

Machine Learning

NYU Shanghai
Spring 2019

Logistics

- **Class webpage:**
 - <https://sites.google.com/a/nyu.edu/nyu-shanghai-introduction-to-machine-learning-spring-2019/>
- **Class:**
 - Tuesday 9:45-11:00
 - Wednesday 8:15-9:45
 - Thursday 9:45-11:00
- **My office hours:**
 - Thursdays 4-5
 - Room 1415

Prerequisites

- Calculus (differential and integral)
- Intro to Computer Programming (Python)
- Desirable:
 - Probability and Statistics
 - Linear Algebra
 - Multivariable Calculus
 - Data structures

Evaluation

- About 7 homeworks (25%)
 - Both theory and programming
 - See collaboration policy on class webpage
- Two exams (40%)
- Final Project (25%): no final exam
- Course participation (10%)

Problem sets

- First assignment out tomorrow! Due Feb 20.
- See problem set policy on course website
 - First try to solve the problems on your own
 - Then, can briefly discuss with other classmates
 - Write-up solutions, code on your own
 - Hand in code as well, all in one document.

Final Project

- Teams of two; no teams of three or one
- Be creative – think of new problems that you can tackle using machine learning
 - Scope: ~40 hours/person
 - Do not be overly ambitious: do something you can complete
- Logistics:
 - Project proposal due April 23.
- Final project format: video presentation

Reference Materials

No textbook required. Readings will come from freely available online material.

The course will draw from:

- David Sontag's Machine Learning and Computational Statistics course (NYU)
- Andrew Ng's Coursera Machine Learning course (Stanford)
- The Deep Learning Book
- Andrew Ng Deep Learning Coursera Course
- David Silver Lectures on Reinforcement Learning

Mathematics

- Machine Learning is all about algorithms implementing heuristics.
- The heuristics are inspired from mathematics.
- This course will be heavy in mathematics, heavy in mathematical notation.
- You should love math and love Python programming if you want to take this course.

Machine Learning: Overview

NYU Shanghai
Spring 2019

Some slides adapted from David Sontag, who adapted the slides from Luke Zettlemoyer, Vibhav Gogate, Pedro Domingos, and Carlos Guestrin

Some Popular Types of Machine Learning

- Supervised Machine Learning
 - Often used in practice
- Unsupervised Machine Learning
- Neural networks
 - Just one of many methodologies for supervised machine learning
- Deep Learning
 - Fancy term for Neural networks ++
- Reinforcement Learning

Machine Learning Examples

- Later we'll survey some examples.
- We'll now instead go through one example of supervised ML in some detail.
- Prepare you for the first problem set.
- The Example:
 - Identify spam emails
 - Classify using perceptron algorithm

Spam filtering

data

prediction

Osman Khan to Carlos

[show details](#) Jan 7 (6 days ago)

[Reply](#)

sounds good
+ok

Carlos Guestrin wrote:

Let's try to chat on Friday a little to coordinate and more on Sunday in person?

Carlos

Welcome to New Media Installation: Art that Learns

Carlos Guestrin to 10615-announce, Osman, Michel [show details](#) 3:15 PM (8 hours ago)

[Reply](#)

Hi everyone,

Welcome to New Media Installation:Art that Learns

The class will start tomorrow.

Make sure you attend the first class, even if you are on the Wait List.

The classes are held in Doherty Hall C316, and will be Tue, Thu 01:30-4:20 PM.

By now, you should be subscribed to our course mailing list: 10615-announce@cs.cmu.edu.

You can contact the instructors by emailing: 10615-instructors@cs.cmu.edu

Natural_LoseWeight SuperFood Endorsed by Oprah Winfrey, Free Trial 1 bottle,
pay only \$5.95 for shipping mfw rlk [Spam](#) | [X](#)

Jaquelyn Halley to nherrlein, bcc: thehorney, bcc: ang [show details](#) 9:52 PM (1 hour ago)

[Reply](#)

=== Natural WeightLOSS Solution ===

Vital Acai is a natural WeightLOSS product that Enables people to lose wieght and cleansing their bodies faster than most other products on the market.

Here are some of the benefits of Vital Acai that You might not be aware of. These benefits have helped people who have been using Vital Acai daily to Achieve goals and reach new heights in there dieting that they never thought they could.

- * Rapid WeightLOSS
- * Increased metabolism - BurnFat & calories easily!
- * Better Mood and Attitude
- * More Self Confidence
- * Cleanse and Detoxify Your Body
- * Much More Energy
- * BetterSexLife
- * A Natural Colon Cleanse

Spam
vs.
Not Spam

Binary classification

- **Input:** email
- **Output:** spam / not spam
- **Setup:**
 - Get a large collection of example emails, each labeled “spam” or “not spam”
 - Note: someone has to hand label all this data!
 - Want to learn to predict class of new, future emails
- **Features:** The attributes used to make the spam / no spam decision
 - Words: “FREE!,” etc.
 - Text Patterns: \$dd, CAPS
 - Metadata: SenderInContacts
 - ...



Dear Sir.

First, I must solicit your confidence in this transaction, this is by virtue of its nature as being utterly confidential and top secret. ...



TO BE REMOVED FROM FUTURE MAILINGS, SIMPLY REPLY TO THIS MESSAGE AND PUT "REMOVE" IN THE SUBJECT.

99 MILLION EMAIL ADDRESSES
FOR ONLY \$99



Ok, I know this is blatantly OT but I'm beginning to go insane. Had an old Dell Dimension XPS sitting in the corner and decided to put it to use, I know it was working pre being stuck in the corner, but when I plugged it in, hit the power nothing happened.

HW Assignment 1 (a)

- Data set: spam_train.txt
 - 5000 emails
 - Each labeled as spam or non-spam
 - Preprocess emails (see assignment)
 - Split the data set:
 - train.txt: first 400 emails
 - validate.txt: last 1000 emails

Homework Assignment 1 (b)

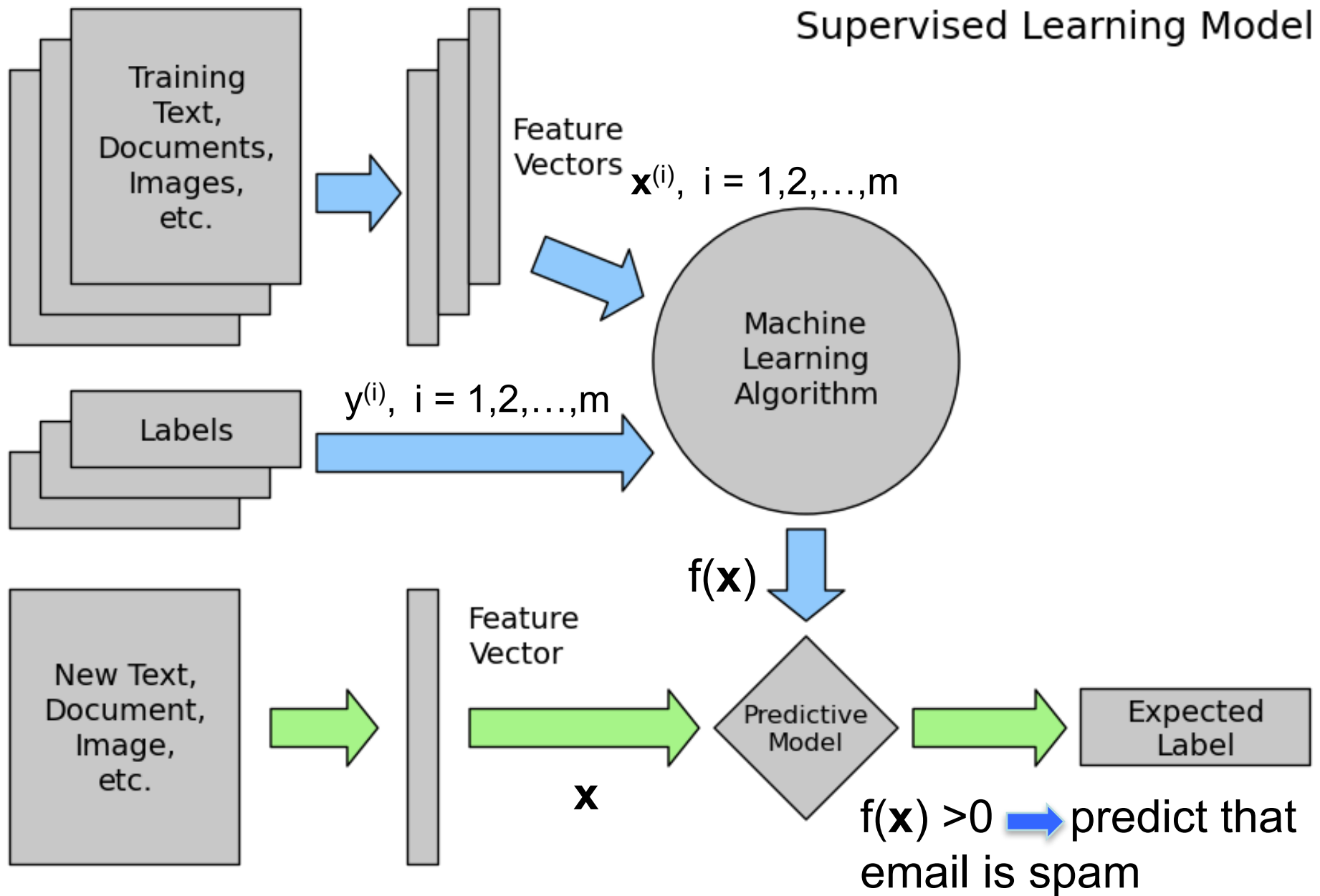
- What are the features?
 - The words in the training set!
 - For simplicity, will not use metadata in email headers
- Collect all the words occurring in train.txt:
 $n = \# \text{ words} = \# \text{ of features}$
- For a given email, let
 - $x_j = 1$ if j th word appears in email
 - $x_j = 0$ otherwise
 - $\mathbf{x} = (x_1, x_2, \dots, x_n)$ “feature vector”

Homework Assignment 1 (c)

- $\mathbf{x} = (x_1, x_2, \dots, x_n)$ “feature vector”
- $\mathbf{x}^{(i)}$ = feature vector for i^{th} email
- Basic idea:
 - If spam feature vectors have different patterns (words) than non-spam vectors, can use feature vector to predict if spam or not!

Homework Assignment 1 (d)

- $\mathbf{x} = (x_1, x_2, \dots, x_n)$ “feature vector”
- $\mathbf{x}^{(i)}$ = feature vector for i^{th} email (“example”)
- $y^{(i)}$ = “label” of i^{th} email
= +1 (spam); = -1 (not spam)
- m = # of emails in train.txt
- Use $\mathbf{x}^{(i)}, y^{(i)}$, $i=1, \dots, m$ to “train model”
- Obtain $f(\mathbf{x})$ = predictor for arbitrary email \mathbf{x}
 > 0 say \mathbf{x} it’s spam
 < 0 say \mathbf{x} it’s not spam

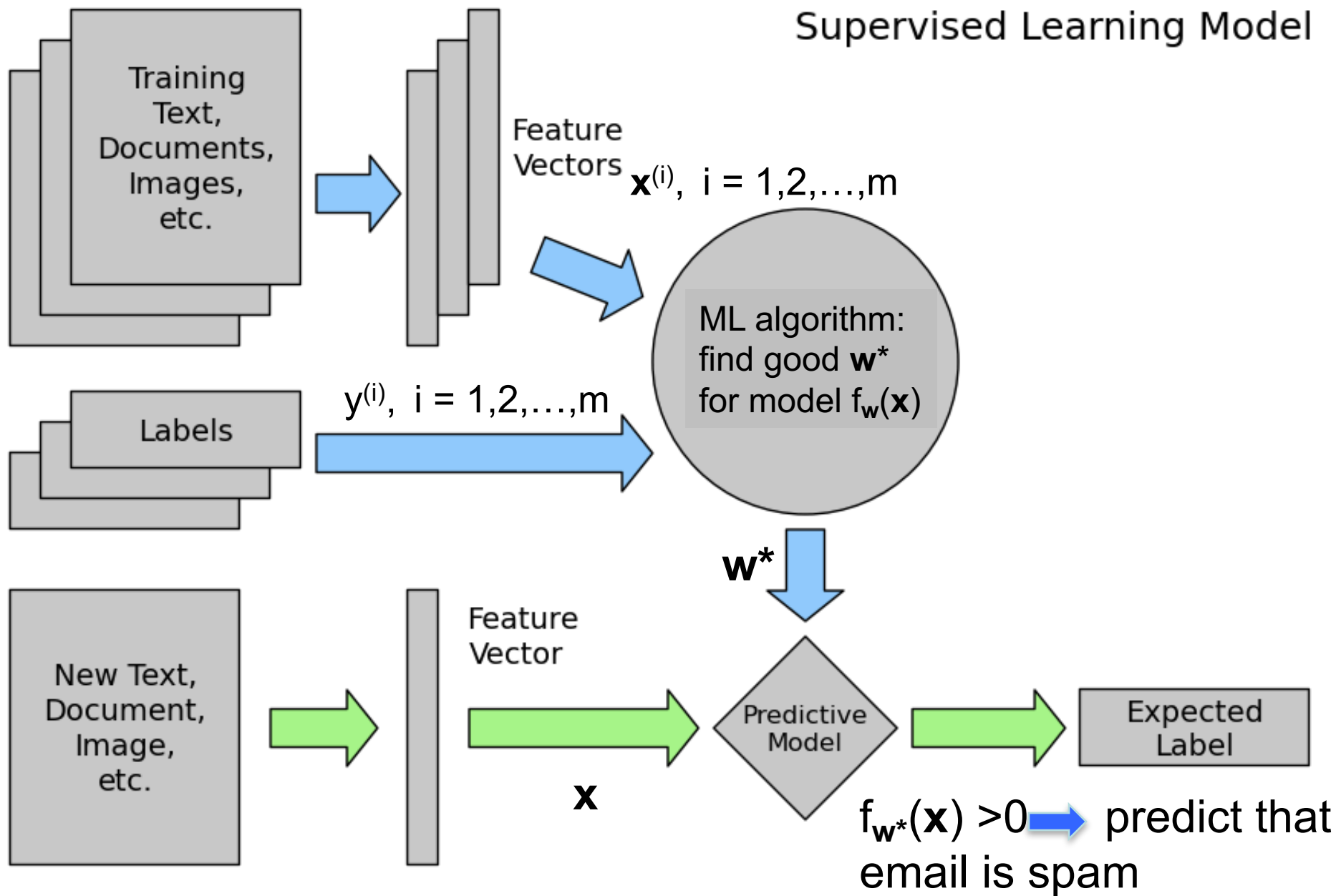


Data in Machine Learning

- **Numbers:** stock market prices; medical sensor data; meteorological data
- **Words:** emails, documents, articles,...
- **Images:** photos on Facebook; medical imaging
- **Sound:** speech; music

The Model

- The model is a parameterized function $f_{\mathbf{w}}(\mathbf{x})$ that maps a feature vector \mathbf{x} to the real numbers
 - (or to a probability distribution)
- \mathbf{w} is a vector of parameters (also called weights).
- For example, $f_{\mathbf{w}}(\mathbf{x}) = w_1x_1 + w_2x_2 + \dots + w_nx_n$ is a *linear model*.
- The training data determines what \mathbf{w} we should use to do prediction.



Review of Vector Algebra

(with geometric interpretations
on white board)

Vector

- $\mathbf{u} = (u_1, \dots, u_n)$ where each u_i , $i=1, \dots, n$, is a real number.
- What is the geometric interpretation?

Vector addition

- $\mathbf{u} = (u_1, \dots, u_n)$, $\mathbf{v} = (v_1, \dots, v_n)$
- $\mathbf{u} + \mathbf{v} =$

Scalar multiplication

- $\mathbf{u} = (u_1, \dots, u_n)$
- $s =$ a real number: “scalar”
- $s \times \mathbf{u} =$

Vector subtraction

- $\mathbf{u} = (u_1, \dots, u_n)$, $\mathbf{v} = (v_1, \dots, v_n)$
- $\mathbf{u} - \mathbf{v} = \mathbf{u} + (-\mathbf{v}) =$

Dot product

- $\mathbf{u} = (u_1, \dots, u_n)$, $\mathbf{v} = (v_1, \dots, v_n)$
- $\mathbf{u} \bullet \mathbf{v} =$

Length of vector (L_2 norm)

- $\mathbf{u} = (u_1, \dots, u_n)$
- $\|\mathbf{u}\|$ = length of vector =
- Also sometimes written as $|\mathbf{u}|$ or $\|\mathbf{u}\|_2$

Line in 2-dim plane

- Let w_1 , w_2 , c be fixed constants; u_1 and u_2 be variables.
- $w_1u_1 + w_2u_2 = c$ defines a line in the (u_1, u_2) plane
- The vector $\mathbf{w} = (w_1, w_2)$ is perpendicular to the line
- Points (u_1, u_2) such that $w_1u_1 + w_2u_2 > c$ lie on one side of the line; points $w_1u_1 + w_2u_2 < c$ lie on the other side of the line.

Plane in 3-dim space

- Let w_1, w_2, w_3, c be fixed constants.
- $w_1u_1 + w_2u_2 + w_3u_3 = c$ defines a plane in 3-dimensional space R^3
- The vector $\mathbf{w} = (w_1, w_2, w_3)$ is perpendicular to plane
 - Cuts R^3 into two halves
 - Points $\mathbf{w} \bullet \mathbf{u} > c$ lie on one side of plane
 - Points $\mathbf{w} \bullet \mathbf{u} < c$ lie on the other side

Hyperplane

- Let $\mathbf{w} = (w_1, w_2, \dots, w_n)$ be any fixed vector and c any scalar
- The equation $w_1 u_1 + w_2 u_2 + \dots + w_n u_n = c$ (equivalently $\mathbf{w} \bullet \mathbf{u} = c$) defines a hyperplane in n -dimensional space.
- \mathbf{w} is perpendicular to plane
 - Cuts \mathbb{R}^n into two halves
 - Points $\mathbf{w} \bullet \mathbf{u} > c$ lie on one side of plane
 - Points $\mathbf{w} \bullet \mathbf{u} < c$ lie on the other side

Back to Supervised Machine Learning

- $\mathbf{x} = (x_1, x_2, \dots, x_n)$ “feature vector”
- $f_{\mathbf{w}}(\mathbf{x})$ is a model
- *For example, Linear Model:*
$$f_{\mathbf{w}}(\mathbf{x}) = w_0 + w_1 x_1 + w_2 x_2 + \dots + w_n x_n$$

where $\mathbf{w} = (w_0, w_1, w_2, \dots, w_n)$
- ML algorithm uses training set to find good \mathbf{w}^* for $f_{\mathbf{w}}(\mathbf{x})$
- Given new email \mathbf{x} , predict \mathbf{x} is SPAM if $f_{\mathbf{w}^*}(\mathbf{x}) > 0$; otherwise not SPAM

Linear classifier

- $f_{\mathbf{w}}(\mathbf{x}) = w_0 + w_1 x_1 + w_2 x_2 + \dots + w_n x_n$
 > 0 say it's spam
 ≤ 0 say it's not spam
- w_0, w_1, \dots, w_n : “weights” (“parameters”)
- Define new feature x_0 with $x_0 := 1$.
- Can now write $f_{\mathbf{w}}(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x}$
- w_0 called the “bias”.

Simple example

- Just 2 features and bias:
 1. “free”
 2. “money”
 3. BIAS (always has value 1)

email
“free money”

x

| | | |
|-------|---|---|
| BIAS | : | 1 |
| free | : | 1 |
| money | : | 1 |
| ... | | |

w

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

w • x

$$\begin{aligned}(1)(-3) &+ \\ (1)(4) &+ \\ (1)(2) &+ \\ \dots & \\ &= 3\end{aligned}$$

w • x > 0 say email is SPAM!!!

How do we choose \mathbf{w} ?

- Desirable property for \mathbf{w} : for all $\mathbf{x}^{(i)}$ in training set,
 $\mathbf{w} \bullet \mathbf{x}^{(i)} > 0$ for all spam ($y^{(i)} = +1$)
 $\mathbf{w} \bullet \mathbf{x}^{(i)} < 0$ for all non-spam ($y^{(i)} = -1$)
- If above property holds, then \mathbf{w} correctly classifies all the training data.
- (Actually, we may not want this for *all* training examples $\mathbf{x}^{(i)}$ as it may lead to over-fitting. More later.)

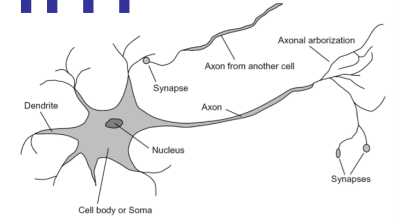
Succinct Equation

- An example $\mathbf{x}^{(i)}$, $y^{(i)}$ is correctly classified if $\mathbf{w} \cdot \mathbf{x}^{(i)} > 0$ when $y^{(i)} = +1$, and $\mathbf{w} \cdot \mathbf{x}^{(i)} < 0$ when $y^{(i)} = -1$
- More succinct way of writing this:
 - An example $\mathbf{x}^{(i)}$, $y^{(i)}$ is correctly classified if $y^{(i)}(\mathbf{w} \cdot \mathbf{x}^{(i)}) > 0$

Perceptron Algorithm

- Will be the first ML algorithm we study in class.
- Relatively simple.
- First of several supervised ML algorithms.

The perceptron algorithm



- 1957: Perceptron algorithm invented by Rosenblatt

Wikipedia: “A handsome bachelor, he drove a classic MGA sports... for several years taught an interdisciplinary undergraduate honors course entitled "Theory of Brain Mechanisms" that drew students equally from Cornell's Engineering and Liberal Arts colleges...this course was a melange of ideas .. experimental brain surgery on epileptic patients while conscious, experiments on .. the visual cortex of cats, ... analog and digital electronic circuits that modeled various details of neuronal behavior (i.e. the perceptron itself, as a machine).”

- Built on work of Hebb (1949); also developed by Widrow-Hoff (1960)
- 1960: Perceptron Mark 1 Computer – hardware implementation
- 1969: Minsky & Papert book shows perceptrons limited to *linearly separable* data, and Rosenblatt dies in boating accident
- 1970' s: Learning methods for two-layer neural networks

Goal of Perceptron Algorithm

- Find \mathbf{w} such that: $w_0 + w_1 x_1^{(i)} + \dots + w_n x_n^{(i)} > 0$ for all examples i that are spam, and $w_0 + w_1 x_1^{(i)} + \dots + w_n x_n^{(i)} < 0$ for all examples i that are not spam.
- \mathbf{w} is said to separate the data, with positive examples on one side of a hyperplane, and negative examples on other side.
- With such a \mathbf{w} , all of the training examples would be correctly classified. We would then expect most of the validation examples to be correctly classified.
- More succinctly, find \mathbf{w} such that $y^{(i)}(\mathbf{w} \bullet \mathbf{x}^{(i)}) > 0$ for all training examples $i = 1, \dots, m$.

The perceptron algorithm

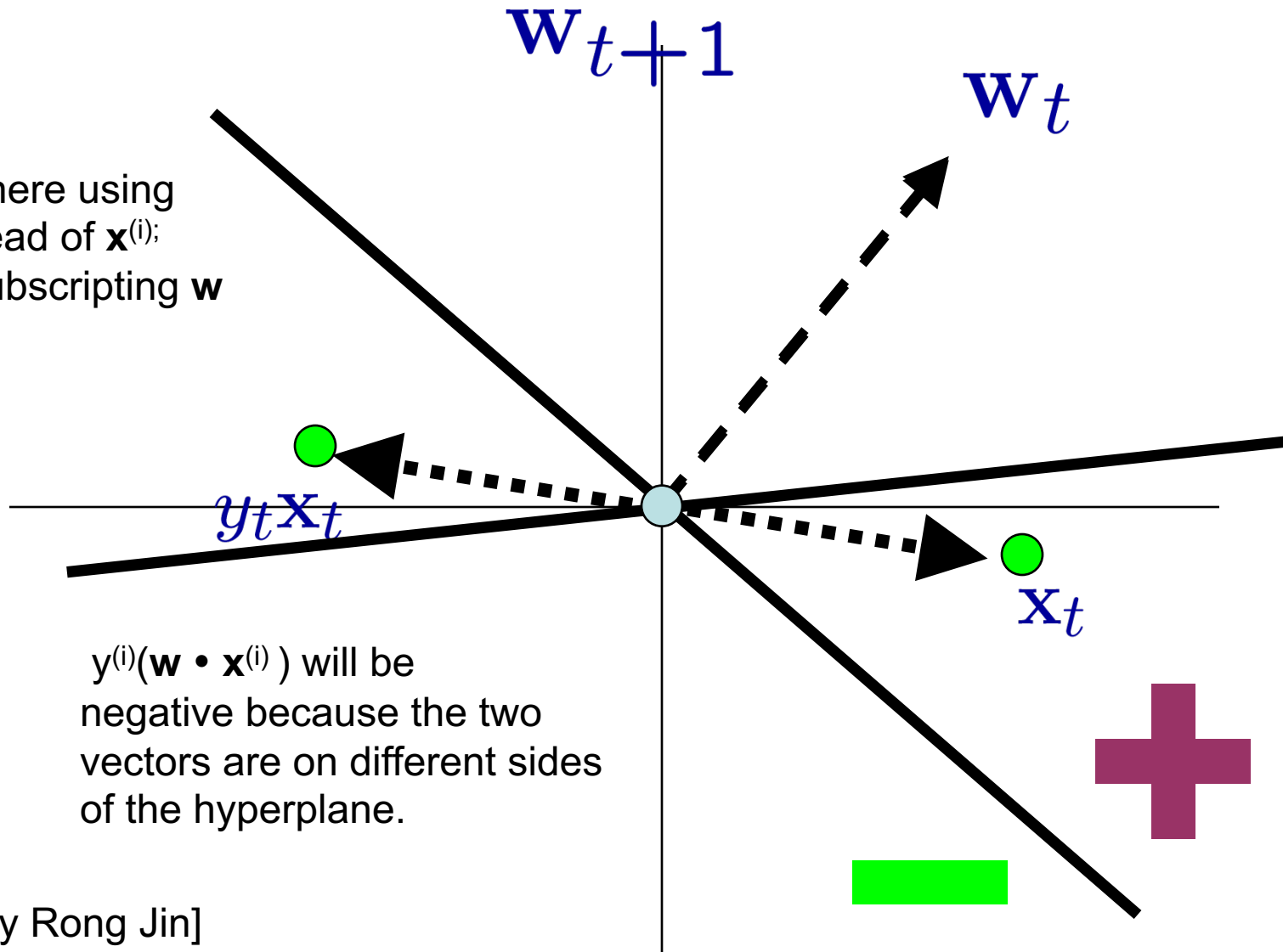
- Start with weight vector = $\mathbf{w} = (0,0,\dots,0)$
- Cycle through the training examples $\mathbf{x}^{(i)}, y^{(i)}$, $i=1,2,\dots,m,1,2,\dots,m,\dots$

- For $i=1,2,\dots,m,1,2,\dots,m,\dots$
 - if $y^{(i)}(\mathbf{w} \bullet \mathbf{x}^{(i)}) > 0$: (example correctly classified)
 $\mathbf{w} = \mathbf{w}$
 - else: (example incorrectly classified)
 $\mathbf{w} = \mathbf{w} + y^{(i)} \mathbf{x}^{(i)}$

If for some \mathbf{w} , get $y^{(i)}(\mathbf{w} \bullet \mathbf{x}^{(i)}) > 0$ for all $i=1,\dots,m$, stop and use that \mathbf{w} !

Geometrical Interpretation

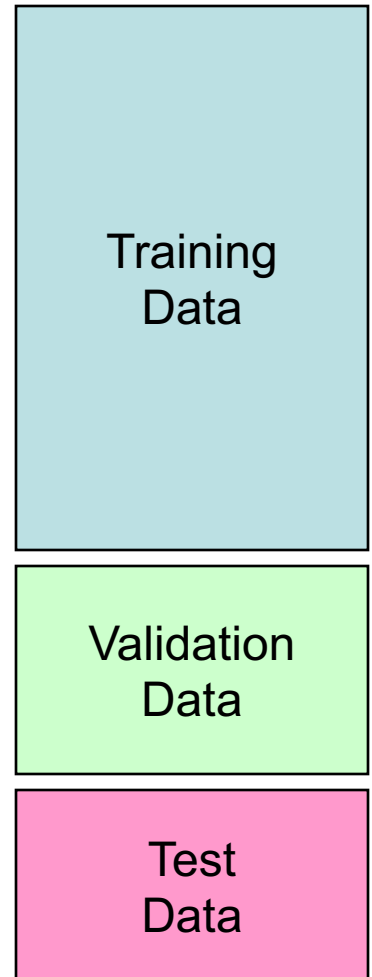
Note: here using \mathbf{x}_t instead of $\mathbf{x}^{(i)}$;
also subscripting \mathbf{w}



ML Methodology

- Experimentation cycle

- Select model, features, algorithm
- Train using training data (ie, use algorithm to find good weights)
- Evaluate using validation data: number of instances in validation set predicted correctly.
- Modify features or model or algorithm
- Train again....
- Find best model, features,...
- Using best model, features, train with both training data and validation data
- Compute final accuracy of test data
- Very important: never “peek” at the test set!



What questions should we ask about a learning algorithm?

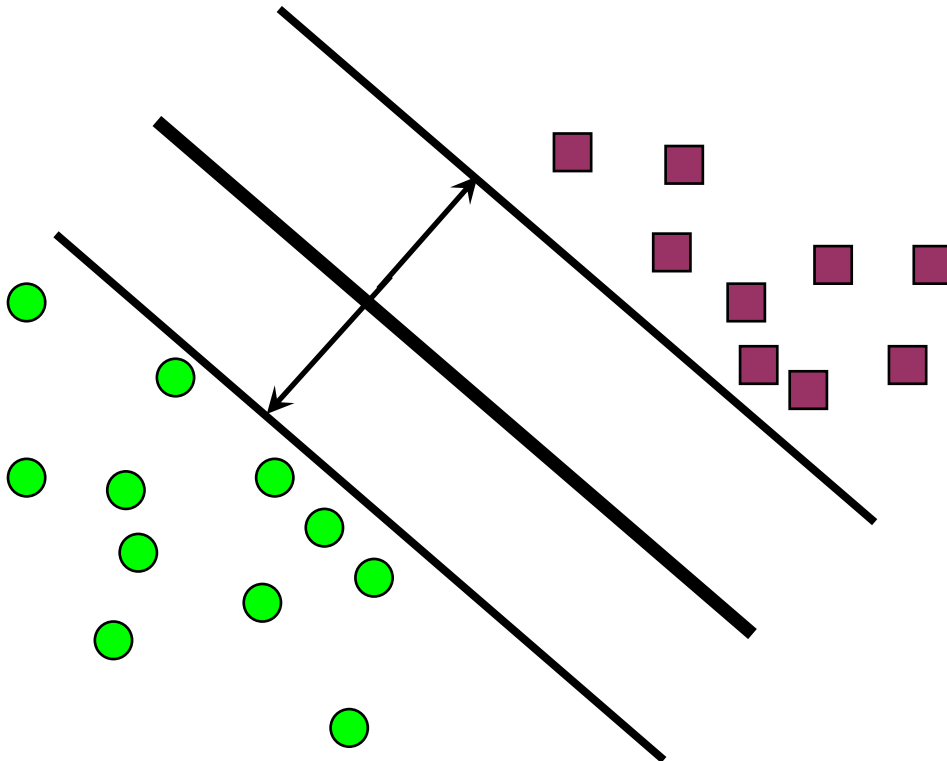
- Will the algorithm converge? If so, how many iterations are required?
- How well does the resulting classifier generalize to unseen data?

Linearly Separable

Definition: The data is linearly separable if there exists \mathbf{w} such that for every example i :

$\mathbf{w} \cdot \mathbf{x}^{(i)} > 0$ when $y^{(i)} = +1$ and

$\mathbf{w} \cdot \mathbf{x}^{(i)} < 0$ when $y^{(i)} = -1$



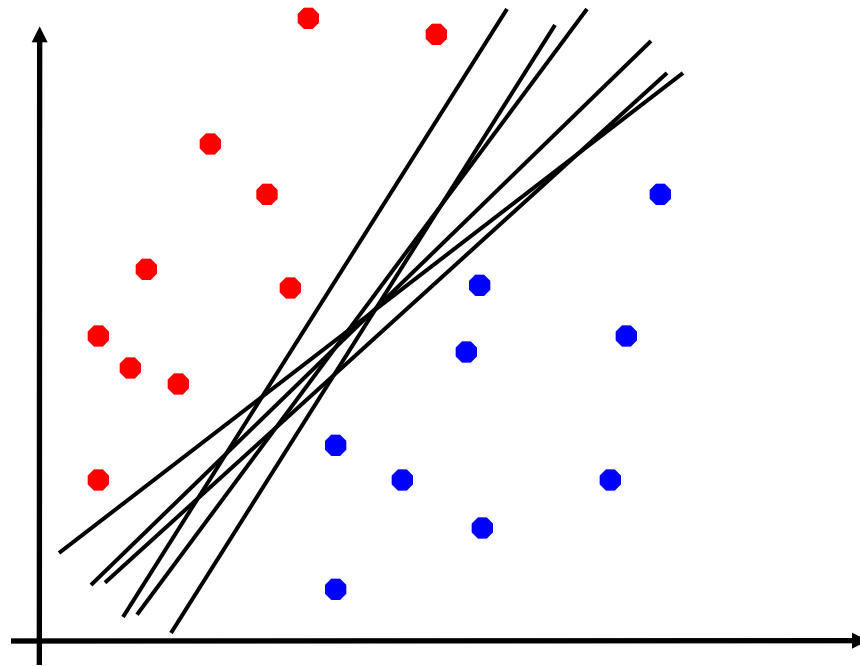
Theorem: If the training data is linearly separable, then the Perceptron algorithm will end and find a separating plane.

Convergence for Perceptron Algorithm

- **Theorem:** If the training data is linearly separable, then the Perceptron algorithm will end and find a separating plane.
- Perceptron algorithm makes a mistake whenever $y^{(i)}(\mathbf{w} \bullet \mathbf{x}^{(i)}) < 0$.
- *Proof strategy:* show that the number of mistakes is bounded.

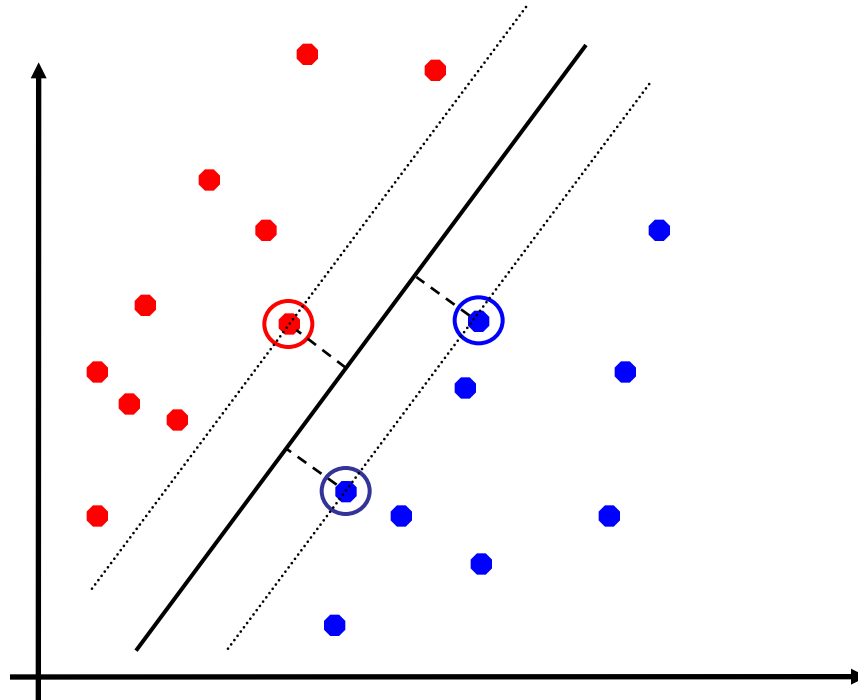
Linear Separators

- Which of these linear separators is optimal?



Later: Support Vector Machines

- SVMs (Vapnik, 1990' s) choose the linear separator with the **largest margin**



- Good according to intuition, theory, practice

Some ML jargon

- Classification
- Binary classification
- Model
- Regression
- Overfitting
- Generalization
- Supervised machine learning
- Unsupervised machine learning

Where is machine
learning used?

Growth of Machine Learning

- Machine learning is preferred approach to
 - Object recognition, face recognition
 - Speech recognition, Natural language processing
 - Medical outcomes analysis
 - Robot control
 - Computational biology
 - Sensor networks
 - ...
- This trend is accelerating
 - Big data
 - Improved machine learning algorithms
 - Faster computers
 - Good open-source software