The Perceptron Algorithm

Dan Ventura

January 8, 2009

Abstract

We give a basic overview of the Perceptron learning algorithm and discuss its relationship to logistic regression. We also give proof of convergence. Note in what follows we assume some familiarity with mathematical notation and with linear and logistic regression.

1 Introduction

The Perceptron learning algorithm is an iterative, data-driven, gradient descent-based technique for finding a best fit sigmoidal function, given a set of tuples of data consisting of pairs of independent variables (inputs) together with an associated dependent variable (output), $\{(\vec{x}, y)_i\}$.

2 The Model

We assume our data can be fit with a model of the form

$$f(x) = \frac{1}{1 + e^{-\vec{w}^T \tilde{x}}} \tag{1}$$

where $\vec{w}^T$ is the transposed (row) vector of weights (free parameters in the hyperplane equation):

$$\vec{w}^T = [w_0, w_1, \ldots, w_n] \tag{2}$$

and $\tilde{x}$ is the (column) vector of independent (input) variables augmented in the initial position with a 1 (for the offset parameter):

$$\tilde{x} = \begin{bmatrix} 1 \\ x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \tag{3}$$

Our job is to find the free parameters (weights) $\vec{w}$, and the Perceptron algorithm does just this.

3 The Algorithm

Inputs:
Training set S with m instances, where the ith instance $= (\tilde{x}, y)_i$
learning rate η
Initialize:
All weight terms $w_j = \texttt{rand}(-0.05, 0.05)$, where $j = 1...n$
while not done **do**
  for $j = 0$ to n **do**
    $\triangle w_j = 0$
  end for
  for each $(\tilde{x}, y)_i$ **do**
    $f(\tilde{x}) = 1/1 + \exp(-\vec{w}^T \tilde{x})$
    $\delta_i = (y - f(\tilde{x}))$
    for $j = 0$ to n **do**
      $\triangle w_j = \triangle w_j + \eta \delta_i \tilde{x}_j$
    end for
  end for
  for $j = 0$ to n **do**
    $w_j = w_j + \triangle w_j$
  end for
end while
return final model $\vec{w}$

4 Convergence

To do so, we need an objective function that tells us when we have found a "good" $\vec{w}$, and our objective function will be sum-squared error (SSE):

$$\epsilon = \sum_y (y - f(x))^2 \tag{4}$$

5 Logistic Regression

Recall that Logistic Regression computes a model based on the natural log of the odds:

$$\ln(\frac{p}{1-p}) = w_0 + w_1 x \tag{5}$$

Solving for p gives

$$\ln(\frac{p}{1-p}) \quad = \quad w_0 + w_1 x$$

$$
\begin{aligned}
\frac{p}{1-p} &= e^{w_0+w_1x} \\
p &= e^{w_0+w_1x} - e^{w_0+w_1x}p \\
p + e^{w_0+w_1x}p &= e^{w_0+w_1x} \\
p(1+e^{w_0+w_1x}) &= e^{w_0+w_1x} \\
p &= \frac{e^{w_0+w_1x}}{1+e^{w_0+w_1x}} \\
p &= \frac{1}{1+e^{-(w_0+w_1x)}}
\end{aligned} \tag{6}
$$

For the case of $n = 1$ independent variable, $\vec{w}^T\tilde{x} = w_0 + w_1x$, making Eqs. 1 and 6 equivalent.