres = 7.3561e-02, subit = 20
out iter = 19, pres = 4.5927e-04, dres = 6.7546e-02, subit = 20
out iter = 20, pres = 3.5000e-06, dres = 6.5227e-02, subit = 20
Running time is 0.3918
classification accuracy on testing data: 95.00%
```

Part 4: SVM-rho08 Test Result

Using a $\beta = 0.2$ and $\lambda = 0.1$ for ALM; c $= 0.45$ and dec-ratio $= 0.5$ for parameters in line search, the testing result for SVM-rho08 is as follows. Considering the length of print, the following printing result only included 20 outer iterations. Note that the classification result is not very satisfactory. The computational time is also long, which implies that the solver efficiency can be improved.

```
>> test_model_SVM_rho08
Testing by student code

out iter = 1, pres = 4.2137e+02, dres = 5.4505e+04, subit = 20
out iter = 2, pres = 4.2648e+02, dres = 3.3706e+04, subit = 20
out iter = 3, pres = 4.2896e+02, dres = 2.2022e+04, subit = 20
out iter = 4, pres = 4.3298e+02, dres = 1.4830e+04, subit = 20
out iter = 5, pres = 4.3743e+02, dres = 1.0156e+04, subit = 20
out iter = 6, pres = 4.4197e+02, dres = 7.0361e+03, subit = 20
out iter = 7, pres = 4.4638e+02, dres = 4.8796e+03, subit = 20
out iter = 8, pres = 4.5069e+02, dres = 3.3364e+03, subit = 20
out iter = 9, pres = 4.5495e+02, dres = 2.2667e+03, subit = 20
out iter = 10, pres = 4.5914e+02, dres = 1.5177e+03, subit = 20
out iter = 11, pres = 4.6329e+02, dres = 1.0144e+03, subit = 20
out iter = 12, pres = 4.6742e+02, dres = 6.8076e+02, subit = 20
out iter = 13, pres = 4.7159e+02, dres = 4.7717e+02, subit = 20
out iter = 14, pres = 4.7581e+02, dres = 3.4866e+02, subit = 20
out iter = 15, pres = 4.8013e+02, dres = 2.6447e+02, subit = 20
```

```
out iter = 16, pres = 4.8454e+02, dres = 2.2514e+02, subit = 20
out iter = 17, pres = 4.8874e+02, dres = 1.5442e+02, subit = 4
out iter = 18, pres = 4.9309e+02, dres = 1.4545e+02, subit = 4
out iter = 19, pres = 4.9751e+02, dres = 1.4145e+02, subit = 4
out iter = 20, pres = 5.0202e+02, dres = 1.3577e+02, subit = 4
Running time is 1.8118
classification accuracy on testing data: 68.00%

Testing by instructor code

out iter = 1, pres = 0.0000e+00, dres = 2.0617e+00, subit = 20
out iter = 2, pres = 0.0000e+00, dres = 1.1143e+00, subit = 20
out iter = 3, pres = 4.1960e-04, dres = 9.6344e-01, subit = 20
out iter = 4, pres = 0.0000e+00, dres = 8.0175e-01, subit = 20
out iter = 5, pres = 4.7455e-04, dres = 6.8857e-01, subit = 20
out iter = 6, pres = 4.7980e-03, dres = 5.5524e-01, subit = 20
out iter = 7, pres = 1.4519e-03, dres = 5.0271e-01, subit = 20
out iter = 8, pres = 0.0000e+00, dres = 4.3816e-01, subit = 20
out iter = 9, pres = 0.0000e+00, dres = 3.7695e-01, subit = 20
out iter = 10, pres = 0.0000e+00, dres = 3.5100e-01, subit = 20
out iter = 11, pres = 0.0000e+00, dres = 3.3167e-01, subit = 20
out iter = 12, pres = 0.0000e+00, dres = 3.1579e-01, subit = 20
out iter = 13, pres = 4.7800e-04, dres = 3.0103e-01, subit = 20
out iter = 14, pres = 3.0787e-03, dres = 2.9638e-01, subit = 20
out iter = 15, pres = 0.0000e+00, dres = 2.8099e-01, subit = 20
out iter = 16, pres = 2.3343e-03, dres = 2.7290e-01, subit = 20
out iter = 17, pres = 4.4062e-04, dres = 2.5993e-01, subit = 20
out iter = 18, pres = 1.5581e-03, dres = 2.5460e-01, subit = 20
out iter = 19, pres = 6.9371e-04, dres = 2.5084e-01, subit = 20
out iter = 20, pres = 8.7837e-04, dres = 2.4451e-01, subit = 20
Running time is 0.2020
classification accuracy on testing data: 80.50%
```

The output figures of SVM-spam, SVM-rho02 and SVM-08 are in the pages following.

Figure 1: SVM-spam

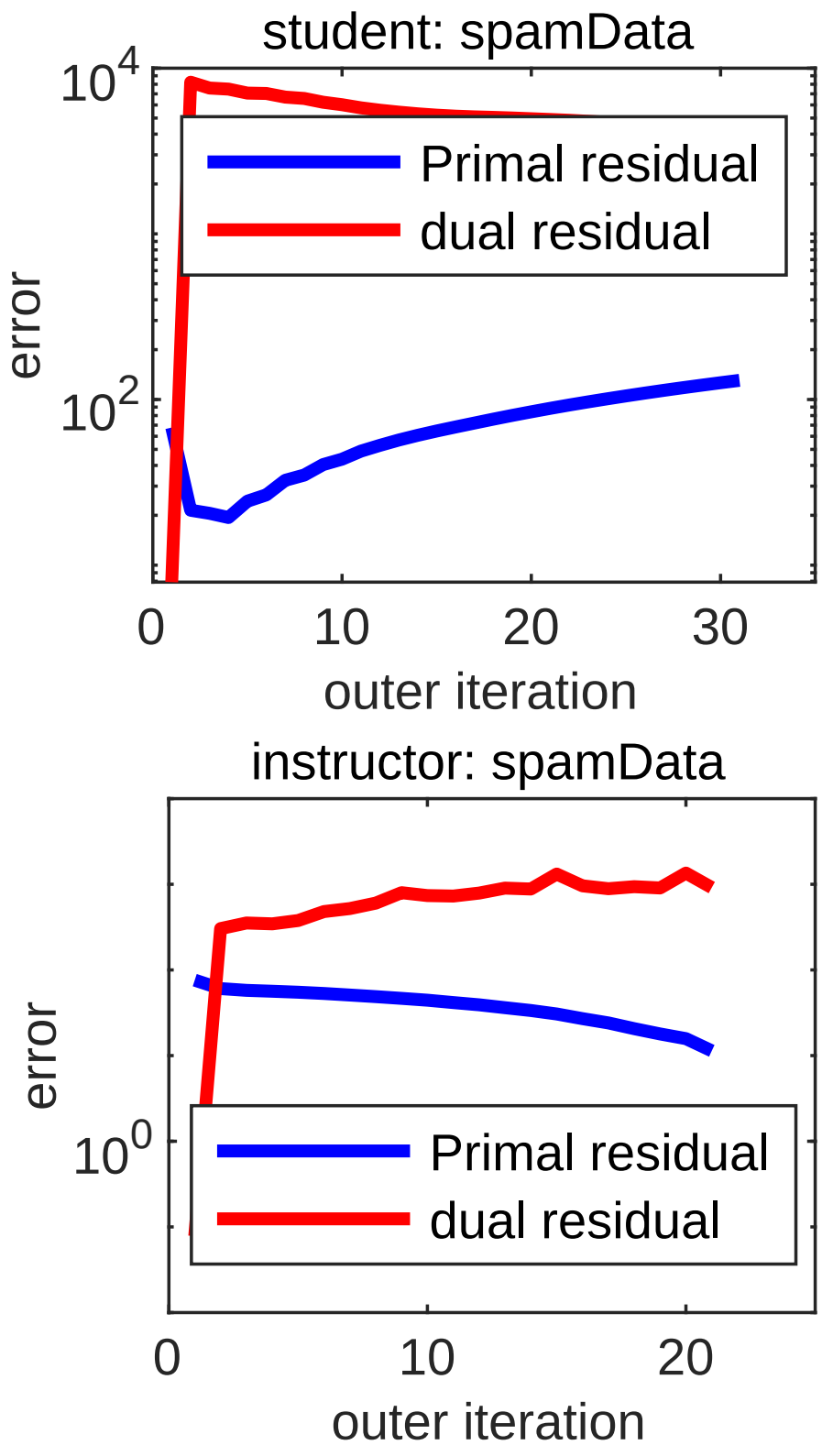


Figure 2: SVM-rho02

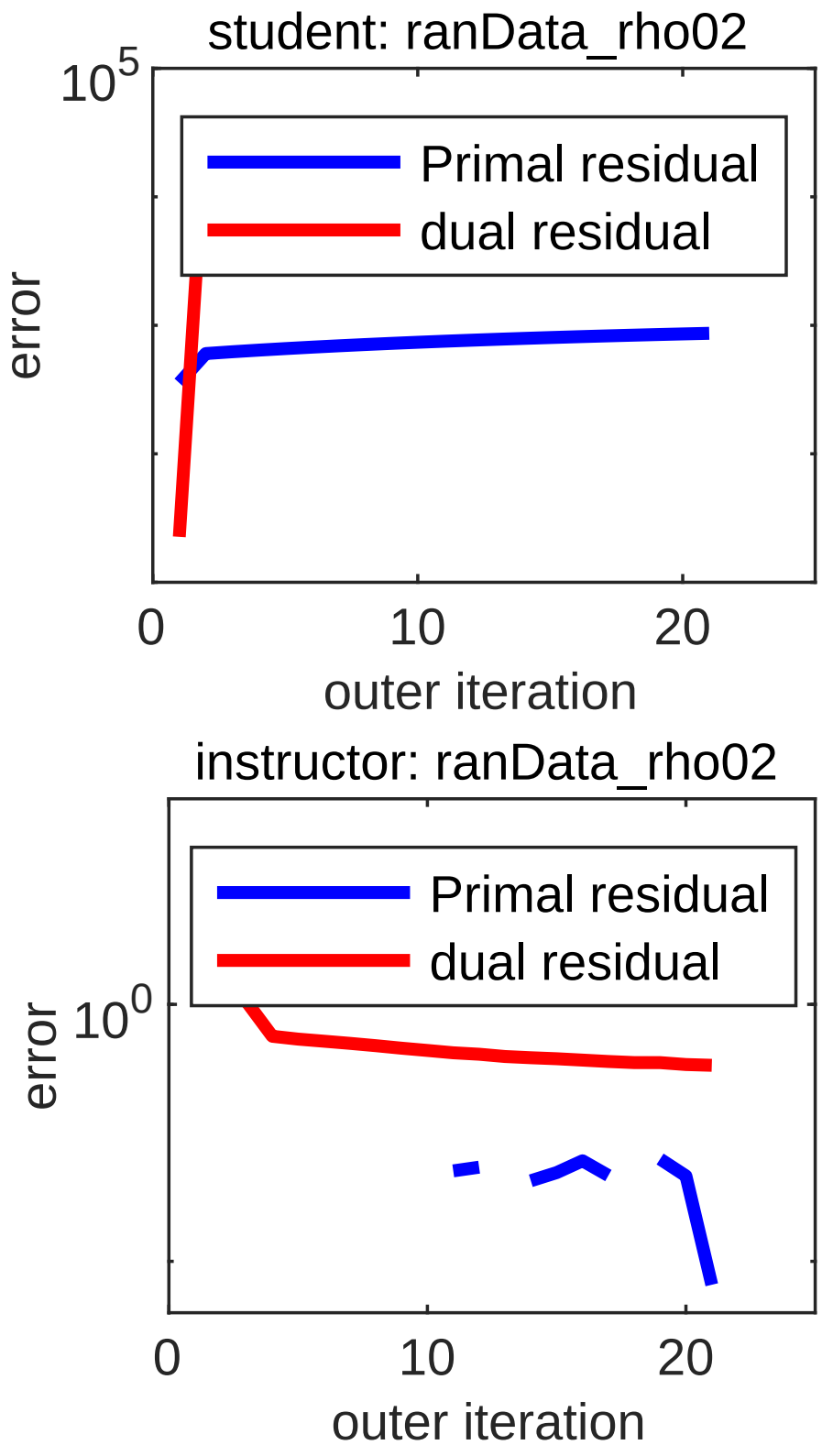
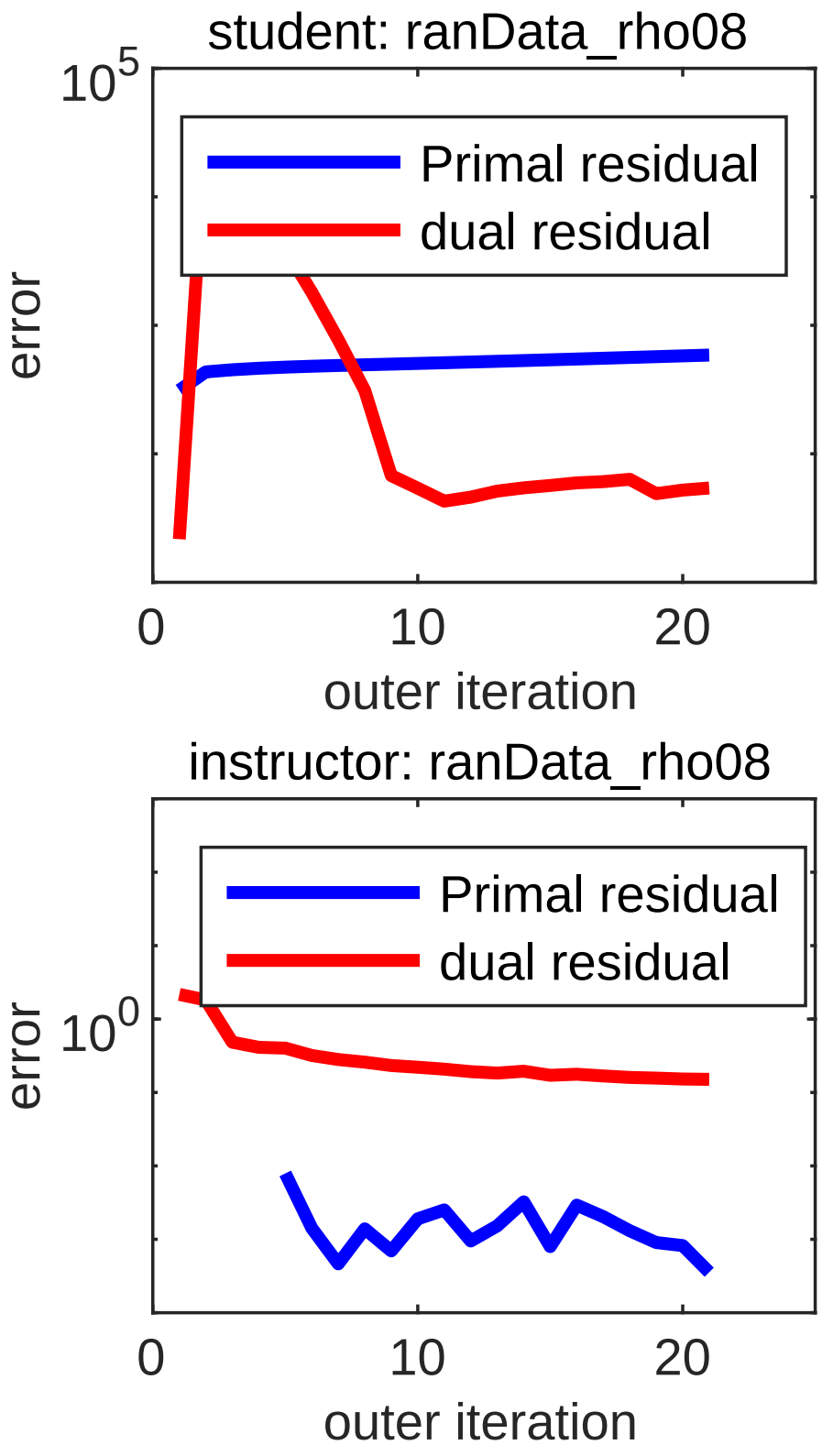


Figure 3: SVM-rho08



Topic II(Bonus)

We use alternating minimization method. Let

$$L(x,y,v) = \frac{1}{2}||XY - M||_F^2 + \sum_{i=1}^{r}[v_i(\sum_{j=1}^{n} Y_{ij} - 1) + \frac{\beta}{2}(\sum_{j=1}^{n} Y_{ij} - 1)^2],\ \beta > 0.$$

The problem is to solve

$$\text{minimize}_{X,Y} L(X,Y,v)$$

Let

$$f(X,Y) = \frac{1}{2}||XY - M||_F^2,\ h_i(X,Y) = \sum_{j=1} nY_{ij} - 1.$$

Dual feasibility for X:

$$\nabla_X f(X,Y) + \sum_{i=1}^{r} u_i \nabla_X h_i(X,Y) = \nabla_X f(X,Y) = 0.$$

Dual feasibility for Y:

$$\nabla_Y f(X,Y) + \sum_{i=1}^{r} u_i \nabla_Y h_i(X,Y) = 0.$$

Primary feasibility:

$$h_i(X,Y) = \sum_{j=1}^{n} Y_{ij} - 1 = 0,\ i = 1, 2, \cdots, r.$$