AI Notes

David Press

01/10/2005

1 Perceptrons

A perceptron is an artificial neuron. It has k inputs with weights $w_1 ... w_k$. When receiving inputs $x_1 ... x_k$ the output is $\sigma\left(\sum_{i=1}^{k} w_i x_i\right)$. For our purposes the output is yes if $w \bullet x \geq 0$ and no otherwise. The goal is to find a set of weights such that the perceptron gives the correct output, at least on the test data.

2 Perceptron Learning Algorithm

Input: $(x_1, y_1)...(x_n, y_n)$

```
for i = 1 to n
  w_i = 0
end
do
  flag := false
  for i = 1 to n
    if (y_i = -1 AND w_i • x_i ≥ 0)OR
       (y_i = 1 AND w_i • x_i < 0)
    then
      w := w + y_i x_i
      flag := true
    end
  end
while flag = true
```

This algorithm continues to change the weights until no weights need to be changed.

3 Convergence of the PLA

Is the PLA guaranteed to stop?

Definition: (x_i, y_i) are linearly separable if and only if $(\exists w_*)\,(\forall i \in \{1, ..., n\})\,(y_i(w \bullet x_i) \geq 0)$

Theorem: If (x_i, y_i) are linearly separable, PLA converges.

aw_* for a > 0 is also a solution, so WLOG, $\|w_*\| = 1$

Proof:

Let

$$m = \min_i |w_* \bullet x_i| > 0$$

and

$$R = \max_i \|x_i\|$$

Also, let w_k be w after the k-th mistake. We will prove that k has an upper bound in terms of m and R, both of which are finite constants in terms of the data.

By definition,

$$w_k = w_{k-1} + yx$$

where y and x are the data point where the (k-1)-th mistake happened. This leads to:

$$w_* \bullet w_k = w_* \bullet w_{k-1} + y(w_* \bullet x)$$

By definition, y must be the same sign as $w_* \bullet x$. This implies that the term $y(w_* \bullet x)$ must be positive and also $\geq m$ (by definition of m)

So,

$$w_* \bullet w_k \geq w_* \bullet w_{k-1} + m$$

By induction:

$$w_* \bullet w_k \geq w_* \bullet 0 + km = km$$

By Cauchy-Schwartz:

$$\|w_*\| \, \|w_k\| \geq w_* \bullet w_k \geq km$$

Since $\|w_*\| = 1$ we have

$$\|w_k\| \geq w_* \bullet w_k \geq km$$

We can deduce the following:

$$\|w_k\|^2 = w_k \bullet w_k = (w_{k-1} + yx) \bullet (w_{k-1} + yx) = \|w_{k-1}\|^2 + y^2 x \bullet x + 2y\,(w_{k-1} \bullet x)$$

The term $2y\,(w_{k-1} \bullet x)$must be negative, because w_{k-1} was a mistaken w.

By definition of R:

$$\|w_k\|^2 \leq \|w_{k-1}\|^2 + x \bullet x \leq \|w_{k-1}\|^2 + R^2$$

By induction:

$$\|w_k\|^2 \leq kR^2$$

So,

$$\sqrt{k}R \geq \|w_k\|$$

Combining the above with our previous result, we get:

$$\sqrt{k}R \geq \|w_k\| \geq w_* \bullet w_k \geq km$$

Thus,

$$\sqrt{k}R \geq km \Leftrightarrow R \geq \sqrt{k}m \Leftrightarrow k \leq \frac{R^2}{m^2}$$

QED

4 Observations and Remarks

1. There are infinitely many classifiers for linearly separable data. The PLA gives just one classifier. A better classifier would be the maximum margin classifier:
$$m = \max_{w_* s.t. \|w_*\|=1} \min_i |w_* \bullet x_i|$$
2. The PLA only finds classifiers for linearly separable data. It would fail on the following data set: $(-1,-1)\,(1,1)(-1,1)(1,-1)$. This is the XOR problem. Also, there are many other linear classifiers (least squares, linear programming)
3. If you look deeper into the kind of data that works, you would see that it has to have a Hilbert Space Structure. That structure guarantees convergence.
4. How well does this algorithm generalize? It is perfect on the training data, but how does it do on novel data?
5. Number 4 leads to the question of "What is the optimal classifier?" We shall see that this is the Bayes Classifier.