# CS57800: Statistical Machine Learning

## Homework 2

Ruoyu Wu
wu1377@purdue.edu

October 12, 2018

# 1 Vanilla Perceptron / Basic Perceptron

## 1.1 Vanilla Percepron Algorithm

**Algorithm 1** VanillaPerceptron

**procedure** Training(D, R, EPOCH)
  $w_d \leftarrow 0$, for all d = 1...D
  $b \leftarrow 0$
  **for** `iter = 1...EPOCH` **do**
    **for** `(x,y) in D` **do**
      $a \leftarrow \Sigma_{d=1}^{D} w_d x_d + b$
      **if** `ya <= 0` **then**
        $w_d \leftarrow w_d + Ryx_d$, for all d = 1...D
        $b \leftarrow b + Ry$
      **end if**
    **end for**
  **end for**
  **return** $w_0, w_1, ..., w_D, b$
**end procedure**

## 1.2 Implementation

See vanilla_perceptron.py.

### 1.3 Size of training set vs. F1

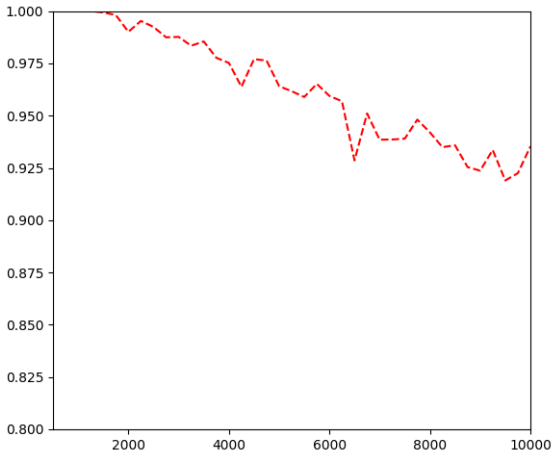


Figure 1: Training size vs F1 on training set

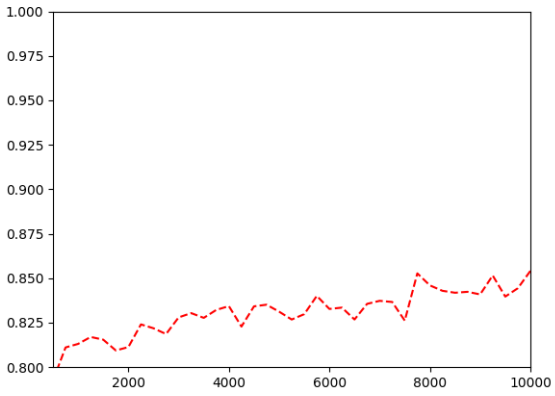


Figure 2: Training size vs F1 on test set

### 1.4 Number of epoch vs. F1

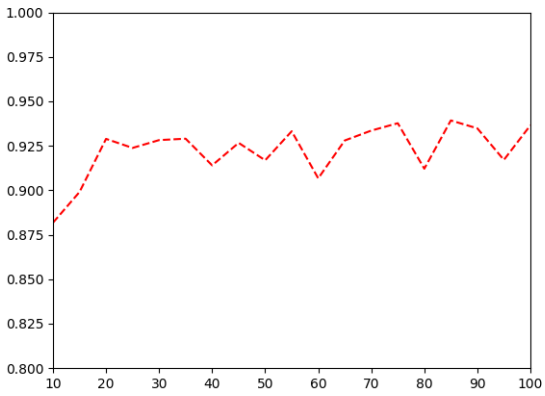


Figure 3: Number of epoch vs F1 on training set

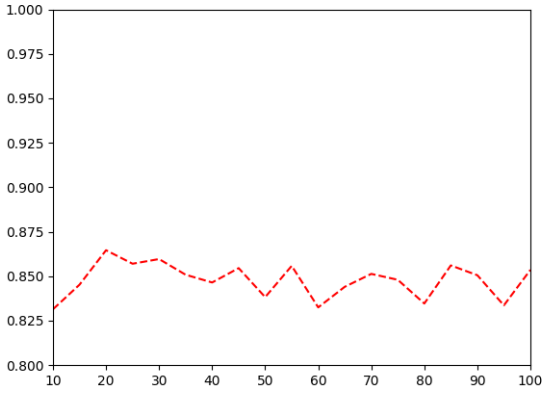


Figure 4: Number of epoch vs F1 on test set

### 1.5 Learning rate vs. F1

The x-axis means the exponent of 0.1. For example, 3 stands for $0.1^3$.

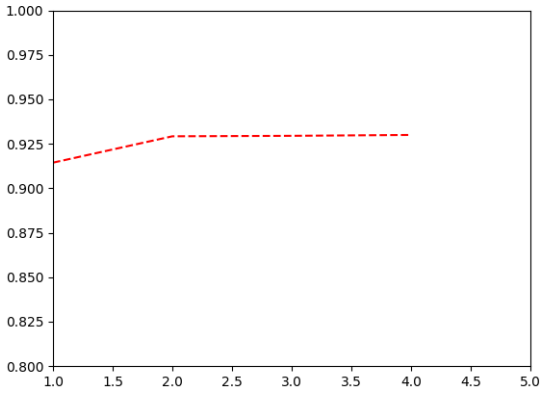


Figure 5: Learning rate vs F1 on training set

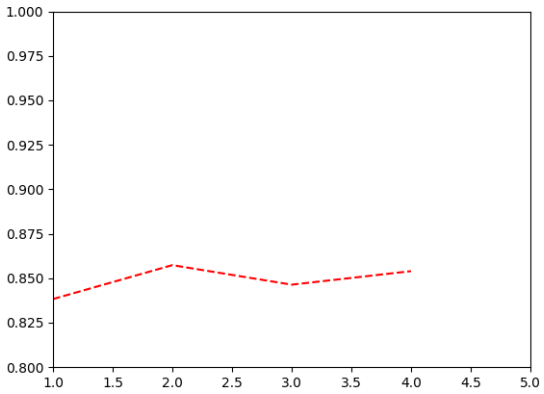


Figure 6: Learning rate vs F1 on test set

## 1.6 Observation from learning curves

### 1.6.1 Effect of the size of training set

As the size of training set grows, F1 scores on training set decreases while F1 scores on test set increases, which is aligned with our expectation. When the size of training set grows, it is harder for the model to find a good hyperplane. But since the model is fed with more data, the hyperplane(our hypothesis) can better approximate the 'real function' with higher possibility. Performance on training set is always better than test set.

### 1.6.2 Effect of number of epoch

The performance on train set and test set both gains when increasing the epoch from 10 to 20, but it does not help afterwards, especially on the test set. Setting epoch too big might lead to over-fitting. Performance on training set is always better than test set.

### 1.6.3 Effect of learning rate

The performance on train set and test set both gains when increasing the learning rate from 0.1 to 0.01, but it seems to have no effect afterwards because of small step size. Performance on training set is always better than test set.

# 2 Average Perceptron

## 2.1 Average Percepron Algorithm

**Algorithm 2** AveragePerceptron

**procedure** TRAINING(D, R, EPOCH)
  $w_d \leftarrow 0$, for all d = 1...D. $b \leftarrow 0$
  $u_d \leftarrow 0$, for all d = 1...D. $\beta \leftarrow 0$
  $c \leftarrow 1$
  **for** `iter = 1...EPOCH` **do**
    **for** `(x,y) in D` **do**
      **if** $y(wx + b) <= 0$ **then**
        $w \leftarrow w + Ryx$
        $b \leftarrow b + Ry$
        $u \leftarrow u + Rycx$
        $\beta \leftarrow \beta + Ryc$
      **end if**
      $c \leftarrow c + 1$
    **end for**
  **end for**
  **return** $w - \frac{1}{c}u, b - \frac{1}{c}\beta$
**end procedure**

## 2.2 Implementation

See average_perceptron.py.

### 2.3 Size of training set vs. F1

The green line indicates the performance on training set and red line on test set.

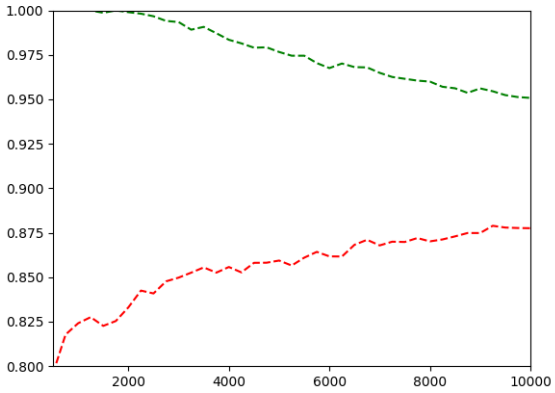


Figure 7: Training size vs F1 on training set and test set

### 2.4 Number of epoch vs. F1

The green line indicates the performance on training set and red line on test set.

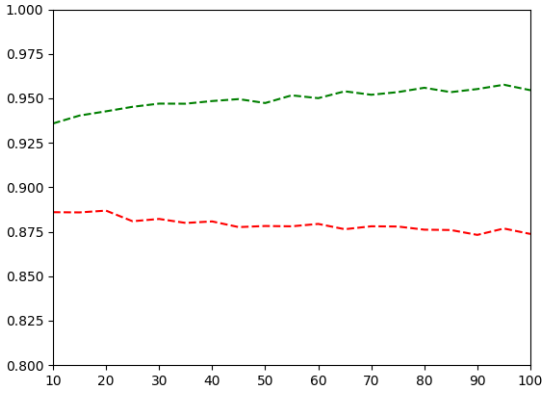


Figure 8: Number of epoch vs F1 on training set and test set

### 2.5 Learning rate vs. F1

The x-axis means the exponent of 0.1. For example, 3 stands for $0.1^3$. The green line indicates the performance on training set and red line on test set.

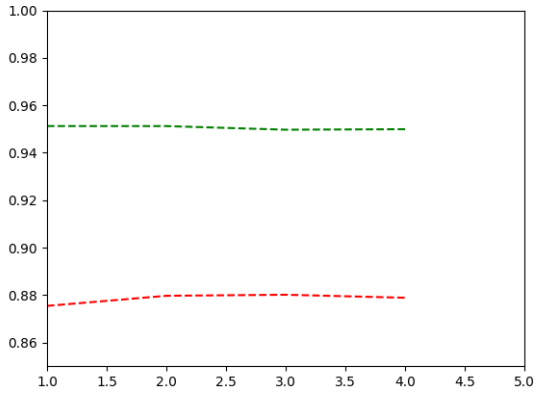


Figure 9: Learning rate vs F1 on training set

## 2.6 Observation from learning curves

### 2.6.1 Effect of the size of training set

As the size of training set grows, the F1 score on training set decreases, while F1 score on testing set increases. Performance on training set is always better than test set.

### 2.6.2 Effect of number of epoch

The performance gets better on training set when epoch increases, while the performance degrades on test set because of over-fitting. Performance on training set is always better than test set.

### 2.6.3 Effect of learning rate

The learning rate does not have many effects on the performance on training set. It does not have many effects on test set neither, but you can see from 1.0 to 2.0 the performance sightly go up and go down slightly after 2.0, which means that $learning\_rate = 0.01$ achieves best performance. Performance on training set is always better than test set.

# 3 Winnow

## 3.1 Winnow Algorithm

**Algorithm 3** Winnow Algorithm

**procedure** TRAINING(D, FACTOR, EPOCH, THETA)
  $w_i \leftarrow 1$, for all i = 1...D
  **for** `iter = 1...EPOCH` **do**
    **for** `(x,y) in D` **do**
      $a \leftarrow \Sigma_{d=1}^{D} w_d x_d$
      **if** $a < THETA \ \& \ y = 1$ **then**
        $w_i \leftarrow Factor * w_i$, for all $x_i = 1$
      **end if**
      **if** $a >= THETA \ \& \ y = -1$ **then**
        $w_i \leftarrow \frac{1}{Factor} * w_i$, for all $x_i = 1$
      **end if**
    **end for**
  **end for**
  **return** $w_0, w_1, ..., w_D$
**end procedure**

## 3.2 Implementation

See willow.py. Please note that I set the value of THETA to 1 in all experiments for simplicity(I do not use THETA as a hyper parameter). willow.py executes in the same way as others, i.e. use command "python willow.py [size of training size] [no of epochs] [factor] [DATA_FOLDER]".

## 3.3 Hyper-parameters

I use factor and the size of training set as the hyper-parameters.

### 3.3.1 Size of training set vs. F1

The green line indicates the performance on training set and red line on test set.

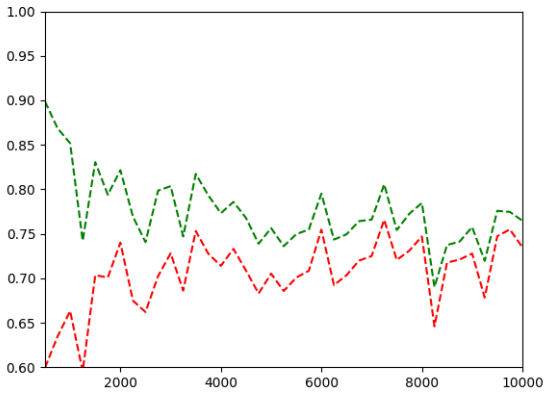


Figure 10: Training size vs F1 on training set and test set

### 3.3.2 Factor vs. F1

The green line indicates the performance on training set and red line on test set.

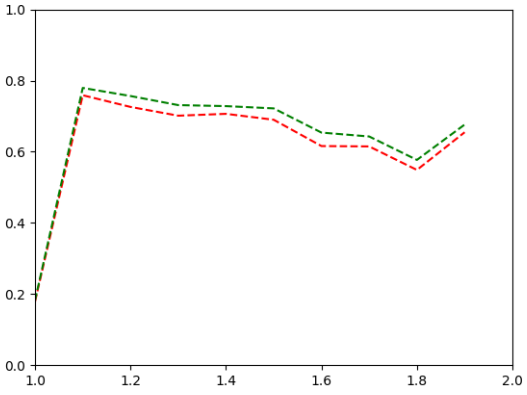


Figure 11: Factor vs F1 on training set

### 3.4 Observation from learning curves

#### 3.4.1 Effect of the size of training set

As the size of training set grows, the F1 score on training set decreases, while F1 score on testing set increases. When the training set is big, the performance on test set is close to test set.

#### 3.4.2 Effect of learning rate

The trend of performance on training set and test set seem to be synchronous. We notice that when $factor = 1, 0$, the weights will not change, so the performance suffers. And the bigger the factor, weights of willow algorithm will change faster(more drastic). We observe that Willow achieves best performance when factor is around 1.1.

## 4 Open Ended Questions

### 4.1

Average perceptron always outperform the vanilla perceptron, and vanilla always outperform the Willow algorithm. And compared to vanilla perceptron, average perceptron is less likely to overfit.

### 4.2

Mistake bound is the maximum number of mistake an online learning model can make in the long run.

### 4.3

**Algorithm 4** LearnBooleanConjunction(V[1...m]. V represents m training examples(n-dim)

1: $result \leftarrow x_1 \wedge \neg x_1 \wedge x_2 \wedge \neg x_2 \wedge ... \wedge x_n \wedge \neg x_n$
2: **for** $i \leftarrow 1tom$ **do**
3: **if** $V[i]islabeledTrue$ **then** remove from $result$ any term of $\neg x_j$, where $V[i]_j = True$ remove from $result$ any term of $x_j$, where $V[i]_j = False$
4: **end if**
5: **end for**
6: return $result$

At first the $result$ has $2^n$ terms, for each mistake, this algorithm at least removes one term. Therefore, the mistake bound for this algorithm is $2^n$.

## 4.4

### 4.4.1 Will both classifiers converge

According to the Perceptron Convergence Theorem, if the data is linearly separable, the perceptron learning algorithm will converge. In this case, data is linearly separable, so both classifiers will converge, no matter how the data is ordered.

### 4.4.2 What will be the training error of each one of the classifiers

If you train these two classifiers with many epochs(enough to converge), then both of the classifiers can achieve 0 error afterwards. Therefore, the training error for both classifiers are 0. The second classifier will converge slower than the first classifier. If this question is asking 'what is the mistake bound of each one of the classifier', then the mistake bound of these are the same. It it because for each epoch, the first perceptron really only have learned from a handful of examples.