Machine Learning

NYU Shanghai

Spring 2017

Logistics

- **Class webpage:**
  - https://sites.google.com/a/nyu.edu/nyu-shanghai-introduction-to-machine-learning-spring-2017/
  - Also a wechat group
- **Class:**
  - Monday, Wed, Fri, 9:45-11:00
- **My office hours:**
  - Wednesdays 11-12 and by appointment
  - Room 1415

Prerequisites

- Calculus (differential and integral)
- Intro to Computer Programming (Python)
- Probability and statistics (co-requisite)

- Desirable:
  - Linear Algebra
  - Multivariable Calculus
  - Data structures

Evaluation

- About 7 homeworks (35%)
  - Both theory and programming
  - See collaboration policy on class webpage
- Quizzes (25%)
- Final Project (35%): no final exam
- Course participation (5%)

Problem sets

- First assignment out today! Due Feb 15.
- 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 on your own
  - Hand in code as well, all in one document.

Final Project

- Teams of two; no teams of three
- Be creative – think of new problems that you can tackle using machine learning
  - Scope: ~40 hours/person
- Logistics:
  - Project proposal due April 1.
- 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)
- Other sources, including research papers

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 2017

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
  - Actually not a class of problems.
  - Just one of many methodologies for supervised machine learning
- Deep Learning
  - Fancy term for Neural networks ++
- Reinforcement Learning

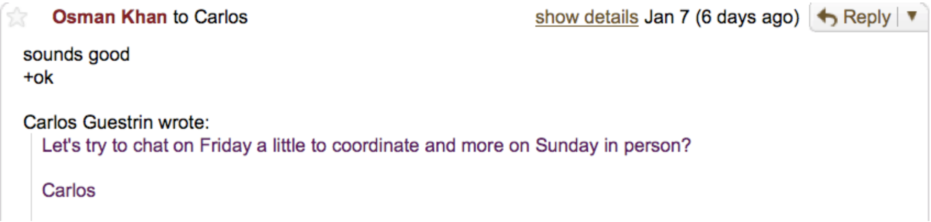
Machine Learning Examples

- Later we’ll survey some examples.
- We’ll now instead go through one example of supervised ML in some detail.
- Prepare you for the problem set.

- The Example:
  - Identify spam emails
  - Classify using perceptron algorithm

Spam filtering

data

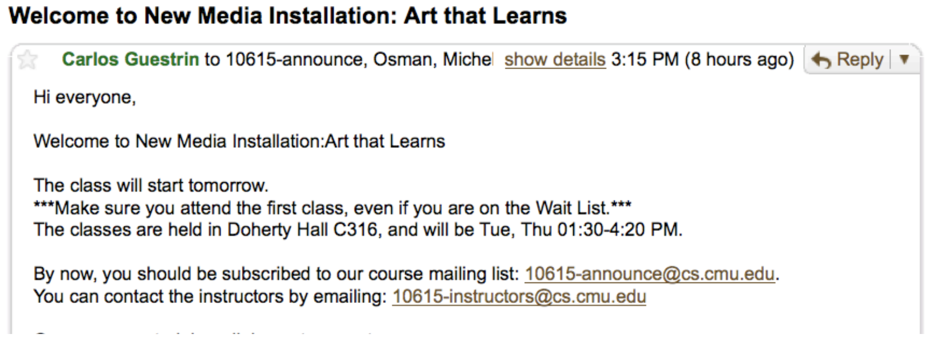


Osman Khan to Carlos show details Jan 7 (6 days ago)

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

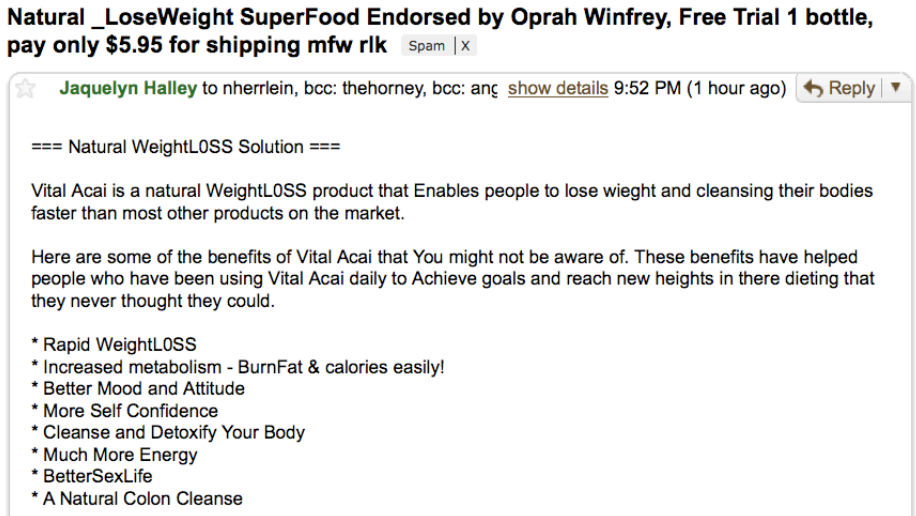
Carlos Guestrin to 10615-announce, Osman, Michel show details 3:15 PM (8 hours ago)

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

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

=== Natural WeightL0SS Solution ===

Vital Acai is a natural WeightL0SS 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 WeightL0SS
* 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

prediction

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 virture of its nature as being utterly confidencial 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, Iknow 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 4000 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 = # words = # of features
- For a given email, let
  - $x_j = 1$ if jth word appears in email
  - $x_j = 0$ otherwise
  - $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ "feature vector"

Homework Assignment 1 (c)

- $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ "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)

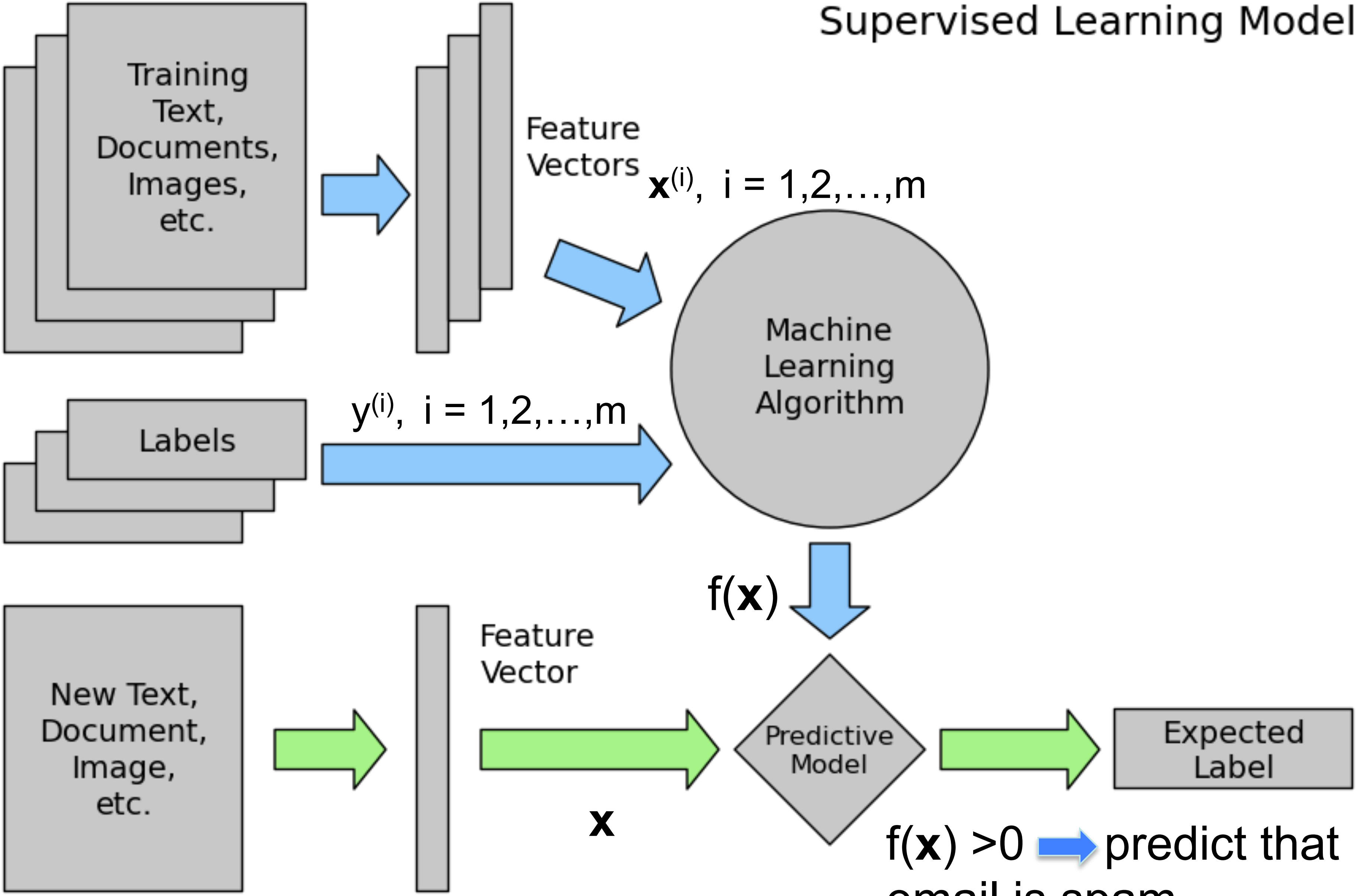
- $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ "feature vector"
- $\mathbf{x}^{(i)}$ = feature vector for ith email ("example")
- $y^{(i)}$ = "label" of ith email

  = +1 (spam); = -1 (not spam)
- m = # of emails in train.txt
- Use $\mathbf{x}^{(i)}, y^{(i)}$, $i=1,\ldots,m$ 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

Supervised Learning Model



Training Text, Documents, Images, etc.

Feature Vectors

$\mathbf{x}^{(i)}$, $i = 1,2,\ldots,m$

Labels

$y^{(i)}$, $i = 1,2,\ldots,m$

Machine Learning Algorithm

$f(\mathbf{x})$

New Text, Document, Image, etc.

Feature Vector

$\mathbf{x}$

Predictive Model

Expected Label

$f(\mathbf{x}) > 0$ → predict that email is spam

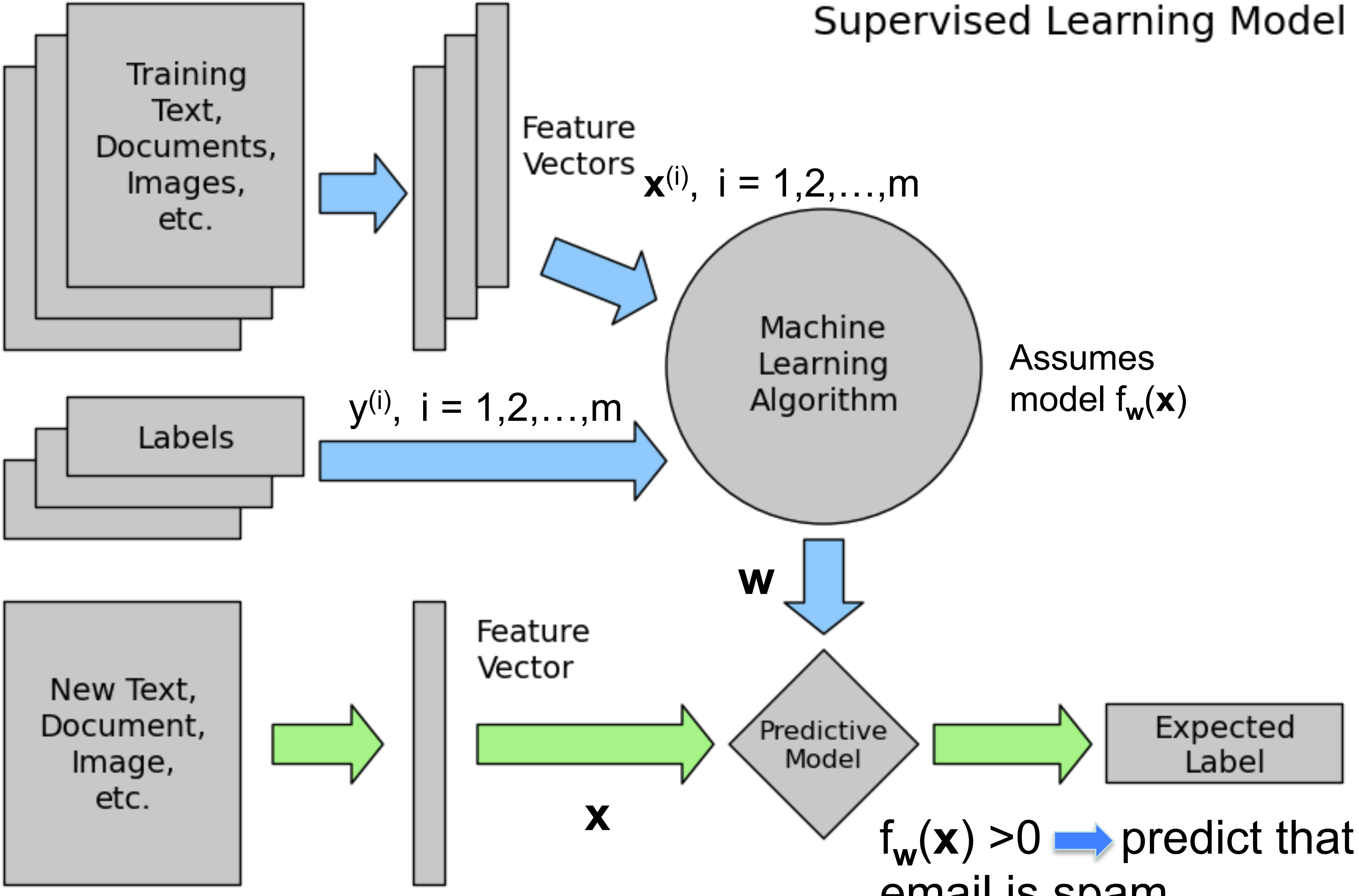
Data in Machine Learning

- **Numbers:** stock market prices; medical sensor data; meteorological data
- **Words:** spam detection; document summarization; spelling correction
- **Images:** photos on Facebook; medical imaging
- **Sound:** speech; music

The Model

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

Supervised Learning Model



Training Text, Documents, Images, etc.

Feature Vectors

$\mathbf{x}^{(i)}$, $i = 1,2,\ldots,m$

Labels

$y^{(i)}$, $i = 1,2,\ldots,m$

Machine Learning Algorithm

Assumes model $f_w(\mathbf{x})$

$\mathbf{w}$

New Text, Document, Image, etc.

Feature Vector

$\mathbf{x}$

Predictive Model

Expected Label

$f_w(\mathbf{x}) > 0$ → predict that email is spam

Review of Vector Algebra

(with geometric interpretations
on white board)

Vector

- $\mathbf{u} = (u_1,\ldots,u_n)$ where each u_i, $i=1,\ldots,n$, is a real number.

- What is the geometric interpretation?

Vector addition

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

Scalar multiplication

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

Vector subtraction

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

Dot product

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

Length of vector (L_2 norm)

- $\mathbf{u} = (u_1, \ldots, u_n)$
- $||\mathbf{u}||$ = 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 **w** = (w_1,w_2,w_3) is perpendicular to plane
  - Cuts R^3 into two halves
  - Points **w•u** > c lie on one side of plane
  - Points **w•u** < c lie on the other side

Hyperplane

- Let **w =** $(w_1,w_2,..,w_n)$ be any fixed vector and c any scalar
- The equation $w_1u_1 + w_2u_2 + \ldots + w_nu_n = c$ (equivalently **w•u** = c) defines a hyperplane in n-dimensional space.
- **w** is perpendicular to plane
  - Cuts R^n into two halves
  - Points **w•u** > c lie on one side of plane
  - Points **w•u** < c lie on the other side

Back to Supervised Machine Learning

- $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ “feature vector”
- ML algorithm uses training set to find function $f(\mathbf{x})$
- Given new email $\mathbf{x}$, predict $\mathbf{x}$ is SPAM if $f(\mathbf{x}) > 0$; otherwise not SPAM
- *Model*: limit our choose of $f(\bullet)$ to a class of functions, for example, linear functions, e.g.

$$f_{\mathbf{w}}(\mathbf{x}) = w_0 + w_1 x_1 + w_2 x_2 + \ldots + w_n x_n$$

where $\mathbf{w} = (w_0, w_1, w_2, .., w_n)$

Linear classifier

- $f_{\mathbf{w}}(\mathbf{x}) = w_0 + w_1\, x_1 + w_2\, x_2 + \ldots + w_n x_n$

  > 0 say it's spam

  ≤ 0 say it's not spam

- $w_0, w_1, \ldots, w_n$: "weights" ("parameters")
- Define $x_0 = 1$. Can write $f_{\mathbf{w}}(\mathbf{x}) = \mathbf{w} \bullet \mathbf{x}$
- 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

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

w • **x** > 0 ➔ SPAM!!!

How do we choose **w**?

- Desirable property for **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 **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 $\mathbf{w} \bullet \mathbf{x}^{(i)} > 0$ when $y^{(i)} = +1$, and $\mathbf{w} \bullet \mathbf{x}^{(i)} \leq 0$ when $y^{(i)} = -1$
- More succinct way of writing this:
  - correctly classified if $y^{(i)}(\mathbf{w} \bullet \mathbf{x}^{(i)}) > 0$
  - incorrectly classified if $y^{(i)}(\mathbf{w} \bullet \mathbf{x}^{(i)}) < 0$

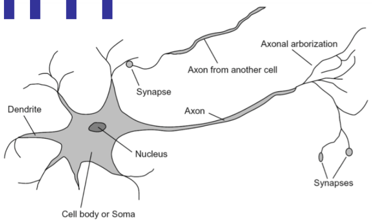
Perceptron Algorithm

- 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 Hebbs (1949); also developed by Widrow-Hoff (1960)

- 1960: Perceptron Mark 1 Computer – hardware implementation

- 1969: Minksky & Papert book shows perceptrons limited to *linearly separable* data, and Rosenblatt dies in boating accident

- 1970’s: Learning methods for two-layer neural networks

[William Cohen]

The perceptron algorithm

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

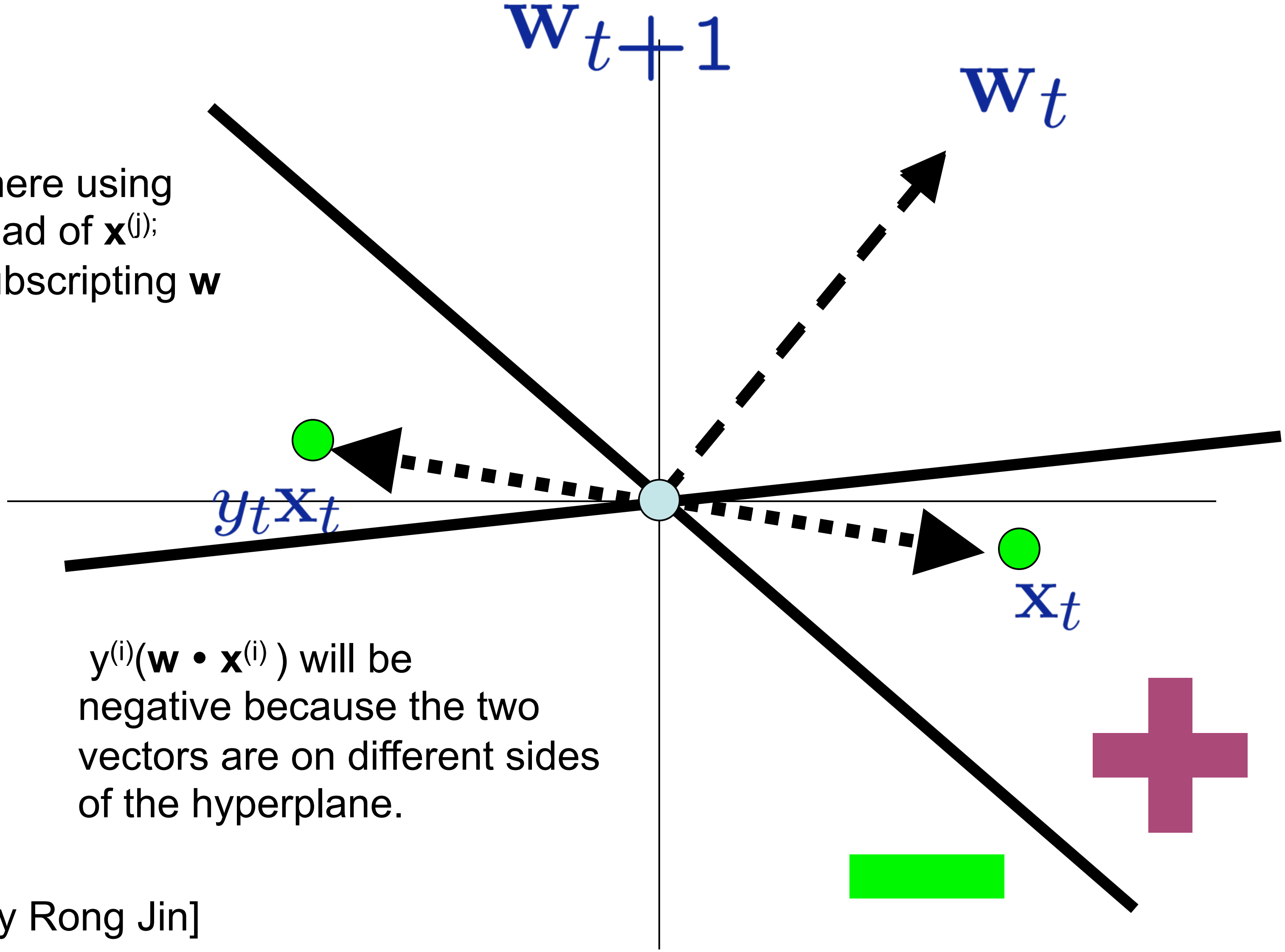
- For i=1,2,..,m,1,2,..,m,…
  - if $y^{(i)}(\mathbf{w} \bullet \mathbf{x}^{(i)}) > 0$: (example correctly classified)

    $\mathbf{w} = \mathbf{w}$
  - 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,…,m, stop and use that $\mathbf{w}$!

Geometrical Interpretation



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

$y^{(i)}(\mathbf{w} \bullet \mathbf{x}^{(i)})$ will be negative because the two vectors are on different sides of the hyperplane.

[Slide by Rong Jin]

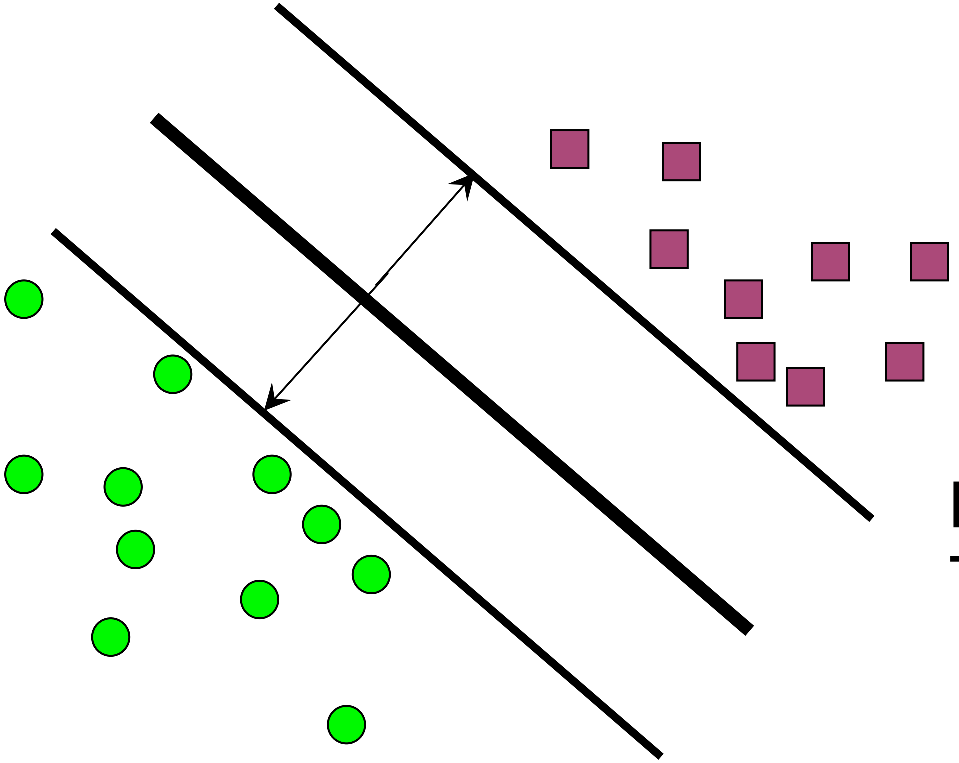
What questions should we ask about a learning algorithm?

- Will the algorithm converge? If so, how many iterations are required?
- If a weight vector with small training error exists, will perceptron find it?
- How well does the resulting classifier generalize to unseen data?

Linearly Separable

Definition: The data is linearly separable if there exists **w** such that for every example i:

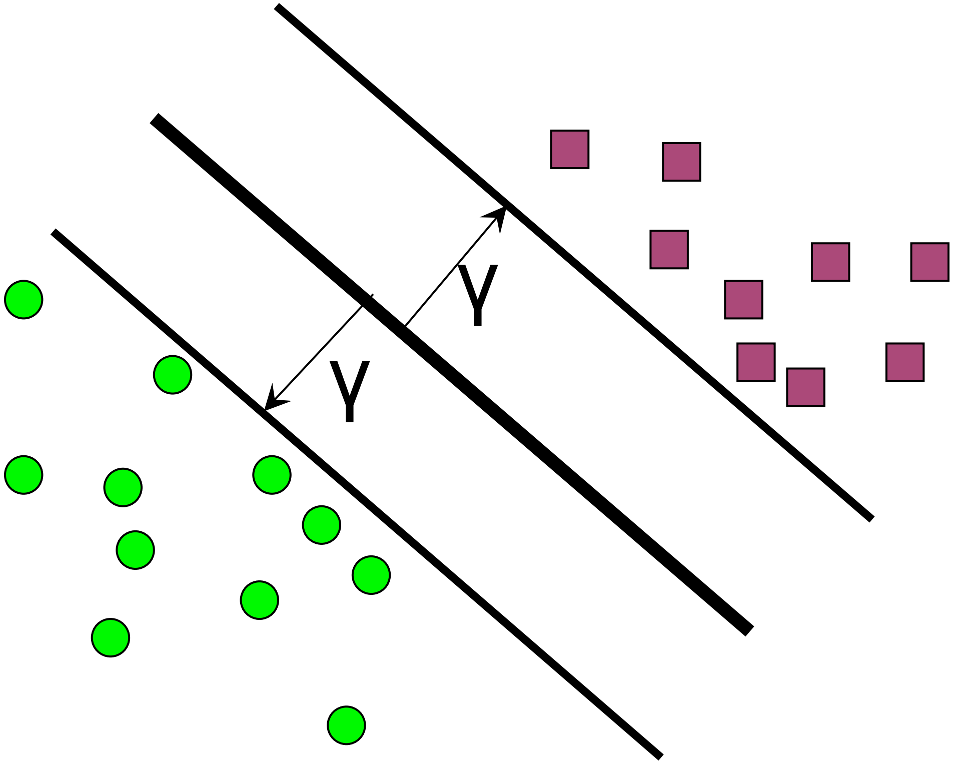
$\mathbf{w} \bullet \mathbf{x}^{(i)} > 0$ when $y^{(i)} = +1$ and
$\mathbf{w} \bullet \mathbf{x}^{(i)} < 0$ when $y^{(i)} = -1$



Let $\gamma = \min | \mathbf{w} \bullet \mathbf{x}^{(i)} | > 0$.
Then $y^{(i)}(\mathbf{w} \bullet \mathbf{x}^{(i)}) \geq \gamma$ for all i.

Geometric Margin

- Suppose data is linearly separable. Then there exists **w** with ||**w**|| =1 and γ > 0 such that $y^{(i)}(\mathbf{w} \bullet \mathbf{x}^{(i)}) \geq \gamma$ for all examples i.



γ is a geometric margin. (It is not necessarily the largest one.)

Mistake Bound for Perceptron

- Assume the data set is linearly separable with a *geometric* margin γ, i.e.,

$$\exists w^* \text{ s.t. } \|w^*\|_2 = 1 \text{ and } \forall t,\ y_t(w^* \cdot x_t) \geq \gamma$$

- Let *R* be such that $\|x_t\|_2 \leq R, \forall t$
- Theorem: The perceptron algorithm will converge, giving a linearly separable hyperplane. The maximum number of mistakes made by the perceptron algorithm is bounded by R^2/γ^2

Proof by induction

Assume we make a mistake for $(\mathbf{x}_t, y_t)$

$$\begin{tikzcd}
\|\mathbf{w}_{t+1}\|_2^2 = \|\mathbf{w}_t + y_t\mathbf{x}_t\|^2 \le \|\mathbf{w}_t\|_2^2 + R^2 \arrow[dd, "\text{A}"'] & \\
& \mathbf{w}_{t+1}^\top\mathbf{w}^* = \mathbf{w}_t^\top\mathbf{w}^* + y_t\mathbf{x}_t^\top\mathbf{w}^* \ge \mathbf{w}_t^\top\mathbf{w}^* + \gamma \arrow[d, "\text{B}"] \\
\|\mathbf{w}_t\|_2^2 \le M_t \cdot R^2 & \mathbf{w}_t^\top\mathbf{w}^* \ge M_t \cdot \gamma
\end{tikzcd}$$

$$\underbrace{\hphantom{M_t \cdot R^2 \qquad\qquad \mathbf{w}_t^\top\mathbf{w}^* \ge M_t}}_{\text{C}}$$

$$M_t \le \frac{R^2}{\gamma^2}$$

[Slide by Rong Jin]

Corollary

If the data is linearly separable, then the perceptron algorithm will converge. Moreover, the number of passes through the the training set is bounded by R^2/γ^2

ML Methodology

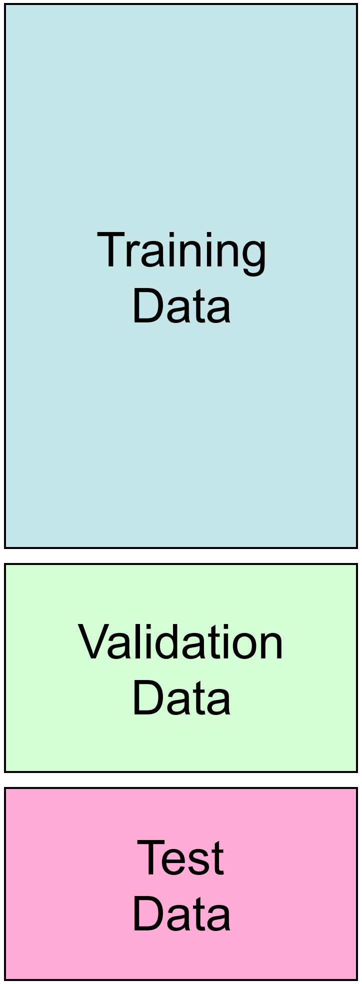
- Data: labeled instances (aka examples), e.g. emails marked spam, not spam
  - Training set
  - Validation set
  - Test set

  Allocate to these three, e.g. 60/20/20

- Features: attribute-value pairs which characterize each x

- Experimentation cycle
  - Select a hypothesis (class of functions) (e.g., linear classifier) Tune function parameters (weights) on *validation* set
  - Compute final accuracy of test set
  - Very important: never “peek” at the test set!

- Evaluation
  - Accuracy: fraction of instances in test set predicted correctly



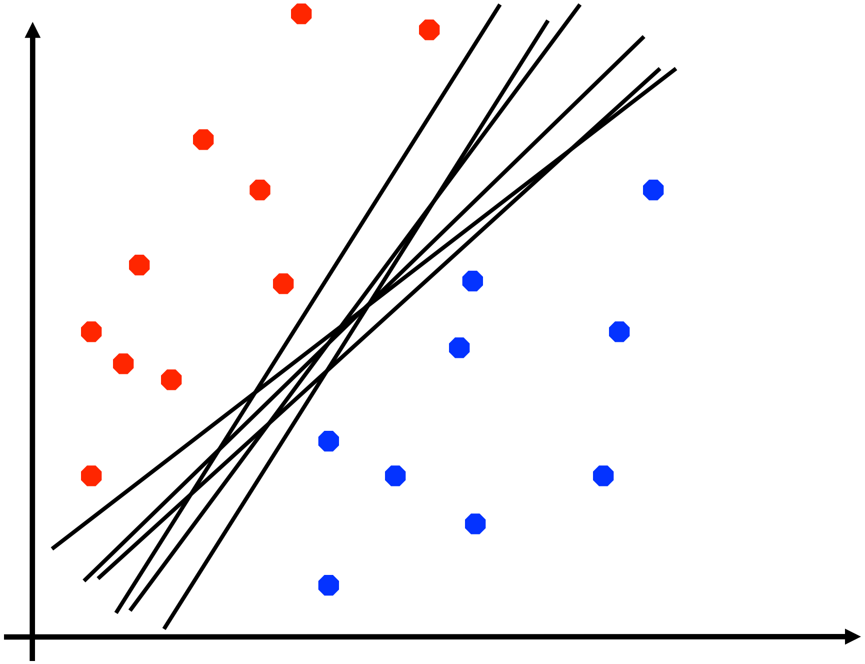
Training Data

Validation Data

Test Data

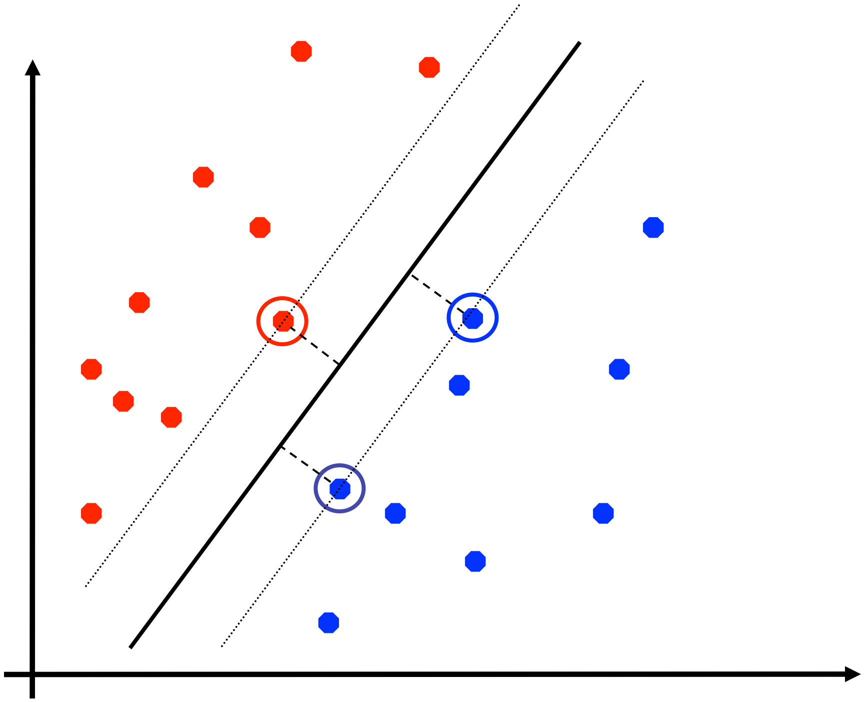
Linear Separators

- 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
- 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