The Perceptron with Dynamic Margin

Constantinos Panagiotakopoulos and Petroula Tsampouka

Physics Division, School of Technology
Aristotle University of Thessaloniki, Greece
costapan@eng.auth.gr, petroula@gen.auth.gr

Abstract. The classical perceptron rule provides a varying upper bound on the maximum margin, namely the length of the current weight vector divided by the total number of updates up to that time. Requiring that the perceptron updates its internal state whenever the normalized margin of a pattern is found not to exceed a certain fraction of this dynamic upper bound we construct a new approximate maximum margin classifier called the perceptron with dynamic margin (PDM). We demonstrate that PDM converges in a finite number of steps and derive an upper bound on them. We also compare experimentally PDM with other perceptron-like algorithms and support vector machines on hard margin tasks involving linear kernels which are equivalent to 2-norm soft margin.

Keywords: Online learning, classification, maximum margin.

1 Introduction

It is a common belief that learning machines able to produce solution hyperplanes with large margins exhibit greater generalization ability [21] and this justifies the enormous interest in Support Vector Machines (SVMs) [21] [2]. Typically, SVMs obtain large margin solutions by solving a constrained quadratic optimization problem using dual variables. In their native form, however, efficient implementation is hindered by the quadratic dependence of their memory requirements in the number of training examples a fact which renders prohibitive the processing of large datasets. To overcome this problem decomposition methods [15] [6] were developed that apply optimization only to a subset of the training set. Although such methods led to improved convergence rates, in practice their superlinear dependence on the number of examples, which can be even cubic, can still lead to excessive runtimes when large datasets are processed. Recently, the so-called linear SVMs [7] [8] [13] made their appearance. They take advantage of linear kernels in order to allow parts of them to be written in primal notation and were shown to outperform decomposition SVMs when dealing with massive datasets.

The above considerations motivated research in alternative large margin classifiers naturally formulated in primal space long before the advent of linear SVMs. Such algorithms are mostly based on the perceptron [16] [12], the simplest online learning algorithm for binary linear classification. Like the perceptron,
they focus on the primal problem by updating a weight vector which represents at each step the current state of the algorithm whenever a data point presented to it satisfies a specific condition. It is the ability of such algorithms to process one example at a time\(^1\) that allows them to spare time and memory resources and consequently makes them able to handle large datasets. The first algorithm of that kind is the perceptron with margin\(^3\) which is much older than SVMs. It is an immediate extension of the perceptron which provably achieves solutions with only up to 1/2 of the maximum margin\(^10\). Subsequently, various algorithms succeeded in approximately attaining maximum margin by employing modified perceptron-like update rules. Such algorithms include ROMMA\(^11\), ALMA\(^5\), CRAMMA\(^19\) and MICRA\(^20\). Very recently, the same goal was accomplished by a generalized perceptron with margin, the margitron\(^14\).

The most straightforward way of obtaining large margin solutions through a perceptron is by requiring that the weight vector be updated every time the example presented to the algorithm has (normalized) margin which does not exceed a predefined value\(^17, 18, 1\). The obvious problem with this idea, however, is that the algorithm with such a fixed margin condition will definitely not converge unless the target value of the margin is smaller than the unknown maximum margin. In an earlier work\(^14\) we noticed that the upper bound \(\|a_t\|/t\) on the maximum margin, with \(\|a_t\|\) being the length of the weight vector and \(t\) the number of updates, that comes as an immediate consequence of the perceptron update rule is very accurate and tends to improve as the algorithm achieves larger margins. In the present work we replace the fixed target margin value with a fraction \(1 - \epsilon\) of this varying upper bound on the maximum margin. The hope is that as the algorithm keeps updating its state the upper bound will keep approaching the maximum margin and convergence to a solution with the desired accuracy \(\epsilon\) will eventually occur. Thus, the resulting algorithm may be regarded as a realizable implementation of the perceptron with fixed margin condition.

The rest of this paper is organized as follows. Section 2 contains some preliminaries and a motivation of the algorithm based on a qualitative analysis. In Sect. 3 we give a formal theoretical analysis. Section 4 is devoted to implementational issues. Section 5 contains our experimental results while Sect. 6 our conclusions.

### 2 Motivation of the Algorithm

Let us consider a linearly separable training set \(\{(x_k, l_k)\}_{k=1}^{m}\), with vectors \(x_k \in \mathbb{R}^d\) and labels \(l_k \in \{+1, -1\}\). This training set may either be the original dataset or the result of a mapping into a feature space of higher dimensionality\(^21, 2\). Actually, there is a very well-known construction\(^4\) making linear separability always possible, which amounts to the adoption of the 2-norm soft margin. By placing \(x_k\) in the same position at a distance \(\rho\) in an additional dimension, i.e. by extending \(x_k\) to \([x_k, \rho]\), we construct an embedding of our data into the so-called augmented space\(^3\). This way, we construct hyperplanes possessing bias

---

\(^1\) The conversion of online algorithms to the batch setting is done by cycling repeatedly through the dataset and using the last hypothesis for prediction.