Slides adapted from Prof. Carpuat

CMSC 422 Introduction to Machine Learning

Lecture 7 The Perceptron

Furong Huang / furongh@cs.umd.edu

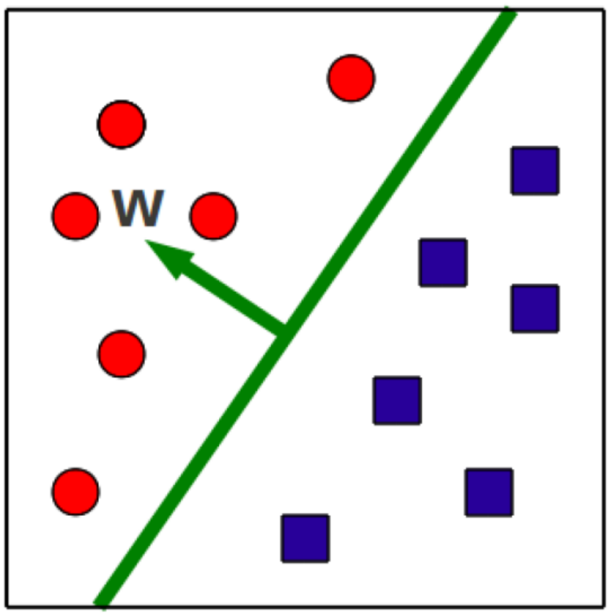
UNIVERSITY OF MARYLAND

This week

- The perception: a new model/algorithm
  - its variants: voted, averaged
  - **convergence proof**
- Fundamental Machine Learning Concepts
  - Online vs. batch learning
  - Error-driven learning
  - **Linear separability and margin of a dataset**

- Project 1 published today

Recap: Perceptron for binary classification



- Classifier = hyperplane that separates positive from negative examples

$$\hat{y} = sign(w^T x + b)$$

- Perceptron training
  - Finds such a hyperplane
  - Online & error-driven

Learning

- Find algorithm that gives us w and b for a given data set $\mathrm{D}\ (\boldsymbol{x}, y)$
  - Many algorithms possible

- Once we know w and b we can predict the class of a new data point $\boldsymbol{x}_i$ by evaluating

$$\hat{y} = sign(w^\top \boldsymbol{x}_i + b)$$

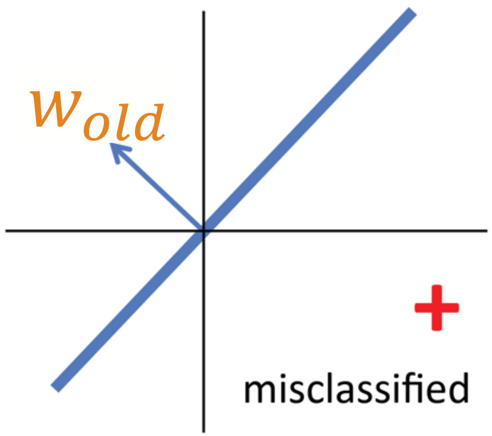
- We learned a particular way of finding these parameters–via the perceptron update rule
  - Iterative online algorithm–visits all the data over epochs

Perceptron update: geometric interpretation

A training example $(\boldsymbol{x}, y)$ is misclassified, i.e.,

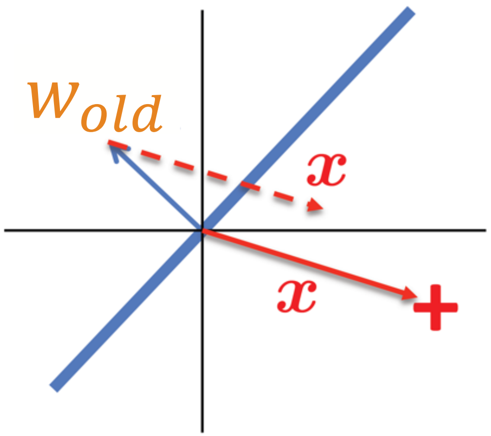
$$sign\left(\boldsymbol{w}_{old}^{\top}\boldsymbol{x} + b\right) \neq y$$

Let's say $y = +1$



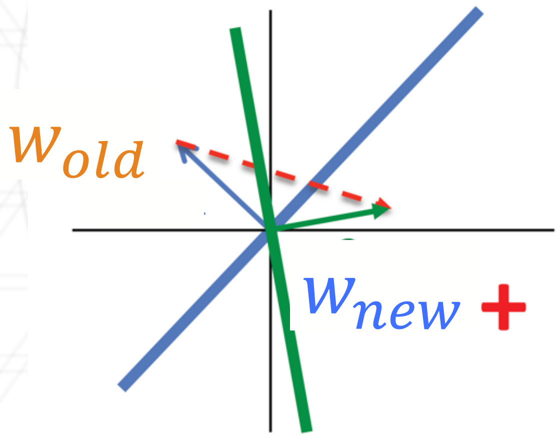
Perceptron update: geometric interpretation

Update: $\boldsymbol{w}_{new} \leftarrow \boldsymbol{w}_{old} + y\boldsymbol{x}$, i.e., $\boldsymbol{w}_{new} \leftarrow \boldsymbol{w}_{old} + \boldsymbol{x}$



Perceptron update: geometric interpretation

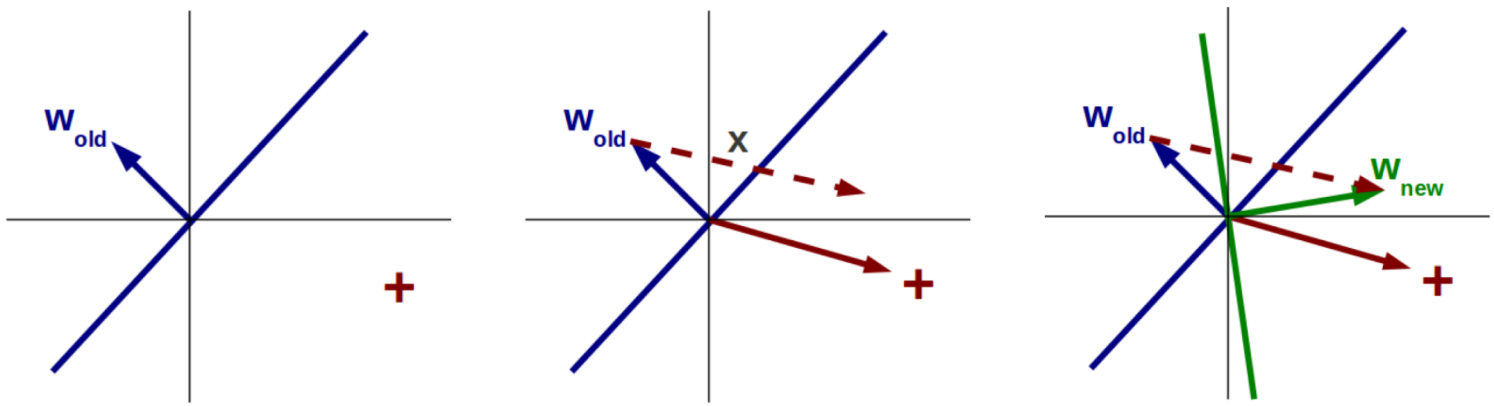
Update: $\boldsymbol{w}_{new} \leftarrow \boldsymbol{w}_{old} + y\boldsymbol{x}$, i.e., $\boldsymbol{w}_{new} \leftarrow \boldsymbol{w}_{old} + \boldsymbol{x}$



Recap: Perceptron updates

Update for a misclassified positive example: $y = 1$

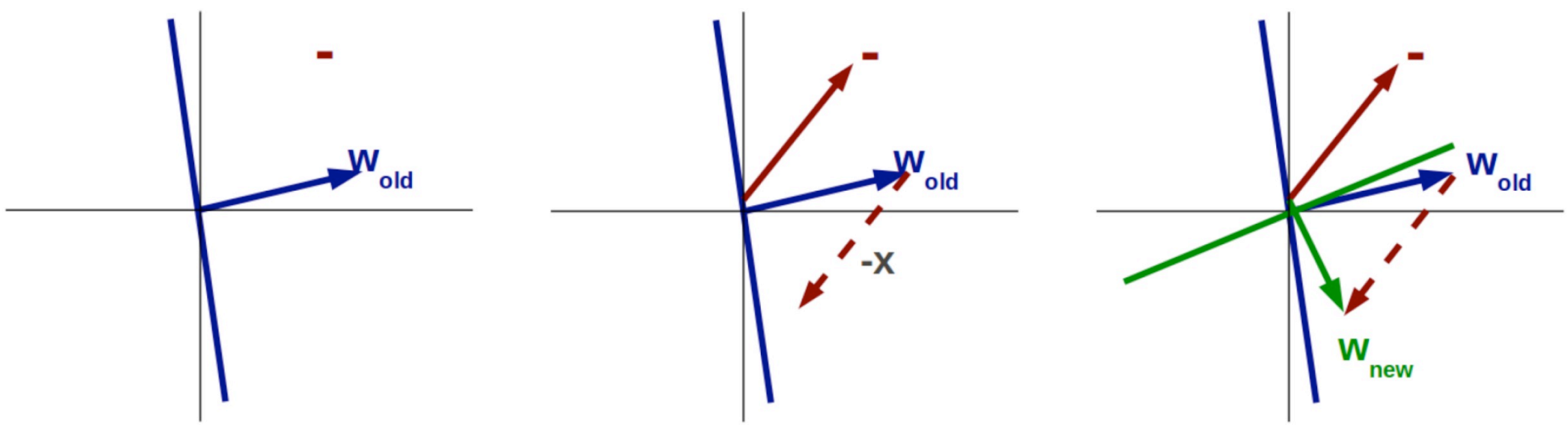
$$\mathbf{w}_{new} = \mathbf{w}_{old} + \mathbf{x}$$



Recap: Perceptron updates

Update for a misclassified negative example: $y = -1$

$$\mathbf{w}_{new} = \mathbf{w}_{old} - \mathbf{x}$$



Today

- Example of perceptron + averaged perceptron training
- Perceptron convergence proof
- Fundamental Machine Learning Concepts
  - Linear separability and margin of a dataset

Standard Perceptron: predict based on final parameters

Algorithm 5 PerceptronTrain($\mathbf{D}$, *MaxIter*)

$w_d \leftarrow 0$, for all $d = 1 \ldots D$ // initialize weights
$b \leftarrow 0$ // initialize bias
for $iter = 1 \ldots MaxIter$ **do**
  for all $(x,y) \in \mathbf{D}$ **do**
    $a \leftarrow \sum_{d=1}^{D} w_d \, x_d + b$ // compute activation for this example
    if $ya \leq 0$ **then**
      $w_d \leftarrow w_d + yx_d$, for all $d = 1 \ldots D$ // update weights
      $b \leftarrow b + y$ // update bias
    end if
  end for
end for
return $w_0, w_1, \ldots, w_D, b$

Predict based on final + intermediate parameters

- The voted perceptron

$$\hat{y} = \text{sign}\left(\sum_{k=1}^{K} c^{(k)} \text{sign}\left(\boldsymbol{w}^{(k)} \cdot \hat{\boldsymbol{x}} + b^{(k)}\right)\right)$$

- The averaged perceptron

$$\hat{y} = \text{sign}\left(\sum_{k=1}^{K} c^{(k)} \left(\boldsymbol{w}^{(k)} \cdot \hat{\boldsymbol{x}} + b^{(k)}\right)\right)$$

- Require keeping track of "survival time" of weight vectors $c^{(1)}, \ldots, c^{(K)}$

Averaged perceptron decision rule

$$\hat{y} = \text{sign}\left(\sum_{k=1}^{K} c^{(k)} \left(\boldsymbol{w}^{(k)} \cdot \hat{\boldsymbol{x}} + b^{(k)}\right)\right)$$

can be rewritten as

$$\hat{y} = \text{sign}\left(\left(\sum_{k=1}^{K} c^{(k)} \boldsymbol{w}^{(k)}\right) \cdot \hat{\boldsymbol{x}} + \sum_{k=1}^{K} c^{(k)} b^{(k)}\right)$$

Averaged Perceptron: predict based on average of intermediate parameters

Algorithm 7 AveragedPerceptronTrain($\mathbf{D}$, $MaxIter$)

$\boldsymbol{w} \leftarrow \langle 0, 0, \ldots 0 \rangle$, $b \leftarrow 0$ // initialize weights and bias
$\boldsymbol{u} \leftarrow \langle 0, 0, \ldots 0 \rangle$, $\beta \leftarrow 0$ // initialize cached weights and bias
$c \leftarrow 1$ // initialize example counter to one
for $iter = 1 \ldots MaxIter$ **do**
  for all $(\boldsymbol{x}, y) \in \mathbf{D}$ **do**
    if $y(\boldsymbol{w} \cdot \boldsymbol{x} + b) \leq 0$ **then**
      $\boldsymbol{w} \leftarrow \boldsymbol{w} + y\, \boldsymbol{x}$ // update weights
      $b \leftarrow b + y$ // update bias
      $\boldsymbol{u} \leftarrow \boldsymbol{u} + y\, c\, \boldsymbol{x}$ // update cached weights
      $\beta \leftarrow \beta + y\, c$ // update cached bias
    end if
    $c \leftarrow c + 1$ // increment counter regardless of update
  end for
end for
return $\boldsymbol{w} - \frac{1}{c}\, \boldsymbol{u}$, $b - \frac{1}{c}\, \beta$ // return averaged weights and bias

Convergence of Perceptron

- The perceptron has converged if it can classify every training example correctly
  - i.e. if it has found a hyperplane that correctly separates positive and negative examples
- Under which conditions does the perceptron converge and how long does it take?

Convergence of Perceptron

Theorem (Block & Novikoff, 1962)

If the training data $D = \{(x_1, y_1), \ldots, (x_N, y_N)\}$ is **linearly separable** with margin γ by a unit norm hyperplane w_* ($||w_*|| = 1$) with $b = 0$,

Then **perceptron training converges after $\frac{R^2}{\gamma^2}$ errors** during training

(assuming ($||x|| < R$) for all x).

Margin of a data set *D*

$$margin(\mathbf{D}, w, b) = \begin{cases} \min_{(x,y)\in\mathbf{D}} y(w \cdot x + b) & \text{if } w \text{ separates } \mathbf{D} \\ -\infty & \text{otherwise} \end{cases} \quad (4.8)$$

Distance between the hyperplane (w,b) and the nearest point in **D**

$$margin(\mathbf{D}) = \sup_{w,b} margin(\mathbf{D}, w, b) \quad (4.9)$$

Largest attainable margin on **D**

Theorem (Block & Novikoff, 1962)

If the training data $D = \{(x_1, y_1), \dots, (x_N, y_N)\}$ is **linearly separable** with margin γ by a unit norm hyperplane w_* ($||w_*|| = 1$) with $b = 0$, then **perceptron training converges after $\frac{R^2}{\gamma^2}$ errors** during training (assuming ($||x|| < R$) for all x).

Proof:

- Margin of $\mathbf{w}_*$ on any *arbitrary example* $(\mathbf{x}_n, y_n)$: $\frac{y_n \mathbf{w}_*^T \mathbf{x}_n}{||\mathbf{w}_*||} = y_n \mathbf{w}_*^T \mathbf{x}_n \geq \gamma$
- Consider the $(k+1)^{th}$mistake: $y_n \mathbf{w}_k^T \mathbf{x}_n \leq 0$, and update $\mathbf{w}_{k+1} = \mathbf{w}_k + y_n \mathbf{x}_n$
- $\mathbf{w}_{k+1}^T \mathbf{w}_* = \mathbf{w}_k^T \mathbf{w}_* + y_n \mathbf{w}_*^T \mathbf{x}_n \geq \mathbf{w}_k^T \mathbf{w}_* + \gamma$ (why is this nice?)
- Repeating iteratively k times, we get $\mathbf{w}_{k+1}^T \mathbf{w}_* > k\gamma$ (1)
- $||\mathbf{w}_{k+1}||^2 = ||\mathbf{w}_k||^2 + 2y_n \mathbf{w}_k^T \mathbf{x}_n + ||\mathbf{x}||^2 \leq ||\mathbf{w}_k||^2 + R^2$ (since $y_n \mathbf{w}_k^T \mathbf{x}_n \leq 0$)
- Repeating iteratively k times, we get $||\mathbf{w}_{k+1}||^2 \leq kR^2$ (2)

Theorem (Block & Novikoff, 1962)

If the training data $D = \{(x_1, y_1), \ldots, (x_N, y_N)\}$ is **linearly separable** with margin γ by a unit norm hyperplane w_* ($||w_*||= 1$) with $b = 0$, then **perceptron training converges after $\frac{R^2}{\gamma^2}$ errors** during training (assuming ($||x||< R$) for all x).

What does this mean?

- Perceptron converges quickly when margin is large, slowly when it is small
- Bound does not depend on number of training examples N, nor on number of features d
- **Proof guarantees that perceptron converges, but not necessarily to the max margin separator**

What you should know

- Perceptron concepts
  - training/prediction algorithms (standard, voting, averaged)
  - convergence theorem and what practical guarantees it gives us
  - how to draw/describe the decision boundary of a perceptron classifier
- Fundamental ML concepts
  - Determine whether a data set is linearly separable and define its margin
  - Error driven algorithms, online vs. batch algorithms

UNIVERSITY OF
MARYLAND

Furong Huang
3251 A.V. Williams, College Park, MD 20740
301.405.8010 / furongh@cs.umd.edu