CPS 571 — HW 4

Shengxin Qian, sq16

1 logistic Regression and Kernels

1.1 Define the reproducing kernel Hilbert space

The kernel Hilbert space of l_2 regularized logistic regression is $k(x, z) = \langle x, z \rangle_{H_k} = x^T * z$, obviously inner product is a valid kernel Hilbert space.
According to the representer theorem, $f^* = \sum_{i=1}^{n} \alpha_i k(x_i, .)$. It represents solving this optimization problem, we only need to solve the α_i and the kernel function could represent the inner product of vectors in Hilbert space.

1.2 Compare logistic loss function and hinge loss function

As we can see in Figure 1, the logistic loss is the approximation of hinge loss, especially when ζ is close to zero.

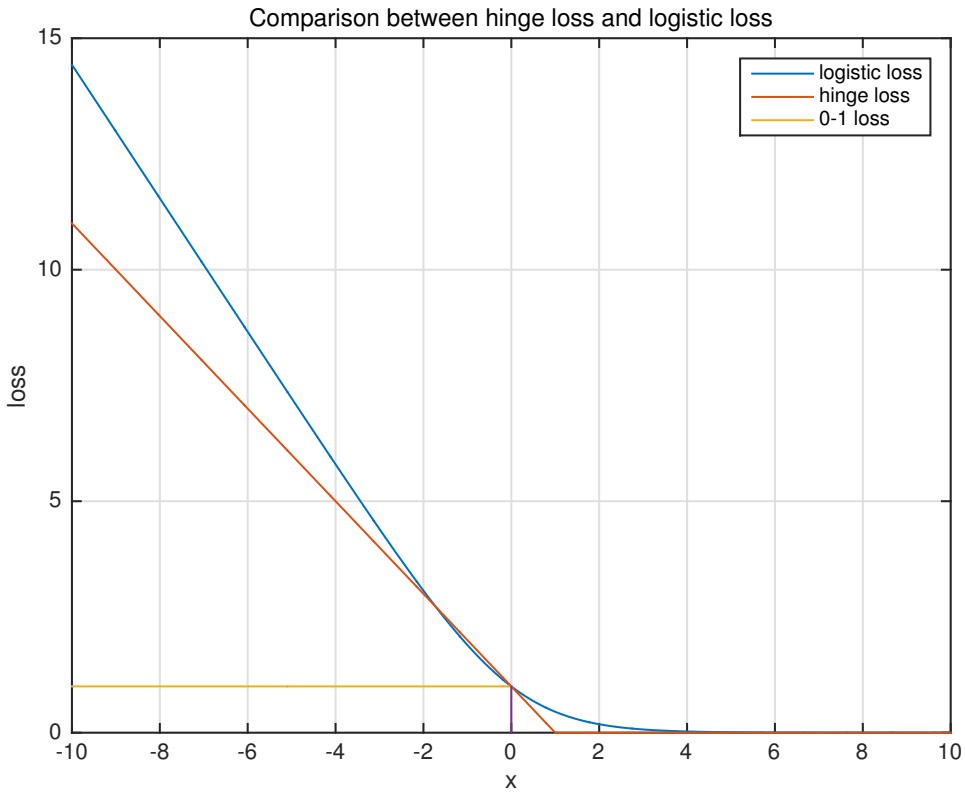

Figure 1: Comparison between logistic loss and hinge loss

1.3 Compare the dual formulation with non-separable SVM

The primal formulation of l_2 regularized logistic regression is:

$$\begin{aligned} \min_{\theta,\ \theta_0,\ \zeta} \max_{\alpha} \ell(\theta, \theta_0, \zeta, \alpha) \ &= \frac{1}{2} \parallel \theta \parallel^2 + C \sum_{i=1}^{n} g(\zeta_i) + \sum_{i=1}^{n} \alpha_i (\zeta_i - y_i (f(x_i) + \theta_0)) \\ & \textit{subject to} \\ & \alpha_i \geq 0, \forall_i \end{aligned} \tag{1}$$

The one of the KKT condition is
Lagrangian stationary

$$\begin{aligned} \frac{\partial \ell}{\partial \theta_0} = 0 \Rightarrow \sum_{i=1}^{n} \alpha_i y_i = 0 \\ \frac{\partial \ell}{\partial \theta} = 0 \Rightarrow \theta^* = \sum_{i=1}^{n} \alpha_i y_i x_i \\ \frac{\partial \ell}{\partial \zeta_i} = 0 \Rightarrow \zeta_i = \ln(\frac{C - \alpha_i}{\alpha_i}) \end{aligned} \tag{2}$$

The dual formulation derived from KKT is

$$\begin{aligned} \max_{\alpha} \ell(\theta^*, \theta_0^*, \zeta^*, \alpha) = -\frac{1}{2} (\sum_{i=1}^{n} \alpha_i y_i x_i)^2 + C \sum_{i=1}^{n} \ln \frac{C}{C - \alpha_i} + \sum_{i=1}^{n} \alpha_i \ln \frac{C - \alpha_i}{\alpha_i} \\ \textit{subject to} \\ 0 \leq \alpha_i \leq C \\ \sum_{i=1}^{n} \alpha_i y_i = 0 \end{aligned} \tag{3}$$

The dual formulation of non-separable SVM derived from KKT is

$$\begin{aligned} \max_{\alpha} \ell(\theta^*, \theta_0^*, \zeta^*, \alpha) = -\frac{1}{2} (\sum_{i=1}^{n} \alpha_i y_i x_i)^2 + \sum_{i=1}^{n} \alpha_i \\ \textit{subject to} \\ 0 \leq \alpha_i \leq C \\ \sum_{i=1}^{n} \alpha_i y_i = 0 \end{aligned} \tag{4}$$

As we can see from the two dual formulations above, the similarity is that the restraints are the same. The difference is the function need to be maximized is different.

2 SVM - Properties of the Maximum Margin Hyperplane

2.1 analytically result

Given optimization problem:

$$\begin{aligned} &\frac{1}{2} \parallel \omega \parallel^2 \\ &subject\ to: \\ &1 - y_i(\omega^t x_i + b) \leq 0 \end{aligned} \tag{5}$$

We can transform it into the primal problem:

$$\begin{aligned} \min_{\omega,\ b} \max_{\alpha} \ell(\omega, b, \alpha) \ &= \frac{1}{2} \parallel \omega \parallel^2 + \sum_{i=0}^{1} \alpha_i [1 - y_i(\omega^t x_i + b)] \\ &subject\ to\ : \\ &\alpha_i \geq 0, \forall_i \end{aligned} \tag{6}$$

According to "Lagrangian stationary"

$$\begin{aligned} &\frac{\partial \ell}{\partial b} = 0 \Rightarrow 0 = \sum_{i=0}^{1} \alpha_i y_i \Rightarrow \alpha_0 = \alpha_1 = \alpha \\ &\frac{\partial \ell}{\partial \omega} = 0 \Rightarrow \omega^* = \sum_{i=0}^{1} \alpha_i y_i x_i \Rightarrow \omega^* = \alpha \sum_{i=0}^{1} y_i x_i \end{aligned} \tag{7}$$

According to KKT condition, we can transform the primal problem to dual problem

$$\begin{aligned} \max_{\alpha} \ell(\omega, b, \alpha) \ \min_{\omega,\ b} \ &= \ \frac{1}{2} \parallel \omega \parallel^2 + \sum_{i=0}^{1} \alpha_i [1 - y_i(\omega^t x_i + b)] \\ \max_{\alpha} \ell(\omega^*, b^*, \alpha) = \ &\frac{1}{2} \parallel \omega^* \parallel^2 + \sum_{i=0}^{1} \alpha [1 - y_i(\omega^{*t} x_i + b^*)] \\ &= 2\alpha - \frac{1}{2}\alpha^2 \parallel x_1 - x_0 \parallel^2 \end{aligned} \tag{8}$$

The result is

$$\alpha^* = \frac{2}{\parallel x_1 - x_0 \parallel^2} \Rightarrow \omega^* = \frac{2(x_1 - x_0)}{\parallel x_1 - x_0 \parallel^2} \tag{9}$$

Because $\alpha_0 = \alpha_1 \neq 0$, both points are support vectors

$$\begin{cases} \omega^T x_1 + b = 1 \\ \omega^T x_0 + b = -1 \end{cases} \Rightarrow \omega^T (x_1 + x_0) + 2b = 0$$
$$b^* = \frac{(x_0 - x_1)^T (x_1 + x_0)}{\| x_1 - x_0 \|^2} \tag{10}$$

2.2 Essence of finding the maximum margin hyperplane

Essentially, finding the maximum margin hyperplane is solving the following question.

$$\begin{aligned} &\frac{1}{2} \| \omega \|^2 \\ &subject\ to: \\ &1 - y_i(\omega^t x_i + b) \leq 0 \end{aligned} \tag{11}$$

Obviously the first part $\frac{1}{2} \| \omega \|^2$ is a convex function. The second part $1 - y_i(\omega^t x_i + b)$ is an affine function (both convex and concave). That is why finding the maximum margin hyperplane is a convex optimization problem.

3 SVM Experiments

3.1 Toy Separable SVM

In order to solve dual problem

$$\max_{\alpha} \ell(\omega^*, b^*, \alpha) = \frac{1}{2} \parallel \omega^* \parallel^2 + \sum_{i=0}^{1} \alpha[1 - y_i(\omega^{*t} x_i + b^*)] \tag{12}$$

We need to use quadratic programming solver to solve a problem specified by

$$\begin{aligned} \min_{x} \frac{1}{2} x^T H x + f^T x \\ subject\ to \\ A * x \leq b \\ Aeq * x = beq \end{aligned} \tag{13}$$

In order to match the form of standard question, we can transform the dual problem into

$$\begin{aligned} \min_{\alpha} \frac{1}{2} \alpha^T (y y^T .* x x^T) \alpha - I^T \alpha \\ subject\ to \\ -I_n \alpha \leq 0 \\ y^T x = 0 \end{aligned} \tag{14}$$

We can get the vector α from the matlab quadratic solver and then derive ω, b from α.

$$\begin{aligned} \omega^* &= (\alpha .* y)^T x \\ b^* &= -\frac{\max_{H_0} \omega^{*T} x_i + \min_{H_1} \omega^{*T} x_i}{2} \end{aligned} \tag{15}$$

As we can see in Figure 2, the dotted line represent the maximum-margin hyperplane. The support vectors were marked with cross. The distribution of red class fits $N([-1,-1], \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix})$ distribution. The distribution of blue class fits $N([1,1], \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix})$ distribution. Obviously, this linear kernel toy SVM works well.

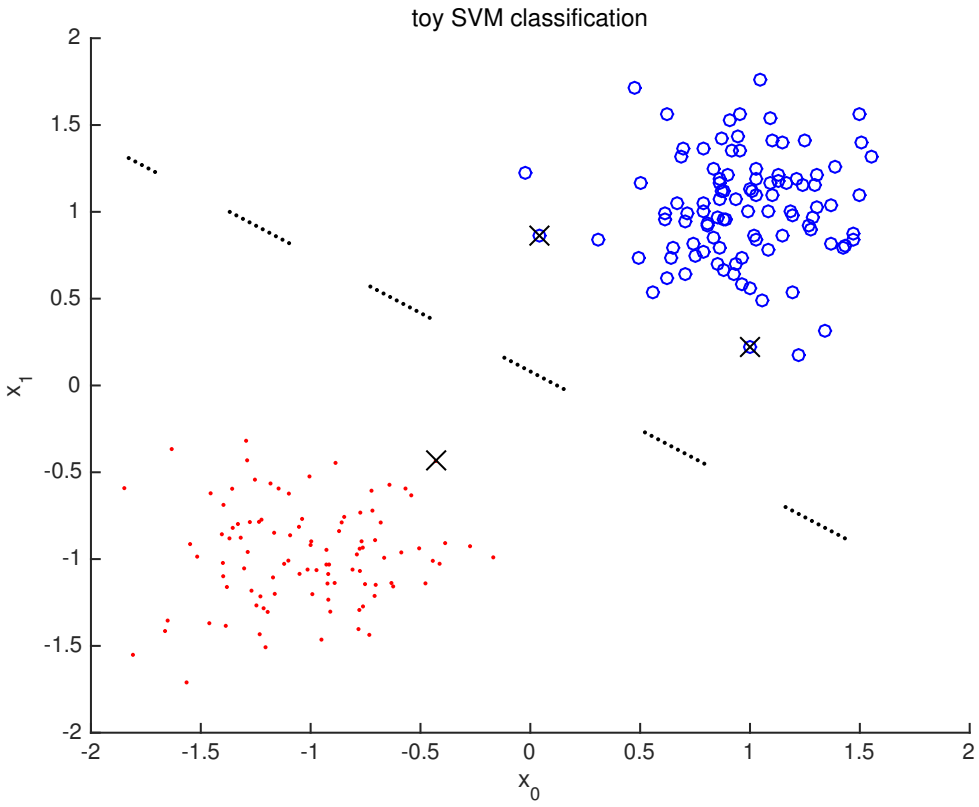

Figure 2: toy SVM classification of 2D Gaussian linear separable data set

3.2 Linear and RBF kernel SVM with creditCard dataset

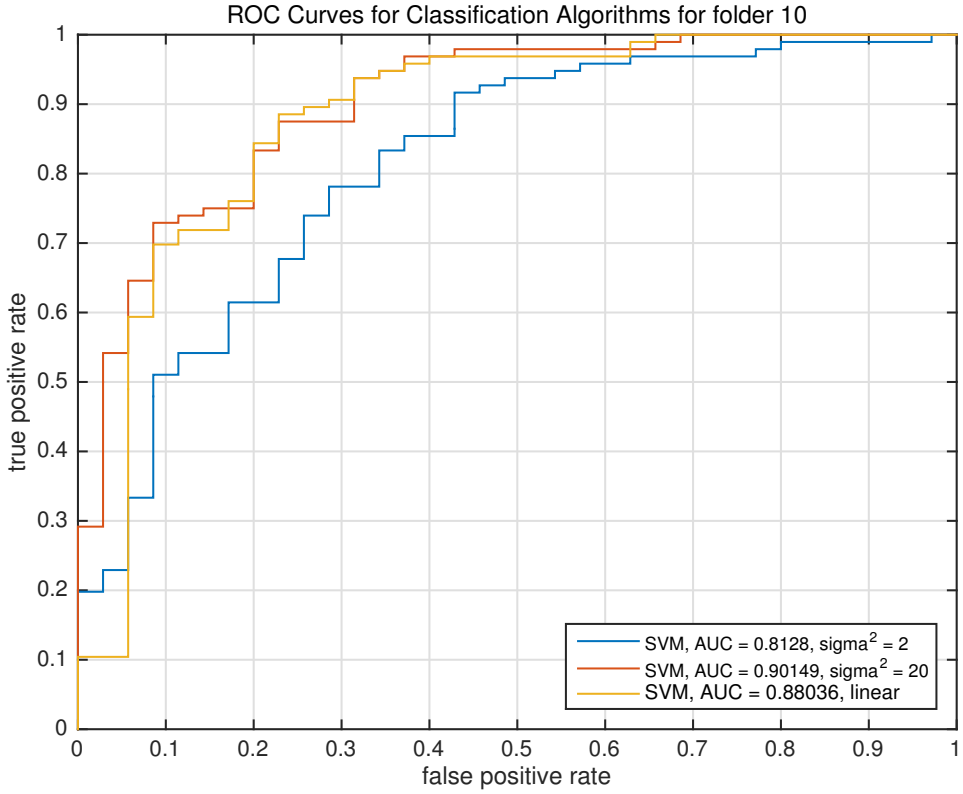

Figure 3: Linear and RBF kernel SVM classification of creditCard dataset

As we can see in Figure 3, with linear kernel, the AUC = 0.88. The AUC of RBF kernel SVM with $\sigma^2 = 2$ is 0.81 and that with $\sigma^2 = 20$ is 0.90. A reasonable explanation could be that $\sigma^2 = 2$ cause overfitting.