utb.

Floris Ernst | Achim Schweikard

Fundamentals of Machine Learning

Support Vector Machines Made Easy

utb 5251

Eine Arbeitsgemeinschaft der Verlage

Böhlau Verlag · Wien · Köln · Weimar
Verlag Barbara Budrich · Opladen · Toronto
facultas · Wien
Wilhelm Fink · Paderborn
Narr Francke Attempto Verlag / expert verlag · Tübingen
Haupt Verlag · Bern
Verlag Julius Klinkhardt · Bad Heilbrunn
Mohr Siebeck · Tübingen
Ernst Reinhardt Verlag · München
Ferdinand Schöningh · Paderborn
transcript Verlag · Bielefeld
Eugen Ulmer Verlag · Stuttgart
UVK Verlag · München
Vandenhoeck & Ruprecht · Göttingen
Waxmann · Münster · New York
wbv Publikation · Bielefeld

Floris Ernst | Achim Schweikard

Fundamentals of Machine Learning

Support Vector Machines Made Easy

UVK Verlag · München

Autoren

Prof. Dr. Floris Ernst und Prof. Dr. Achim Schweikard lehren KI (Künstliche Intelligenz) und Robotik an der Universität Lübeck.

Authors

Prof. Dr. Floris Ernst and Prof. Dr. Achim Schweikard teach AI (artificial intelligence) and robotics at the University of Lübeck.

Umschlagabbildung: © 4X-image – iStock

Bibliografische Information der Deutschen Bibliothek
Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über http://dnb.ddb.de abrufbar.

1. Auflage 2020

© UVK Verlag 2020
– ein Unternehmen der Narr Francke Attempto Verlag GmbH + Co. KG
Dischingerweg 5 · D-72070 Tübingen

Das Werk einschließlich aller seiner Teile ist urheberrechtlich geschützt. Jede Verwertung außerhalb der engen Grenzen des Urheberrechtsgesetzes ist ohne Zustimmung des Verlags unzulässig und strafbar. Das gilt insbesondere für Vervielfältigungen, Übersetzungen, Mikroverfilmungen und die Einspeicherung und Verarbeitung in elektronischen Systemen.

Internet: www.narr.de
eMail: info@narr.de

Einbandgestaltung: Atelier Reichert, Stuttgart
CPI books GmbH, Leck

utb-Nr. 5251
ISBN 978-3-8252-5251-9 (Print)
ISBN 978-3-8385-5251-4 (ePDF)

Contents

Preface

When the first methods for machine learning emerged in the 1970s, it quickly became apparent that the algorithms depended very much on the target field of the application. Examples for such fields were robotics, medical diagnosis, speech recognition, image understanding, chess systems and geometric reasoning. For each application, a new set of methods emerged. This gave rise to the question, whether one could find more general principles, potentially applicable to all domains, and yet powerful enough to solve relevant problems in end-user applications. More recently, when faster computers, and large data sets became available, such principles were found. As one first example, classification and regression with support vector machines was successfully applied to a large range of applications. However, basic methods still must be customized to an application. To be able to do that, users must understand the basic mathematical principles underlying machine learning. The goal of our book is to introduce the methods for machine learning precisely for this purpose: to provide an accessible, yet sufficiently complete basis for machine learning, which will enable users to develop their own methods for new fields. Our book was written for a one-semester course in machine learning at TU München and the University of Lübeck. Students are graduates in Computer Science, Electrical Engineering, Robotics, or Applied Mathematics.

Part I | Support Vector Machines

1 Symbolic Classification and Nearest Neighbour Classification

1.1 Symbolic Classification

Suppose we wish to classify unknown data by computer. Thus, we present data strings describing molecules to the computer, and ask: does this molecule have a specific function in biology? Likewise, the data could also be attributes describing symptoms of diseases instead of molecules. Thus, the attributes could be fever, headache, nausea, etc. We now ask whether a specific disease is present or not.

In both cases, the goal is to classify positive and negative samples. There are two phases: We first present data strings which already have received a classification label by a human. This first phase will be called training phase. We then have a production phase, where we present unclassified data to the system. If the training algorithm works, the system should then be able to decide whether the new data string is a positive or a negative sample.

Thus, we have the following situation:

Training Phase

Input: Sample objects, marked as positive or negative examples.

Goal: To learn which properties or attributes distinguish positive from negative examples.

Production Phase

Input: New sample object, without label.

Output: Label of the new object (states whether it is a positive or negative sample).

A second goal could be: Find rules from facts. We consider again training samples. Each sample is given as a vector with attributes. The samples are either positive or negative, and each attribute can take attribute values.

1.2 Nearest Neighbour Classification

Suppose now, each of our samples has two values: the size and the weight of an object. Clearly, the size and the weight of an object are correlated.

Suppose further, we have two types of objects. Our objects are either metal or wooden.

Marking metal with a '+', and wooden with a '−', we can plot our data. The graph may look like this:

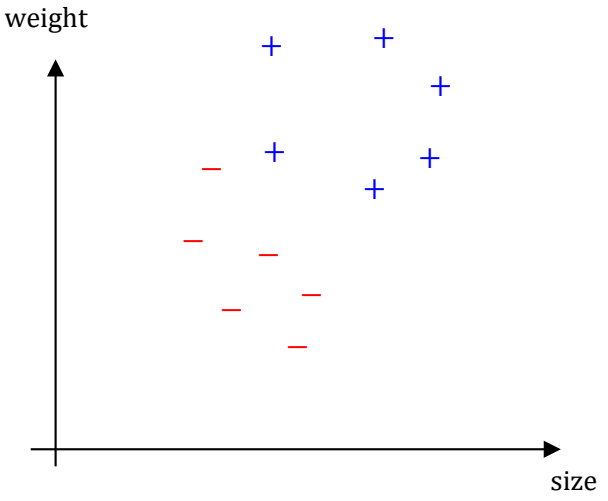

Figure 1 – Example of two classes of objects, plotted by size and weight

The idea in nearest neighbor classification is the following: Given an unknown point, classify it in the same way as its nearest (known) neighbor point in space. To do so, we must find the nearest neighbor of a point U. Here, U is the new, unclassified sample. To do this, we have to perform the following steps:

❶ Let U be the new, unclassified sample
❷ Define a metric $d(U, X)$, the distance of the points U and X in some space
❸ Find the nearest neighbor N of a point U using $d(U, \cdot)$
❹ Assign the class of N to U

If we again look at Figure 1 and add an unknown sample, marked $*$, we will arrive at the following:

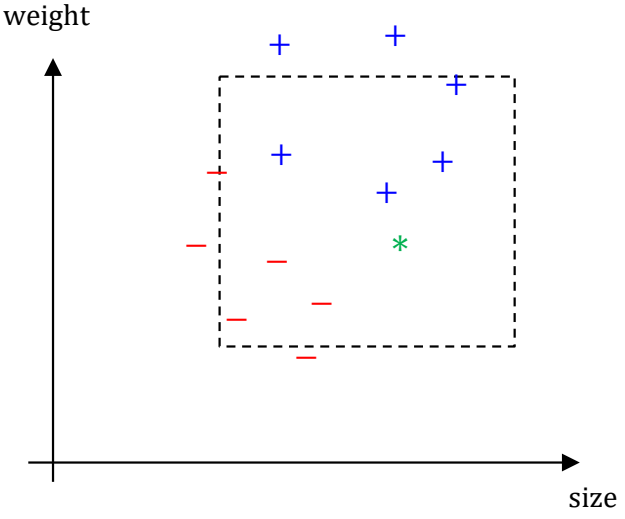

Figure 2 – The same samples as in Figure 1, with a new, unknown sample (marked $*$)

Now, using the metric d as the Euclidean distance, we will find – looking at the dashed region from Figure 2 – the following:

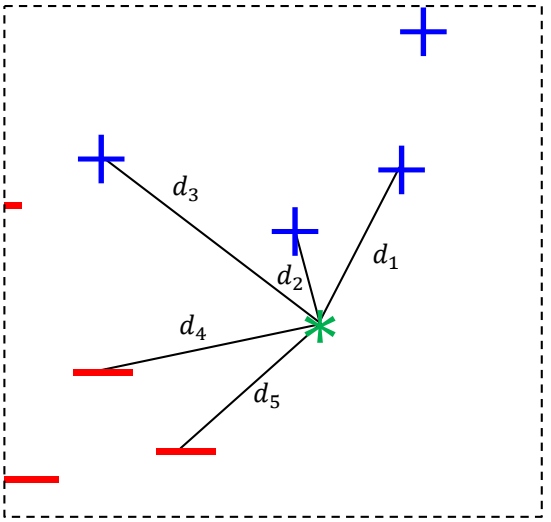

Figure 3 – Subset of Figure 2, showing the distances d between the new sample and the nearest old samples

This gives rise to computing the distances d_1 through d_5 as:

$$d_1 = 3.12 \quad d_2 = 1.77 \quad d_3 = 4.94 \quad d_4 = 4.06 \quad d_5 = 3.43$$

Since the value d_2 is lowest, the corresponding class ($+$) is assigned to the new sample $*$.