10-607 Computational Foundations for Machine Learning
Machine Learning Department
School of Computer Science
Carnegie Mellon University

Proof Techniques + Perceptron Mistake Bound

Matt Gormley
Lecture 3
Oct. 29, 2018

Reminders

- Homework A:
  - Out: Mon, Oct. 28
  - Due: Tue, Oct. 2 at 11:59pm

Q&A

PROOF TECHNIQUES

Proof Techniques

Chalkboard

- Proof by Construction
- Proof by Contradiction
- Proof by Cases
- Proof by Contraposition

THE PERCEPTRON ALGORITHM

Perceptron: History

Imagine you are trying to build a new machine learning technique… your name is Frank Rosenblatt… and the year is 1957

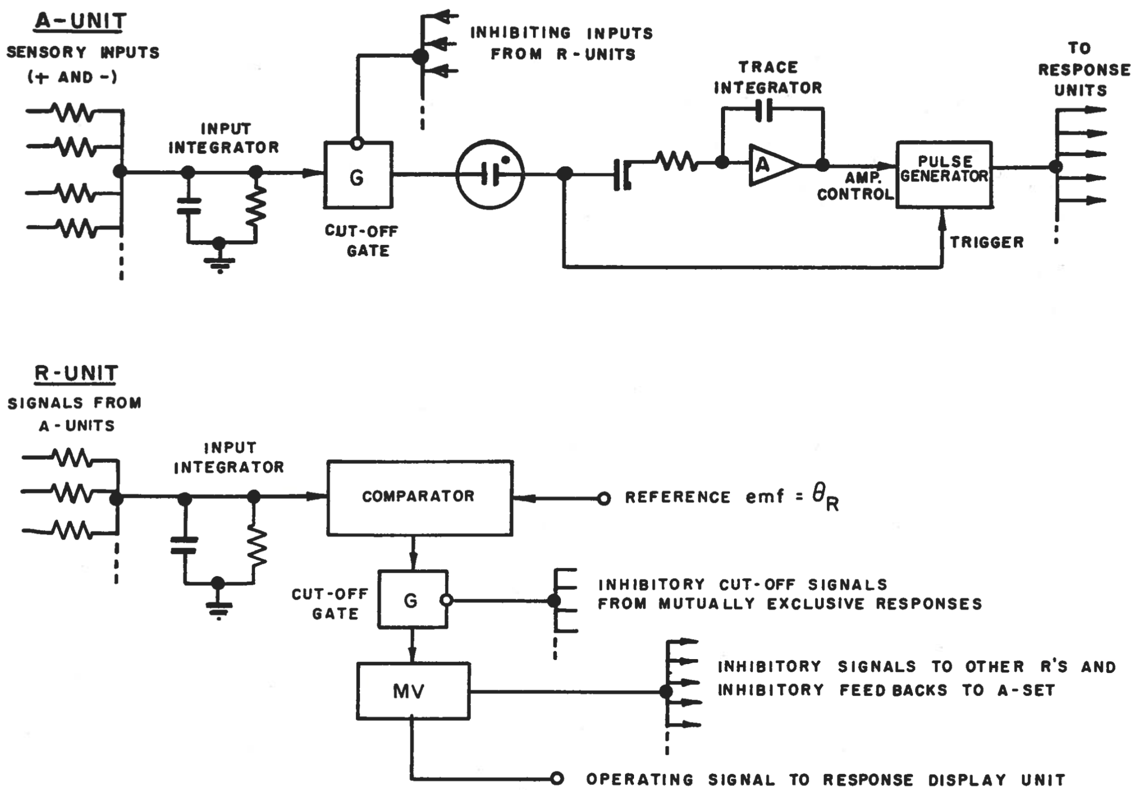
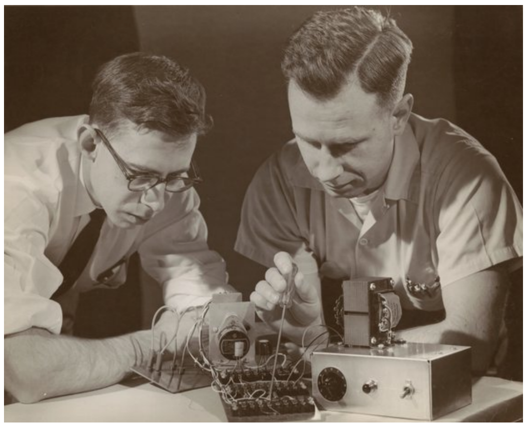
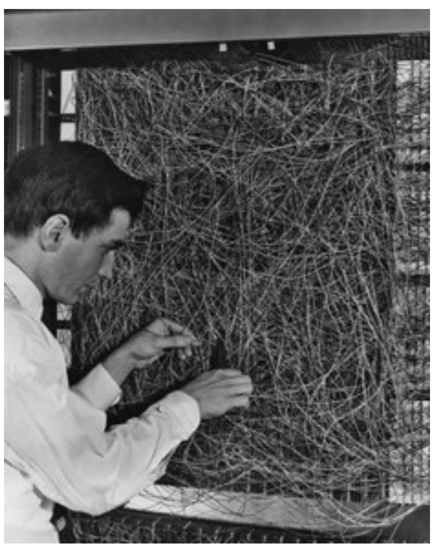


FIGURE 5
DESIGN OF TYPICAL UNITS

Perceptron: History

Imagine you are trying to build a new machine learning technique… your name is Frank Rosenblatt… and the year is 1957

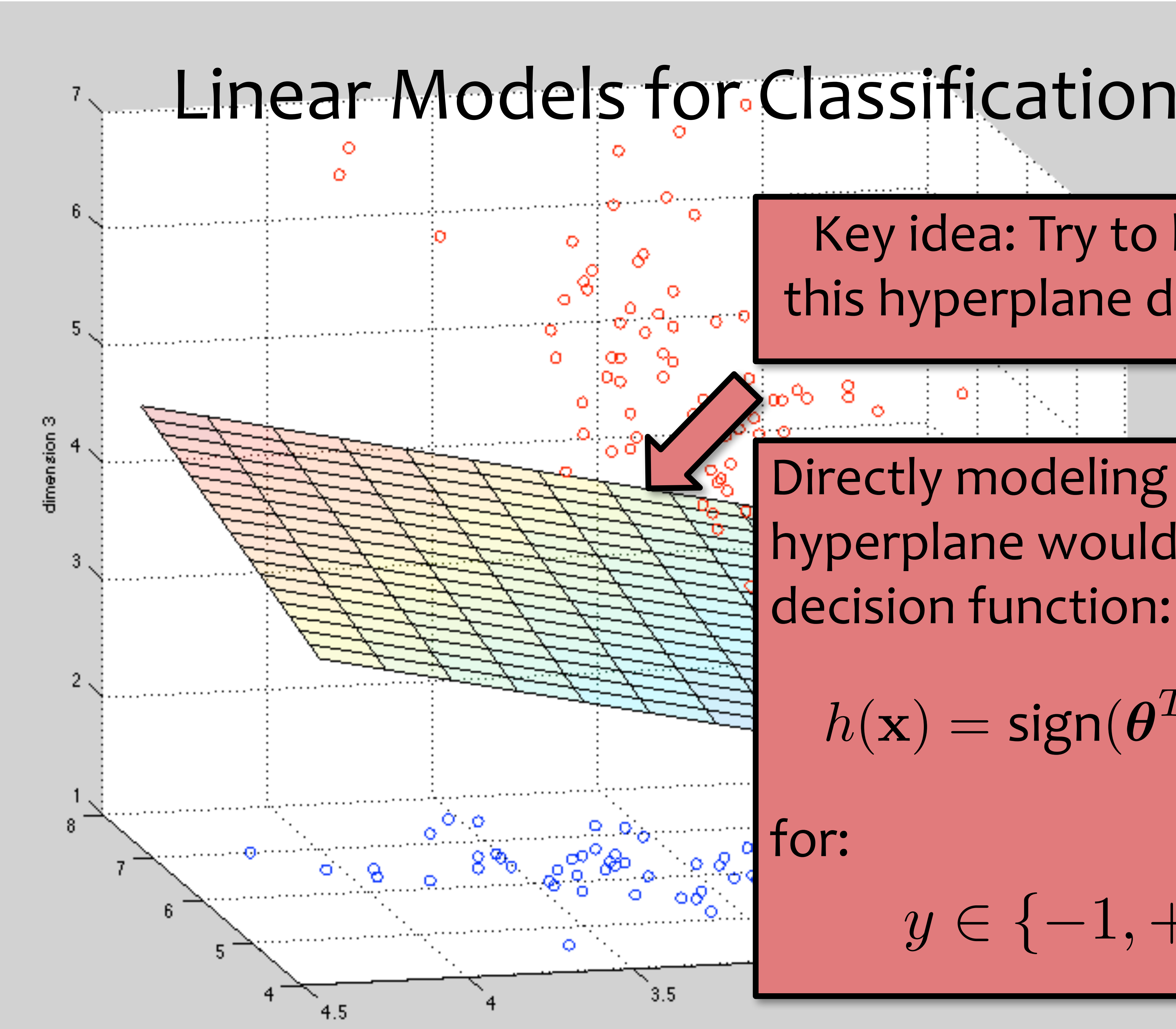




The New Yorker, December 6, 1958 P. 44

Talk story about the perceptron, a new electronic brain which hasn't been built, but which has been successfully simulated on the I.B.M. 704. Talk with Dr. Frank Rosenblatt, of the Cornell Aeronautical Laboratory, who is one of the two men who developed the prodigy; the other man is Dr. Marshall C. Yovits, of the Office of Naval Research, in Washington. Dr. Rosenblatt defined the perceptron as the first non-biological object which will achieve an organization o its external environment in a meaningful way. It interacts with its environment, forming concepts that have not been made ready for it by a human agent. If a triangle is held up, the perceptron's eye picks up the image & conveys it along a random succession of lines to the response units, where the image is registered. It can tell the difference betw. a cat and a dog, although it wouldn't be able to tell whether the dog was to theleft or right of the cat. Right now it is of no practical use, Dr. Rosenblatt conceded, but he said that one day it might be useful to send one into outer space to take in impressions for us.

Linear Models for Classification



Key idea: Try to learn this hyperplane directly

Directly modeling the hyperplane would use a decision function:

$$h(\mathbf{x}) = \text{sign}(\boldsymbol{\theta}^T \mathbf{x})$$

for:

$$y \in \{-1, +1\}$$

Geometry

In-Class Exercise

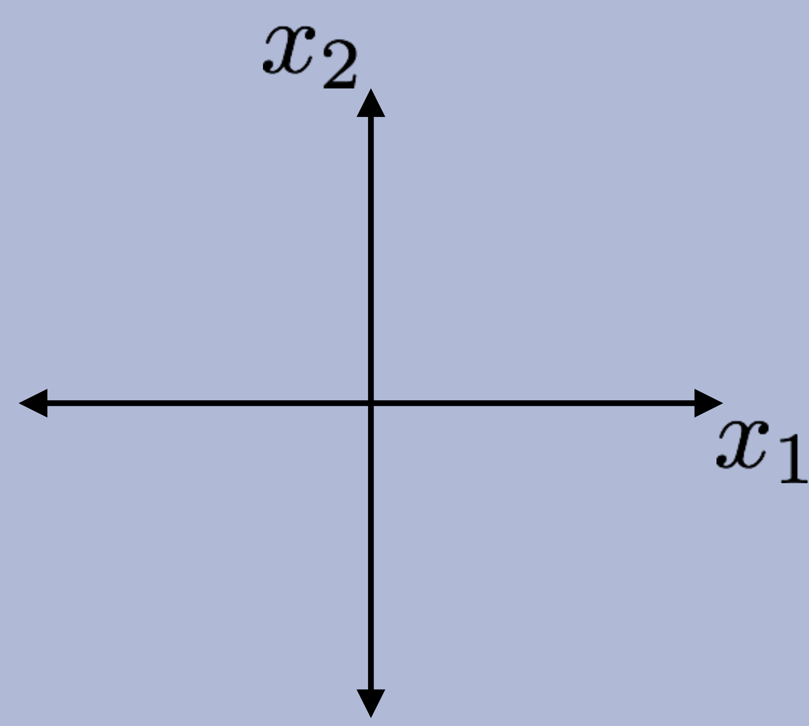
Draw a picture of the region corresponding to:

$$w_1 x_1 + w_2 x_2 + b > 0$$

where $w_1 = 2, w_2 = 3, b = 6$

Draw the vector
$\mathbf{w} = [w_1, w_2]$

Answer Here:

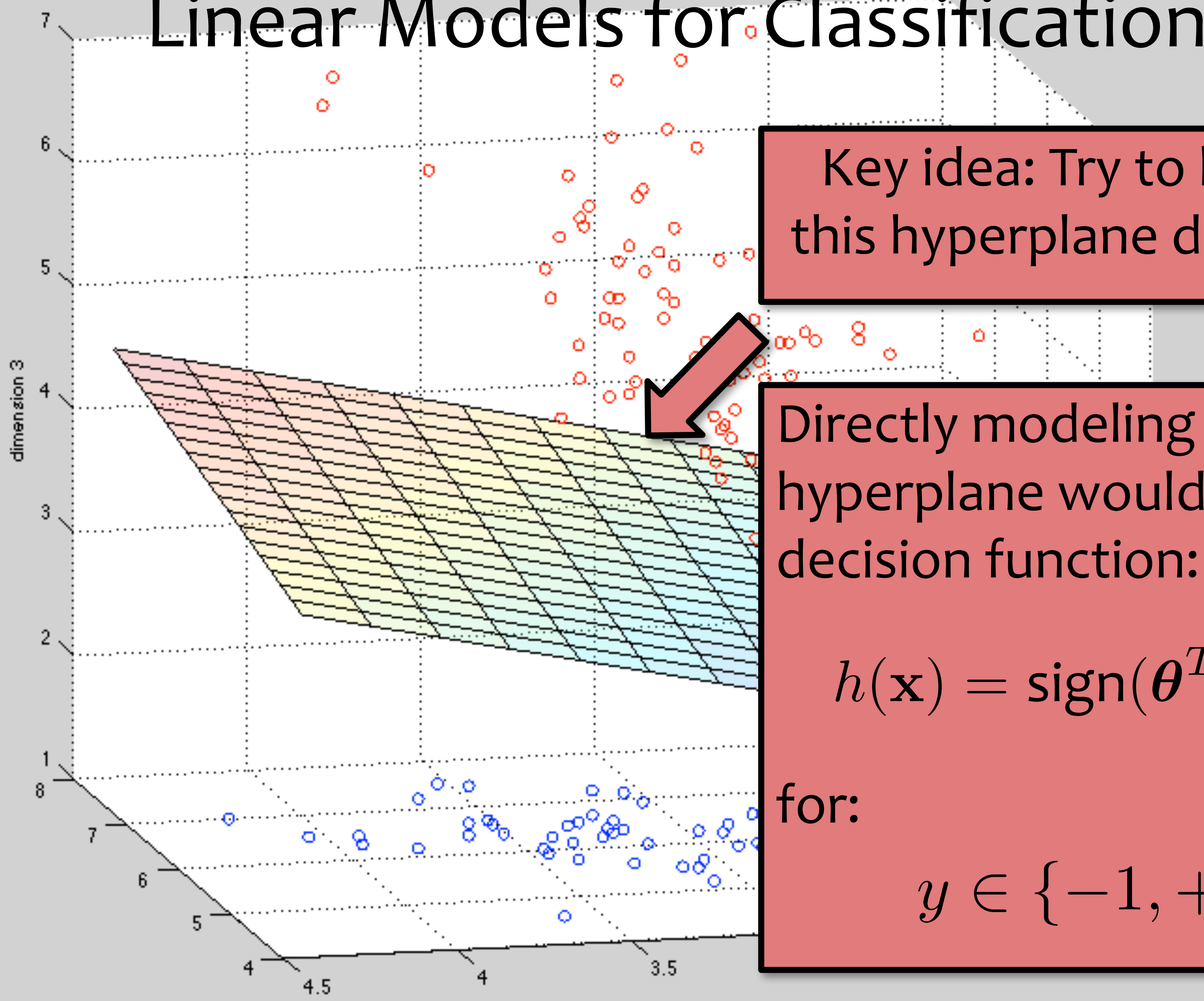


Visualizing Dot-Products

Chalkboard:

– vector in 2D
– line in 2D
– adding a bias term
– definition of orthogonality
– vector projection
– hyperplane definition
– half-space definitions

Linear Models for Classification



Key idea: Try to learn this hyperplane directly

Directly modeling the hyperplane would use a decision function:

$$h(\mathbf{x}) = \text{sign}(\boldsymbol{\theta}^T \mathbf{x})$$

for:

$$y \in \{-1, +1\}$$

Online vs. Batch Learning

Batch Learning

Learn from all the examples at once

Online Learning

Gradually learn as each example is received

Online Learning

Examples

1. **Stock market** prediction (what will the value of Alphabet Inc. be tomorrow?)
2. **Email** classification (distribution of both spam and regular mail changes over time, but the target function stays fixed - last year's spam still looks like spam)
3. **Recommendation** systems. Examples: recommending movies; predicting whether a user will be interested in a new news article
4. **Ad placement** in a new market

Slide adapted from Nina Balcan

Online Learning

For $i = 1, 2, 3, \ldots$:

- **Receive** an unlabeled instance $\mathbf{x}^{(i)}$
- **Predict** $y' = h_{\boldsymbol{\theta}}(\mathbf{x}^{(i)})$
- **Receive** true label $y^{(i)}$
- **Suffer loss** if a mistake was made, $y' \neq y^{(i)}$
- **Update** parameters $\boldsymbol{\theta}$

Goal:

- **Minimize** the number of **mistakes**

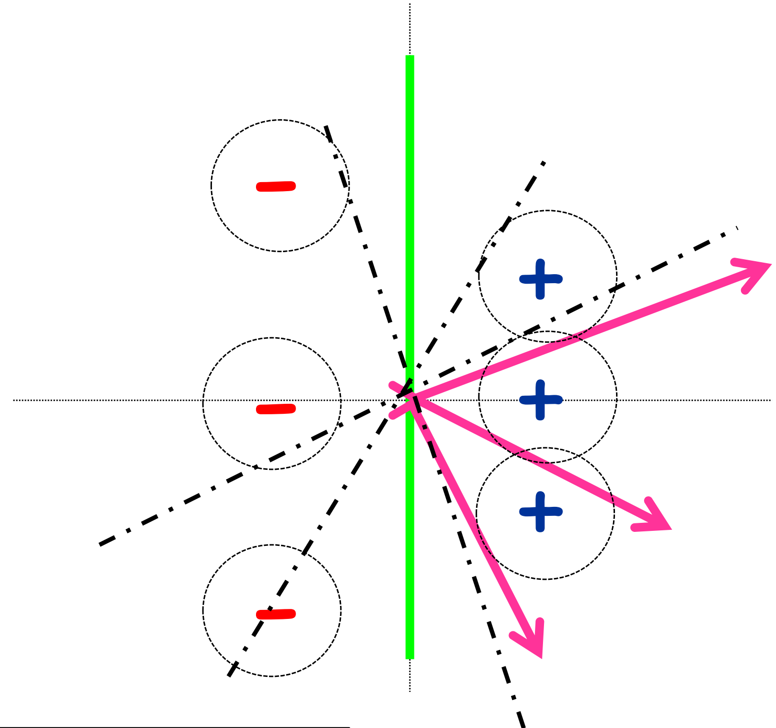
Perceptron

Chalkboard:

- (Online) Perceptron Algorithm
- Why do we need a bias term?
- Inductive Bias of Perceptron
- Limitations of Linear Models

Perceptron Algorithm: Example

Example: $(-1,2) -$ ✗
$(1,0) +$ ✓
$(1,1) +$ ✗
$(-1,0) -$ ✓
$(-1,-2) -$ ✗
$(1,-1) +$ ✓



Perceptron Algorithm: (without the bias term)

- Set t=1, start with all-zeroes weight vector w_1.
- Given example x, predict positive iff $w_t \cdot x \geq 0$.
- On a mistake, update as follows:
  - Mistake on positive, update $w_{t+1} \leftarrow w_t + x$
  - Mistake on negative, update $w_{t+1} \leftarrow w_t - x$

$w_1 = (0,0)$

$w_2 = w_1 - (-1,2) = (1,-2)$

$w_3 = w_2 + (1,1) = (2,-1)$

$w_4 = w_3 - (-1,-2) = (3,1)$

Slide adapted from Nina Balcan