

Introduction to the Perceptron and the application of pattern recognition techniques on real world problems

Peter Pollak, Stefan Sietzen, Albrecht Vlk

January 19, 2014

Abstract

In the following article, we will discuss the application of the Perceptron to calculate the weight vector for a specific training set. We will also show an example of applying different pattern recognition techniques on a real world problem.

1 Theory of the used techniques

First we are going to explain what a Perceptron is and which pattern recognition techniques we have used in our research.

1.1 Perceptron

The perceptron was invented in 1957 by Frank Rosenblatt and is a type of a linear classifier, which means that it makes its predictions based on a linear predictor function combining a set of weights with the feature vector.

The perceptron maps its input value x to an output value $f(x)$ using the function

$$f(x) = \begin{cases} 1, & \text{if } w \cdot x + b > 0 \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

In this function, w describes a vector of real-valued weights, while $w \cdot x$ is the dot product, and b is the constant **bias** which is independent from any input value. Depending on whether the value of $f(x)$ is 0 or 1, x is classified either as a positive or negative instance, if we are dealing with a binary classification problem. If a negative value is chosen for b , the weighted combination of inputs must provide a greater value than $|b|$ to be able to reach a value over 0. In general, the bias alters the position of the decision boundary.

A problem with the perceptron algorithm is that it won't terminate if the

learning set is not linearly separable. If that is not the case, it is not possible to classify all vectors properly. A famous example for that problem is the *Boolean Exclusive-Or problem*.

If the perceptron is used in context of artificial neural networks, it is used an artificial neuron which is using the *Heaviside step function* as the activation function. In combination with neural networks, one must also differ between a **single-layer-perceptron** and a **multi-layer-perceptron**. While a SLP consists of just a single layer of nodes in a directed graph, the MLP can work with several layers together.

1.2 kNN Classifier

The *k-Nearest Neighbors Algorithm* is a non-parametric method which can be used for classification and regression. In our work, we have only experimented with the kNN algorithm as classifier. In this case, the output of the algorithm is a class membership where an object is classified by its neighbors. The new object checks which class is most common among its k nearest neighbors and is then assigned to that class.

1.3 Mahalanobis Distance

The Mahalanobis Distance is a descriptive statistic that provides a relative measure of a data point's distance from a common point and was introduced by P.C. Mahalanobis in 1936. It can be defined in several ways, either by a distance of a multivariate vector x from a group of values with mean μ and a covariance matrix s

$$D_M(x) = \sqrt{(x - \mu)^T S^{-1} (x - \mu)} \quad (2)$$

or as a dissimilarity measure between two random vectors \vec{x} and \vec{y} of the same distribution with the covariance matrix S .

$$d(\vec{x}, \vec{y}) = \sqrt{(\vec{x} - \vec{y})^T S^{-1} (\vec{x} - \vec{y})} \quad (3)$$

The Mahalanobis distance is mostly used in cluster analysis and classification techniques, for instance to detect outliers in the development of linear regression models, and was also widely used in biology.

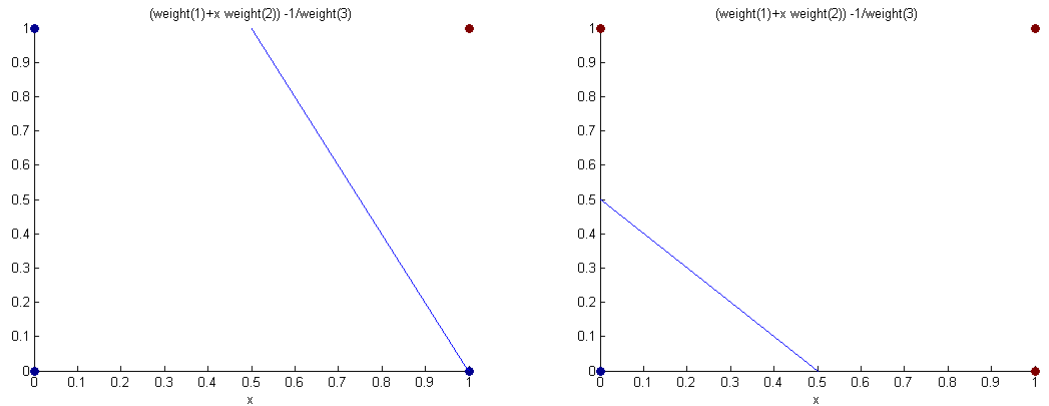
2 Perceptron stuff

Hier kommt die erste Aufgabe hin

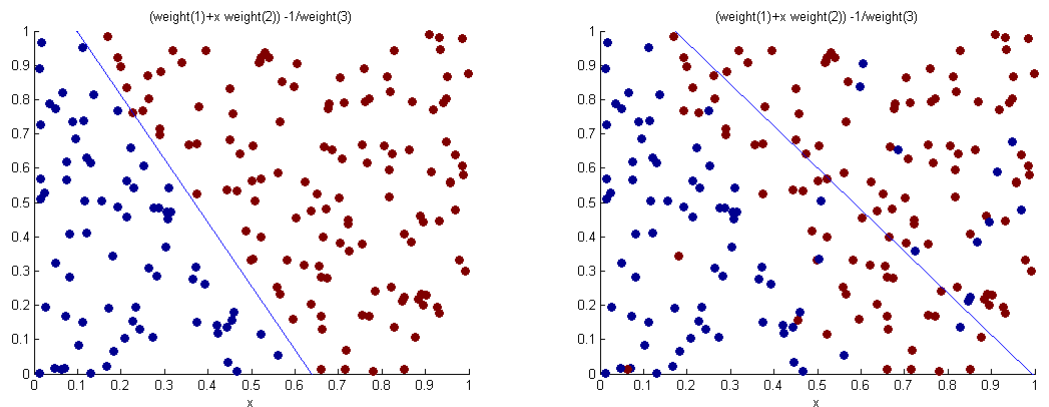
Apply the perceptron to the following problems: OR, AND and XOR. Do not forget to convert your training vectors into homogeneous co ordinates. Discuss the results and the convergence of the algorithm.

Our algorithm works well for linearly separable problems like AND and OR. As expected the algorithm is not able to fully classify XOR since it's not linearly separable. The lines calculated by our algorithm were calculated as shown in the following images.

Points that are classified as +1 and therefore supposed to be above the line are shown in red, points that were supposed to be -1 and below or on the line are shown in blue.



The following images show a dataset of 200 entries with the points classified differently. The image on the left has the points classified in a way so that they're linearly separable and our algorithm performs ideally, while the points in the image on the right are scattered in a way that they can't be distinguished by a line. The drawn line separates the points in a way that 142 of them are classified correctly.



3 Practical Application

Hier erklären wir was wir jetzt vorhaben, wie die Trainingssets bzw Eingangsdaten aussehen und was dann hoffentlich passieren wird / was wir versuchen herauszufinden

Prepare a scientific report comparing the performance of the three classification algorithms on your chosen dataset on the two problems described above. Include graphs showing the different classification rates. Discuss and interpret the results. Can you think of good reasons for the different results produced by the different classifiers? Which combination of the 10 rejected infrared light wavelengths (column 1-10 at file strokefeatures.mat) leads to the best results? Does using fewer features improve the performance? What is a good way to choose the best features to keep? Is your classifier good enough to be used in real life? Your discussion should be precise and scientific.

3.1 Perceptron Stuff

Wie hat das ganze mit Perceptron funktioniert. Verwendete Datasets, Eingaben, Ausgaben, Analyse der Ausgaben: wie richtig ist das ganze, funktioniert es in allen Fällen oder in manchen nicht? Wo (mit welchen Sets / Eingaben) funktioniert es besser, wo schlechter?

3.2 kNN Stuff

Same for kNN

3.3 Mahalanobis stuff

same for mahalanobis

4 Comparison Perceptron / Pattern Recognition

Wie haben die Methoden im Vergleich zueinander abgeschnitten? Wo lagen bei ihnen Schwächen und Stärken? Genauigkeit?

5 Extra-Aufgabe?

6 Conclusion

Hier kommt dann noch eine kurze Zusammenfassung rein.