Machine Learning
Homework 2

Yao Song
301266041

October 20, 2014

Question 1

1. At the green point, $p(C_1|x) = p(C_2|x) = p(C_3|x) = 1/3$
2. When on the red line, two probabilities are equal to each other and they are bigger than the rest one. As an example, if it is on the red line between region 1 and region 2, then $p(C_1|x) = p(C_2|x) > p(C_3|x)$. If we move along the red line, away from the green point, two probabilities will be equally increased and remain equal, the rest one will be decreased. As an example, if we move along the line between region 1 and region 2, we have $p(C_1|x) = p(C_2|x)$, and both $p(C_1|x)$ and $p(C_2|x)$ will increase, $p(C_3|x)$ will decrease.
3. If we move far away from the intersection point in the middle of one region, the probability of that region will increase, but the other two probabilities will both decrease .

Question 2

The exponential kernel can be written as a power series as follows:

$$\begin{aligned}&\exp(k_1(x,x')) = 1 + k_1(x,x') + \frac{1}{2}k_1^2(x,x') + \frac{1}{6}k_1^3(x,x') + \dots \\ &= \sum_{d=0}^{\infty}\frac{1}{d!}k_1^d(x,x') = \sum_{d=0}^{\infty}a_d k_1^d(x,x'), \qquad \text{where} \qquad a_d = \frac{1}{d!}.\end{aligned}$$

From the above equation, we can observe that the exponential kernel is a summation of infinite number of polynomial kernels.According to Eqn.6.15, for any polynomial $q(k_1(x,x')) = a_d k_1^d(x,x')$, there exists a feature space $\phi_d(x)$ such that

$$a_d k_1^d(x,x') = \phi_d(x)^T\phi_d(x')$$

Thus, we get the following equation:

$$\exp(k_1(x,x')) = \sum_{d=0}^{\infty}\phi_d(x)^T\phi_d(x').$$

As a result, the infinite dimensional space of $\exp(k_1(x,x'))$ can be written as $\exp(k_1(x,x')) = \phi_{inf}(x)^T\phi_{inf}(x')$, where $\phi_{inf}(x) = \begin{pmatrix}\phi_0(x)\\ \phi_1(x)\\ \phi_2(x)\\ \dots\end{pmatrix}$ is a infinite dimensional column vector.

Question 3

The corresponding Langrangian is given by

$$L(w, b, \epsilon, a) = \frac{1}{2}\|w\|^2 + C_1 \sum_{n \in N} \epsilon_n + C_2 \sum_{n \in P} \epsilon_n - \sum_n a_n (t_n y(x_n) - 1 + \epsilon_n) - \sum_n \mu_n \epsilon_n,$$

where $a_n \geq 0$ and $\mu_n \geq 0$ are langrangian multipliers.
The KKT conditions are listed as follows:

$$\begin{aligned}
\epsilon_n &\geq 0, && \forall n \\
a_n &\geq 0, && \forall n \\
\mu_n &\geq 0, && \forall n \\
t_n y(x_n) - 1 + \epsilon_n &\geq 0, && \forall n \\
a_n (t_n y(x_n) - 1 + \epsilon_n) &= 0, && \forall n \\
\mu_n \epsilon_n &= 0, && \forall n
\end{aligned}$$

Let us take derivative of the Langrangian over w, b, ϵ_n separately.

$$\begin{aligned}
&\frac{\partial L}{\partial w} = 0 \rightarrow w = \sum_n a_n t_n \phi(x_n) \\
&\frac{\partial L}{\partial b} = 0 \rightarrow \sum_n a_n t_n = 0 \\
&\frac{\partial L}{\partial \epsilon_n} = 0 \rightarrow a_n = C_1 - \mu_n, \forall n \in P, \qquad a_n = C_2 - \mu_n, \forall n \in N
\end{aligned}$$

By using the above results, we can remove w, b, ϵ_n from the Langrangian and get the dual form Langrangian

$$\tilde{L}(a) = \sum_n a_n - \frac{1}{2} \sum_n \sum_m a_n a_m t_n t_m k(x_n, x_m)$$

And the box constraints are

$$\begin{aligned}
0 \leq a_n &\leq C_1, && \forall n \in P \\
0 \leq a_n &\leq C_2, && \forall n \in N \\
\sum_n a_n t_n &= 0
\end{aligned}$$

Question 4

1. The plots are oscillating because the step size is too large. When the step size is large, each time we will jump around the optimal point, but not directly to the optimal point. If the step size is larger, this figure will show a larger oscillation. The results are shown in Fig. 1 and Fig. 2
   We can fix this oscillation problem by setting a smaller step size with care. Now we will go straightforward to the optimal point direction. Note some step size values will not lead to the optimal point, but break the iteration loop early, that is not what we want. We choose the step size to be $eta = 0.001$. Note that this eta is not the best one. There are other eta values which can achieve lower objective values. The results are shown in Fig. 3 and Fig. 4.

2. The results are shown in Fig. 5 and Fig.6

3. The IRLS algorithm takes only 7 iterations to reach convergence. The results form IRLS are shown in Fig.7 and Fig.8.

Question 5

The following figures show the cross validation results. The tuning of C parameter for linear model is shown in Fig. 9. The soft margin constant C will determine the position of the decision boundary. A large C means a large penalty for misclassified points.
The tuning of parameter C and γ for the rbf kernel is shown in Fig. 10. It can achieve a score above 59. The parameter γ is the inverse kernel width. Large value of γ will give the model more flexibility.
The polynomial kernel behaves quite bad in this task.The accuracy stays at around 26.6 for various values of C and degree pairs. So the figure is not shown here.
The tuning of parameter C and γ for the sigmoid kernel is shown in Fig. 11. It can achieve a score above 59.
According to the experimentation results, I decide the best classifier for this task is the linear kernel with C = 150. We can achieve an accuracy of 59.6 with these parameters and Ptest is generated by them.

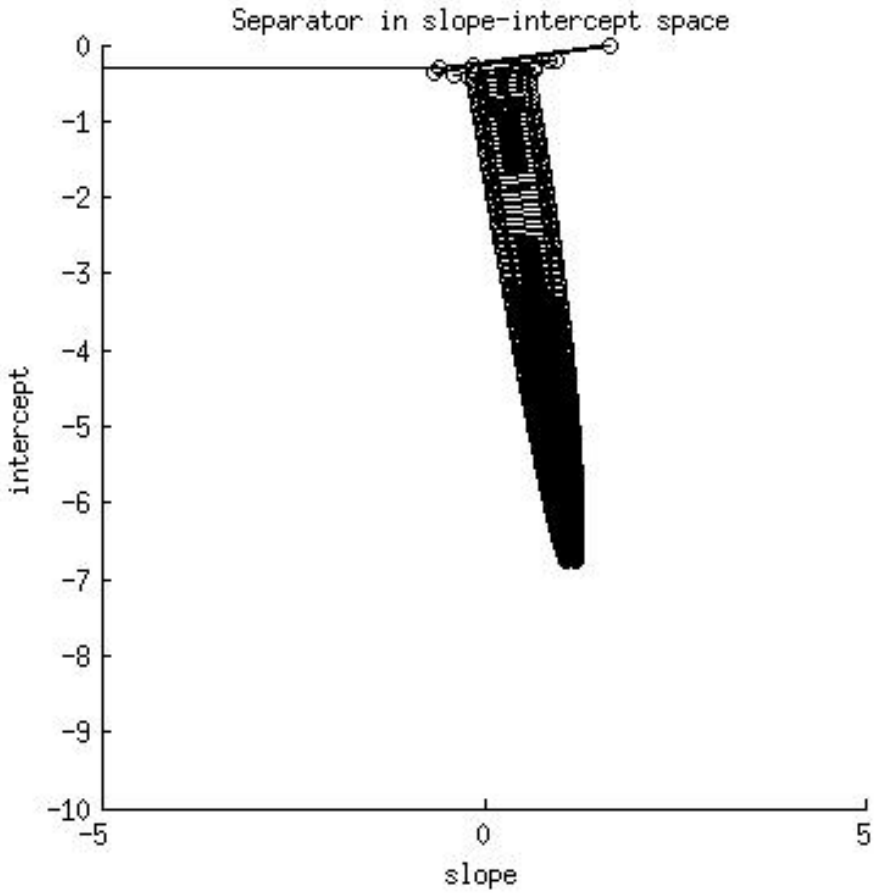

Figure 1: separator path with eta = 0.003, obvious oscillation

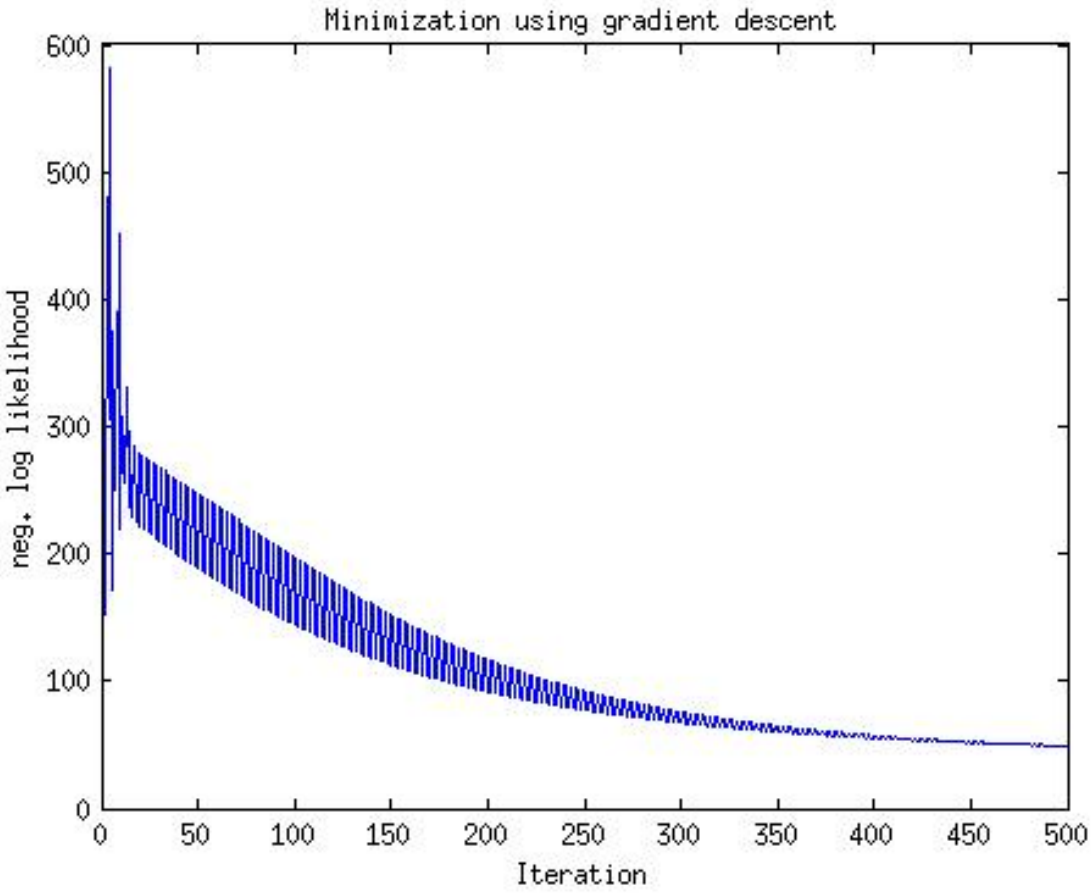

Figure 2: neg. likelihood over iterations with eta = 0.003, obvious oscillation

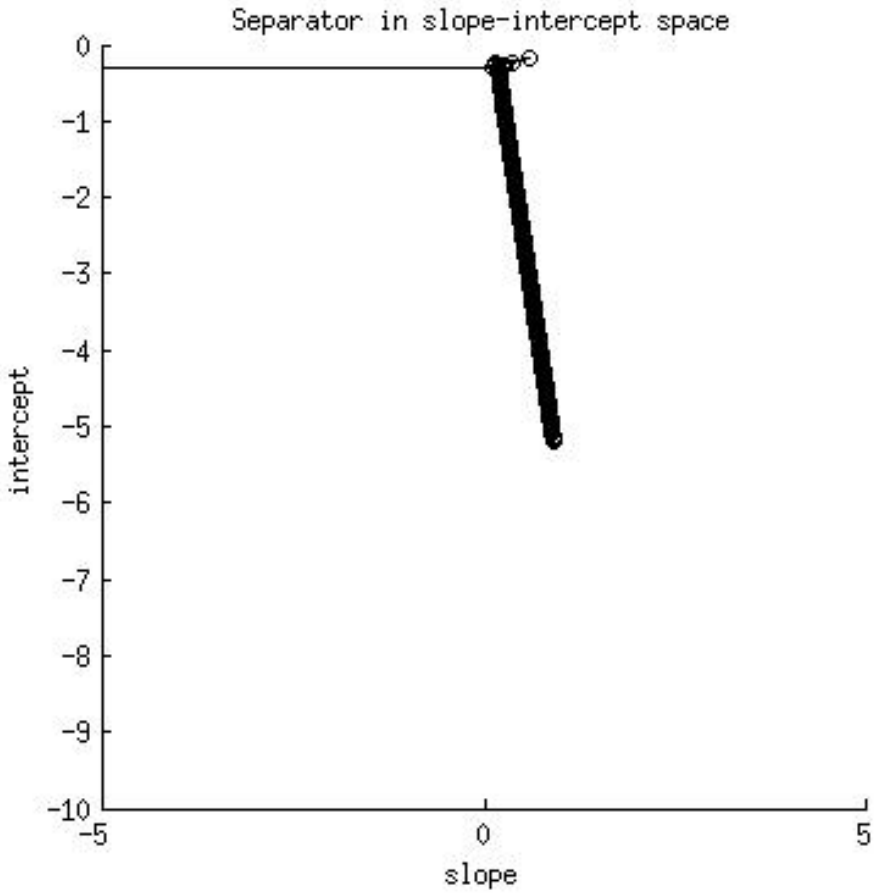

Figure 3: separator path with eta = 0.001, no oscillation

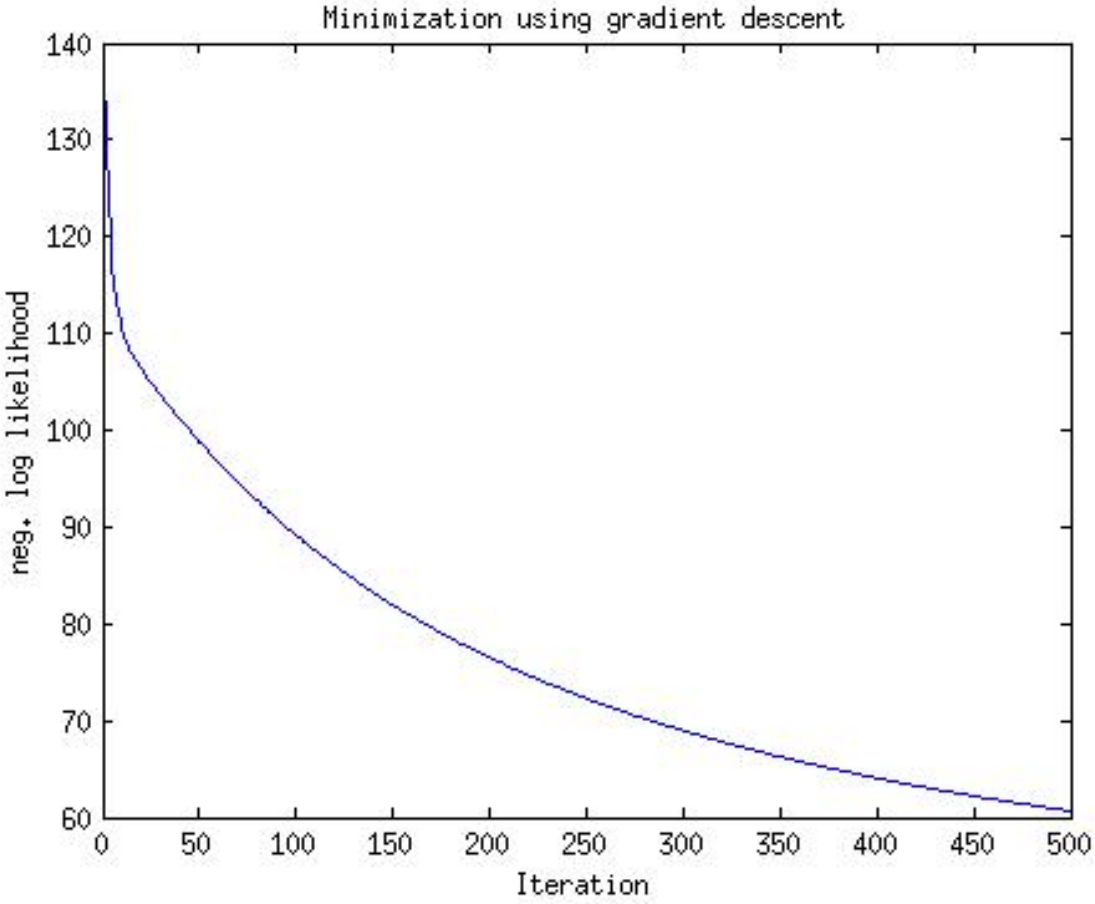

Figure 4: neg. likelihood over iterations with eta = 0.001, no oscillation

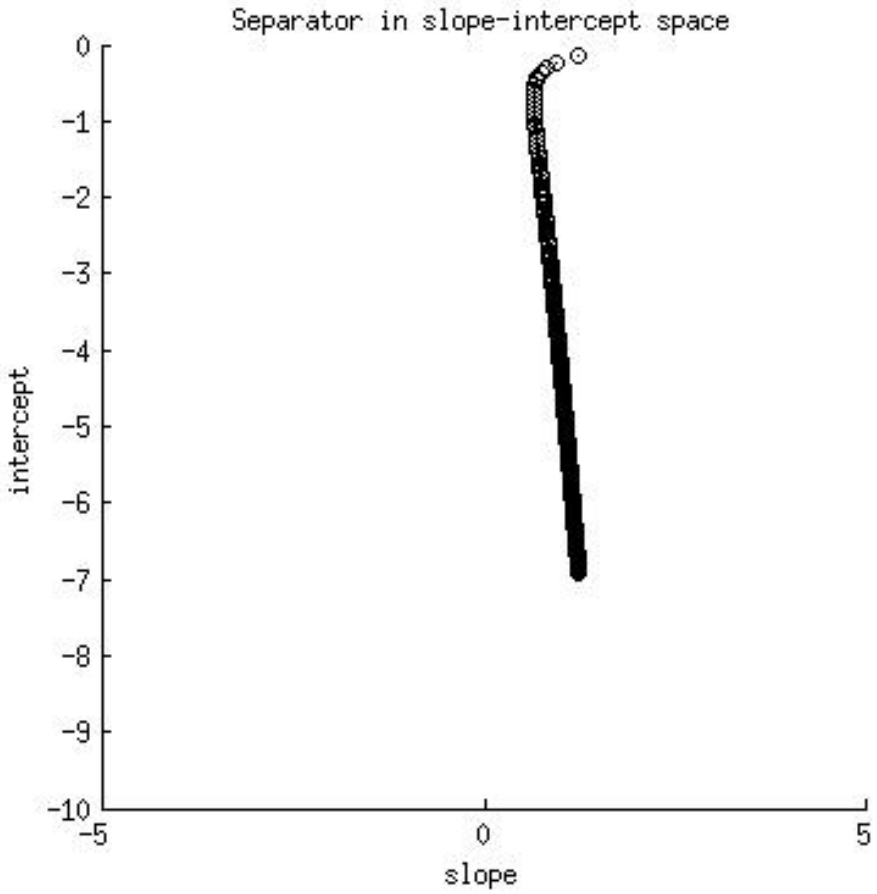

Figure 5: separator path of stochastic gradient descent

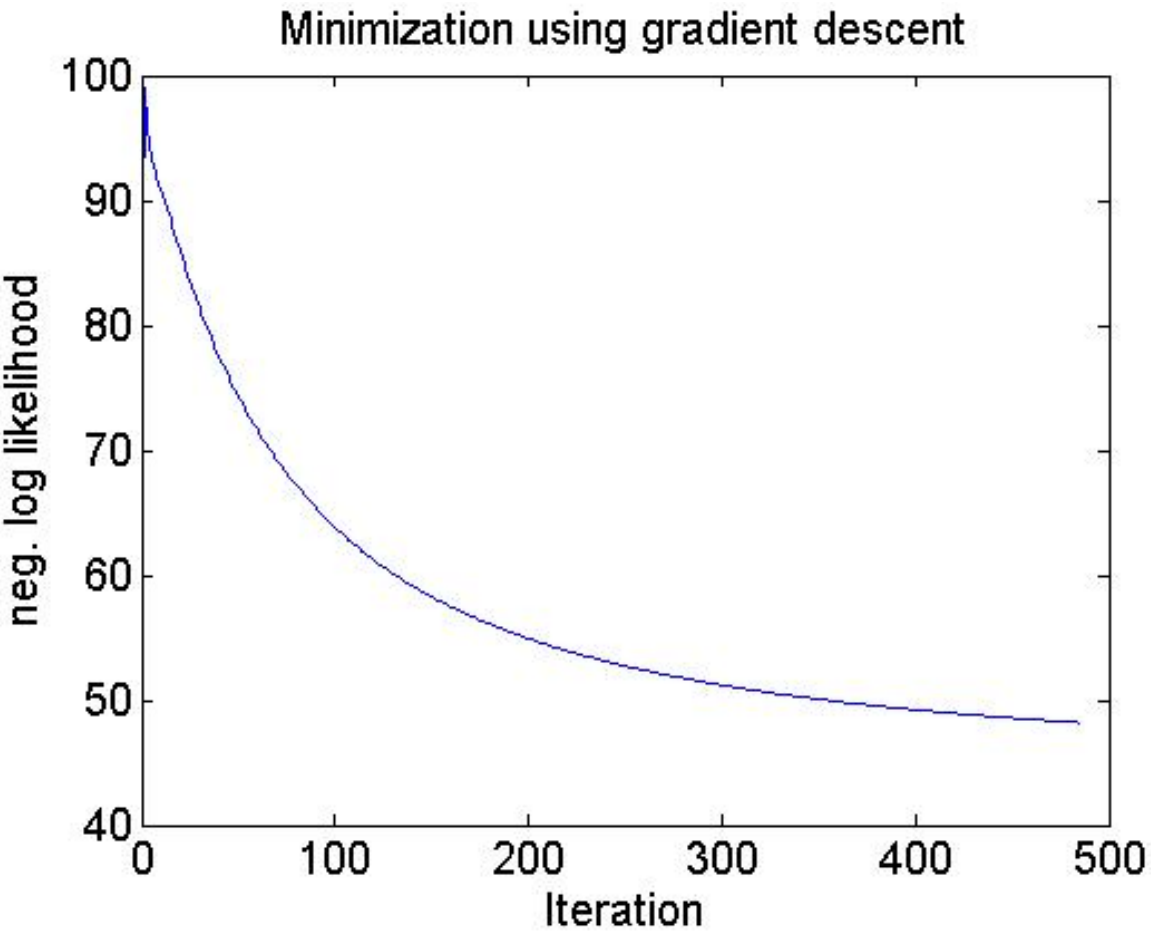

Figure 6: neg. likelihood over iterations of stochastic gradient descent

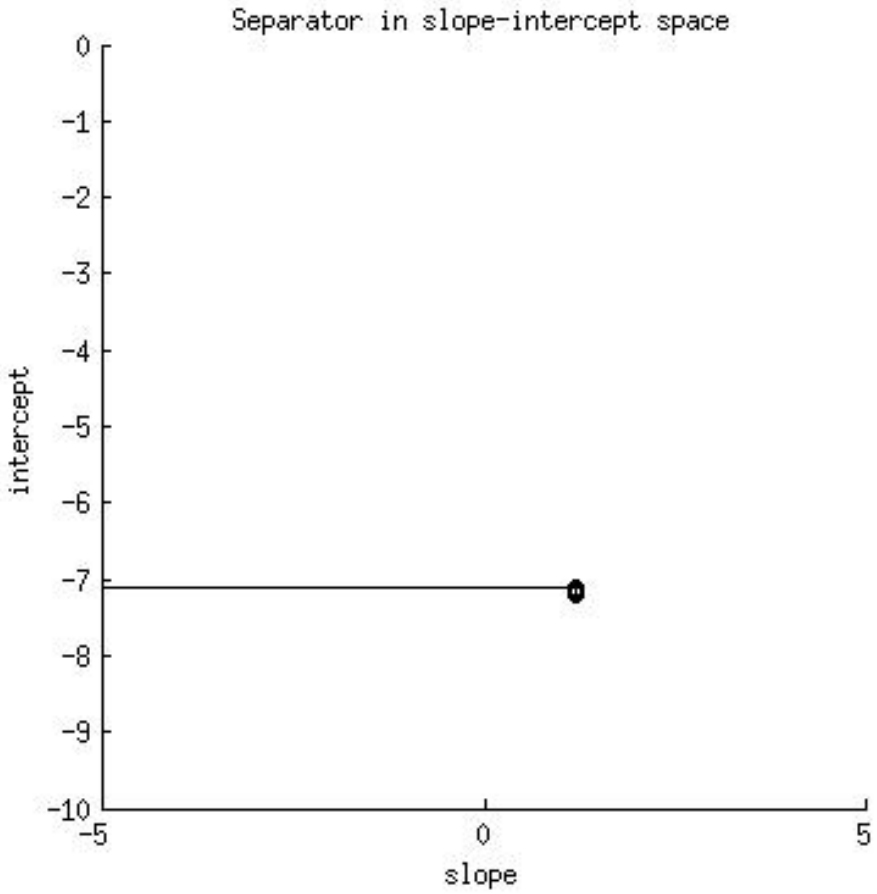

Figure 7: separator path of IRLS

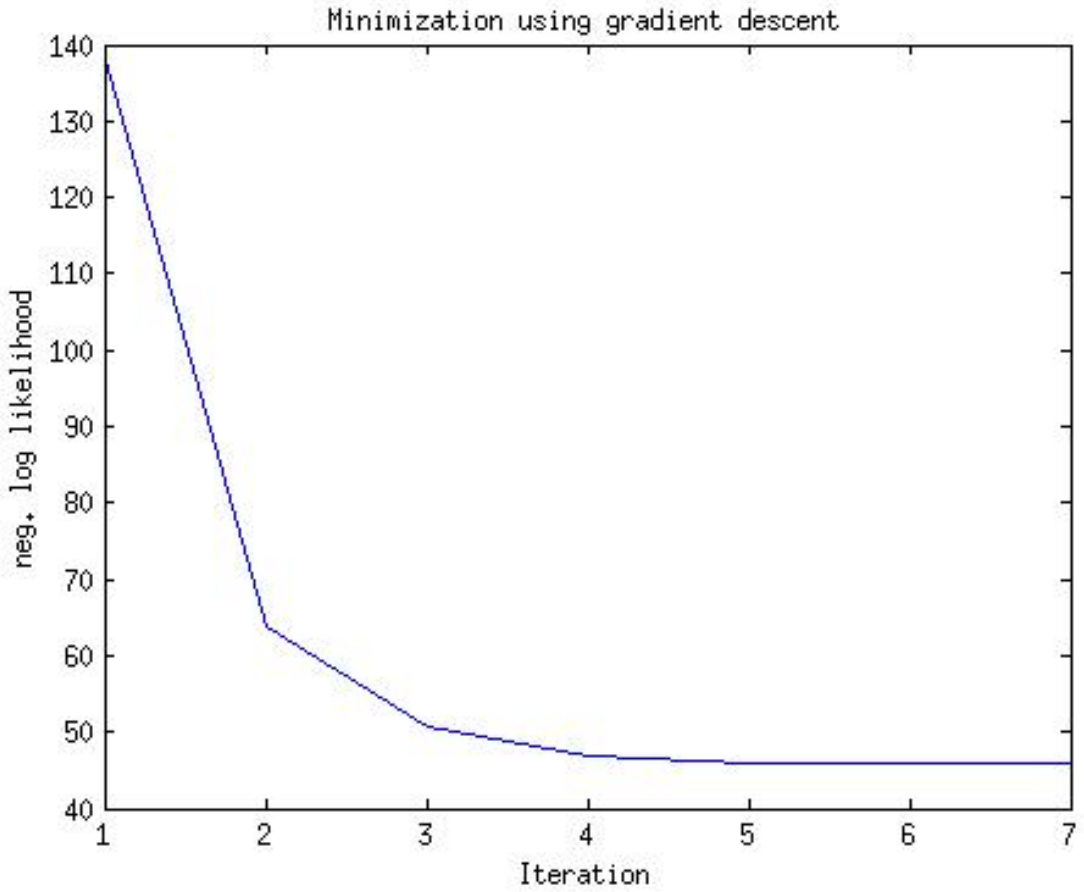

Figure 8: neg. likelihood over iterations of IRLS

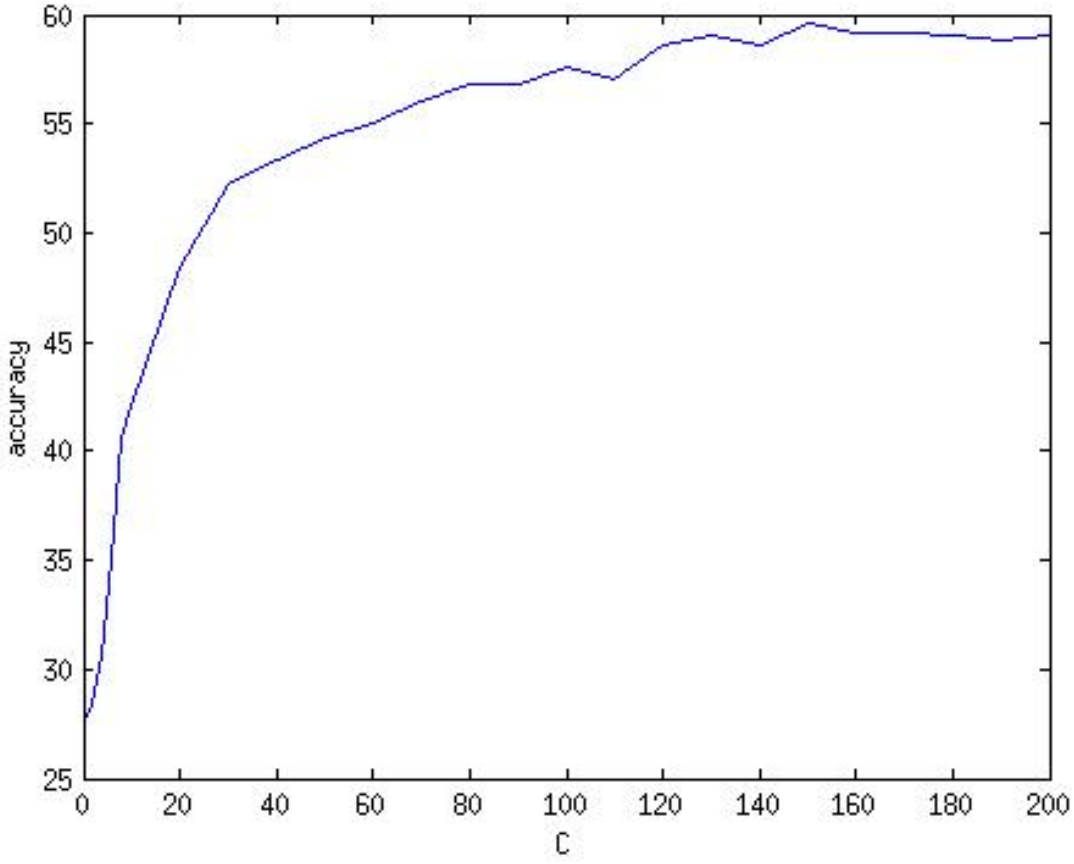

Figure 9: accuracy over the parameter C for linear kernel

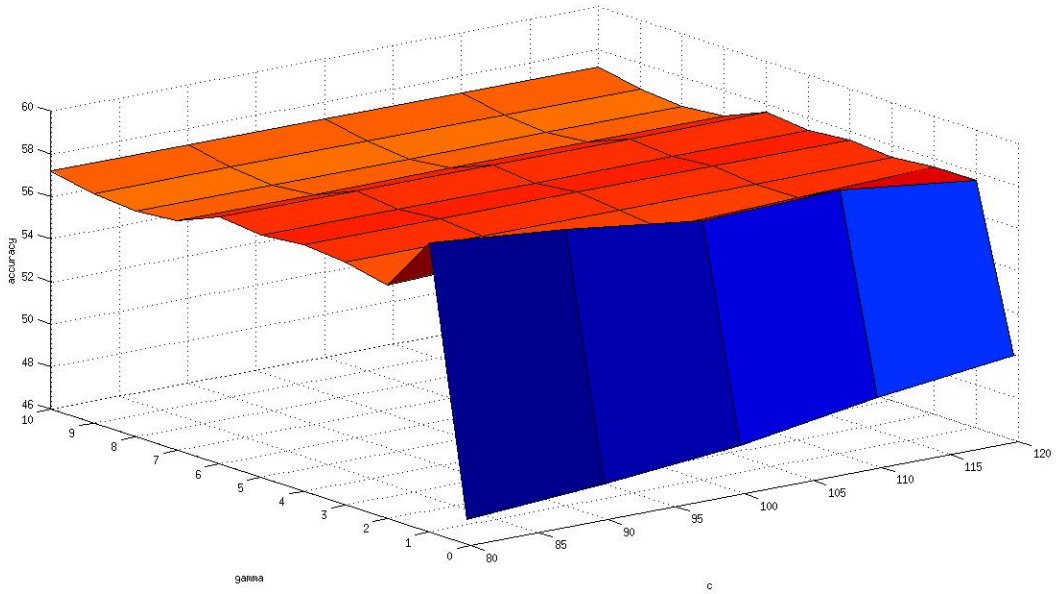

Figure 10: accuracy over parameter C and gamma for rbf kernel

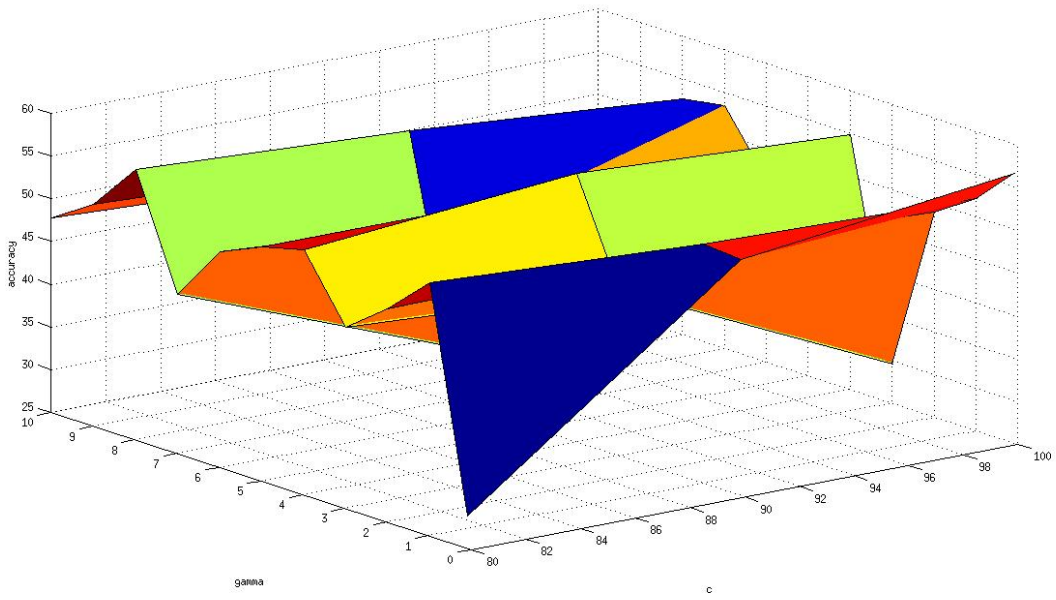

Figure 11: accuracy over parameter C and gamma for sigmoid kernel