An Implementation of Support Vector Machines on Iris Dataset

Ivanna K. Timotius[1], Iwan Setyawan[2], and Andreas A. Febrianto[3]
Department of Electronic Engineering
Satya Wacana Christian University, Salatiga - Indonesia.
ivanna_timotius@yahoo.com[1], dr.isetyawan@gmail.com[2]

Abstract— **Support Vector Machines (SVM) is a set of related supervised learning method used for classification. SVM is used to construct a hyperplane as the decision surface in such a way that the margin of separation between positive and negative examples is maximized. By default, this hyperplane is linear. To improve the classification performance, it is desirable to use a non-linear hyperplane. In order to construct a non-linear hyperplane using SVM, we use kernel functions. This paper presents a comparison of using several kernel functions in the SVM algorithm for Iris dataset classification.**

Index Terms—**Pattern recognition, support vector machines, kernel functions**

I. Introduction

Pattern recognition is the study of how machines can observe the environment and make reasonable decisions about the categories of the patterns [1]. There are many areas that can benefit from a machine's ability to recognize patterns. These include:

- Bioinformatics: Classification of types of genes by analyzing DNA sequences.
- Industrial automation: Automatic classification of defective/non-defective products by analyzing product images.
- Biometric recognition and detection: Identification of humans based on facial patterns (commonly known as "face recognition" application) or classification of "face" and "non-face" areas of an image.

Some approaches have been investigated for use in pattern recognition. The four best known approaches for pattern recognition are [1]:

- Template matching
- Statistical classification
- Syntactic/structural matching
- Neural networks

Template matching is one of the simplest approaches to pattern recognition. This method is computationally demanding. Furthermore, template matching using rigid templates suffers from recognition problems if the templates are somehow distorted. In the statistical approach, each pattern is represented in terms of d features or measurements and is viewed as a point in a d-dimensional space. This approach aims to choose those features that allow pattern vectors belonging to different categories to occupy compact and disjoint regions in a d-dimensional feature space. Therefore, the key problem in this approach is how to separate the patterns from different classes. The syntactic approach is particularly suitable in applications where the patterns are complex. In this approach, the patterns are viewed as being built of simpler sub-patterns (which are themselves built of even simpler sub-sub-patterns). The relationship between each sub-pattern is treated like the syntax of a language. Therefore, a complex pattern is described using simple sub-patterns that are related to each other according to a certain grammatical rule. Neural networks can be viewed as massively parallel computing systems consisting of an extremely large number of simple processors with many interconnections. The main characteristics of neural networks that make them an attractive approach to pattern recognition are that they have the ability to learn complex nonlinear input-output relationships, use sequential training procedures, and adapt themselves to the data.

One method to recognize pattern is called Support Vector Machine (SVM). This method belongs to the statistical classification approach. SVM is a set of related supervised learning method used for classification. It belongs to a family of generalized linear classifiers. The main idea of a support vector machine is to construct a hyperplane as the decision surface in such a way that the margin of separation between positive and negative examples is maximized. The separating hyperplane is defined as a linear function drawn in the feature space [2]. By using kernel functions, the scalar product can be implicitly computed in a kernel feature space, without explicitly using or even knowing the mapping [3].

The Iris flower data set or Fisher's Iris data set is a multivariate data set introduced by Sir Ronald Aylmer Fisher (1936). The Iris dataset is perhaps the best known dataset found in pattern recognition literature. The dataset consists of 3 classes, 50 instances each and 4 numeric attributes where each class refers to a type of Iris plant namely *Iris setosa*, *Iris versicolor*, and *Iris verginica*. The first class is linearly separable from others while that latter are not linearly separable.

This paper presents a comparison of using several kernel functions in the SVM algorithm for Iris dataset classification. In addition, this paper also compares the results with the linear SVM and Nearest Neighbor classifier.

II. Support Vector Machines

For pattern recognition tasks, the construction of an optimal hyperplane is done in a high-dimensional feature space obtained from a nonlinear mapping. Given a training sample $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$, where $\mathbf{x}_i$ is a training vector and y_i is its class label being either +1 or -1, SVM wants to find the weight vector $\mathbf{w}$ and the bias b of the separating hyperplane such that [2][4]:

$$\begin{aligned} y_i(\mathbf{w}^T\varphi(\mathbf{x}_i)+b) &\geq 1-\xi_i, \quad \forall i \\ \xi_i &\geq 0, \quad \forall i \end{aligned} \tag{1}$$

with $\mathbf{w}$ and the slack variables ξ_i minimizing the cost function:

$$\Phi(\mathbf{w},\xi_i) = \tfrac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^N \xi_i \tag{2}$$

where the slack variables ξ_i represent the error measures of data, C is a user-specified positive parameter (which is the penalty assigned to the errors), and $\varphi(\cdot)$ is a nonlinear mapping which maps the data into a higher dimensional feature space from original input space.

The dual problem of SVM is given as follows. Find the Lagrange multipliers $\{\alpha_i\}_{i=1}^N$ that maximize the objective function:

$$Q(\alpha) = \sum_{i=1}^N \alpha_i - \frac{1}{2}\sum_{i=1}^N\sum_{j=1}^N \alpha_i\alpha_j y_i y_j k(\mathbf{x}_i,\mathbf{x}_j)$$

subject to

$$\begin{aligned} &\sum_{i=1}^N \alpha_i y_i = 0 \\ &0 \leq \alpha_i \leq C \qquad \forall i \end{aligned} \tag{3}$$

where C is a user-specified positive parameter.

Having the Lagrange multipliers, the optimum weight vector $\mathbf{w}_o$ could be computed by

$$\mathbf{w}_o = \sum_{i=1}^N \alpha_i y_i \varphi(\mathbf{x}_i) \tag{4}$$

From the Lagrangian function, according to Kuhn-Tucker [5], by taking the samples with $0 < \alpha_i < C$, the bias could be calculated by

$$b = \frac{1}{\#SV}\sum_{\mathbf{x}_i \in SV}\left(\frac{1}{y_i} - \sum_{\mathbf{x}_j \in SV}\alpha_j y_j k(\mathbf{x}_j,\mathbf{x}_i)\right) \tag{5}$$

where $\#SV$ is the number of support vectors with $0 < \alpha_i < C$. For an unseen data $\mathbf{z}$, its predicted class can be obtained by the decision function:

$$D(\mathbf{z}) = \text{sign}\left(\sum_{i=1}^N \alpha_i y_i k(\mathbf{x}_i,\mathbf{z}) + b\right) \tag{6}$$

The requirement on the kernel $k(\mathbf{x}_i,\mathbf{x}_j) = \varphi^T(\mathbf{x}_i)\varphi(\mathbf{x}_j)$ is to satisfy Marcer's theorem. The two common inner-product kernels types of SVM are polynomial learning machine and radial-basis function network. The polynomial learning machines is given by:

$$k(\mathbf{x}_i,\mathbf{x}_j) = (\mathbf{x}_i^T\mathbf{x}_j + 1)^p \tag{7}$$

where p is the polynomial power and the radial-basis function network is given by:

$$k(\mathbf{x}_i,\mathbf{x}_j) = \exp\left(-\tfrac{1}{2\sigma^2}\left\|\mathbf{x}_i - \mathbf{x}_j\right\|^2\right) \tag{8}$$

where σ is the width which is specified by the user.

SVM classification is essentially a binary (two-class) classification technique. For multiclass case, there are two common methods applied: one-against-one (OAO) and one-against-all (OAA) method. The approach that is used in this paper is OAO method. OAO approach involves constructing a machine for each pair of classes resulting in $l(l-1)/2$ machines for l classes. Each classification gives one vote to the winning class and the point is labeled with the class having most votes.

III. Experiment and Results

The experiment is done by using 10 runs of 2-fold cross validation, where the training dataset contains 75 instances randomly sampled, and the testing dataset contains the remaining 75 instances. Two kinds of kernel functions are used: polynomial learning machine and radial-basis function. The linear SVM (without using any kernel function) are also experimented. For each SVM, the parameters C, σ, and p are determined empirically. The classification rate average of the OAO SVM classifier is compared with the Nearest Neighbor classifier. The classification rate averages are summarized in Table 1.

Table 1. Classification Rate Averages and Number of Support Vector

Method	Classification Rate Average	Average Number of Support Vector
SVM Linear	97.20%	8
SVM Polynomial	95.87%	3.4
SVM Radial Basis Function	97.87%	7.9
Nearest Neighbor	95.20%	(75)

According to eq. (6), the number of support vector in the SVM algorithm denotes the number of training samples used in determining the classification hyperplane. Since Nearest Neighbor classifier uses all of the training data in determining the classification boundary, it is the same as having 75 support vectors, although we never named them by support vector.

IV. Discussion/Conclusion

In general, SVM perform better than the Nearest Neighbor :lassifier. By using the radial-basis function as a kernel lunction, SVM performs better than the other methods. The :omputation time required for the testing task depends on the iumber of support vector. The testing stage of the SVM nethod is faster than the Nearest Neighbor classifier.

Acknowledgment

This work is supported by DP2M DIKTI under the research grant 383/SP2H/PP/DP2M/VI/2009.

References

1] A. K. Jain, R. P. W. Duin, and J. Mao, "Statistical Pattern Recognition: A Review," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, no. 1, 2000.
2] S. Haykin, Neural Network: A Comprehensive Foundation, Prentice-Hall, New Jersey, 1999.
3] K. R. Muller, S. Mika, G. Ratsch, K. Tsuda, and B. Scholkopf, "An Introduction to Kernel-Based Learning Algorithms," IEEE Transactions on Neural Networks, vol. 12, no. 2, 2001.
4] V. Vapnik, Statistical learning theory, Springer, Berlin Heidelberg, New York, 1998.
5] R. Fletcher, Practical Methods of Optimization, New York, Wiley, 1987.