Research Report

Introduction

Machine Learning has been recently used to solve research problems in EDA area. It aims at training a model based on experimental data and further using this model to conduct prediction models or make decisions. One important and basic machine-learning algorithm is Support Vector Machine (SVM). In this project, we will apply SVM to solve one specific EDA problem – the sensor placement problem.

Task Description

i.Use SVM to construct a model which can judge whether there occurs an emergency in a position based on the values of sensors.
ii.Discuss the effects changing the parameters and model type.

Task Assignments

i.Jiachen Wang(Leader):Writing the main program and taking most charge of debugging
ii.Hongxiang Gao(Member):Writing the research report and establishing the communication relationship with other groups
iii.Zhaohong Xiu(Member):Taking charge of the logistics and debugging the program partly

Description of our algorithm

*According the description of SVM in the book"*支持向量机通俗导论（理解 *SVM* 的三层境界）*",we decided to make an assumption:*

$$f(\mathrm{x}) = \omega^T x + b$$

in which w is an matrix and "T" means doing the transpose to the matrix,x is a vector so that the product of w(T) and x is a number.
So ,what we need to do is calculate w and b that make fit the equation based on data we get from "core1.mat".
Let us make some definitions:

$$FunctionalMagrin: \hat{\gamma} = y(w^T x + b)$$

$$GeometricMargin: \tilde{\gamma} = \frac{\hat{\gamma}}{\|\omega\|}$$

To make our model more reliable,we need to find w and b that can maximize the geometric margin.
Let us make an assumption that we set the functional margin is constantly equal to 1,so the problem we are meet with has become to maximize 1/丨 w 丨.It's really not easy to work out,but we can find that the dual problem is much easier for us to deal with.
The dual problem is

$$min\frac{1}{2}\|\omega\|^2$$

Obviously,we can deal with it by using Lagrange Function!

So we take the Lagrange Function:

$$L(\omega,b,\alpha)=\frac{1}{2}\|\omega\|^2-\sum_{i=1}^{n}\alpha_i(y_i(\omega^T x+b)-1)$$

After some mathematical processing,we can get some conclusion like below:

$$\omega=\sum_{i=1}^{n}\alpha_i y_i \alpha_i \qquad b=-\frac{1}{2}(\max\omega^T x_i+min\omega^T x_i)$$

So you can see that we can express w and b by using a single variable α.

Next we are going to do is finding an appropriate α.

Since there are some strange points that don't fit our model,we apply Slack Variable so that we can ignore these points.And we use Kernel Functions to deal with non-linear problems.

After these assumptions and complicated definition,now,our task is to find an α *that fit these conditions:*

$$i,min\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j\kappa(x_i,x_j)-\sum_{i=1}^{n}\alpha_i$$
$$ii,0\le\alpha_i\le C$$
$$iii,\sum_{i=1}^{n}\alpha_i y_i=0$$

We are going to apply SMO algorithm to get the α.

Here are the detailed steps:

i,Scanning all the multiplier to find the one that break the KKT condition and set the first one as α(j).

ii,Find an α *that does not break the KKT condition and maximize* $|E\ i-E\ j\ |$ *while the value of E equals f(x)-y.*

iii.Calculate the upper bound,the lower bound and sequential variable η *with sequence showing below:*

$$\begin{cases} L = max\{0, \alpha_j - \alpha_i\} \\ H = max\{C, C + \alpha_j - \alpha_i\} \end{cases}, y_i \neq y_j$$

$$\begin{cases} L = max\{0, \alpha_j + \alpha_i - C\} \\ H = max\{C, \alpha_j - \alpha_i\} \end{cases}, y_i = y_j$$

$$\eta = \kappa(x_i, x_i) + \kappa(x_j, x_j) - 2\kappa(x_i, x_j)$$

iv,Now we can renew α(j) *and* α(i)*as below*

$$\alpha_j : \alpha_j^{new} = \alpha_j^{old} + \frac{y_j(E_i - E_j)}{\eta}$$

$$\alpha_i : \begin{cases} \alpha_j^{new,clipped} = \begin{cases} H, \alpha_j^{new} \geq H \\ \alpha_j^{new}, L < \alpha_j^{new} < H \\ L, \alpha_j^{new} \leq L \end{cases} \\ \alpha_i^{new} = \alpha_i + y_i y_j(\alpha_j - \alpha_j^{new,clipped}) \end{cases}$$

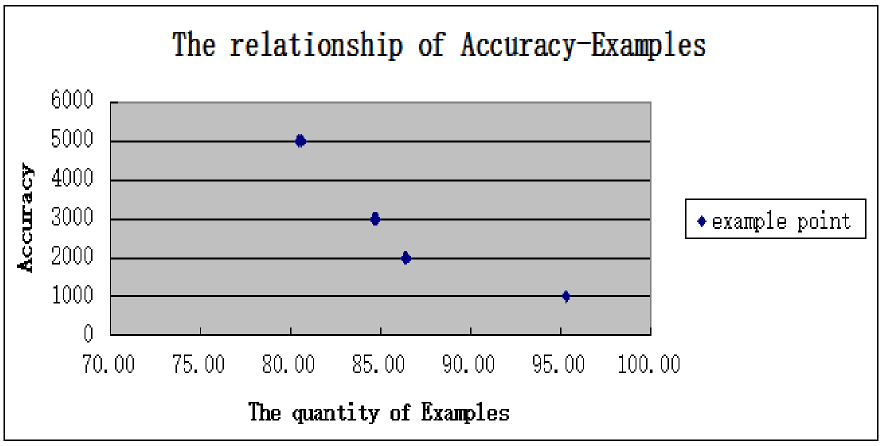
Data Analysis

Examples	*Tolerance*	*C*	*Maxtime*	*The quantity of Yes*	*Accuracy rate*
1-1000	1.00E-03	1000	5	953	95.30
1-2000	1.00E-03	1000	5	1727	86.35
1-2000	1.00E-03	100	5	1729	86.45
1-2000	1.00E-04	1000	5	1728	86.40
1-2000	1.00E-04	100	5	1727	86.35
1-2000	1.00E-03	1	5	1729	86.45
1-3000	1.00E-03	10000	5	2541	84.70
1-3000	1.00E-03	1	5	2542	84.73
1-3000	1.00E-03	0.00000001	5	2539	84.63
1-3000	1.00E-03	10000	8	2543	84.77
1-5000	1.00E-03	10000	5	4024	80.48
1-5000	1.00E-04	10000	5	4032	80.64
1-5000	1.00E-03	100	5	4032	80.64
1-5000	1.00E-03	0.1	5	4032	80.64
1-5000	1.00E-03	0.001	5	4032	80.64

Table A

From Table,we can get some simple conclusions as below:

i,With the same Tolerance,C,Maxtime and different quantity of examples,the accuracy rate is totally different which has shown the regulation that the more examples are,the less accuracy rate is.



ii.With the same Examples,Tolerance,Maxtime and different C,the accuracy rate is not dramatically changed which meas the accuracy may have less relationship with C.

iii,With the same Examples,C,Maxtime and different Tolerance,the accuracy rate doesn't change obviously,but we get the an simple conclusion that the smaller the Tolerance is ,the longer time the program will cost.

iv,With the same Examples,C,Tolerance and Maxtime,the accuracy rate doesn't change obviously,but we get the an simple conclusion that the larger the Maxtime we choose ,the longer time the program will cost.And the effect is considerable!

In these four conclusions we have mentioned,the conclusion ii is actually not true.So let's observe the Table B.

Examples	*The quantity of Yes*	*Accuracy rate*
1-5000	4032	80.62
5001-10000	4073	81.48
10001-15000	4146	82.91
15001-20000	3992	79.85
20001-25000	3740	74.79
25001-30000	3372	67.45
30001-35000	3948	78.95
35001-40000	4064	81.29
40001-45000	4354	87.09
45001-50000	4244	84.88
50001-55000	3897	77.94
55001-60000	4094	81.87
60001-65000	4389	87.79
65001-70000	3496	69.92
70001-75000	3632	72.65
75001-80000	3884	77.68
80001-85000	3737	74.73
85001-90000	4032	80.66
90001-100000	4203	84.04

Table B

In Table B,we keep the quantity of examples,Tolerance,C,and Maxtime while we change the examples,which is from 1-5000 to 90001 to 100000.And we keep C in a tiny value,which is only about 1e-10.So now we get the Table B.

Obviously,the accuracy rate is not stable anyway,it changes dramatically if we change the examples!If we deal with these data statistically,we can get some specific numbers and a simple diagram.

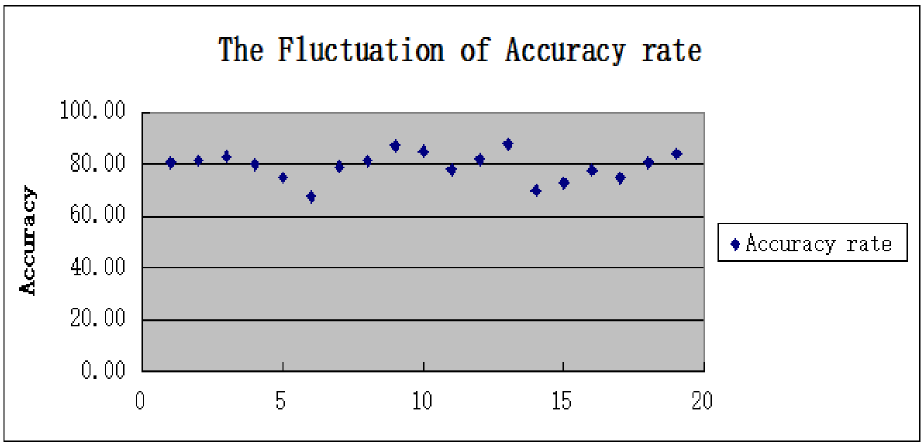


Diagram A

Diagram A shows that if the value of the quantity of examples is a certain number,the value of accuracy rate will fluctuate near a certain number.

And after we do some simple analysis,we can get that the average accuracy rate and the uncertainty:

$$\overline{x} = 79.29, \sigma_x = 5.46$$

Actually the uncertainty is a little bit large which means our model is not accurate enough but since the algorithm is not as impeccable as a professional one ,we still appreciate all that we've done.