Department of Computer Science
UNIVERSITY OF COLORADO **BOULDER**

Why Language is Hard: Structure and Predictions

Introduction to Data Science Algorithms

Jordan Boyd-Graber and Michael Paul

SLIDES ADAPTED FROM LIANG HUANG

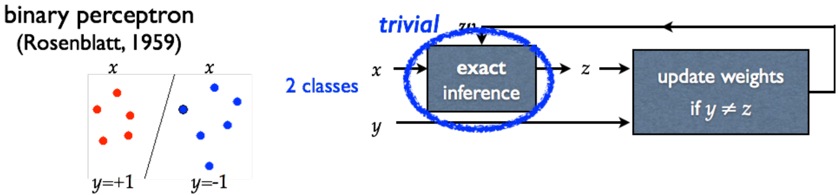
Perceptron Algorithm

- Can we find parameters to minimize errors?
- Rather than just counting up how often we see events?
- Very similar to logistic regression (but 0/1 loss)

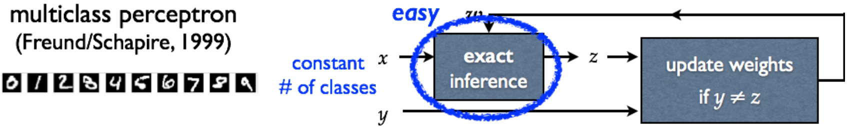
k-means

$\vec{w}_1 \leftarrow \vec{0}$
for $t \leftarrow 1 \dots T$ **do**
  Receive x_t
  $\hat{y}_t \leftarrow \text{sgn}(\vec{w}_t \cdot \vec{x}_t)$
  Receive y_t
  if $\hat{y}_t \neq y_t$ **then**
    $\vec{w}_{t+1} \leftarrow \vec{w}_t + y_t \vec{x}_t$
  else
    $\vec{w}_{t+1} \leftarrow w_t$
return w_{T+1}

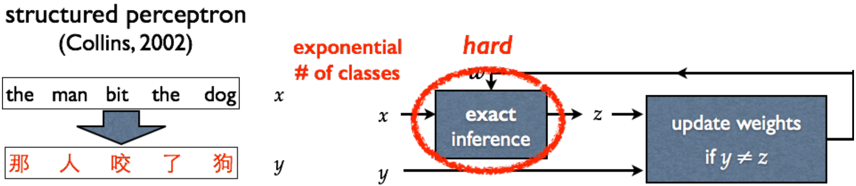
Binary to Structure



Binary to Structure



Binary to Structure



Generic Perceptron

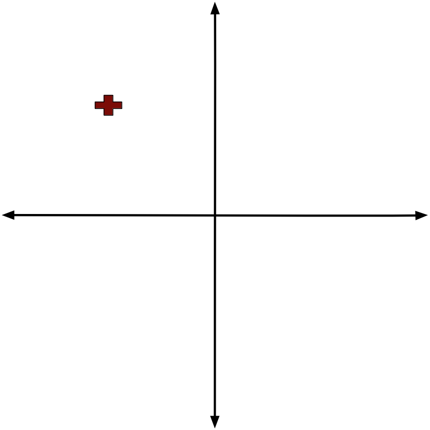
- perceptron is the simplest machine learning algorithm
- online-learning: one example at a time
- learning by doing
  - find the best output under the current weights
  - update weights at mistakes

2D Example

Initially, weight vector is zero:

$$\vec{w}_1 = \langle 0, 0 \rangle \tag{1}$$

Observation 1



$$x_1 = \langle -2, 2 \rangle \tag{2}$$

$$\hat{y}_1 = 0 \tag{3}$$

$$y_1 = +1 \tag{4}$$

Update 1

$$\vec{w}_{t+1} \leftarrow \vec{w}_t + y_t \vec{x}_t \quad (5)$$

$$\vec{w}_2 \leftarrow \quad (6)$$

Update 1

$$\vec{w}_{t+1} \leftarrow \vec{w}_t + y_t \vec{x}_t \tag{5}$$

$$\vec{w}_2 \leftarrow \langle 0, 0 \rangle + \langle -2, 2 \rangle \tag{6}$$

$$\tag{7}$$

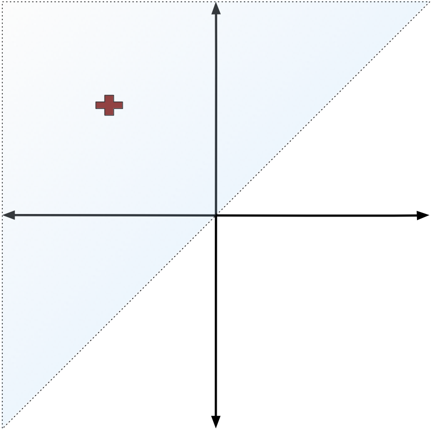
Update 1

$$\vec{w}_{t+1} \leftarrow \vec{w}_t + y_t \vec{x}_t \tag{5}$$

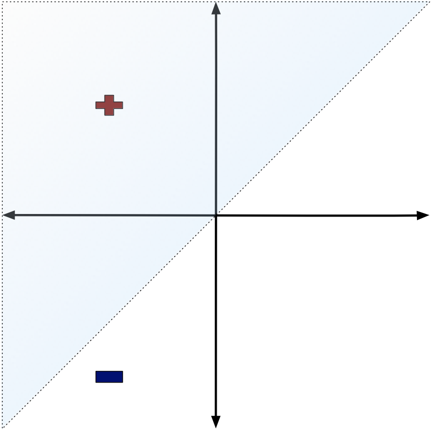
$$\vec{w}_2 \leftarrow \langle 0, 0 \rangle + \langle -2, 2 \rangle \tag{6}$$

$$\vec{w}_2 = \langle -2, 2 \rangle \tag{7}$$

Observation 2



Observation 2



$$x_2 = \langle -2, -3 \rangle \tag{8}$$

$$\hat{y}_2 = +4 + -6 = -2 \tag{9}$$

$$y_2 = -1 \tag{10}$$

Update 2

$$\vec{w}_{t+1} \leftarrow \vec{w}_t \quad (11)$$

$$\vec{w}_2 \leftarrow \quad (12)$$

Update 2

$$\vec{w}_{t+1} \leftarrow \vec{w}_t \tag{11}$$

$$\vec{w}_2 \leftarrow \langle -2, 2 \rangle \tag{12}$$

$$\tag{13}$$

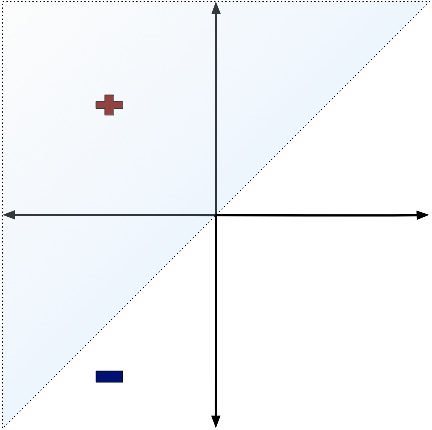
Update 2

$$\vec{w}_{t+1} \leftarrow \vec{w}_t \quad (11)$$

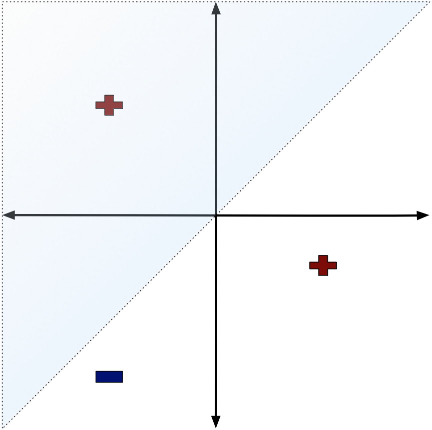
$$\vec{w}_2 \leftarrow \langle -2, 2 \rangle \quad (12)$$

$$\vec{w}_2 = \langle -2, 2 \rangle \quad (13)$$

Observation 3



Observation 3



$$x_3 = \langle 2, -1 \rangle \tag{14}$$

$$\hat{y}_3 = -4 + -2 = -6 \tag{15}$$

$$y_3 = +1 \tag{16}$$

Update 3

$$\vec{w}_{t+1} \leftarrow \vec{w}_t + y_t \vec{x}_t \quad (17)$$

$$\vec{w}_3 \leftarrow \quad (18)$$

Update 3

$$\vec{w}_{t+1} \leftarrow \vec{w}_t + y_t \vec{x}_t \tag{17}$$

$$\vec{w}_3 \leftarrow \langle -2, 2 \rangle + \langle 2, -1 \rangle \tag{18}$$

$$\tag{19}$$

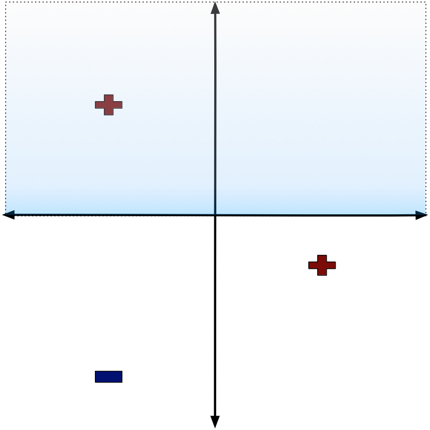
Update 3

$$\vec{w}_{t+1} \leftarrow \vec{w}_t + y_t \vec{x}_t \tag{17}$$

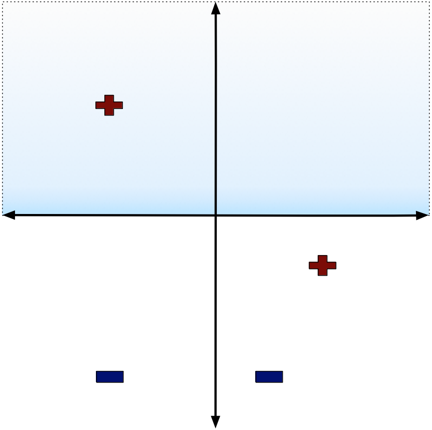
$$\vec{w}_3 \leftarrow \langle -2, 2 \rangle + \langle 2, -1 \rangle \tag{18}$$

$$\vec{w}_3 = \langle 0, 1 \rangle \tag{19}$$

Observation 4



Observation 4



$$x_4 = \langle 1, -4 \rangle \tag{20}$$

$$\hat{y}_4 = -4 \tag{21}$$

$$y_4 = -1 \tag{22}$$

Update 4

$$\vec{w}_4 \leftarrow \tag{23}$$

Update 4

$$\vec{w}_4 \longleftarrow \vec{w}_3 \tag{23}$$

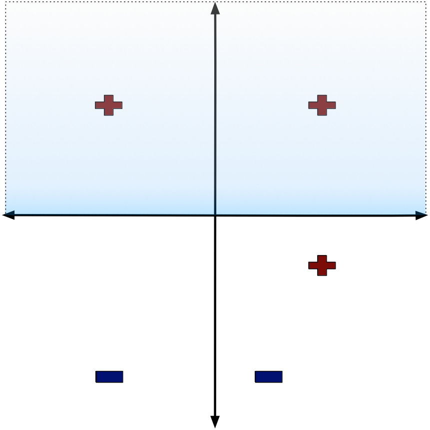
$$\tag{24}$$

Update 4

$$\vec{w}_4 \leftarrow \vec{w}_3 \tag{23}$$

$$\vec{w}_4 = \langle 0, 1 \rangle \tag{24}$$

Observation 5



$$x_5 = \langle 2, 2 \rangle \tag{25}$$

$$\hat{y}_5 = 2 \tag{26}$$

$$y_5 = +1 \tag{27}$$

Update 5

$$\vec{w}_5 \longleftarrow \tag{28}$$

Update 5

$$\vec{w}_5 \longleftarrow \vec{w}_4 \tag{28}$$

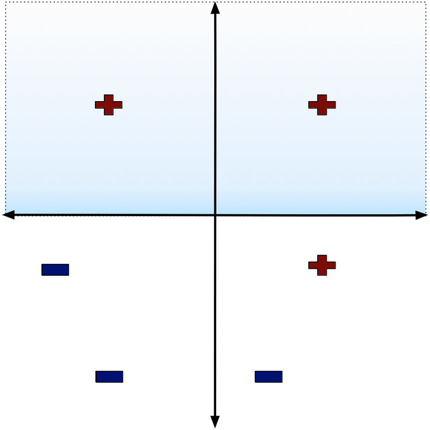
$$\tag{29}$$

Update 5

$$\vec{w}_5 \leftarrow \vec{w}_4 \tag{28}$$

$$\vec{w}_5 = \langle 0, 1 \rangle \tag{29}$$

Observation 6



$$x_6 = \langle 2, 2 \rangle \tag{30}$$

$$\hat{y}_6 = 2 \tag{31}$$

$$y_6 = +1 \tag{32}$$

Update 6

$$\vec{w}_6 \longleftarrow \tag{33}$$

Update 6

$$\vec{w}_6 \longleftarrow \vec{w}_5 \tag{33}$$

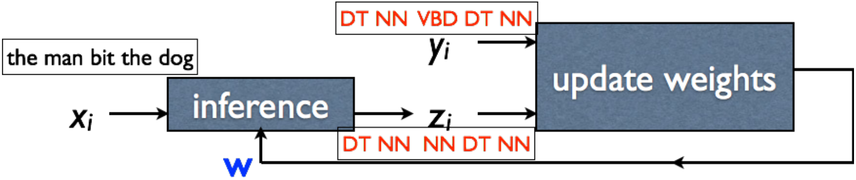
$$\tag{34}$$

Update 6

$$\vec{w}_6 \leftarrow \vec{w}_5 \tag{33}$$

$$\vec{w}_6 = \langle 0, 1 \rangle \tag{34}$$

Structured Perceptron



Perceptron Algorithm

Inputs: Training set (x_i, y_i) for $i = 1 \ldots n$

Initialization: $\mathbf{W} = 0$

Define: $F(x) = \text{argmax}_{y \in \mathbf{GEN}(x)} \mathbf{\Phi}(x, y) \cdot \mathbf{W}$

Algorithm: For $t = 1 \ldots T, i = 1 \ldots n$

$z_i = F(x_i)$

If $(z_i \neq y_i) \quad \mathbf{W} \longleftarrow \mathbf{W} + \mathbf{\Phi}(x_i, y_i) - \mathbf{\Phi}(x_i, z_i)$

Output: Parameters $\mathbf{W}$

POS Example

- gold-standard: DT NN VBD DT NN y
  - the man bit the dog x

$\Phi(x, y)$

- current output: DT NN NN DT NN z
  - the man bit the dog x

$\Phi(x, z)$

- assume only two feature classes
  - tag bigrams t_{i-1} t_i
  - word/tag pairs w_i
- weights ++: (NN,VBD) (VBD, DT) (VBD → bit)
- weights −−: (NN, NN) (NN, DT) (NN → bit)

What must be true?

- Finding highest scoring structure must be really fast (you'll do it often)
- Requires some sort of dynamic programming algorithm
- For tagging: features must be local to y (but can be global to x)