UNIVERSITY AT ALBANY
STATE UNIVERSITY OF NEW YORK

CSI 436/536 (Spring 2026)

Machine Learning

Lecture 7: Linear Classifier

Chong Liu

Department of Computer Science

Feb 16, 2026

Announcement

- HW1 due today.

- HW 2 will be released later today.

Recap: evaluation criteria

- Confusion matrix of binary classification
  - True Positive, False Positive, True Negative, False Negative

- Performance metrics
  - Accuracy
    - Very popular!
  - Precision / Recall / f1 score
  - AUC (Area Under Curve)
    - The larger area, the better performance

Today

- Feature transformation
- Problem of overfitting
- Data splitting methods:
  - Holdout
  - Cross validation
- Key questions about learning linear classifiers
- Perceptron algorithm

How to learn a LINEAR classifier in a non-linearly separable case?

- Training data:

$$(x_1, y_1), ..., (x_n, y_n) \in \mathcal{X} \times \mathcal{Y}$$

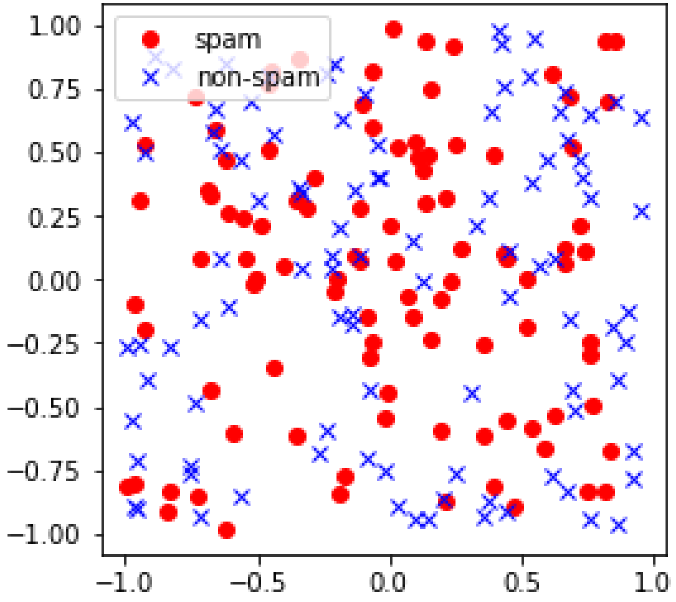
- Solving the following optimization problem:

$$\min_{w \in \mathbb{R}^d} \mathrm{Error}(w) = \frac{1}{n} \sum_{i=1}^{n} \mathbf{1}(h_w(x_i) \neq y_i)$$

- Learning: Find the linear classifier that makes **the smallest number of mistakes** on the training data.

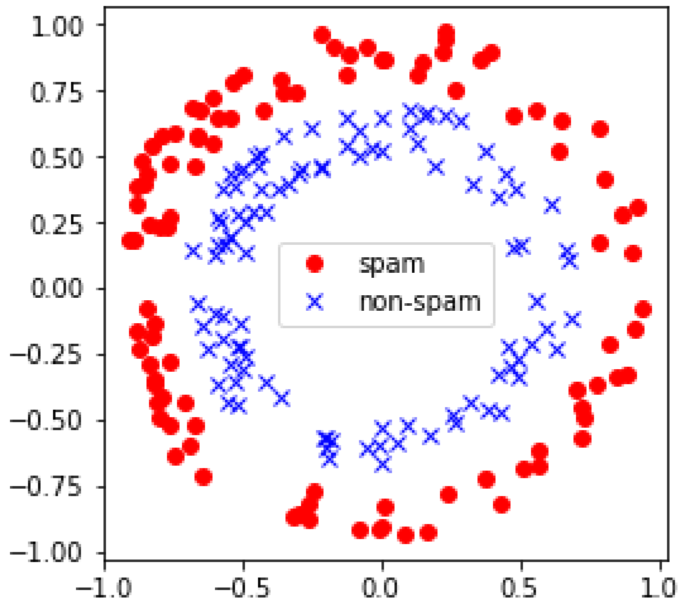
What happens if the linear classifier with the smallest number of mistakes still makes a mistake 49% of the time?

Case 1:



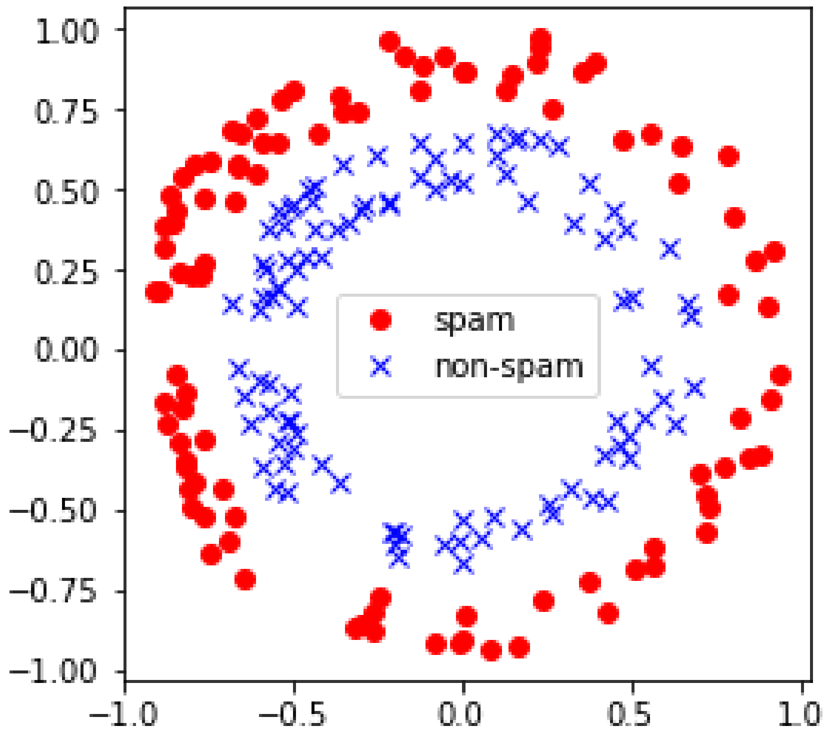
There is no information about the label in the features.
No classifier can do well.

Case 2:



There are some nonlinear classifier that works. But no linear classifiers will do better than random.

Example: Feature transformation

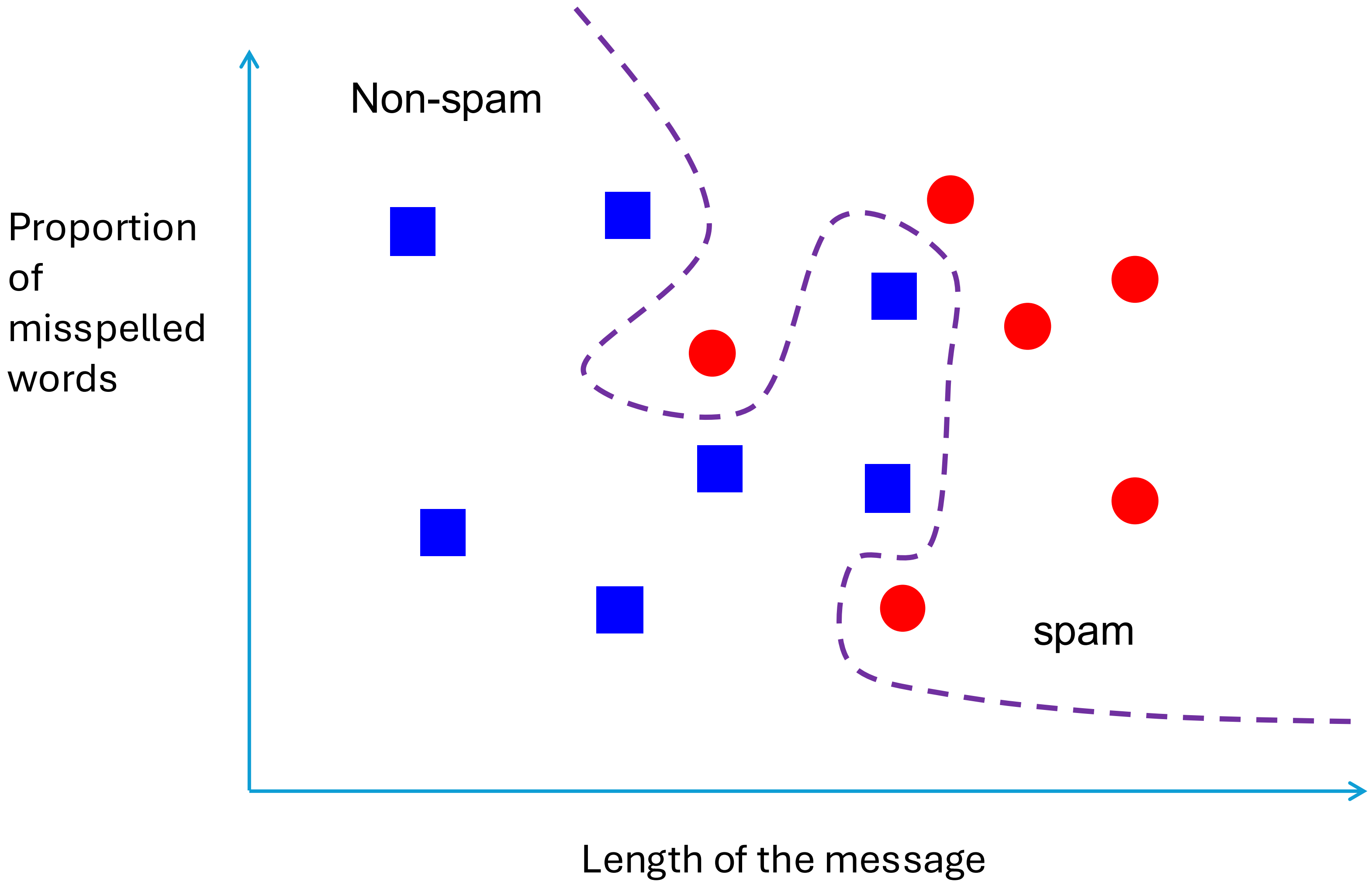


What we can do:

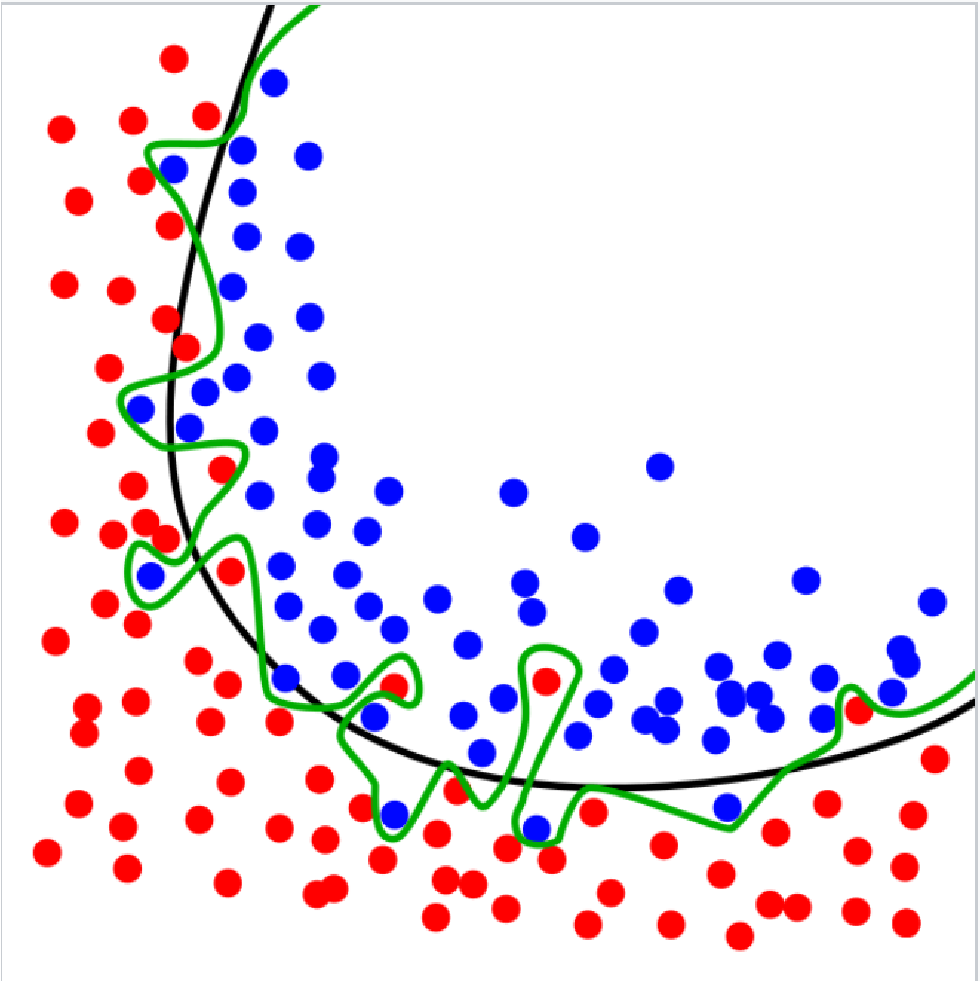
$$(\tilde{x_1}, \tilde{x_2}) = \left(\sqrt{x_1^2 + x_2^2}, \arctan(x_2/x_1)\right)$$

In the redefined space, the two classes are now linearly separable.

Non-linear decision boundary!



The problem of Overfitting

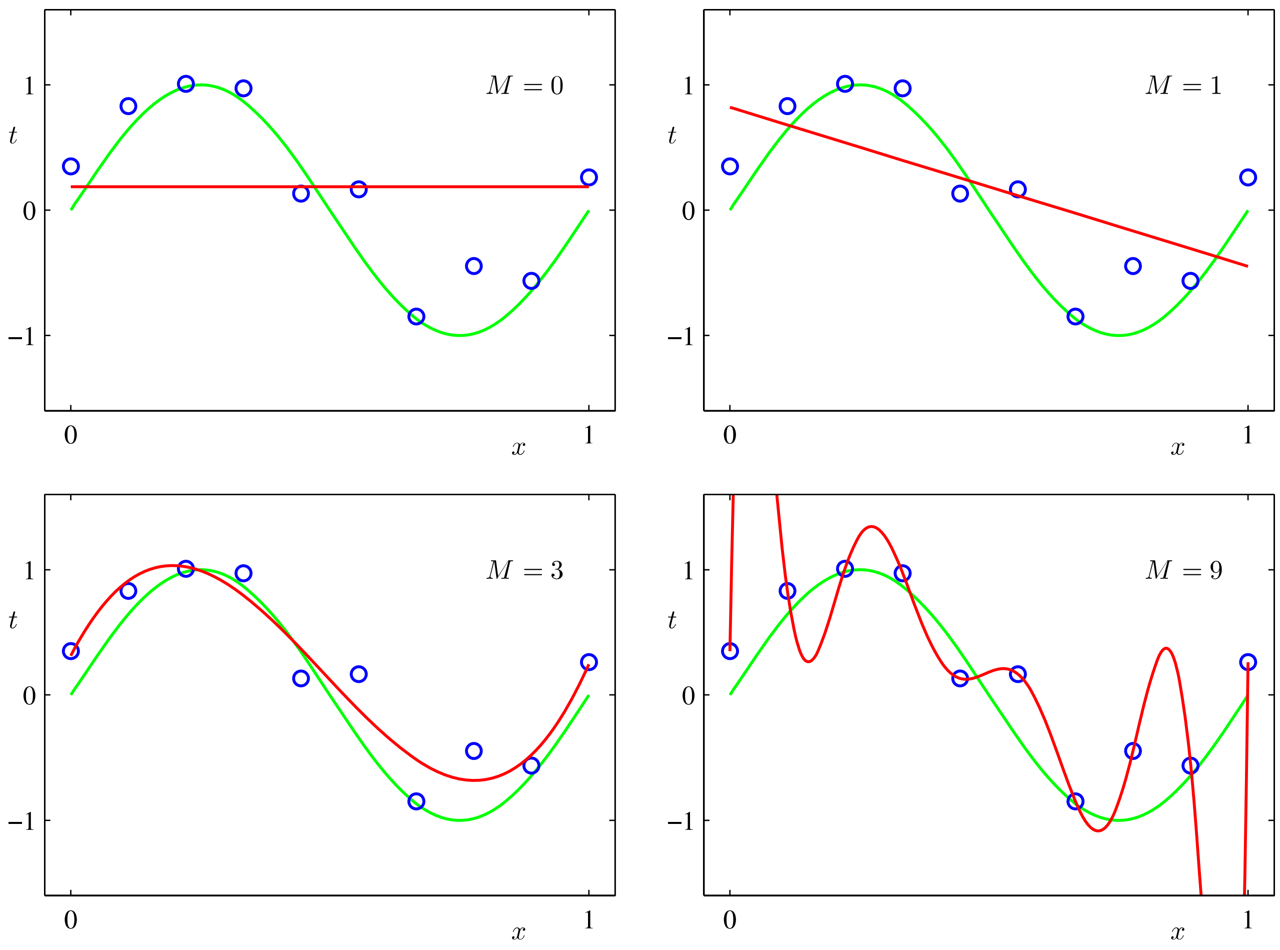


The green line represents an overfitted model.

1. Best follows the training data
2. Too dependent on training data
3. More likely to fail (higher error rate) than black line on new unseen test data

Discussion: examples of overfitting in our learning as human beings?

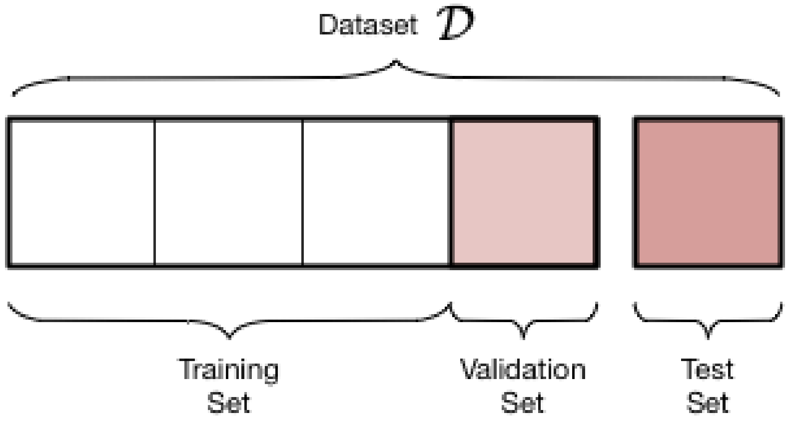
Another example of overfitting in the problem of “Curve Fitting”



Fundamental question in ML:
The learner only sees the "training data" but ideally wants to do well on "new data"!

- The problem of generalization (from training to test).
- All performance metrics we learned before should be calculated on the new test data.
- The issue is that we don't have access to new test data...

Data splitting methods: Holdout

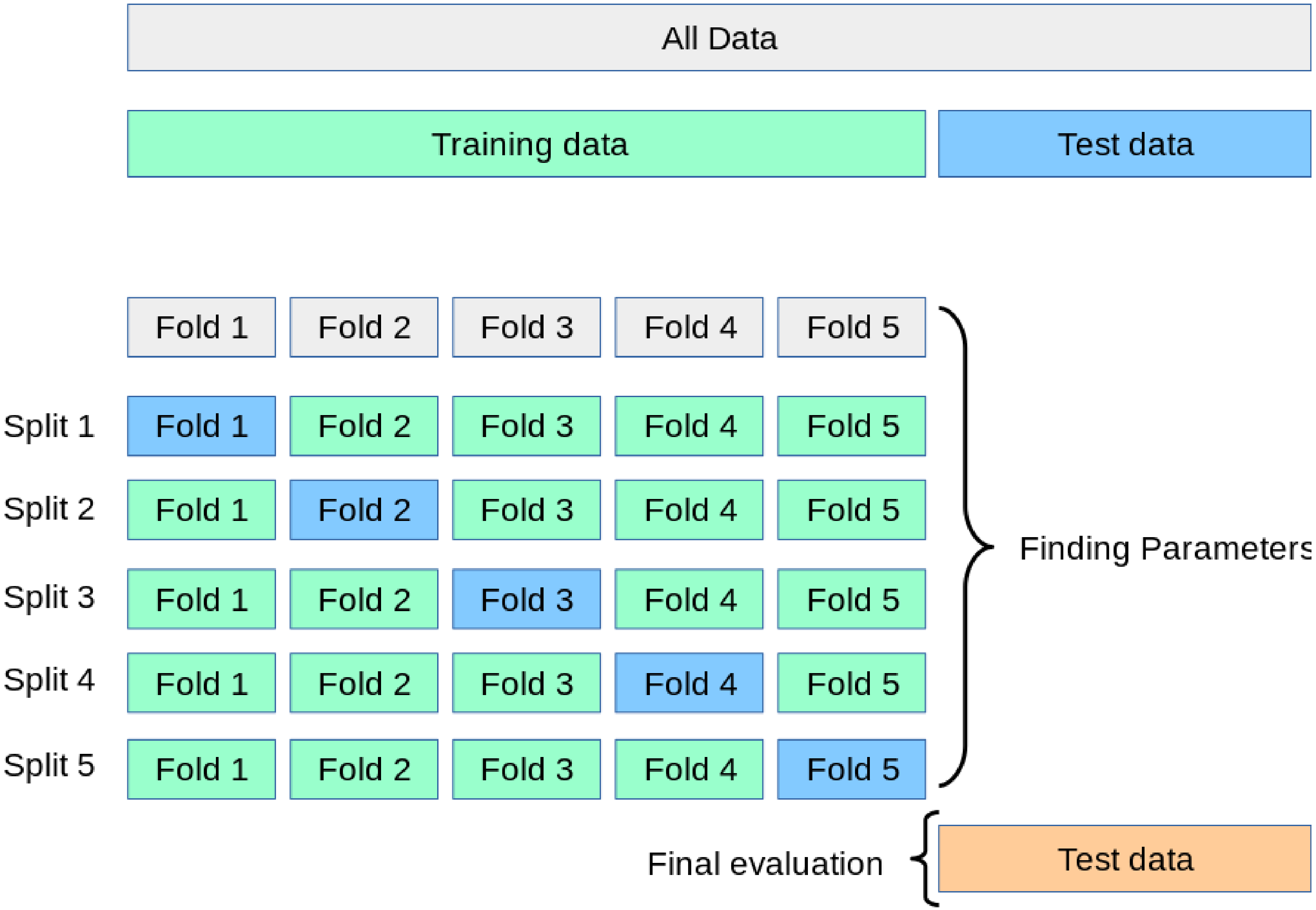


Validation set is used for hyperparameter search (also known as model-selection):

- choosing decision tree vs. linear classifier
- Select features, tune hyperparameters

Test set is used only once to report the final results.

Data splitting: 5-fold cross validation



https://scikit-learn.org/stable/modules/cross_validation.html

Checkpoint

- Deal with linearly non-separatable cases:
  - Feature transformation
  - Non-linear decision boundary
- Problem of overfitting
  - Too dependent on training data and may not good on test data
- Goal of machine learning
  - Learning == search for the best hypothesis in a hypothesis class
  - Ideally, we want to minimize “test error”
    - But all we have access to is the training data
    - We have a practical way --- data-splitting --- to evaluate a classifier

Recap: Linear classifier

- Take input feature vector
  - $\text{Score}(x) = w_0 + w_1x_1 + w_2x_2 + w_3x_3 + w_4x_4$
  - $x_1 = 1(\text{has hyperlinks})$
  - $x_2 = 1(\text{on contact list})$
  - $x_3 = \text{proportion of misspelling}$
  - $x_4 = \text{length}$
- Let label space be $\{-1,1\}$
- Make prediction by thresholding a weighted average of the feature vector at 0:
  - $h_w(x) = \begin{cases} 1, \text{if Score}(x) \geq 0 \\ -1, \text{if Score}(x) < 0 \end{cases}$

Key questions about learning linear classifiers

$$\min_{w \in \mathbb{R}^d} \mathrm{Error}(w) = \frac{1}{n} \sum_{i=1}^{n} \mathbf{1}(h_w(x_i) \neq y_i)$$

1. Does a solution exist?
2. Is the solution unique?
3. Is there an efficient algorithm to find it?
4. How does it work on data points *not* used for training?
5. What are the assumptions needed?

Perceptron algorithm (Rosenblatt, 1957) for linear classifier

THE MARK I PERCEPTRON

NEW NAVY DEVICE LEARNS BY DOING

Psychologist Shows Embryo of Computer Designed to Read and Grow Wiser

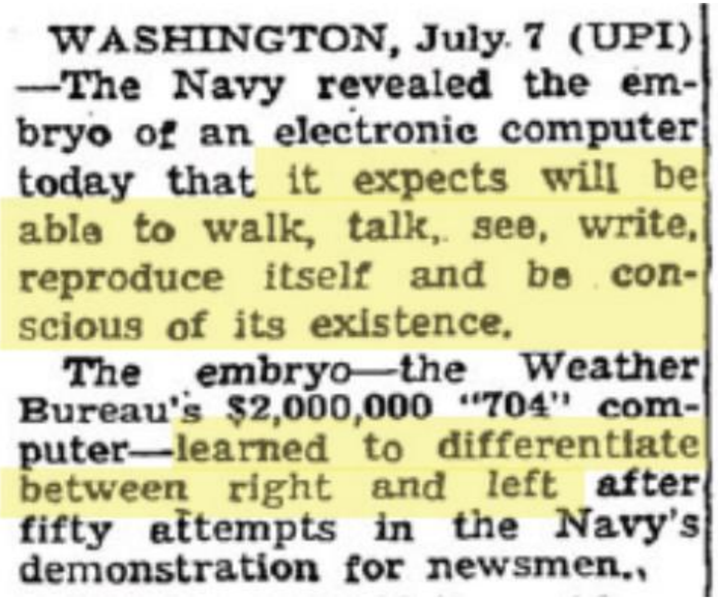
WASHINGTON, July 7 (UPI) —The Navy revealed the embryo of an electronic computer today that it expects will be able to walk, talk, see, write, reproduce itself and be conscious of its existence.

The embryo—the Weather Bureau's $2,000,000 "704" computer—learned to differentiate between right and left after fifty attempts in the Navy's demonstration for newsmen.

The service said it would use this principle to build the first of its Perceptron thinking machines that will be able to read and write. It is expected to be finished in about a year at a cost of $100,000.

Dr. Frank Rosenblatt, designer of the Perceptron, conducted the demonstration. He said the machine would be the first device to think as the human brain. As do human beings, Perceptron will make mistakes at first, but will grow wiser as it gains experience, he said.

Dr. Rosenblatt, a research psychologist at the Cornell Aeronautical Laboratory, Buffalo, said Perceptrons might be fired to the planets as mechanical space explorers.



WASHINGTON, July 7 (UPI) —The Navy revealed the embryo of an electronic computer today that it expects will be able to walk, talk, see, write, reproduce itself and be conscious of its existence.

The embryo—the Weather Bureau's $2,000,000 "704" computer—learned to differentiate between right and left after fifty attempts in the Navy's demonstration for newsmen.

The perceptron algorithm takes an arbitrary sequence of inputs, predict on-the-fly, then update the weights if it makes a mistake!

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 $w_{t+1} \leftarrow w_t - x$

In-class exercise

Example: $(-1,2)\ -$

$(1,0)\ +$

$(1,1)\ +$

$(-1,0)\ -$

$(-1,-2)\ -$

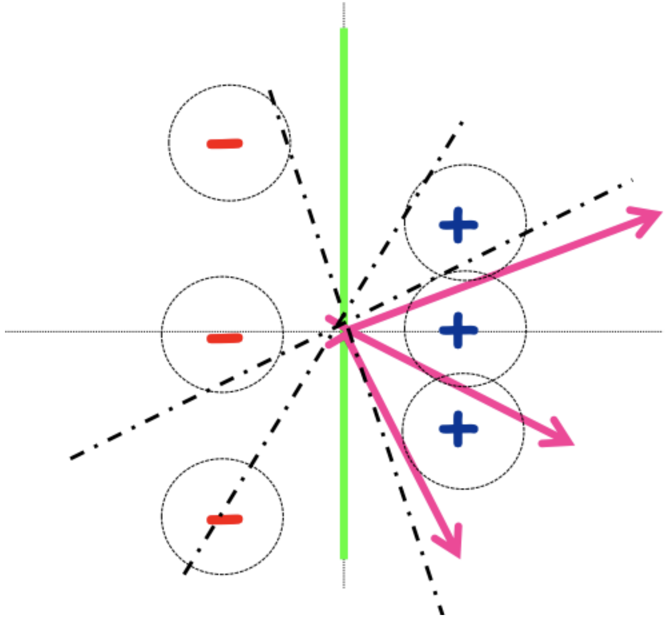
$(1,-1)\ +$

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 $w_{t+1} \leftarrow w_t - x$

Solution

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 $w_{t+1} \leftarrow w_t - x$

$w_1 = (0,0)$

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

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

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