ATLANTIS PRESS
Advances in Computer Science Research, volume 87
3rd International Conference on Mechatronics Engineering and Information Technology (ICMEIT 2019)

Support Vector Machine SMO Algorithms and Their Optimization

Xiujun Wu*
School of Computer Science and Technology, Wuhan University of Technology, Wuhan 430063, China
*2501219891@qq.com

Abstract. Support Vector Machine (SVM) is an important statistical machine learning algorithm, among which SMO algorithm is one of the effective application methods. SMO algorithm decomposes the problem of support vector machine into smaller problems to minimize the sequence. Through the introduction of SMO algorithm and the analysis and summary of the problems, this paper explores the corresponding SMO optimization algorithm, hoping to provide theoretical support for future research.

Keywords: support vector machine, SMO algorithm, Optimization Research.

1. Introduction

With the continuous progress of science and technology and the deepening of research, support vector machine (SVM) has become a research hotspot in the field of machine learning. Compared with other training algorithms, SMO algorithm is faster and easier to practice. However, we find that there are still some improvements in the SMO algorithm. For example, single threshold is one of the reasons why SMO algorithm is inefficient when KKT condition is used to determine the optimal value. In order to solve this problem, two threshold parameters are adopted when KKT condition is used to determine the optimal solution. The Platt algorithm and the improved SMO algorithm are tested using UCI standard database. The results show that the improved algorithm has better performance than Platt's SMO algorithm.

2. Overview of Support Vector Machine Theory

Simply put, support vector machines (SVMs) is a two-class mathematical model. Its function is to divide a hyperplane into samples. The principle followed is to enlarge the interval to the maximum, and eventually transform it into a quadratic programming problem. The result of SVM classifier is optimized, and the risk of the whole sample set can be controlled. This is also one of the ways to minimize the risk of vector machine structure. The low-dimensional input space data is mapped to the high-dimensional feature space by the non-linear mapping function. It is proved that most linear inseparable problems in the input space can be transformed into linear separable problems in the feature space as long as the appropriate mapping function is selected. The solution of this problem makes the support vector machine classifier formally become a general classifier. SMO algorithm is also a decomposition algorithm. Its workspace contains only two data examples. In each iteration, only two Lagrange multipliers are optimized. Because of the linear equality constraints of Lagrange multipliers, this is an achievable minimum optimization problem. Although the sub-problems of quadratic programming have increased in SMO, the overall computer speed has been greatly improved, and the algorithm does not need to deal with large matrices[1].

Therefore, there is no additional requirement for storage space, and large SVM training problems can also be computed by personal computers.

3. Analysis of Problems in SMO Algorithms at Present

The full name of SMO algorithm is Sequence Minimum Optimization. The corresponding samples of two Lagrange multipliers selected in each step are represented as I1 and I2. It is very important to select the method that needs samples I1 and I2 at each step to optimize the performance of the

Copyright © 2019, the Authors. Published by Atlantis Press.
This is an open access article under the CC BY-NC license (http://creativecommons.org/licenses/by-nc/4.0/).

algorithm. SMO algorithm uses two layers of loop: the outer loop chooses I2, the inner loop chooses I1 as a given object, the outer loop first uses non-boundary samples, and chooses samples that violate KKT conditions to adjust until all non-boundary samples satisfy KT conditions [3]. When non-boundary samples are not adjusted in one traversal, all samples are traversed. To test whether the entire set satisfies the KKT condition. If all conditions are satisfied, the algorithm terminates. Otherwise, the algorithm will continue to optimize and traverse the boundary samples again. The inner loop selects another sample to match the sample that violates KKT condition (i.e. to optimize its Lagrange multiplier). As the second sample, the SMO algorithm updates the beta value after optimizing the Lagrange multiplier corresponding to the two samples that need to be optimized, but the value may not be determined (for example, there is no sample with 0 < a < c, that is, the samples are all on the boundary). SMO method is to determine the upper and lower bounds of beta, and then take the average value. The beta value in each iteration process only depends on the optimal value of two variables of the last iteration result, which is used to determine whether the sample satisfies the iteration result. In this way, some samples may reach the optimum value, but the actual results can not meet the corresponding situation of KKT conditions, thus reducing the efficiency of SMO algorithm. Therefore, in the process of SMO algorithm optimization, we need to analyze the basic algorithm of SMO algorithm, and extract the corresponding algorithm essence for optimization processing, in order to summarize the core SMO algorithm, and summarize and deduce the key points of SMO algorithm, check the optimization conditions of samples[2].

4. Analysis of SMO Algorithm after Optimization

In order to reduce the efficiency of SMO algorithm due to the use of the threshold parameter beta, we can try to use two threshold parameters to replace the original form and optimize the derivation. Suppose Fi is valid for all sample I at any time. Define the following relationships:

$$F_{i-\mathrm{low}} = \mathrm{b}_{\mathrm{low}} = \max\{F_i : \mathrm{i} \in I_0 \cup I_3 \cup I_4\} / F_{i-up} = b_{up} = \min\{F_i : i \in I_0 \cup I_1 \cup I_2\}$$

In this way, it is easy to check whether the sample I satisfies the optimization conditions. For example, suppose 1 < 1U12, we only need to check the conditions: F, < Fom. If this condition is satisfied, then there will be a pair of samples violating KKT condition to check whether sample_meets the optimization condition. If it is satisfied, F is used to update (bo, io) or (bp, ip). If not, we choose (i, io) or (i, ip) as the sample pair for this optimization. For example, if I < 1112 and F1 < b, that is, sample I violates the optimization conditions, then we choose sample pair (i, i) as the two samples for this optimization. I f sample I satisfies the optimization condition, then use f update (ip, bu). That is to say, i f f < b, then ip:= i, bp:= f, so in smo algorithm, when the outer loop chooses i2, we no longer use formula (6) as the optimal condition, but use formula (7) as the optimal condition, thus avoiding unnecessary overhead caused by incorrect beta value[3].

In this way, it is easy to check whether sample I satisfies the optimization conditions. For example, suppose 1 < 1U12, we only need to check the conditions: $Fi < Fi_{-low}$ .If this condition is satisfied, some samples violating KKT condition (blow, ilow) and (bup, iup) can be identified. The structure after identification is aggregated, and the relevant samples are calculated synthetically. Then the use requirements of the samples are checked by corresponding mathematical calculation method, and the corresponding optimization is carried out. Use f to update (blow, ilow) or (bup, iup). If not, we choose (ilow, ilow) or (iup, iup). As a sample of this optimization. For example, if I violates the corresponding optimization conditions, i. e. the optimization conditions, then (i, ilow) can be selected as the two samples for this optimization. If sample I satisfies the optimization conditions, Fi updates (iup, bup) are used. That is to say, i f Fi<bup, then iup:= i, bp:= f, so in the SMO algorithm, i f the outer loop is selected as I sample, it is not necessary to use the formula as the optimal condition, thus avoiding the algorithm loss caused by inaccurate beta value. In the specific experimental process, this paper uses the traditional SMO algorithm and the improved SMO algorithm to compare. The hardware environment is 128 megabytes of memory, Pentium 450 processor, and the software

environment is WIN2000 professional + MATLAB 6.1. The first order polynomial function is chosen as the kernel function, and the error control parameter C = 0.1. The data set we selected was UCI voting and diagnostic data[4].

Table 1. Data Set Properties

data set	Sample dimension	Sample size
Votes	16	435
IGA	14	153

Table 2. Operating effect of votes dataset

algorithm	Running time (s)
SMO algorithm	23.7
Improved SMO algorithm	15.6

Table 3. IGA Data Running Effect

algorithm	Running time (s)
SMO algorithm	16.9
Improved SMO algorithm	9.8

In the whole process, the SMO program is compiled by matlab, which is a kind of explanatory language. The actual efficiency of calculation is low. At this time, the actual speed of programming needs to be accelerated by the computer software C language. Based on the calculation of SVM classification algorithm, a new improved SMO algorithm is proposed on the basis of summing up the SMO algorithm, using the standard UCI. Data are integrated to calculate data in an all-round way. On the real surface of the calculation results, the improved SMO algorithm is a little ahead of the traditional SMO algorithm in terms of simplicity and calculation method[5].

References

[1]. Lu Mingzhu, Su Xiaoming, Chen Changzheng, Liu Shixun. Application of improved particle swarm optimization support vector machine in rolling bearing fault diagnosis [J].Machinery and Electronics, 2019,37(01): 42-48.

[2]. Chen Cheng. Particle Swarm Optimization Support Vector Machine for Risk Assessment of Construction Projects [J]. Journal of Inner Mongolia Normal University (Natural Science Chinese Edition), 2019, 48 (01): 86-89.

[3]. Liu Kun. Chaotic prediction of network traffic based on particle swarm optimization support vector machine [J]. Modern electronic technology, 2019,42(02): 120-123.

[4]. Lin Tao, Zhang Li, Cai Ruiqi, Yang Xin, Liu Gang and Liao Wenxue. Performance evaluation of wind turbines based on improved Drosophila algorithm optimization support vector machine [J]. Renewable energy, 2019, 37 (01): 132-137.

[5]. Zhou Shixiang. Vehicle sales forecasting based on genetic algorithm optimization support vector machine [J]. Operations and management, 2019 (01): 128-131.