Support Vector Machine Project

Hao Zheng 11610127
Department of Computer Science and Engineering
Southern University of Science and Technology
11610127@mail.sustc.edu.cn

1. Preliminaries

This project is implementing a support vector machine (SVM) to solve binary classification problem, which is a basic machine learning problem.

1.1. Problem Description

Binary or binomial classification is the task of classifying the elements of a given set into two groups (predicting which group each one belongs to) on the basis of a classification rule. Contexts requiring a decision as to whether or not an item has some qualitative property or some specified characteristic.

1.2. Problem Application

One of the application is cat and dog recognition, in which the input is a image of cat or dog and the output should be the category of the input (cat or dog).

1.3. Algorithm

The algorithm used in this project to optimize SVM is Sequential Minimal Optimization(SMO), which is an algorithm for solving the quadratic programming (QP) problem. As the objective function of SVM is solving a QP problem, SMO can be used to solve SVM.

2. Methodology

2.1. Notation

α_i	Lagrange multipliers
y_i	the label
x_i	the input
K	kernel function
m	the size of training set

2.2. Data Structure

No specific data structure used here.

2.3. Model Design

Consider a binary classification problem with a dataset (x_1, y_1), ..., (x_n, y_n), where x_i is an input vector and $y_i \in \{-1, +1\}$ is a binary label corresponding to it. A soft-margin support vector machine is trained by solving a quadratic programming problem, which is expressed in the dual form as follows:

$$\max_{\alpha} \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j K(x_i, x_j)$$

$$s.t. \sum_{i=1}^{m} \alpha_i y_i = 0,$$

$$\alpha_i \geq 0, i = 1, 2, ..., m$$

SMO is an iterative algorithm for solving the optimization problem described above. SMO breaks this problem into a series of smallest possible sub-problems, which are then solved analytically. Because of the linear equality constraint involving the Lagrange multipliers α_i, the smallest possible problem involves two such multipliers. Then, for any two multipliers α_1 and α_2, the constraints are reduced to:

$$0 \leq \alpha_1, \alpha_2 \leq C, y_1 \alpha_1 + y_2 \alpha_2 = \epsilon$$

And this reduced problem can be solved analytically: one needs to find a minimum of a one-dimensional quadratic function. ϵ is the negative of the sum over the rest of terms in the equality constraint, which is fixed in each iteration.

The algorithm proceeds as follows:

1) Find a Lagrange multiplier α_1 that violates the Karush–Kuhn–Tucker (KKT) conditions for the optimization problem.
2) Pick a second multiplier α_2 and optimize the pair (α_1, α_2).
3) Repeat steps 1 and 2 until convergence.

2.4. Detail of algorithms

In each step, the α_1 is chose from all α iteratively, then the α_2 is chose if violating $0 \leq \alpha_2 \leq C$. Then the i_1, i_2

is transmitted into a step function, which is used to update these two α. The step function is shown as follows:

Input: $i_1 \neq i_2$
0: Calculate $E_1 \Leftarrow f(x_{i1}) - y_{i1}$
0: Compute L and H
0: **if** $L = H$ **then**
0: return
0: **end if**
0: $\eta = \langle x_{i1}, x_{i1} \rangle + \langle x_{i2}, x_{i2} \rangle - 2 \times \langle x_{i1}, x_{i2} \rangle$
0: **if** $\eta \leq 0$ **then**
0: return
0: **end if**
0: Compute and clip new value for α_2
0: $diff = |\alpha_2^{old} - \alpha_2^{new}|$
0: **if** $diff < tolerance$ **then**
0: return
0: **end if**
0: $\alpha_1 = \alpha_1 + y_1 y_2 \times diff$
0: update b
=0

After implementing SMO, I found that the result of SMO is unstable (97% - 99%), thus I run 8 SMO simultaneously to guarantee the stability. The final SVM is chose by the highest accuracy.

3. Empirical Verification

3.1. Dataset

The given dataset is used.

3.2. Performance Measure

As an atomic step, I'll run 8 svm simultaneously and find the max accuracy. After setting a hyperparameter, I'll run the atomic step for 100 times and get the average max accuracy.

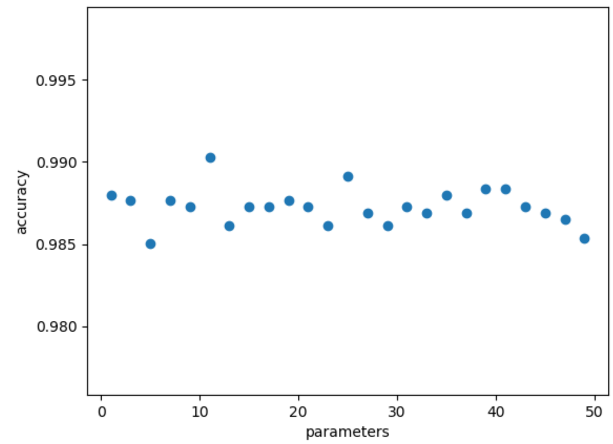
3.3. Hyperparameters

- $tolerance$: the tolerance for violating KKT conditions
- C : the coefficient of soft margin

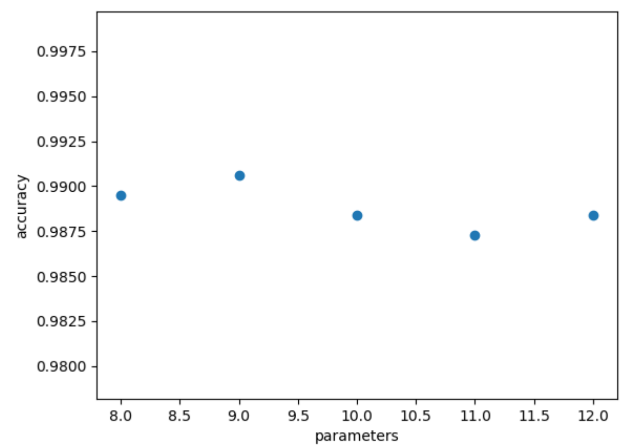
3.4. Experimental Results

For C, I run the parameter test in (1, 3, 5, ..., 49). Then run a more precise test in the highest point.

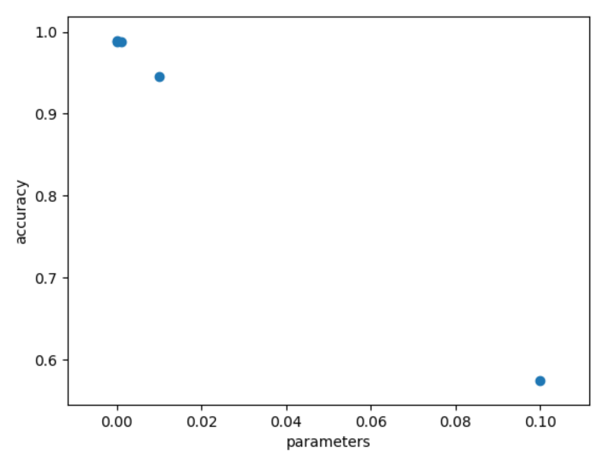
The rough test result:



The precies test result:



For tolerance, I test it exponentially, form (0.1, 0.01, ..., 0.00001) The result is following:



As the scale is decreased exponentially, the point is unclear, but the highest one is 0.0001.

3.5. Conclusion

From results above, the best parameters is as follows:
$C = 12$
$tolerance = 0.0001$

References

[1] Cs229.stanford.edu. (2018). The Simplified SMO Algorithm. [online] Available at: http://cs229.stanford.edu/materials/smo.pdf [Accessed 31 Dec. 2018].

[2] Web.cs.iastate.edu. (2018). Sequential Minimal Optimization for SVM. [online] Available at: http://web.cs.iastate.edu/ honavar/smo-svm.pdf [Accessed 31 Dec. 2018].

[3] Zhihua Zhou, ”Machine Learning”, In China. Beijing: Tsinghua University Press, 2016, pp. 121-140.