Support Vector Machine

Yurong Luo

***Abstract*—Classic statistical learning problems are based on Empirical Risk Minimization (ERM). This kind of learning problems works under very strict assumptions, which may not be satisfied in reality. Moreover, ERM can not trade off bias and variance. Overcoming the assumptions ERM has, Support Vector Machine (SVM) implement Structure Risk Minimization (SRM). The objective of SVM is to create a model with a minimized VC dimension. More specifically, SVM uses a linear separating hyper plane to create a classifier with a maximal margin. Therefore SVM can trade off bias and variance very well. In this paper, in order to master the knowledge of SVM, we intend to realize both linear SVM and nonlinear SVM using polynomial kernel function and Gaussian kernel function.**

***Keywords*—Structure Risk Minimization (SRM), VC dimension, Support Vector Machine (SVM), kernel function**

I. Introduction

Classic statistical learning problems are based on Empirical Risk Minimization (ERM). One of disadvantages of ERM is that it can not decrease generalization error and approximation error at the same time. And it work under very strict assumptions. For example, data can be modeled by a set of linear in parameter functions; a stochastic component of data is the normal probability distribution law, i.e., the underlying joint probability distribution is Gaussian; the induction paradigm for parameter estimation is the maximum likelihood method that is reduced to the minimization of the sum-of-errors-squares cost function. However, all these assumptions may not be satisfied because of the following facts. Modern problems are high-dimensional and scarce, and if the underlying mapping is not very smooth, the linear paradigm needs an exponentially increasing number of terms with an increasing dimensionality of the input space, i.e., with an increase in the number of independent variables; the underlying real-life data generation laws may typically be very far from the normal distribution, etc.

Therefore, in order to deal with the facts mentioned above, SVM is introduced in the framework of SRM and in the theory of VC bounds. It creates a model with a minimized VC dimension. It tells when the VC dimension of model is low, the expected probability of error is low as well, which means good performance on unseen data. More specifically, SVM use a linear separating hyper plane to create a classifier with a maximal margin. So, SVM trade off bias and variance very well.

The rest of paper is organized as follows: In section II, Support Vector Machine is given. The result is presented in section III. The following part is conclusion.

II. Support Vector Machine

Not only SVM can solve classification problems, but also it can be applied in regression. But here, we just introduce three situations in classification.

A. *Linear Maximal Margin Classifier for Non overlapping Classes*

Considering training data are given as

$$(x_1, y_1), (x_2, y_2) \cdots (x_l, y_l), x \in R^n, y \in \{-1, +1\} \tag{1}$$

The purpose of SVM is to find a super hyper plane with the following form

$$d(x, w, b) = w^T x + b \tag{2}$$

Therefore, what SVM does is to minimum of (3) under the condition (4).

$$\frac{1}{2} w^T w \tag{3}$$

$$y_i \left[w^T x_i + b \right] \geq 1, i = 1, l, \tag{4}$$

This problem can be solved either in primal space or in a dual space. The second approach uses the Karush-Kuhn-Tucker (KKT) conditions for the optimum of a constrained function. Finally, it formulated as follows:

$$L_d(\alpha) = -0.5 \alpha^T H \alpha + f^T \alpha \tag{5}$$

Subject to

$$y^T \alpha = 0, \tag{6}$$

$$\alpha > 0 \tag{7}$$

Solutions α of this dual optimization problem determine the parameters w and b of the optimal hyper plane as follow:

$$w = \sum_{i=1}^{l} \alpha_i y_i x_i \tag{8}$$

$$b = \frac{1}{N_{sv}} \left(\sum_{s=1}^{N_{sv}} \left(\frac{1}{y_s} - x_s^T w \right) \right) \tag{9}$$

B. *Linear Soft Margin Classifier for Overlapping Classes*

Different from linear maximal margin classifier for non overlapping classes, SVM minimize the following formulation (10)

$$\frac{1}{2} w^T w + C \left(\sum_{i=1}^{l} \xi_i \right)^k \tag{10}$$

subject to inequality constraints (11)

$$y_i \left[w^T x_i + b \right] \geq 1 - \xi_i, i = 1, l, \xi_i \geq 0 \tag{11}$$

[1]Yurong Luo, Phd student of computer science in Virginia Commonwealth University

Vojislav Kecman, associate professor, department of Computer Science in Virginia Commonwealth University

This problem can also be solved either in primal space or in a dual space. The final formulation is the same as (5) (6), but with the change of α in the following form.

$$C \geq \alpha \geq 0 \tag{12}$$

Then, using (8) and (9) to get the final hyper plane (2).

C. *The Nonlinear Classifier*

The linear classifiers presented in the two previous sections are very limited. However, mostly, not only are classes overlapped but the separation lines are nonlinear hyper surfaces. The solution of SVM in designing nonlinear SVM is to map input vectors $x \in R^n$ into vectors z of a higher-dimensional feature space F, and to solve a linear classification problem in this feature space. The mapping is realized by using the kernel function:

$$K(x_i, x_j) = \phi^T(x_i)\phi(x_j) \tag{13}$$

The role of kernel function is to avoid having to map $\phi(x)$. Instead, the required scalar products in feature space $\phi^T(x_i)\phi(x_j)$ are calculated directly by computing kernels $K(x_i, x_j)$ for given training data vectors in an input space.

Similar with the above two situations, finally, SVM find final hyper plane in the following form:

$$d(x) = \sum_{i=1}^{l} y_i \alpha_i K(x, x_i) + b \tag{14}$$

The solution becomes solve the following formulation

$$L_d(\alpha) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{l} y_i y_j \alpha_i \alpha_j K(x_i, x_j) \tag{15}$$

$$C \geq \alpha \geq 0 \tag{16}$$

$$\sum_{i=1}^{l} \alpha_i y_i = 0 \tag{17}$$

D. *Cross Validation*

The objective of Cross Validation is to find the optimum parameters, i.e., degree of polynomial kernel function, penalty factor C, to make the model more accurate.

The basic idea of Cross Validation is to divide all the datasets into n chunks. This is called n-fold cross validation. Firstly, define your parameters. Secondly, choose one chunk as testing dataset, and the remaining chunks as training dataset. Calculate the total errors after n times. Finally, get the best parameters when the total errors are minimum.

In fact, when implementing cross validation, there are several aspects needed to pay attention to. 1) The dataset must be shuffled to make sure that each chunk not only has positive instances but also negative instances. 2) The dataset should be scaled to a certain scope. 3) All the dataset must be divided only once into n chunks.

III. RESULT

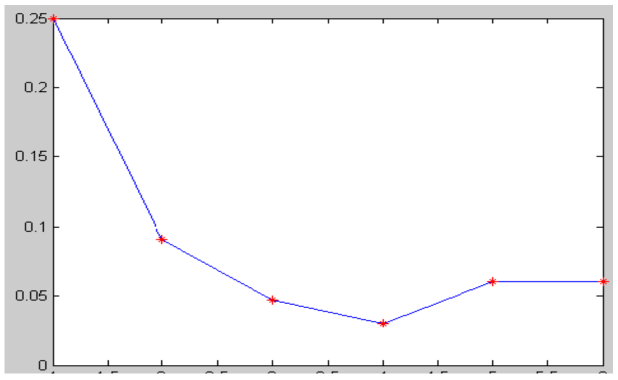
A. *Dataset*

The dataset we use is a well known (and benchmarking) VOTE data. The total number of dataset is 232, and each data has 16 features. The characteristic of the dataset is that it has overlapping data.

B. *Result*

1) Linear Soft Margin Classifier for Overlapping Classes

In this part, we change the penalty factor C (C = [1e-4 1e-3 1e-2 1e-1 1e0 1e1]) and do cross validation. Fig. 1 is the result. From the fig, when C reaches to 1e-1, the error is minimum.



2) Nonlinear Classifier

In this part, we use different kernel function to verify the result.

- Gaussian kernel function

We change the penalty factor C (C = [1e-4 1e-3 1e-2 1e-1 1e0 1e1]), Deviation s (s = [0.5 1 2 5 15]) and do cross validation. Fig. 2 is the result. From the table.1, when C reach to 10 and deviation s reach to 2, the error is minimum.

Table.1 errors with C and s changing

0.4655	0.4655	0.4655	0.4353	0.2241	0.2241
0.4655	0.4655	0.4655	0.4052	0.1595	0.1638
0.4957	0.4655	0.4397	0.0905	0.0474	0.0431
0.4871	0.4655	0.4224	0.0862	0.2500	0.4353
0.4741	0.4655	0.4138	0.1724	0.4612	0.3190

- Polynomial kernel function

We change the penalty factor C (C = [1e-4 1e-3 1e-2 1e-1 1e0 1e1]), degree d (d = [1 2 3]) and do cross validation. Fig. 3 is the result. From the fig, when C reach to 1e-4 and degree d reach to 3, the error is minimum.

Table.1 errors with C and d changing

0.2759	0.1207	0.0474	0.1767	0.3276	0.3362
0.1681	0.1034	0.2371	0.3190	0.1078	0.1078
0.0388	0.1164	0.0474	0.0474	0.0474	0.0474

IV. CONCLUSION

Support Vector Machine is designed in the framework of SRM and in the theory of VC bounds. Not only SVM can implement classification but also regression. In this paper, we do experiment both on linear SVM and nonlinear SVM using polynomial kernel function and Gaussian kernel function. The design of SVM should consider many aspects. For example, before cross validation and classification, the dataset is better to be shuffled and scaled; the calculation of bias just need free support vectors; we should choose the model, which has minimum in error and in support vectors. However, the key point in SVM is to solve QP problem to get penalty factors.

References

[1] Vojislav Kecman. "Learning and Soft Computing: support vector machine, neural networks, and fuzzy logic models". The MIT Press, Cambridge, MA, pp.120-184, 2001