# Title of the Seminar Paper

*Seminar Data Mining*

Chengjie Zhou
Department of Informatics
Technische Universität München
Email: chengjie.zhou@in.tum.de

***Abstract*— Text of the abstract**
***Keywords*— Data Mining**

## I. Introduction

One of the most important ascpects of machine learning is classification. Typical algorithms and models that are used for classification include logistic regression, naive bayes, decision trees and so on. In the early 90s, Vladimir Vapnik and his colleagues developed a new algorithm called *Support Vector Machine (SVM)*, which is an algorithm that is optimized for classification and regression analysis. In this paper, we will focus on the main usages of SVM, including the generalization of linear decision boundaries for classification. We will also discuss the role of kernel in SVM, as well as the implementation, evalution methods, application and many different aspects of SVM.

In order to thoroughly understand the classification problem, we first need to look at a simple example in one dimension. Suppose there are two groups of data points that are distributed separately on the number line. The classification then becomes obvious: The threshold that separates both groups simply lies in the middle of the the most outward data points from both groups. This classification method is called the *maximum margin classifier*, since the margin that separates both groups is maximized.

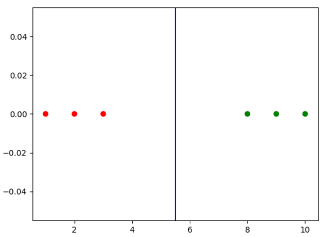


Fig. 1. Maximum Margin Classfier

Nevertheless, the maximum margin classifier is not always applicable, since the data points are not always idealy distributed in a separated way. If an outlier happens to appear in the dataset, it would push the threshold to one side, since the data point is much closer to the other group. This would result in a severe misclassification, since the data that are close to one group now belongs to the other group because of the shift of the threshold. A solution to this problem is to allow some misclassification, so that the threshold has higher bias and is less sensitive to outliers and the classifier performs better when there is new data. This margin that allows some misclassification is called the *soft margin*. The determination of soft margin could be tricky, since there are limitless points of thresholds to choose from. One way to find the optimal threshold is to use Cross Validation. Cross Validation is a method that splits the dataset into several parts. For each repetition, one part of the dataset is used for testing and the rest is used for training. After training through all the repetition, the average position of the threshold represents the most ideal position of the soft margin. This classifier is called soft margin classifier, also known as support vector classifier. The name ”Support Vector” derives from the fact that the data points that are closest to the threshold are called support vectors.

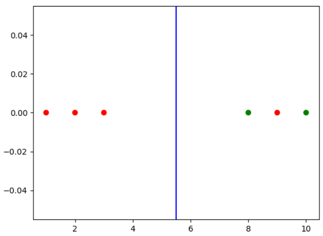


Fig. 2. Support Vector Classfier

In reality, chances are that the data is not only one-dimensional. In fact, almost all of the data that we work with is multi-dimensional. In order to represent the data in a multi-dimensional space, we need to use vectors. In this case, the idea of hyperplane is introduced. A *hyperplane* is a threshold that separates the data in a multi-dimensional space, which is formally defined as a flat affine subspace. [1] The support vector classifier is able to deal with this case as well. It finds the hyperplane that separates the data in a way that the margin is maximized. The exact algorithms and the mathematical derivation will be discussed in the next section.

However, one problem is still not solved. Even though the support vector classifiers allows misclassifications and is less sensitive to outliers, it is still not suitable for data that is not linearly separable.

Usually, we use a *kernel* to deal with this case, which

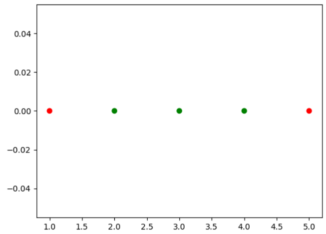


Fig. 3. Non Linearly Separable Data

is a function that maps the data into a higher dimensional space, so that the data becomes linearly separable. The SVM implements the idea of kernel without transforming the data into a higher dimensional space. This concept is called *the kernel trick* and is a crucial part of SVM. This trick allows SVM to cast nonlinear variants to dot products, enabling easier computation and better performance. (55) [2]

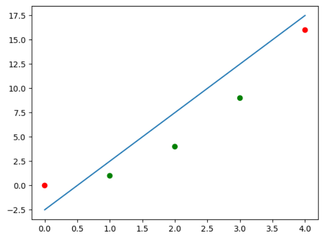


Fig. 4. Kernel Transformation

## II. Implementation

The problem of support vector machine is an optimization problem. The goal is, as mentioned, to find the optimal hyperplane that separates the data in a way that the margin is maximized. The separation of hyperplane is, in this case, fundamental to the functionality of SVM.

The mathematical definition of hyperplane is given by: (Page 418 elements learning)

$$\{x \in \mathbb{R}^p : x^T\beta + \beta_0 = 0\} \tag{1}$$

where $\beta$ is the vector of coefficients and $\beta_0$ is the intercept. In this case, the margin problem becomes a problem of separating the hyperplane. This procedure constructs linear decision boundaries that attempt to separate the data into two or more classes as precisely as possible. To solve this problem, there are two mechanics available: Rosenblatt and Optimal.

The *Rosenblatt's perceptron learning algorithm* aims to find the optimal hyperplane in an iterative way. The algorithm minizies the distance of misclassified points to the decision boundary. The goal of this algorithm is to minimize the following equation

$$\sum_{i \in M} -y_i(x_i^T\beta + \beta_0) \tag{2}$$

where $M$ is the set of misclassified points and $y_i$ is the class label of $x_i$, being either 1 or -1. Using stochastic gradient descent, the algorithm updates the coefficients and intercept. However, the perceptron learning algorithm is not deterministic, since the result depends on the starting values. Furthermore, the algorithm does not converge if the data is not linearly separable, resulting in an infinite loop if the case was not detected beforehand.

The *optimal separating hyperplane* aims to find the hyperplane that maximize the distance to the closet point from either class. (Vapnik). The problem can be mathematically defined as:

$$\max_{\beta,\beta_0} M \tag{3}$$

(Page 130 elements)
TODO: Optimal (Page 132 elements)
The kernel function is defined as:

$$K(x_i, x_j) = \phi(x_i)^T\phi(x_j) \tag{4}$$

## III. Chapter-1

### A. Subchapter

blabla with three references [3]–[5]

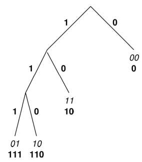


Fig. 5. Tree

TABLE I
BEISPIELTABELLE

| Spalte1 | Spalte2 |
|---|---|
| 0 | 1 |

## IV. Summary and Outlook

blabla

## References

[1] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, second edition ed. New York: Springer, 2009.
[2] T. Hofmann, B. Schölkopf, and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Cambridge, MA: The MIT Press, 2002.
[3] B. Claise, "IPFIX protocol specifications," Internet-Draft, draft-ietf-ipfix-protocol-07, December 2004.
[4] A. C. Snoeren, C. Partridge, L. A. Sanchez, C. E. Jones, F. Tchakountio, S. T. Kent, and W. T. Strayer, "Hash-based IP traceback," in *ACM SIGCOMM 2001 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, 2001.
[5] A. Belenky and N. Ansari, "IP traceback with deterministic packet marking," *IEEE Communications Letters*, vol. 7, no. 4, pp. 162–164, 2003.