

Lecture 15: Machine Learning

COSC 480 Data Science, Spring 2017
Michael Hay

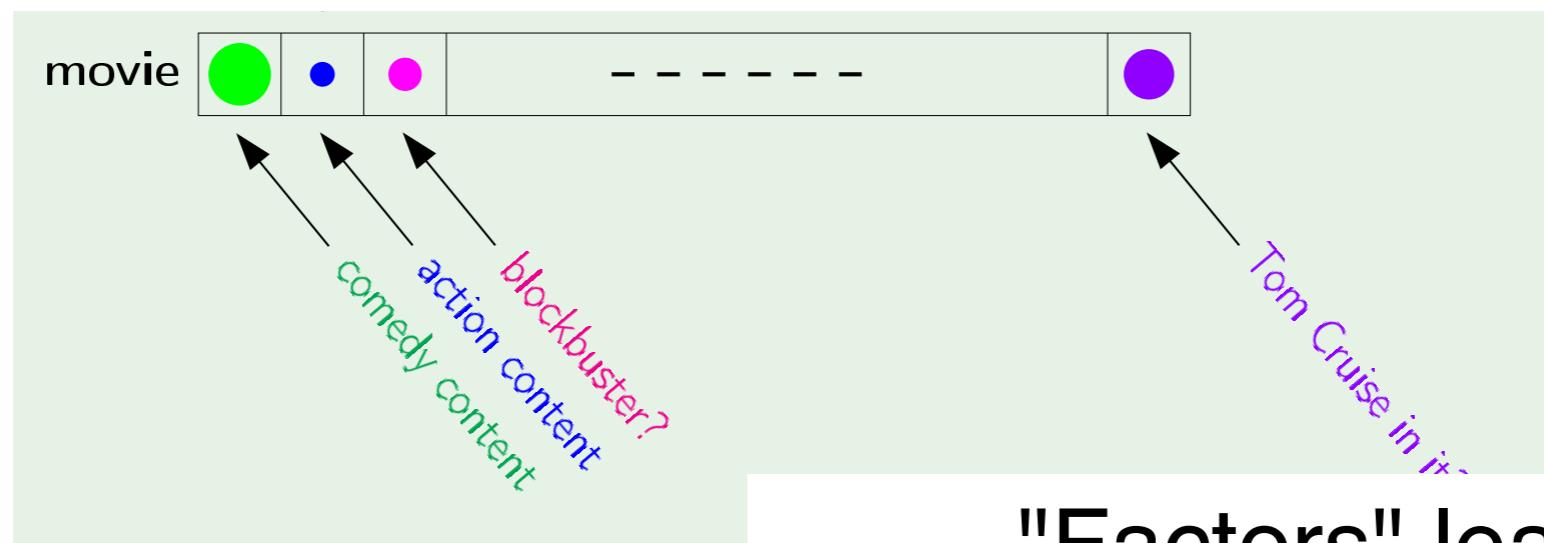
Motivating example #1

- Predict how a viewer will rate a movie
- 10% improvement
→ \$1 million prize!



Movie rating — a solution

Describe movie in terms of different "factors"



"Factors" learned automatically from data.

(Not as easily interpreted as example suggests.)

Example #2: Spam Filter

- Input: email
- Predict: "spam" or "ham"
- Data: old emails, each labeled "spam" or "ham"



- **Features:** The attributes used to make the ham / spam decision
 - Words: FREE!
 - Text Patterns: \$dd, CAPS, spelling errors, ...
 - Non-text: SenderInContacts
 - ...

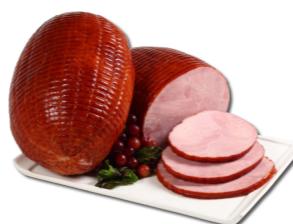


Dear Sir.

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

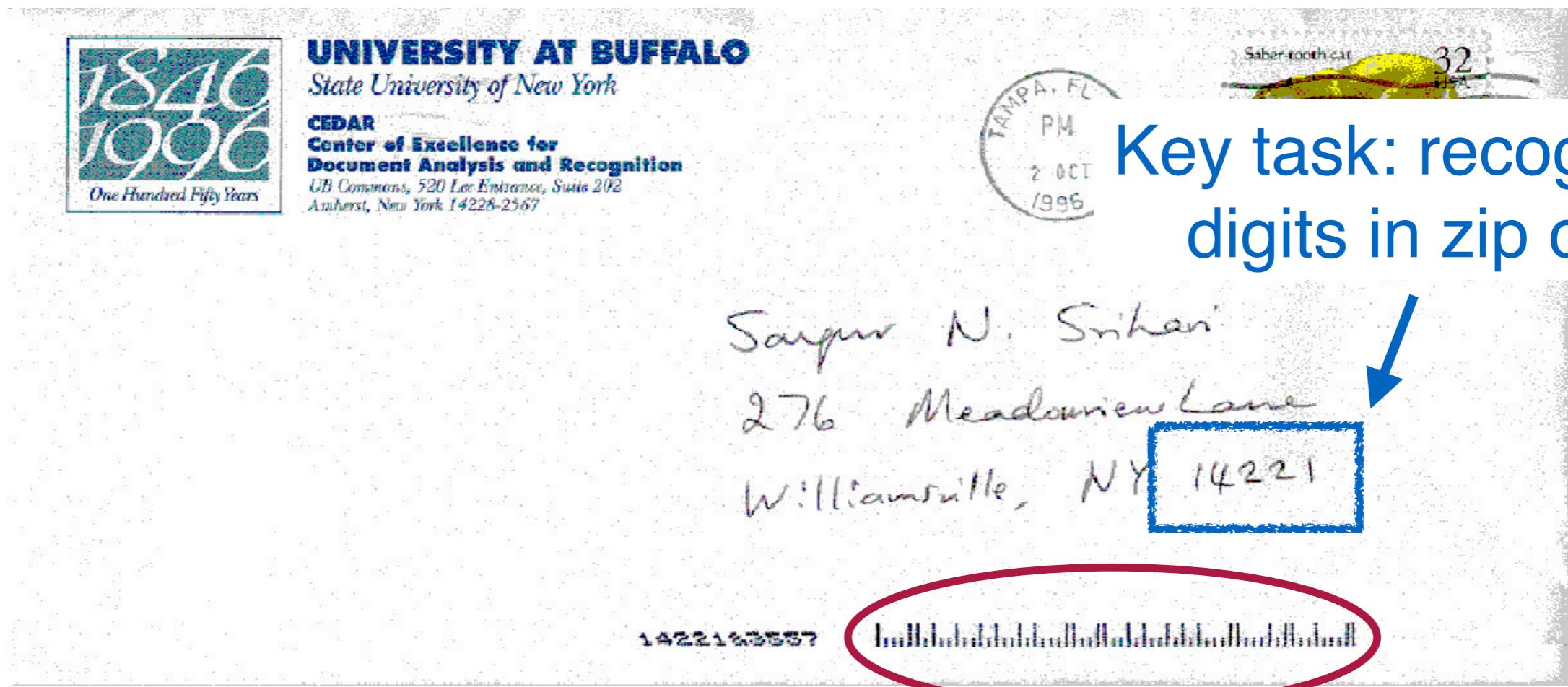
TO BE REMOVED FROM FUTURE MAILINGS, SIMPLY REPLY TO THIS MESSAGE AND PUT "REMOVE" IN THE SUBJECT.

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.

Example #3: Digit Recognition

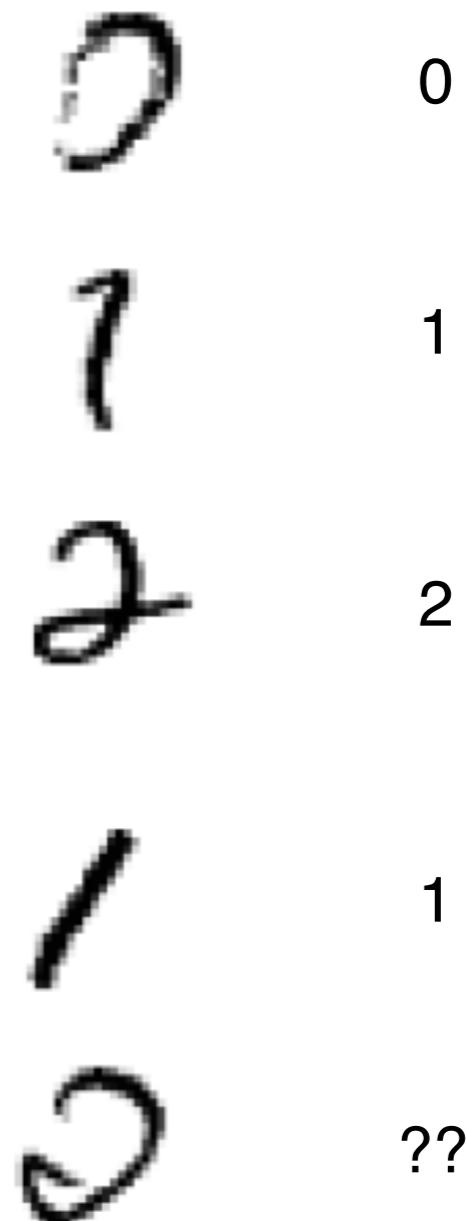


Key task: recognizing
digits in zip code

Postnet Bar Code representing Delivery Point

Example #3: Digit Recognition

- Input: images / pixel grids
- Predict: which digit 0-9
- Data: images of hand-written digits, each labeled
- **Features:** The attributes used to make the digit decision
 - Pixels: (6,8)=ON
 - Shape Patterns: NumComponents, AspectRatio, NumLoops, Vertical/Horizontal Symmetry
 - ...



Essence of machine learning

- A pattern exists
- We cannot pin it down mathematically
- We have data on it

Running example

- Task: credit approval
- Input: applicant information
- Output: approve credit?
- Data: applications of past customers

attribute	value
age	23 years
gender	male
salary	\$45,000
years in residence	1
years in job	1
current debt	\$15,000

Components of learning

Input:

$$X = X_1, X_2, \dots, X_d$$

(customer application)

Feature engineering

- Process of taking raw data (e.g., email) and converting into a set of features
- "Feature" synonymous with attributes, predictor variables, input variables, ...
- ***Big*** impact on learning!

Features can be...

- **binary**: hasALLCAPS (Y or N)
- **numerical**: Number of times Viagra occurs
- **categorical**: Top-level domain of sender (.com, .edu, ...)

Components of learning

Input: $X = X_1, X_2, \dots, X_d$ (customer application)

Output: Y (approve/deny credit?)

Components of learning

Input: $X = X_1, X_2, \dots, X_d$ (customer application)

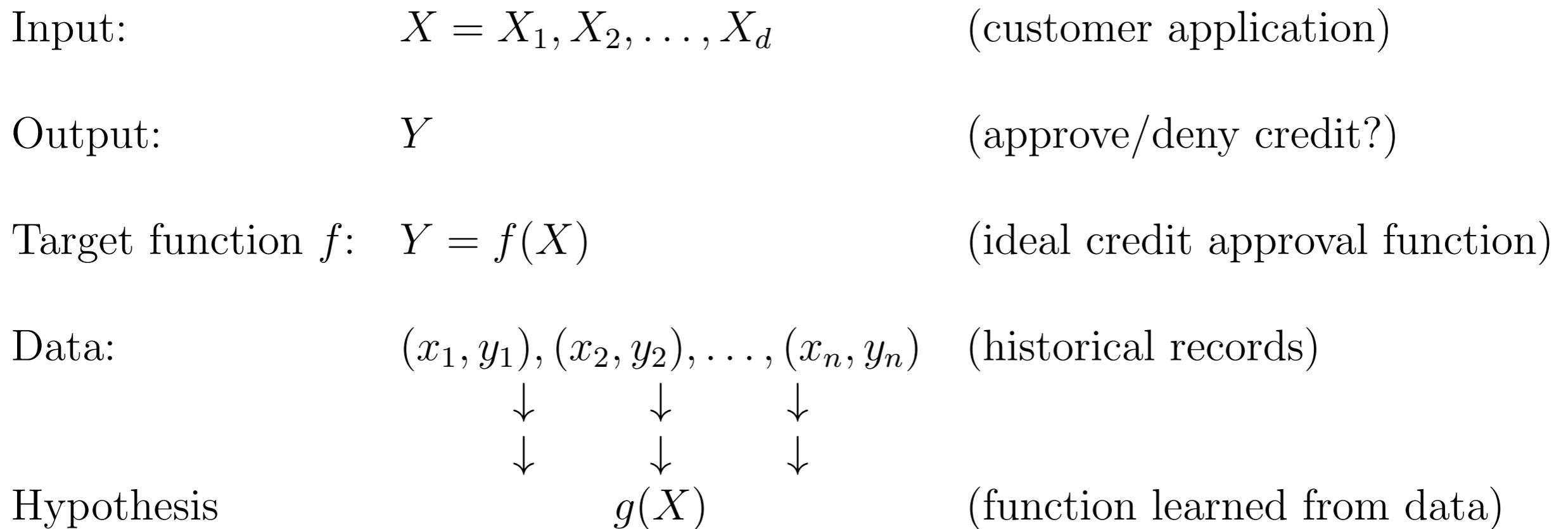
Output: Y (approve/deny credit?)

Target function f : $Y = f(X)$ (ideal credit approval function)

Components of learning

Input:	$X = X_1, X_2, \dots, X_d$	(customer application)
Output:	Y	(approve/deny credit?)
Target function f :	$Y = f(X)$	(ideal credit approval function)
Data:	$(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$	(historical records)

Components of learning



(ideal credit approval function)

Unknown Target Function

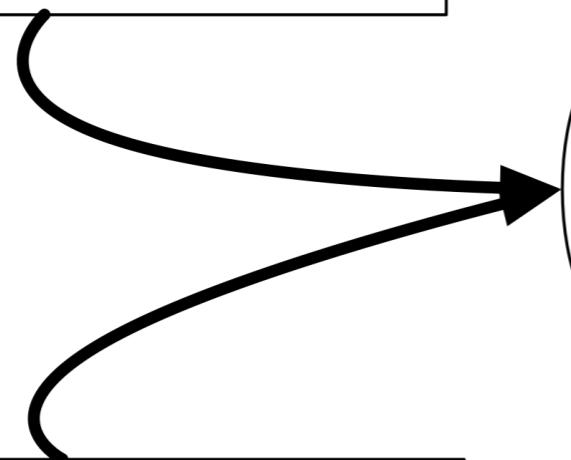
$$f(X) \rightarrow Y$$



(historical records of customers)

Training Examples

$$(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$$



**Learning
Algorithm**



Final Hypothesis
 $g \approx f$

(credit approval function, learned from data)

Hypothesis Set

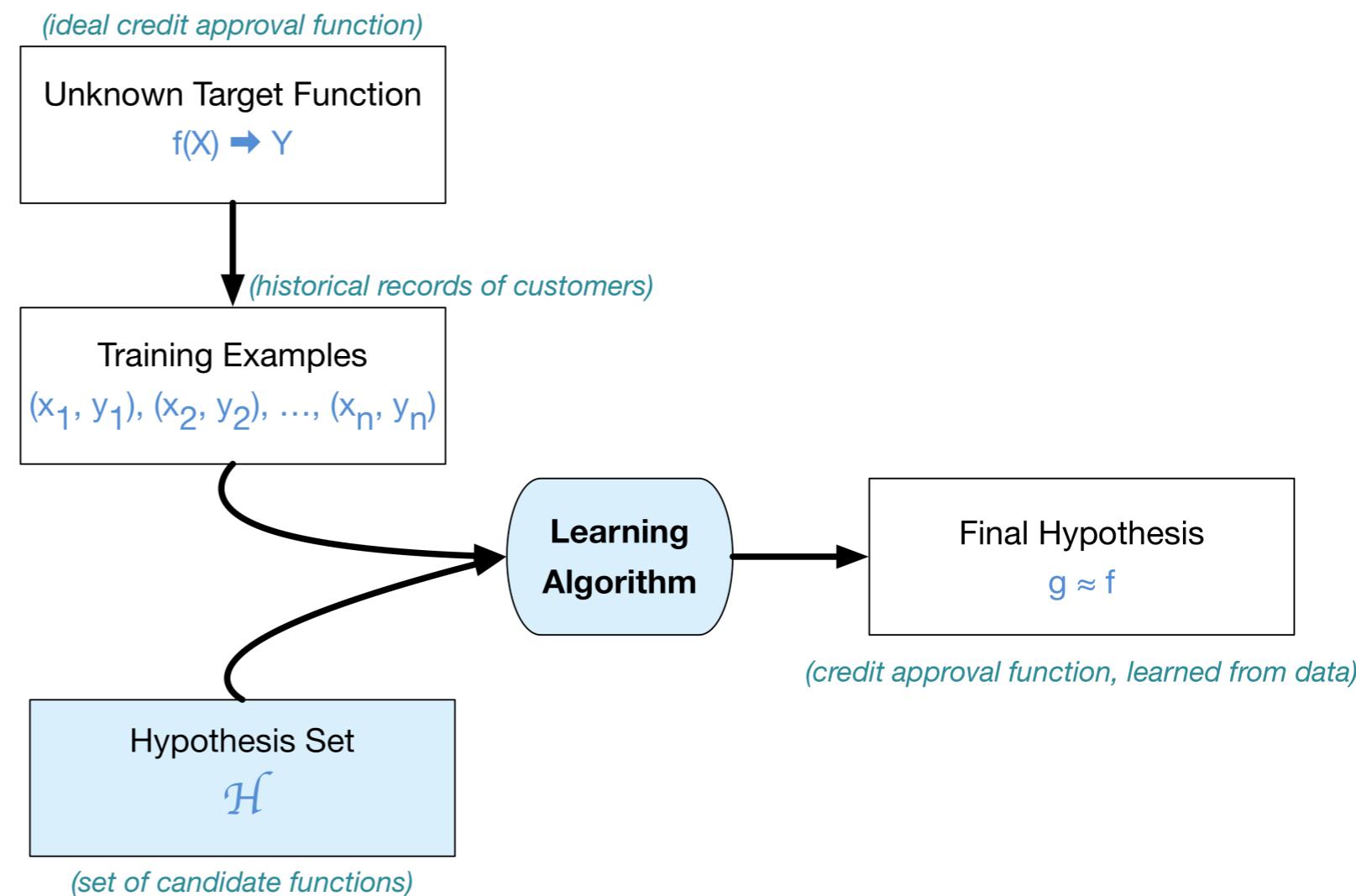
$$\mathcal{H}$$

(set of candidate functions)

Solution components

The solution has two components

- Hypothesis set
- Learning algorithm



5 min. break

Example: perceptron

- Given input $x = (x_1, \dots, x_d)$ (*the d attributes of customer app.*)
- Assign a **weight** w_i for each attribute value x_i . and choose some **threshold** value

Approve credit if $\sum_{i=1}^d w_i x_i \geq \text{threshold}$

Deny credit if $\sum_{i=1}^d w_i x_i < \text{threshold}$

Writing as function $h(x)$

$$h(x) = \text{sign} \left(\left(\sum_{i=1}^d w_i x_i \right) - \text{threshold} \right)$$

$$\text{sign}(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ -1 & \text{if } x < 0 \end{cases}$$

$$= \text{sign} \left(\left(\sum_{i=1}^d w_i x_i \right) + w_0 \right)$$

(rename "threshold" as w_0)

$$= \text{sign} \left(\left(\sum_{i=1}^d w_i x_i \right) + w_0 x_0 \right)$$

(introduce artificial attribute $x_0=1$)

$$= \text{sign} \left(\sum_{i=0}^d w_i x_i \right)$$

Geometry of perceptron

(shown on board)

Exercise

Instructions: ~1 minute to think/answer on your own; then discuss with neighbors; then I will call on one of you

Perceptron: $h(x) = \text{sign} \left(\sum_{i=0}^d w_i x_i \right)$

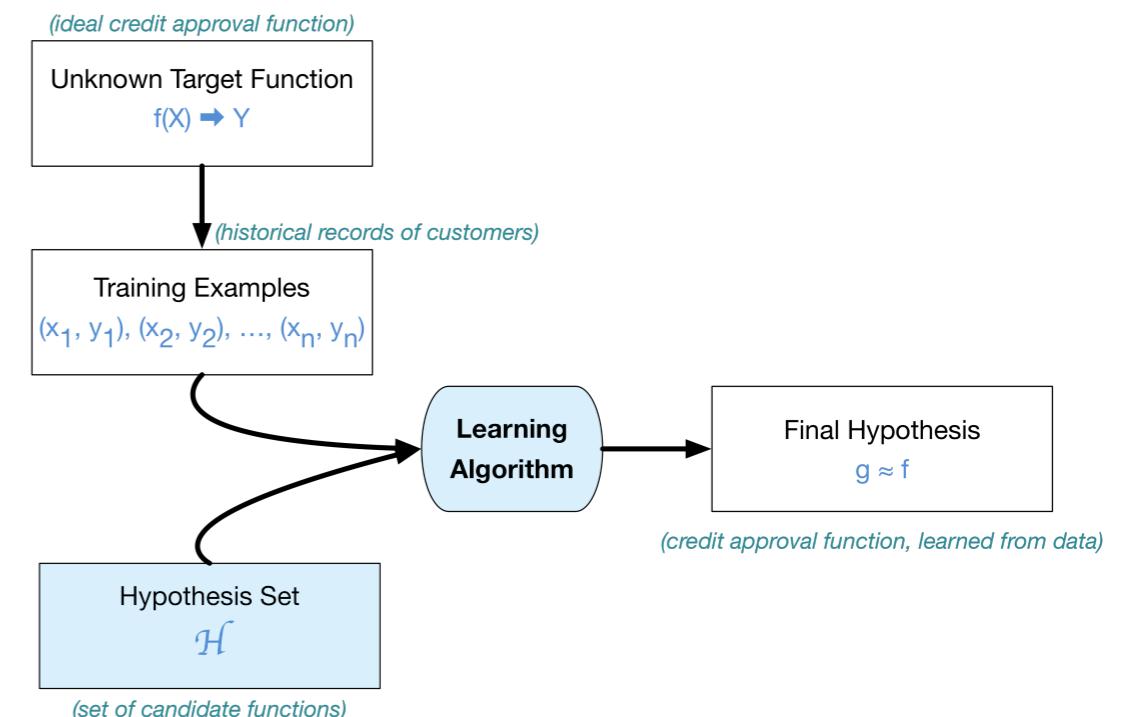
Suppose $x = (x_1, x_2)$ where x_1 is current debt (in thousands of dollars) and x_2 is current salary (in thousands of dollars).

The weights are $w_0=-15$ and $w_1=-2$, and $w_2=1$.

Suppose Bob makes \$40K/year and has \$15K in debt. Would he be approved for credit?

Perceptron = hypothesis set

- Solution components:
 - Hypothesis set
 - Learning algorithm
- **Perception hypothesis set:** what is it?



\mathcal{H} = all possible settings of weights w_0, w_1, \dots, w_d .
(infinite set)

Exercise

Instructions: ~1 minute to think/answer on your own; then discuss with neighbors; then I will call on one of you

Suppose you used a perceptron to classify email as spam or ham. We will interpret +1 to mean that the email is spam and -1 to mean ham.

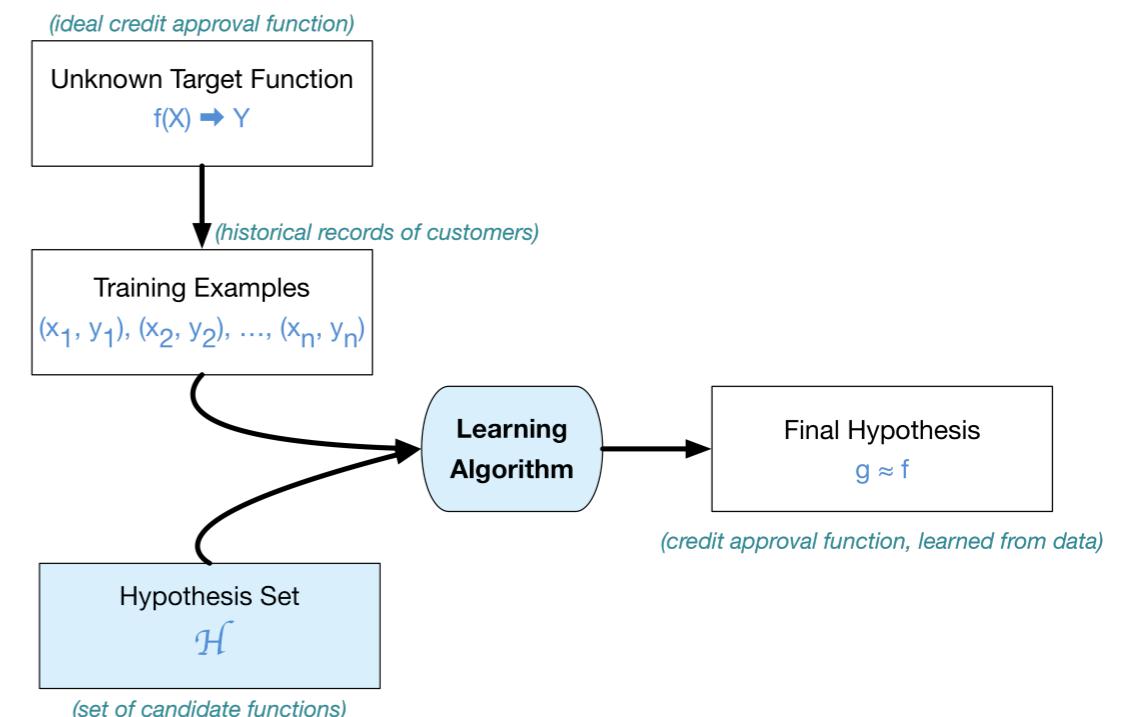
Attributes are word occurrences. For example, x_{Viagra} is 1 if the word "Viagra" appears in the email and 0 if not. We have a feature for every word in language (!).

- Q: Can you think up some words that should receive a positive weight? Negative weight?
- Q: How would an empty email (no words) be classified by the perceptron?

Perceptron learning

- Solution components:

- Hypothesis set
- Learning algorithm



- **Learning algorithm** = procedure for choosing the "best" hypothesis g from among the set \mathcal{H}
- **Perceptron learning algorithm** = procedure for picking values for weights w_0, w_1, \dots, w_d .

Algorithm 1 Algorithm for learning weights of a perceptron

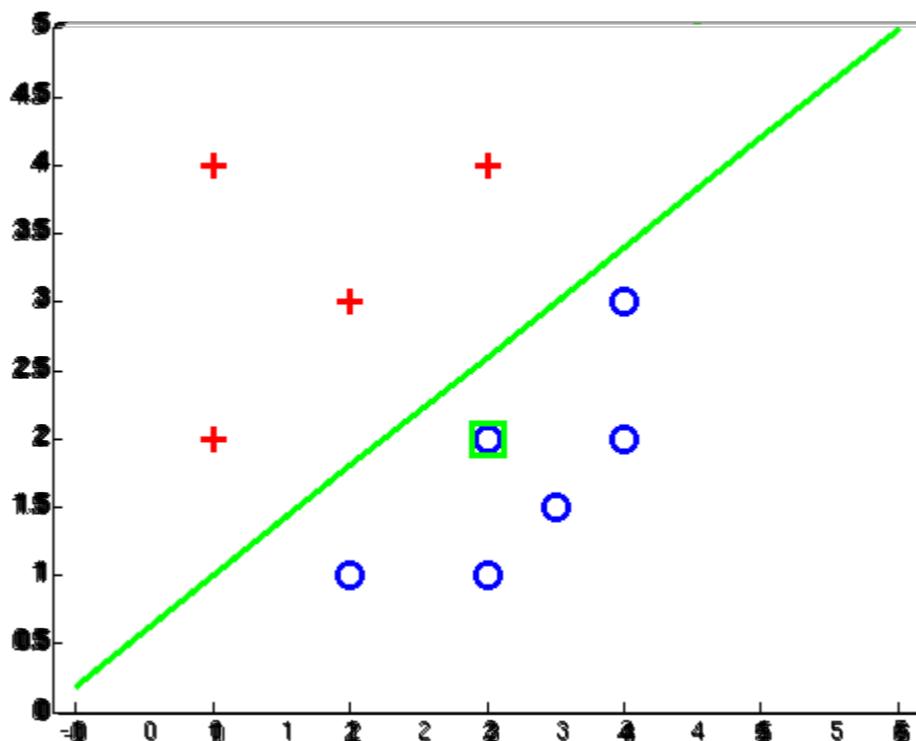
```
1: procedure PERCEPTRONLEARNER(training data)
2:   Training data: collection of  $(x, y)$  pairs where  $x = (x_1, \dots, x_d)$  and  $y = +1$  or  $y = -1$ .
3:   Initialize  $w_0 = w_1 = \dots = w_d = 0$ 
4:   repeat
5:      $\triangleright$  Pick a misclassified example:
6:     Find some  $(\textcolor{blue}{x}, \textcolor{blue}{y})$  pair such that  $h(x) = \text{sign} \left( \sum_{i=0}^d \textcolor{red}{w}_i x_i \right) \neq \textcolor{blue}{y}$ 
7:      $\triangleright$  Update the weight vector:
8:     for  $i = 1$  to  $d$  do
9:       Set  $\textcolor{red}{w}_i \leftarrow \textcolor{red}{w}_i + \textcolor{blue}{y} \times \textcolor{blue}{x}_i$ 
10:      end for
11:    until There are no more misclassified points (or some other stopping criterion)
12:    Return weights  $w_0, \dots, w_d$ .
13: end procedure
```

Intuition:

- Suppose Bob makes \$15K in salary
- Perceptron incorrectly denies Bob credit
 - $h(\text{Bob}) = -1$ but $y = 1$
- Weight update will *increase* the weight on salary, and therefore *increase* $h(\text{Bob})$

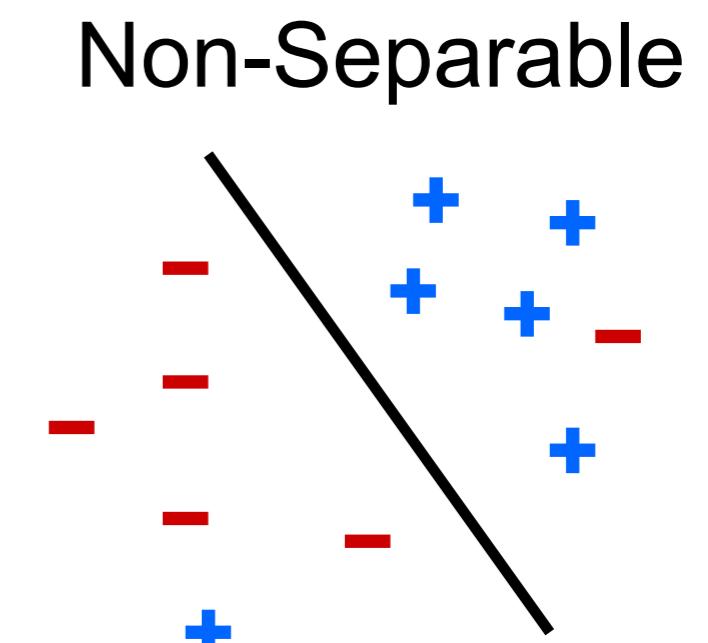
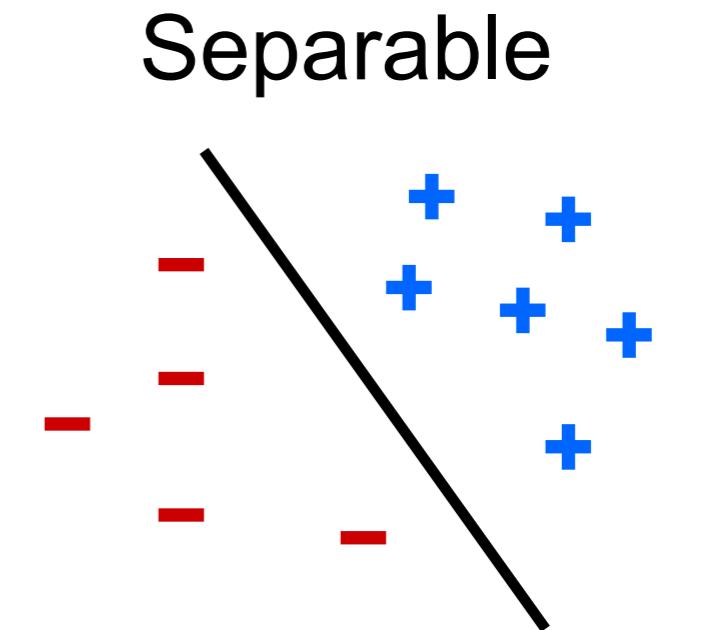
Does this remind you of anything?

Example: Perceptron Learning



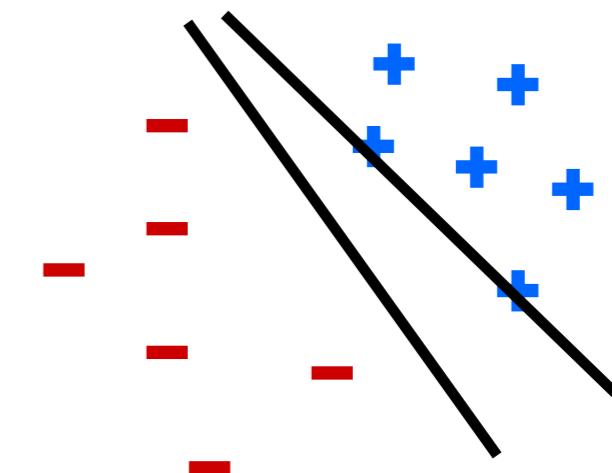
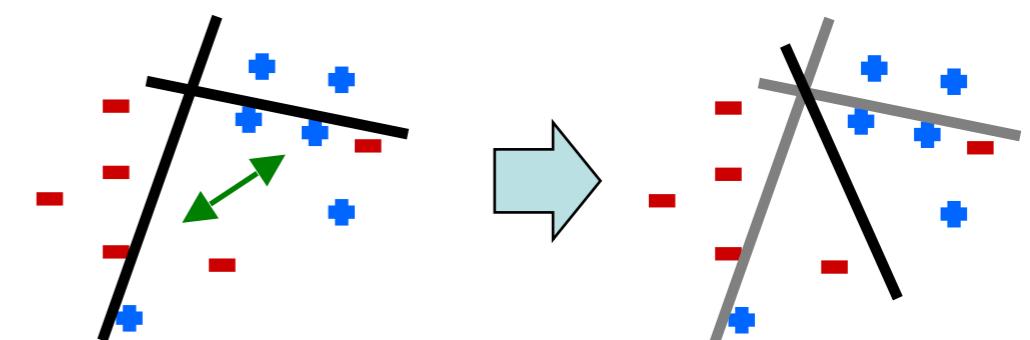
Properties of perceptrons

- **Separable data:** some parameters get the training data perfectly correct
- **Perceptron convergence:** if training data is separable, perceptron will eventually converge (find weights that separate)



Problems with Perceptron

- **Inseparable data:** If the data isn't separable, learning might "thrash"
 - "Pocket algorithm": keep best weights you've seen so far and return this after so many attempts
 - Adding more features can also help
- Sometimes finds a “barely” separating solution.
 - Intuition suggests there are “better” separators.



Essence of machine learning

- A pattern exists
- We cannot pin it down mathematically
- We have data on it

Exercise

Instructions: ~1 minute to think/answer on your own; then discuss with neighbors; then I will call on one of you

Which of the following problems are best suited for Machine Learning?

- A. Classifying numbers into primes and non-primes.
- B. Detecting potential fraud in credit card charges.
- C. Determining the time it would take a falling object to hit the ground.
- D. Determining the optimal cycle for traffic lights in a busy intersection.
- E. More than one above
- F. None of the above

Supervised vs. Unsupervised

- **Supervised**: We are given (x, y) pairs where y is related to x through unknown function $Y = f(X)$. Goal is to find hypothesis g such that $g \approx f$. Types:
 - **Regression**: when y is numerical
 - **Classification**: when y is binary/categorical
- **Unsupervised**: We only get x . Goal is to seek relationships between variables and/or between data points. Typically learn some $g(x)$ that maps x to some simpler representation.
 - **Clustering**: $g(x)$ assigns each x into a group such that "similar" x 's end up in the same group (e.g., cluster movies)

Clustering

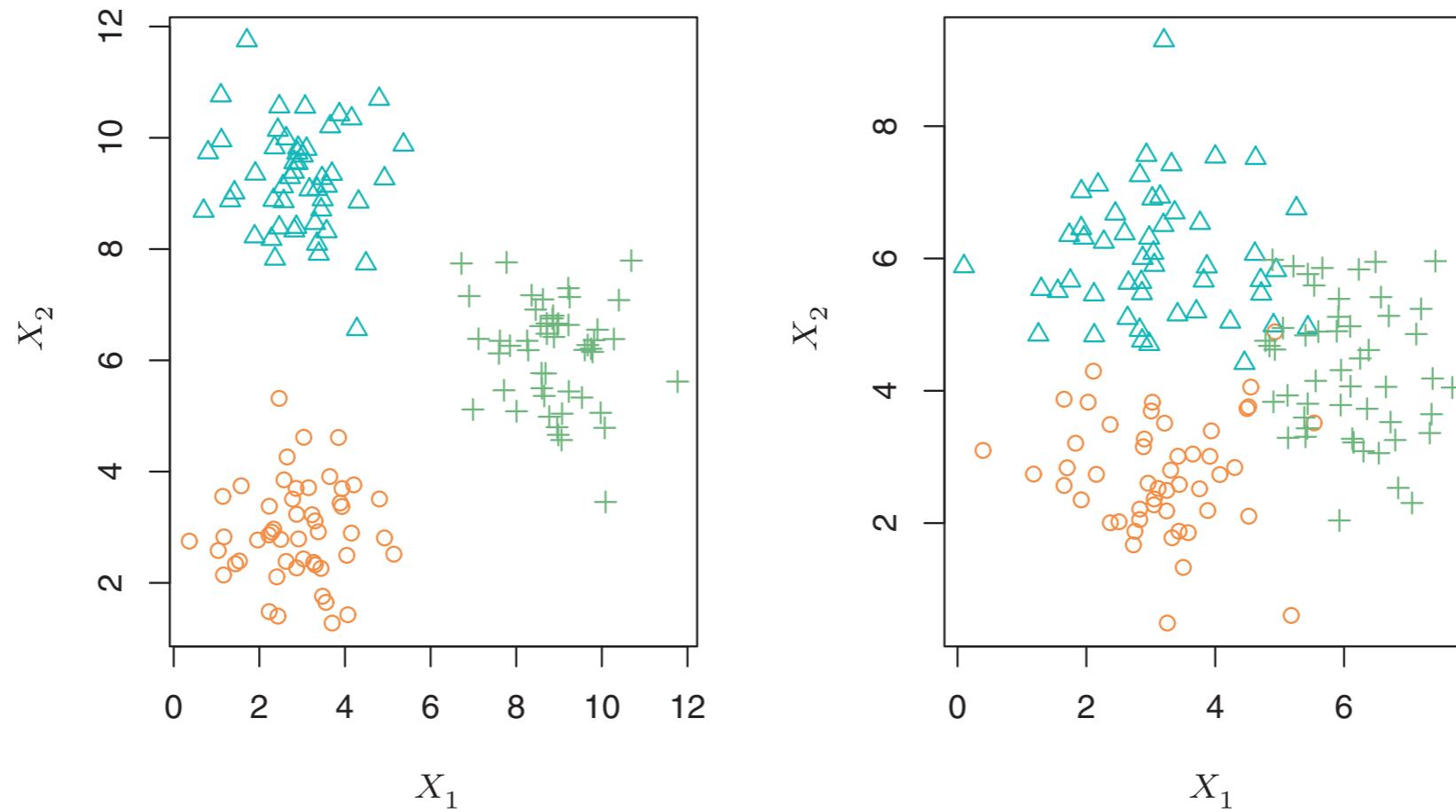


FIGURE 2.8. A clustering data set involving three groups. Each group is shown using a different colored symbol. Left: The three groups are well-separated. In this setting, a clustering approach should successfully identify the three groups. Right: There is some overlap among the groups. Now the clustering task is more challenging.

Parametric vs. Non-parametric

- **Parametric**: means structure of h is fixed except for some parameters.
Hypothesis set = { settings of parameters }
- Perceptron
- Linear regression
- Logistic regression
- Naive Bayes
- **Non-parametric**: form of h is not fixed in advance but adapts with data (typically grows more complex with more data)
 - Decision trees
 - K-nearest neighbors

Generally speaking,
parametric are more
"interpretable" and non-parametric are more "flexible"

Goals of learning

- **Prediction**: use $g(X)$ to predict y on future examples
 - Hypothesis set: anything goes, so long as we can accurate g
- **Inference**: uncover something about the structure of the true (but unknown) function $Y=f(X)$
 - Hypothesis set limited to *interpretable* functions

Generally speaking,
Machine learning focuses on prediction;
Statistics focuses on inference