Support Vector Machines

Prof. Eric A. Suess

April 27, 2020

Introduction

Today we will discuss Support Vector Machines (SVM).

SVM defines a line/surface/hyperplane that divides the data in the feature space.

SVM can model complex relationships in data.

SVM can be used for **classification** and **numeric prediction**.

SVM have been used successfully in **pattern recognition**.

SVMs are most easily understood when used for **binary classification**.

SVM binary classification

SVMs use a linear boundary called a **hyperplane** to partition data into groups of similar elements, indicated by **class values** or labels.

When data can be separated, it is said to be **linearly separable**.

SVMs can also be used with that data is not **linearly separable**.

Sometimes there are many dividing lines.

Finding the maximum margin

The **Maximum Margin Hyperplane (MMH)** creates the greatest separation between the two classes.

The line that gives the greatest separation will **generalize** the best to the future data.

The **support vectors** are the points from each class that are the closest to the MMH. Using the support vectors, it is possible to define the MMH.

The support vectors provide a very compact way to store a classification method.

SVM binary classification

For more details see,

Support-vector network, Machine Learning, Vol. 20, pp. 273-297, by C. Cortes and V. Vapnik (1995).

The case of linearly separable data

When the classes are linearly separable, the MMH is as far away as possible from the outer boundaries of the two groups of data points.

The outer boundaries are called the **convex hull**.

The HMM is the perpendicular bisector of the shortest line between the two convex hulls.

A technique called **quadratic optimization** is used to find the maximum margin hyperplane.

The case of non-linearly separable data

What happens in the case when the data are not linearly separable?

The solution to this issue is the use of a **slack variable**, which creates a **soft margin** that allows some points to fall on the incorrect side of the margin.

A cost value **C** is applied to the points that violate the contraints.

Then the algorithm attemps to minimize the total cost.

Higher the value of **C**, the harder the algorithm tries for 100% separation. So a balance needs to be reached.

Using kernels for non-linear spaces

In many real-world applications, the relationships between variables are non-linear.

A key feature of SVMs is their ability to map the problem into a higher dimensional space or transformed space using a process called the **kernel trick**.

Doing so, *a non-linear relationship may become quite linear*.

Latitude versus Longitude mapped to **Altitude versus Longitude**

Using kernels for non-linear spaces

SVMs with non-linear **kernels** add additional dimensions to the data in order to create separation.

Altitude can be expressed mathematically as an interaction between latitude and longitude.

This allows the SVM to learn concepts that were not explicitly measured in the orginal data.

Strengths of SVMs

- can be used for classification and numeric prediction
- not influences by noisy data
- not prone to overfitting
- may be easier to use than neural networks

Weaknesses of SVMs

- need to try different kernels
- slow to train
- hard to interpret

kernel selection

There is **no rule** for matching a kernel to a particular learning task.

The fit depends heavily on the concept to be learned as well as the amount of traning data and the relationships among the features.

The *choice of kernel is a bit arbitrary*.

Performing OCR with SVMs

The author uses SVMs with an image processing example.

Optical Character Recognition (OCR)

This example actually has at least as much pre-processing as the naive Bayes example.

UCI Machine Learning Repository

Letter Recognition