資工系100062236林修安

Machine Learning Term Project 1 Report

- **Objective Rewritten**

Primal objective:
$$\arg\min_{w,b,\xi} \|w\|_2 + C\sum_{i=1}^{N}\xi_i$$
$$\text{subject to } r^{(i)}(w^T x^{(i)} - b) > 1 - \xi_i, \text{and } \xi_i \geq 0, i = 1,\ldots,N$$

can be rewritten into the following standard QP form:

$$\min_x x^T Q x + c^T x$$
$$\text{subject to } Gx \leq h$$

by setting

$$x = \left[w^T, b, \xi_1, \ldots, \xi_N\right]^T \in \mathbb{R}^{d+N+1}$$

$$Q = \begin{pmatrix} I_d & \mathbf{0} & 0 \\ \mathbf{0} & \ddots & \mathbf{0} \\ 0 & \mathbf{0} & 0 \end{pmatrix} \in \mathbb{R}^{(d+N+1)\times(d+N+1)}$$

$$c = C \times \begin{bmatrix} \mathbf{0} \\ 0 \\ \mathbf{1} \end{bmatrix} \in \mathbb{R}^{(d+N+1)}$$

$$G = \begin{bmatrix} -r^{(1)}\mathbf{x}^{(1)T} & r^{(1)} & -1 & 0 & 0 \\ \vdots & \vdots & 0 & \ddots & 0 \\ -r^{(N)}\mathbf{x}^{(N)T} & r^{(N)} & 0 & 0 & -1 \\ 0 & 0 & -1 & 0 & 0 \\ \vdots & \vdots & 0 & \ddots & 0 \\ 0 & 0 & 0 & 0 & -1 \end{bmatrix} \in \mathbb{R}^{2N\times(d+N+1)}$$

$$h = \begin{bmatrix} \mathbf{-1} \\ \mathbf{0} \end{bmatrix} \in \mathbb{R}^{2N}$$

By combining the variables in one objective matrix, we can obtain w, b, ξ using simpler objective and constraint.

SMO Implementation

The SMO algorithm is performed in `train(X,y)`:

- Initialize $\alpha = 0, g = 1, b = 0$, and the box constraint C
- For instances labeled 1, set their A=0 and B=C;

  for instances labeled -1, set their A=-C and B=0.
- Lift input into feature space using kernel function

Loop until termination {

1. Calculate I_{up} as the set of i of instances such that $r^{(i)}\alpha_i < B^{(i)}$ and I_{down} as the set of j of instances such that $r^{(j)}\alpha_j > A^{(j)}$
2. Find i and j such that $\arg\max_{i \in I_{up}} r^{(i)} g^{(i)}$ and $\arg\min_{j \in I_{down}} r^{(j)} g^{(j)}$
3. SMO terminates if $r^{(i)} g^{(i)} - r^{(j)} g^{(j)} \le 10^{-4}$ {

   (1) Set $b = \dfrac{r^{(i)} g^{(i)} - r^{(j)} g^{(j)}}{2}$ since the two values converge to ρ and $b = \rho$

   (2) Find the indices of support vectors

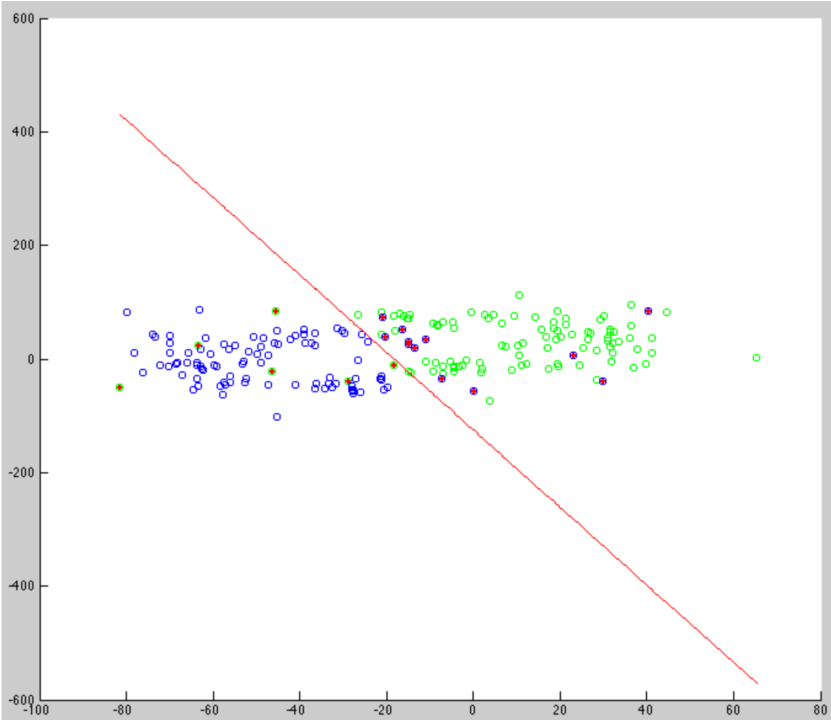
   (3) Multiply the α of support vectors by their labels as the new α in order to calculate w afterward

   }
4. Calculate the search direction $\lambda = \min\left\{ B^{(i)} - r^{(i)}\alpha_i, r^{(j)}\alpha_j - A^{(j)}, \dfrac{r^{(i)} g^{(i)} - r^{(j)} g^{(j)}}{K_{ii} + K_{jj} - 2K_{ij}} \right\}$
5. Update gradient $g^{(t)} = g^{(t)} - \lambda r^{(t)} (K_{it} - K_{jt})$
6. Update α, $\begin{matrix} \alpha_i = \alpha_i + \lambda r^{(i)} \\ \alpha_j = \alpha_j + \lambda r^{(j)} \end{matrix}$

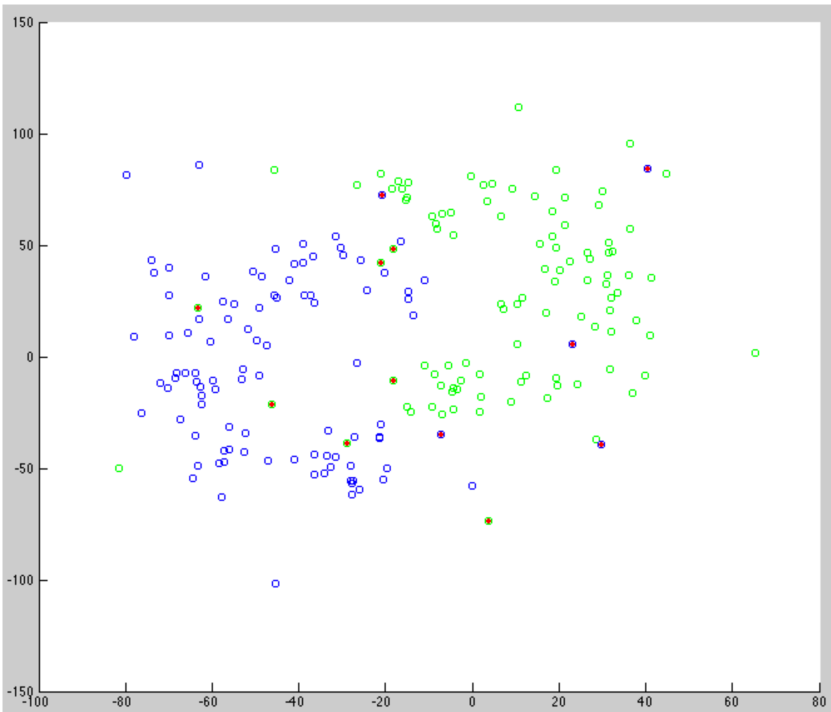
}

In `predict(X)`, map the testing instances also to feature space, and predict the labels using support vectors, and the w and b. If the predicted label is 0, set it to 1.

● **SoftMarginLinearClassifier Decision Boundary Result**

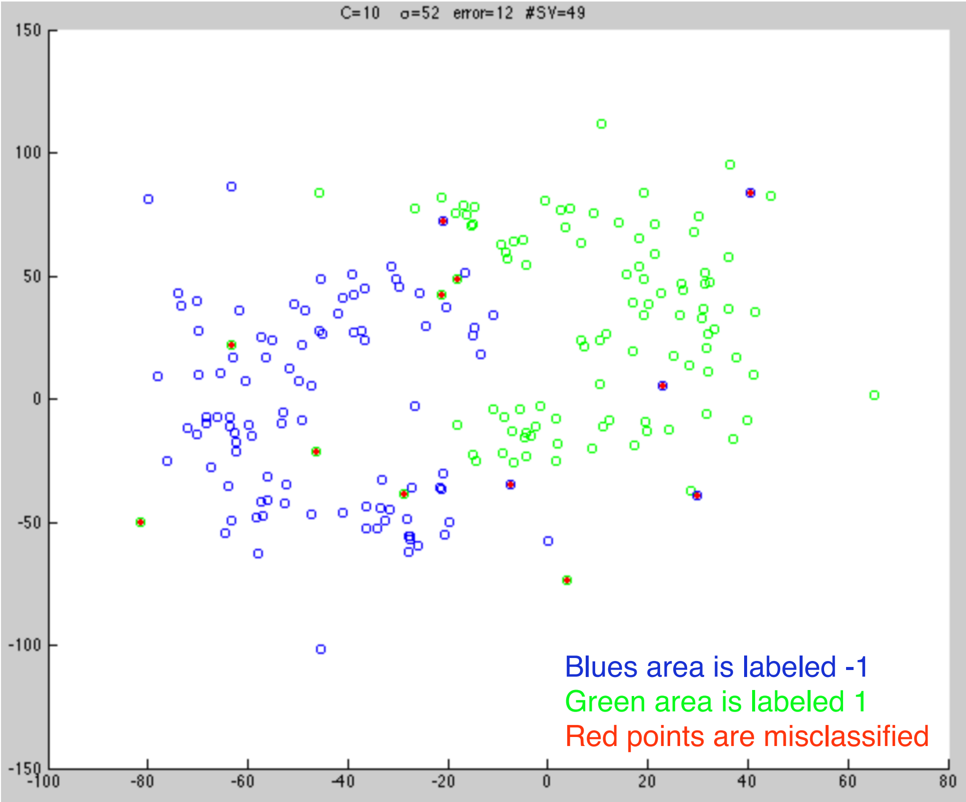


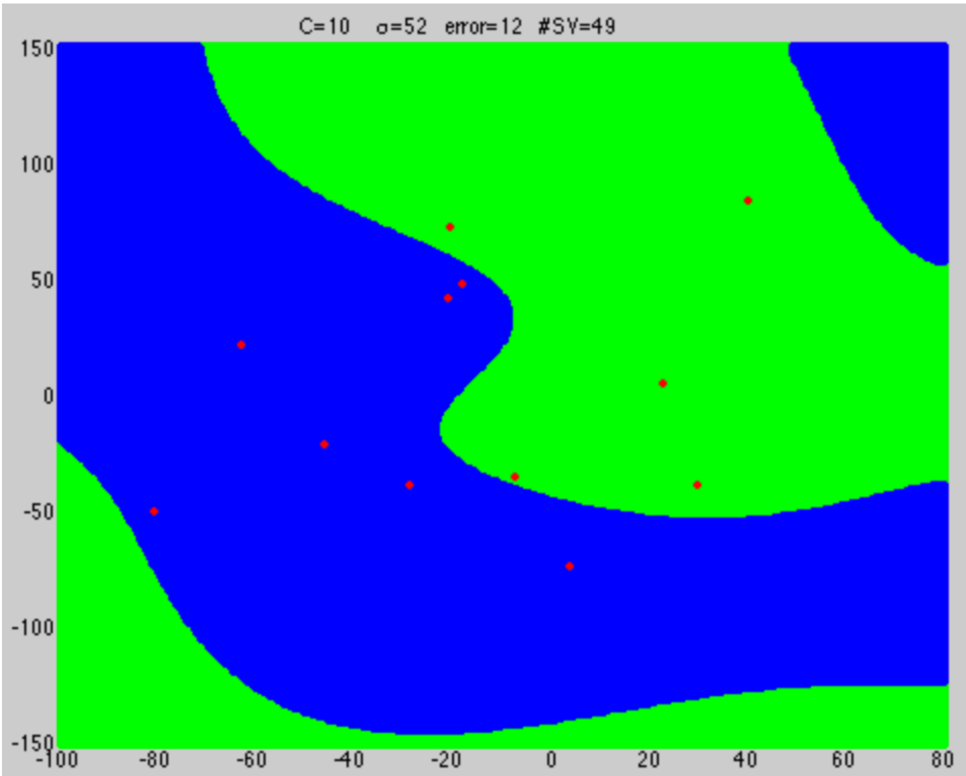
The linear classifier cannot perfectly categorize the points into correct groups, yielding an error of 18.



After lifting the input to feature space using polynomial kernel function with $\gamma = 3$, the error can be reduced to 12.

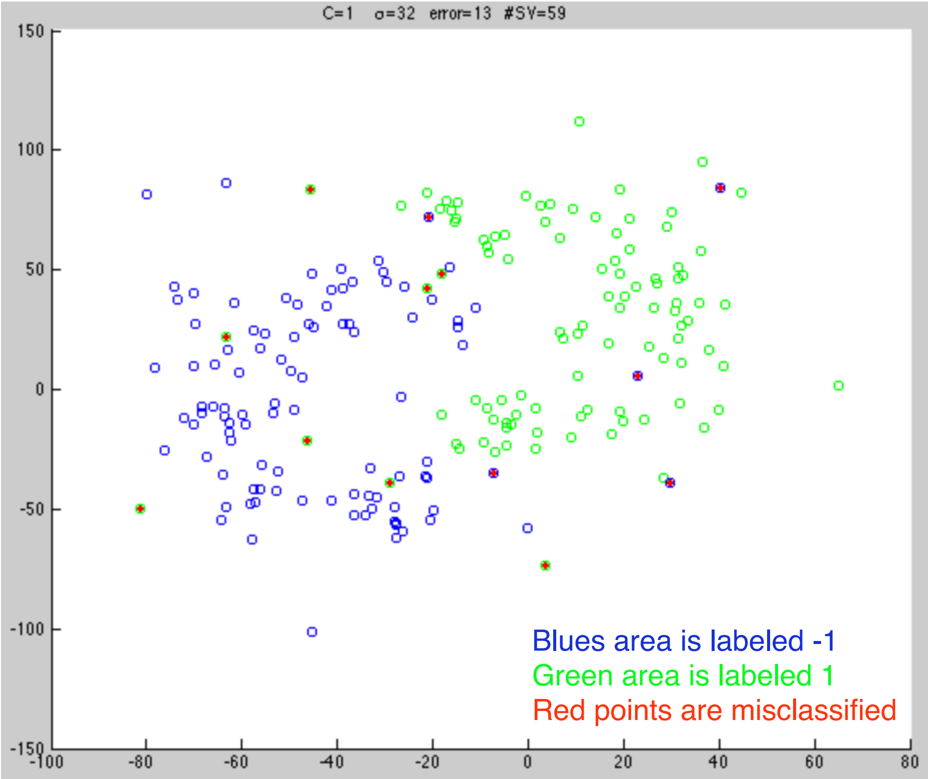
● **SMOClassifier Decision Boundary Result**

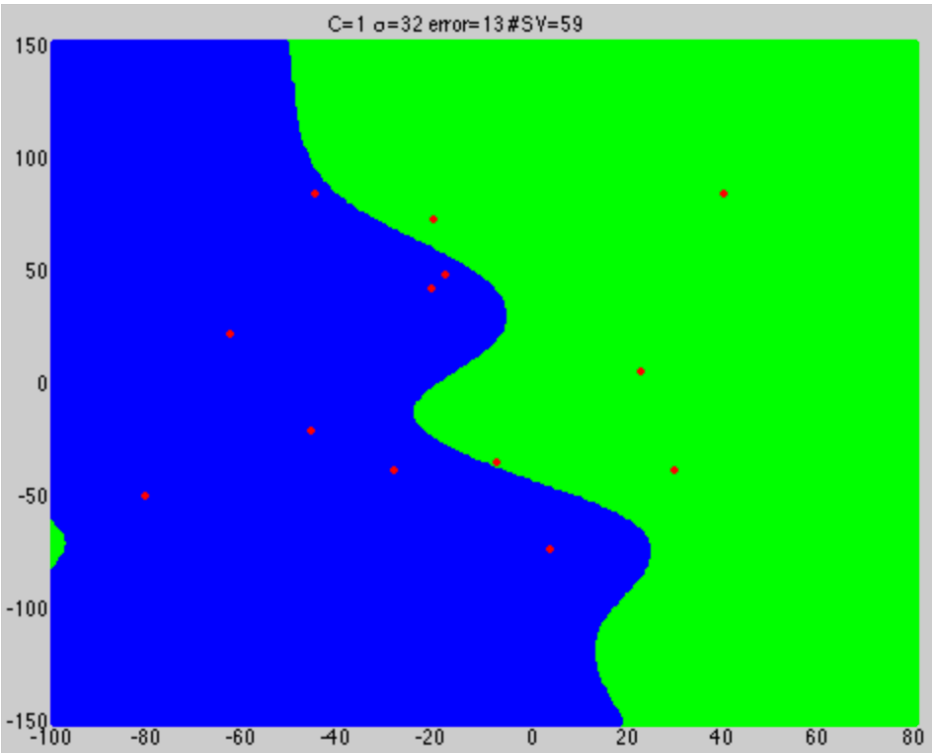




The above figures are results of parameter setting C=10 and σ=52, the error is 12.

But by cross validation with K=10, the CV error = 14, so the model is overfitting.





The above figures are results of parameter setting C=1 and σ=32, the error is 13.

By cross validation with K=10, the CV error =13, being consistent with empirical error.

Moreover, with leave-one-out cross validation, the error is also 13, so the model is more reasonable.

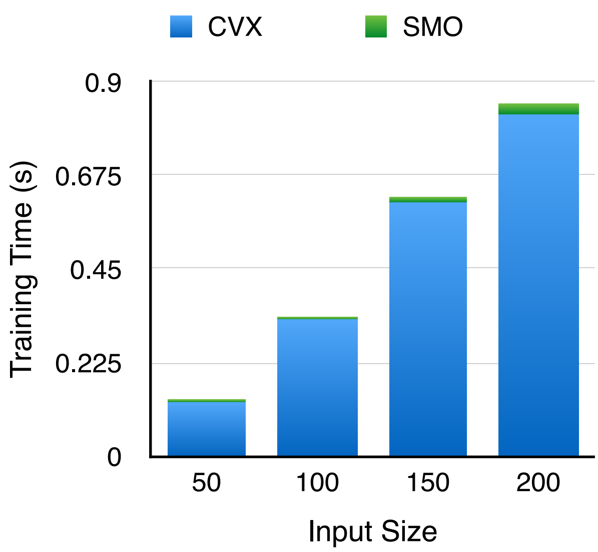
Performance Analysis

Comparing Error Between Classifiers

SoftMarginLinearClassifier give an error of 18, and SMOClassifier gives an error of 13. Since SMOClassifier lifts the input instances to high-dimensional feature space, the instances are more likely to be separated while SoftMarginLinearClassifier only uses a linear model to perform classification.

Comparing Training Time Between Classifiers with Different Training Set Size

Input Size \ Method	CVX	SMO
50	0.129028	0.007982
100	0.325593	0.011893
150	0.610848	0.014821
200	0.823635	0.023462



- Bigger the input size, slower the training time.
- SMO outperforms CVX in training speed.