Homework #1

Answer Sheet

DEADLINE: 04/11/2017, 14:00
INSTRUCTOR: Hsuan-Tien Lin

王冠鈞 b03902027

1. The equation of the hyperplane is:[1]

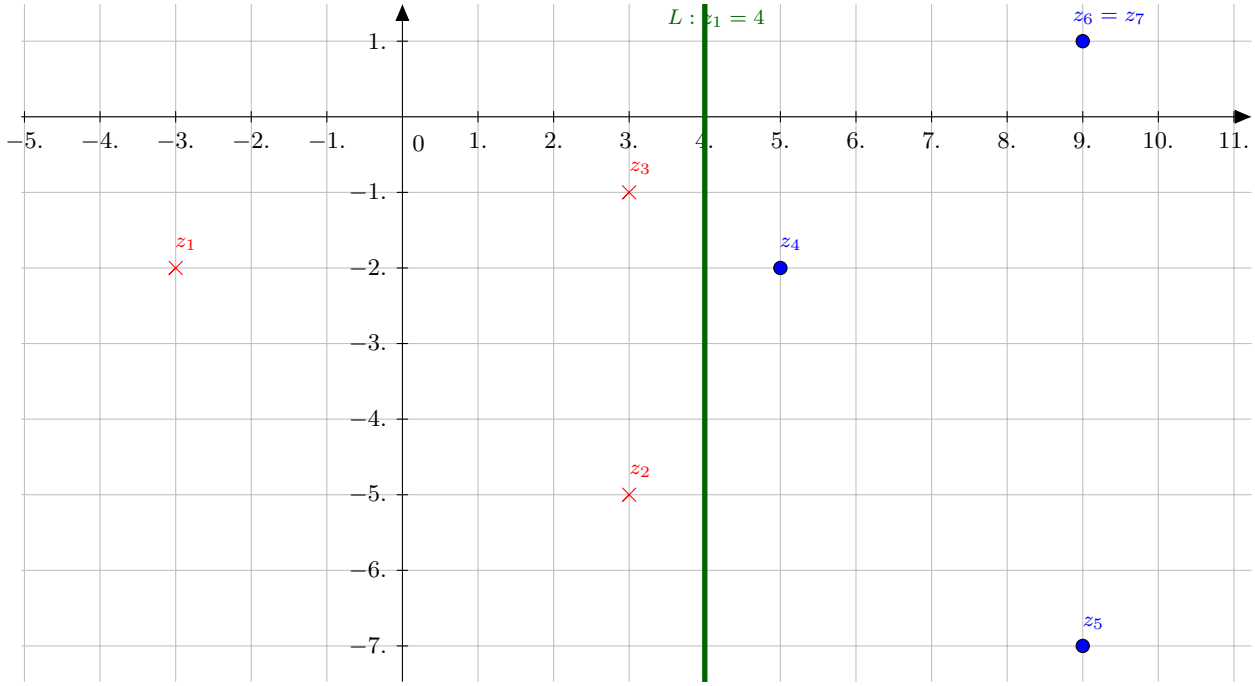
$$z_1 = 4$$

To get this result, we have to first transform all $\mathbf{x_i}$ vectors with $\phi_1(\mathbf{x}), \phi_2(\mathbf{x})$, to the points in the $\mathcal{Z}$ space, say $\mathbf{z_i}$. Then apply the linear SVM optimization problem, and solve it with a QP solver, which can get:

$$(b, \mathbf{w}) = (-4, (1, 0))$$

Thus the equation of the hyperplane is show as above.

Also, we can plot all the transformed points $\mathbf{z_i}$ on the $\mathcal{Z}$ space, as shown below:



And via observation, we can easily find that the equation $z_1 = 4$ is most likely the hyperplane that separates the data, and only $\mathbf{z_2}, \mathbf{z_3}, \mathbf{z_4}$ determines the hyperplane.

2. Instead of using the non-linear transform functions, we use a kernel function to do transformations in this question. Different from the previous question, we should use the first two steps of Kernel Hard-Margin SVM Algorithm to solve the optimal $\boldsymbol{\alpha}$. After feeding all necessary

[1] In this and the following 3 questions, I used the `cvxopt` package in python to compute the answers and I used Geogebra to visualize the results.

data to the QP solver, we can get the optimal $\boldsymbol{\alpha}$[2]:

$$\boldsymbol{\alpha} = (0, \frac{7}{15}, \frac{7}{15}, \frac{8}{15}, \frac{1}{5}, \frac{1}{5}, 0)$$

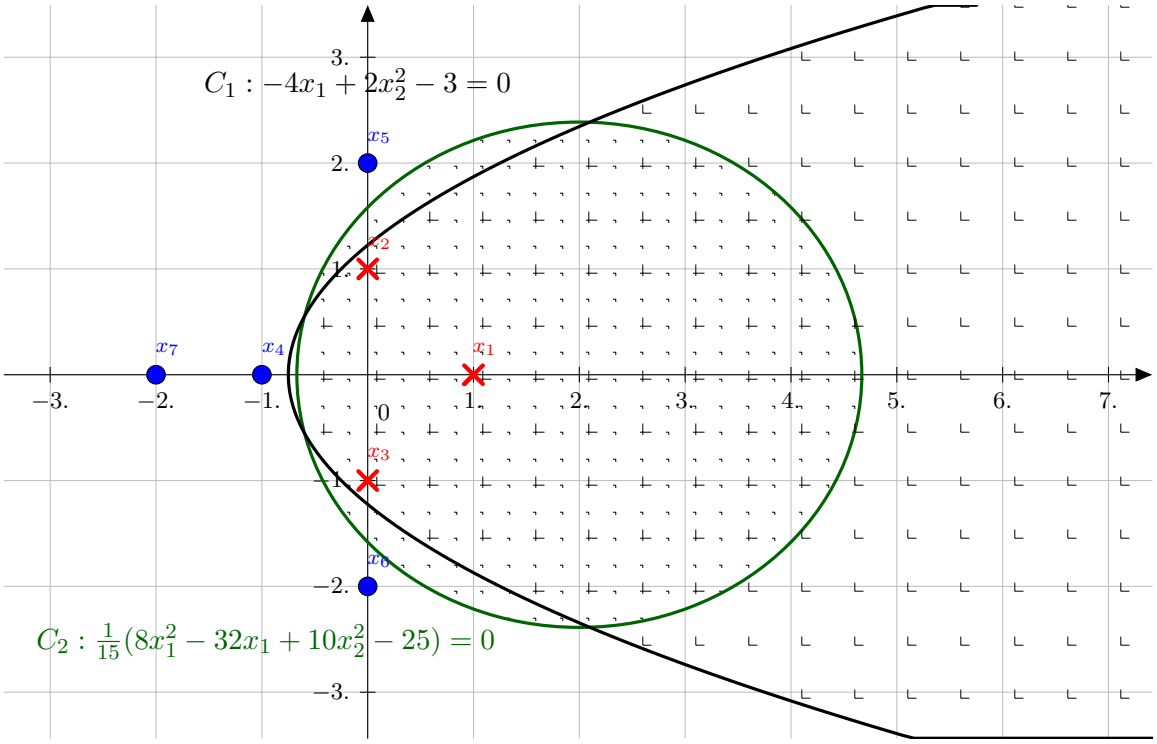
And according to this result, the vectors $\mathbf{x_2}, \mathbf{x_3}, \mathbf{x_4}, \mathbf{x_5}, \mathbf{x_6}$ are support vectors.

3. From the third step of the Kernel Hard-Margin SVM Algorithm we can get $b = -\frac{5}{3}$. And according to the last step[3], the non-linear curve should be:

$$-\frac{7}{15}(2+x_2)^2 - \frac{7}{15}(2-x_2)^2 + \frac{8}{15}(2-x_1)^2 + \frac{1}{5}(2+2x_2)^2 + \frac{1}{5}(2-2x_2)^2 - \frac{5}{3} = 0$$

$$\Rightarrow \frac{1}{15}(8x_1^2 - 32x_1 + 10x_2^2 - 25) = 0$$

4. As the picture shown below (the dotted area denotes the points that $y = -1$), the (parabolic) curve C_1 is the non-linear curve from Question 1 (directly from $4 = \phi_1(\mathbf{x}) = 2x_2^2 - 4x_1 + 1$) and the (elliptic) curve C_2 from Question 3. It's obvious that they are actually different ones since the transformation functions are basically different from each other.



5. Consider for all data $\mathbf{x_i}$, if they are correctly separated, we will set $\xi_i' < 0$, which means the "correctness" of a point (if a point is correctly separated and is further from the separation line, the value will be much more negative), then we can get:

$$\frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{n=1}^{N}\xi_n'^2 > \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{n=1}^{N}\xi_n^2, \forall \xi_i \geq 0$$

[2]The solutions from `cvxopt` are basically numerical values: `[2.081018335064783e-09, 0.4666666707571926, 0.4666666707571876, 0.5333333377102002, 0.20000000265577686, 0.2000000026557744, 5.736471702536082e-10]`, and I regard those with very low power as 0, and consider others to be rational numbers.

[3]$g_{\text{SVM}}(\mathbf{x}) = \text{sign}\left(\Sigma_{\text{SV indices n}}\ \alpha_n y_n K(\mathbf{x_n}, \mathbf{x}) + b\right)$

That is, to minimize the objective function, those "correct" data should set their ξs to 0 instead of negative. Thus, the constraint "$\xi_n \geq 0, \text{for } n = 1, 2, \cdots, N$" is unnecessary here.

We can also compare with the linear-penalty model (P_1). If we allow some $\xi_i < 0$, then the penalty may be eliminated by the correct data, which may lower the incentive to make less mistakes. However in the current model (P_2'), negative ξ_is will merely make the objective function further from optimum, so without such constraint, the optimum will still not make any ξ_i become negative.

6. As the steps demonstrated in the Dual SVM lecture, we can easily write down the Lagrange function:

$$\mathcal{L}((b, \mathbf{w}, \boldsymbol{\xi}), \boldsymbol{\alpha}) = \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{n=1}^{N}\xi_n^2 + \sum_{n=1}^{N}\alpha_n(1 - \xi_n - y_n(\mathbf{w}^T\mathbf{x}_n + b))$$

7. The three-step simplification, which eliminates $b, \boldsymbol{\xi}, \mathbf{w}$, is shown below:

(a) $\frac{\partial\mathcal{L}((b,\mathbf{w},\boldsymbol{\xi}),\boldsymbol{\alpha})}{\partial b} = 0 = -\sum_{n=1}^{N}\alpha_n y_n$, then we can eliminate b by adding the constraint $\sum_{n=1}^{N}\alpha_n y_n = 0$:

$$\max_{\alpha_n \geq 0,\ \sum y_n\alpha_n = 0}\left(\min_{\mathbf{w},\boldsymbol{\xi}}\frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{n=1}^{N}\xi_n^2 + \sum_{n=1}^{N}\alpha_n(1 - \xi_n - y_n(\mathbf{w}^T\mathbf{x}_n))\right)$$

(b) $\frac{\partial\mathcal{L}((b,\mathbf{w},\boldsymbol{\xi}),\boldsymbol{\alpha})}{\partial\xi_n} = 0 = 2C\xi_n - \alpha_n \Rightarrow \xi_n = \frac{\alpha_n}{2C}$, then we can eliminate $\boldsymbol{\xi}$ with $\boldsymbol{\alpha}$ by replacing ξ_n with $\frac{\alpha_n}{2C}$:

$$\max_{\alpha_n \geq 0,\ \sum y_n\alpha_n = 0}\left(\min_{\mathbf{w}}\frac{1}{2}\mathbf{w}^T\mathbf{w} + \sum_{n=1}^{N}\frac{\alpha_n^2}{4C} + \sum_{n=1}^{N}\alpha_n(1 - \frac{\alpha_n}{2C} - y_n(\mathbf{w}^T\mathbf{x}_n))\right)$$

(c) $\frac{\partial\mathcal{L}((b,\mathbf{w},\boldsymbol{\xi}),\boldsymbol{\alpha})}{\partial w_i} = 0 = w_i - \sum_{n=1}^{N}\alpha_n y_n x_{n,i}$, then we can eliminate $\mathbf{w}$ with constraint $\mathbf{w} = \sum_{n=1}N\alpha_n y_n\mathbf{x}_n$:

$$\max_{\alpha_n \geq 0,\ \sum y_n\alpha_n = 0, \mathbf{w} = \sum\alpha_n y_n\mathbf{x}_n}\left(\min_{\mathbf{w}}\frac{1}{2}\mathbf{w}^T\mathbf{w} + \sum_{n=1}^{N}\frac{\alpha_n^2}{4C} + \sum_{n=1}^{N}\alpha_n(1 - \frac{\alpha_n}{2C}) - \mathbf{w}^T\mathbf{w}\right)$$

$$\Longleftrightarrow \max_{\alpha_n \geq 0,\ \sum y_n\alpha_n = 0, \mathbf{w} = \sum\alpha_n y_n\mathbf{x}_n} -\frac{1}{2}\|\sum_{n=1}^{N}\alpha_n y_n\mathbf{x}_n\|^2 - \sum_{n=1}^{N}\frac{\alpha_n^2}{4C} + \sum_{n=1}^{N}\alpha_n$$

(d) And finally,

$$\min_{\alpha}\ \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N}\alpha_n\alpha_m y_n y_m\mathbf{x}_n^T\mathbf{x}_m + \frac{1}{4C}\sum_{n=1}^{N}\alpha_n^2 - \sum_{n=1}^{N}\alpha_n$$

, which is the dual problem we are seeking.

8. When using $\mathbf{z}_n = \phi(\mathbf{x}_n)$, we will be able to transform the current space to $\mathcal{Z}$ space non-linearly, and it will make the separating curve in $\mathcal{X}$ space become a non-linear curve.

The optimization problem when using a kernel $K(\mathbf{x}_n, \mathbf{x}_m)$ is similar to the problem above, with only a few variables replaced:

$$\min_{\alpha}\ \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N}\alpha_n\alpha_m y_n y_m K(\mathbf{x}_n, \mathbf{x}_m) + \frac{1}{4C}\sum_{n=1}^{N}\alpha_n^2 - \sum_{n=1}^{N}\alpha_n$$

9. **[a]** *Not necessarily valid.* I've found a counterexample by using Wolfram Alpha. First, the matrix $\begin{pmatrix} 0.2 & 0.3 \\ 0.3 & 0.5 \end{pmatrix}$ is symmetric, positive semidefinite, and is from some kernel K_1 with $0 < K_1 < 1$. However, the matrix $\begin{pmatrix} 1-0.2 & 1-0.3 \\ 1-0.3 & 1-0.5 \end{pmatrix} = \begin{pmatrix} 0.8 & 0.7 \\ 0.7 & 0.5 \end{pmatrix}$ is not positive semidefinite. Thus the new kernel $K'(\mathbf{x}, \mathbf{x}')$ is not necessarily valid.

[b] *Valid.* Since $0 < 1 - K_1(\mathbf{x}, \mathbf{x}') < 1$, its zero-th power will always become 1. Thus the M_K will be a matrix where all terms $= 1$, and it's apparently symmetric. As for positive semi-definiteness, consider the operation $\mathbf{z}^T M_K \mathbf{z} = \sum\sum z_i z_j m_{ij} = \sum\sum z_i z_j = (\sum z_i)^2 \geq 0$. Thus it is positive semi-definiteness and K is also valid.

[c] I guess that it is valid, but I have no idea how to prove it.

[d] According to the operations of kernels, **[d]** can be regarded as the power of 2 of the kernel in **[c]**, and the multiplicaton of two valid kernels is also valid; that is, if **[c]** is valid, so is **[d]**.

10. *Proof.* Consider the original problem of Soft-Margin SVM Dual:

$$\begin{aligned} (P) \min_{\boldsymbol{\alpha}} \quad & \tfrac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N}\alpha_n\alpha_m y_n y_m K(\mathbf{x}_n, \mathbf{x}_m) - \sum_{n=1}^{N}\alpha_n \\ \text{subject to} \quad & \sum_{n=1}^{N} y_n\alpha_n = 0; \\ & 0 \leq \alpha_n \leq C, \text{for } n = 1, 2, \cdots, N \end{aligned}$$

Let the solution of P be $\boldsymbol{\alpha}^*$ and the classifier g_{SVM} of this SVM is:

$$g_{\text{SVM}}(\mathbf{x}) = \text{sign}\left(\sum_{\text{SV}} \alpha_n y_n K(\mathbf{x}_n, \mathbf{x}) + b\right)$$

$$= \text{sign}\left(\sum_{\text{SV}} \alpha_n y_n K(\mathbf{x}_n, \mathbf{x}) + y_s - \sum_{\text{SV}} \alpha_n y_n K(\mathbf{x}_n, \mathbf{x}_s)\right)$$

Now, when we use the new kernel $\tilde{K}(\mathbf{x}, \mathbf{x}') = pK(\mathbf{x}, \mathbf{x}') + q, p > 0, q \geq 0$, the new classifier g'_{SVM} will be (let $\boldsymbol{\alpha'}$ be the new solution):

$$g'_{\text{SVM}} = \text{sign}\left(\sum_{\text{SV}} \alpha'_n y_n \tilde{K}(\mathbf{x}_n, \mathbf{x}) + y_s - \sum_{\text{SV}} \alpha'_n y_n \tilde{K}(\mathbf{x}_n, \mathbf{x}_s)\right)$$

$$= \text{sign}\left(\sum_{\text{SV}} \alpha'_n y_n pK(\mathbf{x}_n, \mathbf{x}) + y_s - \sum_{\text{SV}} \alpha'_n y_n pK(\mathbf{x}_n, \mathbf{x}_s)\right)$$

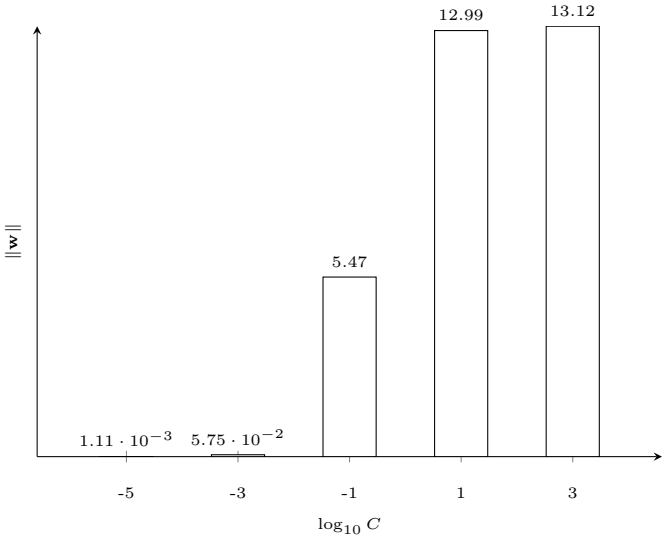
$$= \text{sign}\left(\sum_{\text{SV}} (p\alpha'_n) y_n K(\mathbf{x}_n, \mathbf{x}) + y_s - \sum_{\text{SV}} (p\alpha'_n) y_n K(\mathbf{x}_n, \mathbf{x}_s)\right)$$

That is, if $g'_{\text{SVM}} = g_{\text{SVM}}$, then $p\alpha'_n = \alpha^*_n$, (for every n) could be the solution: the new solutions $\boldsymbol{\alpha'}$ are shrunken by p times from $\boldsymbol{\alpha}$. Geometrically, we can regard the hyperplanes of new objective functions and constraints as being shrunken by p times, so that the new solutions set will be $\boldsymbol{\alpha'}$ (which is also shrunken by p times):

$$\begin{aligned} (P') \min_{\boldsymbol{\alpha}} \quad & \tfrac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N}\alpha'_n\alpha'_m y_n y_m (pK(\mathbf{x}_n, \mathbf{x}_m) + q) - \sum_{n=1}^{N}\alpha'_n \\ \text{subject to} \quad & \tfrac{1}{p}\sum_{n=1}^{N} y_n\alpha'_n = 0; \\ & 0 \leq \alpha'_n \leq \tfrac{C}{p} = \tilde{C}, \text{for } n = 1, 2, \cdots, N \\ \text{where} \quad & \alpha'_i = \tfrac{\alpha_i}{p} \end{aligned}$$

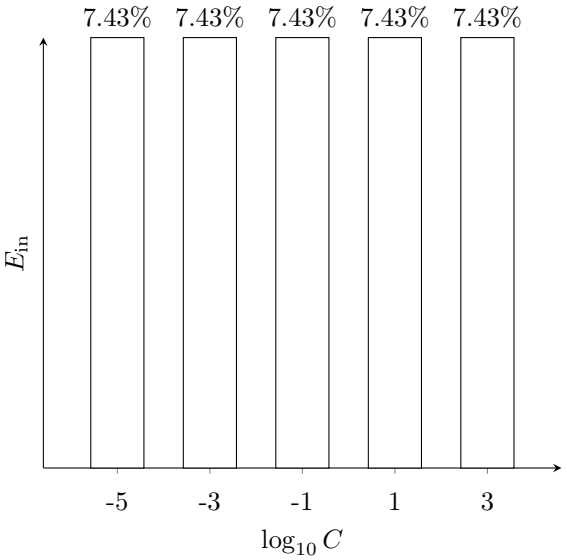
Note that the objective now contains a new term $\frac{q}{2}\sum_{n=1}^{N}\sum_{m=1}^{N}\alpha'_n\alpha'_m y_n y_m$, but according to the constraint $\sum_{n=1}^{N} y_n\alpha'_n = 0$, the new term should always be 0. Thus all functions and constraints are in fact merely shrunken by p times, and thus $\boldsymbol{\alpha'} = \frac{\boldsymbol{\alpha^*}}{p}$. In this situation C is also shrunken by p times, and thus we get $\tilde{C} = \frac{C}{p}$. □

11. Histogram:



From the graph above, it's easy to find that the $\|\mathbf{w}\|$ value increases as $\log_{10} C$ increases. The possible reason is that, according to the implicit constraint $\mathbf{w} = \sum_{n=1}^{N}\alpha_n y_n \mathbf{z}_n$, when C increases, α_n may increase, and this will cause the absolute values some terms of $\mathbf{w}$ to be larger, which will make $\|\mathbf{w}\|$ larger. Some other findings are that the larger C is, the longer it computes[4], and that $\|\mathbf{w}\|$ grows fast at $\log_{10} C = -1$, but slowly at other points.

12. Histogram:

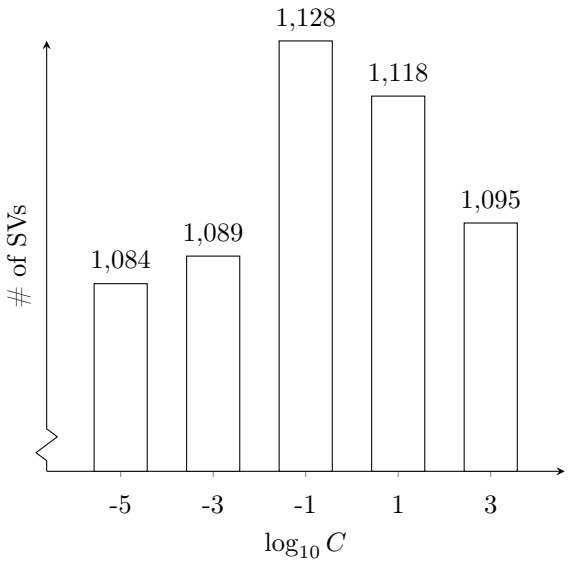


What's interesting is that E_{in} remains the same no matter what C is. Perhaps it's because it has reached its maximum accuracy since C is 10^{-5}, and any more penalty on violations will

[4]The `libsvm` package gives a warning at $C = 1000$, stating that it has reached the maximum number of iterations.

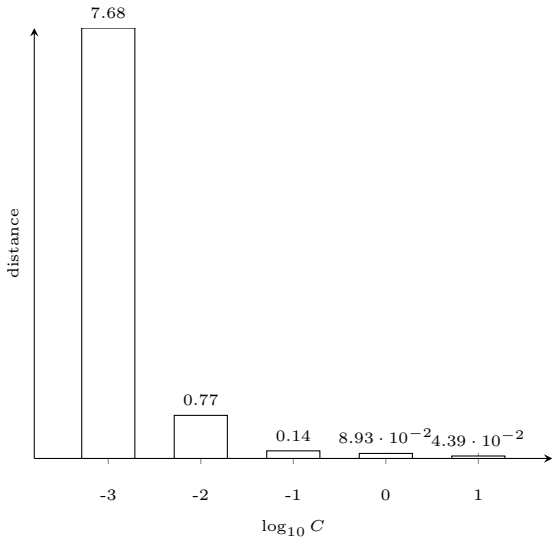
not enable the model to make any less errors.

13. Histogram:



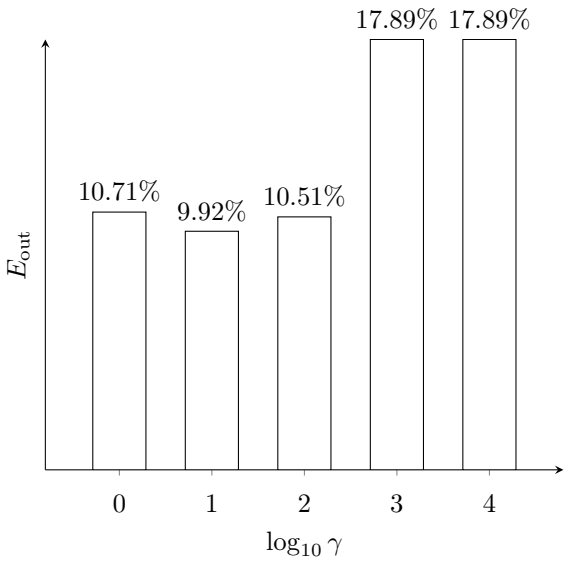
The result is that it reaches a peak when $C = 0.1$.

14. Histogram:



The distince decreases as the $\log_{10} C$ grows. According to the distance formula: $d(\mathbf{z}) = \frac{1}{\|\mathbf{w}\|}|\mathbf{w}^T\mathbf{z} + b|$, and consider a free support vector $\mathbf{x}_s$, $|w^T\mathbf{z} + b| = |\sum_{\mathsf{SV}} \alpha_n y_n K(\mathbf{x}_n, \mathbf{x}_s) + y_s - \sum_{\mathsf{SV}} \alpha_n y_n K(\mathbf{x}_n, \mathbf{x}_s)| = |y_s| = 1$ Thus for any $\mathbf{x}_s$, the distance to the hyperplane will be $\frac{1}{\|\mathbf{w}\|}$ (reverse of $\|\mathbf{w}\|$), and I've mentioned that $\|\mathbf{w}\|$ may increase as C increases, so it may also be the reason that the distance decreases as C increases.

15. Histogram:



According to the result, E_{out} is generally higher when $\log_{10}\gamma$ is large, which means that out-sample errors are higher when we choose a large γ. A possible reason to this result is that large γ values in a Guassian kernel may lead to overfitting, which is not good for data prediction, and thus cause high E_{out}.

16. Histogram: