# Using Support Vector Machine by MATLAB

**Team 7**

**Zemeng Wang**

**Theory**

In machine learning, **support vector machines** (**SVMs**) are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. In addition to performing linear classification, SVMs can efficiently perform a non-linear classification using what is called the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces. It is a model to separate two or more data sets by their attributes.

The SVM is a kind of second class classification model. Its basic model is defined as the linear classifier with the largest interval in the feature space. The learning strategy is the interval maximization, which can be transformed into a convex quadratic programming problem.

 For example, this picture shows a data set with two attributes (X1, X2) and two labels (Black and White). The goal is to find a plane or a line to separate two data set.

In this algorithm, we must have a data set with different attributes, and different labels.

As we can see, H1 cannot separate two data sets. H2 could, but the distance is smaller than H3. So H3 is what our need.

The kernel machine is the main function in SVM. It will use a math way to calculate the function of the inner product of the two vectors in the space after implicit mapping. [1]

**Linear Classifier**

Given some data points, they belong to two different classes, and now find a linear classifier to divide the data into two categories. If x is the data point and y is the category (y can take 1 or -1, representing two different classes), the learning objective of a linear classifier is to find a hyperplane in the n-dimensional data space (Hyper plane), this hyperplane equation can be expressed as (T in wT means transpose):

           [2]

And the value of y could be 1 or -1, that is because of Logistic Regression：



Where x is the n-dimensional eigenvector and the function g is the logistic function.

And the image of  is:


It can be seen that the infinite is mapping to (0,1).

And assume that the function is the probability that the feature belongs to y = 1.



Thus, when we want to distinguish a new feature belongs to which class,

only need to get . If more than 0.5 belongs to y = 1 class, otherwise belong to y = 0 class. [5]

**Definition of Maximum Margin Classifier**

For a data point, the greater the "interval" of the hyperplane from the data point, the greater the confidence of the classification. Therefore, in order to make the confidence of the classification as high as possible, it is necessary to let the selected hyperplane be able to maximize this "interval" value. This interval is half of the Gap in the figure below.



From the previous analysis we can see that the function interval is not suitable for maximizing the interval value because after the hyperplane is fixed, the length of w and the value of b can be scaled. That could make the value of  be arbitrarily large. The function interval  can be arbitrarily large in the case where the hyperplane is kept constant. But the geometric interval is divided by , so that when the scale of w and b geometric interval value is not changed, it only changes with the hyperplane changes, so this is a more appropriate interval. In other words, the "interval" in the hyper grid is the geometric interval.

Thus, the objective function of the maximum margin classifier can be defined as:

[4]

Equivalent to the corresponding constraints in the conditions:, to maximize the 1 / || w || value, and 1 / || w || is the geometric interval.

**Kernel function**

Before we encounter a kernel function, if we use the original method, then we use a linear learner to learn a nonlinear relationship, we need to select a non-linear feature set, and write the data into a new form of expression, which is equivalent to applying a fixed, The data is mapped to the feature space and the linear learner is used in the feature space. Therefore, the hypothesis set considered is a function of this type:



This means that the establishment of a nonlinear learner is divided into two steps: First, a non-linear mapping is used to transform the data into a feature space F, and then use linear learner classification in feature space.
And because the dual form is an important property of the linear learner, which means that the assumption can be expressed as a linear combination of training points, so the decision rules can be used to test points and training points to represent the inner product:



If there is a way to directly compute the inner product <φ (xi · φ (x)) in the feature space, as in the function of the original input point, it is possible to combine the two steps together to establish a non-linear Learning method, so that the method of direct calculation method is called kernel function method:

[3]

**LIBSVM for MATLAB**

LIBSVM is an integrated software for support vector classification, (C-SVC, nu-SVC), regression (epsilon-SVR, nu-SVR) and distribution estimation (one-class SVM). It supports multi-class classification.

Main Function:

Libsvmread (): Read the data set

Svmtrain (): Use the training data to train the model

Svmpredict (): Predict the testing data [6]

**Data**

I use two datasets as my data in MATLAB:

Heart\_scale:

Which is an example data set in LIBSVM, it has 2 classes (good heart &bad heart) and 13 features, with 270 data. And I use the same dataset as my training and testing data.

 Adult\_Data\_Set:

Which is a predict whether income exceeds $50k/year based on census data from UCI. The original Adult data set has 14 features, among which six are continuous and eight are categorical. In this data set, continuous features are discretized into quantiles, and each quantile is represented by a binary feature. Also, a categorical feature with m categories is converted to m binary features. [7]

**Result**

**Heart:**

****

**Adult:**

****

**The meaning of the result:**

#iter: number of iterations

nu: set the parameter nu of nu-SVC, one-class SVM, and nu-SVR (default 0.5)

obj: the minimum value of the SVM file converted to the quadratic programming solution

rho: the constant term b of the decision function

nSV: the number of Support Vector

nBSV: the number of Border Support Vector

**Reference**

[1].Theodoridis, Sergios; and Koutroumbas, Konstantinos; "Pattern Recognition", 4th Edition, Academic Press, 2009, ISBN 978-1-59749-272-0

[2]. Cristianini, Nello; and Shawe-Taylor, John; An Introduction to Support Vector Machines and other kernel-based learning methods, Cambridge University Press, 2000. ISBN 0-521-78019-5 (SVM Book)

[3]. Huang, Te-Ming; Kecman, Vojislav; and Kopriva, Ivica (2006); [Kernel Based Algorithms for Mining Huge Data Sets](http://learning-from-data.com/), in Supervised, Semi-supervised, and Unsupervised Learning, Springer-Verlag, Berlin, Heidelberg, 260 pp. 96 illus., Hardcover, ISBN 3-540-31681-7

[4]. Kecman, Vojislav; Learning and Soft Computing — Support Vector Machines, Neural Networks, Fuzzy Logic Systems, The MIT Press, Cambridge, MA, 2001.

[5].  Cortes, C.; Vapnik, V. Support-vector networks. Machine Learning. 1995, **20** (3): 273–297. doi:10.1007/BF00994018.

[6]. LIBSVM—A Library for Support Vector Machines: **h**ttps://www.csie.ntu.edu.tw/~cjlin/libsvm/

[7]. UCI Machine Learning Repository: http://archive.ics.uci.edu/ml/