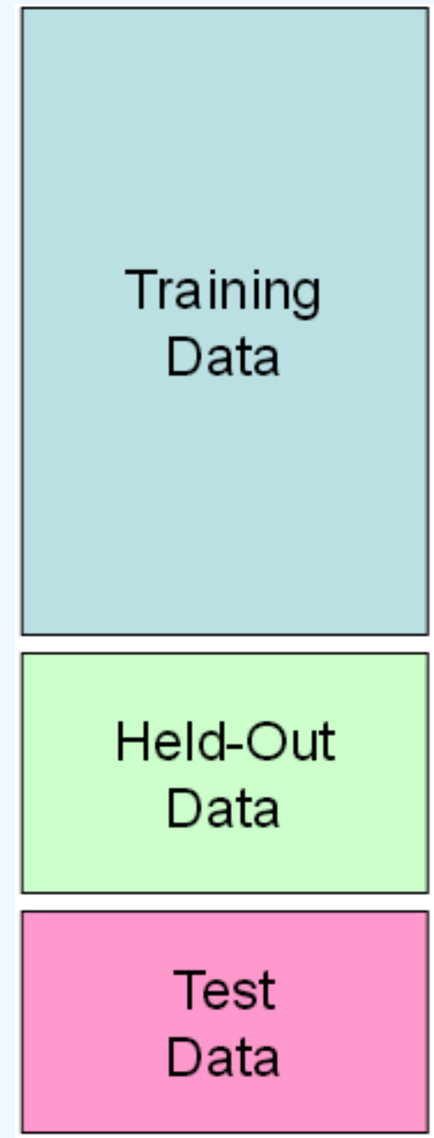
Classification

- Classification
  - Given inputs x, predict labels (classes) y
- Examples
  - Spam detection. input: documents; classes: spam/ham
  - OCR. input: images; classes: characters
  - Medical diagnosis. input: symptoms; classes: diseases
  - Autograder. input: codes; output: grades

Important Concepts

- Data: labeled instances, e.g. emails marked spam/ham
  - Training set
  - Held out set (we will give examples today)
  - Test set
- Features: attribute-value pairs that characterize each x
- Experimentation cycle
  - Learn parameters (e.g. model probabilities) on training set
  - (Tune hyperparameters on held-out set)
  - Compute accuracy of test set
  - Very important: never “peek” at the test set!
- Evaluation
  - Accuracy: fraction of instances predicted correctly
- Overfitting and generalization
  - Want a classifier which does well on test data
  - Overfitting: fitting the training data very closely, but not generalizing well



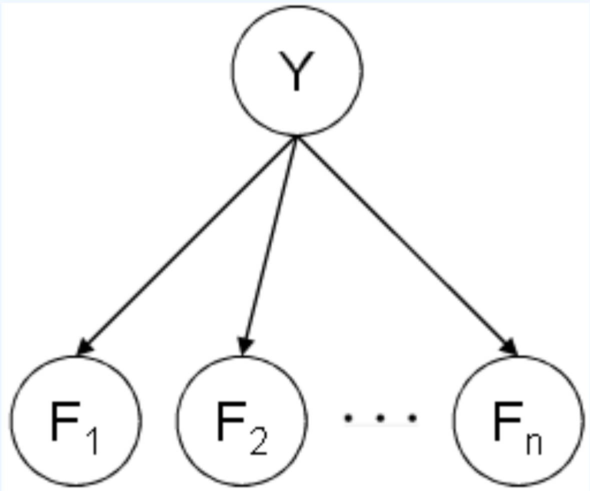
General Naive Bayes

- A general *naive Bayes* model:

$|Y| \times |F|^n$ parameters

$$p(Y, F_1 \cdots F_n) = p(Y) \prod_i p(F_i \mid Y)$$

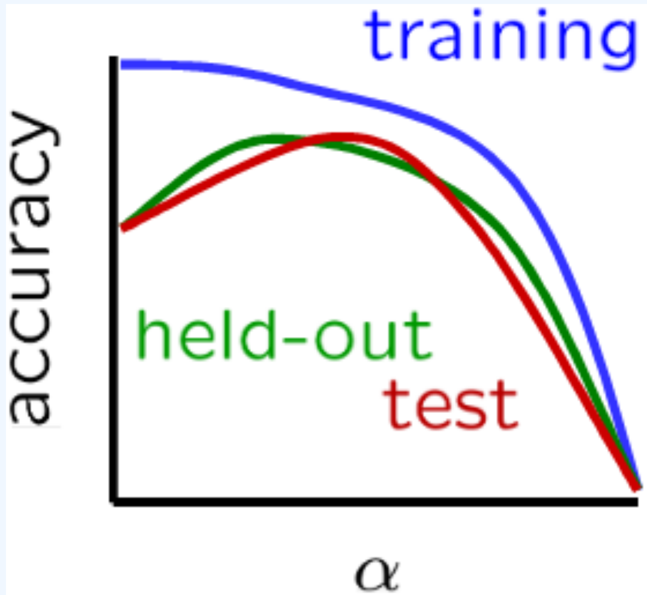
$|Y|$ parameters $\qquad n \times |Y| \times |F|$ parameters



- We only specify how each feature depends on the class
- Total number of parameters is *linear* in n

Tuning on Held-Out Data

- Now we've got two kinds of unknowns
  - Parameters: the probabilities p(Y|X), p(Y)
  - Hyperparameters, like the amount of smoothing to do: k,α
- Where to learn?
  - Learn parameters from training data
  - For each value of the hyperparameters, train and test on the held-out data
  - Choose the best value and do a final test on the test data



Today's schedule

- Binary Linear Classifiers
- Perceptron
- Multi-class Linear Classifiers
- Multi-class Perceptron

Classification: Feature Vectors

x | $f(x)$ | y

```
Hello,

Do you want free printr
cartriges?  Why pay more
when you can get them
ABSOLUTELY FREE!  Just
```

```
# free        : 2
YOUR_NAME     : 0
MISSPELLED    : 2
FROM_FRIEND   : 0
...
```

SPAM
or
+

```
PIXEL-7,12    : 1
PIXEL-7,13    : 0
...
NUM_LOOPS     : 1
...
```

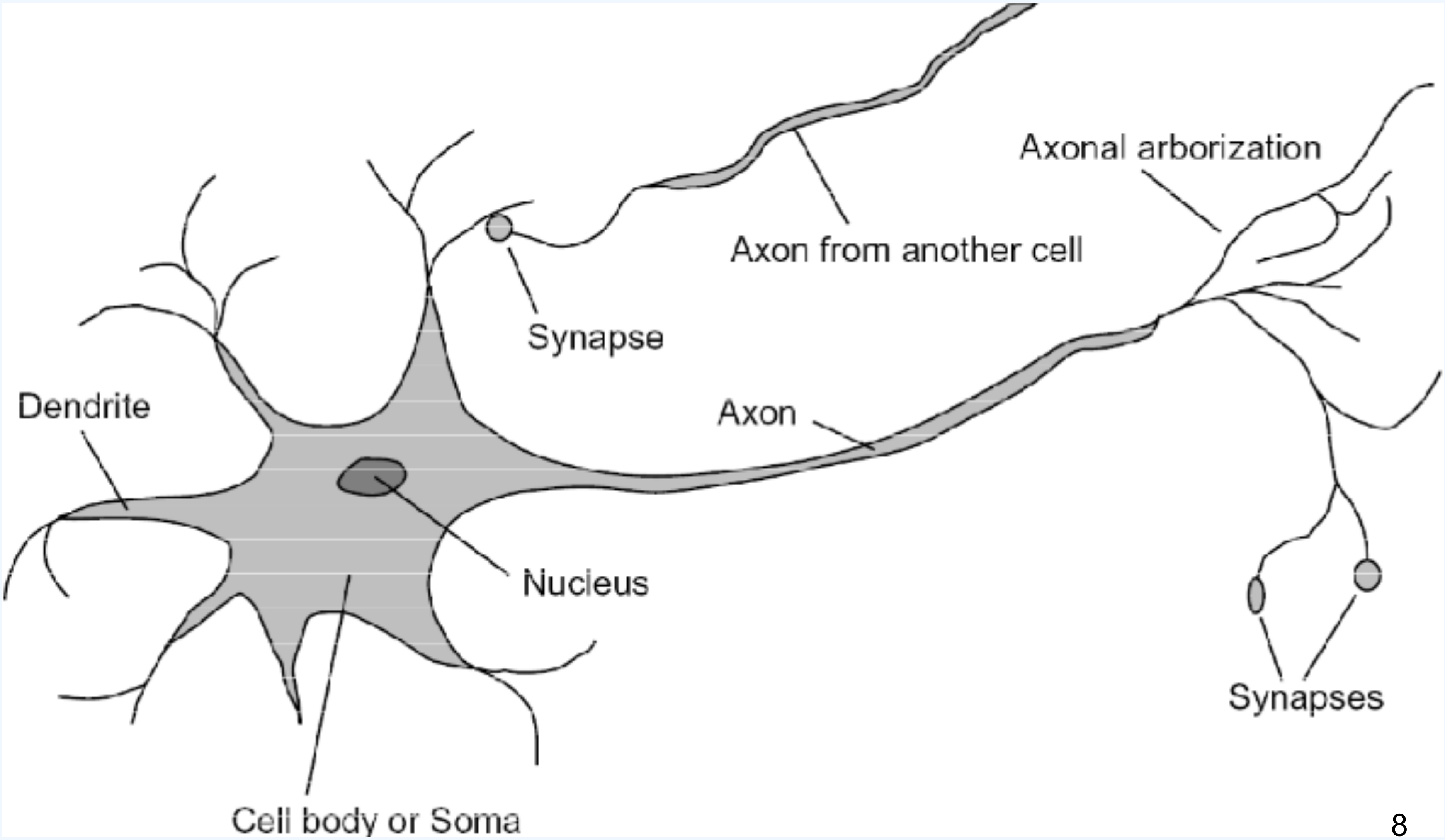
“2”

Outline

- **Binary Linear Classifiers**
- Perceptron
- Multi-class Linear Classifiers
- Multi-class Perceptron

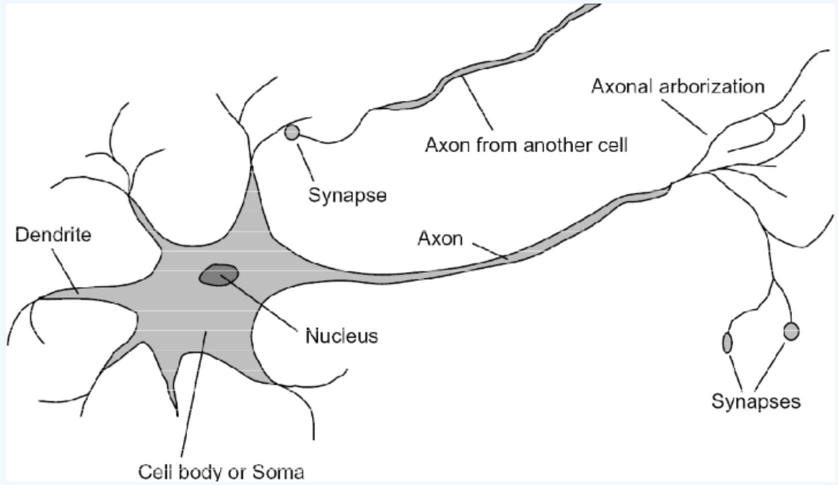
Some (Simplified) Biology

Very loose inspiration: human neurons



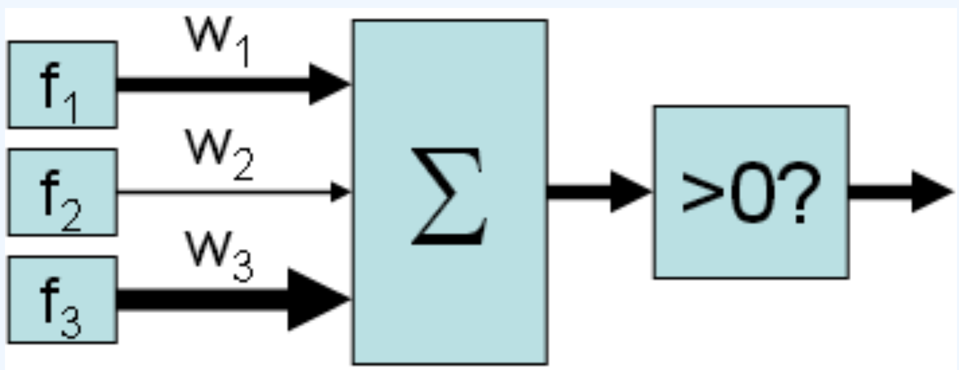
Linear Classifiers

- Inputs are feature values
- Each feature has a weight
- Sum is the activation



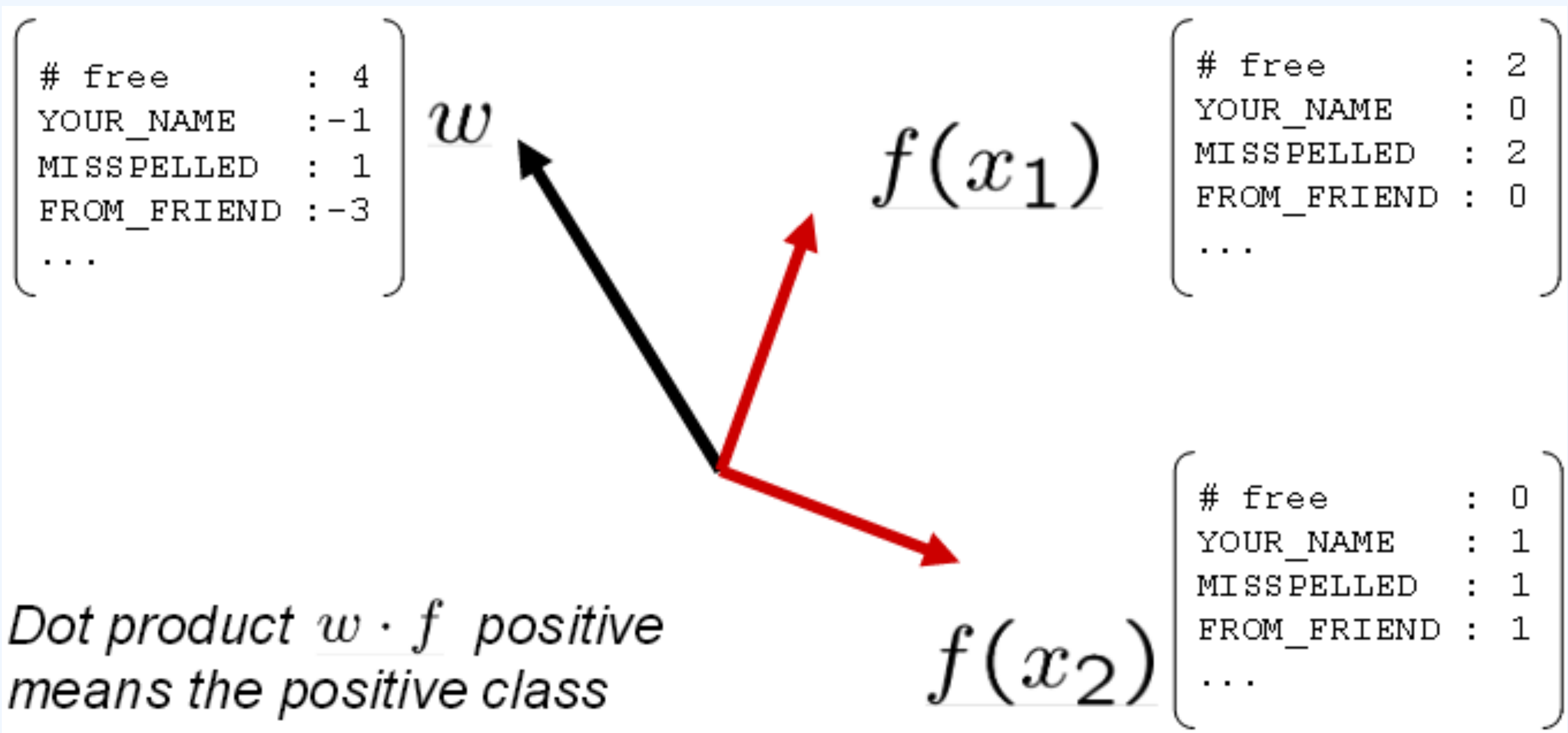
$$\text{activation}_w(x) = \sum_i w_i \cdot f_i(x) = w \cdot f(x)$$

- If the activation is:
  - Positive: output +1
  - Negative, output -1



Classification: Weights

- Binary case: compare features to a weight vector
- Learning: figure out the weight vector from examples



Linear classifiers Mini Exercise

$$f(x_1) = \begin{bmatrix} \text{\# free} & : & 2 \\ \text{YOUR\_NAME} & : & 0 \end{bmatrix} \quad f(x_2) = \begin{bmatrix} \text{\# free} & : & 4 \\ \text{YOUR\_NAME} & : & 1 \end{bmatrix}$$

$$f(x_3) = \begin{bmatrix} \text{\# free} & : & 1 \\ \text{YOUR\_NAME} & : & 1 \end{bmatrix} \quad w = \begin{bmatrix} -1 \\ 2 \end{bmatrix}$$

- 1.Draw the 3 feature vectors and the weight vector w
- 2.Which feature vectors are classified as +? As -?
- 3.Draw the line separating feature vectors being classified + and -.

Linear classifiers Mini Exercise 2---Bias Term

$$f(x_1) = \begin{bmatrix} \text{Bias} & : & 1 \\ \#\ \text{free} & : & 2 \\ \text{YOUR\_NAME} & : & 0 \end{bmatrix} \quad f(x_2) = \begin{bmatrix} \text{Bias} & : & 1 \\ \#\ \text{free} & : & 4 \\ \text{YOUR\_NAME} & : & 1 \end{bmatrix}$$

$$f(x_3) = \begin{bmatrix} \text{Bias} & : & 1 \\ \#\ \text{free} & : & 1 \\ \text{YOUR\_NAME} & : & 1 \end{bmatrix} \quad w = \begin{bmatrix} -3 \\ -1 \\ 2 \end{bmatrix}$$

- 1.Draw the 4 feature vectors and the weight vector w
- 2.Which feature vectors are classified as +? As -?
- 3.Draw the line separating feature vectors being classified + and -.

Linear classifiers Mini Exercise 3--- adding features

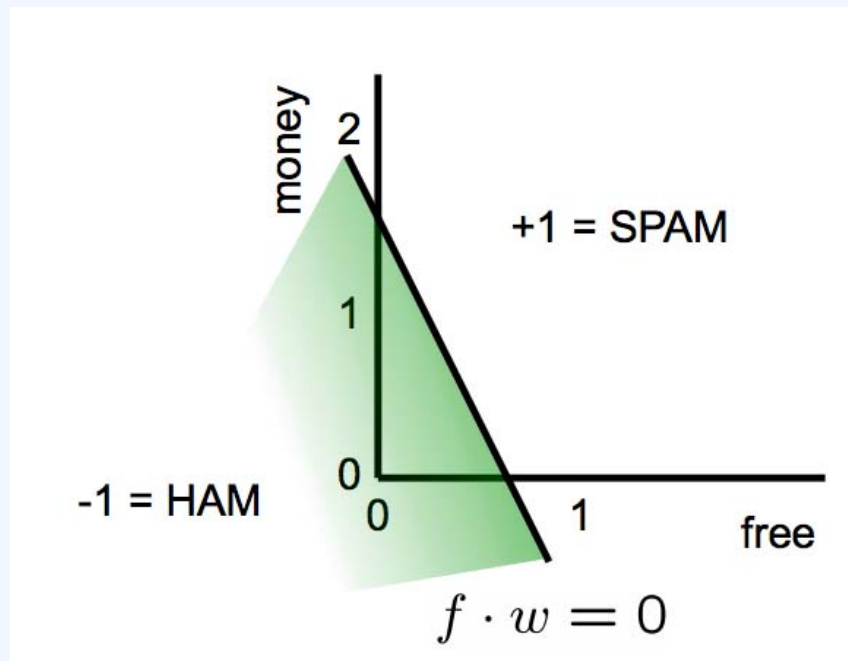
- 1.Draw the 4 1-dimensional feature vectors
  - $f(x_1)=-2$, $f(x_2)=-1$, $f(x_3)=1$, $f(x_4)=2$
- 2.Is there a separating vector w that
  - classifies x_1 and x_4 as +1 and
  - classifies x_2 and x_3 as -1?
- 3. can you add one more feature to make this happen?

Binary Decision Rule

- In the space of feature vectors
  - Examples are points
  - Any weight vector is a hyperplane
  - One side corresponds to Y = +1
  - Other corresponds to Y = -1

w

```
BIAS  : -3
free  :  4
money :  2
...
```



Outline

- Generative vs. Discriminative
- Binary Linear Classifiers
- **Perceptron: how to find the weight vector w from data.**
- Multi-class Linear Classifiers
- Multi-class Perceptron

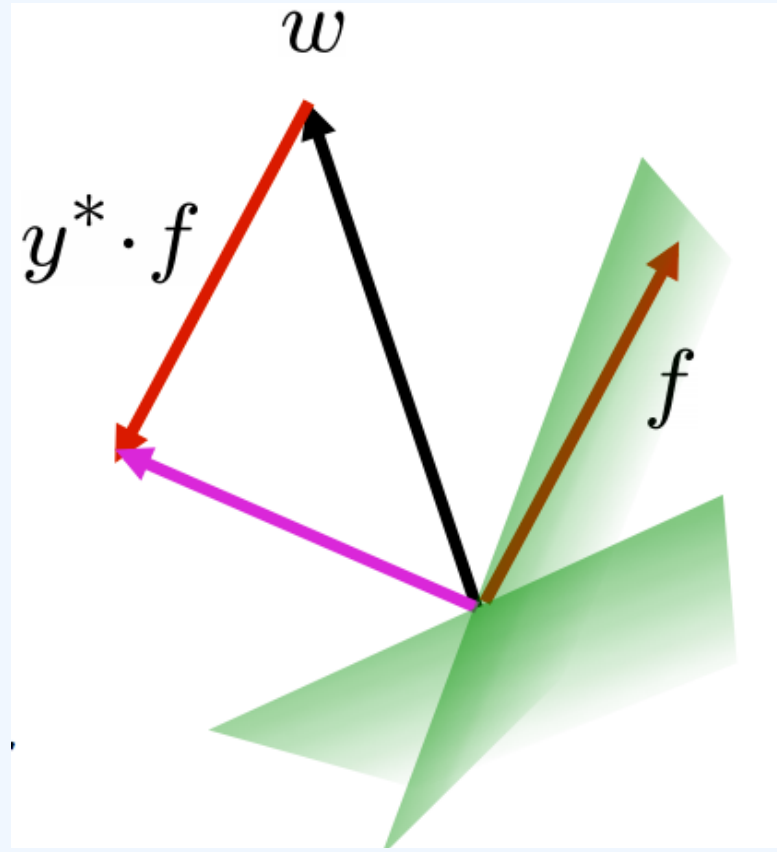
Learning: Binary Perceptron

- Start with weights = 0
- For each training instance:
  - Classify with current weights

$$y = \begin{cases} +1 & \text{if } w \bullet f(x) \geq 0 \\ -1 & \text{if } w \bullet f(x) < 0 \end{cases}$$

  - If correct (i.e. y=y*), no change!
  - If wrong: adjust the weight vector by adding or subtracting the feature vector. Subtract if y* is -1.

$$w = w + y^* \bullet f$$



Outline

- Generative vs. Discriminative
- Binary Linear Classifiers
- Perceptron
- **Multi-class Linear Classifiers**
- Multi-class Perceptron

Multiclass Decision Rule

- If we have multiple classes:
  - A weight vector for each class:

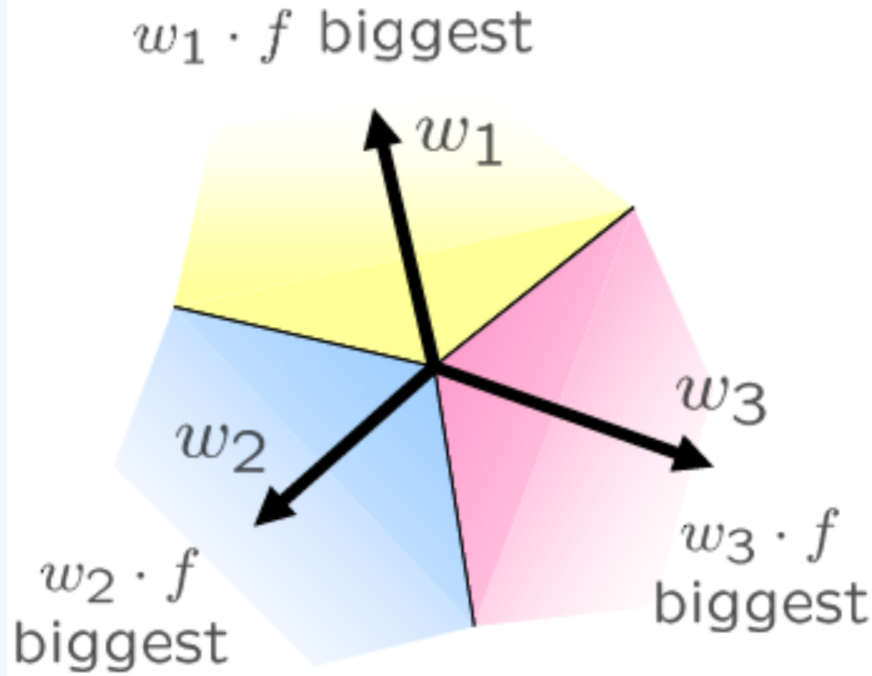
$$w_y$$

  - Score (activation) of a class y:

$$w_y \cdot f(x)$$

  - Prediction highest score wins

$$y = \arg\max_y w_y \cdot f(x)$$



Binary = multiclass where the negative class has weight zero

Example Exercise --- Which Category is Chosen?

"win the vote" →

BIAS	:	1
win	:	1
game	:	0
vote	:	1
the	:	1
...		

w_{SPORTS}

BIAS	:	-2
win	:	4
game	:	4
vote	:	0
the	:	0
...		

w_{POLITICS}

BIAS	:	1
win	:	2
game	:	0
vote	:	4
the	:	0
...		

w_{TECH}

BIAS	:	2
win	:	0
game	:	2
vote	:	0
the	:	0
...		

Exercise: Multiclass linear classifier for 2 classes and binary linear classifier

- Consider the multiclass linear classifier for two classes with $w_1 = \begin{bmatrix} -1 \\ 2 \end{bmatrix}$ $w_2 = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$
- Is there an equivalent binary linear classifier, i.e., one that classifies all points x = (x_1, x_2) the same way?

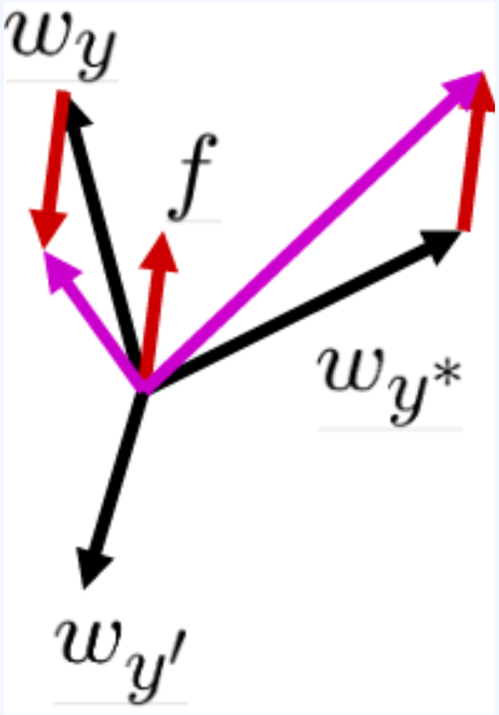
Outline

- Generative vs. Discriminative
- Binary Linear Classifiers
- Perceptron
- Multi-class Linear Classifiers
- **Multi-class Perceptron: learning the weight vectors w_i from data**

Learning: Multiclass Perceptron

- Start with all weights = 0
- Pick up training examples one by one
- Predict with current weights

$$y = \arg\max_y w_y \cdot f(x)$$
$$= \arg\max_y \sum_i w_{y,i} \cdot f_i(x)$$



- If correct, no change!
- If wrong: lower score of wrong answer, raise score of right answer

$$w_y = w_y - f(x)$$
$$w_{y^*} = w_{y^*} + f(x)$$

Example: Multiclass Perceptron

"win the vote"

"win the game"

w_{SPORTS}

```
BIAS  : 0
win   : 0
game  : 0
vote  : 0
the   : 0
...
```

w_{POLITICS}

```
BIAS  : 0
win   : 0
game  : 0
vote  : 0
the   : 0
...
```

w_{TECH}

```
BIAS  : 0
win   : 0
game  : 0
vote  : 0
the   : 0
...
```